Sun Java™ System

Messaging Server 6
管理ガイド

2005Q1

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

Part No: 819-1054

目 次

表目次

図目次

まえがき

このガイドでは、Sun Java™ System Messaging Server と付属するソフトウェアコンポーネントの管理方法について説明します。Messaging Server は、オープンインターネット規格を使用したあらゆる規模の企業およびメッセージングホストの電子メールニーズに対応する、強力で柔軟性のあるクロスプラットフォームのソリューションを提供します。

この章には、以下の項目があります。

対象読者

このマニュアルは、管理するサイトで、Messaging Server の管理や実装に対し、責任ある立場の方を対象としています。『Sun Java System Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) も読んでおく必要があります。

予備知識

このマニュアルは、Messaging Server ソフトウェアのインストールに関する責任者を対象とし、次の一般的な知識を持っていることを前提にしています。

- インターネットおよび WWW (ワールドワイドウェブ)
- Messaging Server のプロトコル
- Sun Java System Administration Server
- Sun Java System Directory Server および LDAP
- Sun Java System Console
- 以下のプラットフォームのシステム管理とネットワーキング
- 一般的な配備アーキテクチャ

このマニュアルの構成

このマニュアルは、次の章および付録から構成されています。

表 1　このマニュアルの構成

章	説明
まえがき	このマニュアルについての一般情報。
第 1 章「インストール後の作業とレイアウト」	正常に機能している Messaging Server を使用できるようにするための手順を説明します。
第 2 章「Sun Java Systems Messaging Server へのアップグレード」	Messaging Server 5.2 から Messaging Server 6 2005Q1 へのアップグレード方法について説明します。
第 3 章「高可用性の構成」	高可用性のクラスタリングソフトウェア Veritas Cluster Server および Sun Cluster を、Messaging Server とともに使用するための設定方法について説明します。
第 4 章「一般的なメッセージング機能の設定」	一般的な Messaging Server タスクについて説明します。
第 5 章「POP、IMAP、および HTTP サービスの設定」	Sun ONE Console またはコマンド行ユーティリティを使って、POP、IMAP および HTTP サービスをサポートするようにサーバーを構成する方法について説明します。
第 6 章「シングルサインオン (SSO) の有効化」	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。
第 7 章「マルチプレクササービスの設定および管理」	標準メールプロトコル (POP、IMAP、および SMTP) 向けの Messaging Multiplexor (MMP) と、Messenger Express Web インタフェース向けの Messenger Express Multiplexor について説明します。

表 1　このマニュアルの構成 (続き)

章	説明
第 8 章「MTA の概念」	MTA の概念について説明します。
第 9 章「MTA のアドレス変換とルーティング」	MTA のアドレス変換とルーティングについて説明します。
第 10 章「MTA サービスと設定について」	一般的な MTA のサービスと設定について説明します。
第 11 章「書き換えルールの設定」	imta.cnf ファイル内で書き換えルールを設定する方法について説明します。
第 12 章「チャネル定義を設定する」	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。
第 13 章「定義済みチャネルを使用する」	MTA の定義済みチャネルの使い方を説明します。
第 14 章「スパムとウィルスのフィルタ処理プログラムを Messaging Server に統合する」	Messaging Server を使用して、スパムおよびウィルスフィルタリングソフトウェアを統合および設定する方法について説明します。
第 15 章「LMTP 配信」	LMTP の運用と開発について説明します。
第 16 章「不在メッセージの自動返信」	不在時の自動返信機構について説明します。
第 17 章「メールのフィルタリングとアクセス制御」	メールをソース (差出人、IP アドレスなど) やヘッダー文字列に基づいてフィルタリングする方法について説明します。
第 18 章「メッセージストアを管理する」	メッセージストアとその管理インタフェースについて説明します。
第 19 章「セキュリティとアクセス制御を設定する」	Messaging Server に対するセキュリティおよびアクセス制御の設定方法について説明します。
第 20 章「Communications Express メールでの S/MIME の管理」	S/MIME を管理する方法について説明します。
第 21 章「ログの管理」	Messaging Server のロギング機能について説明します。
第 22 章「MTA のトラブルシューティング」	MTA の障害追跡のための共通のツール、メソッド、および手続きについて説明します。
第 23 章「Messaging Server を監視する」	Messaging Server の監視について説明します。
付録 A「SNMP サポート」	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。
付録 B「Messaging Server の Event Notification Service (ENS) を管理する」	Messaging Server 内で Event Notification Service Publisher (ENS Publisher) を有効にして Event Notification Service (ENS) を管理する方法について説明します。

表 1　このマニュアルの構成 (続き)

章	説明
付録 C「コンソールインタフェースを使用してメールユーザーとメーリングリストを管理する (推奨しない)」	推奨しません。
付録 D「ショートメッセージサービス (SMS)」	Short Message Service (SMS) を実装する方法について説明します。
付録 E「インストールワークシート」	インストールを計画するためのワークシートを示します。
用語集	このマニュアルセットで使用されている用語の完全なリストです。
索引	索引

このマニュアルの表記ルール

この節の表は、このマニュアルの表記ルールを示します。

書体の表記ルール

次の表は、このマニュアルの書体の表記の種類を示します。

表 2　書体の表記ルール

書体	意味	例
`AaBbCc123` (モノスペース)	コンピュータ画面に表示されるテキスト、またはユーザーが入力するテキストを表します。API および言語要素、HTML タグ、Web サイトの URL、コマンド名、ファイル名、ディレクトリパス名、画面上のコンピュータ出力、サンプルコードを示します。	`.login` ファイルを編集します。 `ls -a` を使用してすべてのファイルを表示します。 `% You have mail.`
`AaBbCc123` (太字のモノスペース)	コード例またはほかの画面上のコンピュータ出力の中でユーザーが入力するテキストを表します。	**`% su`** `Password:`
AaBbCc123 (斜体)	実際の名前または値 (変数など) に置き換えられるコマンドまたはパス名の可変部分を表します。	ファイルは *msg_svr_base*`/bin` ディレクトリにあります。 これらを *class* オプションと呼びます。

記号

次の表は、このマニュアルの記号の表記ルールを示します。

表 3　記号の表記ルール

記号	説明	例	意味
[ ]	省略可能なコマンドオプション。	`ls [-l]`	`-l` オプションの指定は必須ではありません。
{ | }	必須のコマンドオプションの選択肢のセットが含まれます。	`-d {y|n}`	`-d` オプションとともに、`y` 引数または `n` 引数を指定する必要があります。
-	同時に実行する複数のキーストロークを結び付けます。	Control-A	コントロールキーを押しながら A キーを押します。
+	連続する複数のキーストロークを結び付けます。	Ctrl + A + N	コントロールキーを押して離し、続いて次のキーを押します。
>	グラフィカルユーザーインタフェースのメニューの選択を表示します。	「ファイル」>「新規」>「テンプレート」	「ファイル」メニューから「新規」を選択します。「新規」メニューから「テンプレート」を選択します。

デフォルトのパスとファイル名

次の表は、このマニュアルのデフォルトのパスとファイル名を示します。

表 4　デフォルトのパスとファイル名

用語	説明
msg_svr_base	Messaging Server のベースインストールディレクトリを示します。*msg_svr_base* インストールのデフォルト値は、次のとおりです。 Solaris™ システム : `/opt/SUNWmsgsr` Linux システム : `/opt/sun/messaging`

コマンド行プロンプト

このマニュアルの多くの例では、コマンド行プロンプト (たとえば、C シェルの %、Korn/Bourne シェルの $ など) が表示されていません。使用しているオペレーティングシステムによって、コマンド行プロンプトが異なるためです。ただし、特に補足されていないかぎり、コマンドは本書で示すとおりに入力してください。

関連マニュアル

http://docs.sun.comSM Web サイトでは、Sun が提供しているオンラインマニュアルを参照することができます。アーカイブをブラウズすることも、特定のマニュアルのタイトルまたは主題を検索することもできます。

Messaging Server のマニュアル

次の URL を使用すると、Messaging Server のすべてのマニュアルを参照できます。

http://docs.sun.com/app/docs/coll/MessagingServer_05q1?l=ja

次のマニュアルが使用できます。

- 『Sun Java SystemMessaging Server リリースノート』 (http://docs.sun.com/doc/819-0150?l=ja)
- 『Sun Java System Messaging Server 管理ガイド』 (http://docs.sun.com/doc/819-1054?l=ja)
- 『Sun Java System Messaging Server Administration Reference』 (http://docs.sun.com/doc/819-0106)
- 『Sun Java System Messaging Server MTA Developer's Reference』 (http://docs.sun.com/doc/819-0107)
- 『Sun Java System Messenger Express Customization Guide』 (http://docs.sun.com/doc/819-0108)

Messaging Server 製品群には、Sun Java™ System Console、Directory Server、管理サーバーなどほかの製品が含まれています。Sun ONE Messaging Server 製品およびその他の製品のマニュアルは、次の URL で参照できます。

http://docs.sun.com/app/docs/prod/entsys?l=ja#hic

ソフトウェアマニュアルの他にも、Messaging Server ソフトウェアフォーラムで、特定の Messaging Server 製品に関する質問について、技術的なヘルプを参照してください。フォーラムには、以下の URL をご利用ください。

http://swforum.sun.com/jive/forum.jspa?forumID=15

Communications Services のマニュアル

次の URL を使用すると、Communications Services 製品のすべてのマニュアルを参照できます。

http://docs.sun.com/app/docs/coll/MessagingServer_05q1?l=ja

次のマニュアルが使用できます。

- 『Sun Java System Messenger Express Customization Guide』 (http://docs.sun.com/doc/819-0108)
- 『Sun Java System Communications Services Delegated Administrator 管理ガイド』 (http://docs.sun.com/doc/819-1101?l=ja)
- 『Sun Java System Communications Services 配備計画ガイド』 (http://docs.sun.com/doc/819-1069?l=ja)
- 『Sun Java System Communcations Services Schema Reference』 (http://docs.sun.com/doc/819-0113)
- 『Sun Java System Communications Services Schema Migration Guide』 (http://docs.sun.com/doc/819-0112)
- 『Sun Java System Event Notification Service Guide』 (http://docs.sun.com/doc/819-0109)
- 『Sun Java System Communications Express 管理ガイド』 (http://docs.sun.com/doc/819-1065?l=ja)
- 『Sun Java System Communications Express Customization Guide』 (http://docs.sun.com/doc/819-0116)

オンラインでこのマニュアルを入手するには

『Messaging Server 管理ガイド』は、PDF および HTML 形式でオンラインで入手できます。以下の URL をご利用ください。

(http://docs.sun.com/doc/819-1054?l=ja)

Sun のオンラインリソースへのアクセス

製品のダウンロード、プロフェッショナルサービス、パッチとサポート、詳細な開発者関連情報などについては、次のサイトを参照してください。

- ダウンロードセンター
  http://wwws.sun.com/software/download/
- プロフェッショナルサービス
  http://www.sun.com/service/sunps/sps.html
- Sun Enterprise Service、Solaris パッチ、サポート
  http://sunsolve.sun.com/
- 開発者関連情報
  http://developers.sun.com/prodtech/index.html

Sun テクニカルサポートへの連絡

この製品に関して、製品マニュアルにない技術的な質問がある場合は、
http://www.sun.com/service/contacting まで問い合せてください。

関連する外部 Web サイトの参照

このマニュアルで述べる外部 Web サイトの可用性について Sun は責任を負いません。こうしたサイトやリソース上またはこれらを通じて利用できるコンテンツ、広告、製品、その他の資料について Sun は推奨しているわけではなく、Sun はいかなる責任も負いません。こうしたサイトやリソース上で、またはこれらを経由して利用できるコンテンツ、製品、サービスを利用または信頼したことに伴って発生した (あるいは発生したと主張される) いかなる損害や損失についても、Sun は直接的にも間接的にも、一切の責任を負いません。

Sun では、お客様のご意見を歓迎します。

Sun では、マニュアルをよりよいものにするため、お客様のご意見やご提案をお待ちしております。

http://docs.sun.com にアクセスして、「コメントの送信」をクリックしてください。オンラインフォームに、マニュアルのタイトルと Part No. を記入してください。Part No. は 7 桁または 9 桁の数字で、マニュアルの表紙またはドキュメントの上部にあります。たとえば、このマニュアルのタイトルは『Sun Java System Messaging Server 6 2005Q1 管理ガイド』で、Part No. は 819-1054 です。

Sun では、お客様のご意見を歓迎します。

第 1 章

インストール後の作業とレイアウト

この章では、読者があらかじめ『Sun Java System Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) を読み、Sun Java™ Enterprise System インストーラを使用して Messaging Server をインストールしていることを前提にしています (詳細は、『Sun Java Enterprise System インストールガイド』(http://docs.sun.com/doc/819-0808?l=ja) を参照)。次の作業を実行すると、Messaging Server を機能させることができます。さらに配備をカスタマイズすることも、ユーザーとグループのプロビジョニングや移行を行うこともできます。カスタマイズについては、このマニュアルの後続の章で説明します。プロビジョニングについては、『Sun Java System Communications Services Delegated Administrator 管理ガイド』(http://docs.sun.com/doc/819-1101?l=ja) を参照してください。

この章には、以下の節があります。

UNIX システムのユーザーとグループを作成するには

システムユーザーが特定のサーバープロセスを実行するとき、そのプロセスを実行するための適切なアクセス権を持つように、ユーザーに権限が付与されていることが必要です。

まず、すべての Sun Java System サーバーで有効な 1 つのシステムユーザーアカウントとグループを設定します。次に、そのユーザーが所有するディレクトリとファイルに対し、アクセス権を設定します。設定のための手順を、次に示します。

注 セキュリティの理由から、配備によっては、サーバーごとに異なるシステム管理者を設定することが望ましい場合もあります。そのためには、サーバーごとに異なるシステムユーザーとグループを作成します。たとえば、Messaging Server と Web Server にそれぞれ異なるシステムユーザーを設定し、Messaging Server のシステム管理者は Web Server を管理できないようにします。

1. スーパーユーザーとしてログインします。
2. システムユーザーが所属するグループを作成します。以下の例では、`mailsrv` グループが作成されます。

```
# groupadd mail
```

3. システムユーザーを作成して、作成したグループと関連付けます。さらに、そのユーザーのパスワードを設定します。次の例では、ユーザー `mail` が作成され、`mailsrv` グループと関連付けられます。

```
# useradd -g mail mailsrv
```

`useradd` コマンドと `usermod` コマンドは `/usr/sbin` にあります。詳細については、UNIX のマニュアルページを参照してください。

4. 作成したシステムグループにユーザーが追加されていることを確認するために、/etc/group ファイルと /etc/passwd ファイルをチェックすることが必要になる場合もあります。

注 Messaging Server のインストール前に UNIX のシステムユーザーやグループを設定しない場合は、53 ページの「Messaging Server の初期実行時設定を作成するには」の実行時に設定することができます。

Messaging Server 設定用に Directory Server を準備するには

この節では、Directory Server 設定スクリプト (comm_dssetup.pl) の実行方法について説明します。このスクリプトは、Messaging Server、Calendar Server、または User Management Utility の設定を処理するように LDAP Directory Server を設定します。comm_dssetup.pl スクリプトは、Directory Server に新しいスキーマ、インデックス、および設定データを設定することによって、Directory Server を準備します。Messaging Server や Communications Express を新しくインストールした場合は、このスクリプトを実行する必要があります。Directory Server に依存するコンポーネント製品のいずれかをアップグレードする場合も、最新の comm_dssetup.pl を実行することをお勧めします。

以下の項目について説明します。

- 43 ページの「comm_dssetup.pl の場所」
- 44 ページの「comm_dssetup.pl 要件」
- 45 ページの「comm_dssetup.pl スクリプトを実行するには」

comm_dssetup.pl の場所

旧バージョンの Java Enterprise System では、このユーティリティは Messaging Server および Calendar Server にバンドルされていて、別にインストールする必要はありませんでした。しかし Java Enterprise System 2005Q1 では、このスクリプトは別個にインストール可能な共有コンポーネントとなりました。

comm_dssetup.pl をインストールするには、次のいずれかの方法を選んでください。

- Java Enterprise System インストールプログラムを使用している場合、コンポーネント選択パネルで comm_dssetup.pl を選択します (Selecting Directory Server も、同様に comm_dssetup.pl を自動的に選択する)。

- 旧バージョンの Java Enterprise System からのアップグレードで、Java Enterprise System インストールプログラムを使用していない場合は、次のパッチをダウンロードします。

  Solaris Sparc: 118245、118242

  Solaris x86: 118256、118243

  Linux: 118247 のみ

このようにインストールを実行すると、comm_dssetup.pl は次のディレクトリに含まれます。

Solaris:/opt/SUNWcomds/sbin

Linux: /opt/sun/comms/dssetup/sbin

comm_dssetup.pl 要件

comm_dssetup.pl スクリプトを実行する前に、以下の要件を確認してください。

- comm_dssetup.pl スクリプトを実行する前に、ディレクトリサーバーをインストールして設定する必要があります。
- comm_dssetup.pl スクリプトをスーパーユーザーとして実行します。
- comm_dssetup.pl を実行してから、Messaging Server、Calendar Server、Communications Express、または User Management Utility Initial Runtime Configuration プログラムを実行します。
  - 通常、ある製品 (Calendar Server など) に対してディレクトリサーバーで comm_dssetup.pl を実行する場合、別の製品 (Messaging Server など) に対しては、両方の製品が同じディレクトリサーバーを使用するのであれば、再度実行する必要はありません。ただし、comm_dssetup の実行時に回答した内容を変更する場合は、comm_dssetup を再度実行する必要があります。たとえば、commdirmig を実行した (Sun LDAP Schema 1 から Sun LDAP Schema 2 に移行した) などの理由で、次の Messaging Server 設定には別のユーザー / グループサフィックスを使用したい場合などです。
- comm_dssetup.pl スクリプトは、使用しているディレクトリサーバーマシン上で実行する必要があります。
- ディレクトリサーバーが実行中であることを確認してから、comm_dssetup.pl を実行します。
- 新しいバージョンの Messaging Server をインストールするときは常に、Directory Server マシンで新しいバージョンの comm_dssetup.pl を実行する必要があります。新しいスキーマおよび新しいインデックスが各 Messaging Server ディストリビューションに追加されている場合があります。

- 設定データとユーザーデータおよびグループデータが別々のディレクトリインスタンスに分割されている場合、両方のインスタンスで comm_dssetup スクリプトを実行する必要があります。
- UNIX システムの場合、バージョン互換の問題を避けるために、Perl は Directory Server に付属のバージョン *dir_server_root*/bin/slapd/admin/bin/perl を使用してください。
- comm_dssetup.pl をリモートディレクトリサーバーで実行する場合は次の手順を行います。
  - dssetup.zip ファイルを、*msg_svr_base*/install ディレクトリからリモートディレクトリサーバーにコピーします。このファイルは、/tmp や /var/tmp などのディレクトリにコピーすることもできます。すべての Directory Server マシンに comm_dssetup をインストールする代わりに、この zip ファイルをほかの Directory Server マシンにコピーすることもできます。
  - dssetup.zip ファイルを解凍します (このファイルには comm_dssetup.pl および必須スキーマが含まれている)。
  - リモートディレクトリサーバーで comm_dssetup.pl スクリプトを実行します。
- レプリケートされたディレクトリサーバーを実行する場合は、マスターディレクトリとレプリカディレクトリに対して comm_dssetup.pl スクリプトを必ず実行する必要があります。
- Directory Server 設定スクリプト (comm_dssetup.pl) を実行して Messaging Server の設定用に Directory Server を準備する場合は、991 ページの表 E-3 にインストールパラメータを記入してください。Messaging Server の初期実行時設定では、これらのパラメータの一部が必要になります。

comm_dssetup.pl スクリプトを実行するには

comm_dssetup.pl は、次のどちらかのモードで実行することができます。

- 45 ページの「インタラクティブモード」
- 51 ページの「サイレントモード」

『Sun Java System Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) のインストールワークシートを使用して、回答を記録してください。

インタラクティブモード

引数なしで comm_dssetup.pl を実行すると、以下のように質問されます。

1. 概要

```
# perl comm_dssetup.pl

Welcome to the Directory Server preparation tool for Java
Enterprise Communications Server.
(Version X.X Revision X.X)

This tool prepares your directory server for Sun Java System
Messaging Server install.

The logfile is /var/tmp/dssetup_YYYYMMDDHHSS

Do you want to continue [y]:
```

続行するには Enter キーを押します。終了するには No と入力します。

2. Directory Server のインストールルート

```
Please enter the full path to the directory where the Java
Enterprise Directory Server was installed.

Directory server root [/var/opt/mps/serverroot]
```

Directory Server マシン上の Directory Server のインストールルートの場所を指定します。Linux では Directory Server のルートの場所が異なることに注意してください。

3. Directory Server インスタンス

```
Please select a directory server instance from the following
list:

[1]   slapd-varrius

Which instance do you want [1]:
```

マシン上に Directory Server reside の複数のインスタンスがある場合は、Messaging Server とともに設定するインスタンスを選びます。

4. Directory Manager 識別名 (DN)

```
Please enter the directory manager DN [cn=Directory Manager]:
Password:
```

Directory Manager DN (`cn=Directory Manager`) は、組織ツリー内のユーザーデータおよびグループデータの責任を持つ管理者です。このスクリプトで指定する Directory Manager DN は、Directory Server インストールおよび Messaging Server インストールで設定する DN と同じものであることを確認します。

5. ユーザーおよびグループの Directory Server

```
Will this directory server be used for users/groups [Yes]:
```

`Yes` と入力した場合は、ユーザー / グループツリーについてさらに質問されます。

`No` と入力した場合、このディレクトリインスタンスは設定データの保存のみに使用されると見なされ、スキーマファイルの更新についての質問に進みます。設定ディレクトリインスタンスに対するこのスクリプトの実行が終了したあと、インストールプロセスに移る前に、ユーザーデータおよびグループデータを保存するディレクトリインスタンスに対してこのスクリプトを実行する必要があります。

6. ユーザーおよびグループベースのサフィックス

```
Please enter the Users/Groups base suffix [o=usergroup]:
```

ユーザーおよびグループのベースサフィックスは、ユーザーエントリおよびグループエントリのネームスペースを保持する、組織ツリー内のトップエントリです。選択するユーザーおよびグループのベースサフィックスは、Directory Server インストールおよび Messaging Server インストールで指定したものと同じベースサフィックスであることを確認します。

注 Access Manager がインストールされている場合は、Access Manager のインストール時に指定したサフィックスが、この質問に答えて指定するものと同じであることを確認します。同じサフィックスを使用しない場合、Messaging Server は Access Manager のインストールを認識できません。

組織ツリーの詳細については、『Sun Java Enterprise System 2003Q4 インストールガイド』の第 12 章「Messaging Server 6.0 のプロビジョニングとスキーマの概念」(http://docs.sun.com/source/817-4242-10/provisioning-concepts.html) を参照してください。

7. スキーマタイプ

```
There are 3 possible schema types:
  1   - schema 1 for systems with iMS 5.x data
  1.5 - schema 2 compatibility for systems with iMS 5.x data
        that has been converted with commdirmig
  2   - schema 2 native for systems using Access Manager

Please enter the Schema Type (1, 1.5, 2) [1]:
```

Sun LDAP Schema 1 を使用する予定の場合は、オプション 1 を選択します。

Sun LDAP Schema 2 (互換モード) を使用する予定の場合は、オプション 1.5 を選択します。詳細は、『Sun Java System Communications Services Schema Migration Guide』を参照してください。

Sun LDAP Schema 2 (ネイティブモード) を使用する予定の場合は、オプション 2 を選択します。

Access Manager がインストールされていない場合に、comm_dssetup.pl が終了することはなくなりました。代わりに、Access Manager がインストールされていないことを警告し、Schema 2 をインストールするオプションを提供します。警告画面は次のようになります。

```
Please enter the Schema Type (1, 1.5, 2) [1]: 2

Access Manager has not been configured for this new user/group suffix

You can opt to continue, but you will not be able to use features that
depend on Access Manager

Are you sure you want this schema type? [n]:
```

スキーマオプションの詳細は、『Sun Java System Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) を参照してください。

8. ドメインコンポーネント (DC) ツリーのベースサフィックス

```
Please enter the DC Tree base suffix [o=internet]:
```

注 手順 7 でオプション 1 または 1.5 を選択した場合は、DC ツリーのベースサフィックスを入力するよう求められます。オプション 2 の「Sun LDAP Schema 2 - ネイティブモード」を選択した場合、この入力は求められません。

DC ツリーは、ローカル DNS 構造をミラー化したものであり、ユーザーとグループのデータエントリを含む組織ツリーへのインデックスとしてシステムにより使用されます。DC ツリーのベースサフィックスは、DC ツリーの最上位エントリの名前です。デフォルトの `o=internet` か別の名前のどちらかを選択できます。

DC ツリーまたは組織ツリーの詳細については、『Sun Java Enterprise System 2003Q4 インストールガイド』の第 12 章「Messaging Server 6.0 のプロビジョニングとスキーマの概念」(http://docs.sun.com/source/817-4242-10/provisioning-concepts.html) を参照してください。

9. スキーマファイルの更新

```
Do you want to update the schema files [yes]:
```

`Yes` と答えると、新しい要素がスキーマに追加されます。新しいバージョンの Messaging Server をインストールするたびに、新しいスキーマファイルで Directory を更新することをお勧めします。

10. 新しいインデックスの設定

```
Do you want to configure new indexes [yes]:
```

手順 5 (ユーザーとグループの Directory Server) に対して `Yes` と答えた場合、新しいインデックスを設定するかどうか尋ねられます。新しいインデックスは、キャッシュを作成してディレクトリ検索の効率を向上させるために使用されます。この質問には `Yes` と答えることをお勧めします。ただし、次のような状況では、インデックスを作成する必要はありません。

- レプリカを提供するためだけに使用される、マスターのユーザー / グループ Directory Server である。つまり、このユーザー / グループ Directory Server に対して直接クエリーが実行されることはない。
- 本稼働用のユーザー / グループ Directory Server に多数のエントリがあり、インデックス作成のために長い停止時間が発生するのは望ましくない。

11. 設定の要約

```
Here is a summary of the settings that you chose:
  Server Root                          : /var/opt/mps/serverroot/
  Server Instance                      : slapd-varrius
  Users/Groups Directory               : Yes
  Update Schema                        : yes
  Schema Type                          : 1
  DC Root                              : o=internet
  User/Group Root                      : o=usergroup
  Add New Indexes                      : yes
  Directory Manager DN                 : cn=Directory Manager

Now ready to generate a shell script and ldif file to modify the
Directory.
No changes to the Directory Server will be made this time.

Do you want to continue [y]:
```

ディレクトリ設定が更新される前に、設定の要約が表示されます。この時点では変更は加えられません。

注 手順 7 で、オプション 2 の「Sun LDAP Schema 2 (ネイティブモード)」を選択した場合、設定の要約の DC Root は、User/Group Root に入力した値と同じにになります。

設定を変更したい場合は、No と入力し、スクリプトを再度実行します。

Yes と入力して続行すると、comm_dssetup.pl スクリプトは、LDIF ファイルと、ディレクトリサーバー内のインデックスとスキーマの更新に使われるシェルスクリプトを作成します。

/var/tmp/dssetup_*YYYYMMDDHHMMSS*.sh
/var/tmp/dssetup_*YYYYMMDDHHMMSS*.ldif

ここで、*YYYYMMDDHHMMSS* は、ファイルの作成された時刻と日付のスタンプを示します。

注 スクリプトをこの時点で実行するか、あとで実行するかを選択できます。スクリプトをここで実行する場合は、続行するかどうかを尋ねられたときに Yes と入力します。スクリプトをあとで実行する場合は、/var/tmp/dssetup_*YYYYMMDDHHMMSS*.sh を使用して、スクリプトを呼び出すことができます。

サイレントモード

サイレントモードを有効にするには、一度にすべての引数を指定します。

構文

```
# perl comm_dssetup.pl -i yes|no -c Directory_Server_Root -d
Directory_instance -r DC_tree -u User_Group_suffix -s yes|no -D
"DirectoryManagerDN" -w password -b yes|no -t 1|1.5|2 -m yes|no [-S
path-to-schema-files]
```

オプション

このコマンドのオプションは、以下のとおりです。

オプション	説明
-i yes|no	次の質問に答えます。「Do you want to configure new indexes?」新しいインデックスを設定する場合は、yes を指定します。新しいインデックスを設定しない場合は、no を指定します。
-c *Directory_Server_Root*	Directory Server ルートのパス名。 例 : /var/opt/mps/serverroot/
-d *Directory_instance*	Directory Server インスタンスのサブディレクトリ。 例 : slapd-budgie
-r *DC_tree*	DC ツリーのサフィックス。例 : o=internet
-u *User_Group_suffix*	ユーザー / グループサフィックス。例 : o=usergroup
-s yes|no	次の質問に答えます。「Do you want to update the schema?」スキーマファイルを更新する場合は、yes を指定します。スキーマファイルを更新しない場合は、no を指定します。
-D *DirectoryManagerDN*	Directory Manager の DN。例 : "cn=Directory Manager"
-w *password*	Directory Manager のパスワード

オプション	説明
-b yes|no	次の質問に答えます。「Will this directory server be used for users/groups?」ディレクトリサーバーを設定してユーザー / グループ用に使用する場合は、yes を指定します。このディレクトリを設定データのみに使用する場合は、no を指定します。
-t 1|1.5|2	Messaging Server 用に使用するスキーマバージョンを決定します。 • Sun LDAP Schema 1 の場合は、1 を選択します。 • Sun LDAP Schema 2 (互換モード) の場合は、1.5 を選択します。詳細は、『Sun Java System Communications Services Schema Migration Guide』を参照してください。 • Sun LDAP Schema 2 (ネイティブモード) の場合は、2 を選択してください。
-m yes|no	次の質問に答えます。「Do you want to modify the directory server?」ディレクトリを変更する場合は、yes を指定します。ディレクトリを変更しない場合は、no を指定します。
-R <yes|no>	-m yes を指定した場合に、新しいインデックスが見つかるとインデックスの再作成を実行します。
-S *path-to-schema-files*	スキーマファイルへのディレクトリパスを指定します。 例 : ./schema

例

```
# perl comm_dssetup.pl -i yes -c /var/opt/mps/serverroot -d
slapd-budgie
-r o=internet -u o=usergroup -s yes -D "cn=Directory Manager" -w
password -b yes -t 1 -m yes
```

comm_dssetup.pl スクリプトのオプションすべての設定が終わると、実際にスクリプトが実行される前に、次のような概要画面が表示されます。

```
Here is a summary of the settings that you chose:
  Server Root                          : /var/opt/mps/serverroot/
  Server Instance                      : slapd-budgie
  Users/Groups Directory               : Yes
  Update Schema                        : yes
  Schema Type                          : 1
  DC Root                              : o=internet
  User/Group Root                      : o=usergroup
  Add New Indexes                      : yes
  Schema Directory                     : ./schema
  Directory Manager DN                 : "cn=Directory Manager"
```

各オプションの詳細は、45 ページの「インタラクティブモード」の節に記載されています。

Messaging Server の初期実行時設定を作成するには

初期実行時設定プログラムは、Messaging Server を起動して実行する設定を提供します。このプログラムの目的は、「初期実行時設定」を作成して、一般的に機能する Messaging Server 設定を行うことです。この設定を基に、必要に応じてカスタマイズを行うことができます。このプログラムは 1 回だけ実行するように意図されています。このプログラムを実行すると、2 回目以降は設定が上書きされます。初期実行時設定を変更するには、この節と『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) に記載されている設定ユーティリティを使用してください。

Messaging Server の前提条件

初期実行時設定プログラムを実行する前に、次の作業を行う必要があります。

- 管理サーバーをインストールして設定します。(『Sun Java Enterprise System インストールガイド』(http://docs.sun.com/doc/819-0808?l=ja) を参照)。管理サーバーの設定を行わなくても Messaging Server を設定することはできますが、その場合、コンソールを使用して Messaging Server サーバーを管理することはできません。
- Directory Server をインストールして設定します。(『Sun Java Enterprise System インストールガイド』を参照)
- `comm_dssetup.pl` プログラムを実行します (43 ページの「Messaging Server 設定用に Directory Server を準備するには」を参照)。
- 付録 E「インストールワークシート」にあるチェックリストに、管理サーバーと Directory Server のインストールパラメータおよび設定パラメータを記録します。

Messaging Server の設定チェックリスト

Messaging Server の初期実行時設定プログラムを実行するときは、992 ページの表 E-4にパラメータを記入してください。いくつかの質問に対する回答には、『Communications Services 配備計画ガイド』にある Directory Server および管理サーバーのインストールチェックリストを参照してください。

configure プログラムの実行

次の手順では、Messaging Server 初期実行時設定の方法を紹介していきます。

1. 次のコマンドを使って、Messaging Server 初期実行時設定を起動します。

```
/msg_svr_base/sbin/configure [flag]
```

Messaging Server をリモートシステムで設定する場合は、`xhost`(1) コマンドを使用する必要が生じることがあります。

表 1-1 に、configure プログラムを使って設定できるオプションフラグを示します。

表 1-1 Messaging Server の configure プログラムのオプションフラグ

フラグ	説明
-nodisplay	コマンド行の設定プログラムを起動します。
-noconsole	GUI ユーザーインタフェースプログラムを起動します。
-state *[statefile]*	サイレントインストールファイルを使用します。-nodisplay および -noconsole フラグとともに使用する必要があります。「サイレントインストールを実行するには」を参照してください。

configure コマンドを実行すると、次の設定プログラムが起動します。

2. ようこそ

   設定プログラムの最初のパネルは、著作権ページです。続行するには「次へ」を選択し、終了するには「キャンセル」を選択します。管理サーバーを設定しなかった場合は警告が表示されるので、OK を選択して続行します。

3. 完全修飾ホスト名 (FQHN) を入力します。

   これは、Messaging Server が動作するマシン名です。Java Enterprise System インストーラを使用してサーバーをインストールしたときに、すでに物理ホスト名を指定済みである場合があります。しかし、クラスタ環境をインストールする場合は、論理ホスト名を使用する必要があります。ここで、最初に指定していたホスト名を変更できます。

4. 設定およびデータファイルを保存するディレクトリを選択します。

   Messaging Server の設定およびデータファイルを保存するディレクトリを選択します。*msg_svr_base* の下にはないパス名を指定します。設定とデータディレクトリへのシンボリックリンクが *msg_svr_base* に作成されます。これらのシンボリックリンクの詳細については、70 ページの「インストール後のディレクトリレイアウト」を参照してください。

   これらのファイルを保存するのに十分なディスク容量があることを確認します。

5. コンポーネントが読み込まれていることを示す、小さいウィンドウが表示されます。

   コンポーネントの読み込みには数分かかることがあります。

6. 設定するコンポーネントを選択します。

   設定を行う Messaging コンポーネントを選択します。

- Message Transfer Agent: ルーティング、ユーザーメールの配信、および SMTP 認証を処理します。MTA は、ホストドメイン、ドメイン別名、サーバー側フィルタの機能をサポートします。
- メッセージストア : メッセージストアを介して、一貫性のあるメッセージングサービスを提供します。メッセージストアには、HTTP、POP、および IMAP プロトコルを介してアクセスできます。メッセージストアを設定するだけの場合、MTA も選択する必要があります。
- Messenger Express: メッセージストアからメッセージを取得する HTTP プロトコルを処理します。Messenger Express を設定するだけであれば、メッセージストアと MTA も選択する必要があります。
- Messaging Multiplexor: 組織内の複数のメッセージングサーバーマシンに対するプロキシとして機能します。ユーザーは Multiplexor サーバーに接続し、Multiplexor サーバーが各接続を適切なメールサーバーにリダイレクトします。このコンポーネントは、デフォルトでは無効になっています。MMP とメッセージストアのチェックを行うと、それらが同じシステム上で有効になります。設定後、ポート番号を変更するための警告メッセージが表示されます (この手順については、72 ページの「インストール後のポート番号」を参照)。

  MMP を設定するには、第 7 章「マルチプレクササービスの設定および管理」と『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) を参照してください。

設定するコンポーネントにチェックマークを付け、設定しないコンポーネントのチェックマークを外します。

7. 設定されたファイルを所有するシステムユーザー名とグループを入力します。

   システムユーザーおよびグループの設定の詳細は、42 ページの「UNIX システムのユーザーとグループを作成するには」を参照してください。

8. Configuration Directory Server パネル

   Configuration Directory LDAP URL、管理者、およびパスワードを入力します。これは管理サーバー設定から取得されます。

   Directory Server のインストールから Configuration Server LDAP URL を収集します。『Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) にある Directory Server インストールワークシートを参照してください。

Directory Manager には、Directory Server、および Directory Server を使用するすべての Sun Java System サーバー (Messaging Server など) に対する包括的な管理権限が付与されています。また、Directory Server 内のすべてのエントリの管理に必要なすべてのアクセス権も与えられています。デフォルトかつ推奨される識別名 (DN) は、`cn=Directory Manager` です。これは Directory Server の設定時に設定されます。

注 デフォルト以外の値を選択すると、管理サーバーと Configuration Directory Server の間に不一致が発生します。この場合、設定後に手動での作業が必要になります。したがって、このエントリを変更するのは、十分な知識がある場合だけにしてください。

9. User/Group Directory Server パネル

   ユーザーおよびグループの Directory LDAP URL、管理者、およびパスワードを入力します。

   ホストからユーザーおよびグループのサーバー LDAP URL 情報を収集し、Directory Server インスタンスからポート番号情報を収集します。『Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) にある Directory Server インストールワークシートを参照してください。

   Directory Manager には、Directory Server、および Directory Server を使用するすべての Sun Java System サーバー (Messaging Server など) に対する包括的な管理権限が付与されており、Directory Server 内のすべてのエントリの管理に必要なすべてのアクセス権が与えられています。デフォルトかつ推奨される識別名 (DN) は、`cn=Directory Manager` です。これは Directory Server の設定時に設定されます。

   レプリケートされた Directory Server インスタンスに対してインストールする場合は、マスターディレクトリではなく、レプリカの資格情報を指定する必要があります。

10. ポストマスターの電子メールアドレス

    ポストマスターの電子メールアドレスを入力します。

    管理者が頻繁に監視するアドレスを選択します。たとえば、siroe ドメインのポストマスターの場合は `pma@siroe.com` です。このアドレスは「Postmaster」から始めることはできません。

    電子メールアドレスのユーザーは自動的には作成されません。このため、プロビジョニングツールを使って作成する必要があります。

11. 管理者アカウントのパスワード

    PAB 管理者と SSL パスワードに対してだけでなく、サービス管理者、サーバー、ユーザーおよびグループの管理者、エンドユーザー管理権限に対しても使用する初期パスワードを入力します。

    初期実行時設定のあとで、このパスワードを個々の管理者アカウント用に変更することもできます。詳細は、106 ページの「パスワードを変更するには」を参照してください。

12. デフォルトの電子メールドメイン

    デフォルトの電子メールドメインを入力します。

    この電子メールドメインは、ほかにドメインが指定されていない場合に使用されるデフォルトです。たとえば、デフォルトの電子メールドメインが `siroe.com` の場合、ドメインのないユーザー ID に宛てられたメッセージはこのドメインに送られます。

    User Management Utility (Sun LDAP Schema 2 を含むユーザーとグループのプロビジョニングのためのコマンド行インタフェース) を使用する場合は、設定中に同じデフォルトのドメインを指定します。詳細については、『Sun Java System Communications Services Delegated Administrator 管理ガイド』(`http://docs.sun.com/doc/819-1101?l=ja`) を参照してください。

13. 組織 DN

    ユーザーやグループを作成する組織の DN を入力します。デフォルトは、電子メールドメインの前にユーザー / グループサフィックスを付けたものです。

    たとえば、ユーザー / グループサフィックスが `o=usergroup` で、電子メールドメインが siroe.com の場合、デフォルトは `o=siroe.com, o=usegroup` です。ここで `o=usergroup` は、43 ページの「Messaging Server 設定用に Directory Server を準備するには」で指定したユーザー / グループのディレクトリサフィックスです。

    組織 DN と同じユーザー / グループのディレクトリサフィックスを選択すると、ホストドメインを作成するよう決定した場合に移行の問題が起きることがあります。初期実行時設定でホストドメインを設定する場合はユーザーおよびグループサフィックスの 1 つ下のレベルの DN を指定してください。

14. 設定準備完了

    設定プログラムは、マシンに十分なディスク容量があるかを確認してから、設定の準備が完了したコンポーネントの概略を示します。

    Messaging のコンポーネントを設定するには、「すぐに設定」を選択します。設定変数を変更するには、「戻る」を選択します。また、設定プログラムを終了するには、「キャンセル」を選択します。

15. Starting Task Sequence、Sequence Completed、および Installation Summary パネル

    「インストールの概要」パネルで「詳細」を選択することで、インストールの状態を確認することができます。プログラムを終了するには、「閉じる」を選択します。

    ログファイルは、*/msg_svr_base*/install/configure_*YYYYMMDDHHMMSS*.log に作成されます。ここで、*YYYYMMDDHHMMSS* は、設定の 4 桁の年、月、日、時、分、および秒を示します。

    これで、Messaging Server の初期実行時設定が設定されます。設定パラメータに変更を加える場合は、このマニュアルでその手順に該当する部分を参照してください。

Messaging Server を起動するには、次のコマンドを使用します。

```
/opt/SUNWmsgsr/sbin/start-msg
```

サイレントインストールを実行するには

Messaging Server 初期実行時設定プログラムは、自動的にサイレントインストール状態ファイル (saveState というファイル) を作成します。このファイルは、Messaging Server Solaris パッケージがインストールされている配備で、追加の Messaging Server インスタンスをすばやく設定するために使用できます。設定プロンプトで指定したすべての値が、そのファイルに記録されます。

サイレントインストールを実行すると、configure プログラムは、サイレントインストール状態ファイルを読み取ります。configure プログラムは、Messaging Server のその後の初期実行時設定では同じ質問を繰り返さずに、このファイルの値を使用します。したがって、新しいインストールで状態ファイルを使用すると、ユーザーは一切質問を受けることがありません。その代わりに、新しいインストールパラメータとして状態ファイルの値が自動的に適用されます。

サイレントインストール状態ファイル saveState は、*msg_svr_base*/install/configure_*YYYYMMDDHHMMSS* ディレクトリに保存されます。ここで、*YYYYMMDDHHMMSS* は、saveState ファイルの 4 桁の年、月、日、時、分、および秒を示します。

サイレントインストール状態ファイルを使用して、配備先の別のマシン上に別の Messaging Server インスタンスを設定するには、次の手順に従います。

1. サイレントインストール状態ファイルを、新しいインストールを行うマシン上の一時的な領域にコピーします。

2. 必要に応じて、サイレントインストール状態ファイルを見直して編集します。

   多くの場合、状態ファイルのパラメータおよび指定の変更が必要になります。たとえば、新しいインストールのデフォルトの電子メールドメインが、状態ファイルに記録されているデフォルトの電子メールドメインと異なる場合などです。状態ファイルに記録されたパラメータが、このインストールに自動的に適用されることを忘れないでください。

3. 次のコマンドを実行して、サイレントインストールファイルを使ってほかのマシンを設定します。

   msg_svr_base/sbin/configure -nodisplay -noconsole -state ¥
   fullpath/saveState

   ここで、*fullpath* は、saveState ファイルがある場所の完全なディレクトリパスです (この節の手順 1 を参照)。

注 サイレントインストールプログラムを実行すると、サイレントインストールによって、新しい状態ファイルが、*msg_svr_base*/install/configure_*YYYYMMDDHHMMSS*/saveState ディレクトリに作成されます。ここで、*YYYYMMDDHHMMSS* は、saveState ファイルの 4 桁の年、月、日、時、分、および秒を示します。

Directory Server のレプリカに対して Messaging Server をインストールするには

Directory Server のマスターに対して Messaging Server をインストールする場合、次のことによって制限を受ける可能性があります。

- Directory Server のマスターの資格を持っていない。
- Messaging Server が Directory Server マスターと直接通信できない。

Directory Server のレプリカに対して Messaging Server をインストールするには、次の手順に従います。

1. 44 ページの「comm_dssetup.pl 要件」に記載されているように、Directory Server のレプリカを含むすべての Directory Server に対して comm_dssetup.pl プログラムを実行します。
2. 「Messaging Server の初期実行時設定を作成するには」の手順 8 と手順 9 で説明されているように、レプリケートされた Directory Server の資格を使用して、Messaging configure プログラム (*msg_svr_base*/sbin/configure に存在する) を実行します。

Directory Server 管理者を設定しようとすると、無効な権限のため、configure プログラムは失敗します。ただし、Directory Server のレプリカに適切な権限を許可するために必要な *msg_svr_base*/config/*.ldif ファイルが生成されます。

3. *.ldif ファイルを Directory Server マスターに移動します。
4. *.ldif ファイルで ldapmodify コマンドを実行します。

   ldapmodify または *msg_svr_base*/install/configure_*YYYYMMDDHHMMSS*.log の詳細は、Sun Java System Directory Server のマニュアルを参照してください。
5. configure プログラムを再度実行します。

   これで、Directory Server のレプリカ (およびマスター) が、Messaging Server で適切に機能するように設定されます。

Messaging Server プロビジョニングツールをインストールするには

以下の節では、サポートされているプロビジョニングツールに関するインストール情報の概要について説明します。

- 61 ページの「Delegated Administrator for Messaging」
- 63 ページの「LDAP プロビジョニングツール」
- 42 ページの「UNIX システムのユーザーとグループを作成するには」

Delegated Administrator for Messaging

Messaging Server には、iPlanet Delegated Administrator (Sun LDAP Schema 1) と Communications Services Delegated Administrator (Sun LDAP Schema 2) という 2 つの GUI プロビジョニングツールが用意されています。この節では前者について説明します。後者の詳細については、『Sun Java System Communications Services Delegated Administrator 管理ガイド』(http://docs.sun.com/doc/819-1101?l=ja) を参照してください。

iPlanet Delegated Administrator (Sun LDAP Schema 1) をインストールするには、Sun ソフトウェアページからダウンロードする必要があります。ダウンロード場所の情報については、ご利用の Sun Java System 代理店にお問い合わせください。

注

iPlanet Delegated Administrator は、Messaging Server と Web Server をインストールして設定してからしかインストールできません。iPlanet Delegated Administrator のインストールの詳細については、iPlanet Delegated Administrator のマニュアルを参照してください。

iPlanet Delegated Administrator を利用できるのは、既存の Messaging Server 5.x インストールに対して Messaging Server 6 をインストールする場合だけです。Messaging Server 製品を初めて使用する場合は利用できません。

iPlanet Delegated Administrator は、Sun Java System Web Server 6.0 (以前の Messaging Server 5.2 製品のみにバンドルされている) とともに使用する必要があります。Web Server 6.1 (Java Enterprise System インストーラにバンドルされている) を iPlanet Delegated Administrator とともに使用することはできません。

『Sun Java System Messaging Server リリースノート』(http://docs.sun.com/doc/819-0150?l=ja) を必ずお読みください。

インストール手順の要約 : iPlanet Delegated Administrator for Messaging を Messaging Server とともにインストールして設定するには、次の手順を行います。

注

次の製品をインストールするときは、Java Enterprise System インストーラを使用します。これらの製品の中には、独自の設定を持つものもあれば、製品のコンフィギュレータが Java Enterprise System インストーラ / コンフィギュレータに埋め込まれているものもあります。詳細は、該当の製品のマニュアルを参照してください。

1. Sun Java System Directory Server 5.2 がインストールされ設定されていることを確認します。

   詳細は、『Sun Java System Directory Server インストールガイド』を参照してください。

2. Messaging Server をインストールして設定します。

   Messaging Server は、Sun Java System Access Manager がインストールされないため、Sun LDAP Schema 1 を検出します。

3. 以前の Messaging Server 5.2 バンドルから Sun Java System Web Server 6.0 をインストールします。

   Sun Java System Web Server のマニュアルと Sun Java System Delegated Administrator のマニュアルを参照してください。

4. iPlanet Delegated Administrator for Messaging 1.2 パッチ 2 をインストールします。最新バージョンの入手については、Sun サポート代理店にお問い合わせください。

   iPlanet Delegated Administrator のマニュアルを参照してください。

LDAP プロビジョニングツール

Sun LDAP Schema 1 のユーザーとグループは、LDAP ディレクトリツールを使用してプロビジョニングすることができます。Schema 2 はサポートされていません。

インストール手順の要約 :

1. Directory Server がインストールされていない場合は、必ずインストールして設定してください。

   詳細は、『Sun Java Enterprise System インストールガイド』(http://docs.sun.com/doc/819-0808?l=ja) を参照してください。

2. Access Manager を設定して、Directory Server 内のデータを認識します。

   Access Manager が LDAP ディレクトリ内のデータを認識できるようにするためには、Access Manager によって管理されるすべての組織、グループ、およびユーザーのエントリに、特殊なオブジェクトクラスを追加する必要があります。まだ追加していない場合は、新しいアカウントのプロビジョニングを開始する前に実行してください。これらのオブジェクトクラスをディレクトリに自動的に追加できるよう、サンプルスクリプトが Access Manager 製品にバンドルされています。これらのインストール後の手順の詳細は、『Sun Java System Access Manager Migration Guide』(http://docs.sun.com/doc/817-7645) を参照してください。

3. このマニュアルを参照して、Messaging Server をインストールして設定します。

   Messaging Server は、Access Manager がインストールされているかどうかに従って、どの Sun Java System LDAP Schema を使用するかを検出します。

4. Sun Java System Web Server 6.1 をインストールおよび設定して、Messenger Express でメールのフィルタリングを有効にします。メールのフィルタリングの有効化の詳細は、69 ページの「Messenger Express および Communications Express メールフィルタの設定」を参照してください。Web Server をインストールするには、『Sun Java Enterprise System インストールガイド』を参照してください。

   メールのフィルタリングはプロビジョニングツールではありませんが、この機能は以前の GUI バージョンの Delegated Administrator for Messaging に存在していました。

5. LDAP プロビジョニングの実行については、Sun Java System Messaging Server のマニュアルを参照してください。

   Sun LDAP Schema 1 の LDAP プロビジョニングの場合は、『Messaging Server 5.2 Provisioning Guide』と『Sun Java System Communications Services Schema Reference Manual』を参照してください。『Sun Java System Schema Reference Manual』には、Sun LDAP Schema 1 と v.2 の両方のオブジェクトクラスと属性が記載されています。

SMTP リレーブロッキング

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、サーバーがあるホスト以外のシステムのこと)。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。

インストール後、自分のサイトのニーズを満たすように手動で設定を変更することが必要です。特に、Messaging Server が、内部システムと SMTP リレーを許可するサブネットを認識するようにします。この設定を変更しなければ、MTA 設定のテスト時に問題が生じる可能性があります。

IMAP クライアントと POP クライアントが Messaging Server システムの SMTP サーバー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。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、ファイル *msg_svr_base*`/config/mappings` にあります。

たとえば、IP アドレスが `192.45.67.89` の 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

  $(192.45.67.89/24)  $Y
  127.0.0.1  $Y
  *  $N
```

この例の最初のエントリでは、$(IP-pattern/signicant-prefix-bits) 構文を使用して、24 ビットの 192.45.67.89 すべてに一致する IP アドレスが内部として認識されるように指定しています。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ように指定しています。

最後の $N エントリの前に別の IP アドレスやサブネットを指定して、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、IP アドレスまたはサブネット (サブネットの指定には $(.../...) 構文を使用) を左側に、$Y を右側に指定する必要があります。また、既存の $(.../...) エントリを変更して、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 192.45.67.0 サブネットを所有する場合は、マッピングテーブルの最初のエントリを次のように変更します。

```
INTERNAL_IP

  $(192.45.67.89/24)  $Y
  127.0.0.1  $Y
  *  $N
```

また、サイトが 192.45.67.80 ～ 192.45.67.99 の範囲の IP アドレスだけを持つ場合は、次の手順を行います。

```
INTERNAL_IP

  ! Match IP addresses in the range 192.45.67.80-192.45.67.95
  $(192.45.67.80/28)  $Y
  ! Match IP addresses in the range 192.45.67.96-192.45.67.99
  $(192.45.67.96/30)  $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、*msg_svr_base*/sbin/imsimta test -match ユーティリティが便利です。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も便利です。

INTERNAL_IP マッピングテーブルを変更したあとは、変更を有効にするために、必ず、*msg_svr_base*/sbin/imsimta cnbuild および *msg_svr_base*/sbin/imsimta restart ユーティリティを実行してください。

マッピングファイルと一般的なマッピングテーブルの形式、および imsimta コマンド行ユーティリティについては、『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) を参照してください。また、INTERNAL_IP マッピングテーブルについては、550 ページの「SMTP リレーを追加するには」を参照してください。

システム再起動後の起動

起動スクリプト *msg_svr_base*/lib/Sun_MsgSvr を使用すると、システム再起動後に Messaging Server を起動できるようになります。デフォルトでは、このスクリプトを実行しない限り、システム再起動後に Messaging Server が起動することはありません。また、このスクリプトは MMP が有効な場合に MMP を起動できます。

Sun_MsgSvr を有効にするには、次の手順を行います。

1. Sun_MsgSvr スクリプトを /etc/init.d ディレクトリの中にコピーします。
2. Sun_MsgSvr スクリプトの所有権とアクセスモードを次のように変更します。

表 1-2 Sun_MsgSvr に対する所有権とアクセスモードの変更

所有権 (chown(1M))	グループ所有権 (chgrp(1M))	アクセスモード (chmod(1M))
root (スーパーユーザー)	sys	0744

3. /etc/rc2.d ディレクトリに移動して、次のシンボリックリンクを作成します。

```
ln /etc/init.d/Sun_MsgSvr S92Sun_MsgSvr
```

4. /etc/rc0.d ディレクトリに移動して、次のシンボリックリンクを作成します。

```
ln /etc/init.d/Sun_MsgSvr K08Sun_MsgSvr
```

sendmail クライアントの処理

エンドユーザーが sendmail クライアントからメッセージを送信する場合、Messaging Server がプロトコル経由でクライアントとともに動作するよう設定することができます。ユーザーは、引き続き UNIX sendmail クライアントを使用することができます。

sendmail 設定ファイルを作成し修正すると、sendmail クライアントと Messaging Server の間での互換性を確保できます。

注 システムに新しい sendmail パッチを適用するたびに、後述の「Solaris 9 以降」の手順で説明しているように、submit.cf ファイルを変更する必要があります。Solaris 8 では、次の手順に従ってください。

以前のバージョンの Messaging Server では、アップグレードすると /usr/lib/sendmail バイナリが sendmail 製品のコンポーネントで置き換えられていました。アップグレード時のこの置換は、Messaging Server 6 2005Q1 では発生しなくなりました。したがって、適切なバージョンの /usr/lib/sendmail バイナリを最新の sendmail パッチから入手する必要があります。

Solaris 8

Solaris 8 オペレーティングシステムの場合は、次の手順に従ってください。

1. ディレクトリ /usr/lib/mail/cf 内のファイル main-v7sun.mc を検索して、このファイルのコピーを作成します。

   この節の例では、sunone-msg.mc というコピーが作成されます。

2. sunone-msg.mc ファイルで、MAILER マクロの前に以下の行を追加します。

```
FEATURE('nullclient', 'smtp:rhino.west.sesta.com')dnl
MASQUERADE_AS('west.sesta.com')dnl
define('confDOMAIN_NAME', 'west.sesta.com')dnl
```

「Messaging Server の初期実行時設定を作成するには」内の 58 ページの「デフォルトの電子メールドメイン」で説明しているように、rhino.west.sesta.com はローカルホスト名で、west.sesta.com はデフォルトの電子メールドメインです。HA 環境では論理ホスト名を使用してください。高可用性の論理ホスト名の詳細については、第 3 章「高可用性の構成」を参照してください。

3. sunone-msg.mc ファイルをコンパイルします。

```
/usr/ccs/bin/make sunone-msg.cf
```

sunone-msg.mc は、sunone-msg.cf を出力します。

4. /etc/mail ディレクトリにある既存の sendmail.cf ファイルのバックアップコピーを作成します。
   a. /usr/lib/mail/cf/sunone-msg.cf をコピーして、名前を sendmail.cf ファイルに変更します。
   b. 新しい sendmail.cf ファイルを /etc/mail ディレクトリに移動します。

Solaris 9 以降

Solaris 9 プラットフォームでは、sendmail は setuid プログラムではありません。現在は、setgid プログラムです。

Solaris プラットフォームで sendmail 設定ファイルを作成するには、次の手順を行います。

1. ディレクトリ /usr/lib/mail/cf 内でファイル submit.mc を検索し、そのファイルのコピーを作成します。

   この節の例では、sunone-submit.mc というコピーが作成されます。

2. ファイル sunone-submit.mc で次の行を変更します。

```
FEATURE('msp')dn
```

から

```
FEATURE('msp', 'rhino.west.sesta.com')dnl
```

ここで、rhino.west.sesta.com はローカルホスト名です。

「Messaging Server の初期実行時設定を作成するには」内の 58 ページの「デフォルトの電子メールドメイン」で説明しているように、rhino.west.sesta.com はローカルホスト名で、west.sesta.com はデフォルトの電子メールドメインです。HA 環境では論理ホスト名を使用してください。高可用性の論理ホスト名の詳細については、第 3 章「高可用性の構成」を参照してください。

3. sunone-submit.mc ファイルをコンパイルします。

```
/usr/ccs/bin/make sunone-submit.cf
```

sunone-submit.mc は、sunone-submit.cf を出力します。

4. /etc/mail ディレクトリにある既存の submit.cf ファイルのバックアップコピーを作成します。
   a. /usr/lib/mail/cf/sunone-submit.cf ファイルをコピーして、名前を submit.cf ファイルに変更します。
   b. 新しい submit.cf ファイルを /etc/mail ディレクトリに移動します。

Messenger Express および Communications Express メールフィルタの設定

メールフィルタは、Messenger Express および Communications Express を経由してアクセスできます。Communications Express だけを使用する場合は、.war ファイルを配備する必要はありません。ただし、Messenger Express のメールフィルタを配備するには、次のコマンドを実行する必要があります。

Web Server を Web コンテナとして使用している場合

```
# cd web_svr_base/bin/https/httpadmin/bin/
# ./wdeploy deploy -u /MailFilter -i https-srvr_instance -v
https-virtual_srvr_instance msg_svr_base/SUNWmsgmf/MailFilter.war
```

Application Server を Web コンテナとして使用している場合

```
# cd app_svr_base/sbin
# ./asadmin
asadmin> deploy --user admin msg_svr_base/SUNWmsgmf/MailFilter.war
```

どちらの場合も、次の configutil パラメータを設定し、mshttpd を再起動してください。

```
# cd msg_svr_base/sbin/
# ./configutil -o "local.webmail.sieve.port" -v "WS_port_no|AS_port_no"
# ./stop-msg http
# ./start-msg http
```

管理コンソールを使用して .war ファイルを配備することもできます。詳細については、『Sun Java System Web Server 6.1 管理者ガイド』(http://docs.sun.com/app/doc/819-0822?l=ja) または『Sun Java System Application Server Enterprise Edition 8.1 管理ガイド』(http://docs.sun.com/doc/819-1551?l=ja) を参照してください。

エンドユーザー向けのメールフィルタに関する情報は、Messenger Express および Communications Express のオンラインヘルプファイルに記載されています。

パフォーマンスとチューニング

『Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) で、特に Messaging Server アーキテクチャのパフォーマンスの考慮に関連する節を参照してください。

インストール後のディレクトリレイアウト

Sun Java System Messaging Server のインストール後、そのディレクトリおよびファイルは表 1-3 に示した構成で配置されます。この表はすべてを網羅したものではありません。典型的なサーバー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 1-3　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	デフォルトの位置および説明
Messaging Server Base (*msg_svr_base*)	/opt/SUNWmsgsr/ (デフォルトの位置) Messaging Server マシン上に存在するディレクトリで、サーバープログラムや設定、保守、および情報ファイルを維持するためのものです。 マシンごとに 1 つの Messaging Server Base ディレクトリのみが許可されます。

表 1-3　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	デフォルトの位置および説明
設定 config	*msg_svr_base*/config/ imta.cnf ファイルや msg.conf ファイルなどの Messaging Server 設定ファイルのすべてが含まれます。 Solaris および Linux プラットフォームのみ : このディレクトリは、初期実行時設定で指定したデータと設定のディレクトリ (デフォルト : /var/opt/SUNWmsgsr/) のサブディレクトリである config にシンボリックリンクしています (UNIX プラットフォーム)。
ログ log	*msg_svr_base*/log/ mail.log_current ファイルをはじめとする Messaging Server のログファイルが含まれます。 Solaris および Linux プラットフォームのみ : このディレクトリは、初期実行時設定で指定したデータと設定のディレクトリ (デフォルト : /var/opt/SUNWmsgsr/) のサブディレクトリである log にシンボリックリンクしています (UNIX プラットフォーム)。
データ data	*msg_svr_base*/data/ (必須の位置) データベース、設定、ログファイル、サイトのプログラム、キュー、ストア、およびメッセージファイルが含まれます。 data ディレクトリには、config ディレクトリと log ディレクトリが含まれています。 Solaris および Linux プラットフォームのみ : このディレクトリは、初期実行時設定で指定したデータと設定のディレクトリ (デフォルト : /var/opt/SUNWmsgsr/) にシンボリックリンクしています (UNIX プラットフォーム)。
システム管理者プログラム sbin	*msg_svr_base*/sbin/ (必須の位置) imsimta、configutil、stop-msg、start-msg、uninstaller などの、Messaging Server システム管理者が実行できるプログラムとスクリプトが含まれます。
ライブラリ lib	*msg_svr_base*/lib/ (必須の位置) 共有ライブラリ、プライベートの実行可能プログラムとスクリプト、デーモン、およびカスタマイズできないコンテンツデータファイルが含まれます。例 : imapd および qm_maint.hlp。

表 1-3　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	デフォルトの位置および説明
SDK インクルードファイル include	*msg_svr_base*/include/ (必須の位置)
	SDK (Software Development Kit) 用のメッセージヘッダーファイルを格納します。
例 examples	*msg_svr_base*/examples/ (必須の位置)
	Messenger Express AUTH SDK などの、各種 SDK の例が含まれます。
インストールデータ install	*msg_svr_base*/install/ (必須の位置)
	インストールログファイル、サイレントインストールファイル、出荷時設定ファイル、および初期実行時設定ログファイルなどの、インストール関連のデータファイルが含まれます。

インストール後のポート番号

インストールおよび初期実行時設定プログラムで、各種サービス用のポート番号が選択されます。これらのポート番号は 1 から 65535 までの任意の値を指定できます。

表 1-4 に、インストール後に指定されるポート番号のリストを示します。

表 1-4　インストール中に指定されるポート番号

ポート番号	サービス (configutil パラメータ)
389	Directory Server をインストールするマシン上の標準 Directory Server LDAP ポート。このポートは、Directory Server インストールプログラムに指定されています。(local.ugldapport)
110	標準 POP3 ポート。同一マシンにインストールされている場合、このポート番号は MMP のポートと競合する可能性があります。(service.pop.port)
143	標準 IMAP4 ポート。同一マシンにインストールされている場合、このポート番号は MMP のポートと競合する可能性があります。(service.imap.port)
25	標準 SMTP ポート。(service.http.smtpport)
80	Messenger Express HTTP ポート。同一マシンにインストールされている場合、このポート番号は Web Server のポートと競合する可能性があります。(service.http.port)

表 1-4　インストール中に指定されるポート番号 (続き)

ポート番号	サービス (configutil パラメータ)
992	SSL を使用した POP3 ポート。暗号化された通信に使用されます。(service.pop.sslport)
993	SSL を使用した IMAP ポート。暗号化された通信に使用されます。同一マシンにインストールされている場合、このポート番号は MMP のポートと競合する可能性があります。(service.imap.sslport)
443	SSL を使用した HTTP ポート。暗号化された通信に使用されます。(service.http.sslport)
7997	Messaging and Collaboration ENS (イベント通知サービス) ポート
27442	内部製品通信用にジョブコントローラが使用するポート。
49994	内部製品通信用に Watcher が使用するポート。Watcher の詳細については、『Sun Java System Messaging Server 管理ガイド』を参照してください。(local.watcher.port)
ユーザー指定	管理サーバーの HTTP ポート (コンソール要求の待機用)。

製品が特定の組み合わせで同一のマシンにインストールされている場合、ポート番号が競合する可能性があります。競合が発生する可能性があるポート番号を表 1-5 に示します。

表 1-5　競合が発生する可能性があるポート番号

ポート番号の競合	ポート	ポート
143	IMAP サーバー	MMP IMAP プロキシ
110	POP3 サーバー	MMP POP3 プロキシ
993	SSL を使用した IMAP	SSL を使用した MMP IMAP プロキシ
80	Web Server ポート	Messenger Express

可能であれば、ポート番号が競合する製品は、別々のマシンにインストールすることをお勧めします。これができない場合は、競合する製品のいずれかのポート番号を変更する必要があります。

ポート番号を変更するには、configutil ユーティリティを使用します。完全な構文と使用方法については、『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) を参照してください。

次の例は、service.http.port.configutil パラメータを使用して、Messenger Express HTTP ポート番号を 8080 に変更します。

```
configutil -o service.http.port -v 8080
```

第 2 章

Sun Java Systems Messaging Server へのアップグレード

この章では、Messaging Server 5.2 から Messaging Server 6 2005Q1 へのアップグレード方法について説明します。

始める前に

アップグレードを実行する前に、次の点を確認してください。

- Messaging Server 6 2005Q1 が、Messaging Server 5.2 システムと同じシステムまたは異なるシステム上にインストールされて設定されている。

注

以前のバージョンの Messaging Server とは異なり、既存の Messaging Server をアップグレードするには、まず Messaging Server 6 2005Q1 をインストールして設定する必要があります。

また、バージョン 5.2 より古いバージョンの Messaging Server には、このアップグレードプログラムは使用できません。したがって、まず Messaging Server 5.2 に移行またはアップグレードし、Messaging Server 6 2005Q1 をインストールしてから、このアップグレードプログラムを実行します。Messaging Server 5.2 への移行の詳細は、『iPlanet Messaging Server 5.2 Migration Guide』(http://docs.sun.com/source/816-6017-10/index.html) を参照してください。

- 既存の Messaging Server 5.2 インストールが、`imsimta dirsync` ではなく MTA ダイレクト LDAP 検索によって設定されている。

- また、Messaging Server 6 2005Q1 は複数のインスタンスをサポートしません。Messaging Server バージョン 5.2 のインスタンスが複数ある場合、Messaging Server 6 2005Q1 にアップグレードできるインスタンスは 1 つだけです。さらに、アップグレードユーティリティで複数のインスタンスの移行を何回も試みると、設定が上書きされます。

アップグレードプロセスの概要

Messaging Server 5.2 から Messaging Server 6 2005Q1 へのアップグレードには、3 つのステップがあります。これらのプロセスを以下の項目で説明します。

1. 76 ページの「設定を更新するアップグレードファイルの作成」(`UpgradeMsg5toMsg6.pl`)
2. 80 ページの「アップグレードユーティリティの実行」(`do_the_upgrade.sh`)
   - MTA の設定 (`make_mta_config_changes.sh`)
   - configutil の各パラメータ (`make_configutil_changes.sh`)
   - バックアップ設定 (`make_backup_config_changes.sh`)
   - mboxlist データベース (`make_mboxlistdb_changes.sh`)
3. 83 ページの「ユーザーメールボックスの移行」(省略可)

設定を更新するアップグレードファイルの作成

この節では、Messaging Server 6 2005Q1 システム上で設定を更新するために、特殊なアップグレードファイルを作成する方法を説明します。

アップグレードファイルについて

アップグレードユーティリティを実行して Messaging Server 5.2 を 6 に移行する前に、まず Perl スクリプト `UpgradeMsg5toMsg6.pl` を実行する必要があります。このスクリプトは *msg_svr_base*`/sbin` にあります。

`UpgradeMsg5toMsg6.pl` は、5.2 設定ファイルを Messaging Server 6 設定ファイルと比較し、設定ファイルごとに、`*.CHANGES` ファイルと `*.MERGED` ファイルの 2 セットを作成します。

*.CHANGES ファイルと *.MERGED ファイルは、ワークスペースディレクトリ /var/tmp/UpgradeMsg5toMsg6.ScratchDir 内に生成されます。

*.CHANGES ファイルは、Messaging Server 5.2 と Messaging Server 6 2005Q1 の間の、設定ファイルの重要な相違点を示します。このファイルでは、Messaging Server 6 2005Q1 でのみ見つかった設定エンティティ、Messaging Server 6 2005Q1 では廃止された Messaging Server 5.2 の設定エンティティ、および Messaging Server 5.2 でのみ見つかった設定エンティティを特に示しています。すべての *.CHANGES ファイルが設定ファイルのバージョン間の相違点を示すわけではなく、また、すべての設定ファイルが *.CHANGES ファイルを生成するわけではありません。

*.MERGED ファイルは、Messaging Server 5.2 と Messaging Server 6 の設定値と設定を統合したものです。一般に次のどちらかの場合に、Messaging Server 5.2 の設定パラメータ値は、Messaging Server 6 2005Q1 でも保持されます。

- Messaging Server 6 2005Q1 にデフォルト値がない場合。
- 5.2 の設定で指定した値がデフォルト値ではない場合。

表 2-1 で、*.MERGED または *.CHANGES ファイルを生成する設定ファイルのリストを示します。

表 2-1 *.MERGED または *.CHANGES ファイルを生成する Messaging Server 設定ファイル

設定情報	説明	*.MERGED ファイルを生成	*.CHANGES ファイルを生成
job_controller.cnf	ジョブコントローラファイル	X	X
変換	変換ファイル	X	
*channel*_option。*channel* は SMTP チャネル	SMTP チャネルオプションファイル	X	
native_option	ネイティブチャネルのオプションファイル (*channel*_option の例外)	X	X
*channel*_headers.opt。*channel* は SMTP チャネル	ヘッダーオプションファイル	X	
dispatcher.cnf	ディスパッチャファイル	X	X
imta_tailor	テイラーファイル	X	X
option.dat	グローバルな MTA オプションファイル	X	X
エイリアス	エイリアスファイル	X	

表 2-1　*.MERGED または *.CHANGES ファイルを生成する Messaging Server 設定ファイル (続き)

設定情報	説明	*.MERGED ファイルを生成	*.CHANGES ファイルを生成
imta.cnf	MTA 設定ファイル。組み込み参照 (ファイルディレクトリの場所など) のみが変更されます。書き換え規則およびチャネル設定は、5.2 の設定が保持されます。imta.cnf ファイルに LMTP を含めるには、LMTP 情報を Messaging Server 6 の imta.cnf ファイルからコピーします。	X	場合によっては、*.CHANGES ファイルが生成されることがあります。
マッピング	マッピングファイル	X	
mappings.locale	ローカライズされたマッピングファイル	X	
internet.rules	インターネットルールの設定ファイル	X	
backup-groups.conf	バックアップグループ定義	X	X
configutil	local.conf および msg.conf 設定ファイルにある設定パラメータの変更。		X

UpgradeMsg5toMsg6.pl Perl スクリプトの実行

UpgradeMsg5toMsg6.pl を実行して、設定の更新に使用できるファイルのセットを作成するには、次の手順に従います。

1. この時点で、5.2 システムと Messaging Server 6 2005Q1 システムの両方を実行することができます。
2. Messaging Server 5.2 および 6 バージョンが同じマシン上にない場合は、Messaging Server 5.2 の *server-root* ディレクトリを転送、抽出して、Messaging Server 6 2005Q1 システムにコピーします。これらのサーバーバージョンが同じマシン上にインストールされている場合は、この手順は省くことができます。

   メッセージストアがシステム間で転送するには大きすぎる場合は、サーバーインスタンスの不可欠な部分だけを新しいシステムに転送することができます。UpgradeMsg5toMsg6.pl には、この詳細を説明したコメントが含まれています。

Messaging Server 5.2 のストアデータを Messaging Server 6 2005Q1 システムにコピーする必要はありません。ただし、アップグレードプロセス中に Messaging Server 5.2 の mboxlist ディレクトリがアクセス可能であることを確認する必要があります。

3. 5.2 バージョンの msg-*instance* と Messaging Server 6 2005Q1 バージョンの *msg_svr_base* に対して、UpgradeMsg5toMsg6.pl アップグレードスクリプトを実行します。このスクリプトは *msg_svr_base*/sbin にあります。次に例を示します。

```
perl UpgradeMsg5toMsg6.pl /usr/sunone/server5/msg-budgie ¥
  /opt/SUNWmsgsr
```

ここで、/usr/sunone/server5/msg-budgie は 5.2 Messaging Server の msg-*instance* で、/opt/SUNWmsgsr は Messaging Server 6 2005Q1 の *msg_svr_base* です。

*.MERGED ファイルと *.CHANGES ファイル (表 2-1 を参照) が作成されます。

4. *.MERGED ファイルをよく確認してください。推奨されている設定を使用しない場合は、設定を手動で調整する必要があります。

このユーティリティは、Messenger Express カスタマイズファイルを更新しません。そのため、Messaging Server 5.2 の関連情報を保持しながら Messaging Server 6 2005Q1 の新情報を追加するには、これらのファイルを手動で変更する必要があります。

アップグレードユーティリティの実行

この節では、do_the_upgrade.sh ユーティリティについて説明します。このユーティリティは /var/tmp/UpgradeMsg5toMsg6.ScratchDir にあり、4 つのサブスクリプトで構成されているシェルスクリプトです。この節には、以下の項目があります。

アップグレードユーティリティの概要

do_the_upgrade.sh ユーティリティは 4 つのシェルスクリプトで構成されています。これらのスクリプトは、*.MERGED ファイルを使用して、Messaging Server 6 2005Q1 システムに含まれる MTA 設定の設定ディレクトリとファイルディレクトリの場所、configutil パラメータ、バックアップパラメータ、および mboxlist データベースを更新します。

このユーティリティでは、do_the_upgrade.sh ユーティリティを実行するほかに、do_the_upgrade.sh ユーティリティを構成するスクリプト (make_mta_config_changes.sh、make_configutil_changes.sh、make_backup_config_changes.sh、および make_mboxlistdb_changes.sh) の 1 つまたは複数を個別に実行することができます。

MTA リレーマシンを Messaging Server 5.2 から Messaging Server 6 2005Q1 にアップグレードする場合、make_mta_config_changes.sh と make_backup_config_changes.sh を実行します (82 ページの「バックアップ設定」を参照)。

do_the_upgrade.sh ユーティリティやサブスクリプトを実行する場合、Messaging Server の 5.2 や 6 2005Q1 が起動や実行をしていないことを確認してください。

do_the_upgrade.sh ユーティリティの実行

do_the_upgrade.sh ユーティリティを実行するには、次のように指定します。

1. Messaging Server 5.2 と 6 の両方をシャットダウンします。
2. 次のようにユーティリティを実行します。

```
# sh /var/tmp/UpgradeMsg5toMsg6.ScratchDir/do_the_upgrade.sh
```

do_the_upgrade.sh スクリプトの実行後に、5.2 のパーティションパスの参照を継続するか (すると Messaging Server 5.2 の *server-root* ディレクトリは削除できなくなる)、5.2 のストアパーティションを Messaging Server 6 2005Q1 ディレクトリが割り当てられる場所に手動で移動するかそのどちらかを選択できます。この手順は Messaging Server の再起動のまえに実行しておくことが必要です。

MTA の設定

do_the_upgrade.sh ユーティリティを構成するサブスクリプトのうち、MTA のアップグレード設定を行うものは、make_mta_config_changes.sh と呼ばれ、/var/tmp/UpgradeMsg5toMsg6.ScratchDir にあります。

make_mta_config_changes.sh スクリプトは、*.MERGED サーバー設定ファイルをバックアップし、Messaging Server 6 2005Q1 のファイルディレクトリ構造内の元の名前と位置に戻します。

このスクリプトによるファイルの名前の変更と移動が完了すると、imsimta cnbuild コマンドが自動的に実行されて MTA 設定が再コンパイルされます。

注 MTA リレーマシンを Messaging Server 5.2 から Messaging Server 6 2005Q1 にアップグレードする場合、make_mta_config_changes.sh と make_backup_config_changes.sh を実行します (82 ページの「バックアップ設定」を参照)。

configutil の各パラメータ

do_the_upgrade.sh ユーティリティを構成するサブスクリプトのうち、configutil のアップグレード設定を行うものは、make_configutil_changes.sh スクリプトと呼ばれ、/var/tmp/UpgradeMsg5toMsg6.ScratchDir にあります。

make_configutil_changes.sh スクリプトは、msg.conf および local.conf ファイル内の新規またはアップグレードされたパラメータを組み込みます。Messaging Server 6 2005Q1 の configutil パラメータにデフォルト値が指定されていない場合、Messaging Server 5.2 の値が Messaging Server 6 2005Q1 バージョンに繰り越されます。

バックアップ設定

do_the_upgrade.sh ユーティリティを構成するサブスクリプトのうち、バックアップのアップグレード設定を行うものは、make_backup_config_changes.sh スクリプトと呼ばれ、/var/tmp/UpgradeMsg5toMsg6.ScratchDir にあります。

make_backup_config_changes.sh スクリプトは、backup-groups.conf ファイルにあるようなバックアップサービスの設定をアップグレードします。

mboxlist データベース

do_the_upgrade.sh ユーティリティを構成するサブスクリプトのうち、mboxlist データベースのアップグレード設定を行うものは、make_mboxlistdb_changes.sh スクリプトと呼ばれ、/var/tmp/UpgradeMsg5toMsg6.ScratchDir にあります。

make_mboxlistdb_changes.sh スクリプトは、5.2 の mboxlist データベースを転送してアップグレードし、それを Messaging Server 6 2005Q1 ディレクトリ構造にアップグレードします。このスクリプトは、4 つの *.db ファイル (folder.db、quota.db、peruser.db、および subscr.db) を、Messaging Server 5.2 システム上の *server-root*/msg-*instance*/store/mboxlist から Messaging Server 6 2005Q1 システム上の *msg_svr_base*/data/store/mboxlist にコピーします。

ユーザーメールボックスの移行

この節では、ユーザーメールボックスを Messaging Server 5.2 システムから Messaging Server 6 2005Q1 システムに移行する方法について説明します。Messaging Server 5.2 から Messaging Server 6 にアップグレードする場合で、メッセージストアデータベース全体をアップグレードするときは、この手順を行う必要はありません。前の節で説明した `make_mboxlistdb_changes.sh` スクリプトを使用して、より効率よくデータベースをアップグレードできます。

この手順は次の場合に実行する必要があります。

- Microsoft Windows から UNIX に、または UNIX から Microsoft Windows に移行する場合。
- 一度にメッセージストア全体を移行しない場合。
- ユーザーの名前を変更する必要がある場合。`UID`、ドメイン名、およびデフォルトドメインの変更を含む。

この手順を使ってメールボックスを移行する場合は、パーティションパスを Messaging Server 5.2 パーティションにマップしないでください。また、`make_mboxlist_changes.sh` スクリプトを実行しないでください。

アップグレードスクリプトで生成される `make_configutil_changes.sh` スクリプトによって、Messaging Server 5.2 パーティションにマップするパーティションパスが自動的に設定されます。これは手動で変更する必要があります。また、`do_the_upgrade.sh` スクリプトから `make_mboxlistdb_changes.sh` スクリプトの呼び出しを削除する必要があります。

ユーザーメールボックスのデータを Messaging Server 5.2 から Messaging Server 6 2005Q1 にオンラインで移動するには、この節で説明する手順に従ってください。データの移動中に Messaging Server を停止する必要はありません。

以下の項目の概略について説明しています。

- 83 ページの「条件」
- 84 ページの「移行手順」

条件

移行の要件は、新旧両方の Messaging Server で `stored` が実行されていることだけです。

移行手順

ユーザーメールボックスを 5.2 システムから Messaging Server 6 2005Q1 システムに移行するには、次の手順に従います。

1. データ移動プロセスが完了するまでメールボックスにアクセスできないことを、あらかじめユーザーに通知します。データを移動する前に、ユーザーがメールシステムからログアウトしていることを確認します。
2. 保留キューにある着信メッセージを保留し、IMAP、POP、および HTTP を介してメールボックスにアクセスできないようにするために、5.2 メッセージストアのすべてのユーザーエントリの `mailUserStatus` ユーザー LDAP 属性を、`active` から `hold` に変更します。

   `mailUserStatus` の詳細については、『Sun Java System Communications Services Schema Reference Manual』(http://docs.sun.com/doc/819-0113) を参照してください。
3. このプロセス中に 5.2 と 6 2005Q1 の両方の Messaging Server が、起動して実行されていることを確認します。
4. すべてのユーザーエントリ内の `mailHost` 属性を、古いメールサーバーから新しいメールサーバーに変更します。

   そのためには、次の `ldapsearch` コマンドを使用して、`mailHost` 属性の変更が必要なユーザーエントリを検索します。

```
ldapsearch -h ldap.siroe.com -b "o=internet" ¥
   "(&(objectclass=maildomain)(mailHost=oldmail.siroe.com))"
```

   次に、`ldapmodify` コマンドを使用して、これらのエントリを新しいメールサーバーに正しく変更します。Messaging Server または Directory Server に付属している `ldapmodify` を使用してください。Solaris の `ldapmodify` は使用しないでください。

   `mailhost` の詳細は、『Sun Java System Communications Services Schema Reference Manual』を参照してください。
5. 古いシステムで、`backup-groups.conf` ファイルを使用してユーザーエントリを均等なグループに分割します (ユーザー名をファイルに入れ、手順 6 で `-u` オプションを使用する方法もある)。
6. ユーザーデータを、Messaging Server 5.2 メッセージストアから Messaging Server 6 2005Q1 メッセージストアに移動します。

これを行うには、imsbackup ユーティリティを使用して Messaging Server 5.2 メッセージストアをバックアップし、imsrestore ユーティリティを使用してメッセージストアを Messaging Server 6 2005Q1 に復元します。たとえば、oldmail.siroe.com から newmail.siroe.com にメールボックスを移行するには、oldmail.siroe.com で次のコマンドを実行します。

```
/<server-root>/bin/msg/store/bin/imsbackup -f- /<instance>/<group>      ¥
| rsh newmail.siroe.com /opt/SUNWmsgsr/lib/msg/imsrestore.sh   ¥
-f- -cy -v1
```

バックアップと復元のセッションを同時に複数 (グループごとに 1 つ) 実行すると、新しいメッセージストアへの転送速度を最適化できます。imsbackup ユーティリティと imsrestore ユーティリティの詳細は、『Messaging Server Reference Manual』(http://docs.sun.com/doc/819-0106) および 637 ページの「メッセージストアのバックアップと復元を行う」 を参照してください。

7. システムの新しいデフォルトのメッセージングサーバーに Messaging Server 6 2005Q1 を設定します。

   newmail.siroe.com (以前 oldmail.siroe.com によってホストされていたドメインを管理するサーバー) をポイントするように、oldmail.siroe.com の A レコードを変更します。

8. 次のコマンドを発行して、Messaging Server 5.2 システムの保留キューにあるメッセージを解放します。

```
imsimta process_held -uid=user -domain=domain
```

   user はユーザー ID で、*domain* はユーザーが常駐するドメインです。

9. ユーザークライアントで新しいメールサーバーが指定されていることを確認します。

   アップグレードが終了したら、ユーザーに、メールクライアントプログラムから新しいメールサーバーをポイントしてもらいます (この例では、oldmail.siroe.com から newmail.siroe.com をポイントしてもらう)。

   別の方法として、MMP の使用があります。MMP を使用した場合、ユーザーが新しいメールサーバーでクライアントを直接ポイントする必要はありません。MMP は、LDAP ユーザーエントリに保存されている mailHost 属性から情報を取得し、その情報を新しいサーバーに自動的にリダイレクトします。

第 3 章

高可用性の構成

この節では、Veritas Cluster Server または Sun Cluster の高可用性のクラスタリングソフトウェアを構成し、Messaging Server で使用するための準備に必要な情報について説明します。『Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) で高可用性に関する章を読んでおくことを前提にしています。また、Veritas または Sun Cluster Server のマニュアルで、詳細な計画、インストール手順、必要なパッチなどの情報を必要に応じて参照してください。

表 3-1 は、Messaging Server で現時点でサポートされている Sun Cluster Server および Veritas Cluster Server のバージョンのリストです。

表 3-1 Sun Cluster Server および Veritas Cluster Servers のサポートされているバージョン

クラスタ	サポートされているバージョン
Sun Cluster Server	Sun Cluster 3.1
Veritas Cluster Server	Veritas Cluster Server 1.3、Veritas Cluster Server 2.0、および Veritas Cluster Server 3.5

この章には、以下の節があります。

クラスタエージェントのインストール

クラスタエージェントは、クラスタフレームワークのもとで動作する Messaging Server プログラムです。

Sun Cluster Messaging Server エージェント (`SUNWscims`) は、Java Enterprise System インストーラから Sun Cluster 3.1 を選択したときにインストールされます。Veritas Cluster Messaging Server エージェント (`SUNWmsgvc`) は、Java Enterprise System CD の Messaging Server の `Product` サブディレクトリにあります (`Solaris_sparc/Product/messaging_svr/Packages/SUNWmsgvc`)。VCS クラスタエージェントをインストールするには、`pkgadd`(1M) コマンドを使用する必要があります。

Messaging Server および高可用性の注意

Messaging Server および高可用性 (Veritas Cluster および Sun Cluster の両方に適用される) のインストールに関して次のことに注意してください。

- クラスタリングソフトウェアは、Messaging Server のインストールと設定を行う前にインストールする必要があります。その場合は、Messaging Server の HA 論理ホスト名が現在指定しているクラスタノードで、インストールを実行してください。ノード名の入力を要求されたら、クラスタエイリアスを入力してください。Messaging Server をインストールする場合は、インストール先のノードはクラスタの論理名であり、物理名とは無関係であることを Administration Server に報告してください。
- Messaging Server の初期実行時設定 (53 ページの「Messaging Server の初期実行時設定を作成するには」を参照) を実行する際に、Messaging Server のクラスタの完全指定 HA 論理ホスト名を指定してください。
- Messaging Server を設定するには、クラスタホスト名を使用してください。それ以外の方法で設定した場合、クラスタホスト名を使用して再設定する必要があります。

useconfig ユーティリティの使用

`useconfig` ユーティリティを使用することで、HA 環境の複数のノード間で単一の設定を共有することができます。このユーティリティは、既存の設定をアップグレードまたは更新するものではありません。

たとえば、最初のノードをアップグレードする場合は、Java Enterprise System インストーラからインストールを行なってから、Messagng Server を設定します。そのあと、Java Enterprise System インストーラを使って Messaging Server パッケージをインストールする 2 番目のノードにフェイルオーバーします。ただし、初期実行時設定プログラム (configure) をもう一度実行する必要はありません。代わりに、useconfig ユーティリティを使用することができます。

このユーティリティを使用するには、useconfig ユーティリティを実行して、以前の Messaging Server 設定を指定します。

```
msg_svr_base/sbin/useconfig install/configure_YYYYMMDDHHMMSS
```

configure_*YYYYMMDDHHMMSS* は以前の構成設定ファイルです。

新しいノードでは、共用ディスク上の *msg_svr_base*/data/setup ディレクトリに configure_*YYYYMMDDHHMMSS* があります。

89 ページの「Veritas Cluster Server エージェントのインストール」および 94 ページの「Sun Cluster エージェントのインストール」の以下の節には、useconfig ユーティリティをいつ使用できるかが記載されています。

Veritas Cluster Server エージェントのインストール

Messaging Server は、Veritas Cluster Server 1.3、2.0、および 3.5 とともに設定することができます。この節で示す手順では、Veritas Cluster 3.5 のみを取り上げています。Veritas 1.3 および 2.0 については、『Sun Java Enterprise System インストールガイド』(http://docs.sun.com/doc/819-0808?l=ja) を参照してください。

次の手順を実行する前に、Veritas Cluster Server マニュアルを再度お読みください。

注

- Veritas Volume Manager (VxVM) には、別のライセンスを必要とするクラスタ機能があります。この機能では、Sun Cluster 3.0 グローバルファイルシステム同様、共有ストレージ上のファイルシステムを概観できます。詳細は、Veritas Cluster Server のマニュアルを参照してください。
- `FsckOpt` は、3.5 より前の Veritas リリースではオプションでした。しかし、`Mount` リソースの設定には必須です。`FsckOpt` には `-y` または `-n` を付ける必要があります。そうしないと、リソースがオンラインになりません。
- Veritas Cluster Server 2.0 Explorer は、Veritas Cluster Server 3.5 の管理には使用できません。

Java Enterprise System インストーラからの Messaging Server のインストールと HA の設定が完了したあと、HA サポートに関連するその他の手順を 99 ページの「サーバー上での IP アドレスのバインド」で確認してください。

Veritas Cluster Server 要件

- Veritas Cluster Software は、あらかじめインストールされ設定されています。
- 次の手順 (90 ページの「VCS 3.5 インストールおよび設定上の注意」) で説明しているように、Messaging Server の Veritas Cluster Agent パッケージは、両方のノードで Messaging Server ソフトウェアとともにインストールします。

VCS 3.5 インストールおよび設定上の注意

次の手順では、Veritas Cluster Server 3.5 を使用して Messaging Server を HA サービスとして設定する方法を説明します。

デフォルトの `main.cf` 設定ファイルは、`VCSweb` アプリケーションを起動する `ClusterService` と呼ばれるリソースグループを設定します。このグループには、`csgnic` や `webip` などの、ネットワーク論理ホスト IP リソースが含まれます。また、`ntfr` リソースは、イベント通知用に作成されます。

1. ノードの 1 つから Cluster Explorer を起動します。

   この Veritas Cluster Server の手順では、Messaging Server を HA サービスとして設定する際にグラフィカルユーザーインタフェースを使用すると仮定しています。

Cluster Explorer を起動するには、以下のコマンドを実行します。

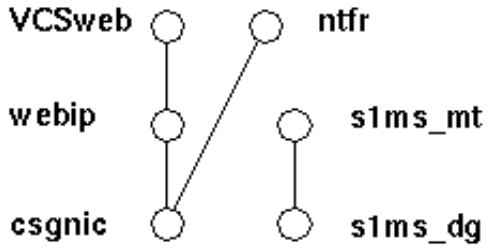
```
# /opt/VRTSvcs/bin/hagui
```

GUI を使用するために、VRTScscm パッケージをインストールする必要があります。

2. タイプ DiskGroup の s1ms_dg ディスクグループリソースを追加して、有効にします。
3. タイプ Mount の s1ms_mt マウントリソースを追加します。
   a. Veritas Cluster Server 2.0 とは異なり、-y (または -n) を FsckOpt に追加する必要があります。NULL オプションは、Mount をハングアップさせます。fsck_vxfs の詳細については、Solaris のマニュアルページを参照してください。
   b. リンクしているリソースがまだ有効になっていない場合は、「リンク」ボタンを使って、必ずそのリソースを有効にしてください。
4. s1ms_mt と s1ms_dg の間のリンクを作成します。リソース s1ms_mt を有効にします。

   次の図は依存関係ツリーを示します。

   図 3-1　Veritas Cluster Server 依存関係ツリー



5. 管理サーバーと Messaging Server を選択して、Java Enterprise System インストーラを実行します。
   a. 管理サーバーの設定中にホスト名の入力を求められたときは、必ず論理ホスト名を指定してください。

b. 主要ノード (たとえば、Node_A) から Messaging Server 初期実行時設定を実行して、Messaging Server をインストールします。

c. pkgadd(1M) コマンドを使用して、Veritas Cluster Server エージェントパッケージ、SUNWmsgvc (Java Enterprise System CD の Messaging Server Product サブディレクトリ内) をインストールします。

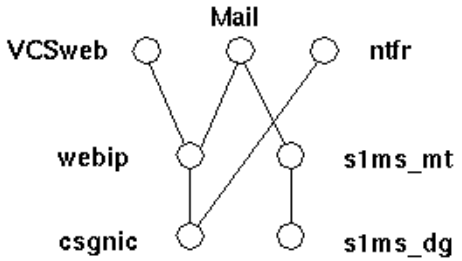
これで、Messaging Server と Veritas エージェントが Node_A にインストールされます。

6. バックアップノード (たとえば、Node_B) に切り替えます。
7. Java Enterprise System インストーラを実行して、バックアップノード (Node_B) に Messaging Server をインストールします。
8. Messaging Server をインストールしたあと、useconfig ユーティリティを使用することにより、バックアップノード (Node_B) に追加の初期実行時設定を作成する必要がなくなります。useconfig ユーティリティを使用することで、HA 環境の複数のノード間で単一の設定を共有することができます。このユーティリティは、既存の設定をアップグレードまたは更新するものではありません。88 ページの「useconfig ユーティリティの使用」を参照してください。

これで、Veritas エージェントが Node_B にインストールされます。

9. Cluster Explorer で、「File」メニューから「Import Types...」を選択し、ファイル選択ボックスを表示します。
10. /etc/VRTSvcs/conf/config ディレクトリから MsgSrvTypes.cf タイプをインポートします。このタイプファイルをインポートします。このファイルを検索するときは、クラスタノード上にいる必要があります。
11. ここで、タイプ MsgSrv のリソース (たとえば、Mail) を作成します。このリソースは、論理ホスト名プロパティの設定に必要です。
12. Mail リソースは、s1ms_mt および webip によって決まります。以下の依存関係ツリーに示されているように、リソース間のリンクを作成します。

図 3-2 Veritas Cluster 依存関係ツリー



a. すべてのリソースを有効にし、Mail をオンラインにします。

b. すべてのサーバーが起動されます。

13. Node_A に切り替えて、高可用性の設定が機能しているかどうかをチェックします。

14. グループ属性 OnlineRetryLimit を 3 から 0 に変更します。変更しないと、フェイルオーバーしたサービスが同じノード上で再起動することがあります。

MsgSrv 属性

この節では、mail リソースの動作を管理する MsgSrv のその他の属性について説明します。Veritas Cluster Server で Messaging Server を設定するには、表 3-2 を参照してください。

表 3-2 Veritas Cluster Server 属性

属性	説明
FaultOnMonitorTimeouts	設定解除 (=0) の場合、監視 (プローブ) のタイムアウトはリソースの障害としては扱われません。推奨される設定は 2。監視のタイムアウトが 2 回になると、リソースが再起動またはフェイルオーバーします。
ConfInterval	障害や再起動がカウントされる時間間隔。サービスがこの時間の間オンラインになっていると、以前の履歴は消去されます。600 ミリ秒を推奨。
ToleranceLimit	リソース FAULTED を宣言するために監視が OFFLINE を返す回数。この値は 0 (デフォルト) のままにします。

Sun Cluster エージェントのインストール

この節では、Sun Cluster の Highly Available (HA) Data Service のインストールおよび設定方法を説明します。このインストール手順は、Sun Cluster 3.1 に適用されます。この節には、以下の項目があります。

Sun Cluster 3.1 のマニュアルは、次の場所で参照できます。

`http://docs.sun.com/app/docs/prod/cluster?l=ja#hic`

Veritas File System (VxFS) は、Sun Cluster 3.1 でサポートされています。

Sun Cluster の要件

ここでは、次の条件を前提としています。

- Sun Cluster 3.1 が、必須パッチを含む Solaris 8 または 9 オペレーティングシステム上にインストールされ設定されています。
- Sun Cluster エージェント `SUNWscims` が、使用しているシステム上にインストールされています。
- 論理ボリュームを作成する場合、Solstice DiskSuite または Veritas Volume Manager のどちらかが使用されています。

HAStoragePlus について

Sun Cluster 環境内でローカルにマウントされたファイルシステムの可用性を高めるために、HAStoragePlus リソースタイプを使用することを強くお勧めします。Sun Cluster グローバルデバイスグループに常駐するファイルシステムは、HAStoragePlus で使用できます。HAStorage のようなグローバルにマウントされたファイルシステムとは異なり、HAStoragePlus は一定期間に 1 つのクラスタノードでのみ利用できます。これらのローカルにマウントされたファイルシステムは、フェイルオーバーモードとフェイルオーバーリソースグループのみで使用できます。HAStorage の GFS (グローバルファイルシステム) とは異なり、HAStoragePlus は FFS (フェイルオーバーファイルシステム) を提供します。

HAStoragePlus には、多くの利点があります。

- HAStoragePlus は、グローバルファイルサービスのレイヤーを完全にバイパスします。ディスク入出力が集中するデータサービスの場合、これによりパフォーマンスが大幅に向上します。
- HAStoragePlus は、グローバルファイルサービスのレイヤーでは動作しないものを含め、あらゆるファイルシステム (UFS、VxFS、など) で動作可能です。Solaris オペレーティングシステムでサポートされているファイルシステムであれば、HAStoragePlus で使用できます。

HAStoragePlus の詳細は、『Sun Cluster 3.1 Data Service Planning and Administration Guide』を参照してください。

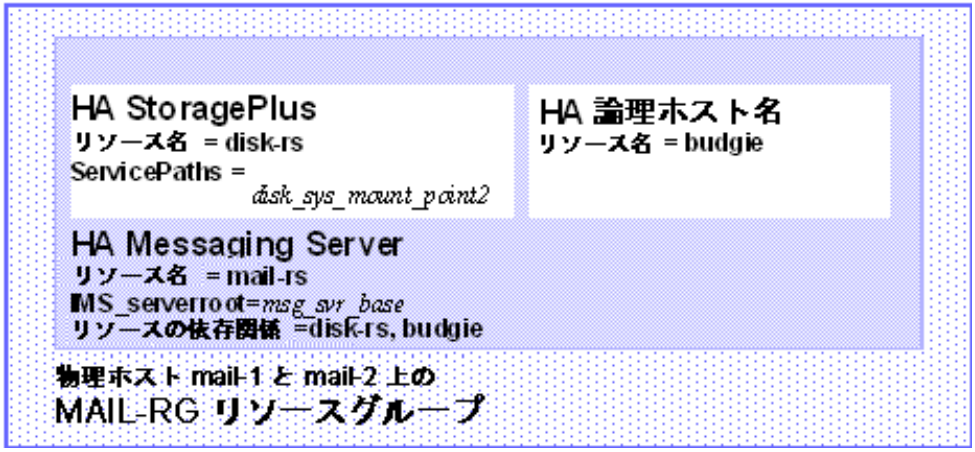
Sun Cluster と HA StoragePlus での Messaging Server の設定

この節では、Sun Cluster 3.1 の Sun Java System Messaging Server の HA サポートおよび HA StoragePlus を設定する方法を、単純な例を使って説明します。

HA を設定したら、99 ページの「サーバー上での IP アドレスのバインド」で、HA サポートに関連する追加の手順を確認してください。

以下の例では、メッセージングサーバーが HA 論理ホスト名および IP アドレスによって設定されていると仮定しています。物理ホスト名は `mail-1` と `mail-2` で、HA 論理ホスト名は `budgie` とします。図 3-3 に、Messaging Server HA サポートの構成時に作成する各種の HA リソースの入れ子の依存関係を示します。

図 3-3　単純な Sun Java System Messaging Server HA 構成



1. スーパーユーザーになり、コンソールを開きます。

   以下の Sun Cluster コマンドを実行するには、スーパーユーザーとしてログインする必要があります。また、メッセージ出力を表示するコンソールまたはウィンドウを /dev/console に設定する必要があります。

2. 必要なリソースタイプを追加します。

   使用するリソースタイプを Sun Cluster が認識できるように設定します。これを行うには、次のように、scrgadm -a -t コマンドを使用します。

   ```
   # scrgadm -a -t SUNW.HAStoragePlus
   # scrgadm -a -t SUNW.ims
   ```

3. Messaging Server のリソースグループを作成します。

   この作業をまだ実行していない場合は、リソースグループを作成し、Messaging Server を実行するクラスタノードにそのグループが表示されるようにします。次のコマンドは、MAIL-RG というリソースグループを作成し、クラスタノードの mail-1 および mail-2 にこのグループを表示します。

   ```
   # scrgadm -a -g MAIL-RG -h mail-1,mail-2
   ```

   リソースグループには、任意の名前を使用できます。

4. HA 論理ホスト名リソースを作成し、リソースグループを起動します。

   この作業をまだ実行していない場合は、HA 論理ホスト名リソースを作成して有効にし、これをリソースグループ内に配置します。次のコマンドは、論理ホスト名 budgie を使用して、これを実行します。-j オプションが省略されているので、作成したリソースの名前も budgie になります。

   ```
   # scrgadm -a -L -g MAIL-RG -l budgie
   # scswitch -Z -g MAIL-RG
   ```

5. HAStoragePlus リソースを作成します。

   Messaging Server が依存するファイルシステムの HAStoragePlus リソースタイプを作成する必要があります。次のコマンドは、disk-rs という HAStoragePlus リソースを作成し、ファイルシステム *disk_sys_mount_point* を、その制御下に配置します。

   ```
   # scrgadm -a -j disk-rs -g MAIL-RG ¥
   -t SUNW.HAStoragePlus ¥
   -x ServicePaths=disk_sys_mount_point-1, disk_sys_mount_point-2
   ```

   ServicePaths= の後ろに、Messaging Server が依存するクラスタファイルシステムのマウントポイントをコンマで区切って列挙します。上の例では、2 つのマウントポイント、*disk_sys_mount_point-1* と *disk_sys_mount_point-2* が指定されています。一方のサーバーが別のファイルシステムに依存する場合は、追加の HA ストレージリソースを作成し、手順 10 でその依存関係を指定します。

6. 管理サーバーをインストールして設定します (『Sun Java Enterprise System インストールガイド』を参照)。

   完全指定のドメイン名を指定するときは、手順 4 で作成した HA 論理ホスト名を使用してください。

7. Messaging Server をインストールして設定します。53 ページの「Messaging Server の初期実行時設定を作成するには」を参照してください。

   a. 初期実行時設定で、設定ディレクトリを指定するよう求められます (53 ページの「Messaging Server の初期実行時設定を作成するには」を参照)。必ず、HAStoragePlus リソースの共用ディスクのディレクトリパスを使用してください。

   b. 次のコマンドを実行して、Sun Cluster の下で watcher プロセスを有効にしてください。

```
configutil -o local.autorestart -v 1
```

   watcher プロセスの詳細については、113 ページの「障害が発生したサービスや応答がないサービスの自動再起動」を参照してください。

8. `ha_ip_config` スクリプトを実行して、`service.listenaddr` と `service.http.smtphost` を設定し、高可用性用に `dispatcher.cnf` および `job_controller.cnf` ファイルを設定します。このスクリプトでは、物理 IP アドレスではなく論理 IP アドレスがこれらのパラメータやファイルに設定されます。このスクリプトはまた、watcher プロセス (`local.watcher.enable` を 1 に設定) と自動再起動プロセス (`local.auto.restart` を 1 に設定) を有効にします。

   スクリプトの実行手順については、99 ページの「サーバー上での IP アドレスのバインド」を参照してください。

   `ha_ip_config` スクリプトは、(設定とデータ用の) 共用ディスクがあるマシン上で、1 回だけ実行する必要があります。

9. `imta.cnf` ファイルを変更して、出現するすべての物理ホスト名をクラスタの論理名に置き換えます。

10. HA Messaging Server リソースを作成します。

    HA Messaging Server リソースを作成し、これをリソースグループに追加します。このリソースは、HA 論理ホスト名リソースと HA ディスクリソースに依存します。

    HA Messaging Server リソースを作成するときは、Messaging Server のトップレベルディレクトリへのパス (*msg_svr_base* パス) を指定する必要があります。これには、次に示すように、`IMS_serverroot` 拡張プロパティを使用します。

```
# scrgadm -a -j mail-rs -t SUNW.ims -g MAIL-RG ¥
          -x IMS_serverroot=msg_svr_base ¥
          -y Resource_dependencies=disk-rs,budgie
```

上記のコマンドは、*msg_svr_base* ディレクトリ内の IMS_serverroot にインストールされている Messaging Server に、mail-rs という Messaging Server リソースを作成します。この HA Messaging Server リソースは、HA ディスクリソース disk-rs、および HA 論理ホスト名 budgie に依存します。

Messaging Server が追加のファイルシステムとの依存関係を持つ場合は、そのファイルシステム用に追加の HA ストレージリソースを作成できます。上記のコマンドの Resource_dependencies オプションに、追加する HA ストレージリソースの名前が含まれていることを確認してください。

11. /etc/vfstab ファイルから global という語を削除します。ブートアップでは、/etc/vbstab が「no」に設定されている必要があります。詳細については、Sun Cluster 3.1 のマニュアルを参照してください。

    HAStoragePlus で vfstab ファイルが有効になる前に、最初に、現在のグローバルファイルシステムであるファイルシステムを umount する場合があります。そのあと、HAStoragePlus で vfstab ファイルを有効にし、ファイルシステムを再マウントすることができます。

12. Messaging Server リソースを有効にします。

    HA Messaging Server リソースを有効にし、その Messaging Server をオンラインにします。これを実行するには、次のコマンドを使用します。

    ```
    # scswitch -e -j mail-rs
    ```

    このコマンドは、MAIL-RG リソースグループの mail-rs リソースを有効にします。MAIL-RG リソースはすでにオンラインになっているので、このコマンドで、mail-rs リソースもオンラインにします。

13. リソースの動作を確認します。

    scstat コマンドを使用して、MAIL-RG リソースグループがオンラインになっているかどうかを確認します。診断情報があれば、コンソールデバイスに出力されるので、画面で確認できます。また、syslog ファイル /var/adm/messages で参照することもできます。

14. フェイルオーバーを適切に動作させるため、もう 1 つのクラスタノードにリソースグループの処理を継続させます。

    手動でリソースグループの処理を別のクラスタノードに継続させます。フェイルオーバー先のノードでスーパーユーザー権限を持っていることを確認してください。

    scstat コマンドを使用して、現在リソースグループの処理を実行している (オンラインになっている) ノードを確認します。たとえば、オンラインノードが mail-1 の場合は、次のコマンドを使用して、mail-2 に処理を継続させます。

```
# scswitch -z -g MAIL-RG -h mail-2
```

最初のノードをアップグレードする場合は、Java Enterprise System インストーラからインストールを行なってから、Messagng Server を設定します。そのあと、Java Enterprise System インストーラを使って Messaging Server パッケージをインストールする 2 番目のノードにフェイルオーバーします。ただし、初期実行時設定プログラム (configure) をもう一度実行する必要はありません。代わりに、useconfig ユーティリティを使用することができます。

サーバー上での IP アドレスのバインド

「対称」高可用性モデルまたは「N + 1」高可用性モデルを使用する場合は、Sun Cluster Server を Messaging Server に対応させるために、構成で注意すべき事項がいくつかあります。

サーバー上で動作する Messaging Server は、正しい IP アドレスによってバインドされる必要があります。これは HA 環境で Messaging を正しく設定するために必要です。

Messaging Server を HA 対応に構成する過程で、Messaging Server がバインドされて接続を待機するインタフェースアドレスを設定します。デフォルトでは、各サーバーは使用可能なすべてのインタフェースアドレスにバインドされます。ただし、HA 環境では、HA 論理ホスト名に関連付けられたインタフェースアドレスに限定して各サーバーをバインドする必要があります。

上記のようなバインドが簡単に行えるように、特定の Messaging Server インスタンスに属するサーバーが使用するインタフェースアドレスの構成を行うためのスクリプトが用意されています。このスクリプトでは、ユーザーが所有する IP アドレス、またはサーバーが使用する HA 論理ホスト名に関連付ける IP アドレスから、適切なインタフェースアドレスを特定します。

このスクリプトは、以下の設定ファイルを修正または作成することによって、構成を変更します。

msg_svr_base/config/dispatcher.cnf

このファイルでは、SMTP サーバーおよび SMTP 送信サーバーの INTERFACE_ADDRESS オプションを追加または変更します。

msg_svr_base/config/job_controller.cnf

このファイルでは、ジョブコントローラの INTERFACE_ADDRESS オプションを追加または変更します。

最後に、POP、IMAP、および Messenger Express HTTP サーバーが使用する configutil service.listenaddr および service.http.smtphost パラメータを設定します。

元の設定ファイルがある場合、それらのファイルは *.pre-ha という名前に変更されます。

このスクリプトを実行するには、次の手順に従います。

1. スーパーユーザーになります。
2. *msg_svr_base*/sbin/ha_ip_config を実行します。
3. スクリプトによって、以下の質問が表示されます。Control キーを押しながら d キーを押すと、どの質問の段階でもスクリプトを中止できます。デフォルトの設定は、角括弧 ([ ]) 内に表示されています。デフォルトの設定を選択する場合は、Return キーを押します。

   a. Logical IP address: 論理 IP アドレス。Messaging Server が使用する論理ホスト名に割り当てられた IP アドレスを指定します。この IP アドレスは、「123.456.78.90」のように、ドット付きの 10 進形式で指定する必要があります。

      論理 IP アドレスは、configutil パラメータ service.http.smtphost に自動的に設定されます。このパラメータにより、クラスタ内でメッセージングシステムが実行されているマシンを参照することができます。たとえば Messenger Express を使用している場合、サーバーは、送信メールの送り先メールホストを判断できます。

   b. Messaging Server Base (*msg_svr_base*): Messaging Server をインストールする最上位ディレクトリの絶対パスを指定します。

   c. 選択した項目を変更するかどうか。これまでに回答した内容でよい場合は、「no」と答えて、設定の変更を確定します。回答を変更する場合は、「yes」と答えます。

注

また、ha_ip_config スクリプトでは、パラメータ local.autorestart および local.watcher.enable を使用して、2 つの新しいプロセス watcher および msprobe を自動的に有効にします。これらの新しいパラメータは、メッセージングサーバーの状況を監視する際に役立ちます。プロセスの障害や反応しないサービスによって、特定の障害を示すログメッセージが生成されます。これで、クラスタエージェントは watcher プロセスを監視するようになり、このプロセスが終了すると必ずフェイルオーバーします。Sun Cluster を正しく機能させるために、パラメータを有効にしておく必要があります。

watcher および msprobe プロセスの詳細は、113 ページの「障害が発生したサービスや応答がないサービスの自動再起動」を参照してください。

高可用性の構成の解除

この節では、高可用性の構成を解除する方法を説明します。高可用性をアンインストールするには、Veritas または Sun Cluster のマニュアルに記載されている手順に従ってください。

高可用性の構成の解除手順は、Veritas Cluster Server か Sun Cluster のどちらを削除するかによって異なります。

この節には、以下の項目があります。

Veritas Cluster Server の構成の解除

Veritas Cluster Server の高可用性コンポーネントの構成を解除するには、次の手順に従います。

1. `iMS5` サービスグループをオフラインにし、そのリソースを無効にします。
2. `mail` リソース、`logical_IP` リソース、および `mountshared` リソースの間の依存関係を解除します。
3. `iMS5` サービスグループをオンラインに戻します。`sharedg` リソースが有効になります。
4. インストール時に作成した Veritas Cluster Server リソースをすべて削除します。
5. Veritas Cluster Server を停止し、両方のノードで次のファイルを削除します。

   ```
   /etc/VRTSvcs/conf/config/MsgSrvTypes.cf
   /opt/VRTSvcs/bin/MsgSrv/online
   /opt/VRTSvcs/bin/MsgSrv/offline
   /opt/VRTSvcs/bin/MsgSrv/clean
   /opt/VRTSvcs/bin/MsgSrv/monitor
   /opt/VRTSvcs/bin/MsgSrv/sub.pl
   ```

6. Messaging Server のエントリを両方のノードの `/etc/VRTSvcs/conf/config/main.cf` ファイルから削除します。
7. 両方のノードから `/opt/VRTSvcs/bin/MsgSrv/` ディレクトリを削除します。

Sun Cluster 3. x の Messaging Server HA サポートの構成の解除

この節では、Sun Cluster の HA 構成を取り消す方法を説明します。ここでは、94 ページの「Sun Cluster エージェントのインストール」で説明した単純な例の構成を前提としています。ほかの構成では、特定のコマンド (たとえば、手順 3) が異なる場合がありますが、それ以外の手順は同じです。

1. スーパーユーザーになります。

   以下の Sun Cluster コマンドを実行するには、スーパーユーザーになる必要があります。

2. リソースグループをオフラインにします。

   リソースグループのすべてのリソースをシャットダウンするには、次のコマンドを実行します。

   ```
   # scswitch -F -g MAIL-RG
   ```

   これで、リソースグループ内のすべてのリソース (Messaging Server や HA 論理ホスト名など) がシャットダウンされます。

3. 個々のリソースを無効にします。

   次のコマンドで、リソースグループからリソースを 1 つずつ無効にします。

   ```
   # scswitch -n -j mail-rs
   # scswitch -n -j disk-rs
   # scswitch -n -j budgie
   ```

4. リソースグループから個々のソースを削除します。

   リソースを無効にしたら、次のコマンドで、リソースグループからリソースを 1 つずつ削除できます。

   ```
   # scrgadm -r -j mail-rs
   # scrgadm -r -j disk-rs
   # scrgadm -r -j budgie
   ```

5. リソースグループを削除します。

   リソースグループからすべてのリソースを削除したら、次のコマンドで、リソースグループそのものを削除できます。

   ```
   # scrgadm -r -g MAIL-RG
   ```

6. リソースタイプを削除します (省略可)。

   クラスタからリソースタイプを削除する必要がある場合は、次のコマンドを実行します。

```
# scrgadm -r -t SUNW.ims
# scrgadm -r -t SUNW.HAStoragePlus
```

第 4 章

一般的なメッセージング機能の設定

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、Sun ONE Server Console (以下、省略してコンソールという) またはコマンド行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。

パスワードを変更するには

初期設定 (53 ページの「Messaging Server の初期実行時設定を作成するには」を参照) で同じパスワードを持つ多数の管理者を設定しているため、これらの管理者のパスワードを変更したい場合もあります。

初期実行時設定の際にデフォルトのパスワードが設定されるパラメータと、それらを変更するためのユーティリティを確認するには、表 4-1 を参照してください。`configutil` ユーティリティを使用してパスワード変更するパラメータについては、『Sun ONE Messaging Server Reference Manual』で構文の詳細と使用方法を確認してください。

表 4-1 Messaging Server の初期実行時設定で設定されるパスワード

パラメータ	説明
`local.ugldapbindcred`	`configutil` ユーティリティを使って設定したユーザー / グループ管理者のパスワード
`local.service.pab.ldappasswd`	`configutil` ユーティリティを使って設定した、PAB 検索のバインド DN によって指定されたユーザーのパスワード
キーファイルの SSL パスワード	`sslpassword.conf` ファイル内に直接設定されているパスワード
サービス管理者の資格	`ldapmodify` コマンドを使って LDAP ディレクトリに直接設定されている資格
Delegated Administrator のサービス管理者	Sun LDAP Schema 1 が有効になっていて、iPlanet Delegated Administrator ユーティリティを使用している場合は、この管理者のパスワードを変更するだけで済みます。 Delegated Administrator サービス管理者のパスワードを変更するには、Sun ONE Console、LDAP ディレクトリ (`ldapmodify` コマンドで)、または Delegated Administrator の UI でパスワードを変更します。
ストア管理者	ストア管理者のパスワードを変更するには、Sun ONE Console または LDAP ディレクトリ (`ldapmodify` コマンドで) のどちらかでパスワードを変更できます。

以下の例では、`local.enduseradmincred configutil` パラメータを使用してエンドユーザー管理者のパスワードを変更します。

```
configutil -o local.enduseradmincred -v newpassword
```

メールユーザー、メーリングリスト、およびドメインを管理する

すべてのユーザー、メーリングリスト、およびドメインの情報は、LDAP ディレクトリ内に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、組織のユーザーとして扱われます。

LDAP ディレクトリ内のユーザー情報は、各ユーザー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。ユーザー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、直属の部下名、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ーがメールアカウントを持っているはずです。Messaging Server の場合、メールアカウント情報はサーバーにローカルには保存されません。これは、LDAP ユーザー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ー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。

メールユーザー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ー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。これを行うには、Sun LDAP Schema 2 対応の Delegated Administrator およびメッセージング用 iPlanet Delegated Administrator (Sun LDAP Schema 1 対応) の Delegated Administrator コマンド行ユーティリティを使うか、Sun LDAP Schema 1 の LDAP ディレクトリを直接変更します。

Messaging Server からユーザーを削除するには

1. `commadmin user delete` コマンドを実行して、ユーザーを削除済みとマークします (『Sun Java System Communications Services Delegated Administrator 管理ガイド』(http://docs.sun.com/doc/819-1101?l=ja) を参照)。
2. ユーザーからサービスを削除します。

   サービスとしては、メールボックスやカレンダなどがあります。Messaging Server の場合、このプログラムの名前は `msuserpurge` です (『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) を参照)。カレンダサービスの場合、このプログラムの名前は `csclean` です (『Sun Java System Calendar Server 管理ガイド』(http://docs.sun.com/doc/819-1476?l=ja) を参照)。
3. `commadmin domain purge` コマンドを呼び出して、ユーザーを永久に削除します。

Messaging Server からドメインを削除するには

1. commadmin domain delete コマンドを実行して、ドメインを削除済みとマークします (『Sun Java System Communications Services Delegated Administrator 管理ガイド』(http://docs.sun.com/doc/819-1101?l=ja) を参照)。
2. そのドメインのユーザーからサービスを削除します。

   サービスとしては、メールボックスやカレンダなどがあります。Messaging Server の場合、このプログラムの名前は msuserpurge です (『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) を参照)。カレンダサービスの場合、このプログラムの名前は csclean です (『Sun Java System Calendar Server 管理ガイド』(http://docs.sun.com/doc/819-1476?l=ja) を参照)。
3. commadmin domain purge コマンドを呼び出して、ドメインを永久に削除します。

Sun ONE Console を使った Messaging Server の管理

Messaging Server のインストールプロセスと初期実行時設定プログラムが完了すると、管理コンソールから Messaging Server を起動することができます。ディレクトリサーバーとメッセージングサーバーが同じマシンに存在する場合は、Console インタフェースを使用して両方のサーバーを管理できます。

コンソールを起動するには、/var/opt/mps/serverroot/start console コマンドを実行します。

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、Sun ONE Server Console を使って情報フォームを表示します。

情報フォームを表示するには、次の手順に従います。

1. コンソールで、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ーのアイコンを選択します。
3. 左側のペインの「設定」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、サーバー名、サーバー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

サービスを起動および停止する方法は、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。HA 配備で `stop-msg` を試みると、HA 設定が検出されたという警告と適切なシステムの停止方法が示されます。

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。ほかの Messaging Server サービス (たとえば、SMTP) を個別に起動、再起動、停止するための特定の HA コマンドはないことに注意してください。ただし、`stop-msg` *service* コマンドを実行して、`imap`、`pop`、`sched` などの個々のサーバーを停止または再起動することはできます。

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 4-2 Sun Cluster 3.0/3.1 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	`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 4-3 Veritas 1.3、2.0、2.1、および 3.5 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、コンソールまたはコマンド行を使って起動および停止できます。ほかに必要な操作は、サーバー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、特定の Messaging Server 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、保守、修復、セキュリティ上の必要からサーバー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、HTTP など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、126 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 : サーバープロセスがクラッシュすると、ほかのプロセスがハングアップ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ー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。自動再起動 (113 ページの「障害が発生したサービスや応答がないサービスの自動再起動」を参照) を使用していない場合で、サーバー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 : コンソールでは、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したりできます。

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が表示されます。このフォームでは、その他のステータス情報も表示できます。

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. コンソールで、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. 「設定」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、サーバー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、サーバー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、サーバー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。

コマンド行 : `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 (`smtp`、`imap`、`pop`、`store`、`http`、`ens`、`sched`) を起動または停止できます。次に例を示します。

msg_svr_base`/sbin/start-msg imap`
msg_svr_base`/sbin/stop-msg pop`
msg_svr_base`/sbin/stop-msg sched`
msg_svr_base`/sbin/stop-msg smtp`

サービスを停止または起動するには、サービスは有効になっている必要があります。112 ページの「起動するサービスを指定するには」を参照してください。

注	`start-msg smtp` および `stop-msg smtp` コマンドを実行すると、SMTP サーバーだけでなく、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、ディスパッチャおよびジョブコントローラに対して start/stop msg コマンドを使用します。詳細は、『Messaging Server Reference Manual』を参照してください。

起動するサービスを指定するには

デフォルトでは、start-msg を使って次のサービスが起動されます。

```
# ./start-msg
Connecting to watcher ...
Launching watcher ...
Starting ens server .... 21132 21132
Starting store server .... 13457 13457 21133
checking store server status ... ready
Starting imap server .... 21135 21135
Starting pop server .... 21138 21138
Starting http server .... 21141 21141
Starting sched server .... 21143 21143
Starting dispatcher server .... 21144 21144
Starting job_controller server .... 21146 21146
```

これらのサービスは、次の configutil パラメータを有効化または無効化することによって制御できます。service.imap.enable、service.pop.enable、service.http.enable、local.smsgateway.enable、local.snmp.enable、local.imta.enable、local.mmp.enable、local.ens.enable、および local.sched.enable。IMAP を無効にするには、service.imap.enable と service.imap.enablesslport の両方を 0 に設定する必要があります。POP および HTTP の場合も同様です。これらのパラメータの機能の詳細については、『Sun Java System Messaging Server Administration Reference』を参照してください。

障害が発生したサービスや応答がないサービスの自動再起動

Messaging Server では、watcher と msprobe の 2 つのプロセスが提供されています。これらのプロセスによってサービスは透過的に監視され、クラッシュしたり応答がなくなった (ハングアップしている) 場合には、自動的に再起動されます。watcher はサーバークラッシュを監視し、msprobe は応答時間をチェックすることでサーバーハングアップを監視します。サーバーでエラーが発生した場合や、サーバーが要求に応答しなくなった場合、サーバーは自動的に再起動されます。表 4-4 を参照。

表 4-4 watcher と msprobe で監視されるサービス

watcher (クラッシュ)	msprobe (応答しないハングアップ)
IMAP、POP、HTTP、ジョブコントローラ、ディスパッチャ、メッセージストア (stored)、imsched、MMP (LMTP/SMTP サーバーはディスパッチャによって監視され、LMTP/SMTP クライアントは job_controller によって監視される)。	IMAP、POP、HTTP、証明書、ジョブコントローラ、メッセージストア (stored)、imsched、ENS、LMTP、SMTP

local.watcher.enable=on (デフォルト) を設定すると、プロセスの失敗と応答しないサービスが監視され、default ログファイルに特定の失敗を示すエラーメッセージが記録されます。サーバーの自動再起動を有効にするには、configutil のパラメータ local.autorestart を yes に設定します。デフォルトでは、このパラメータは no に設定されています。

メッセージストアのサービスのどれかが失敗またはフリーズした場合、起動時に有効にしたすべてのメッセージストアのサービスが再起動されます。たとえば、imapd が失敗すると、少なくとも stored および imapd が再起動されます。POP または HTTP サーバーなど、メッセージストアのほかのサービスが実行されている場合、それらのサービスも失敗や成功にかかわらず再起動されます。

自動再起動は、メッセージストアユーティリティが失敗またはフリーズした場合にも機能します。たとえば、mboxutil が失敗またはフリーズした場合、すべてのメッセージストアサービスが再起動されます。ただし、ユーティリティは再起動されません。msprobe は 10 分ごとに実行されています。サービスとプロセスの再起動は 10 分間に最大 2 回実行されます (local.autorestart.timeout を使用して設定可能)。

local.autorestart が yes に設定されているかどうかにかかわらず、サービスはシステムによって監視され、失敗または応答なしのエラーメッセージがコンソールおよび *msg_svr_base*/data/log/watcher に送信されます。watcher はデフォルトではポート 49994 を待機しますが、これは local.watcher.port を使って設定可能です。

watcher ログファイルが *msg_svr_base*/data/log/watcher に生成されます。このログファイルは、ログが記録されるシステムでは管理されません (ロールオーバーおよびパージは行われない)。このログファイルにはすべてのサーバーの起動と停止が記録されます。ログの例を次に示します。

```
watcher process 13425 started at Tue Oct 21 15:29:44 2003

Watched 'imapd' process 13428 exited abnormally
Received request to restart:  store imap pop http
Connecting to watcher ...
Stopping http server 13440 .... done
Stopping pop server 13431 ... done
Stopping pop server 13434 ... done
Stopping pop server 13435 ... done
Stopping pop server 13433 ... done
imap server is not running
Stopping store server 13426 .... done
Starting store server .... 13457 13457 13457
checking store server status ...... ready
Starting imap server ..... 13459 13459 13459
Starting pop server ....... 13462 13462 13462
Starting http server ...... 13471 13471 13471
```

この機能の設定方法の詳細については、859 ページの「msprobe および watcher 関数を使用した監視」を参照してください。

msprobe は imsched によって制御されます。imsched がクラッシュすると、このイベントは watcher によって検出され、再起動がトリガされます (自動再起動が有効になっている場合)。ただし、ごくまれに imsched がハングアップした場合は、kill *imsched_pid* を使用して imsched を強制終了する必要があります。これにより、imsched が watcher によって再起動されます。

高可用性の配備での自動再起動

可用性が高い配備での自動再起動には、次の configutil パラメータを設定する必要があります。

表 4-5　HA 自動再起動パラメータ

パラメータ	説明 /HA 値
local.watcher.enable	watcher の有効化。On (デフォルトは On)
local.autorestart	autorestart の有効化。On
local.autorestart.timeout	再試行失敗のタイムアウト。ここに指定した時間内でサーバーに 3 回以上障害が発生すると、システムはサーバーの再起動を試行しなくなります。HA システムでこれが発生すると、Messaging Server がシャットダウンし、別のシステムへのフェイルオーバーが行われます。値 (秒単位で指定) は、msprobe の間隔 (local.schedule.msprobe) よりも長い時間に設定する必要があります。
local.schedule.msprobe	msprobe の実行スケジュール。crontab 形式でスケジュールを示す文字列 (617 ページの表 18-10 を参照)。デフォルトは 600 秒

自動タスクをスケジュールするには

Messaging Server は、imsched というプロセスを使って一般的なタスクスケジュールを行うメカニズムを提供します。これは、Messaging Server のプロセスをスケジュールするためのものです。Messaging Server 以外のタスクのスケジューリングはサポートされていません。これは、local.schedule.*taskname* configutil パラメータを設定して有効にします。スケジュールを変更している場合は、コマンド stop-msg sched および start-msg sched を使用してスケジューラを再起動するか、refresh sched でスケジューラプロセスを更新する必要があります。

パラメータには、コマンドとコマンドを実行するスケジュールが必要です。形式は次のとおりです。

```
configutil -o local.schedule.taskname -v "schedule"
```

taskname は、このコマンドとスケジュールの組み合わせを示す一意の名前です。

schedule は次の形式をとります。

minute hour day-of-month month-of-year day-of-week command args

command args (コマンド引数) は、Messaging Server の任意のコマンドとその引数をとることができます。コマンドのパス名は、完全指定でなければなりません。

minute hour day-of-month month-of-year day-of-week (分 時 日付 月 曜日) は、コマンドを実行するスケジュールです。UNIX の `crontab` の書式に従います。

値は空白文字またはタブ文字で区切られ、値の範囲は、分は 0 ～ 59、時は 0 ～ 23、日付は 1 ～ 31、月は 1 ～ 12、曜日は 0 ～ 6 (0= 日曜日) となります。各時間フィールドには、アスタリスク (すべての取りうる値)、コンマ区切りの値のリスト、またはハイフンで区切られた 2 つの値による範囲を使用することもできます。日は日付と曜日の両方を使用して指定します。両方の使用を指定した場合には、どちらの値も必要です。たとえば、月の 17 日目と火曜日を設定すると、コマンドは、火曜日で、かつ 17 日である場合だけ実行されます。スケジュールのパラメータの設定方法例については、617 ページの表 18-10 を参照してください。

スケジューラを変更している場合は、コマンド `stop-msg sched` および `start-msg sched` を使用してスケジューラを再起動するか、SIGHUP をスケジューラプロセスに送信する必要があります。

```
kill -HUP scheduler_pid
```

スケジューラの例

次の例では、冗長モードで `imexpire` を 12:30am、8:30am、および 4:30pm に実行します。

```
configutil -o local.schedule.rm_messages -v "30 0,8,16 * * *"
/opt/SUNWmsgsr/sbin/imexpire -v
```

次の例では、MTA チャネルキューのメッセージカウンタを 20 分おきに表示します。

```
configutil -o local.schedule.counters -v "20,40,60 * * * *"
/opt/SUNWmsgsr/sbin/imsimta qm counters -show > temp.txt
```

次の例では、`imsbackup` を月曜日から金曜日の真夜中 (12 am) に実行します。

```
configutil -o local.schedule.msbackup -v "0 0 * * 1-5"
/opt/SUNWmsgsr/sbin/imsbackup -f backupfile /primary
```

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、新規ユーザーに送る電子メールグリーティングメッセージを作成できます。

コンソール : コンソールを使って新規ユーザー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. コンソールで、新規ユーザーへのグリーティングを設定する Messaging Server を開きます。
2. 「設定」タブをクリックします。左側のペインでサーバー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、新規ユーザー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。
5. 保存を完了します。

コマンド行 : コマンド行を使って新規ユーザー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 Message
```

Message には少なくとも件名行を含むヘッダーがあり、$$、メッセージ本文がその後に続いている必要があります。$ は、新しい行を表します。

たとえばこのパラメータを有効にするため、次のように設定変数を設定することができます。

```
configutil -o gen.newuserforms - v 'Subject:Welcome!!$$ Sesta.com
welcomes you to the premier internet experience in Dafandzadgad!
```

使用しているシェルによっては、$ の前に特殊文字を追加して、$ が持つ特殊な意味をエスケープする必要があることもあります (ほとんどの場合、$ はシェルのエスケープ文字)。

ドメイン単位のグリーティングメッセージを設定するには

新規のホストしているドメインを作成する場合は常に、サポートされている言語のドメイン単位のグリーティングメッセージを作成することをお勧めします。これを行わない場合は、gen.newuserforms によって設定されている一般的なグリーティングメッセージが送信されます。

新規ユーザーへのグリーティングメッセージは、ドメインごとに設定できます。メッセージは、ユーザー、ドメイン、またはサイトの優先言語に応じて変えることができます。これを行うには、対象の LDAP ドメインエントリの mailDomainWelcomeMessage 属性を設定します。属性の構文は次のとおりです。

mailDomainWelcomeMessage;lang-*user_prefLang*
mailDomainWelcomeMessage;lang-*domain_prefLang*
mailDomainWelcomeMessage;lang-*gen.sitelanguage*

次の例では、英語のドメインのグリーティングメッセージが設定されています。

```
mailDomainWelcomeMessage;lang-en:Subject:Welcome!!$$Welcome to the
mail system.
```

次の例では、フランス語のドメインのグリーティングメッセージが設定されています。

```
mailDomainWelcomeMessage;lang-fr:Subject:Bienvenue!!$$Bienvenue a
siroe.com!
```

上記の例から、次のことを仮定します。

- ドメインは siroe.com である
- 新規ユーザーはこのドメインに所属している
- LDAP 属性 preferredlanguage で指定されているように、ユーザーが優先する言語はフランス語である
- siroe.com では、上記の英語およびフランス語のグリーティングメッセージが使用可能である
- gen.sitelanguage で指定されているように、サイト言語は en である

サポートされるロケールおよびその言語値タグの一覧は、『Directory Server Reference Manual』(http://docs.sun.com/source/816-6699-10/ax_inter.html#18744) を参照してください。

ユーザーは、初めてログインしたとき、フランス語のグリーティングメッセージを受信します。フランス語のグリーティングメッセージが使用不可の場合、英語のグリーティングメッセージを受信します。

グリーティングメッセージの動作方式

グリーティングメッセージは、LDAP 属性 mailDomainWelcomeMessage と configutil パラメータ gen.newuserforms の両方によって設定されます。メッセージが選択される順序を、優先順位の高い順に次に示します。

mailDomainWelcomeMessage;lang-*user_prefLang*
mailDomainWelcomeMessage;lang-*domain_prefLang*
mailDomainWelcomeMessage;lang-*gen.sitelanguage*
mailDomainWelcomeMessage
gen.newuserforms;lang-"$user-*prefLang*"
gen.newuserforms;lang-"$domain-*prefLang*"
gen.newuserforms;lang-"$gen.*sitelanguage*"
gen.newuserforms

アルゴリズムは次のように機能します。ドメインが存在しない場合 (または存在してもドメイン単位のグリーティングメッセージが提供されない場合)、gen.newuserforms パラメータが指定されていれば、このパラメータを使ってグリーティングメッセージが設定されます。ユーザーに優先言語があり (preferredlanguage LDAP 属性で設定)、gen.newuserforms;lang-*user_prefLang* が設定されていれば、ユーザーはサーバーに最初にログインしたときにグリーティングメッセージを受信します。gen.newuserforms;lang-*gen.sitelanguage* が設定されていて、preferredlanguage が設定されていない場合で、サイト言語が設定 (gen.sitelanguage パラメータを使用) されている場合、ユーザーはメッセージを受信します。言語タグのパラメータが設定されていない場合は、タグなしの gen.newuserforms が設定されていれば、そのメッセージがユーザーに送信されます。いずれの値も設定されていない場合は、ユーザーはグリーティングメッセージを受信しません。

ユーザーがドメインに所属している場合は、上記の説明と同様に、ユーザーは mailDomainWelcomeMessage;lang-xx のうちのいずれかを受信します。受信するメッセージは、どのメッセージがリストおよび所定の順序で使用可能であるかによって異なります。

次に例を示します。ドメインは fantasia.com で、ドメインの優先言語はドイツ語 (de) です。しかし、このドメインの新規ユーザーの優先言語はトルコ語 (tr) です。サイト言語は英語です。使用できる値は次のとおりです (mailDomainWelcomeMessage は、ドメイン fantasia.com の属性)。

mailDomainWelcomeMessage;lang-fr
mailDomainWelcomeMessage;lang-ja
gen.newuserforms;lang-de
gen.newuserforms;lang-en
gen.newuserforms

アルゴリズムに従って、ユーザーに送信されるメッセージは gen.newuserforms;lang-de になります。

ユーザーの優先言語を設定するには

管理者は、ユーザーの LDAP エントリの属性 preferredLanguage を設定することで、GUI およびサーバーで生成されるメッセージの優先言語を設定できます。

サーバーの管理ドメイン外のユーザーにメッセージを送信する場合、サーバーはそのユーザーの優先言語は判断できません。ただし、その着信メッセージが、ヘッダーに優先言語が指定された着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、ユーザー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 ヘッダーフィールド
2. 元のメッセージの Preferred-Language ヘッダーフィールド
3. 元のメッセージの X-Accept-Language ヘッダーフィールド
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、ドメインの LDAP エントリの属性 preferredLanguage を設定することでドメインの優先言語を設定できます。

サーバーサイト言語を設定するには

以下の手順に従って、サーバーのデフォルトサイト言語を指定できます。ユーザー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 : コンソール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. 「設定」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. 保存を完了します。

コマンド行 : 次に示すように、コマンド行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いずれかの言語を指定できます。サポートされるロケールおよびその言語値タグの一覧は、『Directory Server Reference Manual』を参照してください (http://docs.sun.com/source/816-6699-10/ax_inter.html#18744)。

ディレクトリ検索をカスタマイズするには

Messaging Server は、Sun Java System 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 およびコンソールには、多く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。次に例を示します。

- Messaging Server をはじめてインストールする際に、サーバーの構成設定を入力します。これらの設定は、中央の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。
- メールユーザー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、その情報はユーザーディレクトリと呼ばれるディレクトリに保存されます。サーバーグループの管理サーバーはインストール時に設定されています。この設定によって、ユーザーやグループにアクセスしたとき、コンソールは管理トポロジが定義されている設定ディレクトリにデフォルトで接続します。「管理トポロジ」とは、同じ設定ディレクトリおよびユーザーディレクトリを共有する Sun Java System サーバーの集まりのことです。
- 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 はユーザー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 は管理サーバーが使用するのと同じユーザーディレクトリ内を検索します。
- メールルーティングの検索のためにユーザーの認証を行います。

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。

- コンソールの「管理サーバー」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 (詳細は、『Sun ONE Server Console 5.2 Server Management Guide』の管理サーバーに関する章を参照)。
- ユーザーやグループの情報を変更する場合は、コンソールの「ユーザーおよびグループ」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ーディレクトリに一時的に接続することができます (詳細は、『Sun ONE Server Console 5.2 Server Management Guide』のユーザーとグループに関する章を参照)。

- コンソールの「Messaging Server」インタフェースを使用すると、管理サーバーで定義されているデフォルトとは別のユーザー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。

別のユーザーディレクトリに接続してユーザー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、サーバーの管理ドメインを定義しているユーザーディレクトリがドメイン内のすべてのサーバー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ーディレクトリを指定した場合は、コンソールの「ユーザーおよびグループ」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ー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。

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

1. コンソールから、LDAP 接続をカスタマイズする Messaging Server を開きます。
2. 「設定」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ー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、『Sun ONE Server Console 5.2 Server Management Guide』の管理サーバーに関するを参照してください。
5. ユーザーディレクトリの接続設定を変更するには、「Messaging Server 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、『Directory Server 管理ガイド』を参照)。

   ホスト名: インストールのユーザー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 がユーザー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 (ユーザー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ー部分に対する検索特権がある、ユーザー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. ユーザー検索のために 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、サーバー構成に保存されているパスワードは更新されますが、LDAP サーバー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ーのパスワードを変更します。このユーザーアカウントは、設定属性 `local.ugldaphost` に指定されているディレクトリサーバー内にあります。

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ーディレクトリに戻るには、「Messaging Server 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。

コマンド行 : 次に示すように、コマンド行でユーザーディレクトリの接続設定の値を設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。

Messaging Server 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 ]
```

ユーザー検索用の 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[:port_number]
```

ユーザー検索用の 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
```

ユーザー検索用の 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
```

ユーザー検索用の 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

コンソールを使用すると、Messaging Server の SSL (Secure Sockets Layer) 暗号化および認証を有効にしたり、サーバー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 19 章「セキュリティとアクセス制御を設定する」の「SSL を有効にし暗号化方式を選択するには」の節で説明します。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も記載されています。

LDAP サーバーフェイルオーバーを設定する

複数の LDAP サーバーをユーザーまたはグループディレクトリとして指定することができます。これによって、1 つのサーバーに障害が発生しても別のサーバーが処理を引き継ぎます。

1. `local.ugldaphost` を複数の LDAP マシンに設定します。次に例を示します。

   ```
   configutil -o local.ugldaphost -v "server1 server2 ..."
   ```

2. `local.ugldapuselocal` を `yes` に設定します。これによって、ユーザーまたはグループの LDAP 設定データはローカル設定ファイルに保存されます。これ以外の場合は、LDAP に保存されます。次に例を示します。

   ```
   configutil -o local.ugldapuselocal -v yes
   ```

リストにある最初のサーバーに障害が発生した場合、既存の LDAP 接続はダウンしたとみなされ、新しい接続が確立されます。新規の LDAP 接続が必要な場合、LDAP ライブラリはすべての LDAP サーバーをリストされている順序で試します。

ユーザーまたはグループディレクトリ用のフェイルオーバーと同様に、設定ディレクトリ用のフェイルオーバーサーバーを設定することもできます。設定属性は `local.ldaphost` です。

第 5 章

POP、IMAP、および HTTP サービスの設定

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、エンドユーザーは 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、Sun ONE Console またはコマンド行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サーバーを構成する方法について説明します。

Simple Mail Transfer Protocol (SMTP) サービスの設定については、第 10 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見出しの修正 (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手順については、135 ページの「POP サービスを設定するには」、137 ページの「IMAP サービスを設定するには」、および 139 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 (109 ページの「サービスを起動および停止する」を参照)。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」したサービスは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、サーバー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。また、112 ページの「起動するサービスを指定するには」でも説明されています。特定の LDAP 属性 `mailAllowedServiceAccess` を設定することにより、ユーザー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。

ポート番号を指定する

各サービスに対して、サーバー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、サーバーが 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、サーバーが 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、サーバーが 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 サーバーインスタンスがある場合や、同じホストマシンを IMAP サーバーおよび Messaging Multiplexor サーバー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 7 章「マルチプレクササービスの設定および管理」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、POP、および 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 サポートの詳細については、679 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 (推奨) 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 (681 ページの「管理コンソールからの証明書の入手」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 POP

POP 用の個別 SSL ポートは、デフォルトでは 995 です。STLS コマンドを使用して、通常の POP ポートで SSL を開始することもできます (135 ページの「POP サービスを設定するには」を参照)。

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 HTTPS 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、サーバー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ーには表示されない) は、サーバーが Sun Java System Messaging Server であることを証明するもので、そこにはサーバー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ほか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。

Sun ONE 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` の構文の詳細については、『Sun Java System Messaging Server Administration Reference』(`http://docs.sun.com/doc/819-0106`) を参照してください。

ログインの要件

ユーザーは 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。このユーザー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、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手順については、135 ページの「POP サービスを設定するには」、137 ページの「IMAP サービスを設定するには」、および 139 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

メールクライアントによっては、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@domain と似ているからです。これらのクライアントの例には、Microsoft 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ーに @ 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。

ドメイン名を使用せずにログインを許可するには

標準のログインでは、ユーザーがユーザー ID、区切り文字、ドメイン名の順に入力し、次にパスワードを入力する必要があります。ただし、インストール時に指定したデフォルトのドメインのユーザーは、ドメイン名や区切り文字を入力しなくてもログインできます。

ほかのドメインのユーザーもユーザー ID だけでログインできるようにする (ドメイン名と区切り文字を不要にする) には、sasl.default.ldap.searchfordomain を 0 に設定します。ユーザー ID はディレクトリツリー全体で一意になるようにする必要があります。ユーザー ID が一意でない場合、ドメイン名を指定せずにログインすると失敗します。

ログイン時にユーザーが入力する必要がある属性を変更することもできます。たとえば、電話番号 (telephoneNumber) や従業員番号 (employeeID) を入力させたい場合は、configutil のパラメータ sasl.default.ldap.searchfilter で定義されている LDAP 検索を変更します。このパラメータは、ドメイン単位の inetdomainsearchfilter 属性のグローバルなデフォルト設定であり、同じ構文に従っています。

これらのパラメータの詳細については、『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) を参照してください。

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、ユーザーは POP、IMAP、または HTTP メールクライアントにパスワードを入力して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ーに送信すると、サーバーはそのパスワードを使ってユーザーを認証します。ユーザーが認証されると、アクセス制御ルールに基づき、そのサーバー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、ユーザー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、パスワードまたは SSL 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、サーバー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暗号化方式を使って、サーバーとの 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 のユーザー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ーのサポートする最大長より大きい場合、またはサーバー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。さまざまな暗号化方式とキー長をサポートするようにサーバーを設定する方法については、687 ページの「SSL を有効化し暗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

Sun Java System サーバーでは、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ー認証もサポートしています。サーバーとの 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ーの証明書を提示します。証明書の信頼性が確認されると、そのユーザーは認証済み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、689 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なった場合は、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ー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、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手順については、135 ページの「POP サービスを設定するには」、137 ページの「IMAP サービスを設定するには」、および 139 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ーマシンにおいて特に効果があります。多くのサーバー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、サーバー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、サーバーマシンの入出力サブシステムに負担をかけるため、実際にサーバー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。サーバー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ーが IMAP に接続すると、接続は通常ユーザーが終了するか、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ー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ー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 や 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させます。Messaging Server 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、673 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ーの要求により、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ー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。

通常、アクティブな POP 接続では、サーバーリソースと帯域幅について、アクティブな IMAP 接続よりもはるかに厳しい要件が求められます。これは、POP 接続が継続的にメッセージをダウンロードしている間、IMAP 接続はほとんどずっとアイドル状態だからです。POP のセッション数は、少なくすることをお勧めします。逆に言えば、POP 接続は電子メールのダウンロードに要する時間しか継続しないため、アクティブな POP ユーザーはその時間のほんの数パーセントしか接続できません。これに対し、IMAP 接続は一連のメールチェックの間ずっと接続状態です。

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。サーバー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、実際にサーバー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ー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、サーバー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、ユーザーは新たに接続するときに再び認証する必要があります。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、673 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アップするなど) 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 ユーザー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 サーバー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、サーバーは 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、再び認証し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、673 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

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、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、695 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 710 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 Sun ONE 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。完全なリストは、『Sun Java System Messaging Server Administration Reference』にあります。

詳細は、以下を参照してください。

- 126 ページの「サービスの有効化と無効化」
- 129 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」
- 126 ページの「ポート番号を指定する」
- 132 ページの「プロセス当たりの接続数」
- 134 ページの「アイドル接続を切断する」
- 133 ページの「プロセス当たりのスレッド数」
- 131 ページの「プロセス数」

コンソール : コンソールを使用して POP サービスを設定するには、次の手順に従います。

1. 構成を行う Messaging Server を Sun ONE Console から開きます。
2. 「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、「次のポートで POP サービスを有効」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、132 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、134 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、133 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、131 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. 保存を完了します。

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。

コマンド行 : 次に示すように、コマンド行から 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 number
```

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 number
```

接続の最大アイドル時間を設定する

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

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 number
```

最大プロセス数を設定する

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

SSL を使用した POP を有効にする

```
configutil -o service.pop.enablesslport -v 1configutil -o
service.pop.sslport -v 995
```

SSL が正しく設定されている場合、TLS もサポートされます。

プロトコルにようこそ見出しを指定する

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

IMAP サービスを設定するには

`configutil` コマンドまたは Sun ONE 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、『Sun Java System Messaging Server Administration Reference』にあります。詳細は、以下を参照してください。

- 126 ページの「サービスの有効化と無効化」
- 126 ページの「ポート番号を指定する」
- 130 ページの「パスワードに基づくログイン」
- 132 ページの「プロセス当たりの接続数」
- 134 ページの「アイドル接続を切断する」
- 133 ページの「プロセス当たりのスレッド数」
- 131 ページの「プロセス数」

コンソール：コンソールを使用して IMAP サービスを設定するには、次の手順に従います。

1. 構成を行う Messaging Server を Sun ONE Console から開きます。
2. 「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、「次のポートで 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、132 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、134 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、133 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、131 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. 保存を完了します。

コマンド行 : 次に示すように、コマンド行から 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 number
```

「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 number
```

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.imap.plaintextmincipher -v value
```

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 number
```

接続の最大アイドル時間を設定する

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

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 number
```

最大プロセス数を設定する

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

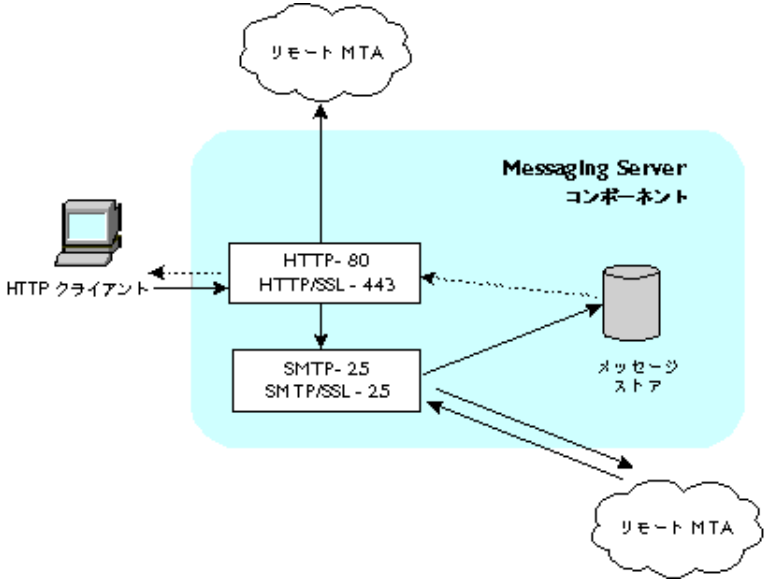
プロトコルにようこそ見出しを指定する

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

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 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 Messaging Server の一部である特殊な Web サーバーに送信します。その後、HTTP サービスは、図 5-1 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。Messaging Server を 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。

図 5-1　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。完全なリストは、『Sun Java System Messaging Server Administration Reference』にあります。詳細は、以下を参照してください。

- 126 ページの「サービスの有効化と無効化」
- 126 ページの「ポート番号を指定する」

- 130 ページの「パスワードに基づくログイン」
- 132 ページの「プロセス当たりの接続数」
- 134 ページの「アイドル接続を切断する」
- 134 ページの「HTTP クライアントをログアウトする」
- 133 ページの「プロセス当たりのスレッド数」
- 131 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 サービスに特有なものもあります。

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ー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 : Sun ONE Console を使用して HTTP サービスを設定するには、次の手順に従います。

1. 構成を行う Messaging Server を Sun ONE Console から開きます。
2. 「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、「次のポートで HTTP サービスを有効」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、132 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、134 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、134 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、133 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、131 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、140 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、140 ページの「MTA 設定」を参照してください。
11. 保存を完了します。

コマンド行：以下に示すように、コマンド行を使用して HTTP 属性の値を設定できます (詳細は、『Sun Java System Messaging Server Administration Reference』 (http://docs.sun.com/doc/819-0106) を参照)。

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 number
```

「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 number
```

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 value
```

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 number
```

接続の最大アイドル時間を設定する

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

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 number
```

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 number
```

最大プロセス数を設定する

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

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 dirpath
```

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。

代わりの MTA ホスト名を指定する

```
configutil -o service.http.smtphost -v hostname
```

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 portnum
```

第 6 章

シングルサインオン (SSO) の有効化

シングルサインオンとは、エンドユーザーからの 1 回の認証 (つまり、ユーザー ID とパスワードを使用したログイン) に対し、複数のアプリケーションへのアクセス権を付与する機能のことです。Sun Java System Access Manager (旧称 Identity Server) は、Sun Java System サーバーの SSO に使用される正規のゲートウェイです。SSO が設定されたほかのサーバーにアクセスするには、Access Manager にログインする必要があります。

たとえば、設定が適切なら、ユーザーは Sun Java System Access Manager のログイン画面でサインインでき、再度サインインすることなく別のウィンドウで Messenger Express にアクセスできます。同様に、Sun Java System Calendar Server の設定が適切なら、ユーザーは Sun Java System Access Manager のログイン画面でサインインでき、その後、再度サインインすることなく別のウィンドウで Calendar Server にアクセスできます。

Messaging Server には、SSO を配備する方法が 2 つあります。1 つは Sun Java System Access Manager を使用する方法、もう 1 つは通信サーバーの信頼できるサークル技術を使用する方法です。信頼できるサークルを使用することは、SSO の実装方法として長く採用されてきました。信頼できるサークルには Access Manager SSO では使用できない機能もありますが、今後の開発では Access Manager に照準が合わせられるため、この方法を使用することはお勧めしません。ただし、この章では、次の節で両方の方法について説明します。

Sun Java System サーバー用の Access Manager SSO

この節では、Access Manager を使用する SSO について説明します。この章には、以下の節があります。

SSO の制限事項と注意事項

- Messenger Express セッションは、Access Manager セッションが有効な場合に限り有効です。ユーザーが Access Manager からログアウトした場合、Web メールセッションは自動的に終了します (シングルサインオフ)。
- 同時に実行される SSO アプリケーションは、同一の DNS ドメインに存在する必要があります (cookie ドメインとも呼ばれる)。
- SSO アプリケーションには、Access Manager の確認 URL (ネーミングサービス) へのアクセス権が与えられている必要があります。
- ブラウザには cookie が必要です。

Messaging Server を設定して SSO をサポートする

Messaging Server SSO は、4 つの `configutil` パラメータによってサポートされます。4 つのパラメータのうち、Messaging Server で SSO を有効にするには、`local.webmail.sso.amnamingurl` の 1 つのみが必要とされます。SSO を有効にするには、このパラメータを、Access Manager がネーミングサービスを実行している URL に設定します。通常、この URL は http://*server*/amserver/namingservice となります。次に例を示します。

```
configutil -o local.webmail.sso.amnamingurl -v
http://sca-walnut:88/amserver/namingservice
```

注 Access Manager SSO は、古い SSO メカニズムを有効にする `local.webmail.sso.enable` を確認しません。`local.webmail.sso.enable` は `off` のままにしておくか、解除してください。これ以外の設定では、古い SSO メカニズムに必要とされる存在しない設定パラメータについての警告メッセージが記録されます。

表 6-3 で示されている SSO の設定パラメータは `configutil` コマンドを使用して変更できます。

表 6-1 Access Manager 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.amnamingurl	Access Manager がネーミングサービスを実行する URL。Access Manager を使用するシングルサインオンに必須の変数です。通常、この URL は `http://`*server*`/amserver/namingservice` です。 デフォルト : 設定なし。
local.webmail.sso.amcookiename	Access Manager の cookie 名。Access Manager が別の cookie 名を使用するように設定されている場合、その名前は Messaging Server で `local.webmail.sso.amcookiename` として設定する必要があります。これによって、Messaging Server でシングルサインオンを処理する場合の検索対象を指定できます。デフォルト値は `iPlanetDirectoryPro` です。Access Manager がデフォルト設定の場合は変更しないでください。 デフォルト : `iPlanetDirectoryPro`
local.webmail.sso.amloglevel	AMSDK ログレベル。Messaging Server で使用される SSO ライブラリには、Messaging Server とは別の独自のロギングメカニズムがあります。SSO ライブラリのメッセージは、*msg_svr_base*`/log` の下にある `http_sso` と呼ばれるファイルに記録されます。デフォルトでは、*info* 以上のメッセージのみが記録されますが、ログレベルを 1 ～ 5 の値 (1 = errors、2 = warnings、3 = info、4 = debug、5 = maxdebug) に設定することで、ログレベルを上げることは可能です。ライブラリでのメッセージの重要性の概念は Messaging Server と異なること、また、レベルを *debug* に設定すると無意味なデータが大量に記録されることに注意してください。さらに、`http_sso` ログファイルは、共通の Messaging Server ログコードで管理されないこと、クリーンアップされたりロールオーバーされたりすることがないことにも注意してください。デフォルトよりも高いログレベルに設定した場合は、システム管理者の責任でクリーンアップを行います。 デフォルト : 3

表 6-1　Access Manager 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.webmail.sso.singlesignoff	Messaging Server から Access Manager へのシングルサインオフ。Access Manager は中央の認証オーソリティであるため、シングルサインオフは常に Access Manager から Messaging Server の順で有効になります。このオプションを使用すると、サイトで Web メールの *logout* ボタンによって Access Manager からもユーザーを (カスタマイズ作業を保存して) ログアウトするかどうか設定できます。デフォルトでは、このオプションは有効になっています。このオプションを無効にした場合、デフォルトの Web メールクライアントからログアウトしたユーザーは自動的に再度ログインされます。ログアウトはルートドキュメントを参照し、ルートドキュメントは Access Manager cookie が存在していてそれが有効である限り受信箱画面を参照するためです。したがって、このオプションを無効に設定したサイトでは、Web メールのログアウト時に発生するアクションをカスタマイズする必要があります。 デフォルト : yes

SSO のトラブルシューティング

SSO に関して問題がある場合は、まず Web メールのログファイル *msg_svr_base*/log/http でエラーをチェックする必要があります。ログレベルを上げると作業がしやすくなります (configutil -o logfile.http.loglevel -v debug)。これで解決しない場合は、*msg_svr_base*/log/http_sso の amsdk メッセージをチェックしてから、amsdk ログレベルを上げてください (configutil -o local.webmail.sso.amloglevel -v 5)。サーバーを再起動しないとログレベルの変更が反映されないことに注意してください。

SSO の問題が解決しない場合は、Access Manager と Messaging Server の両方について、完全修飾ホスト名をログイン時に使用していることを確認してください。cookie は同一ドメインのサーバー間でのみ共有され、ブラウザはローカルサーバー名に使われるドメインを認識していないため、ブラウザで完全修飾ホスト名を使用しないと SSO は機能しません。

信頼できるサークル SSO (従来システム)

この節では、信頼できるサークル SSO について説明します。この方法による SSO を使用することはお勧めしません。今後の開発では Access Manager に照準が合わせられるためです。ただし、現時点では、信頼できるサークル SSO では使用可能で、Access Manager SSO では使用不可な機能もあります。この節には、以下の項があります。

信頼できるサークル SSO の概要と定義

SSO の配備に先立ち、次の用語を理解しておくことは重要です。

- SSO: シングルサインオン。1 つのアプリケーションにサインインするとほかのアプリケーションにもアクセスできること。ユーザー ID はすべてのアプリケーションにおいて同じです。
- **信頼できるアプリケーション** : SSO スキーム (SSO プレフィックス) を共有し、互いの cookie と確認を信頼し合うアプリケーション群。ピア SSO アプリケーションとも呼ばれます。
- **信頼できるサークル** : 信頼できるアプリケーションのグループ。同一の SSO プレフィックスを共有します。
- **SSO プレフィックス** : SSO を配備する担当者によって定義され、アプリケーションで認識されている文字列。同一の信頼できるサークル内に存在するほかのアプリケーションで生成された cookie の検出に使用されます。異なるプレフィックスを持つアプリケーションは同一サークル内に存在しないため、このようなアプリケーション間を移動する場合、ユーザーは再認証する必要があります。構成設定でプレフィックスには最後に「-」が付く場合があります。
- **アプリケーション ID** (appid): SSO を配備する担当者によって SSO サークル内の各アプリケーションに定義された一意の文字列。
- SSO cookie: ユーザーがあるアプリケーションで認証済みであることをブラウザが記憶するために使用されるトークン。cookie 名は、*SSO_prefix-application ID* という形式をとります。cookie の値は SSO キーであり、通常、アプリケーションによって生成されるセッション ID となります。

- cookie ドメイン : アプリケーションが cookie の送信を制限されているドメイン。これは DNS で意味するところのドメインです。
- 確認 URL: あるアプリケーションによって cookie の確認に使用され、別のアプリケーションによって検出される URL。

信頼できるサークル SSO アプリケーション

SSO を実装する前に、信頼できるサークルに含めるアプリケーションを考慮する必要があります。信頼できるサークルに含めることのできるアプリケーションは、Messenger Express (Messenger Express Multiplexor が付属された、または付属されていない)、Calendar Express、および旧バージョンのメッセージング用 iPlanet Delegated Administrator (Sun LDAP Schema 1 しかサポートしないので推奨しない) です。

表 6-2 に、SSO を介して互いにアクセス可能なアプリケーションを示します。ユーザーの視点でとらえると、行見出しで示されているアプリケーションのいずれかにログインし、ユーザー ID とパスワードを再入力せずに列見出しで示されているアプリケーションにアクセスできた場合、SSO は機能します。

表 6-2　SSO の相互運用性

アクセス先: / アクセス元 :	Calendar Express	Messenger Express	Messenger Express Multiplexor	Delegated Administrator
Calendar Express	SSO	SSO	SSO	SSO
Messenger Express	SSO	N/A	N/A	SSO
Messenger Express Multiplexor	SSO	N/A	N/A	SSO
Delegated Administrator	SSO	SSO	SSO	N/A

信頼できるサークル SSO の制限事項

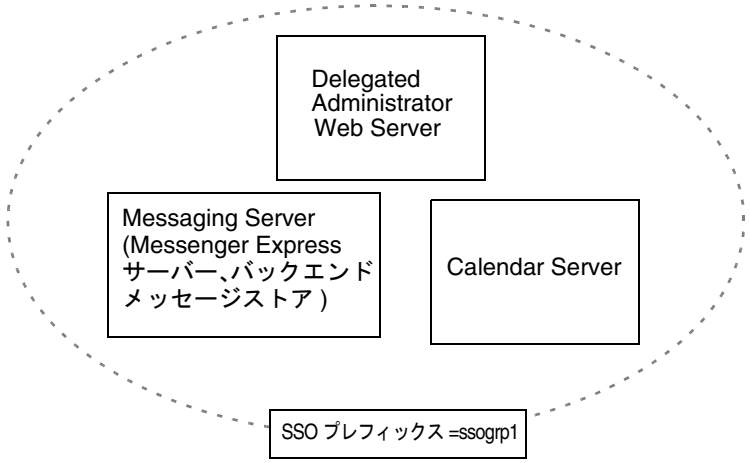
- 同時に実行される SSO アプリケーションは、同一ドメインに存在する必要があります。
- SSO アプリケーション群には、互いの SSO の確認 URL へのアクセス権が与えられている必要があります。
- ブラウザは cookie をサポートしている必要があります。

- セキュリティのためには、ブラウザが実行されている共有マシン上で SSO を使用しないでください。
- 別の ID に切り替えるには、ブラウザを再起動する必要があります。
- Messenger Express と Sun Java System Calendar Server の両方でシングルサインオフが有効になっている場合に Sun Java System Calendar Server からログアウトすると、本来ならば Messenger Express には再度ログインする必要があります。Messenger Express からログアウトすると、Sun Java System Calendar Server に再度ログインする必要があります。ただし、現在のところこのように機能しません。一方をログアウトしても、もう一方ではログインされたままの場合があります。

信頼できるサークル SSO 配備の例

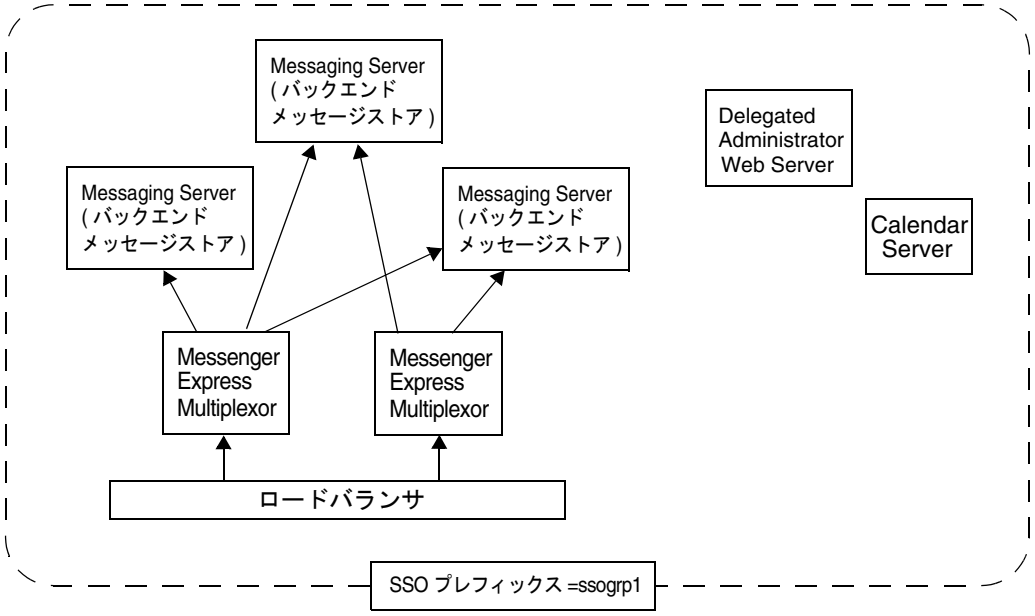
もっとも単純な SSO 配備の場合、Messenger Express と メッセージング用 iPlanet Delegated Administrator のみが使用されます。これより複雑な場合は、Calendar Express (同一マシン上または別のマシン上) が追加されます。追加するには、同一の信頼できるサークル内に存在できるように、同一の SSO プレフィックスを使用します。図 6-1 はこのことを示しています。

図 6-1　単純な SSO 配備



さらに複雑な配備では、Messenger Express Multiplexors およびロードバランサが追加されます。

図 6-2　複雑な SSO 配備



信頼できるサークル SSO の設定

この節では、Messenger Express、メッセージング用 iPlanet Delegated Administrator、および Calendar Manager 用の SSO の設定について説明します。

1. Messenger Express に SSO の設定をします。
   a. 適切な SSO `configutil` パラメータを設定します。

Messenger Express で Delegated Administrator とのシングルサインオンを有効にするには、次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 siroe.com と仮定)。パラメータの詳細については、表 6-3 を参照してください。設定を行うにはルートユーザーである必要があります。cd で *instance_root* に移動します。

```
configutil -o local.webmail.sso.enable -v 1
configutil -o local.webmail.sso.prefix -v ssogrp1
      ssogrp1 は iDA で使用されるデフォルトの SSO プレフィックスです。
      別のプレフィックスを選択することもできますが、デフォルトを使用すると
      iDA や iCS を設定するときにプレフィックスを入力せずに済みます。
configutil -o local.webmail.sso.id -v ims5
      ims5 は Messenger Express (ME) をほかのアプリケーションから識別するために
      付ける名前です。
configutil -o local.webmail.sso.cookiedomain -v ".siroe.com"
      上記のドメインは、ME/ ブラウザクライアントでサーバーとの接続に使用される
      ドメインと一致する必要があります。
      したがって、このサーバー上のホストしているドメインが xyz.com である場合、
      DNS にある実際のドメインを使用する必要があります。
      この値はピリオドで始まります。
configutil -o local.webmail.sso.singlesignoff -v 1
configutil -o local.sso.ApplicationID.verifyurl -v ¥
      "http://ApplicationHost:port/verifySSO?"
      ApplicationID は SSO アプリケーションに付ける名前です ( 例 :
      Delegated Administrator 用 ida、Calendar Server 用 ics50)。
      ApplicationHost:port は、アプリケーションのホストとポート番号です。
      Messaging Server 以外の各アプリケーションごとに、これらの行のいずれかが
      あります。次に例を示します。
      configutil -o local.sso.ida.verifyurl -v ¥
         "http://siroe.com:8080/verifySSO?"
```

b. 設定を変更後、Messenger Express http サーバーを再起動します。

```
cd instance_root
./stop-msg http
./start-msg http
```

2. Directory Server の SSO を設定します。

a. ディレクトリでプロキシユーザーアカウントを作成します。

プロキシユーザーアカウントを使って、Delegated Administrator はプロキシ認証を行うために Directory Server にバインドできます。次の LDIF コード (proxy.ldif) を使って、ldapadd を使うプロキシユーザーアカウントのエントリを作成できます。

ldapadd -h mysystem.siroe.com -D "cn=Directory Manager" -w *password* **-v -f proxy.ldif**

```
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
```

b. プロキシユーザーアカウント認証に適切な ACI を作成します。

ldapmodify ユーティリティを使用して、Delegated Administrator のインストール時に作成した各サフィックスの ACI を作成します。

osiroot - ユーザーデータを保存するために入力したサフィックス (デフォルトは o=isp)。osiroot は組織ツリーのルートです。

dcroot - ドメイン情報を保存するために入力したサフィックス (デフォルトは o=internet)。

osiroot - 設定情報を保存するために入力したサフィックス。これはユーザーデータを保存するために入力した値と同一になります。

以下に、先に作成したプロキシユーザーの osiroot の ACI エントリ (aci1.ldif) 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";)
```

ldapmodify -h siroe.com -D "cn=Directory Manager" -w *password* **-v -f aci1.ldif**

dcroot に同様の ACI エントリ (aci2.ldif) を作成します。

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

ldapmodify -h siroe.com -D "cn=Directory Manager" -w *password* **-v -f aci2.ldif**

3. Delegated Administrator を設定します。

   a. プロキシユーザー証明書およびコンテキストの cookie 名を Delegated Administrator resource.properties ファイルに追加します。

      Delegated Administrator の *iDA_server_root*/nda/classes/netscape/nda/servlet/resource.properties ファイルの次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*
      NDAAuth-singleSignOnId=*SSO_Prefix*-
      NDAAuth-applicationId=*DelAdminID*

      次に例を示します。

      ```
      LDAPDatabaseInterface-ldapauthdn=
         uid=proxy, ou=people, o=siroe.com, o=isp
      LDAPDatabaseInterface-ldapauthpw=proxypassword
      NDAAuth-singleSignOnId=ssogrp1-
      NDAAuth-applicationId=ida
      ```

   b. 対象となるサーバーの確認 URL を追加します。

      受け取る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ーに、確認 URL を指定します。

      次の例では、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 msg5 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-msg5=http://*webmail_hostname*:*port*/VerifySSO?
      verificationurl-ssogrp1-ida=http://*iDA_hostname*:*port*/VerifySSO?
      verificationurl-ssogrp1-ics50=http://*iCS_hostname*:*port*/VerifySSO?

4. Delegated Administrator のシングルサインオン cookie 情報を追加し、UTF8 パラメータのエンコーディングを有効にします。

   a. Delegated Administrator のコンテキスト識別子を定義します。

      Web_Server_Root/https-*instancename*/config/servlets.properties を編集し、servlet.*.context=ims50 というテキストを含んでいるすべての行のコメントを解除します。* は任意の文字列を示しています。

   b. Enterprise Server 設定のコンテキストの cookie 名を指定します。

      Enterprise Server ファイル *Web_Server_Root*/https-*instancename*/config/contexts.properties を編集し、ファイルの下部の #IDACONF-Start 行の前に次の行を追加します。

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

   c. ims5 コンテキストの UTF8 パラメータエンコーディングを有効にします。

      Enterprise Server 設定の ims5 コンテキストの UTF8 パラメータエンコーディングを有効にするには、次のエントリを Enterprise Server の *WebServer_Root*/https-*instancename*/config/contexts.properties ファイルに追加します。

      ```
      context.ims50.parameterEncoding=utf8
      ```

5. Messenger Express を再起動します。

   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 Messenger Express を再起動します。

   *WebServer_Root***/https-i***instance_name***/stop**
   *WebServer_Root***/https-***instancename***/start**

6. SSO グループに Calendar を配備している場合は、Calendar Server を設定します。

   ics.conf を編集し、以下を追加します。

   ```
   sso.appid = "ics50"
   sso.appprefix = "ssogrp1"
   sso.cookiedomain = ".red.iplanet.com"
   sso.enable = "1"
   sso.singlesignoff = "true"
   sso.userdomain = "mysystem.red.iplanet.com"
   sso.ims5.url="http://mysystem.red.iplanet.com:80/VerifySSO?"
   sso.ida.url=http://mysystem.red.iplanet.com:8080/VerifySSO?
   ```

7. Calendar Server を再起動します。

   start-cal

8. Messenger Express http サーバーを再起動します。

   msg_svr_base/sbin**/stop-msg http**
   msg_svr_base/sbin**/start-msg http**

Messenger Express 信頼 SSO 設定のパラメータ

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 6-3 に、パラメータを示します。configutil の詳細については、『Messaging Server Reference Manual』を参照してください。

表 6-3　信頼できるサークルのシングルサインオンパラメータ

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 値を設定します。*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 です。たとえば、Delegated Administrator のデフォルトの *appid* は、nda45 であり、実際の値は Delegated Administrator の resource.properties ファイルのエントリ NDAAuth-applicationID で指定されています。 信頼されている各 ピア 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ります。確認 URL の標準形は次のようになります。 http://*nda-host:port*/VerifySSO? 複数の Messenger Express Multiplexors と (Messenger Express を実行している) Message Store サーバーの前で、または Calendar フロントエンドの前でロードバランサを使用する場合、各物理的システムに、異なる appid を verifyurl にある実際のホスト名とともに割り当ててください。これによって、cookie の確認に正しいシステムが使用されます
local.webmail.sso.cookiedomain	このパラメータの文字列値は、Messenger Express HTTP サーバー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ます。デフォルト値は null (空白)。 このドメインは、Messenger Express ブラウザでサーバーへのアクセスに使用される DNS ドメインと一致する必要があります。ホストしているドメインの名前ではありません。
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、ログイン成功時にクライアントに 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します。 ゼロ以外の値に設定した場合、サーバーはすべての SSO 機能を実行します。 ゼロに設定した場合、サーバーはどの SSO 機能も実行しません。 デフォルト値はゼロ。

表 6-3　信頼できるサークル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.webmail.sso.id	このパラメータの文字列値は、Messenger Express HTTP サーバー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 として使用されます。デフォルト値は null (空白)。 これは任意の文字列です。この値は Delegated Administrator の resource.properties ファイルに指定した値と一致する必要があります。resource.properties での対応するエントリは次のようになります。 Verifycationurl-XXX-YYY=http://webmailhost:webmailport/VerifySSO? XXX は、上記の local.webmail.sso.prefix 値セット、YYY は、ここで設定される local.webmail.sso.id の値です。
local.webmail.sso.prefix	このパラメータの文字列値は、Messenger Express HTTP サーバーによって設定された SSO cookie をフォーマットするときのプレフィックスとして使用されます。このプレフィックスの付いた SSO cookie だけがサーバーによって認識され、ほかの SSO cookie はすべて無視されます。 このパラメータの値が null (空白) の場合は、事実上、サーバー上のすべての SSO 機能が無効になります。 デフォルト値は null (空白)。 この文字列は、メッセージング用 iPlanet Delegated Administrator の resource.properties ファイルで使用されている文字列から末尾に付いている「-」を除いたものと一致する必要があります。次に例を示します。 NDAAuth-singleSignOnID=ssogrp1- この場合、ここで設定する値は ssogrp1 です。
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 の値に一致するプレフィックス値を持つクライアント上の SSO cookie がすべて消去されます。 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 を消去します。 デフォルト値はゼロ。

第 7 章

マルチプレクササービスの設定および管理

この章では、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 インタフェース用の 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、以下の項目があります。

マルチプレクササービス

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ー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 Messaging Server で別途管理され、Messenger Express Multiplexor は 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ーでは、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、ユーザーメールボックスとユーザー接続を複数のサーバーに振り分けると、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、小さなサーバーマシンを数台使う方が費用効率が高い場合があります。

メールサーバー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。ユーザー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ー間でのユーザー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。

- **ユーザー管理の簡易化**

  すべてのユーザーが 1 台 のサーバー (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、すべてのユーザー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ーを実行し、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ー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ーの Messaging Server を検出します。このため、システム管理者は、ユーザーに意識させることなく、ユーザーを簡単に新しいサーバーに移動することができます。管理者はユーザー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ーのエントリを更新することができます。ユーザーのメールアドレス、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ーを異なるマシンに割り当てることができます。たとえば、重要なユーザー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、ユーザーが低速で接続 (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、小型のサーバー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。
- **最小限のユーザーダウンタイム** : マルチプレクサを使用すると、大規模なユーザー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、ユーザーダウンタイムを抑えることができます。あるサーバーが故障しても、影響を受けるのはそのサーバーのユーザーだけです。
- **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ー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。

Messaging Multiplexor について

Sun Java System Messaging Multiplexor (MMP) は、複数のバックエンドメッセージングサーバーの単一接続ポイントとして機能する特別なメッセージングサーバーです。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 のユーザー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。すべてのユーザーは、単一の Multiplexor サーバーに接続します。Multiplexor サーバーは、各接続を適切な Messaging Server にリダイレクトします。

多数のユーザーに電子メールサービスを提供する場合は、ユーザーには複数の 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 Messaging Server の一部として提供されます。MMP は Messaging Server やほかの Sun Java System サーバー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。

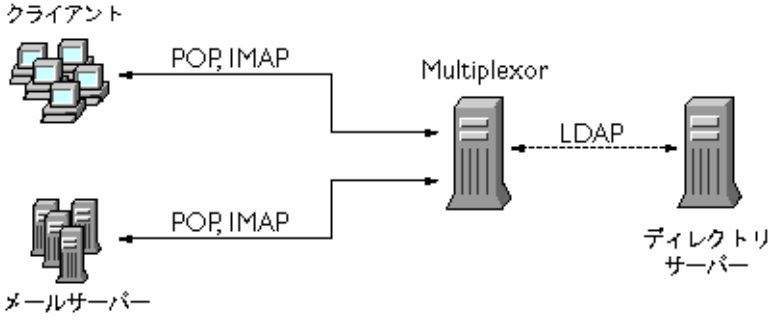
- メールクライアントとの非暗号化通信および暗号化 (SSL) 通信。
- 証明書に基づくクライアント認証 (162 ページの「証明書に基づくクライアント認証」を参照)。
- ユーザーの事前認証 (163 ページの「ユーザーの事前認証」を参照)。
- さまざまな IP アドレスを待機し、ユーザー ID にドメイン名を自動的に付与する仮想ドメイン (163 ページの「MMP 仮想ドメイン」を参照)。
- 複数のサーバーに対する MMP のインストール (『Sun Java Enterprise System インストールガイド』を参照)。

- 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、706 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

MMP は、メールユーザーを複数のサーバーマシンに分散させるマルチスレッドサーバーです。MMP は、ユーザーメールボックスがあるサーバーマシン宛の着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 はユーザーの正しいサーバーを判断し、そのサーバーに接続し、クライアントとサーバー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、ユーザー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、ユーザーの効率を向上させ、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 7-1 に、MMP をインストールした場合のサーバーとクライアントの関係を示します。

図 7-1　MMP をインストールした場合のクライアントとサーバー



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ーをインストールした場合と同様、各ユーザーは特定の 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、ユーザー接続は次の手順で確立されます。

1. ユーザーのクライアントが MMP に接続し、予備的な認証情報 (ユーザー名) が受け入れられます。
2. MMP は Directory Server に照会して、そのユーザー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

Messaging Multiplexor は、Messaging Server とメールクライアント間の暗号化 (SSL) 通信および非暗号化通信をサポートしています。Messaging Server は、新しい証明書データベース形式 (cert8.db) をサポートしています。

SSL を有効にすると、MMP 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待機するように 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、`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 サーバーがセキュアサーバー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 サーバーポートを指定する必要があります。詳細は、169 ページの「SSL を使用する MMP を設定する」を参照してください。

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 サーバー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 パラメータの一覧は、『Sun Java System Messaging Server Administration Reference』(`http://docs.sun.com/doc/819-0106`) を参照してください。

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 の正しいユーザー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。161 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、一意のユーザー 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 または POP 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ー 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ー 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。詳細は、161 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 つインストールします。詳細は、683 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。

ユーザーの事前認証

MMP には、着信ユーザー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ーを事前認証するオプションがあります。

注	ユーザーの事前認証を有効にすると、サーバー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*) user *name* pre-authenticated - client IP *address,* server IP *address*

date は yyyymmdd 形式、*time* はサーバーで設定された hhmmss 形式の時刻であり、*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ーの 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、サーバー 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。

ユーザーは、省略形のユーザー ID または完全指定のユーザー 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。省略形のユーザー 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、複数のホストしているドメインをサポートするサイトでは、サーバー IP アドレスと MMP 仮想ドメインにそれぞれのホストしているドメインを関連付けるだけで、省略形のユーザー ID を使用できるようになります。

特定のホストしているドメインのユーザー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ーでも LDAP のユーザー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。

Sun LDAP Schema 2 が有効になると (『Sun Java Enterprise System インストールガイド』および『Sun Java System Communications Services Schema Reference Manual』を参照)、特定のドメインのユーザーサブツリーは、そのドメインの組織ノードの下のサブツリーにあるすべてのユーザーになります。

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、以下の構文を使用します。

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
RestrictPlainPasswords
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、『Messaging Server Reference Manual』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、706 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。690 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 を設定する

Messaging Server の初期実行時設定中に、MMP をマシンに設定するかどうかを選択できます。Messaging Server と同一のマシンに設定することも、別のマシンに設定することもできます。

注 MMP は DNS の結果をキャッシュしません。Messaging Server の本稼働配備では、高性能キャッシュを使用する DNS サーバーがローカルネットワークに必要です。

以下の節では、MMP の設定方法について説明します。

MMP の詳細については、次のマニュアルで参照できます。

- 『Sun Java System Messaging Server Administration Reference』:「MMP Syntax and Structure」(http://docs.sun.com/doc/819-0106)

MMP を設定する前に

MMP を設定する前に、次の操作を実行します。

1. MMP を設定するマシンを選択します。MMP 専用のマシンを使用することをお勧めします。

注	POP または IMAP サーバーを実行するマシンでは、MMP を有効にしないことをお勧めします。 Messaging Server と同じマシンに MMP をインストールする場合は、POP サーバーおよび IMAP サーバーを標準以外のポートに設定する必要があります。標準以外のポートを使用すれば、MMP サーバーと Messaging Server のポートが互いに競合することはありません。

2. MMP を設定するマシンに、MMP で必要な UNIX システムユーザーを作成します。この新規ユーザーは、UNIX システムグループに属している必要があります。42 ページの「UNIX システムのユーザーとグループを作成するには」を参照してください。
3. Messaging Server で使用する Directory Server とホストマシンの設定が完了していない場合は、それらを設定します。43 ページの「Messaging Server 設定用に Directory Server を準備するには」を参照してください。
4. バックエンドサーバーより前に MMP がアップグレードされた場合、ユーザーは `ImapProxyAService.cfg` の `Capability` オプションを、古いバックエンドから発行された `capability` コマンドへの応答と一致するように設定する必要があります。この設定は次のようになります。

   ```
   IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ NAMESPACE UIDPLUS CHILDREN
   LANGUAGE XSENDER X-NETSCAPE XSERVERINFO
   ```

   改行は編集上明確にする目的で使用されること、および設定値は 1 行に配置する必要があることに注意してください。

Multiplexor の設定

MMP を設定するには、Messaging Server の設定プログラムを使用する必要があります。このプログラムには、Messaging Multiplexor を有効にするかどうかを選択するオプションがあります。設定プログラムの詳細については、53 ページの「Messaging Server の初期実行時設定を作成するには」を参照してください。

MMP を設定するには、次の手順に従います。

1. MMP をインストールおよび設定するマシンに Sun Java System Messaging Server をインストールします。

2. Messaging Server の初期実行時設定を作成して MMP を設定します。53 ページの「Messaging Server の初期実行時設定を作成するには」を参照してください。

   例外として、Messaging Server をインストールする場合は、Messaging Multiplexor オプションのみをチェックするようにしてください。

Multiplexor のファイル

Messaging Multiplexor のファイルは、*msg_svr_base*/config 設定ファイルディレクトリに格納されています。表 7-1 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。MMP 設定パラメータの詳細については、『Sun Java System Messaging Server Administration Reference』を参照してください。

表 7-1　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。このファイルは、start-msg mmp で MMP を最初に起動したときにのみ作成されます。
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。このファイルは、start-msg mmp で MMP を最初に起動したときに作成されます。
AService.cfg	起動するサービス、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	起動するサービス、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。このファイルは、start-msg mmp で MMP を最初に起動したときに作成されます。
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 を有効にする場合は必須です。POP before SMTP を有効にしない場合でも、SSL ハードウェアのサポートを最大にするのに役立ちます。POP before SMTP の詳細については、706 ページの「POP before SMTP を有効にする」を参照してください。
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。このファイルは、start-msg mmp で MMP を最初に起動したときに作成されます。

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 を設定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。

Multiplexor の起動

Messaging Multiplexor のインスタンスを起動、停止、更新するには、次の表 7-2 に示すコマンドのうちの 1 つを使用します。これらのコマンドは *msg_svr_base*/sbin ディレクトリにあります。

表 7-2 MMP コマンド

オプション	説明
start-msg mmp	MMP を起動します。これは、別のインスタンスが起動されている場合でも可能です。
stop-msg mmp	最後に起動した MMP を停止します。
refresh mmp	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更新するようにします。

既存の MMP の変更

既存の MMP インスタンスを変更するには、必要に応じて、ImapProxyAService.cfg または PopProxyAService.cfg、あるいはその両方の設定ファイルを編集します。これらの設定ファイルは、*msg_svr_base*/config サブディレクトリにあります。

SSL を使用する MMP を設定する

SSL を使用するように MMP を構成するには、次の手順に従います。

注 ここでは、メッセージストアまたは MTA を持たないマシンに MMP をインストールすることを前提としています。

1. 管理サーバーがインストールされている場合は、管理コンソールを使用して SSL サーバー証明書をインストールします。それ以外の場合は、NSS ツールを使用します。690 ページの「ネットワークセキュリティサービスツール」を参照してください。

   http://docs.sun.com/db/doc/816-5572-10 を参照してください。

2. 管理サーバーがインストールされている場合は、操作を簡略化するために、コマンド行で次のシンボリックリンクを作成します。

   ```
   cd msg_svr_base/config
   ln -s /var/mps/serverroot/alias/admin-serv-instance-cert7.db cert7.db
   ln -s /var/mps/serverroot/alias/admin-serv-instance-key3.db key3.db
   ```

   さらに、これらのファイルが、MMP を実行するシステム ID に属していることを確認します。Messaging Server は、新しい証明書データベース形式 (cert8.db) をサポートしています。

3. `sslpassword.conf` ファイルは Messaging Server の初期実行時設定で設定されているので、新しく設定する必要はありません。53 ページの「Messaging Server の初期実行時設定を作成するには」を参照してください。

注 手順 1 ～ 8 を実行する代わりに、次のファイルをコピーする方法もあります。既存の Messaging Server または Directory Server の `cert7.db`、`key3.db`、`secmod.db`、および `sslpassword.conf` の各ファイル。コピー元のサーバーには、同じドメインに対する適切なサーバー証明書とキーがあらかじめインストールされている必要があります。

4. `ImapProxyAService.cfg` ファイルを編集して、関連のある SSL 設定のコメント記号を削除します。

5. SSL と POP を使用する場合は、`PopProxyAService.cfg` ファイルを編集して、関連のある SSL 設定のコメント記号を削除します。

   さらに、`AService.cfg` ファイルを編集して、`ServiceList` 設定の「`110`」のあとに「`|995`」を追加します。

6. ImapProxyAService.cfg ファイルと PopProxyAService.cfg ファイルに、BindDN オプションと BindPass オプションが設定されていることを確認します。

   さらに、DefaultDomain オプションには、デフォルトドメイン (資格のないユーザー名で使用するドメイン) を設定する必要があります。

サーバー側のみで SSL を使用する場合は、これで作業は完了です。*msg_svr_base*/sbin ディレクトリで次のコマンドを使用して MMP を起動します。

```
start-msg mmp
```

クライアント証明書を使用したログインを行う場合は、次の手順に従います。

1. クライアント証明書とそれに署名した CA 証明書のコピーを入手します。
2. 以前と同じように、MMP をインストールしたマシン上で Sun ONE Console を起動します。ただし、この時に信頼できる認証局として CA 証明書をインポートします。
3. Messaging Server のインストール時に作成したストア管理者 (Store Administrator) を使用します。

   詳細は、580 ページの「ストアへの管理者によるアクセスを指定する」を参照してください。
4. MMP の certmap.conf ファイルを作成します。次に例を示します。

   ```
   certmap default  default
   default:DNComps
   default:FilterComps   e=mail
   ```

   これは、LDAP サーバーの mail 属性を調べて、証明書 DN の e フィールドと一致するものを検索することを意味します。
5. ImapProxyAService.cfg ファイルを編集し、以下の設定を行います。

   a. CertMapFile には、certmap.conf を設定します。

   b. StoreAdmin と StorePass には、手順 3 の値を設定します。

   c. UserGroupDN には、ユーザー / グループツリーのルートを設定します。
6. POP3 によるクライアント証明書を必要とする場合は、PopProxyAService.cfg ファイルに対して、手順 5 の操作を繰り返します。
7. MMP がまだ実行されていない場合は、*msg_svr_base*/sbin ディレクトリで次のコマンドを使用して MMP を起動します。

   ```
   start-msg mmp
   ```

8. クライアント証明書をクライアントにインポートします。Netscape™ Communicator では、鍵 (セキュリティ) のアイコンをクリックし、「証明書」の「本人」を選択し、次に、「証明書のインポート ...」を選択して画面の指示に従います。

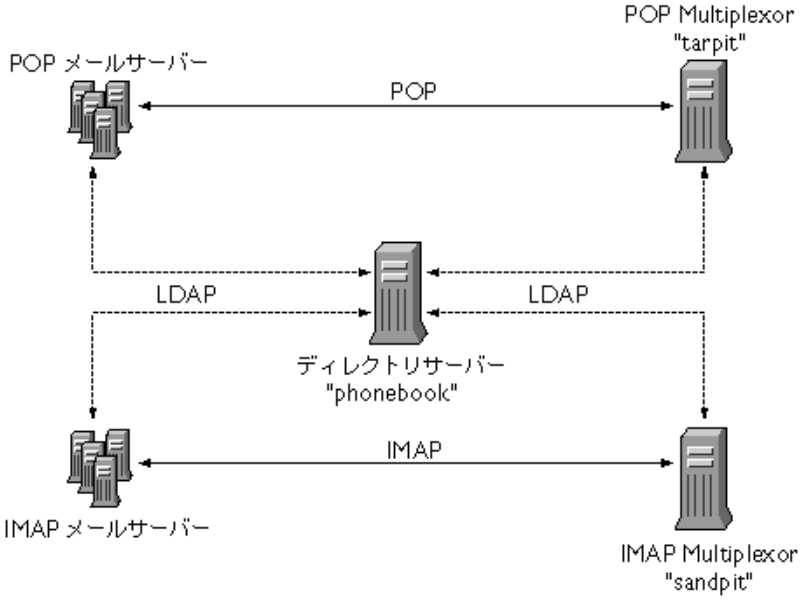
注	すべてのログインでクライアント証明書を使用する場合は、すべてのユーザーがこの手順を実行する必要があります。

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例を想定します。POP および IMAP のユーザー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、それぞれのサーバーは 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。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)。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ています。

このトポロジを、図 7-2 に示します。

図 7-2　複数の MMP による複数の Messaging Server のサポート



IMAP の構成例

図 7-2 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 はユーザーメールボックスの情報を扱うホスト `phonebook` の LDAP サーバーと通信し、適切な IMAP サーバー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.siroe.com/o=internet
default:LogDir              /opt/SUNWmsgsr/config/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
BINARY LANGUAGE XSENDER X-NETSCAPE XSERVERINFO"
default:SearchFormat        (uid=%s)
default:SSLEnable           yes
default:SSLPorts            993
default:SSLSecmodFile       /opt/SUNWmsgsr/config/secmod.db
default:SSLCertFile         /opt/SUNWmsgsr/config/cert7.db
default:SSLKeyFile          /opt/SUNWmsgsr/config/key3.db
default:SSLKeyPasswdFile    ""
default:SSLCipherSpecs      all
default:SSLCertNicknames    Siroe.com Server-Cert
default:SSLCacheDir         /opt/SUNWmsgsr/config
default:SSLBacksidePort     993
default:VirtualDomainFile   /opt/SUNWmsgsr/config/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          /opt/SUNWmsgsr/config/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 7-2 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 はユーザーメールボックス情報を扱うホスト phonebook の LDAP サーバーと通信し、適切な POP サーバー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.siroe.com/o=internet
default:LogDir               /opt/SUNWmsgsr/config/log
default:LogLevel             5
default:BindDN               "cn=Directory Manager"
default:BindPass             password
default:BacksidePort         110
default:Timeout              1800
default:SearchFormat         (uid=%s)
default:SSLEnable            no
default:VirtualDomainFile    /opt/SUNWmsgsr/config/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           /opt/SUNWmsgsr/config/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
```

MMP のタスク

この節では、その他の MMP 設定タスクについて説明します。次のようなタスクがあります。

- 175 ページの「MMP を使ったメールアクセスを設定するには」
- 175 ページの「MMP LDAP サーバーフェイルオーバーを設定するには」

MMP を使ったメールアクセスを設定するには

MMP は自動的には設定されないので、明示的に設定する必要があります。また、MMP では `PORT_ACCESS` マッピングテーブルは使用されません。MMP を使用している場合、特定の IP アドレスからの SMTP 接続を拒否するには、`TCPAccess` オプションを使用する必要があります。このオプションの構文は `mailDomainAllowedServiceAccess` と同じです (『Sun Java System Communications Services Schema Reference Manual』(`http://docs.sun.com/doc/819-0113`) を参照)。また、697 ページの「フィルタの構文」でも説明されています。

MMP LDAP サーバーフェイルオーバーを設定するには

複数の LDAP サーバーを MMP として指定することができます。これによって、1 つのサーバーに障害が発生しても別のサーバーが処理を引き継げるようになります。`PopProxyAservice.cfg` または `IMAPProxyAservice.cfg` を次のように修正します。

`default:LdapUrl "ldap://`*ldap01.yourdomain ldap02.yourdomain*`/o=INTERNET"`

Messenger Express Multiplexor について

Sun Java System 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ーです。Messenger Express は、Sun Java System 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。すべてのユーザーがこのメッセージングプロキシサーバーに接続し、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、メールユーザーには複数の Messaging Server が単一のホストであるかのように表示されます。

Messaging Multiplexor (MMP) は POP および IMAP サーバーに接続しますが、Messenger Express Multiplexor は HTTP サーバーに接続します。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については、第 19 章「セキュリティとアクセス制御を設定する」の「セキュリティとアクセス制御」を参照してください。

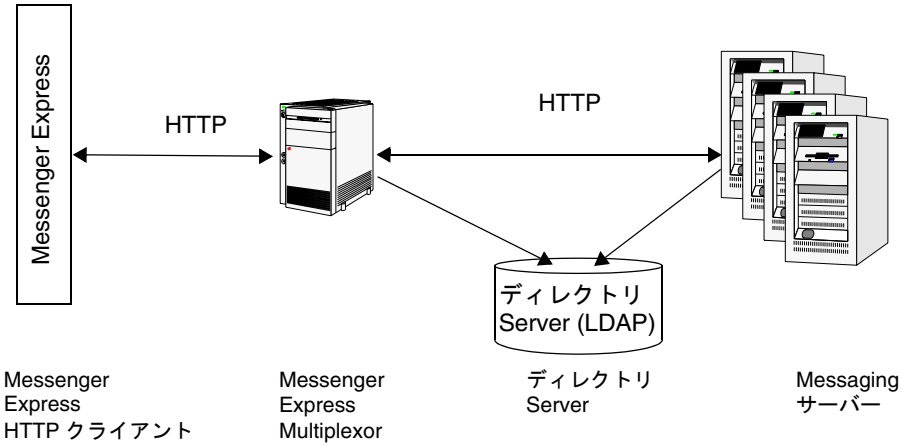
- ホストしているドメイン

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、ロギングと設定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ーで構成されており、ユーザーが 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、複数のサーバー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 Messenger Express にログオンすると Multiplexor に接続します。Multiplexor はユーザーの正しいサーバーを判断し、そのサーバーに接続し、クライアントとサーバー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、ユーザー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、ユーザーの効率を向上させ、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。177 ページの図 7-3 に、Messaging Server での Messenger Express Multiplexor の位置を示します。

図 7-3　iPlanet Messenger Express Multiplexor の概要



Messenger Express Multiplexor は、Messenger Express クライアントと 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ーをインストールした場合と同様、各ユーザーは特定の 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、ユーザー接続は次の手順で確立されます。

1. ユーザーのクライアントが Messenger Express Multiplexor に接続し、予備的な認証情報が受け入れられます。
2. Messenger Express Multiplexor は Directory Server に照会して、そのユーザー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。

Messaging Express Multiplexor を設定する

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。

プロキシマシンに Messaging Server をインストールするには

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。インストール手順については、『Sun Java Enterprise System インストールガイド』を参照してください。

Messaging Server は、バックエンドメッセージングサーバーを指す 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ーは、Messenger Express Multiplexor を介して、Messeging Server でユーザーを認証するために使用します。

Messenger Express 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メッセージングサーバーの情報を集めます。

   バックエンドメッセージングサーバーのディレクトリで `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ー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 *msg_svr_base*`/sbin/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ーの値と同じであることを確認します。

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。

- 179 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メッセージングサーバー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ーが使用する、ユーザーおよびグループの 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ーが使用するものと同じ値、または同じデータを含むレプリケートされた LDAP サーバーを指定します。

- `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ー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ー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ー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 を有効にするには

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 *msg_svr_base*/sbin/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。

非ローカルユーザー (ログインしたサーバーにメールホストがないユーザー) がログインした場合、local.service.http.proxy の値が 0 であれば、このユーザーは自分のホストに転送されます。ユーザーは、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ーからは Messaging Server 全体が 1 台のホストであるかのように見えます。

ローカルユーザー (ログインしたサーバーがメールホストであるユーザー) の場合は、local.service.http.proxy のパラメータ値とは無関係にサーバー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ーとローカルユーザーを共存させることもできます。

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には

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*

   次に例を示します。

   ```
   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ー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ー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、ユーザーからは 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

ユーザー 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表示するには、*msg_svr_base*/`log` 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、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には

Messenger Express Multiplexor の SSL (Secure Sockets Layer) の設定と管理については、687 ページの「SSL を有効化し暗号化方式を選択するには」を参照してください。

複数のプロキシサーバーを設定するには

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ー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 を管理するには

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 Messaging Server を使う場合は、Messenger Express のスタティックファイルを更新してサーバー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スタティックファイルは、ユーザーのメールホストではなく Messenger Express Multiplexor から直接提供されます。Multiplexor が *msg_svr_base*/config/html ディレクトリにあるこれらの設定ファイルを見つけます。

サーバー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 *msg_svr_base*/config/html ディレクトリの内容 (Messenger Express インタフェースを構成するスタティック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ーで Messaging Server 6 2003Q4 を使用し、Messenger Express Multiplexor には Messaging Server 6 2005Q1 をインストールしている場合は、Messenger Express Multiplexor の *msg_svr_base*/config/html ディレクトリの内容を、Messaging Server 6 2003Q4 を使用するバックエンドサーバー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、Messaging Server 6 2003Q4 から Messaging Server 6 2005Q1 にアップグレードするときに、Messenger Express Multiplexor サーバーの *msg_svr_base*/config/html ディレクトリにあるスタティックファイルも更新することができます。

Messenger Express Multiplexor を使用するバックエンドメッセージングサーバーのポートを設定するには

Messenger Express Multiplexor を使用するバックエンド HTTP メッセージングサーバーのポートを設定する場合は、Multiplexor マシンで次の configutil コマンドを使用します。

```
local.service.http.proxy.port.hostname
```

hostname は、バックエンド HTTP メッセージングサーバーのホストです。

たとえば、バックエンドメッセージングサーバーのホスト名が sesta.com で、ポート番号が 8888 の場合、コマンドは次の形式になります。

```
configutil -o local.service.http.proxy.port.store.sesta.com -v 8888
```

独自のポートを持っているバックエンドメッセージストアを除き、local.service.proxy.port はすべてのバックエンドメッセージストアに適用されます (local.service.proxy.admin と同じ)。

シングルサインオンを設定するには

シングルサインオンは、Messenger Express Multiplexor マシンで設定します。Messaging (HTTP) Server と同様に、次の追加設定が必要です。

```
configutil -o local.service.http.proxy.admin -v store_administrator
```

store_administrator は、バックエンドメッセージングサーバーのインストール中に指定したバックエンドストアの管理者です。

```
configutil -o local.service.http.proxy.adminpass -v store_admin_password
```

store_admin_password は、バックエンドメッセージングサーバーのインストール中に指定したバックエンドストア管理者のパスワードです。

異なるストア管理者とパスワードを使用する複数のバックエンドメッセージングサーバーを使用している場合は、次のように Messenger Express Multiplexor の各設定変数に完全修飾ホスト名を追加して、それらを個別に設定できます。

```
configutil -o local.service.http.proxy.admin.hostname -v store_administrator
```

```
configutil -o local.service.http.proxy.adminpass.hostname -v
store_admin_password
```

hostname は、バックエンド HTTP メッセージングサーバーのホストです。*store_administrator* および *store_admin_password* は、バックエンド HTTP メッセージングサーバーのインストール中に指定したバックエンドストア管理者およびパスワードです。

ユーザーをバックエンドサーバーにログインさせるために、Messenger Express Multiplexor で `proxyauth` ログインコマンドを使用します。`proxyauth` を有効にするには、バックエンドメッセージストアで次の `configutil` パラメータを使用します。

```
configutil -o service.http.allowadminproxy -v 1
```

注 シングルサインオンが Messenger Express Multiplexor を通じて有効化されている場合は、バックエンド HTTP メッセージングサーバーで設定する必要はありません。

Identity Server を使用して Messenger Express Multiplexor SSO を設定するには、次の `configutil` パラメータを有効にします。

```
./configutil -o local.webmail.sso.amcookiename -v iPlanetDirectoryPro
```

```
./configutil -o local.webmail.sso.amnamingurl -v ¥
        http://identity host:identity port/amserver/namingservice
```

第 8 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 (Message Transfer Agent) は Messaging Server (187 ページの図 8-1) の構成要素です。もっとも基本的なレベルにおいては、MTA はメッセージルーターです。MTA はほかのサーバーからメッセージを受信してアドレスを読み取り、最終的な宛先 (通常はユーザーのメールボックス) に向けて次のサーバーにルーティングします。

長年にわたって、MTA には多くの機能が追加され、サイズ、能力、および複雑さも増してきました。これらの MTA 機能は重複していますが、通常は次のように分類できます。

- **ルーティング**: メッセージを受け取り、エイリアスである場合などは必要に応じて展開または変換して、次のサーバー、チャネル、プログラム、ファイルなどにルーティングします。ルーティング機能が拡張され、管理者がメッセージのルーティングについての内部的および外部的な方法を指定できるようになりました。たとえば、SMTP 認証、各種 SMTP コマンドおよびプロトコルの使用、TCP/IP または DNS 検索のサポート、ジョブの送信、プロセス制御やメッセージキューイングなどを指定できます。

- **アドレス書き換え**：通常、エンベロープアドレスはルーティングプロセスの一環として書き換えられますが、エンベロープまたはヘッダーアドレスを、より希望に沿った適切な形式に書き換えることができます。
- **フィルタ**：MTA は、アドレス、ドメイン、感染のおそれのあるウィルスやスパムの内容、サイズ、IP アドレス、ヘッダーの内容などに基づいて、メッセージをフィルタ処理できます。フィルタ処理されたメッセージは、破棄、拒否、変更、ファイルやプログラムに送信、またはユーザーのメールボックスに向けて送信する途中に次のサーバーに送信することができます。
- **コンテンツの変更**：メッセージのヘッダーまたはコンテンツを変更することができます。たとえば、特定のクライアントから読み取り可能なメッセージや、特定の文字セットをサポートするメッセージを作成したり、スパムやウィルスのチェックを行います。
- **監査**：だれが、いつ、どこから、どのようなメッセージを送信したかを追跡します。

これらの機能は、188 ページの図 8-2 に示される多数のサブコンポーネントおよびプロセスでサポートされています。この章では、これらのサブコンポーネントとプロセスについて説明します。また、システム管理者は多数のツールを使用して、これらの機能を有効化および設定することができます。このようなツールには、MTA オプション、`configutil` パラメータ、マッピングテーブル、キーワード、チャネル、書き換えルールなどがあります。これらのツールについては、後に MTA の章で説明します。

- 第 9 章「MTA のアドレス変換とルーティング」
- 第 10 章「MTA サービスと設定について」
- 第 11 章「書き換えルールの設定」
- 第 12 章「チャネル定義を設定する」
- 第 13 章「定義済みチャネルを使用する」
- 第 14 章「スパムとウィルスのフィルタ処理プログラムを Messaging Server に統合する」
- 第 15 章「LMTP 配信」
- 第 16 章「不在メッセージの自動返信」
- 第 17 章「メールのフィルタリングとアクセス制御」
- 第 19 章「セキュリティとアクセス制御を設定する」
- 第 21 章「ログの管理」
- 第 22 章「MTA のトラブルシューティング」
- 第 23 章「Messaging Server を監視する」

図 8-1　Messaging Server の簡易コンポーネント表示 (Messenger Express は表示されていない)

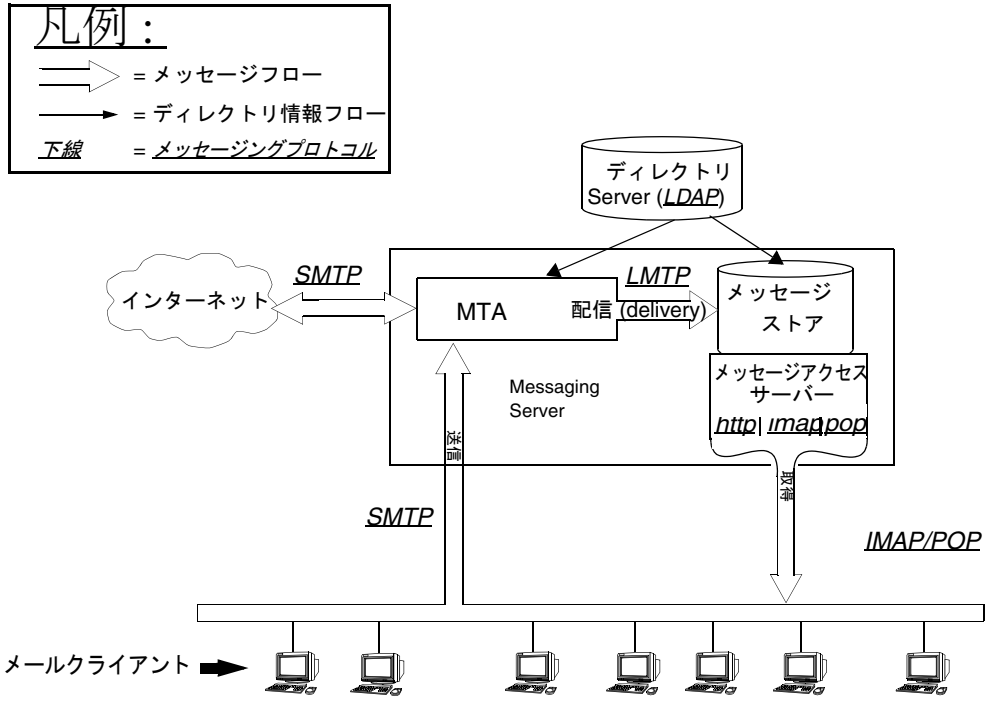
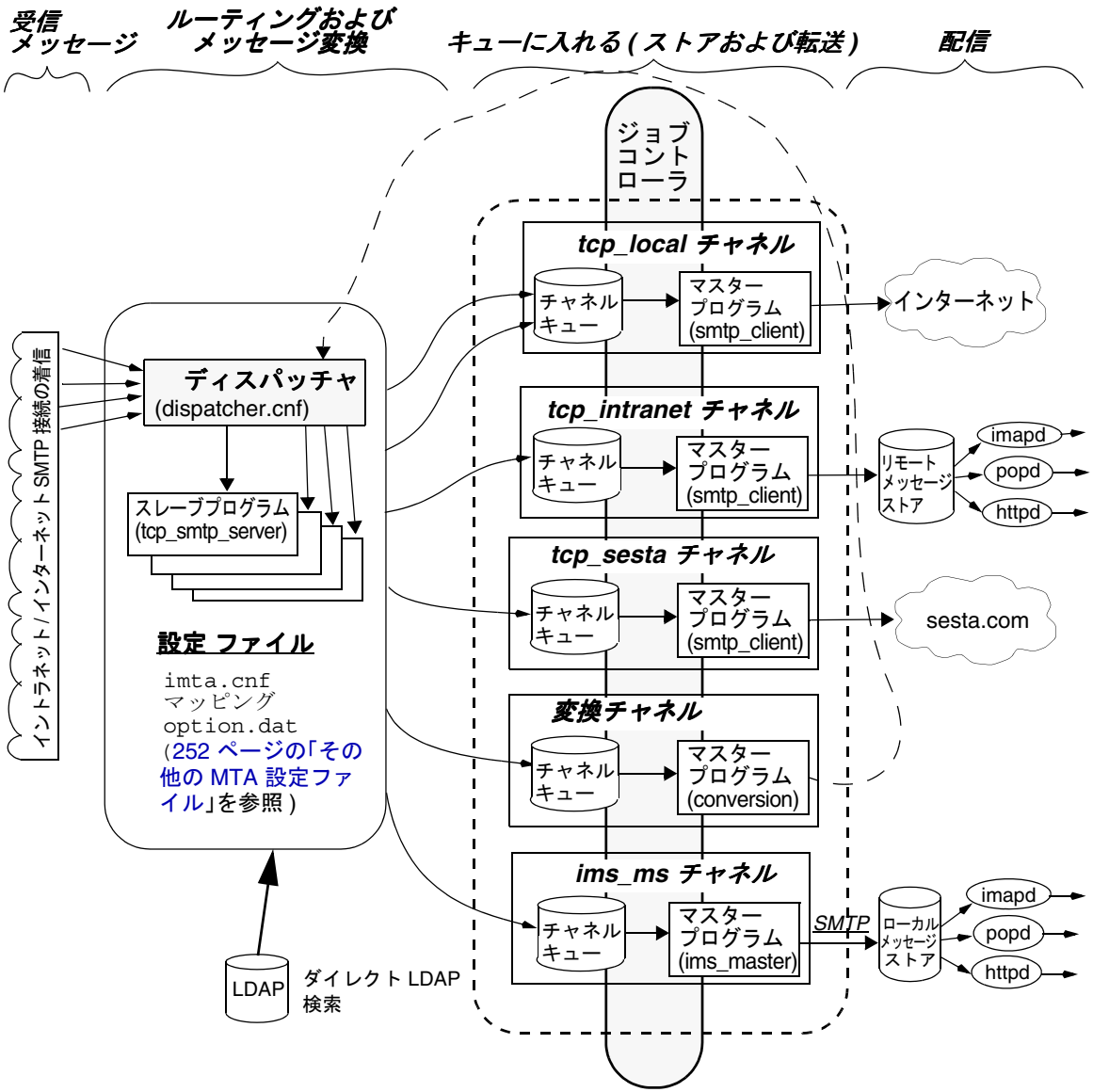


図 8-2　MTA のアーキテクチャ



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 8-2)。MTA は非常に複雑なコンポーネントであること、図 8-2 はシステムを通じて配信されるメッセージの簡略図であることに注意してください。実際、この図は、システムを通じて配信されるすべてのメッセージを厳密に示しているわけではありません。ただし、概念を説明するという目的は十分に果たしています。

ディスパッチャと SMTP サーバー (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 (マルチスレッド接続ディスパッチエージェント) 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。ディスパッチャは、各サービスのマルチスレッドプロセスのプールを管理します。さらにセッションが要求されると、ディスパッチャは SMTP サーバー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。ディスパッチャのプロセスプール内のプロセスは、複数の接続を同時に処理することもあります。ディスパッチャとスレーブプログラムにより、着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 17 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。着信したアドレスの `From:` や `To:` を必要な形式に書き換える。
- チャネルへのキューイング。アドレスに書き換えルールを適用し、メッセージを送信するチャネルを決定する。

詳細は、191 ページの「ディスパッチャ」を参照してください。

ルーティングとアドレス書き換え

メッセージは SMTP サーバーによってキューに入れられますが、変換チャネルや再処理チャネルなど、いくつかのほかのチャネルによってもキューに入れられることがあります。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。

- エイリアスを展開する。
- アドレスに書き換えルール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、書き換えルールと呼びます (193 ページの「書き換えルール」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、書き換えルールと同様、`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
tcp_auth
tcp_intranet-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、『Sun Java System Messaging Server Administration Reference』と第 12 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。

アドレスのローカル部分と受信フィールドは通常は 7 ビット文字なので注意してください。MTA がこれらのフィールドで 8 ビット文字を読み取った場合、8 ビットそれぞれをアスタリスクに変換します。

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ープロセスが 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 サーバープロセスを同時に実行し、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、それぞれのサーバーで 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 サーバープロセスを作成し、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、今後の着信接続を予測して、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、254 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。

サーバー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、新規サーバープロセスの作成や、アイドル状態の古いサーバー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、着信接続用に一定のサーバー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ー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、動作中のサーバー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、サーバープロセス間で接続を分散できます。MIN_CONNS はサーバープロセスが「十分にビジー」であることを示す接続数を指定し、MAX_CONNS はサーバープロセスが「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、現在のサーバープロセス数が MIN_PROCS 未満である場合、または既存のサーバープロセスがすべて「十分にビジー」(各サーバープロセスに対し、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ー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ー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ー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、254 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。

```
start-msg 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 start-msg コマンドが組み込まれています。そのため、組み込まれた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。

```
stop-msg dispatcher
```

ディスパッチャの終了時にサーバー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。

書き換えルール

書き換えルールには、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。

書き換えルール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。これは、次の 2 つから構成されます。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。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。

書き換えルール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。`$U` は、書き換えられたアドレスでも同じユーザー名を使うように指定します。`%` は、書き換えられたアドレスでも同じドメイン区切り文字を使用するように指定します。`$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。詳細は、第 11 章「書き換えルールの設定」を参照してください。

書き換えルールの設定の詳細については、234 ページの「MTA 設定ファイル」および第 11 章「書き換えルールの設定」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ーに相当します。

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。

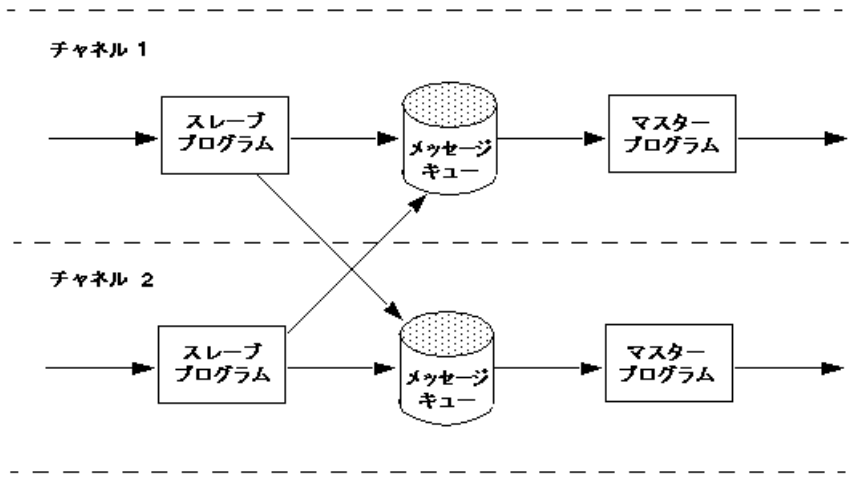
通常、スレーブプログラムは、MTA が外部要求に応答するための着信接続を受け入れます。スレーブチャネルプログラムには、以下のような機能があります。

- 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、書き換えルール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 8-3 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ルールを適用し、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。

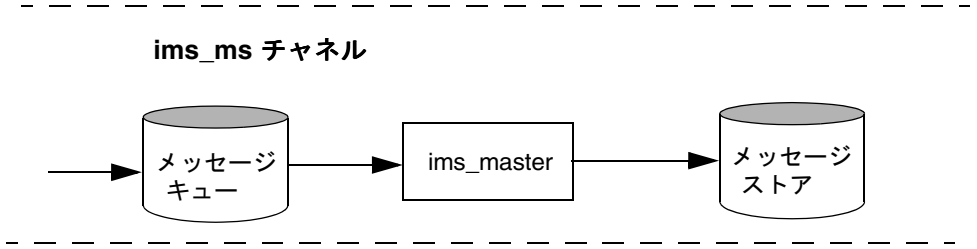
マスタープログラムは、キューからメッセージを取り出し、メッセージのネットワーク転送を開始します。ただし、マスタープログラムは、自分のチャネルキューにあるメッセージしか取り出せません。

図 8-3　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 8-4 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 8-4　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 *msg_svr_base*/data/queue/*channel*/* に保存されます。

警告　MTA キューディレクトリ (imta_tailor ファイル内の IMTA_QUEUE の値) に、ファイルまたはディレクトリを追加しないでください。これを行うと問題が発生します。MTA キューディレクトリに個別のファイルシステムを使用するときは、マウントポイントの下にサブディレクトリを作成し、そのサブディレクトリを IMTA_QUEUE の値として指定してください。

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、書き換えルールのあとに記載されています (234 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、書き換えルール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ルールで使用され、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、次の例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。

```
! test.cnf - An example configuration file.
!
! 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 tcp_auth
tcp_intranet-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 tcp_intranet-daemon はチャネルタグです。チャネルタグは、書き換えルール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、『Sun Java System Messaging Server Administration Reference』と第 12 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、常にファイルの最初のチャネルブロックでローカルチャネル l が示されます (例外は defaults チャネルであり、このチャネルはローカルチャネルの前に出現)。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、256 ページの「オプションファイル」および 254 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。設定チャネルの詳細については、第 12 章「チャネル定義を設定する」を参照してください。MTA チャネルの作成の詳細については、234 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ー、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 は LDAP ディレクトリに直接アクセスします。詳細は、第 9 章「MTA のアドレス変換とルーティング」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数が制限に達したためにジョブを開始できない場合は、ジョブコントローラは別のジョブが終了するまで待機します。ジョブ数の超過が解消されると、ジョブコントローラは別の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。プールの詳細については、257 ページの「ジョブコントローラファイル」および 375 ページの「チャネル実行ジョブの処理プール」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解放するのに十分な数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の設定とプールの詳細については、257 ページの「ジョブコントローラファイル」および 370 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。

```
start-msg 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

```
stop-msg 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 9 章

MTA のアドレス変換とルーティング

Messaging Server 6 2003Q4 より前の Messaging Server では、LDAP サーバーに保存された情報からコンパイルされたデータベースより、すべてのユーザー、ドメイン、およびグループデータにアクセスしていました。LDAP サーバーでディレクトリ情報が更新されると、データベース情報は `dirsync` というプログラムによって同期化されていました。Messaging Server MTA は、LDAP ディレクトリに直接アクセスします。この章では、ダイレクト LDAP データアクセスを使用する MTA 内のデータフローについて説明します。この章には、以下の節があります。

ダイレクト LDAP のアルゴリズムと実装

ここでは、ダイレクト LDAP 処理について説明します。

ドメインローカリティの判別

アドレスの変換とルーティングのプロセスでは、*user@domain* という形式のアドレスを元に、*domain* がローカルであるかどうかが最初にチェックされます。

書き換えルールの機能

MTA の書き換えルールには、提示された文字列がローカルで処理する必要のあるドメインであるかどうかをチェックする機能が追加されています。この新機能は、メタキャラクタ $V または $Z によってアクティブ化されます。これらの新しいメタキャラクタは、その後にパターン文字列が続くという点で、構文的には従来のメタキャラクタ $N、$M、$Q、および $C と同様です。$N、$M、$Q、および $C の場合、パターンはソースチャネルまたは宛先チャネルのいずれかと照合されます。$V および $Z の場合、パターンはドメインであり、チェック内容はそのドメインがローカルであるかどうかです。$V によってローカルドメインでない場合にルールエラーが発生し、$Z によってローカルドメインの場合にルールエラーが発生します。

これらのメタキャラクタの処理は、次の手順で実装します。

1. Messaging Server は、現在のドメインがディレクトリ内の有効なドメインエントリと一致するかどうかをチェックします。エントリが存在しない場合は、手順 3 に進みます。

2. そのドメインのエントリがディレクトリ内にある場合、LDAP_DOMAIN_ATTR_ROUTING_HOSTS MTA オプション (デフォルトは mailRoutingHosts) で指定されている属性がドメインエントリから取得されます。この属性が存在する場合、このドメイン内のユーザーを処理できるホストの一覧が示されます。この一覧は、local.hostname configutil パラメータで指定されているホストおよび local.imta.hostnamealiases configutil パラメータで指定されているホストの一覧と比較されます。これらのオプションはそれぞれ、LDAP_LOCAL_HOST および LDAP_HOST_ALIAS_LIST の各 MTA オプションで指定変更できます。一致するものがある場合またはドメインに属性が存在しない場合、ドメインはローカルです。一致するものがない場合、ドメインはローカルではありません。

   mailRoutingHosts 属性が原因でローカルでないと見なされるドメインの処理は、ROUTE_TO_ROUTING_HOST MTA オプションの設定によって異なります。このオプションが 0 (デフォルト) に設定されている場合、アドレスはそのままローカルでないものとして扱われ、MTA の書き換えルールを使用してルーティングが決定されます。このオプションが 1 に設定されている場合、LDAP_DOMAIN_ATTR_ROUTING_HOSTS MTA オプションで最初にリストされている値から成るソースルートがアドレスの先頭に追加されます。

3. ドメインエントリが見つからない場合、ドメインの左側の構成要素が削除され、手順 1 に戻ります。残っている構成要素がない場合は、手順 4 に進みます。

   ドメインツリーの上位にさかのぼった結果、domain.com がローカルとして認識された場合、domain.com のサブディレクトリはすべてローカルとして認識されます。この方法が不適当な状況が発生する可能性もあるため、この動作を制御する MTA オプション DOMAIN_UPLEVEL が提供されています。具体的には、DOMAIN_UPLEVEL のビット 0 (値 = 1) を設定解除すると、ドメインの構成要素を削除して再試行する動作は無効になります。DOMAIN_UPLEVEL のデフォルト値は 0 です。

4. この時点で、バニティドメインのチェックを実行する必要があります。バニティドメインにはドメインエントリがありません。代わりに、特殊なドメイン属性を 1 つまたは複数のユーザーエントリに付加することで、バニティドメインを指定します。バニティドメインのチェックは、DOMAIN_MATCH_URL MTA オプションで指定されている LDAP URL を使用して LDAP 検索を開始することによって実行されます。このオプションの値は次の値に設定する必要があります。

   ldap:///$B?msgVanityDomain?sub?(msgVanityDomain=$D)

   $B によって local.ugldapbasedn configutil パラメータの値が置き換えられます。これはディレクトリ内のユーザーツリーのベースです。LDAP_USER_ROOT MTA オプションを使用すると、この MTA 専用の configutil オプションの値を変更できます。

   この検索で実際に返される値は重要ではありません。重要なのは、返される値があるかどうかです。返される値がある場合は、ドメインはローカルであると見なされます。返される値がない場合は、ドメインはローカルではないと見なされます。

ドメインローカリティのドメインマップの判別

ディレクトリ内の有効なドメインエントリを検索するためにどのような処理が実行されるかを知っておくことも有益です。処理はスキーマレベルに固有です。Sun LDAP Schema 1 の場合は、次のとおりです。

1. ドメインをドメインツリーのベース DN に変換します。これは、ドメインを一連の dc コンポーネントに変換し、ドメインルートサフィックスを追加することによって実行されます。デフォルトのサフィックスは、service.dcroot configutil パラメータから取得されます。デフォルトのサフィックスは o=internet です。a.b.c.d という形式のドメインは一般に、dc=a,dc=b,dc=c,dc=d,o=internet に変換されます。service.dcroot configutil パラメータは、LDAP_DOMAIN_ROOT MTA オプションを設定することで無効にできます。

2. 手順 1 で見つかったベース DN を持ち、inetDomain または inetDomainAlias のいずれかのオブジェクトクラスを持つエントリを検索します。このために使用される検索フィルタは、LDAP_DOMAIN_FILTER_SCHEMA1 MTA オプションを設定することで無効にできます。このオプションを使用すると、デフォルトの (|(objectclass=inetDomain)(objectclass=inetdomainalias)) に戻ります。

3. 何も見つからない場合は、エラー終了します。

4. エントリのオブジェクトクラスが見つかり、それが inetDomain である場合、ドメインエントリに関連付けられた inetDomainBaseDn 属性が存在するかどうかをチェックします。存在する場合は、あとでユーザーエントリの検索に使用できるように保存され、処理は終了します。存在しない場合、エントリはドメインエイリアスであると見なされ、処理は手順 5 に進みます。MTA オプション LDAP_DOMAIN_ATTR_BASEDN を使用すると、inetDomainBaseDN の使用が無効になります。

5. エントリはドメインエイリアスであるはずなので、aliasedObjectName 属性によって参照されているエントリを検索し、手順 4 に戻ります。aliasedObjectName 属性が存在しない場合、処理はエラー終了します。aliasedObjectName 属性の使用に代わる手段は、MTA オプション LDAP_DOMAIN_ATTR_ALIAS を使用して指定することができます。

   処理が手順 4 に戻るのは、1 回だけです。ドメインエイリアスでさらにドメインエイリアスを参照することは許されていません。

Sun LDAP Schema 2 で実行される処理は、上記の処理より簡単です。ディレクトリ内でオブジェクトクラス sunManagedOrganization を持つエントリが検索されます。ここではドメインは sunPreferredDomain 属性または associatedDomain 属性のいずれかの値として示されています。この目的のための sunPreferredDomain および associatedDomain の各属性の使用は、必要に応じてそれぞれ、MTA オプション LDAP_ATTR_DOMAIN1_SCHEMA2 および LDAP_ATTR_DOMAIN2_SCHEMA2 で無効にできます。検索は、service.dcroot configutil パラメータで指定されているルートの下で実行されます。service.dcroot configutil パラメータは、LDAP_DOMAIN_ROOT MTA オプションを設定することで無効にできます。また、Schema 2 のドメインエントリに inetDomainBaseDn 属性は必須ではありません。ドメインエントリにこの属性がない場合、ユーザーツリーのベース自身がドメインエントリであると見なされます。

ドメインローカリティ情報のキャッシュ

ドメイン書き換え処理が実行される頻度とディレクトリ照会 (特にバニティドメインチェック) の負担から、ドメインについての情報は、否定的なものと肯定的なものの両方をキャッシュする必要があります。これは、開鎖型の、動的に拡張されたメモリ内ハッシュテーブルを使用して実装します。キャッシュの最大サイズは DOMAIN_MATCH_CACHE_SIZE MTA オプションで設定します (デフォルトは 100000)。キャッシュ内のエントリのタイムアウトは DOMAIN_MATCH_CACHE_TIMEOUT MTA オプションで設定します (デフォルトは 600 秒)。

エラー処理

サーバーエラーが発生すると、ドメインがローカルであるかどうかを判別することができなくなるため、このプロセス時の一時的なサーバーエラーには慎重に対処する必要があります。このような場合、一般的に次の 2 つの結果がもたらされる可能性があります。

1. 一時 (4*xx*) エラーをクライアントに返し、あとでそのアドレスを使用して再試行するように指示する。
2. アドレスを受け入れるが、再処理チャネルのキューに入れ、あとでローカルで再試行できるようにする。

上記の選択肢はいずれも、すべての場合に適切であるとは限りません。たとえば、結果 1 は、リモート SMTP リレーと通信している場合に適切です。一方、結果 2 は、ローカルユーザーからの SMTP 送信を処理している場合に適切です。

同じパターンを持つ複数のルールを使用して一時エラーを処理することは理論的には可能ですが、このような照会を繰り返すことによるオーバーヘッドは、キャッシュが配置されている場合であっても容認できるものではありません。したがって、ドメイン書き換えでは、成功または失敗して次のルールに進むという単純な照合方式は不適切です。ドメイン検索が失敗した場合は、代わりに、MTA オプション DOMAIN_FAILURE で指定されている特殊なテンプレートが使用されます。$V の処理が失敗すると、このテンプレートが、現在処理されている書き換えルールテンプレートの残りの部分の代わりに使用されます。

ドメインチェック書き換えルールのパターン

このドメインチェックは、ほかの書き換えルールが起動する前に実行される必要があります。この順序は、ルールの左側に特別な $* を配置することによって確保されます。$* パターンは、ほかのどのルールよりも先にチェックされます。

すべてのメカニズムを統合する

上記に示したすべての機能を考慮すると、imta.cnf で必要とされる新しい書き換えルールは次のようになります。

```
$*        $E$F$U%$H$V$H@localhost
```

また、option.dat ファイルの DOMAIN_FAILURE MTA オプションの値は、次のように設定されている必要があります。

```
reprocess-daemon$Mtcp_local$1M$1~-error$4000000?Temporary lookup
failure
```

この書き換えルールでは、localhost はローカルチャネルに関連付けられているホスト名です。ここで示した DOMAIN_FAILURE オプションの値は、デフォルト値であるので、通常の状況では option.dat に記述する必要はありません。

ここでの順序付けは特に複雑です。MTA は、アドレスが再構築された後、ただしルートが追加される前に $V をチェックします。これによって、MTA は、一時的な検索エラーが発生した場合にルートを変更することができます。保留中チャネルの一致チェックは、挿入ポイントが変更されるたびに適用されます。これにより、2 番目の $H に続く @ がチェックを開始します。このチェックが成功した場合、テンプレートの残りの部分が適用され、書き換え処理が終了します。このチェックが失敗した場合、書き換えは失敗し、次の適用可能な書き換えルールで書き換えが続行されます。一時的なエラーが原因でチェックが実行できない場合、テンプレート処理は、DOMAIN_FAILURE MTA オプションで指定されている値を使用して続行されます。このテンプレートの値によって、まずルーティングホストが reprocess-daemon に設定されます。次に、MTA が何らかの再処理チャネルまたは tcp_local を処理しているか

どうかがチェックされます。MTA がこのようなチャネルを処理している場合、ルールは継続し、ルーティングホストが無効とされ、一時的なエラーが結果として示されます。MTA がこのようなチャネルを処理していない場合、ルールは打ち切られて正常に終了します。その結果、再処理チャネルへのアドレスは書き換えられます。

ローカルアドレスのエイリアス展開

アドレスがローカルチャネルに関連付けられていると決定されると、アドレスのエイリアス展開が自動的に実行されます。エイリアス展開プロセスでは、大量の情報ソースを調べます。これには次の情報ソースが含まれます。

1. エイリアスファイル (コンパイルされた設定の一部)。
2. エイリアスデータベース。
3. エイリアス URL。

正確にどのエイリアスソースがチェックされるか、およびチェックされる順序については、option.dat ファイルの ALIAS_MAGIC MTA オプションによって決まります。ダイレクト LDAP では、このオプションを 8764 に設定します。これによって、ALIAS_URL0 MTA オプションで指定されている URL が先にチェックされ、以降は ALIAS_URL1 MTA オプションで指定されている URL、ALIAS_URL2 MTA オプションで指定されている URL、エイリアスファイルの順序でチェックされます。エイリアスデータベースは、この設定がアクティブな場合はチェックされません。

LDAP URL を使用するエイリアスチェック

LDAP のエイリアスチェックは、2 つの特殊な LDAP URL をエイリアス URL として指定することで実装されます。最初の URL では通常のユーザーとグループが処理され、後続のエイリアス URL ではバニティドメインが処理されます。最初の URL を ALIAS_URL0 として、次のように指定します。

```
ALIAS_URL0=ldap:///$V?*?sub?$R
```

$V メタキャラクタ

メタキャラクタの展開は、URL 検索より先に実行されます。ALIAS_URL0 値に使用されている 2 つのメタキャラクタは $V と $R です。

$V メタキャラクタは、アドレスのドメイン部分をベース DN に変換します。これは、前出の「書き換えルールの機能」で説明されている、$V 書き換えルールメタキャラクタによって実行される最初の処理と似ています。$V 処理では、次の手順が実行されます。

1. 現在のドメインのユーザーエントリのベース DN を取得します。

2. 現在のドメインに関連付けられている標準ドメインを取得します。Sun LDAP Schema 1 では、ドメインエントリの inetCanonicalDomainName 属性が存在する場合、この属性で標準ドメイン名が指定されています。この属性がない場合、標準ドメイン名は実際のドメインエントリの DN から率直な方法で構築された名前になります。この名前は、実際のドメインがエイリアスである場合、実際のドメインとは異なります。標準ドメイン名を保存するために使用される名前属性は、option.dat ファイルの LDAP_DOMAIN_ATTR_CANONICAL MTA オプションで無効にできます。

   Sun LDAP Schema 2 では、標準名は SunPreferredDomain 属性の値です。

3. ベース DN が存在する場合は、ベース DN が URL の $V と置き換えられます。

4. この時点で、このエントリの適用可能なすべてのホストしているドメインが判別されます。これは、標準ドメイン (DOMAIN_UPLEVEL のビット 2 (値 = 4) が設定解除されている場合) または現在のドメイン (DOMAIN_UPLEVEL のビット 2 (値 = 4) が設定されている場合) のいずれかを service.defaultdomain configutil パラメータと比較することによって実行されます。一致しない場合、エントリはホストしているドメインのメンバーです。service.defaultdomain configutil パラメータは、option.dat ファイルにある LDAP_DEFAULT_DOMAIN MTA オプションを設定することで無効にできます。

5. ベース DN の判別が失敗した場合、ドメインの左側の構成要素が削除され、手順 1 に戻ります。構成要素が残っていない場合、置換は失敗します。

$V は、オプションの数値引数も受け入れます。1 に設定されている場合 (たとえば、$1V)、ドメインツリーのドメイン解決が失敗したことは無視され、local.ugldapbasedn configutil オプションで指定されたユーザーツリーのベースが返されます。

ドメインのベース DN の取得に成功すると、MTA はあとで必要になるいくつかのドメイン属性も取得します。取得される属性の名前は、option.dat ファイルにある次の MTA オプションで設定します。

- LDAP_DOMAIN_ATTR_UID_SEPARATOR (デフォルト domainUidSeparator)
- LDAP_DOMAIN_ATTR_SMARTHOST (デフォルト mailRoutingSmartHost)
- LDAP_DOMAIN_ATTR_CATCHALL_ADDRESS (デフォルト mailDomainCatchallAddress)
- LDAP_DOMAIN_ATTR_BLOCKLIMIT (デフォルト mailDomainMsgMaxBlocks)
- LDAP_DOMAIN_ATTR_REPORT_ADDRESS (デフォルト mailDomainReportAddress)
- LDAP_DOMAIN_ATTR_STATUS (デフォルト inetDomainStatus)
- LDAP_DOMAIN_ATTR_MAIL_STATUS (デフォルト mailDomainStatus)
- LDAP_DOMAIN_ATTR_CONVERSION_TAG (デフォルト mailDomainConversionTag)
- LDAP_DOMAIN_ATTR_FILTER (デフォルト mailDomainSieveRuleSource)

- LDAP_DOMAIN_ATTR_DISK_QUOTA (デフォルトなし)
- LDAP_DOMAIN_ATTR_MESSAGE_QUOTA (デフォルトなし)
- LDAP_DOMAIN_ATTR_AUTOREPLY_TIMEOUT (デフォルトなし)
- LDAP_DOMAIN_ATTR_NOSOLICIT (デフォルトなし)
- LDAP_DOMAIN_ATTR_OPTIN (デフォルトなし)
- LDAP_DOMAIN_ATTR_RECIPIENTLIMIT (デフォルトなし)
- LDAP_DOMAIN_ATTR_RECIPIENTCUTOFF (デフォルトなし)
- LDAP_DOMAIN_ATTR_SOURCEBLOCKLIMIT (デフォルトなし)

URL からマッピングを呼び出す

ドメインからベース DN へのマッピングを別の方法で実行するまれなケースが発生する場合があります。このようなケースに対応するために、URL 解決プロセスには、MTA マッピングを呼び出す機能があります。これは、次の一般的な形式の一連のメタキャラクタ列を使用して実行されます。

$|/*mapping-name*/*mapping-argument*|

二重引用符 (") 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あり、マッピング名または引数のいずれかで使用される文字ではないものを選択する必要があります。

$R メタキャラクタ

$R メタキャラクタは、URL 用に適切なフィルタを提供します。目的は、特定のユーザーまたはグループの電子メールアドレスを含んでいる可能性のあるすべての属性を検索するフィルタを生成することです。検索対象になる属性のリストは、configutil パラメータ local.imta.mailaliases で指定します。このパラメータが設定されていない場合は、configutil パラメータ local.imta.schematag が調べられ、その値に応じて適切なデフォルト属性の集合が次のように選択されます。

```
sims401      mail,rfc822mailalias
nms41        mail,mailAlternateAddress
ims50        mail,mailAlternateAddress,mailEquivalentAddress
```

local.imta.schematag の値はコンマ区切りのリストにできます。複数のスキーマがサポートされている場合は、組み合わせて重複を削除した属性のリストが使用されます。LDAP_SCHEMATAG MTA オプションは、MTA 専用の local.imta.schematag の設定を無効にするために使用できます。

また、フィルタは、最初に指定されたアドレスを検索するだけではなく、同じローカル部分を持ちながらドメイン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。このドメインは「$V メタキャラクタ」の手順 2 で保存されたものです。ドメインツリー検索の反復性は、この 2 つのアドレスが異なる可能性があることを意味します。この追加チェックは、option.dat ファイルにある DOMAIN_UPLEVEL MTA オプションのビット 1 (値 = 2) によって制御されます。このビットを設定すると、追加アドレスチェックが有効になります。DOMAIN_UPLEVEL のデフォルト値は 0 です。

たとえば、ドメイン siroe.com がドメインツリーに表示されると仮定します。Sun LDAP Schema 1 が有効であり、次のアドレスを検索すると仮定します。

```
u@host1.siroe.com
```

$R および ims50 schematag の展開の結果から得られるフィルタは、次のようになります。

```
(|(mail=u@siroe.com)
      (mail=u@host1.siroe.com)
      (mailAlternateAddress=u@siroe.com)
      (mailAlternateAddress=u@host1.siroe.com)
      (mailEquivalentAddress=u@siroe.com)
      (mailEquivalentAddress=u@host1.siroe.com))
```

また、DOMAIN_UPLEVEL が 3 ではなく 1 に設定されている場合、フィルタは次のようになります。

```
(|(mail=u@host1.siroe.com)
      (mailAlternateAddress=u@host1.siroe.com)
      (mailEquivalentAddress=u@host1.siroe.com))
```

フェッチする属性を決定する

返される属性のリストとして * が URL で指定されている場合、アスタリスクを MTA が使用できる属性のリストで置き換えます。このリストは、MTA が使用するオプションを指定する、各種の MTA オプション設定から動的に生成されます。

LDAP エラーを処理する

この時点で、結果の URL を使用して LDAP 検索が実行されます。何らかの LDAP エラーが発生した場合、処理は一時的なエラー (4xx error in SMTP) を示して終了します。LDAP 操作は成功したものの、結果の生成に失敗した場合は、LDAP_DOMAIN_ATTR_CATCHALL_ADDRESS MTA オプションから取得される、ドメインのキャッチオールアドレス属性がチェックされます。この属性が設定されている場合は、その値で現在のアドレスが置き換えられます。

キャッチオールアドレス属性が設定されていない場合は、LDAP_DOMAIN_ATTR_SMARTHOST MTA オプションから取得される、ドメインのスマートホスト属性がチェックされます。この属性が設定されている場合、次の形式のアドレスが作成されます。

@*smarthost*:*user*@*domain*

エイリアス処理はこの結果で正常終了します。また、LDAP_DOMAIN_ATTR_CONVERSION_TAG MTA オプションから取得する、ドメインの変換タグ (存在する場合) がアドレスに追加されます。これによって、スマートホストへの転送前に変換が実行されます。ドメインのキャッチオールアドレスまたはスマートホストがない場合は、このエイリアス URL の処理はエラー終了します。

LDAP 結果のサニティチェック

LDAP 検索の結果が返された後、検索結果に 1 つのエントリのみが存在することを確認するチェックが実行されます。複数のエントリが存在する場合は、各エントリがユーザーまたはグループにとって正しいオブジェクトクラスを持っているか、deleted ステータスになっていないか、ユーザーの場合は UID があるかどうかがチェックされます。このチェックに合格しないエントリは無視されます。このチェックによって複数のエントリが 1 つに絞られた場合、処理は続行されます。それ以外の場合は、重複するディレクトリまたはあいまいなディレクトリであることを示すエラーが返されます。

バニティドメインのサポート

ALIAS_URL0 チェックは、標準的なユーザーまたはホストしているドメイン内のユーザーのためのチェックです。このチェックが失敗すると、バニティドメインチェックも実行されます。これは、次のエイリアス URL を使用して実行されます。

```
ALIAS_URL1=ldap:///$B?*?sub?(&(msgVanityDomain=$D)$R)
```

キャッチオールアドレスのサポート

@*host* という形式のキャッチオールアドレスのチェックは、mailAlternateAddress 属性で設定する必要があります。この形式のワイルドカード指定は、ホストしているドメインおよびバニティドメインの両方で許可されています。この場合の適切なエイリアス URL は次のとおりです。

```
ALIAS_URL2=ldap:///$1V?*?sub?(mailAlternateAddress=@$D)
```

注 +* サブアドレス置換メカニズムは、常にダイレクト LDAP モードでキャッチオールアドレスとともに動作していましたが、置換される文字列はローカル部分全体ではなく、サブアドレスのみでした。このメカニズムは変更されて、元のアドレスのローカル部分全体がサブアドレスとしてキャッチオールアドレスにプラグインされるようになりました。

たとえば、foo+bar@domain.com という形式のアドレスを想定する場合、domain.com ドメインにはローカルユーザー foo は存在せず、domain.com のキャッチオールアドレスが bletch+*@example.com である場合、結果としてアドレスは bletch+foo+bar@example.com となります。これは、以前は bletch+bar@example.com でした。

LDAP 結果を処理する

LDAP エイリアス結果は、順序依存性のあるいくつかの段階で処理されます。以下の項目で、これらの段階について説明します。

オブジェクトクラスチェック

エイリアス検索が成功した場合、エントリのオブジェクトクラスがチェックされ、ユーザーまたはグループに適したオブジェクトクラスのセットを含んでいることが確認されます。ユーザーおよびグループに必要なオブジェクトクラスのセットは、通常、どのスキーマがアクティブであるかよって異なります。これは、local.imta.schematag 設定で決定されます。

表 9-1 に、さまざまな schematag 値から得られるユーザーおよびグループのオブジェクトクラスを示します。

表 9-1 さまざまな schematag 値から得られるオブジェクトクラス

schematag	ユーザーオブジェクトクラス	グループオブジェクトクラス
sims40	inetMailRouting+inetmailuser	inetMailRouting+inetmailgroup
nms41	mailRecipient + nsMessagingServerUser	mailGroup
ims50	inetLocalMailRecipient+inetmailuser	inetLocalMailRecipient+inetmailgroup

この表の情報は、ほかのスキーマタグの処理と同様に、ハードコード化されています。ただし、option.dat ファイルには、LDAP_USER_OBJECT_CLASSES と LDAP_GROUP_OBJECT_CLASSES の 2 つの MTA オプションがあり、別のオブジェクトクラスのセットを指定することが可能です。前者はユーザー用、後者はグループ用です。

たとえば、ims50,nms41 のスキーマタグ設定は、次のオプション設定と同等です。

```
LDAP_USER_OBJECT_CLASSES=inetLocalMailRecipient+inetmailuser,
mailRecipient+nsMessagingServerUser
```

```
LDAP_GROUP_OBJECT_CLASSES=inetLocalMailRecipient+inetmailgroup,mail
Group
```

LDAP 結果にユーザーまたはグループに適した正しいオブジェクトセットがない場合、LDAP 結果は無視されます。MTA は、ユーザーまたはグループを処理しているかどうかを判断し、この情報を保存します。保存された情報は、あとで繰り返し使用されます。

上記で説明したオブジェクトクラス設定は、ユーザーまたはグループに適した正しいオブジェクトクラスがエントリにあるかどうかをチェックするために使用できる、実際の LDAP 検索フィルタを構築するためにも使用されます。このフィルタは、$K メタキャラクタを経由してアクセスできます。また、コマンド imsimta cnbuild -option が使用されたときの LDAP_UG_FILTER と同様、チャネルプログラムで使用するために MTA の設定に内部的に保存され、MTA オプションファイル option.dat に書き込まれます。このオプションは、ファイルに書き込むだけです。MTA がオプションファイルから読み取ることはありません。

エントリステータスチェック

次のエントリのステータスがチェックされます。2 つのステータス属性があり、1 つは一般的なエントリ用、もう 1 つはメールサービス専用です。

表 9-2 に、有効化されているスキーマに応じてチェック対象になる、schematag エントリ内の一般およびメール固有のユーザー属性またはグループ属性を示します。

表 9-2 チェック対象の属性

schematag	タイプ	一般	メール固有
sims40	ユーザー	inetsubscriberstatus	mailuserstatus
sims40	グループ	none	inetmailgroupstatus
nms41	ユーザー	none	mailuserstatus
nms41	グループ	none	none
Messaging Server 5.0	ユーザー	inetuserstatus	mailuserstatus
Messaging Server 5.0	グループ	none	inetmailgroupstatus

必要に応じて、option.dat ファイルにある LDAP_USER_STATUS および LDAP_GROUP_STATUS の MTA オプションを使用して、別の一般ステータス属性を選択することができます。前者はユーザー用、後者はグループ用です。メール固有のユーザーおよびグループのステータス属性は、LDAP_USER_MAIL_STATUS および LDAP_GROUP_MAIL_STATUS の各 MTA オプションで制御します。

このチェックで使用されるもう 1 つの要素は、ドメイン自体のステータス (LDAP_DOMAIN_ATTR_STATUS および LDAP_DOMAIN_ATTR_MAIL_STATUS) です。全部で 4 つのステータス属性があります。これらのステータスは、次に示す順序で考慮されることによって組み合わせられます。

1. ドメインステータス
2. ドメインメールステータス
3. ユーザーまたはグループのステータス
4. メールユーザーまたはメールグループのステータス

これらのうち、「active」以外のステータスを示す最初のステータスは、ほかのステータスより優先されます。これ以外に許容されるステータス値は、「inactive」、「deleted」、「removed」、「disabled」、「hold」、および「overquota」です。「hold」、「disabled」、および「removed」ステータスは、メールドメイン、メールユーザー、またはメールグループのみに指定されます。「overquota」ステータスは、メールドメインステータスまたはメールユーザーステータスとしてのみ指定されます。

特定のステータス属性が存在しない場合、すべてのステータスはデフォルトの「active」になります。不明なステータス値は、「inactive」として解釈されます。

4 つのステータスが組み合わせられると、ユーザーまたはグループに次のステータスが可能になります。「active」、「inactive」、「deleted」、「removed」、「disabled」、「hold」、および「overquota」。active ステータスの場合、エイリアス処理が続行されます。inactive または overquota ステータスの場合、4xx (一時的) エラーが発生し、アドレスはただちに拒否されます。deleted、removed、および disabled ステータスの場合、5xx (永続的) エラーが発生し、アドレスはただちに拒否されます。hold ステータスの場合、ステータス処理に関しては active として扱われますが、内部フラグが設定されます。これによって、あとで配信オプションが考慮される際、既存のオプションはいずれも、単一の「hold」エントリが含まれているオプションリストで上書きされます。

UID チェック

次に必要な処理は、エントリの UID を考慮することです。UID はさまざまな目的で使用されます。UID はユーザーエントリの一部である必要があり、グループエントリに含まれていることもあります。UID がないユーザーエントリは無視され、このエイリアス URL の処理はエラー終了します。ホストしているドメインのエントリの UID は、

実 UID、区切り文字、およびドメインで構成できます。MTA では実 UID のみを必要とするので、ほかの構成要素が存在する場合は、option.dat ファイルにある LDAP_DOMAIN_ATTR_UID_SEPARATOR MTA オプションで取得したドメイン区切り文字を使用して削除されます。

あまりないことですが、uid 以外の属性で UID が保存される場合には、別の属性を使用するように LDAP_UID MTA オプションで設定できます。

メッセージの取得

次に、メッセージ取得アドレスを指定するために使用される LDAP 属性がチェックされます。この目的で使用される属性は、LDAP_CAPTURE MTA オプションで指定されている必要があります。デフォルトはありません。この属性の値はアドレスとして扱われます。特殊な「取得」通知が生成され、このアドレスに送信されます。この通知には、現在のメッセージが添付されています。また、取得アドレスは、アドレスが以後、エンベロープ from: アドレスとして表示される場合に、アドレスリバースキャッシュをシードするために使用されます。

リバースキャッシュをシードする

次に、プライマリアドレスおよびユーザーエントリに添付されたエイリアスが考慮されます。この情報は、アドレスリバースキャッシュをシードするために使用されます。この情報は、現在のアドレス変換プロセスでは使用されません。最初に、プライマリアドレス、個人名、受取人制限、受取人の遮断、およびソースブロック制限の各属性が考慮されます。プライマリアドレスは通常、mail 属性に保存されています。別の属性は、LDAP_PRIMARY_ADDRESS MTA オプションを適切に設定することによって指定できます。当然、プライマリアドレスはそれ自身にリバースされます。これ以外の属性には、デフォルトの属性はありません。これらの属性を使用する場合は、LDAP_PERSONAL_NAME (528 ページの「不在返信メッセージの自動返信の属性」を参照)、LDAP_RECIPIENTLIMIT、LDAP_RECIPIENTCUTOFF (408 ページの「メッセージの受取人を制限する」を参照)、および LDAP_SOURCEBLOCKLIMIT (404 ページの「絶対的なメッセージサイズ制限を指定する」を参照) の各 MTA オプションで指定する必要があります。このときに、対応するドメインレベルの受取人制限、受取人の遮断、ソースブロック制限の各属性も考慮されます。ユーザーレベルの設定は、ドメインレベルの設定より完全に優先されます。

次に、セカンダリアドレスが考慮され、各セカンダリアドレスのキャッシュエントリが作成されます。セカンダリアドレスには 2 種類あります。アドレスリバースの対象になるものと、ならないものです。両者とも、アドレスリバースキャッシュを適切にシードするためには考慮される必要があります。メッセージ取得要求があるかどうかをあらゆる場合にチェックする必要があるためです。

リバース対象になるセカンダリアドレスは通常、mailAlternateAddress 属性に保存されています。別の属性は、LDAP_ALIAS_ADDRESSES MTA オプションで指定できます。リバース対象にならないセカンダリアドレスは通常、mailEquivalentAddress 属性に保存されています。別の属性は、LDAP_EQUIVALENCE_ADDRESSES MTA オプションで指定できます。

メールホストおよびルーティングアドレス

ここでは、mailhost および mailRoutingAddress の各属性が考慮されます。考慮される実際の属性は、LDAP_MAILHOST および LDAP_ROUTING_ADDRESS の各 MTA オプションで変更できます。これらの属性は同時に機能し、現時点でアドレスを有効化するべきかどうか、または別のシステムに転送するべきかどうかを決定します。

最初に、mailhost がこのエントリにとって有効であるかどうかが判断されます。エントリ対してアクティブな配信オプションの事前チェックは、エントリがメールホスト固有であるかどうかを確認するために実行されます。メールホスト固有でない場合、mailhost チェックは省略されます。このチェック方法については、217 ページの「配信オプションの処理」を参照し、特に # フラグについての説明を確認してください。

ユーザーエントリの場合、mailhost 属性を有効にするには、この属性がローカルシステムを特定している必要があります。mailhost 属性は、local.hostname configutil パラメータの値および local.imta.hostnamealiases configutil パラメータによって指定されている値のリストと比較されます。mailhost 属性は、これらのいずれかと一致した場合、ローカルホストを特定していると見なされます。

一致が見つかった場合、エイリアスをローカルで有効にすることができ、エイリアス処理は続行されます。一致が見つからない場合、メッセージを有効にするには、メールホストに転送する必要があります。次の形式の新しいアドレスが構築されます。

@mailhost : user@domain

これがエイリアス展開操作の結果になります。

欠落している mailhost 属性の処理は、エントリがユーザーであるかグループであるかによって異なります。ユーザーの場合、メールホストは不可欠であり、mailhost 属性が存在しない場合は次の形式の新しいアドレスが構築されます。

@smarthost : user@domain

このとき、LDAP_DOMAIN_ATTR_SMARTHOST MTA オプションによって決定されたドメインのスマートホストが使用されます。ドメインのスマートホストが存在しない場合は、エラーが表示されます。

グループの場合、メールホストは必須ではなく、メールホストの欠落は、任意の場所でグループが拡張可能であるという意味に解釈されます。したがって、エイリアス処理は続行されます。

mailRoutingAddress 属性によって、最後に 1 つ問題が追加されます。この属性が存在する場合、エイリアス処理は mailRoutingAddress を結果として終了します。ただし、メールホストが存在する場合、そのメールホストが mailRoutingAddress にソースルートとして追加されます。

その他の属性のサポート

次に、mailMsgMaxBlocks 属性が考慮されます。最初に、この属性は、LDAP_DOMAIN_ATTR_BLOCKLIMIT MTA オプションから返されたドメインのブロック制限で最小化されます。現在のメッセージのサイズが制限を超過していると認識された場合、エイリアス処理はサイズ超過エラーで終了します。サイズが不明である場合、または制限を超過していない場合、この制限は保存され、あとでメッセージ自体がチェックされるときに再チェックされます。mailMsgMaxBlocks の使用は、LDAP_BLOCKLIMIT MTA オプションで変更できます。

次に、いくつかの属性に対してアクセスと保存が行われます。最終的には、これらの属性はキューファイルエントリに書き込まれ、ims_master チャネルプログラムによって使用されます。このプログラムはその後、この属性を使用してストアのユーザー情報キャッシュを更新します。個々のユーザーの属性が見つからない場合、ドメインレベルの属性を使用してデフォルトを設定できます。

この処理は、LDAP エントリがユーザーではなくグループのものである場合、または LDAP エントリが LDAP ディレクトリではなくエイリアスキャッシュに由来する場合は、スキップされます。後者の基準の背後にある論理は、この情報を頻繁に更新することは不必要であるということと、エイリアスキャッシュを使用すれば、更新が行われるべき時期についての合理的な基準が提供されるということです。取得される属性の名前は、さまざまな MTA オプションによって設定されます。

表 9-3 に、取得されるディスク制限容量とメッセージ制限容量の各属性を設定する MTA オプションを示します。

表 9-3 取得されるディスク制限容量とメッセージ制限容量の各属性を設定する MTA オプション

MTA オプション	属性
LDAP_DISK_QUOTA	mailQuota
LDAP_MESSAGE_QUOTA	mailMsgQuota

次に、いくつかの属性があとでメタキャラクタの置換との関連で使用できるように保存されます。

表 9-4 に、MTA オプション、デフォルトの属性、およびメタキャラクタを示します。

表 9-4　MTA オプション、デフォルトの属性、メタキャラクタ

MTA オプション	デフォルトの属性	メタキャラクタ
LDAP_PROGRAM_INFO	mailProgramDeliveryInfo	$P
LDAP_DELIVERY_FILE	mailDeliveryFileURL	$F
LDAP_SPARE_1	デフォルトなし	$1E $1G $E
LDAP_SPARE_2	デフォルトなし	$2E $2G $G
LDAP_SPARE_3	デフォルトなし	$3E $3G
LDAP_SPARE_4	デフォルトなし	$4E $4G
LDAP_SPARE_5	デフォルトなし	$5E $5G

追加の属性用のスペアスロットが含まれています。これらを使用することによってカスタマイズされたアドレス拡張機能を構築できます。

次に、mailconversiontag 属性に関連付けられている値がすべて、現在の変換タグのセットに追加されます。この属性の名前は、LDAP_CONVERSION_TAG MTA オプションで変更できます。ドメインの mailDomainConversionTag 属性に値が関連付けられている場合は、その値も同様に追加されます。

配信オプションの処理

次に、mailDeliveryOption 属性がチェックされます。この属性の名前は、LDAP_DELIVERY_OPTION MTA オプションで変更できます。これは複数の値を指定できるオプションであり、この値によってエイリアス変換プロセスで生成されたアドレスが決まります。また、許可される値は、ユーザーとグループで異なります。両者に許可される値は、program、forward、および hold です。ユーザーにのみ許可される値は、mailbox、native、unix、および autoreply です。グループにのみ許可される値は、members、members_offline、および file です。

mailDeliveryOption 属性から適切なアドレスへの変換は、DELIVERY_OPTIONS MTA オプションによって制御されます。このオプションは、許可される mailDeliveryOption 値それぞれがどんなアドレスを生成するかだけではなく、mailDeliveryOption に許可される値は何か、またそれぞれの値がユーザー、グループ、あるいはその両方に該当するかどうかを指定します。

このオプションの値は、deliveryoption=template ペアのコンマ区切りのリストで構成され、各ペアにはオプションの単一文字のプレフィックスが 1 つまたは複数付いています。

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,                            ¥
     /hold=@hold-daemon:$A,                               ¥
     *unix=$M@native-daemon,                              ¥
     &file=+$F@native-daemon,                             ¥
     &@members_offline=*,                                 ¥
     program=$M%$P@pipe-daemon,                           ¥
     #forward=**,                                         ¥
     *^!autoreply=$M+$D@bitbucket
```

各配信オプションは、可能な mailDeliveryOption 属性値に対応します。対応するテンプレートは、URL 処理の場合と同じメタキャラクタの置換スキームを使用して結果のアドレスを指定します。

表 9-5 に、DELIVERY_OPTIONS オプションで使用可能な単一文字のプレフィックスを示します。

表 9-5 DELIVERY_OPTIONS MTA オプション内のオプションで使用する単一文字のプレフィックス

文字プレフィックス	説明
@	メッセージを再処理チャネルにリダイレクトする必要があることを示すフラグを設定します。現在のユーザーやグループの処理は中止されます。再処理チャネルから発信されるメッセージについては、このフラグは無視されます。
*	ユーザーに適用される配信オプション。
&	グループに適用される配信オプション。
$	このユーザーまたはグループの展開は遅延されることを示すフラグを設定します。
^	不在期間の開始と終了をチェックして配信オプションが有効化されているかどうかを確認する必要があることを示すフラグを設定します。
#	エントリの指定メールホストに対してこの配信オプションの展開を行う必要がないことを示すフラグを設定します。つまり、続くエントリはメールホストとは無関係です。これによって、MTA は、指定されたユーザーまたはグループのすべての配信オプションがメールホストと無関係かどうかを確認します。無関係である場合、MTA はメールホストにメッセージを送信する必要はなく、エントリで即座に動作できます。
/	この配信オプションによって生成されたすべてのアドレスを保留にするフラグを設定します。これらの受取人アドレスが記述されているメッセージファイルには、.HELD 拡張が追加されます。

表 9-5　DELIVERY_OPTIONS MTA オプション内のオプションで使用する単一文字のプレフィックス (続き)

文字プレフィックス	説明
!	自動返信が MTA によって内部的に処理される必要があることを示すフラグを設定します。このプレフィックスは、自動返信の配信オプションに使用した場合にのみ意味を持ちます。このオプションの値は、メッセージを bitbucket チャネルに送信するものである必要があります

* と & のいずれも存在しない場合、配信オプションは、ユーザーとグループの両方に適用されるものと見なされます。

配信オプションで使用するその他のメタキャラクタ

MTA の URL テンプレート機能の新しい使用方法をサポートするために、その他のメタキャラクタがいくつか追加されています。次のようなタスクがあります。

表 9-6 に、配信オプションで使用するその他のメタキャラクタとその説明を示します。

表 9-6　配信オプションで使用するその他のメタキャラクタ

メタキャラクタ	説明
$¥	後続のテキストを小文字にします。
$^	後続のテキストを大文字にします。
$_	後続のテキストの大文字と小文字を変換しません。
$*n*A	アドレスの *n* 番目の文字を挿入します。最初の文字は文字 0。*n* が省略されている場合は、アドレス全体が置換されます。このメタキャラクタは、自動返信ディレクトリパスを構築するために使用されます。
$D	アドレスのドメイン部分を挿入します。
$*n*E	*n* 番目のスペア属性の値を挿入します。*n* が省略されている場合は、最初の属性が使用されます。
$F	配信ファイル名 (mailDeliveryFileURL 属性) を挿入します。
$*n*G	*n* 番目のスペア属性の値を挿入します。*n* が省略されている場合は、2 番目の属性が使用されます。
$*n*H	元のアドレスのドメインの、0 から数えて *n* 番目のコンポーネントを挿入します。*n* が省略されている場合、デフォルトは 0。
$*n*I	エイリアスに関連付けられているホストしているドメインを挿入します。このメタキャラクタは、整数パラメータ *n* を受け入れます。このパラメータのセマンティクスについては、表 9-7 を参照。
$*n*J	ホストドメインの、0 から数えて *n* 番目の部分を挿入します。*n* のデフォルトは 0。

表 9-6 配信オプションで使用するその他のメタキャラクタ (続き)

メタキャラクタ	説明
$*n*O	現在のアドレスに関連付けられているソースルートを挿入します。このメタキャラクタは、整数パラメータ n を受け入れます。このパラメータのセマンティクスについては、表 9-7 を参照。
$K	ユーザーまたはグループのオブジェクトクラスと一致する LDAP フィルタを挿入します。出力専用の MTA オプション LDAP_UG_FILTER の説明を参照。
$L	アドレスのローカル部分を挿入します。
$*n*M	UID の *n* 番目の文字を挿入します。最初の文字は文字 0。n が省略されている場合は、UID 全体が置換されます。
$P	プログラム名 (mailProgramDeliveryInfo 属性) を挿入します。
$*n*S	現在のアドレスに関連付けられているサブアドレスを挿入します。このメタキャラクタは、整数パラメータ n を受け入れます。このパラメータのセマンティクスについては、表 9-7 を参照。
$*n*U	現在のアドレスのメールボックス部分から引用符が削除された形式での、n 番目の文字を挿入します。最初の文字は文字 0。n が省略されている場合は、引用符なしのメールボックス全体が置換されます。
$*n*X	メールホストの *n* 番目のコンポーネントを挿入します。*n* が省略されている場合は、メールホスト全体が挿入されます。

表 9-7 に、各整数パラメータに対応する $nI および $nS のメタキャラクタの動作を示します。

表 9-7 $nI および $nS のメタキャラクタの動作変更を制御する整数

整数	動作の説明
0	値が使用不可である場合に失敗します (デフォルト)。
1	ある値が使用可能である場合に値を挿入します。それ以外の場合は何も挿入しません。
2	ある値が使用可能である場合に値を挿入します。それ以外の場合は何も挿入せず、前の文字を削除します (この特殊な動作は、ims-ms チャネルによって必要とされる)。
3	ある値が使用可能である場合に値を挿入します。それ以外の場合は何も挿入せず、後続の文字を無視します。

メタキャラクタに加えて、表 9-8 で示すように、2 つの特殊なテンプレート文字列があります。

表 9-8 特殊なテンプレート文字列

特殊なテンプレート文字列	説明
*	グループの拡張を実行します。グループの拡張を実行するこの値はユーザーエントリに対しては無効です。
**	LDAP_FORWARDING_ADDRESS MTA オプションによって指定されている属性を拡張します。これによってデフォルトの mailForwardingAddress. になります。

たとえば、グループ拡張の場合、ユーザーの mailDeliveryOption 値が mailbox に設定されていると、UID、パーセント記号 (適用可能な場合はこのあとにホストしているドメインが続く)、プラス記号 (指定されている場合はこのあとにサブアドレスが続く)、および @ims-ms-daemon で構成される新規アドレスが作成されます。

配信オプションのデフォルト

この時点でアクティブな配信オプションのリストが空である場合、リストの最初のオプション (通常はメールボックス) がユーザー用にアクティブ化され、リストの 2 番目のオプション (通常はメンバー) がグループ用にアクティブ化されます。

開始日と終了日のチェック

配信オプションリストが読み取られた後、開始日と終了日のチェックが実行されます。それぞれの属性名は、LDAP_START_DATE (デフォルトは vacationStartDate) および LDAP_END_DATE (デフォルトは vacationEndDate) の各 MTA オプションで制御します。1 つ以上のアクティブな配信オプションで ^ プレフィックス文字を指定した場合、これらのオプションの値は、現在の日付と照らしてチェックされます。現在の日付がこれらのオプションで指定されている範囲に含まれていない場合、プレフィックス ^ 付きの配信オプションは、アクティブなセットから削除されます。詳細は、528 ページの「不在返信メッセージの自動返信の属性」を参照してください。

Optin 属性と Presence 属性

LDAP_OPTIN MTA オプションを使用すると、スパムフィルタの Optin 値のリストを含んでいる LDAP 属性を指定できます。このオプションが指定されている場合で、かつ属性が存在する場合は、現在のスパムフィルタの optin リストに追加されます。LDAP_DOMAIN_ATTR_OPTIN MTA オプションで設定されているドメインレベルの属性によって設定されている値もリストに追加されます。

LDAP_PRESENCE MTA オプションを使用すると、ユーザーの存在情報を返す解決可能な URL を指定できます。このオプションが指定されている場合で、かつ属性が存在する場合、その値は Sieve による存在テストに関連して使用できるように保存されます。LDAP_DOMAIN_ATTR_PRESENCE MTA オプションによって設定されるドメインレベル属性は、ユーザーエントリの値がない場合に、この URL のソースとして使用されます。

Sieve フィルタの処理

次に、このエントリに適用される Sieve フィルタがあるかどうかについて mailSieveRuleSource 属性がチェックされます。この属性が存在する場合、属性はこの時点でパースされ、保存されます。この属性の値としては、完全な Sieve スクリプトが含まれている単一の値または各値に 1 個の Sieve スクリプトが含まれている複数の値の 2 つの形式が可能です。後者の形式は、Web フィルタ作成インタフェースによって作成されます。それぞれの値を順番に並べて適切につなげるための特別なコードが使用されます。

mailSieveRuleSource 属性の使用は、LDAP_FILTER MTA オプションで変更できます。

据え置き処理の制御

次に、mailDeferProcessing 属性がチェックされます。この属性は、LDAP_REPROCESS MTA オプションで変更できます。この属性が存在し、no に設定されている場合、処理は通常どおりに続行されます。属性が yes に設定されていて、現在のソースチャネルが再処理チャネルではない場合、このエントリの拡張は異常終了し、元の *user@domain* アドレスが再処理チャネルのキューに入れられます。この属性が存在しない場合、配信オプションの処理に関連付けられている据え置き処理の文字プレフィックスの設定がチェックされます (例については、「配信オプションの処理」を参照)。文字プレフィックスが設定されている場合、処理は据え置きとなります。設定されていない場合、ユーザーのデフォルトは no です。グループのデフォルトは、MTA オプション DEFER_GROUP_PROCESSING で制御されます。このオプションのデフォルトは 1 (yes) です。この時点で、ユーザーエントリのエイリアス処理は終了します。

グループ拡張属性

その他のいくつかの属性はグループ拡張に関連付けられており、この時点で処理される必要があります。これらの属性の名前はすべて、さまざまな MTA オプションで設定可能です。

表 9-9 に、デフォルトの属性名、属性名を設定する MTA オプション、および MTA による属性の処理方法を示します。この表での要素の順序は、各グループ属性が処理される順序を示しています。正しく動作するには、この順序が不可欠です。

表 9-9　グループ拡張のデフォルト属性および設定用 MTA オプション

デフォルトの属性	(属性名を設定する MTA オプション) 属性の処理方法
mgrpMsgRejectAction	(LDAP_REJECT_ACTION) 後続のアクセスチェックのいずれかが失敗した場合の処理を制御する単一値の属性。TOMODERATOR の 1 つの値だけが定義されており、設定すると、mgrpModerator 属性で指定したモデレータにアクセスのエラーをすべてリダイレクトするよう、MTA に指示します。デフォルト (およびこの属性のほかの値すべて) では、エラーを報告し、メッセージは拒否されます。
mailRejectText	(LDAP_REJECT_TEXT) この属性の最初の値に格納された最初の行が保存されます。後続の認証属性のいずれかが原因でメッセージが拒否された場合、このテキストが返されます。テキストは SMTP 応答に表示されるため、現在のメッセージング規格に準拠するには、値は US-ASCII に制限する必要があります。
mgrpBroadcasterPolicy	(LDAP_AUTH_POLICY) グループへの送信を行うために必要な認証のレベルを指定します。可能なトークンは SMTP_AUTH_REQUIRED または AUTH_REQ であり、どちらも、グループへの送信を行う場合に差出人を特定するために SMTP AUTH コマンドを使用する必要があることを意味します。また、PASSWORD_REQUIRED、PASSWD_REQUIRED、または PASSWD_REQ も可能なトークンであり、これらはどれも、グループへの送信を行うために、mgrpAuthPassword 属性で指定されているパスワードがメッセージの Approved: ヘッダーフィールドに存在する必要があることを意味します。OR は、このリストの OR_CLAUSES MTA オプションの設定を 1 に変更します。AND は、このリストの OR_CLAUSES MTA オプションの設定を 0 に変更します。NO_REQUIREMENTS は演算を行いません。複数の値を指定でき、各値を、コンマ区切りのトークンのリストにすることも可能です。 SMTP AUTH が呼び出された場合は、後続の認証チェックが、MAIL FROM アドレスではなく、SASL レイヤーによって提供された電子メールアドレスに照らして実行されることも意味します。

表 9-9　グループ拡張のデフォルト属性および設定用 MTA オプション (続き)

デフォルトの属性	(属性名を設定する MTA オプション) 属性の処理方法
mgrpAllowedDomain	(LDAP_AUTH_DOMAIN) このグループへのメッセージの送信を許可されたドメイン。OR_CLAUSES MTA オプションが 0 (デフォルト) に設定されているとき、一致がない場合はアクセスチェックが失敗したことを意味し、後続のテストはすべて省略されます。OR_CLAUSES MTA オプションが 1 に設定されているとき、一致がない場合は「failure pending」フラグが設定されます。アクセスチェックが成功するためには、ほかのいくつかのアクセスチェックが成功する必要があります。送信者が LDAP_AUTH_URL と一致することがすでに確認されている場合、このチェックは省略されます。複数の値を指定でき、グローバルな形式のワイルドカードも使用できます。
mgrpDisallowedDomain	(LDAP_CANT_DOMAIN) このグループへのメッセージの送信を許可されていないドメイン。一致がある場合は、アクセスチェックが失敗したことを意味し、後続のテストはすべて省略されます。送信者が LDAP_AUTH_URL と一致することがすでに確認されている場合、このチェックは省略されます。複数の値を指定でき、グローバルな形式のワイルドカードも使用できます。
mgrpAllowedBroadcaster	(LDAP_AUTH_URL) このグループへのメッセージの送信を許可されているメールアドレスを特定する URL。複数の値を指定できます。各 URL はアドレスのリストに拡張され、各アドレスは現在のエンベロープ from アドレスに照らしてチェックされます。OR_CLAUSES MTA オプションが 0 (デフォルト) に設定されているとき、一致がない場合はアクセスチェックが失敗したことを意味し、後続のテストはすべて省略されます。OR_CLAUSES MTA オプションが 1 に設定されているとき、一致がない場合は「failure pending」フラグが設定されます。アクセスチェックが成功するためには、ほかのいくつかのアクセスチェックが成功する必要があります。一致がある場合は、後続のドメインアクセスチェックも省略されます。実行される展開は、すべてのアクセス制御チェックを無効にした場合の SMTP EXPN に似ています。
mgrpDisallowedBroadcaster	(LDAP_CANT_URL) このグループへのメッセージの送信を許可されていないメールアドレスを特定する URL。複数の値を指定できます。各 URL はアドレスのリストに拡張され、各アドレスは現在のエンベロープ from アドレスに照らしてチェックされます。一致がある場合は、アクセスチェックが失敗したことを意味し、後続のテストはすべて省略されます。実行される展開は、すべてのアクセス制御チェックを無効にした場合の SMTP EXPN に似ています。

表 9-9　グループ拡張のデフォルト属性および設定用 MTA オプション (続き)

デフォルトの属性	(属性名を設定する MTA オプション) 属性の処理方法
mgrpMsgMaxSize	(LDAP_ATTR_MAXIMUM_MESSAGE_SIZE) グループへ送信できる最大のメッセージサイズ (バイト数)。この属性は廃止されましたが、下位互換性を保つためにサポートされています。代わりに新しい mailMsgMaxBlocks を使用する必要があります。
mgrpAuthPassword	(LDAP_AUTH_PASSWORD) リストに投稿するために必要なパスワードを指定します。mgrpAuthPassword 属性が存在することによって、再処理は通過します。メッセージが再処理チャネルのキューに入れられると、ヘッダーからパスワードが取得され、エンベロープに配置されます。その後、再処理中に、パスワードはエンベロープから取得され、この属性に照らしてチェックされます。また、実際に使用されているパスワードのみがヘッダーフィールドから削除されます。 OR_CLAUSES MTA オプションは、この属性に対しても、ほかのアクセスチェック属性に対する場合と同様に機能します。
mgrpModerator	(LDAP_MODERATOR_URL) この属性によって指定される URL のリスト。一連のアドレスに拡張されます。このアドレスリストの解釈は、LDAP_REJECT_ACTION MTA オプションの設定によって異なります。LDAP_REJECT_ACTION が TOMODERATOR に設定されている場合、この属性によって、アクセスチェックのいずれかが失敗した場合のメッセージ送信先となるモデレータのアドレスが指定されます。LDAP_REJECT_ACTION が設定されていない場合、または別の値が設定されている場合は、アドレスリストはエンベロープ from アドレスと比較されます。一致が存在する場合、処理は続行されます。一致が存在しない場合、メッセージはこの属性で指定されているすべてのアドレスに再送信されます。この属性の拡張は、この属性の値をグループの URL リストにすることによって実装されます。RFC822 アドレスまたはグループに関連付けられた DN のリストはすべて消去され、グループ用の配信オプションは、members に設定されます。また、この表にリストされている後続のグループ属性は無視されます。
mgrpDeliverTo	(LDAP_GROUP_URL1) URL のリストであり、展開すると、メーリングリストのメンバーのアドレスが一覧表示されます。
memberURL	(LDAP_GROUP_URL2) URL のリストであり、展開すると、メーリングリストのメンバーのアドレスが一覧表示されます。

表 9-9　グループ拡張のデフォルト属性および設定用 MTA オプション (続き)

デフォルトの属性	(属性名を設定する MTA オプション) 属性の処理方法
uniqueMember	(LDAP_GROUP_DN) グループメンバーの DN のリスト。DN はサブツリー全体を示す場合があります。一意のメンバー DN は、LADP URL に埋め込むことによって拡張されます。使用する URL は、GROUP_DN_TEMPLATE MTA オプションで正確に指定します。このオプションのデフォルト値は、次のとおりです。 ldap:///$A?mail?sub?(mail=*) $A は、uniqueMember DN の挿入点を指定しています。
mgrpRFC822MailMember	(LDAP_GROUP_RFC822) このリストのメンバーのメールアドレス。
rfc822MailMember	(LDAP_GROUP_RFC822) rfc822MailMember は下位互換性のためにサポートされています。任意の指定グループで rfc822MailMember または mgrpRFC822MailMember のいずれかを使用できますが、両方同時には使用できません。
mgrpErrorsTo	(LDAP_ERRORS_TO) エンベロープ発信元 (MAIL FROM) アドレスを、属性によって指定されている任意の値に設定します。
mgrpAddHeader	(LDAP_ADD_HEADER) 属性で指定されているヘッダーを、ヘッダートリミング ADD オプションにします。
mgrpRemoveHeader	(LDAP_REMOVE_HEADER) 指定されているヘッダーを、ヘッダートリミング MAXLINES=-1 オプションにします。
mgrpMsgPrefixText	(LDAP_PREFIX_TEXT) 指定テキストがある場合は、それをメッセージテキストの先頭に追加します。
mgrpMsgSuffixText	(LDAP_SUFFIX_TEXT) 指定テキストがある場合は、それをメッセージテキストの末尾に追加します。
No Default	(LDAP_ADD_TAG) 指定されたテキストが件名に存在するかどうかをチェックします。存在しない場合は、テキストを件名のフィールドの先頭に追加します。

次の最終的な属性は、SMTP の EXPN コマンドの一部として、特殊なグループ拡張の場合にチェックされます。mgmanMemberVisibility または expandable です。LDAP_EXPANDABLE MTA オプションを使用すると、チェック対象としてさまざまな属性を選択できます。指定可能な値は以下のとおりです。anyone (だれでもグループを拡

張できる)、all または true (ユーザーは SASL で認証されていないと、拡張が許可されない)、および none (拡張は許可されていない) です。認識不能な値は、none と解釈されます。属性が存在しない場合、EXPANDABLE_DEFAULT MTA オプションによって拡張を許可するかどうかが制御されます。

エイリアスエントリは、ドメインエントリと似た方法でキャッシュされます。エイリアスキャッシュを制御する MTA オプションは、ALIAS_ENTRY_CACHE_SIZE (デフォルト 1000 エントリ) および ALIAS_ENTRY_CACHE_TIMEOUT (デフォルト 600 秒) です。このエイリアス用に LDAP から返される値は、キャッシュに保管されます。

エイリアスエントリのネガティブキャッシングは、ALIAS_ENTRY_CACHE_NEGATIVE MTA オプションで制御します。ゼロ以外の値の場合、エイリアス一致エラーのキャッシュが有効になります。値がゼロの場合は無効になります。デフォルトでは、エイリアスエントリのネガティブキャッシングは無効になっています。無効なアドレスが繰り返し指定されることは、実際は頻繁には起こり得ないという理論です。また、ネガティブキャッシングが実行されることによって、ディレクトリに追加された新規ユーザーをタイムリーに認識できなくなる場合があります。ただし、バニティドメインが多用されている状況では、サイトはエイリアスのネガティブキャッシングを有効にすることを検討する必要があります。ALIAS_URL0 で指定されている URL によって実行される検索は、成功する可能性が低くなります。

アドレスリバース

ダイレクト LDAP を使用してアドレスリバースを実行するには、まず、USE_REVERSE_DATABASE の値を 4 に設定します。これによってリバースデータベースの使用が無効になります。その後、前述したルーティング機能を使用します。以前のバージョンでは、次の形式のリバース URL の指定からアドレスリバースが開始されました。

```
REVERSE_URL=ldap:///$V?mail?sub?$Q
```

$V メタキャラクタについては、すでにエイリアス URL の関連で説明したとおりです。ただし、$Q メタキャラクタは、エイリアス URL で使用される $R メタキャラクタと非常によく似ていますが、アドレスリバース専用に使用されます。$R とは異なり、$Q では、アドレスリバースの候補であるアドレスを含んでいる属性を検索するフィルタが生成されます。検索対象になる属性のリストは、MTA オプション LDAP_MAIL_REVERSES で指定します。このオプションが設定されていない場合は、local.imta.schematag configutil パラメータが調べられ、その値に応じて適切なデフォルト属性の集合が選択されます。

表 9-10 に、local.imta.schematag の値と選択されるデフォルト属性を示します。

表 9-10 local.imta.schematag の値と属性

スキーマタグ値	属性
sims40	mail,rfc822mailalias
nms41	mail,mailAlternateAddress
ims50	mail,mailAlternateAddress

ただし、$Q の使用は、現在は不適切になっています。メッセージの取得やその他の機能を正しく実行するために、アドレスリバースの機能は向上されており、一致があるという事実に加えて、一致した属性に注意を払うようになっています。つまり、$Q の代わりに $R を使用してフィルタを指定する必要があります。また、$N メタキャラクタが追加されていますが、これはアドレスリバース対象の属性のリストを返します。結果のオプション値は、次のとおりです。

```
REVERSE_URL=ldap:///$V?$N?sub?$R
```

local.imta.schematag はコンマ区切りのリストにできます。複数のスキーマがサポートされている場合は、組み合わせて重複を削除した属性のリストが使用されます。

また、フィルタは、最初に指定されたアドレスを検索するだけでなく、同じローカル部分を持ちながらもドメイン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。このドメインは 202 ページの手順 2 で保存されたものです。ドメインツリー検索の反復性は、この 2 つのアドレスが異なる可能性があることを意味します。

たとえば、ドメイン siroe.com がドメインツリーに存在し、MTA によって次のアドレスが認識されたと仮定します。

```
u@host1.siroe.com
```

$R および ims50 schematag の展開の結果から得られるフィルタは、次のようになります。

```
(|(mail=u@siroe.com)
(mail=u@host1.siroe.com)
(mailAlternateAddress=u@siroe.com)
(mailAlternateAddress=u@host1.siroe.com)
(mailEquivalentAddress=u@siroe.com)
(mailEquivalentAddress=u@host1.siroe.com))
```

リバース URL によって、正規化されたアドレスを含んでいる属性が明示的に指定されています。これは通常、メール属性です。

URL が構築された後、LDAP 検索が実行されます。検索が成功した場合、最初に返された属性値によって元のアドレスが置き換えられます。検索が失敗した場合、またはエラーが発生した場合は、元のアドレスは変更されません。

アドレスリバース処理が実行される頻度、特にメッセージヘッダーに表示されるアドレスの数および必要なディレクトリ照会による負担を考慮すると、否定的な結果と肯定的な結果の両方をキャッシュする必要があります。これは、開鎖型の、動的に拡張されたメモリ内ハッシュテーブルを使用して実装します。キャッシュの最大サイズは REVERSE_ADDRESS_CACHE_SIZE MTA オプションで設定します (デフォルトは 100000)。キャッシュ内のエントリのタイムアウトは REVERSE_ADDRESS_CACHE_TIMEOUT MTA オプションで設定します (デフォルトは 600 秒)。実際は、キャッシュにはアドレス自体が保存され、LDAP URL や LDAP 結果は保存されません。

非同期 LDAP 動作

非同期検索では、パフォーマンス問題の原因となる可能性のある大きな LDAP 結果全体をメモリ内に保存する必要がありません。MTA では、さまざまなタイプの検索を非同期で実行するための機能が提供されます。

非同期 LDAP 検索の使用は、MTA オプション LDAP_USE_ASYNC で制御します。このオプションはビットエンコードされた値です。各ビットは、設定されている場合、MTA 内の特定の LDAP の使用と連動して、非同期 LDAP 検索の使用を有効にします。

表 9-11 に、option.dat ファイルの LDAP_USE_ASYNC MTA オプションに設定するビットと値を示します。

表 9-11 LDAP_USE_ASYNC MTA オプションの設定

ビット	値	LDAP の具体的な使用法
0	1	LDAP_GROUP_URL1 (mgrpDeliverTo) URL
1	2	LDAP_GROUP_URL2 (memberURL) URL
2	4	LDAP_GROUP_DN (UniqueMember) DN
3	8	auth_list、moderator_list、sasl_auth_list、および sasl_moderator_list の非定位置リストパラメータ URL
4	16	cant_list、sasl_cant_list 非定位置リストパラメータ URL
5	32	originator_reply 非定位置リストパラメータ URL
6	64	deferred_list、direct_list、hold_list、nohold_list 非定位置リストパラメータ URL

表 9-11　LDAP_USE_ASYNC MTA オプションの設定 (続き)

ビット	値	LDAP の具体的な使用法
7	128	username_auth_list、username_moderator_list、username_cant_list 非定位置リストパラメータ URL
8	256	エイリアスファイルリストの URL
9	512	エイリアスデータベースリストの URL
10	1024	LDAP_CANT_URL (mgrpDisallowedBroadcaster) 外部レベル URL
11	2048	LDAP_CANT_URL 内部レベル URL
12	4096	LDAP_AUTH_URL (mgrpAllowedBroadcaster) 外部レベル URL
13	8192	LDAP_AUTH_URL 内部レベル URL
14	16384	LDAP_MODERATOR_URL (mgrpModerator) URL

LDAP_USE_ASYNC MTA オプションのデフォルトは 0 です。つまり、非同期 LDAP 検索はデフォルトでは無効です。

設定のまとめ

ダイレクト LDAP を有効にするには、次の MTA オプションを設定する必要があります。

```
ALIAS_MAGIC=8764
ALIAS_URL0=ldap:///$V?*?sub?$R
USE_REVERSE_DATABASE=4
USE_DOMAIN_DATABASE=0
REVERSE_URL=ldap:///$V?mail?sub?$Q
```

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を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)
```

これらのオプションのうち最後のものは、ホストドメインとバニティドメインの両方で、ローカル部分にワイルドカードが指定されているケースも処理することに注意してください。ワイルドカードが指定されたローカル部分のサポートが必要であり、バニティドメインのサポートが不要な場合は、次のオプションを代わりに使用してください。

```
ALIAS_URL1=ldap:///$V?*?sub?&(mailAlternateAddress=@$D)
```

filter ssrd:$A 句は、MTA 設定ファイル (imta.cnf) 内の ims-ms チャネル定義から削除する必要があります。

第 10 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 設定をコンパイルする

`imta.cnf`、`mappings`、`aliases`、`option.dat` などの MTA 設定ファイルを変更した場合は、`imsimta refresh` コマンドを使って必ず設定をコンパイルしなおす必要があります (『Sun Java System Messaging Server Administration Reference』を参照)。このコマンドによって、設定ファイルが共有メモリ内の単一のイメージ (UNIX の場合)、またはダイナミックリンクライブラリ (NT の場合) にコンパイルされます。

コンパイルされた設定には、静的な部分と動的で再読み込み可能な部分があります。動的な部分が変更された場合に `imsimta reload` を実行すると、実行中のプログラムによって動的なデータが再読み込みされます。動的な部分とは、マッピングテーブル、エイリアス、検索テーブルです。

設定情報のコンパイルは、主にパフォーマンス向上のために行います。コンパイルされた設定を使用するもう 1 つの利点は、設定の変更を簡単にテストできることです。これは、コンパイルされた設定が使用されているときに設定ファイル自体は「実行中」ではないからです。

チャネルプログラムなどの MTA コンポーネントは、設定ファイルの読み込みが必要になるたびに、コンパイルされた設定が存在するかどうかをチェックします。存在する場合は、そのイメージが実行中のプログラムに添付されます。イメージの添付処理に失敗すると、MTA は代わりに古い方法であるテキストファイルの読み込みを実行します。

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 *msg_svr_base*`/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ルールが含まれています。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。通常、デフォルトの `imta.cnf` を使用することでシステムは良好に機能します。

この節では、MTA 設定ファイルについて簡単に説明します。MTA 設定ファイルを構成する書き換えルールとチャネル定義の詳細については、第 11 章「書き換えルールの設定」および第 12 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、書き換えルール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 (書き換えルール) 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、ドメイン書き換えルールとチャネル定義の 2 つの部分から構成されます。ドメイン書き換えルールがファイルの最初に現れ、チャネル定義とは 1 つの空白行で区切られています。チャネル定義は集合的にチャネルテーブルと呼ばれます。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、書き換えルール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。書き換えルール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。

```
! test.cnf - 設定ファイルの例。(1)
!
! これは、単に設定ファイルの例です。
! システムで使用するためのものではありません。
!
! パート I: 書き換えルール
a      $U@a-daemon (2)
b      $U@b-daemon
c      $U%c@b-daemon
d      $U%d@a-daemon
             (3)
! パート II: チャネル定義
l               (4)
local-host

a_channel defragment charset7 usascii (5)
a-daemon

b_channel noreverse notices 1 2 3
b-daemon

</opt/SUNWmsgsr/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. 書き換えルールは設定ファイルの前半部分にあります。書き換えルール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、書き換えルール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。その後、チャネルブロック間が空白行で区切られます (例外は defaults チャネルであり、このチャネルは l チャネルの前に出現)。

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. 設定ファイルには、ほかのファイルの内容をインクルードすることができます。行の 1 桁目に「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。インクルードファイルは、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 10-1 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 10-1 アドレスおよび関連チャネル

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、193 ページの「書き換えルール」、196 ページの「チャネル定義」、および第 11 章「書き換えルールの設定」を参照してください。

注 imta.cnf ファイルを変更した場合は、必ず MTA 設定をコンパイルしなおしてください。233 ページの「MTA 設定をコンパイルする」を参照してください。

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 `mappings` 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。

マッピングテーブルは、MTA `mappings` ファイルに保存されています。これは、MTA `tailor` 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 *msg_svr_base*`/config/mappings` です。`mappings` ファイルの内容は、再読み込み可能なセクションとしてコンパイルされた設定に取り込まれます (233 ページの「MTA 設定をコンパイルする」を参照)。`mappings` ファイルは、誰でも読み取り可能でなければなりません。誰でも読み取り可能でアクセスできない場合は、誤作動をまねくことになります。`mappings` ファイルを変更した場合は、必ず MTA 設定をコンパイルしなおしてください。233 ページの「MTA 設定をコンパイルする」を参照してください。

表 10-2 に、このマニュアルで説明するマッピングテーブルの一覧を示します。

表 10-2　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
`AUTH_REWRITE`	367 ページ	authrewrite キーワードとともに、認証操作 (SASL) で取得したアドレス情報によってヘッダーとエンベロープアドレスを変更するために使用されます。
`CHARSET-CONVERSION`	448 ページ	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。

表 10-2　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
COMMENT_STRINGS	390 ページ	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ます。
CONVERSIONS	430 ページ	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ます。
FORWARD	271 ページ	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います。
FROM_ACCESS	533 ページ	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します。このテーブルは、To アドレスが不適切な場合に使用します。
INTERNAL_IP	550 ページ	内部のシステムとサブネットを認識します。
MAIL_ACCESS	533 ページ	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着信接続をブロックする場合に使用します。
NOTIFICATION_LANGUAGE	275 ページ	通知メッセージをカスタマイズまたはローカライズします。
ORIG_MAIL_ACCESS	533 ページ	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着信接続をブロックする場合に使用します
ORIG_SEND_ACCESS	533 ページ	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、着信接続をブロックします。
PERSONAL_NAMES	391 ページ	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ます。
PORT_ACCESS	533 ページ	IP 番号に基づいて着信接続をブロックする場合に使用します。
REVERSE	267 ページ	内部形式から公のアドバタイズ形式にアドレスを変換します。

表 10-2　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
SEND_ACCESS	533 ページ	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、着信接続をブロックします。
*SMS_Channel*_TEXT	918 ページ	サイト定義のテキストの変換に使用されます。
X-ATT-NAMES	439 ページ	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ます。
X-REWRITE-SMS-ADDRESS	917 ページ	ローカル SMS アドレスの妥当性チェックに使用されます。

マッピングファイルのファイルフォーマット

mappings 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ます。名前には常に、最初の列にアルファベット文字がきます。テーブル名の次には必ず空白行が続き、その後にテーブルのエントリが続きます。エントリは、ゼロまたはそれ以上のインデント行で構成されます。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。各テーブル名の後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。コメントは、1 つめのカラムに記述され、感嘆符 (!) から始まります。

つまり、ファイルフォーマット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、最高 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、大きな CPU 容量およびメモリ容量を要することになる)。252 バイト以上の長い行は、¥ (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 1 つめのカラムの前にある空白スペースを削除してはなりません。

mappings 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

mappings ファイルにほかのファイルをインクルードすることができます。次の形式の行を使用します。

```
<file-spec
```

これによって、mappings 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 (ディレクトリ等) が必要です。この方法で含めるファイルは、誰でも読み取り可能でなければなりません。mappings 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。インクルードファイルは、3 階層までネストできます。インクルードファイルは、mappings ファイルといっしょに読み込まれます。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

mappings 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 (*) はゼロまたはそれ以上の文字と一致し、パーセント記号 (%) は 1 つの文字に一致します。ドル記号 ($) をアスタリスク、パーセント記号、スペース、およびタブの前に置くことによって、それらの記号を文字として使用できるようになります。アスタリスクまたはパーセント記号を文字として使用した場合は、それらの特殊な定義が無効になります。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、1 つめのドル記号によって、2 つめのドル記号を文字として認識されるようになります。

表 10-3 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 つの文字に一致します。
*	左から右への最大限の一致を使用して、ゼロ以上の文字を一致します
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します。
修飾子	**説明**
$_	左から右への最低限の一致を使用します。
$@	後続のワイルドカード、またはグロブの「保存」をオフにします。
$^	後続のワイルドカードまたはグロブの「保存」をオンにします。デフォルト設定。
グロブワイルドカード	**説明**
$A%	A ～ Z および a ～ z のアルファベットのうち、1 つの文字に一致します。
$A*	A ～ Z および a ～ z のアルファベットが 0 個以上含まれた文字列に一致します。
$B%	1 桁の 2 進数 (0 または 1) に一致します。
$B*	0 またはそれ以上の桁数の 2 進数 (0 または 1) に一致します。
$D%	1 桁の 10 進数 (0 ～ 9) に一致します。
$D*	0 またはそれ以上の桁数の 10 進数 (0 ～ 9) に一致します。
$H%	1 桁の 16 進数 (0 ～ 9 または A ～ F) に一致します。
$H*	0 またはそれ以上の桁数の 16 進数 (0 ～ 9 または A ～ F) に一致します。

表 10-3 マッピングパターンのワイルドカード (続き)

$O%	1 桁の 8 進数 (0 ～ 7) に一致します。
$O*	0 またはそれ以上の桁数の 8 進数 (0 ～ 7) に一致します。
$S%	1 つの記号セット文字 (例 : 0 ～ 9、A ～ Z、a ～ z、_、$) に一致します。
$S*	ゼロまたはそれ以上の記号セット文字、すなわち 0 ～ 9、A ～ Z、a ～ z、_、$ に一致します。
$T%	1 つのタブ、垂直タブ、またはスペース文字に一致します。
$T*	ゼロまたはそれ以上のタブ、垂直タブ、またはスペース文字に一致します。
$X%	$H% と同義。
$X*	$H* と同義。
$[c]%	文字 c に一致します。
$[c]*	文字 c の不定発生に一致します。
$[c_1 c_2 ... c_n]%	文字 c_1、c_2、または c_n の発生の 1 つに一致します。
$[c_1 c_2 ... c_n]*	文字 c_1、c_2、または c_n の不定発生に一致します。
$[c_1 -c_n]%	c_1 から c_n までの文字のいずれか 1 つに一致します。
$[c_1 -c_n]*	c_1 から c_n までの文字の不定発生に一致します。
$< IPv4>	ビットを無視して、IPv4 アドレスに一致します。
$(IPv4)	プレフィックスビットを維持した状態で、IPv4 アドレスに一致します。
${IPv6}	1 組の IPv6 アドレスに一致します。

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、構文にドル記号を 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、文字列 a/b/c をパターン $_*/$_* と比較した場合、左の $_* は a と、右の $_* は b/c と一致します。

IP の照合

IPv4 プレフィックスの照合では、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プレフィックス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スラッシュを付け、照合を確認する際に無視する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。

ドル記号に数字 *n* が続いている場合は置換を呼び出します。ドル記号の後ろにアルファベット文字が続くものは「メタキャラクタ」と呼ばれます。メタキャラクタ自体はテンプレートで生成された出力文字列に出現しませんが、特殊な置換や処理で使われます。特殊な置換および標準処理のメタキャラクタの一覧は、表 10-4 を参照してください。その他のメタキャラクタ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 のいずれかのメタキャラクタがある場合、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 $R のどのメタキャラクタも含まれていない場合は、$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 10-4　マッピングテンプレートの置換とメタキャラクタ

置換シーケンス	置き換える内容
$n	左から右にゼロから数えて *n* 番めのワイルドカードのフィールド。
$#...#	シーケンス番号の置換
$]...[	LDAP により URL 検索が行われます。結果として、置換が行われます。
$|...|	指定されたマッピングテーブルを、与えられた文字列に適用します。
${...}	一般データベースの置換。
$}*domain*,*attribute*{	ドメイン単位の属性にアクセスする機能を追加します。*domain* は該当するドメインであり、*attribute* はドメインに関連付けられた属性です。このドメインが存在して属性を有している場合、その初期値はマッピングの結果に代入されます。属性かドメインのどちらかが存在しない場合、マッピングエントリは失敗します。 *attribute* には、ドメイン LDAP 属性か、下記のように定義された特殊な属性を指定できます。 _base_dn_ - ドメインのユーザーエントリのベース DN _domain_dn_ - ドメインエントリ自体の DN _domain_name_ - ドメインの名前 (エイリアスではない) _canonical_name_ - ドメインに関連付けられた標準名
$[...]	サイト提供のルーチンを起動し、結果の置換を行います。

表 10-4　マッピングテンプレートの置換とメタキャラクタ (続き)

置換シーケンス	置き換える内容
メタキャラクタ	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します。
$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します。
$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します。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します。後続の照合エントリにメタキャラクタ $C、$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	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します。
$nA	現在のアドレスの 0 の位置から左に n 番目の文字を挿入します。n を省略した場合、アドレス全体が挿入されます。
$nX	メールホストの 0 から左に n 番目のコンポーネントを挿入します。n を省略した場合、メールホスト全体が挿入されます。
$?x?	マッピングエントリが x パーセントの割合で成功します。
$¥	後続のテキストを小文字にします。
$^	後続のテキストを大文字にします。
$_	後続のテキストを元々の状態で残します。
$=	後続の置換文字が、LDAP 検索フィルタへの挿入に適した引用の対象となるようにして、その部分を大文字に変換します。
$:*x*	指定したフラグが設定されている場合にのみ、一致します。
$;*x*	指定したフラグがクリアの場合にのみ、一致します。

ワイルドカードフィールドの置換 ($n)

ドル記号に数字 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で置き換えられます。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 にも一致するので、出力文字列として USER@1234.psi.siroe.com が生成されます。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 ($¥、$^、$_)

メタキャラクタ $¥ は後続のテキストを小文字に変換し、メタキャラクタ $^ は後続のテキストを大文字に変換します。また、メタキャラクタ $_ は、後続のテキストを元の大文字または小文字の状態で残します。たとえば、これらのメタキャラクタは、マッピングを使って大文字または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)

メタキャラクタ $C、$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。これらのメタキャラクタには、次の効果があります。

- $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キャラクタ $C、$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。一般データベースの置換やランダム値で制御されるエントリなど、「成功」または「失敗」するエントリに $C、$L、または $R のフラグを設定するには、メタキャラクタ $C、$L、または $R をエントリの成功または失敗する部分の左側に配置します。これを行わないと、エントリの残りの部分が失敗した場合、フラグが表示されません。

特殊なフラグの確認

マッピングプローブの中には、特殊なフラグセットを持つものがあります。これらは設定可能なフラグであり、それらが存在するかどうかは $: および $; テストの一般的なマッピングテーブル機能を使用して確認されます。$:x はフラグ x が設定されている場合のみ、エントリを一致させます。$:x はフラグ x がクリアの場合にのみ、エントリを一致させます。特定のマッピングテーブルに適用される特殊なフラグについては、各マッピングテーブルの説明を参照してください (534 ページの表 17-2 の $A、$T、$S、$F、および $D を参照)。

フラグチェックが成功するとエントリが成功して終了するが、フラグチェックが失敗するとマッピング処理を続行する必要があるという場合、エントリはフラグチェックの左側に $C メタキャラクタを配置し、フラグチェックの右側に $E フラグを配置する必要があります。

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メタキャラクタ $?x? 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。残りの割合でエントリは「失敗」し、マッピングエントリの入力文字列は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、必ずしも同義ではない)。*x* には、成功率を実数で指定します。

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)。エントリが失敗する残りの 75 パーセントの割合では、そのエントリの最初の $C によって MTA は次のエントリからマッピングを続行しますが、接続試行は拒否され、Try again later（あとでもう一度試行してください）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。

以下のいずれかの構文を使用できます。

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

```
$#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の完全なファイル指定です。オプションの引数 *radix* で出力するシーケンス値の基数を、*width* で出力する桁数を指定します。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。

オプションの引数 *m* はモジュラスです。この 4 番目の引数が指定されている場合、挿入される値はファイル mod *m* から取得されたシーケンス番号です。デフォルトでは、モジュラスの処理を行わないようになっています。

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、誰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 ユーザーアカウント (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、特殊な方法で処理されます。*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識別名で、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 mappings 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への入力文字列として *argument* を使用します。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キャラクタを使用する場合は、処理制御メタキャラクタ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、処理制御メタキャラクタが処理されません。

一般検索テーブルまたはデータベース置換 (${...})

${*text*} 形式の置換は、特殊な方法で処理されます。*text* 部分は、一般検索テーブルや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。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* がテーブルで一致すると、テーブル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* がテーブル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。

一般検索テーブルを使用している場合、MTA オプションの `use_text_databases` の下位ビットを設定する必要があります。つまり、奇数に設定する必要があります。`general.txt` を変更した場合は、`imsimta cnbuild` を使用してコンパイルし、`imsimta reload` を使用して再読み込み可能なデータを再読み込みすることで、MTA 設定にコンパイルする必要があります。

一般データベースを使用している場合、データベースが適切に動作するためには、データベースは誰にでも読み取り可能でなければなりません。

一般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キャラクタを使用する場合は、処理制御メタキャラクタをマッピングテーブルテンプレート内の一般テーブル置換の左側に配置します。そうしないと、一般テーブルの置換が「失敗」したときに、処理制御メタキャラクタが処理されません。

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見つけて呼び出すために使用されます。UNIX では、MTA は `dlopen` および `dlsym` を使って共有ライブラリ *image* からルーチン *routine* をダイナミックにロードし、呼び出します。そのとき、ルーチン *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キャラクタを使用する場合は、処理制御メタキャラクタ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、処理制御メタキャラクタが処理されません。

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、ロード監視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、誰でも読み取り可能でなければなりません。

UTF-8 文字列の生成

一般的なマッピングテーブル機能で、Unicode 文字の値から UTF-8 文字列を生成できます。次の形式の Unicode メタキャラクタ列があるとします。

```
$&A0A0,20,A1A1&
```

この場合、A0A0、20 および A1A1 の位置にある文字を含む UTF-8 文字列が生成されます。

その他の MTA 設定ファイル

imta.cnf ファイルのほかにも、Messaging Server には MTA サービスの設定に役立ついくつかの設定ファイルがあります。表 10-5 にファイルの一覧を示します。

imta.cnf、mappings、aliases、option.dat などの MTA 設定ファイルを変更した場合は、必ず設定をコンパイルしなおす必要があります (『Sun Java System Messaging Server Administration Reference』の imsimta refresh コマンドを参照)。

表 10-5　MTA 設定ファイル

ファイル	説明
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装します。 *msg_svr_base*/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します。 *msg_svr_base*/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します。 *msg_svr_base*/config/conversions

表 10-5　MTA 設定ファイル

ファイル	説明
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。 *msg_svr_base*/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。 /*msg_svr_base*/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します。 /*msg_svr_base*/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。 /*msg_svr_base*/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。 /*msg_svr_base*/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。 /*msg_svr_base*/config/imta_tailor
一般検索テーブル (オプション)	一般検索機能は一般データベースと同等です。再読み込み可能なコンパイル済み設定の一部です。 場所といくつかの調整パラメータを指定するファイル。 /*msg_svr_base*/config/general.txt
正引き検索テーブル (オプション)	To: アドレスの検索機能。正引きデータベースと同等。再読み込み可能なコンパイル済み設定の一部です。 /*msg_svr_base*/config/forward.txt
リバース検索テーブル (オプション)	From: アドレスのリバース検索機能。リバースデータベースと同等。再読み込み可能なコンパイル済み設定の一部 /*msg_svr_base*/config/reverse.txt

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、264 ページの「エイリアス」を参照してください。

aliases ファイルの変更後は、MTA を再起動するか、imsimta reload コマンドを実行してください。

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。チャネルオプションファイルは MTA 設定ディレクトリに保存し、*x*_option という名前を付けてください。*x* はチャネル名です。たとえば、*msg_svr_base*/config/imta/tcp_local_option のようになります。詳細については、348 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構文の詳細については、『Messaging Server Reference Manual』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。

このファイルの構文の詳細については、427 ページの「変換チャネル」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 では、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 ファイルを編集して変更することができます (概念の詳細は、191 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フォーマットは、ほかの MTA 設定ファイルのフォーマットに似ています。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 はオプション名で、*value* はオプションを設定する文字列または整数です。*option* が整数の値を受け入れる場合は、*b%v* の文字列表記ルールを使って基数を指定することができます。この場合、*b* は底 10 で表す基数であり、*v* は底 *b* で表す実際の値です。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。

```
! オプションの最初のセットは、[SERVICE=xxx] ヘッダーなし
! で表示された、すべてのサービスに適用されるデフォルトオプション
! ! です。
!
MIN_PROCS=0
MAX_PROCS=5
MIN_CONNS=5
MAX_CONNS=20
MAX_LIFE_TIME=86400
MAX_LIFE_CONNS=100
MAX_SHUTDOWN=2
!
! ディスパッチャで使用できるサービスを定義する
!
[SERVICE=SMTP]
PORT=25
IMAGE=msg_svr_base/lib/tcp_smtp_server
LOGFILE=msg_svr_base/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、『Messaging Server Reference Manual』を参照してください。

マッピングファイル

mappings ファイルでは、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。1 つめ (左側) のカラムには入力文字列が、2 つめ (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。

mappings ファイルによって、MTA は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。mappings ファイルの構文および test -mapping コマンドの詳細については、237 ページの「マッピングファイル」および『Messaging Server Reference Manual』を参照してください。

mappings ファイルの変更後は、MTA を再起動するか、imsimta reload コマンドを実行してください。

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、許可する書き換えルールの数を設定したりできます。

option.dat では、#、!、または ; で始まる行はコメント行として処理されます。先行する行の末尾に、続きがあることを示す ¥ がある場合でも同様です。配信オプションなど、これらの文字を含む長いオプションの場合には注意が必要です。

配信オプションの場合は、自然なレイアウトは # または ! で始まる継続行になりますが、確実で整然とした回避方法はあります。

オプションファイルの構文の詳細については、『Messaging Server Reference Manual』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 *msg_svr_base*/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、『Messaging Server Reference Manual』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、199 ページの「ジョブコントローラ」、375 ページの「チャネル実行ジョブの処理プール」、および 376 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、次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。

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。一般に、チャネルマスタープログラムは、そのチャネルの MTA メッセージキューにメッセージが保存されている場合に呼び出されます。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は使用されません。これは、すべての SMTP サーバーからの要求に応じて、ネットワークサービスである SMTP サーバーが着信 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 10-1 に、ジョブコントローラ設定ファイルの例を示します。表 10-6 に、使用可能なオプションを示します。

コード例 10-1 ジョブコントローラ設定ファイルの例 (UNIX)

```
!MTA ジョブコントローラ設定ファイル
!
! グローバルデフォルト
tcp_port=27442          (1)
secret=never mind
slave_command=NULL      (2)
max_life_age=3600       (3)
!
!
! プールの定義
!
[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
!
! チャネル定義
!
!
[CHANNEL=l]               (6)
master_command=msg_svr_base/lib/l_master
!
[CHANNEL=ims-ms]
master_command=msg_svr_base/lib/ims_master
!
[CHANNEL=tcp_*]           (7)
anon_host=0
master_command=msg_svr_base/lib/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. このグル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
```

プールの詳細については、375 ページの「チャネル実行ジョブの処理プール」を参照してください。ジョブコントローラファイルの構文の詳細については、『Messaging Server Reference Manual』を参照してください。

表 10-6　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します。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ます。デフォルトで、ジョブコントローラはすべてのアドレスにバインドします (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られません。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がありません。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、不適切かもしれません。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」アダプタを持っている場合で、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お勧めします。
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します。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ります。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ません。メールメッセージは常にディスクに書き込まれるため、失われることはありませんが、ジョブコントローラが認識するメッセージ数がこの値の半分になるまで配信されません。この時点では、ジョブコントローラが imsimta cache -sync コマンドを模倣してプールディレクトリをスキャンします。 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プール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します。デフォルトでは 4 時間ごとにスキャンされます (ジョブコントローラが起動してから 4 時間ごと)。*time_spec* のフォーマットは、*HH*:*MM*/*hh*:*mm* または /*hh*:*mm*。*hh*.*mm* 変数は、イベントの間隔を時間数 (*h*) と分数 (*m*) で示します。*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す。たとえば 15:45/7:15 と指定すると、15:45 にイベントが開始し、その後 7 時間 15 分ごとにイベントが実行されます。

表 10-6　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
TCP_PORT=*integer*	ジョブコントローラが要求パケットをリッスンする TCP ポートを指定します。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かぎり変更しないでください。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 (*msg_svr_base*/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ります。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ます。
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します。JOB_LIMIT は各プールに個別に適用されます。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す。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ます。このオプションは、[CHANNEL] セクション内では無視されます。
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します。この値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ます。[POOL] セクション内では、このオプションが無視されます。
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します。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ます。デフォルト値が指定されていない場合は、1800 (30 分) が使用されます。
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ます。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ます。デフォルト値が指定されていない場合は 300 が使用されます。
SLAVE_COMMAND=*file_spec*	チャネルを実行し、そのチャネルに入れる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します。ほとんどの場合、MTA チャネルには SLAVE_COMMAND がありません。その場合は、予約値である NULL を指定します。この値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ます。[POOL] セクション内では、このオプションが無視されます。

エイリアス

MTA には、ローカルシステムに関連付けられ、実際のユーザーと必ずしも対応しない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。エイリアスは、メーリングリストの作成、メールの転送、およびユーザーの別名の設定に役立ちます。エイリアス解決の処理方法については、206 ページの「$V メタキャラクタ」を参照してください。

aliases ファイルまたはエイリアスデータベースで定義されている旧形式のメーリングリストは、非定位置パラメータ [capture] をとるようになりました。[capture] パラメータが使用されている場合、このパラメータで指定するのは、LDAP でユーザーまたはグループに適用される LDAP_CAPTURE 属性で指定される取得アドレスと同じセマンティクスの取得アドレスです。

エイリアスデータベース

エイリアスデータベースの使用はお勧めしません。代わりに aliases ファイルを使用してください。このファイルは imsimta reload コマンドを使用して動的に再読み込みできます。

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。

注	データベースの形式は固有のものです。データベースを直接編集しないでください。必要な変更はすべてディレクトリで行なってください。

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。よい例として、Postmaster エイリアスが挙げられます。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、このファイルの設定は無視されます。imsimta reload コマンドを発行するか MTA を再起動すると変更が有効になります。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、常に書き換えルールを使用するようにしてください。第 11 章「書き換えルールの設定」を参照してください。

このファイルでは、一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。

ファイルフォーマットは以下のとおりです。

user@domain:<*address*> （ホストしているドメイン内のユーザー用）

user@domain:<*address*> （ホストしていないドメイン内のユーザー用。例 : デフォルトドメイン）

次に例を示します。

```
! /var/mail/ ユーザー
inetmail@siroe.com:inetmail@native-daemon

! メッセージストアユーザー
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、誰でも読み込み可能であることなど)。

インクルードファイルの内容は、aliases ファイル内の参照ポイントに挿入されます。インクルード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。インクルードファイルの形式は、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、インクルードファイルにほかのファイルを含めることも可能です。インクルードファイルは、3 階層までネストすることができます。

コマンド行ユーティリティ

Messaging Server には、MTA に関する各種保守、テスト、管理などのタスクを実行するためのコマンド行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。MTA コマンド行ユーティリティの詳細については、『Messaging Server Reference Manual』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 17 章「メールのフィルタリングとアクセス制御」および第 19 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 固有のログファイルはすべて、ログディレクトリ (*msg_svr_base*/log) 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 21 章「ログの管理」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 (「リバースデータベース」とも呼ばれる) と REVERSE マッピングテーブルを使っ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、常に書き換えルールを使用するようにしてください。第 11 章「書き換えルールの設定」を参照してください。

リバースデータベースでは、各ユーザーの公式アドレスはディレクトリ内のユーザーエントリの 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。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、*msg_svr_base*/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 *msg_svr_base*/data/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、mappings 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 mappings 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し、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実行されるマッピング) の形式は、次のとおりです。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]NewAddr
esS
```

REVERSE マッピングテーブルのフラグを表 10-7 に示します。

表 10-7　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します。
$N	アドレスは変更されません。
$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します。
$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します。

表 10-7　REVERSE マッピングテーブルのフラグ (続き)

フラグ	説明
$F	パターンを正引きデータベースエントリとして追加します。
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します。
$:E	エンベロープアドレスのみを照合します。
$:F	前方を探すアドレスのみを照合します。
$:R	後方を探すアドレスのみを照合します。
$:I	メッセージ ID のみを照合します。

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、『Sun Java System Messaging Server Administration Reference』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、『Sun Java System Messaging Server Administration Reference』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、『Sun Java System Messaging Server Administration Reference』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 REVERSE マッピングの例を次に示します。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、ユーザー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ーを指定しています。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 メタキャラクタでは、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ー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@siro
e.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)。

正引き検索テーブルと FORWARD アドレスのマッピング

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。これは、エンベロープ To: アドレスは、メッセージがメールシステムで処理される過程で次々と書き換えられ、変更されるからです。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正規化機能は、エンベロープ To: アドレスには不適切です。

MTA では豊富な機能が使用でき、エンベロープ To: アドレスの置換が実行できます。エイリアスファイル、エイリアスデータベース、および一般検索テーブルによって、この機能が提供されます。

MTA では、正引き検索テーブルや FORWARD マッピングも提供されており、パターンに基づく転送、ソース固有の転送、アドレスの自動登録などの特殊な転送に使用されます。ただし、正引き検索テーブルや FORWARD マッピングは、特殊なアドレス転送のための機能であることに注意してください。ほとんどのアドレス転送には、MTA のほかの転送機構を使用したほうがパフォーマンスは向上します。

エンベロープ To: アドレス用のさまざまな置換機構では、リバース検索テーブル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等の機能は現時点ではありません。エンベロープ To: アドレス用のマッピング機能が必要とされる状況が発生することもあります。

FORWARD マッピングテーブル

FORWARD マッピングテーブルでは、パターンに基づいた転送を行うための機能が提供されます。また、ソース固有の転送を行うための機構も提供されます。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、エンベロープ To: アドレスは結果の文字列で置き換えられ、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。このほかのフラグの一覧は、表 10-8 を参照してください。

表 10-8　FORWARD マッピングテーブルフラグの各フラグの説明

フラグ	説明
$D	出力文字列を使って書き換えプロセスを再び実行します
$G	正引き検索テーブルの使用が有効になっている場合に、出力文字列を使って正引き検索テーブルをスキャンします
$H	正引き検索テーブルまたは FORWARD マッピングの検索続行を無効にします
$I	.HELD ファイルとしてメッセージを保留します
$N	アドレスは変更されません
$Y	出力文字列を新規アドレスとして使用します

FORWARD マッピングが存在する場合は、正引き検索テーブルの検索が行われる前に参照されます。FORWARD マッピングが一致し、フラグ $G が付いていれば、FORWARD マッピングの結果は正引き検索テーブルに対してチェックされます。ただし、USE_FORWARD_DATABASE が適切に設定されていて正引き検索テーブルの使用が有効になっている必要があります。チャネル固有の正引き検索テーブルの使用が指定されている場合は、正引き検索テーブルの検索が行われる前に、ソースアドレスとソースチャネルが FORWARD マッピングの結果の前に付けられます。一致する FORWARD マッピングエントリが $D を指定している場合、FORWARD マッピング (およびオプションの正引き検索テーブルの検索) の結果を使用して MTA アドレス書き換えプロセスが再び実行されます。一致する FORWARD マッピングエントリが $H を指定している場合、それ以上の FORWARD マッピングまたはデータベースの検索は、$D を使用したことによる後続のアドレス書き換えプロセスの間に実行されません。

以下に、複雑な REVERSE マッピングおよび FORWARD マッピングの使用例を示します。mr_local チャネルに関連付けられている am.sigurd.innosoft.com というシステム (仮のドメイン) が、次の一般的な形式の RFC 822 アドレスを生成すると仮定します。

```
"lastname, firstname"@am.sigurd.example.com
```

または

```
"lastname,firstname"@am.sigurd.example.com
```

これらのアドレスは完全に正しいものですが、RFC 822 の構文ルールに完全準拠していないほかのメーラー (たとえば、引用符で囲まれたアドレスを適切に処理しないメーラー) では混乱が生じることがあります。そのため、引用を必要としないアドレス形式のほうが、多くのメーラーで機能する傾向があります。次はその一例です。

```
firstname.lastname@am.sigurd.example.com
```

複雑な FORWARD マッピングおよび REVERSE マッピングの例

```
REVERSE

 *|mr_local|"*,$ *"@am.sigurd.innosoft.com $Y"$1,$ $2"@am.sigurd.innosoft.com
 *|mr_local|"*,*"@am.sigurd.innosoft.com   $Y"$1,$ $2"@am.sigurd.innosoft.com
 *|*|"*,$ *"@am.sigurd.innosoft.com        $Y$3.$2@am.sigurd.innosoft.com
 *|*|"*,*"@am.sigurd.innosoft.com          $Y$3.$2@am.sigurd.innosoft.com
 *|mr_local|*.*@am.sigurd.innosoft.com     $Y"$2,$ $1"@am.sigurd.innosoft.com
 *|*|*.*@am.sigurd.innosoft.com            $Y$2.$3@am.sigurd.innosoft.com


FORWARD

 "*,$ *"@am.sigurd.innosoft.com            $Y"$0,$ $1"@am.sigurd.innosoft.com
 "*,*"@am.sigurd.innosoft.com              $Y"$0,$ $1"@am.sigurd.innosoft.com
 *.*@am.sigurd.innosoft.com                $Y"$1,$ $0"@am.sigurd.innosoft.com
```

上記の例では、サンプルのマッピングテーブルの目的には 3 段階あります。(1) 上記の 3 種類のアドレス形式をすべて使用可能にする。(2) 必要に応じて形式を変換し、元の形式のアドレスのみを mr_local チャネルに提示する。(3) 必要に応じて形式を変換し、新しい引用符なしの形式のアドレスのみをほかのすべてのチャネルに提示する (例で示した REVERSE マッピングでは、MTA オプション USE_REVERSE_DATABASE のビット 3 が設定されていると仮定)。

正引き検索テーブル

アドレス転送を自動登録またはソース固有にする必要がある場合には、正引き検索テーブルを使用します。通常、単純なメッセージの転送に正引き検索テーブル使用することは適切ではありません。このような転送には、aliases ファイルまたはエイリアス検索テーブルを使用するほうが効率的です。デフォルトでは、正引き検索テーブルは一切使用されません。使用するには、USE_FORWARD_DATABASE オプションを使用して明示的に有効にする必要があります。正引き検索テーブルの検索は、アドレス書き換えの後、エイリアス展開が実行され、FORWARD マッピングがチェックされた後で実行されます。正引き検索テーブルの検索が成功した場合、結果の置換済みアドレスを使用して MTA アドレス書き換えプロセスが初めからやり直されます。

正引き検索テーブルに使用できる機構には、メモリ内ハッシュテーブルと従来のデータベースの 2 つがあります。テーブルのサイズが極端に大きい場合を除いて、ハッシュテーブルを使用することをお勧めします (1,000 はそれほど大きいとされません。100,000 が目安)。ハッシュテーブルは、use_text_database オプションにビット 3 (値 34) を設定すること、および use_forward_database を設定することによって有効になります。ハッシュテーブルは、*msg_svr_base*/configure/forward.txt から読み込まれ、設定の再読み込み可能な部分にコンパイルされます。imsimta relaod コマンドを使用すると、これをアクティブな MTA プロセスに再読み込みできます。

正引きデータベースは、crdb ユーティリティを使用してソーステキストファイルから作成された MTA crdb データベースです。デフォルトでは、ソーステキストファイルは次のような形式になっています。

```
user1@domain1    changedmailbox1@changeddomain1
user2@domain2    changedmailbox@changeddomain2
```

ただし、USE_FORWARD_DATABASE オプションでビット 3 が設定されていてソース固有の正引きデータベースの使用が有効になっている場合は、ソーステキストファイルの形式は次のようになります。

```
source-channel|source-address|original-address  changed-address
```

たとえば、次のようなエントリがあるとします。

```
tcp_limited|bob@blue.com|helen@red.com  "helen of
troy"@siroe.com
```

この例では To: アドレス helen@red.com が "helen of troy"@siore.com にマッピングされます (メッセージの差出人が bob@blue.com で、キューに入れられるチャネルが tcp_limted であると仮定した場合)。

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、すなわちステータス通知は、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するステータス通知を作成することもできます。

デフォルトでは、ステータス通知は、*msg_svr_base*/config/locale/C ディレクトリに保存されています (*msg_svr_base*/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

*.txt ファイルのメッセージテキストは、1 行につき 78 文字以内である必要があります。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、Messaging Server のアップグレード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。

ステータス通知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、ロケールまたはカスタマイズ、あるいはその両方のそれぞれに return_*.txt ファイルの全セットを作成し、それを別々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、もう 1 つ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、3 つ目のディレクトリには特殊な不特定多数宛メールに対する通知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、ユーザーのそれぞれのニーズに合わせて変更することができます。

日本語などの 2 バイト文字の場合は、日本語でテキストを作成してから、そのテキストを ASCII 形式に変換してから、% 文字がないかどうかをチェックしてください。不測の % 文字が存在する場合は、%% で置き換えてください。

ステータス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 以外のステータス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)。%H は、表 10-9 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 ステータスタイプです。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 10-9　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します。
%C	メッセージがキューに入っていた時間の単位[1] に展開します。
%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] に展開します。
%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] に展開します。
%S [%s]	以前展開した数値が 1 以外の場合は、S または s に展開します。例 : たとえば、「%C day%s」は、メッセージがキューに入っていた日数によって「1 day」または「2 days」などに展開できます。
%U [%u]	使用する時間の単位[1] (時間または日) に展開します。例 : たとえば、「%C %U%s」は、メッセージがキューに入っていた日数または時間数と MTA オプション RETURN_UNITS の値によって「2 日」や「1 時間」などに展開できます。RETURN_UNITS=1 (時間) を設定していて、ローカライズされた通知メッセージをサイトで使用している場合は、英語以外のすべての言語に関して、return_delayed.txt と return_timedout.txt を編集し、「日」に相当する単語を「時間」に相当する単語で置き換える必要があります。たとえば、フランス語では、jour(s) を heure(s) と置き換えます。ドイツ語では、Tag(e) を Stunde(n) と置き換えます。スペイン語では、dìa(s) を hora(s) と置き換えます
%R	メッセージの受取人のリストに展開します。
%%	% (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ます。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ます。

配信ステータス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

配信ステータス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ー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ーにフランス語の通知を返すことができます。

ステータス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。詳細は、275 ページの「ステータス通知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 (*msg_svr_base*/config/mappings) では、送信元メッセージ (通知が送信される原因であるメッセージ) の属性 (言語、国、ドメイン、アドレスなど) に応じて使用される、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パースされ、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 1 人目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-recipien
t $Idirectory-spec
```

dsn-type-list は、配信ステータス通知の種類のコンマ区切りリストです。複数の種類を指定する場合はコンマで区切ります。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、永続的配信不能を示すメッセージ (「そのようなユーザーはありません」など)。return_failed.txt ファイルが使用されます。

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

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します。メッセージは送り返されます。return_timedout.txt ファイルが使用されます。

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します。return_delayed.txt ファイルが使用されます。

deferred - 「delayed」に類似した配信不能通知。ただし、MTA が配信試行を続行する期間は表示されません。return_deferred.txt ファイルが使用されます。

forwarded - このメッセージに対して配信確認が要求されていたが、このメッセージは配信確認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します。return_forwarded.txt ファイルが使用されます。

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。標準の言語コードの値のリストは、*msg_svr_base*/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: ヘッダー行に指定した内容が入ります。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: アドレスです。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、元のメッセージの宛先のエンベロープ To: アドレス (メッセージが複数の受取人に届かない場合は 1 人目の受取人アドレス) 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、報告を受けるエンベロープ To: アドレスは dan@siroe.com 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。

コード例 10-2　　通知言語マッピングテーブルの例

```
NOTIFICATION_LANGUAGE

! 優先言語 : 指定されたヘッダー値
!
   *|*|fr|*|*     $I/lc_messages/table/notify_french/
   *|*|es|*|*     $IIMTA_TABLE/notify_spanish/
   *|*|en|*|*     $I/imta/lang/
!
! 優先言語の値が指定されていない場合は、
! ドメイン名の国別コードに基づいて通知を選択します。例 : PF= フランス領ポリネシア、BO= ボリビア
!
   *|*|*|*.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。

生成された通知の国際化

配信ステータス通知および MDN (message disposition notification) の両方に、2 つのオプションファイルが使用できます。これらは、生成された通知をより柔軟に国際化するためのものです。次の 2 種類があります。

IMTA_LANG:return_option.dat (DSN)
IMTA_LANG:disposition_option.dat (MDN)

これらのファイルに使用できるオプションを、表 10-10 に示します。

表 10-10 配信ステータス通知および MDN (message disposition notification) のオプション

オプション	説明
DAY (DSN)	RETURN_UNITS=0 (デフォルト) が設定されている場合に、%U または %u と置き換えて挿入される文字。%U と %u とは区別されません (デフォルトでは、英語の「Day」と「day」が区別して置換される)。
DIAGNOSTIC_CODE (DSN)	DSN の最初の受取人ごとのセクションの作成に使用した「Diagnostic code:」文字のオーバーライド。このフィールドには、DSN の最初の部分で使用したのと同じ文字セットを指定する必要があります。
HOUR (DSN)	RETURN_UNITS=1 が設定されている場合に、%U または %u と置き換えて挿入される文字。%U と %u とは区別されません (デフォルトでは、英語の「Hour」と「hour」が区別して置換される)。
n.n.n (DSN)	DSN の受取人ごとのセクションの作成時に、受取人単位のステータスの数字と一致するオプション名があるかどうかがチェックされます。一致するものがある場合、対応する文字が DSN に挿入されます。また、上で指定された REASON オプションの結果が長さ 0 の文字列である場合、REASON フィールドには文字は挿入されません。
ORIGINAL_ADDRESS (DSN)	DSN の最初の受取人ごとのセクションの作成に使用した「Original address:」文字のオーバーライド。このフィールドには、DSN の最初の部分で使用したのと同じ文字セットを指定する必要があります。
REASON (DSN)	DSN の最初の受取人ごとのセクションの作成に使用した「Reason:」文字のオーバーライド。このフィールドには、DSN の最初の部分で使用したのと同じ文字セットを指定する必要があります。
RECIPIENT_ADDRESS (DSN)	DSN の最初の受取人ごとのセクションの作成に使用した「Recipient address:」文字のオーバーライド。このフィールドには、DSN の最初の部分で使用したのと同じ文字セットを指定する必要があります。
RETURN_PERSONAL (DSN および MDN)	From: フィールドと一緒に使用される個人名のフィールドのオーバーライド。このフィールドは RFC 2047 エンコードされている必要があります。このオプションを指定しない場合、RETURN_PERSONAL MTA オプションで設定された値が使用されます。

表 10-10　配信ステータス通知および MDN (message disposition notification) のオプション (続き)

オプション	説明
SUBJECT (DSN および MDN)	Subject: フィールドのオーバーライド。この値は、通知に個々の件名のフィールドが含まれない場合にのみ使用されます。このフィールドは RFC 2047 エンコードされている必要があります。このオプションが使用されず、通知に件名が含まれない場合は、適切な件名が作成されます。
TEXT_CHARSET (MDN)	MDN の最初の部分および件名が変換されるべき文字セット。デフォルトでは、変換を行わないようになっています。

ステータス通知メッセージの追加機能

ステータス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。

ステータス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にある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。

例 :

```
notices 1 2 3
```

この例では、すべてのメッセージについて、一時的な配信不能のステータス通知メッセージが 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、一時的な配信不能の通知が 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。RETURN_UNITS=1 に設定した場合は、通知を 1 時間おきに受信するには、返送ジョブが 1 時間おきに実行されるようにスケジュールする必要もあります。返送ジョブが 1 時間ごとに実行されると、このジョブによって mail.log* ファイルも 1 時間ごとにロールオーバーされます。mail.log ファイルが 1 時間ごとにロールオーバーされないようにするには、imta.tailor ファイルの IMTA_RETURN_SPLIT_PERIOD テイラーファイルオプションを 24 に設定します。返送ジョブのスケジュールは、local.schedule.return_job configutil パラメータで制御します。

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。

ステータス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 が通知メッセージ (バウンスメッセージ、配信確認メッセージなど) を生成するとき、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 に提示される場合があります。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ないことをお勧めします。このようなサイトでは、元の形式の外部用アドレスのみをステータス通知メッセージに含めるほうが適しています。デフォルトは includefinal であり、最終的な形式の受取人アドレスが含められます。suppressfinal を使用すると、元の形式のアドレスが存在する場合、MTA は最終的な形式のアドレスをステータス通知メッセージに含めません。

useintermediate キーワードでは、リストの展開後、ユーザー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、最終形式が使用されます。

ポストマスターへのステータス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、Errors-to: ヘッダー行やエンベロープ 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、警告のステータス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配信の詳細については、以後の節および表 10-11 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 10-11 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。キーワード : warnpost、copywarnpost、errwarnpost、nowarnpost キーワードは、警告メッセージを postmaster に送ることを制御するために使用されます (表 10-11 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ます。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、あるいはローカルポストマスターのアドレスを入れるかを制御します。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制御します。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、ユーザー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 10-11 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ステータス通知メッセージを作成する際に使用されるポストマスターの返信アドレスは、postmaster@*local-host* です。この *local-host* の部分は、ローカルホストの正式な名前 (ローカルチャネルの名前) で、ポストマスターの個人名は「MTA e-Mail Interconnect」です。この場合、ポストマスターのアドレスの選択には注意してください。不正なアドレスを選択すると、高速のメッセージループが発生し、非常に多数のエラーメッセージが返されることになります。

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。このようなオプションが設定されていない場合は、RETURN_ADDRESS オプションと RETURN_PERSONAL オプションでデフォルトを設定します。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ー名 postmaster (大文字のみ、小文字のみ、またはその両方) 宛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のメッセージを処理する方法を制御します。

表 10-11 ポストマスターと差出人に送信される通知メッセージのキーワード

キーワード	説明
返送メッセージの内容	**通知のアドレスの指定**
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します。
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します。
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します。
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します。
返送メッセージ	**配信不能な返送メッセージの処理方法。**

表 10-11　ポストマスターと差出人に送信される通知メッセージのキーワード (続き)

キーワード	説明
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します。
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します。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ります。
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します。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ありません。
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ません。
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します。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ます。
copywarnpost	未配信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ます。
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します。
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ません。
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の指定。**
postheadonly	ポストマスターにヘッダーだけを返送します。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、ユーザー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 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りません。
postheadbody	ヘッダーとメッセージの内容の両方を返送します。
返送メッセージの内容	**通知のアドレスの指定**
includefinal	配信通知の中に最終的な形式のアドレス (受取人アドレス) を含めます。

表 10-11　ポストマスターと差出人に送信される通知メッセージのキーワード (続き)

キーワード	説明
returnenvelope	空白のエンベロープ返信アドレスの使用を制御します。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ます。 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、あるいはローカルポストマスターのアドレスを入れるかを制御します。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。 ビット 1 (値 = 2) は、MTA がすべての空白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制御します。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。
suppressfinal	オリジナル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アドレス形式を表示しないようにします。
useintermediate	リストの展開後、ユーザー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、最終形式が使用されます。
返送メッセージの内容	**通知のアドレスの指定**
aliaspostmaster	正式なチャネル名でのユーザー名ポストマスター宛のメッセージは postmaster@local-host にリダイレクトされます。local-host には、ローカルホスト名 (ローカルチャネルの名前) が入ります。
returnaddress	ローカルポストマスターの返信アドレスを設定します。
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します。
returnpersonal	ローカルのポストマスターに対する個人名を設定します。
noreturnpersonal	ポストマスター個人名に RETURN_PERSONAL オプション値を使用します。

MDN (Message Disposition Notifications) を制御する

MDN (Message Disposition Notification) は、MTA によって差出人またはポストマスター (あるいはその両方) に送信される電子メールレポートであり、メッセージの配信状態を報告します。たとえば、メッセージが Sieve フィルタによって拒否された場合、差出人に MDN が送信されます。MDN は、開封確認、確認通知、受信通知、配信確認とも呼ばれます。Sieve スクリプト言語は一般に、メッセージフィルタリングおよび不在返信メッセージに使用されます。

MDN 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

MDN の変更とローカライズについての手順は、わずかな相違を除いて、配信ステータス通知メッセージのカスタマイズとローカライズで説明されている手順と同様です。277 ページの「配信ステータス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」および 281 ページの「生成された通知の国際化」を参照してください。

マッピング (DISPOSITION_LANGUAGE マッピングと呼ばれる) は、ステータス通知を国際化するために使用される notification_language マッピングテーブル (280 ページのコード例 10-2) と同等です。

ただし、このマッピングに対する MDN のプローブは、次の形式をとります。

```
type|modifiers|source-channel|header-language|return|recipient
```

説明 :

type はディスポジションタイプで、次のいずれかを指定できます。displayed、dispatched、processed、deleted、denied、または failed。

modifiers は、ディスポジション修飾子をコンマで区切って示したリストです。現在指定できるのは次のとおりです。error、warning、superseded、および expired。

source-channel は、MDN を生成するソースチャネルです。

header-language は言語で、次のいずれかのオプションで指定します。accept-language、preferred-language、または x-accept-language (これらオプションのうち最初に指定されているものが MTA で使用される)。

return は、通知の返信先アドレスです。

recipient は、ディスポジションの対象のアドレスです。

ディスポジションマッピングの結果には、2 ～ 3 個の情報が含まれます。各情報は縦棒 (|) で区切られています。最初の情報は、開封通知のテンプレートファイルが置かれているディレクトリです。2 番目の情報は、ディスポジションテキストだけに適用される文字セットです。一部のディスポジション (特に自動返信エコーまたは不在時の Sieve 処理に対する :mime パラメータの使用によって生成されたディスポジション) では、テンプレートファイルが使用されず、その結果、テンプレートファイルから文字セットを継承することができないため、この情報は必要です。3 番目の情報は、通知の件名行のオーバーライドです。この情報は、マッピングによって $T フラグも設定されている場合にのみ使用されます。

以下の追加テンプレートファイルは、MDN を構築するときに使用されます。

```
disposition_deleted.txt disposition_failed.txt
disposition_denied.txt disposition_prefix.txt
disposition_dispatched.txt disposition_processed.txt
disposition_displayed.txt disposition_suffix.txt
disposition_option.opt
```

これらのテンプレートファイルの使用は、ステータス通知メッセージの場合のさまざまな return_*.txt ファイルの使用に相当します。*.txt ファイルのメッセージテキストは、1 行につき 78 文字以内である必要があります。

第 11 章

書き換えルールの設定

この章では、imta.cnf ファイル内で書き換えルール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 10 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、アドレスリバースデータベース、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、常に書き換えルールを使用するようにしてください。

注

imta.cnf ファイル内の書き換えルールを変更する場合は、imsimta restart 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 サーバー)。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起動については、『Messaging Server Reference Manual』を参照してください。

書き換えルールの構造

書き換えルールは、MTA 設定ファイルである imta.cnf の上半分に表示されます。設定ファイルに、各ルールが 1 行ごとに記述されています。空白行ではないコメントを、ルールとルールの間に入力できます。書き換えルールは空白行で終わり、その後にチャネル定義が続きます。設定ファイル内の書き換えルールセクションの例を以下に示します。

```
! 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。
! システムで使用するためのものではありません。
!
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。
```

書き換えルールは次の 2 つの部分から構成されます。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。書き換えルールの構造は以下のとおりです。

patterntemplate

pattern

ドメイン名の中の検索する文字列を指定します。表 11-3 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の部分と一致する場合、書き換えルールはアドレスに適用されます。パターンはスペースでテンプレートと区切る必要があります。パターンの構文については、294 ページの「書き換えルールのパターンとタグ」を参照してください。

template

以下のいずれかの形式です。

UserTemplate%*DomainTemplate*@*ChannelTag* [コントロール]

UserTemplate@*ChannelTag* [コントロール]

UserTemplate%*DomainTemplate* [コントロール]

UserTemplate@*DomainTemplate*@*ChannelTag* [コントロール]

UserTemplate@*DomainTemplate*@*SourceRoute*@*ChannelTag* [コントロール]

ここで、

UserTemplate は、アドレスのユーザー部分を書き換える方法を指定します。置換シーケンスを使用して、オリジナルのアドレスの一部、またはデータベース検索の結果を表すことができます。書き換えられたアドレスを作成するために、置換シーケンスはそれが表すものと置き換えられます。表 11-4 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、306 ページの「テンプレートの置換と書き換えルールのコントロールシーケンス」を参照してください。

DomainTemplate は、アドレスのドメイン部分を書き換える方法を指定します。*UserTemplate* と同様、*DomainTemplate* には置換シーケンスを入力できます。

ChannelTag は、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ルールとそのチャネル定義に記述されます。

controls を使用して、ルールの適用を制限することができます。コントロールシーケンスの中には、ルールの先頭に指定するものと、ルールの最後に指定するものとがあります。コントロールについては、306 ページの「テンプレートの置換と書き換えルールのコントロールシーケンス」を参照してください。

テンプレートの構文については、298 ページの「書き換えルールテンプレート」を参照してください。

書き換えルールのパターンとタグ

この節には、以下の項があります。

書き換えルールのほとんどのパターンは、該当のホストだけと一致する特定のホスト名か、サブドメイン全体の任意のホスト / ドメインと一致するサブドメインパターンのいずれかで構成されます。

たとえば、以下の書き換えルールのパターンは、指定したホストだけと一致する特定のホスト名で構成されます。

```
host.siroe.com
```

次の書き換えルールのパターンは、サブドメイン全体の任意のホストまたはドメインと一致するサブドメインのパターンで構成されます。

```
.siroe.com
```

ただし、このパターンは、ホスト名 `siroe.com` 自体とは一致しません。ホスト名 `siroe.com` 自体と一致させるには、別の `siroe.com` パターンが必要になります。

MTA は、特定のホスト名で始まるホスト / ドメイン名を書き換えてから、固有性を少なくするよう、増分で名前を生成しようとします。つまり、より固有な書き換えルールパターンは、より一般的な書き換えルールパターンに優先して使用されます。たとえば、設定ファイルに以下の書き換えルールパターンが指定されているとします。

```
hosta.subnet.siroe.com
.subnet.siroe.com
.siroe.com
```

書き換えルールパターンに基づいて、`jdoe@hosta.subnet.siroe.com` のアドレスは書き換えルールパターン `hosta.subnet.siroe.com` と一致し、`jdoe@hostb.subnet.siroe.com` のアドレスは書き換えルールパターン `.subnet.siroe.com` と一致し、`jdoe@hostc.siroe.com` のアドレスは書き換えルールパターン `.siroe.com` と一致します。

特に、インターネットのサイトではサブドメイン書き換えルールパターンを含む書き換えルールの使用が一般的です。通常、このようなサイトにはそれ自体の内部ホストおよびサブネットの多数の書き換えルールがあり、`internet.rules` ファイルからその設定に、トップレベルインターネットドメインの書き換えルールが組み込まれます (*msg_svr_base*`/config/internet.rules`)。

インターネット宛先 (より特定の書き換えルールを通じて処理される内部ホスト宛先を除く) 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。

- トップレベルインターネットドメインと一致するパターンを含む書き換えルール
- 送信する TCP/IP チャネルへのパターンなどと一致するアドレスを書き換えるテンプレート

```
!    Ascension Island
.AC                          $U%$H$D@TCP-DAEMON
. [text
.     removed for
.              brevity]
!      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.]
```

すでに説明したより一般的な種類のホストまたはサブドメインの書き換えルールパターンのほか、書き換えルールではいくつかの特殊なパターンも使われます。これについては、表 11-1 で要約し、以降の項で説明します。

表 11-1　書き換えルールの特殊パターンの要約

パターン	説明 / 使用目的
$*	任意のアドレスと一致します。このルールが指定されている場合、それがファイル内のどの位置にあっても、最初に適用されます。
$%	パーセントハックルール。A%B という形式のホスト / ドメイン仕様と一致します。
$!	bang-style ルール。B!A という形式のホスト / ドメイン仕様と一致します。
[ ]	IP リテラル全一致ルール。任意の IP ドメインリテラルと一致します。
.	任意のホスト / ドメイン仕様と一致します。たとえば、joe@[129.165.12.11]

Messaging Server には、このような特殊なパターンのほか、書き換えルールパターンに現れることのあるタグという概念があります。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、どの書き換えルール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、297 ページの「タグ付き書き換えルールセット」を参照してください。

パーセントハックに一致するルール

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 つのルールが適用されます (これらのアドレス形式については、298 ページの「書き換えルールテンプレート」を参照)。このもう 1 つのルールがパーセントハックルールです。形式は `$%` です。形式が変更されることはありません。このルールは、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ルールを含む) 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。

パーセントハックルールは、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ルール

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 つのルールが適用されます。このルールが bang-style ルールです。形式は `$!` です。形式が変更されることはありません。このルールは、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ルールを含む) 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 ルールを使用すると、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ルール

特殊パターン「.」(ドット文字) は、ほかに一致するルールがなく、ホストまたはドメイン仕様がチャネルテーブル内で見つからない場合に、任意のホストまたはドメイン仕様に一致します。つまり、「.」ルールは、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、全一致ルールが一致し、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 は単一のドット記号「.」に展開します。したがって、全一致ルールのテンプレートでは、$D の使用が制限されます。

タグ付き書き換えルールセット

書き換えプロセスを実行するにあたり、別のルールセットを追加するとうまくいく場合があります。別のルールセットを追加するには、書き換えルールタグを使用します。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、書き換えルールテンプレート内の $T という置換文字列を使って一致する書き換えルール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、別のルール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。

ルールにより、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。

書き換えルールテンプレート

以下の節では、書き換えルールのテンプレートの形式について説明します。表 11-2 にテンプレートの形式を示します。

表 11-2　書き換えルールのテンプレートの形式の要約

テンプレート	ページ	使用目的
A%B	299	A は新しいユーザー / メールボックスの名前になり、B は新しいホスト / ドメイン仕様になります。繰り返し書き換えます。
A@B	299	A%B@B として扱われます。
A%B@C	299	A は新しいユーザー / メールボックスの名前になり、B は新しいホスト / ドメイン仕様になり、ホスト C と関連するチャネルにルーティングされます。
A@B@C	299	A@B@C@C として扱われます。
A@B@C@D	299	A は新しいユーザー / メールボックスの名前になり、B は新しいホスト / ドメイン仕様になり、C をソースルートとして挿入し、ホスト D と関連するチャネルにルーティングされます。

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。ルールは、アドレスのユーザー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* [コントロール]

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ルールに使用されます。ルール適用後は、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、ルールを適用すると書き換えプロセスが終了)。

UserTemplate%*DomainTemplate* [コントロール]

たとえば、以下に示すルールを使うと、.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
```

繰り返しルールを使用する場合には、「ルールループ」が生じないように特別な注意が必要です。そのため、特に必要がない限り、繰り返し書き換えルールの使用を控えることをお勧めします。繰り返しルールを使用する際には、imsimta test -rewrite コマンドを使ってルールをテストするとよいでしょう。test -rewrite コマンドについては、『Messaging Server Reference Manual』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* と同じように機能します (最初の区切り文字が異なることに注意)。ただし、*ChannelTag* はソースルートとしてアドレスに挿入される点で異なります。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、以下に示すルールを使うと、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
```

書き換えルールテンプレートにおける大文字と小文字の区別

書き換えルール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ルール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、置換部分に特定の大文字または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、$¥ は後ろに続く置換部分を小文字にし、$^ は後ろに続く置換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、以下のルールを使うと、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ルールを適用する方法

以下に、MTA が指定アドレスに書き換えルール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ルールを検索します。
3. 一致する書き換えルールが見つかると、MTA により、そのルール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、このルール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、ルールに一致するメッセージは配信不能となります。

動作 1: 最初のホストまたはドメイン仕様を抽出する

アドレス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 アドレスルール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または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. アットマーク @ 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な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 11-3 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 11-3 抽出されるアドレスとホスト名

アドレス	最初のホスト ドメイン仕様	コメント
user@a	a	「省略形」のドメイン名。
user@a.b.c	a.b.c	「完全指定」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	省略形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。

表 11-3　抽出されるアドレスとホスト名 (続き)

アドレス	最初のホストドメイン仕様	コメント
@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」と呼ばれます。
user%A	A	
user%A%B	B	
user%%A%B	B	
A!user	A	「bang-style」のアドレス。UUCP によく使用されます。
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、パーセント記号はアットマーク (@) と同じように解釈されます (アットマークがない場合)。このルールは 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ー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」アドレスルールに従います。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: 書き換えルールを検索する

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 は書き換えルールを調べてその仕様の処理方法を明らかにします。ホストまたはドメイン仕様は、各ルールのパターン部分 (各ルールの左側) 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。

ホストまたはドメイン仕様がどのパターンにも一致しない場合は、ホストまたは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、その新しいホストまたは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換えルールだけで、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ルールテーブルだけで、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、一致するルールが見つかるか、ホストまたは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、指定した内容にもっとも近いドメインから始めて、徐々に広範な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。

- ホストまたはドメイン仕様が比較文字列 `spec_1` と `spec_2` の初期値として使用されます (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ルール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ます。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します。
- 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ます。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` に変換されます。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します。
- 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ます。`.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「.」で置き換えられます。削除後の `spec_1` 文字列の長さがゼロでない場合は、動作 1 に戻ります。削除後の新しい文字列の長さがゼロの場合 (たとえば、置換後の文字列が「.」だった場合) は、検索プロセスが失敗に終わり、マッチングプロセスが終了します。

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

ホストまたはドメイン仕様が書き換えルールに一致すると、そのホストまたはドメイン仕様はルール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ー名。
2. アドレスの新しいホストまたはドメイン仕様。
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

ホストまたは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またはドメイン仕様がない場合は、書き換え後の仕様が抽出された元の仕様に置き換えられ、書き換えプロセスが終了します。
- 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または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 (初期設定) のホストまたはドメイン仕様が削除されます。次にそのアドレスから新しいホストまたはドメイン仕様が抽出され、プロセス全体が繰り返されます。書き換えプロセスは、すべてのホストまたはドメイン仕様がなくなるか、あるいはローカルでないチャネルまたは

ルートローカルでないチャネルを介したルートが見つかるまで続けられます。MTA がソースルート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ます。

書き換えルールの失敗

ホストまたはドメイン仕様がどの書き換えルールにも一致せず、デフォルトのルールもない場合には、「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または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。

書き換え後の構文チェック

書き換えルールが適用されたあとのアドレスに対し、構文チェックは行われません。これは意図的なものです。構文チェックを行わないようにすることで、書き換えルール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ルール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。

通常のホストまたは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ルールを使って、ホストまたはドメイン仕様の書き換えが行われます。ルール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ルールが適用されます。

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ルール「.」という順に検索が実行されます。

テンプレートの置換と書き換えルールのコントロールシーケンス

置換を使用して、書き換えられたアドレスに文字列を挿入することによって、ユーザー名またはアドレスを書き換えます。この値は、使用される特定の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ー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ー名部分 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、指定した書き換えルールの適用に対して追加の条件を課します。書き換えルールのパターン部がチェックされるホストまたはドメイン仕様と一致する必要があるだけでなく、書き換えられているアドレスの他の側面も、コントロールシーケンスまたはシーケンスによる条件設定と一致する必要があります。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。以下の書き換えルールは、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
```

ドメインまたはホスト仕様が書き換えルールのパターン部分と一致しても、そのルールのテンプレートの中のコントロールシーケンスによって生じる基準のすべてとは一致しない場合、書き換えルールは失敗し、適用可能なほかのルールの検索が続けられます。

表 11-4 では、テンプレートの置換とコントロールシーケンスを要約しています。

表 11-4 書き換えルールテンプレートの置換とコントロールシーケンスの要約

置換シーケンス	置き換える内容
$D	一致するドメイン仕様の部分。
$H	ホストまたはドメイン仕様 (パターンのドットの左側) の一致しない部分。
$L	ドメインリテラル (パターンリテラルのドットの右側) の一致しない部分。
$U	オリジナルのアドレスのユーザー名。
$nA	現在のアドレスの 0 の位置から左に n 番目の文字を挿入します。n を省略した場合、アドレス全体が挿入されます。
$nX	メールホストの 0 から左に n 番目のコンポーネントを挿入します。n を省略した場合、メールホスト全体が挿入されます。
$0U	オリジナルのアドレスのローカル部分 (ユーザー名) からサブアドレスを除いたもの。
$1U	存在する場合は、オリジナルのアドレスのローカル部分 (ユーザー名)。
$$	リテラルのドル記号 ($) を挿入します。
$%	リテラルのパーセント記号 (%) を挿入します。
$@	リテラルのアットマーク (@) を挿入します。
$¥	該当部分を小文字にします。
$^	該当部分を大文字にします。
$_	元の大文字と小文字を使用します。

表 11-4 書き換えルールテンプレートの置換とコントロールシーケンスの要約 (続き)

置換シーケンス	置き換える内容
$=	後続の置換文字が、LDAP 検索フィルタへの挿入に適した引用の対象となるようにして、その部分を大文字に変換します。
$W	ランダムで一意な文字列に置換します。
$] . . . [	LDAP 検索 URL ルックアップ。
$ (テキスト)	一般データベースの置換。検索に失敗すると、ルールは失敗します。
${ . . . }	指定したマッピングを、与えられた文字列に適用します。
$[...]	カスタマ提供のルーチンを起動し、結果の置換を行います。
$&*n*	左から右にゼロから数えて、一致しない (またはワイルドカードの) ホストの *n* 番目の部分。
$!*n*	右から左にゼロから数えて、一致しない (またはワイルドカードの) ホストの *n* 番目の部分。
$**n*	左から右にゼロから数えて、一致するパターンの *n* 番目の部分。
$#*n*	右から左にゼロから数えて、一致するパターンの *n* 番目の部分。
$*n*D	一致するドメイン仕様の部分で、左側の 0 から n 番目までの部分が残されます
$*n*H	一致しないホスト / ドメイン仕様の部分で、左側の 0 から n 番目までの部分が残されます
コントロールシーケンス	**書き換えルールの効果**
$1M	チャネルが内部再処理チャネルの場合のみ適用されます。
$1N	チャネルが内部再処理チャネルではない場合のみ適用されます。
$1~	保留状態のチャネルの照合チェックを実行します。チェックに失敗した場合、現在の書き換えルールテンプレートの処理は正常に終了します。
$A	ホストがアットマーク (@) の右にある場合に適用されます
$B	ヘッダー / 本文のアドレスのみに適用されます
$C*channel*	*channel* に送信中の場合は失敗します
$E	エンベロープアドレスのみに適用されます
$F	前方を探すアドレス (例、To:) のみに適用されます
$M*channel*	*channel* がアドレスを書き換えている場合のみ適用されます
$N*channel*	*channel* がアドレスを書き換えている場合は失敗します
$P	ホストがパーセント記号の右にある場合に適用されます
$Q*channel*	*channel* に送信している場合のみ適用されます

表 11-4 書き換えルールテンプレートの置換とコントロールシーケンスの要約 (続き)

置換シーケンス	置き換える内容
$R	後方を探すアドレス (例、From:) にのみ適用されます
$S	ホストがソースルートからの場合に適用されます
$T*newtag*	書き換えルールタグを新規タグに設定します
$V ホスト	ホスト名が LDAP ディレクトリ (DC ツリー内または仮想ドメインとしてのいずれか) に定義されていない場合、失敗します。LDAP 検索がタイムアウトになると、書き換えパターンのホスト名の後の直後の文字の残りの部分は、MTA オプションの文字列 DOMAIN_FAILURE と置き換えられます。
$X	ホストが感嘆符の左にある場合に適用されます
$Z*host*	ホスト名が LDAP ディレクトリ (DC ツリー内または仮想ドメインとしてのいずれか) に定義されている場合、失敗します。LDAP 検索がタイムアウトになると、書き換えパターンのホスト名の後の直後の文字の残りの部分は、MTA オプションの文字列 DOMAIN_FAILURE と置き換えられます。
$?*errmsg*	書き換えに失敗すると、デフォルトのエラーメッセージの代わりに *errmsg* が返されます。エラーメッセージは US ASCII 文字でなければなりません。
$*number*?*errmsg*	書き換えに失敗すると、デフォルトのエラーメッセージの代わりに *errmsg* が返され、SMTP 拡張エラーコードが *a.b.c* に設定されます。 • *a* は、番号 /1000000 (最初の桁) • *b* は (番号 /1000)、余り 1000 (桁 2 から 4 の値) • *c* は 番号、余り 1000 (最後の 3 桁の値)。 以下の例では、エラーコードを 3.45.89 に設定しています。 `$3045089?the snark is a boojum`

ユーザー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ー名 (RFC 822「ローカル部」) に置き換えられます。この場合、a."b" 形式のユーザー名は "a.b" に置き換えられます。RFC2822 では、RFC 822 における古い構文の使用は推奨されていません。今後、より新しい構文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ー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 はユーザー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。

ホストまたはドメインと IP リテラルの置換 : $D、$H、$nD、$nH、$L

$H はすべて、ルールに一致しなかったホストまたはドメイン仕様の部分に置き換えられます。また、$D はすべて、ルールに一致したホストまたは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ルールに一致したとします。

```
host.siroe.com     $U%$1D@TCP-DAEMON
```

このルールが適用されると、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の部分 1 (siroe) 以降の部分 (siroe.com) に置き換えられます。

$L は、書き換えルール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 @ 文字は書き換えルールテンプレートのメタキャラクタです。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、$% は単一のパーセント記号 % (この場合、パーセントはテンプレートのフィールド区切り文字として解釈されない) に、$@ は単一のアットマーク @ (同様に、フィールド区切り文字として解釈されない) に展開されます。

LDAP クエリー URL の置換 : $]...[

`$]ldap-url[` 形式の置換シーケンスは LDAP クエリー URL として解釈され、LDAP クエリー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識別名で、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。書き換えルールの場合、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。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ルールは、たとえば次のようになります。この例で、作成された LDAP クエリー内の 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、書き換えルールの 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 11-5 に LDAP URL 置換シーケンスの一覧を示します。

表 11-5　LDAP URL 置換シーケンス

置換シーケンス	説明
$$	リテラル $ 文字
$~ アカウント	ユーザー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 (完全指定ドメイン名の最初の部分)
$L	~ または _ などの特別な先頭文字を除くユーザー名
$S	サブアドレス
$U	ユーザー名

MTA は、書き換えルールおよびマッピング内で実行された検索結果の URL をキャッシュするようになりました。この新しい URL 結果キャッシュは、URL_RESULT_CACHE_SIZE (デフォルト : 10000 エントリ) および URL_RESULT_CACHE_TIMEOUT (デフォルト : 600 秒) の 2 つの新しい MTA オプションによって制御されます。

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、imsimta 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、再帰的参照のループを避けるために、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、次の書き換えルールに 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) が書き換えルール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、ユーザー名 = jdoe、ホストまたはドメイン仕様 = 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 : ${...}

.SIROENET $($H) ${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、アドレスのユーザー名部分を選択しながら分析したり変更したり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て呼び出すのに使用します。UNIX では、MTA は dlopen および dlsym を使って、指定されたルーチンを共有ライブラリイメージからダイナミックにロードし、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きの long 型整数です。入力時に *argument* には書き換えルール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ルールテンプレートで "$[*image,routine,argument*]" に置換されます。routine は、書き換えルール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。

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、あるタイプのネームサービスに対して呼び出しを実行し、その結果を使ってアドレスを変化させることができます。次の書き換えルールを使って、ホスト siroe.com に対して前方を探すアドレス (例 : To: アドレス) のディレクトリサービス検索が次のように実行されることがあります。$F を指定すると、この書き換えルールを前方を探すアドレスだけに使用することができます。詳細は、318 ページの「方向および位置に固有の書き換えルール ($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ルールに一致すると、メモリ内に LOOKUP_IMAGE (UNIX の共有ライブラリ) がロードされ、argument パラメータとして jdoe を使って LOOKUP ルーチンが呼び出されます。その後、LOOKUP ルーチンは、John.Doe%eng.siroe.com などの別のアドレスを result パラメータに入れ、書き換えルール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 (299 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、書き換えるホストまたはドメイン仕様からサブドメイン部分を 1 つ抽出するためのものです。表 11-6 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 11-6 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または一致した部分) 内の n 番目の要素を表します (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」となります。要求された要素が存在しない場合は、書き換えは失敗します。
$!n	ホスト仕様 (ワイルドカードに一致しなかったまたは一致した部分) 内の n 番目の要素を表します (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」となります。要求された要素が存在しない場合は、書き換えは失敗します。
$*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 番目の要素を表します (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」となります。要求された要素が存在しない場合は、書き換えは失敗します。
$#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 番目の要素を表します (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」となります。要求された要素が存在しない場合は、書き換えは失敗します。

jdoe@eng.siroe.com というアドレスが次の書き換えルール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

$W コントロールシーケンスは、大文字の英数字からなる繰り返し不可能な固有のテキスト文字列を挿入します。$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ルール ($M、$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ルール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ルールチェックは行われません。そのチャネルに rules が指定されている場合は、チャネル固有の書き換えルール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ルールチェックは、ルール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、アットマーク (@)、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、$M チャネルを使用したときにチャネルが現在書き換えを行なっているチャネルでない場合は、ルールが適用されません。また、$N チャネルを使用したときにチャネルが書き換えを行なっている場合も、ルールが適用されません。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、ルールが適用されます。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、ルール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ルール ($C、$Q)

メッセージをキューに入れるチャネルに依存する書き換えルール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ルールチェックは行われません。宛先チャネルに rules が指定されている場合は、チャネル固有の書き換えルール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: アドレスのみに依存します。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、宛先チャネルが決まると、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際に $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ルールチェックは、ルール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、アットマーク (@) 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、$Q チャネルを使用したときにチャネルが宛先チャネルでない場合は、ルールが適用されません。また、$C チャネルを使用したときにチャネルが宛先である場合にも、ルールは適用されません。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、ルールが適用されます。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、ルール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ルール ($B、$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ルール、またはヘッダーアドレスにのみ適用される書き換えルール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、書き換えルール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。$F コントロールシーケンスを使うと、前方を探すアドレスである場合に書き換えルールが適用されます。$R コントロールシーケンスを使うと、後方を探すアドレスである場合に書き換えルールが適用されます。

ホスト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ルール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
- アットマーク (@) 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。

- ルール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、$S を使用します。
- ルールをアットマーク (@) の右側にあるホストに一致させるには、$A を使用します。
- ルールを % 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。

- ルールを感嘆符 (!) 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、ルールの適用が失敗に終わります。これらのシーケンスは、1 つの書き換えルール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、ルールはソースルート内のホスト名またはアットマーク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、ルール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)

現在の書き換えルールタグを変更するには、$T コントロールシーケンスを使用します。書き換えルールタグはすべての書き換えルールパターンの先頭に付けられ、その後、設定ファイルやドメインデータベースで書き換えルールパターンの検索が行われます。$T の直後からアットマーク、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、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、以下のようなルール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 とする)。

```
internet                $S$U@localhost$Tmtcp-force|

mtcp-force|.$U%$H@TCP-DAEMON
```

最初のルールは、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ルールが一致することはありません。最後に、デフォルトのルールがタグとともに試され、2 番目のルールに移ります。このルールでは、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、「不正なホストまたは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、ルール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、$? シーケンスを使用します。$? の直後からアットマーク (@)、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。

$? を含むルールもほかのルールと同じように動作します。特別なケースとして、$? だけを含むルール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ルールがあるとします。

```
. . $?Unrecognized address; contact postmaster@siroe.com
```

この例で、認識されないホストまたはドメイン仕様は、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。

多数の書き換えルールを扱う

MTA は常に `imta.cnf` ファイルからすべての書き換えルールを読み取り、メモリ内のハッシュテーブルにそれらのルールを保存します。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ルール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、書き換えルール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 個以上の書き換えルール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ルール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、設定ファイルのルール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、指定されたルール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、ルール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ルールが無効になります。デフォルトでは、ドメインデータベースはホストしているドメインに関連する書き換えルール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 *msg_svr_base*`/data/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。

書き換えルールをテストする

書き換えルールをテストするには `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、『Messaging Server Reference Manual』を参照してください。

書き換えルールの例

以下に、書き換えルールの例とそれらのルールによってサンプルアドレスがどの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、次の例で示す書き換えルールが含まれているとします。

```
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 11-7 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 11-7 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

表 11-7 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@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
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、これらの書き換えルール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、ユーザーに送ります。cs.cmu.edu を 1 つの部分からなる短形式の名前に追加し、再試行します。.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、もう一度試行します。また、.cs.siroe も .cs.siroe.edu に変換し、もう一度試行します。

名前が sd.cs.siroe.edu (ユーザー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、そこに送ります。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) に送ります。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) に送ります。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) に送ります。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 に送ります。

上記の例のように、書き換えルールによってアドレスのユーザー名 (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。アドレスのユーザー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 (ホストまたはドメイン仕様をアドレスのユーザー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 12 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 10 章「MTA サービスと設定について」、196 ページの「チャネル定義」、および 234 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。

注　imta.cnf 内のチャネル定義を変更する場合は、imsimta restart 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 サーバー)。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起動については、『Messaging Server Reference Manual』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 12-1　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	382	822	382	addreturnpath	389	addrsperfile	407
Aliasdetourhost	415	aliaslocal	392	aliaspostmaster	285	allowetrn	352
allowswitchchannel	364	alternatechannel	405	alternateblocklimit	405	alternatelinelimit	405
alternaterecipientlimit	405	authrewrite	367	backoff	374	bangoverpercent	384
bangstyle	382	bidirectional	373	blocketrn	352	blocklimit	404
cacheeverything	361	cachefailures	361	cachesuccesses	361	channelfilter	413
charset7	355	charset8	355	charsetesc	355	checkehlo	351
commentinc	390	commentmap	390	commentomit	390	commentstrip	390
commenttotal	390	connectalias	385	connectcanonical	385	copysendpost	284
copywarnpost	284	daemon	365	datefour	397	datetwo	397
dayofweek	397	defaulthost	386	defaultmx	363	defaultnameservers	364
deferralrejectlimit	419	deferred	373	defragment	400	dequeue_removeroute	393
destinationfilter	413	destinationnosolicit	418	destinationspamfilterXoptin	414	disableetrn	352
dispositionchannel	412	disconnectbadauthlimit	403	disconnectbadcommandlimit	410	domainetrn	352

表 12-1　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
domainvrfy	353	dropblank	388	ehlo	351	eightbit	355
eightnegotiate	355	eightstrict	355	errsendpost	284	errwarnpost	284
expandchannel	380	expandlimit	380	expnallow	354	expndisable	354
expndefault	354	exproute	384	fileinto	413	filesperjob	376
filter	413	forwardcheckdelete	361	forwardchecknone	361	forwardchecktag	361
header_733	382	header_822	382	header_uucp	382	headerlabelalign	398
headerlimit	408	headerlinelength	398	headerread	395	headertrim	395
holdexquota	407	holdlimit	380	identnone	362	identnonelimited	362
identnonenumeric	362	identnonesymbolic	362	identtcp	362	identtcplimited	362
identtcpsymbolic	362	ignoreencoding	400	immnonurgent	373	improute	384
includefinal	283	indenttcpnumeric	362	inner	395	innertrim	395
interfaceaddress	360	interpretencoding	400	language	399	lastresort	364
linelength	402	linelimit	404	localvrfy	353	logging	411
logheader	411	loopcheck	412	mailfromdnsverify	354	master	373
master_debug	411	maxblocks	401	maxheaderaddrs	398	maxheaderchars	398
maxjobs	376	maxlines	401	maxprocchars	398	maysaslserver	366
maytls	369	maytlsclient	369	maytlsserver	369	missingrecipientpolicy	387
msexchange	369	multiple	407	mustsaslserver	366	musttls	369
musttlsclient	369	musttlsserver	369	mx	363	namelengthlimit	408
nameservers	364	noaddreturnpath	389	nobangoverpercent	384	noblocklimit	404
nocache	361	nochannelfilter	413	nodayofweek	397	nodefaulthost	386
nodeferred	373	nodefragment	400	nodestinationfilter	413	nodropblank	388
noehlo	351	noexproute	384	noexquota	407	nofileinto	413
nofilter	413	noheaderread	395	noheadertrim	395	noimproute	384
noinner	395	noinnertrim	395	nolinelimit	404	nologging	411
noloopcheck	412	nomailfromdnsverify	354	nomaster_debug	411	nomsexchange	367
nomx	363	nonrandomemx	363	nonurgentbackoff	374	nonurgentblocklimit	378

表 12-1　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
nonurgentnotices	283	noreceivedfor	389	noreceivedfrom	389	noremotehost	386
norestricted	388	noreturnaddress	285	noreturnpersonal	285	noreverse	388
normalbackoff	374	normalblocklimit	378	normalnotices	283	norules	393
nosasl	366	nosaslserver	366	nosaslswitchchannel	366	nosendetrn	352
nosendpost	284	noservice	381	noslave_debug	411	nosmtp	351
nosourcefilter	413	noswitchchannel	364	notices	283	notificationchannel	412
notls	369	notlsclient	369	notlsserver	369	novrfy	353
nowarnpost	284	nox_env_to	396	parameterlengthlimit	408	percentonly	384
percents	382	personalinc	391	personalmap	391	personalomit	391
personalstrip	391	pool	375	port	360	postheadbody	285
postheadonly	285	randommx	363	receivedfor	389	receivedfrom	389
recipientcutoff	408	recipientlimit	408	rejectsmtplonglines	407	remotehost	386
restricted	388	returnaddress	285	returnenvelope	285	returnpersonal	285
reverse	388	routelocal	385	rules	393	rules	393
saslswitchchannel	366	sendetrn	352	sendpost	284	sensitivitycompanyconfidential	399
sensitivitynormal	399	sensitivitypersonal	399	sensitivityprivate	399	service	381
sevenbit	355	silentetrn	352	single	407	single_sys	365
slave	373	slave_debug	411	smtp	351	smtp_cr	351
smtp_crlf	351	smtp_crorlf	351	smtp_lf	351	sourceblocklimit	404
sourcecommentinc	390	sourcecommentmap	390	sourcecommentomit	390	sourcecommentstrip	390
sourcecommenttotal	390	sourcefilter	413	sourcenosolicit	418	sourcepersonalinc	391
sourcepersonalmap	391	sourcepersonalomit	391	sourcepersonalstrip	391	sourceroute	382
sourcespamfilterXoptin	414	streaming	357	subaddressexact	392	subaddressrelaxed	392
subaddresswild	392	subdirs	410	submit	413	suppressfinal	283

表 12-1　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
switchchannel	364	threaddepth	379	tlsswitchchannel	369	transactionlimit	378
truncatesmtplonglines	407	unrestricted	388	urgentbackoff	374	urgentblocklimit	378
urgentnotices	283	useintermediate	283	user	413	uucp	382
viaaliasoptional	394	viaaliasrequired	394	vrfyallow	353	vrfydefault	353
vrfyhide	353	warnpost	284	wrapsmtplonglines	407	x_env_to	396

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。次のような種類があります。

表 12-2　機能別チャネルキーワード

キーワード	ページ	定義
アドレス処理		
733	382	エンベロープで % ルーティングを使用します。percents と同義。
822	382	エンベロープでソースルートを使用します。sourceroute と同義。
addreturnpath	389	このチャネルのキューに入れるメッセージに Return-Path: ヘッダーを追加します。
aliaslocal	392	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します。
authrewrite	367	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します。
bangoverpercent	384	A!B%C を A!(B%C) としてグループ化します。
bangstyle	382	エンベロープで UUCP! ルーティングを使用します。uucp と同義。
defaulthost	386	アドレスを完成させるためにドメイン名を指定します
dequeue_removeroute	393	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。
exproute	384	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します。
holdlimit	380	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、メッセージを保留します。
improute	384	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ます
missingrecipientpolicy	387	受取人ヘッダーがないメッセージを有効にする (どのヘッダーを追加するか指定する) ポリシーを設定します。
noaddreturnpath	389	メッセージをキューに入れる際に、メッセージに Return-Path: ヘッダーを追加しません。
nobangoverpercent	384	A!B%C を (A!B)%C としてグループ化します。
nodefaulthost	386	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ません
noexproute	384	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ません
noimproute	384	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ません
noreceivedfrom	389	元のエンベロープの From: アドレスを含めずに Received: ヘッダー行を作成します。
noremotehost	386	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します
norestricted	388	unsrestricted と同じ。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
noreverse	388	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します
norules	393	このチャネル固有の書き換えルールを確認しません。
percentonly	384	bang パスを無視します。エンベロープで % ルーティングを使用します。
percents	382	エンベロープで % ルーティングを使用します。733 と同義。
remotehost	386	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します
restricted	388	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します。
reverse	388	アドレスリバースデータベースまたは REVERSE マッピングに対してアドレスを確認します
routelocal	385	アドレスをチャネルに書き換える際に、MTA がアドレスのすべての明示的ルーティングの短絡化を試行するようにします。
rules	393	このチャネル固有の書き換えルールを確認します。
sourceroute	382	822 と同義。
subaddressexact	392	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いません。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ります。
subaddressrelaxed	392	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します。
subaddresswild	392	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します。
unrestricted	388	RFC 1137 エンコーディングとデコーディングを実行しないように MTA に指示します。
uucp	382	エンベロープで UUCP! ルーティングを使用します。bangstyle と同義。
viaaliasoptional	394	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ありません。
viaaliasrequired	394	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ります。
添付と MIME 処理		
defragment	400	このチャネルのキューに入れられる部分メッセージは、代わりに再組み立てチャネルのキューに入れられます。
ignoreencoding	400	着信メッセージの Encoding: ヘッダーを無視します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
interpretencoding	400	着信メッセージの Encoding: ヘッダーを必要に応じて解釈します。
nodefragment	400	再組み立てを無効にします。
文字セットと 8 ビットデータ		
charset7	355	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	355	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	355	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	355	チャネルが 8 ビット文字をサポートします。
eightnegotiate	355	チャネルが 8 ビット転送の使用をネゴシエートします (可能な場合)。
eightstrict	355	ネゴシエーションが行われていない 8 ビットデータがメッセージヘッダーに含まれている場合は、そのメッセージを拒否します。
sevenbit	355	8 ビット文字をサポートしません。8 ビット文字はエンコードされる必要があります。
MTA キュー領域でのファイル作成		
addrsperfile	407	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の制限
expandchannel	380	expandlimit の適用による遅延拡張を実行するチャネルを指定します。
expandlimit	380	アドレスの数がこの制限を超えた場合、着信メッセージを「オフライン」で処理します。
multiple	407	メッセージファイル内の受取人数を制限しません。ただし SMTP チャネルのデフォルトは 99 です。
single	407	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 つずつ作成されます。
single_sys	407	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。
subdirs	410	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します。
ヘッダー		
authrewrite	367	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します。
commentinc	390	メッセージのヘッダー行内のコメントをそのままにします。
commentmap	390	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します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
commentomit	390	メッセージのヘッダー行内のコメントを取り除きます。
commentstrip	390	メッセージのヘッダー行内のコメントフィールドから問題を起こす文字を取り除きます。
commenttotal	390	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します。ただし、推奨しません。
datefour	397	すべての年表示フィールドを 4 桁に展開します。
datetwo	397	4 桁の日付表示から先頭の 2 桁を削除します。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でください。
dayofweek	397	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します。
defaulthost	386	アドレスを完成させるためにドメイン名を指定します
dropblank	388	着信メッセージから不正な空白ヘッダーを削除します。
header_733	382	メッセージヘッダーで % ルーティングを使用します。
header_822	382	メッセージヘッダーでソースルートを使用します。
headerlabelalign	398	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します。整数値の引数をとります。
headerlinelength	398	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します。
headerread	395	オリジナルのメッセージヘッダーが処理される前に、メッセージがキューに入れられたときに、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ルールを適用します (注意して使用すること)。
headertrim	395	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ルールを適用します。
header_uucp	382	ヘッダーで ! ルーティングを使用します
inner	395	メッセージをパースして、内部ヘッダーを書き換えます。
innertrim	395	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ルールを適用します (注意して使用すること)。
language	399	ヘッダーにデフォルトの言語を指定します。
maxheaderaddrs	398	1 行に表示できるアドレスの数を指定します。
maxheaderchars	398	1 行に表示できる文字数を指定します。
missingrecipientpolicy	387	受取人ヘッダーがないメッセージを有効にする (どのヘッダーを追加するか指定する) ポリシーを設定します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
nodayofweek	397	日付 / 時刻ヘッダーから曜日情報を削除します。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でください。
nodefaulthost	386	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ません。
nodropblank	388	着信メッセージから不正な空白ヘッダーを削除しません。
noheaderread	395	オプションファイルからのヘッダートリミングルールを適用しません。
noheadertrim	395	オプションファイルからのヘッダートリミングルールを適用しません。
noinner	395	内部のメッセージヘッダー行を書き換えません。
noinnertrim	395	内部のメッセージヘッダーにヘッダートリミングルールを適用しません。
noreceivedfor	389	エンベロープ受取人の情報を含めずに `Received:` ヘッダー行を作成します。
noreceivedfrom	389	元のエンベロープの From: アドレスを含めずに `Received:` ヘッダー行を作成します。
noremotehost	386	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します
noreverse	388	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します
norules	393	このチャネル固有の書き換えルールを確認しません。
nox_env_to	396	`X-Envelope-to` ヘッダー行を削除します。
personalinc	391	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します。
personalmap	391	PERSONAL_NAMES マッピングテーブルを通じて、個人名を実行します。
personalomit	391	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します。
personalstrip	391	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します。
receivedfor	389	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 Received: ヘッダー行に `To:` アドレスを含めます。
receivedfrom	389	MTA がエンベロープ `From:` アドレスを変更した場合、着信メッセージの `Received:` ヘッダー行を作成する際に元のエンベロープの `From:` アドレスを含めます。
remotehost	386	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します

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
restricted	388	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します。
reverse	388	アドレスリバースデータベースまたは REVERSE マッピングに対してアドレスを確認します
rules	393	このチャネル固有の書き換えルールを確認します。
sensitivitycompanyconfidential	399	`Companyconfidential` が、受け付けるメッセージの重要度の上限です。
sensitivitynormal	399	`Normal` が、受け付けるメッセージの重要度の上限です。
sensitivitypersonal	399	`Personal` が、受け付けるメッセージの重要度の上限です。
sensitivityprivate	399	`Private` が、受け付けるメッセージの重要度の上限です。
sourcecommentinc	390	着信メッセージのヘッダー行にコメントを残します。
sourcecommentmap	390	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します。
sourcecommentomit	390	着信メッセージの To:、From:、Cc: などのヘッダー行からコメントを削除します。
sourcecommentstrip	390	着信メッセージのヘッダー行内のコメントフィールドから問題を起こす文字を削除します。
sourcecommenttotal	390	着信メッセージから、( ) 内に入っているコメントを削除します。
sourcepersonalinc	391	着信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します。
sourcepersonalmap	391	ソースチャネルを通じて個人名を実行します。
sourcepersonalomit	391	着信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します。
sourcepersonalstrip	391	着信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します。
unrestricted	388	RFC 1137 エンコーディングとデコーディングを実行しないように MTA に指示します。
x_env_to	396	`X-Envelope-to` ヘッダー行の生成を有効にします。
着信チャネルの一致と切り替え		
allowswitchchannel	364	`switchchannel` チャネルからこのチャネルへの切り替えを許可します
nosaslswitchchannel	366	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ません
noswitchchannel	364	このチャネルへの、またはこのチャネルからの切り替えを行いません。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
switchchannel	364	サーバー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ます。
saslswitchchannel	366	クライアントが SASL の使用に成功した場合、着信接続が指定のチャネルに切り替えられます。
tlsswitchchannel	369	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ます。
ログ記録とデバッグ		
logging	411	キューに対するメッセージの出入りをログに記録し、特定のチャネルのログ機能を有効にします。
loopcheck	412	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します。
master_debug	411	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します。
nologging	411	キューに対するメッセージの出入りをログに記録しません。
noloopcheck	412	SMTP EHLO 応答見出しに文字列がありません。
nomaster_debug	411	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いません。
noslave_debug	411	スレーブのデバッグ出力を生成しません。
slave_debug	411	スレーブのデバッグ出力を生成します。
長いアドレスリストやヘッダー		
expandchannel	380	expandlimit の適用による遅延拡張を実行するチャネルを指定します。
expandlimit	380	アドレスの数がこの制限を超えた場合、着信メッセージを「オフライン」で処理します。
holdlimit	380	アドレスの数がこの制限を越えた場合、メッセージを保留します。
maxprocchars	398	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	413	チャネルフィルタファイルの場所。`destinationfilter` と同じ。
destinationfilter	413	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。
destinationspamfilterXoptin	414	スパムのフィルタ処理のソフトウェア X によってこのチャネル宛てのメッセージを起動します。
fileinto	413	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します。
filter	413	ユーザーフィルタファイルの場所を指定します。
nochannelfilter	413	送信メッセージに対するチャネルフィルタリングを行いません。`nodestinationfilter` とも呼ばれます。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
nodestinationfilter	413	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ません。
nofileinto	413	メールボックスフィルタ fileinto のオペレータが効果を発揮しません。
nofilter	413	ユーザーメールボックスのフィルタリングを実行しません。
nosourcefilter	413	着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ません。
sourcefilter	413	着信メッセージ用のチャネルフィルタファイルの場所を指定します。
sourcespamfilterXoptin	414	スパムのフィルタ処理のソフトウェア X によってこのチャネルから発信されるメッセージを起動します。
非請求の SMTP 拡張のサポート		
sourcenosolicit	418	このチャネルから送信されたメール内でブロックされる請求フィールドの値のコンマ区切りの一覧を指定します。
destinationnosolicit	418	このチャネルのキューに入れられたメール内で受け入れられない請求フィールドの値のコンマ区切りの一覧を指定します。
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 275 ページを参照)		
aliaspostmaster	285	正式なチャネル名でのユーザー名が postmaster であるユーザーに宛てられたメッセージは postmaster@ ローカルホストにリダイレクトされます。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) が入ります。
copysendpost	284	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ます。
copywarnpost	284	未配信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ます。
errsendpost	284	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します。
errwarnpost	284	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します。
includefinal	283	配信通知の中に受取人アドレスの最終的な形式を含めます。
nonurgentnotices	283	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します。
noreturnaddress	285	ポストマスターアドレス名に RETURN_ADDRESS オプション値を使用します。
noreturnpersonal	285	ポストマスター個人名に RETURN_PERSONAL オプション値を使用します。
normalnotices	283	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します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
nosendpost	284	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ません。
notices	283	通知を送り、メッセージを返すまでの時間を指定します。
nowarnpost	284	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ません。
postheadbody	285	ヘッダーとメッセージの内容の両方を返送します。
postheadonly	285	ポストマスターにヘッダーだけを返送します。
returnaddress	285	ローカルポストマスターの返信アドレスを設定します。
returnenvelope	285	空白のエンベロープ返信アドレスの使用を制御します。
returnpersonal	285	ローカルのポストマスターに対する個人名を設定します。
sendpost	284	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します。
suppressfinal	283	オリジナル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アドレス形式を表示しないようにします。
urgentnotices	283	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します。
useintermediate	283	リストの展開後、ユーザー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。
warnpost	284	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します。
処理制御とジョブ送信 **(より大きい機能単位については 371 ページの表 12-7 を参照)**		
backoff	374	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	373	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	373	`Deferred-delivery:` ヘッダー行を認識し、許可します。
expandchannel	380	expandlimit の適用による遅延拡張を実行するチャネルを指定します。
expandlimit	380	アドレスの数がこの制限を超えた場合、着信メッセージを「オフライン」で処理します。
filesperjob	376	1 つのジョブで処理できるキューエントリの数。
immnonurgent	373	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します。
master	373	マスタープログラムによって処理されるチャネル (`master`)。
maxjobs	376	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	373	`Deferred-delivery:` ヘッダー行が許可されないように指定します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
nonurgentbackoff	374	優先度が低いメッセージの配信試行頻度。
nonurgentblocklimit	378	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) に設定します。該当するメッセージは次の定期ジョブまで処理されません。
normalbackoff	374	優先度が標準であるメッセージの配信試行頻度。
normalblocklimit	378	指定値以上のサイズを持つメッセージの優先度を「低」に設定します。
noservice	381	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 を使用して有効にします。
pool	375	チャネル用のプールを指定します。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します。
service	381	CHARSET-CONVERSION エントリにかかわらず、無条件でサービス変換を有効にします。
slave	373	スレーブプログラム (slave) によって処理されるチャネル。
threaddepth	379	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。
transactionlimit		1 つの接続について許されるメッセージの数を制限します。
urgentbackoff	374	優先度が高いメッセージの配信試行頻度。
urgentblocklimit	378	指定値以上のサイズを持つメッセージの優先度を「標準」に設定します。
ユーザー	413	pipe チャネルでどのユーザー名で実行するかを示すのに使用されます。
重要度の上限		
sensitivitycompanyconfidential	399	受け付けるメッセージの重要度の上限。
sensitivitynormal	399	Normal が、受け付けるメッセージの重要度の上限です。
sensitivitypersonal	399	Personal が、受け付けるメッセージの重要度の上限です。
sensitivityprivate	399	Private が、受け付けるメッセージの重要度の上限です。
メッセージの制限、ユーザー制限容量、権限、認証の試行		
alternatechannel	405	alternateblocklimit、alternatelinelimit、および alternaterecipientlimit の代替宛先チャネル
alternateblocklimit	405	メッセージが alternativechannel に送信される前に、メッセージのブロック数の制限を指定します。
alternatelinelimit	405	メッセージが alternativechannel に送信される前に、メッセージの行数の制限を指定します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
alternaterecipientlimit	405	メッセージが alternativechannel に送信される前に、メッセージの受取人数の制限を指定します。
blocklimit	404	1 つの着信メッセージに対して許可されている MTA ブロックの最大数。
disconnectbadauthlimit	403	1 つのセッションで許可される認証の試行失敗回数の制限。この回数に達するとセッションの接続は切断されます。
disconnectbadcommandlimit	410	セッションの不良コマンド数を制限します。
disconnectrecipientlimit	410	セッションの受取人数を制限します。
disconnectrejectlimit	410	拒否される受取人数を制限します。
disconnecttransactionlimit	410	トランザクション数を制限します。
headerlimit	408	プライマリ (もっとも外側の) メッセージヘッダーの最大サイズの制限
holdexquota	407	制限容量を超過したユーザーに対するメッセージを保留します。
holdlimit	380	アドレスの数がこの制限を越えた場合、着信メッセージを保留します。
linelength	402	チャネルごとに許可される最大のメッセージ行の長さを制限します。
linelimit	404	1 つのメッセージに対して許可される最大の行数を指定します。
maxblocks	401	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。
maxlines	401	1 つのメッセージに許可する最大行数を指定します。
nameparameterlengthlimit	408	name content-type および filename content-disposition パラメータが切り捨てられる位置を制御します。
noblocklimit	404	1 つのメッセージに許可される MTA ブロックの数に制限はありません。
noexquota	407	制限容量を超過したユーザーに宛てられたメッセージをすべて差出人に送り返します。
nolinelimit	404	1 つのメッセージに許可される行数に制限はありません。
nonurgentblocklimit	378	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) に設定します。該当するメッセージは次の定期ジョブまで処理されません。
normalblocklimit	378	指定値以上のサイズを持つメッセージの優先度を「低」に設定します。
parameterlengthlimit	408	general content-type および content-disposition パラメータが切り捨てられる位置を制御します。
recipientcutoff	408	受取人がこの値を超えるとメッセージを拒否します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
recipientlimit	408	メッセージが受け付ける受取人アドレスの数を制限します。
rejectsmtplonglines	407	1000 文字よりも長い行 (CRLF を含む) を含むメッセージを拒否します。
sourceblocklimit	404	1 つの着信メッセージに対して許可されている MTA ブロックの最大数。
truncatesmtplonglines	407	1 行が 1000 文字を超えるとそれ以降の文字を切り捨てます。
wrapsmtplonglines	407	1 行が 1000 文字を超えると折り返します。
urgentblocklimit	378	指定値以上のサイズを持つメッセージの優先度を「標準」に設定します。
SMTP 認証、SASL、TLS (より大きい機能単位については 366 ページを参照)		
authrewrite	367	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します。
maysaslserver	366	クライアントが SASL 認証を使用することを許可します。
maytls	369	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。
maytlsclient	369	MTA SMTP クライアントは TLS をサポートする SMTP サーバーにメッセージを送信する際に TLS を使用します。
maytlsserver	369	MTA SMTP サーバーが 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します。
msexchange	369	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します。
mustsaslserver	366	SMTP サーバーは、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。
musttls	369	MTA は送着信接続に必ず TLS を使用します。
musttlsclient	369	MTA SMTP クライアントは、メッセージの送信に必ず TLS を使用します (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	369	MTA SMTP サーバーが STARTTLS 拡張をサポートすることを通知し、メッセージを着信する際に TLS を使用します。
nomsexchange	367	デフォルト。
nosasl	366	SASL 認証は許可されず、試行もされません。
nosaslserver	366	SASL 認証は許可されません。
notls	369	TLS 認証は許可されません。試行もされません。
notlsclient	369	送信接続時に MTA SMTP クライアントは TLS を使用しません (送信接続時に STARTTLS コマンドが発行されない)。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
notlsserver	369	着信接続時に MTA SMTP サーバーは TLS の使用を許可しません (SMTP サーバーもコマンド自体も STARTTLS 拡張に通知しない)。
saslswitchchannel	366	クライアントが SASL の使用に成功した場合、着信接続が指定のチャネルに切り替えられます。
tlsswitchchannel	369	クライアントが TLS ネゴシエーションに成功した場合、着信接続が指定のチャネルに切り替えられます。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。
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 349 ページの表 12-4 を参照)		
allowetrn	352	ETRN コマンドを処理します。
blocketrn	352	ETRN コマンドをブロックします。
checkehlo	351	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します。
disableetrn	352	ETRN SMTP コマンドのサポートを無効にします。
domainetrn	352	ドメインを指定する ETRN コマンドだけを処理します。
domainvrfy	353	完全なアドレスを使用して VRFY コマンドを発行します。
ehlo	351	初期接続に SMTP EHLO コマンドを使用します。
eightbit	355	チャネルが 8 ビット文字をサポートします。
eightnegotiate	355	チャネルが 8 ビット転送の使用をネゴシエートします (可能な場合)。
eightstrict	355	ネゴシエーションが行われていない 8 ビットデータがメッセージヘッダーに含まれている場合は、そのメッセージを拒否します。
expnallow	354	`DISABLE_EXPAND` SMTP チャネルオプションによって SMTP サーバーレベルで `EXPN` が無効にされている場合でも、EXPN を許可します。
expndisable	354	`EXPN` を無条件で無効にします。
expndefault	354	SMTP サーバーの設定で `EXPN` が許可されていれば EXPN を許可します
localvrfy	353	ローカルアドレスを使用して VRFY コマンドを発行します。
mailfromdnsverify	354	`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ます。
noehlo	351	EHLO コマンドを使用しません。
nomailfromdnsverify	354	`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ません。
nosendetrn	352	ETRN コマンドを送信しません。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
nosmtp	351	SMTP プロトコルをサポートしません。これがデフォルトです。
novrfy	353	VRFY コマンドを発行しません。
sendetrn	352	ETRN コマンドを送信します。
sevenbit	355	8 ビット文字をサポートしません。8 ビット文字はエンコードされる必要があります。
silentetrn	352	チャネル情報をエコーせずに ETRN コマンドを処理します。
smtp	351	SMTP プロトコルをサポートします。キーワード `smtp` は、すべての SMTP チャネルで必須です (このキーワードは `smtp_crorlf` と同等)。
smtp_cr	351	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ます。
smtp_crlf	351	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_crorlf	351	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す。
smtp_lf	351	キャリッジリターン (CR) なしの、ラインフィーダ (LF) のみを使用できます。
streaming	357	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します。
vrfyallow	353	VRFY コマンドに対して詳細な情報を提供する応答を出します。
vrfydefault	353	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します。
vrfyhide	353	SMTP VRFY コマンドに対してあいまいな応答を出します。
TCP/IP 接続と DNS 検索のサポート **(より大きい機能単位については 358 ページの表 12-5 を参照)**		
cacheeverything	361	すべての接続情報をキャッシュします。
cachefailures	361	接続失敗に関する情報だけをキャッシュします。
cachesuccesses	361	接続成功に関する情報だけをキャッシュします。
connectalias	385	受取人のアドレスに書かれているホストに配信します。
connectcanonical	385	MTA が接続するシステムのホストエイリアスに接続します。
daemon	365	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します。
defaultmx	363	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します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
defaultnameservers	364	TCP/IP スタックが選択したネームサーバーを照合します。
forwardcheckdelete	361	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します。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ます。
forwardchecknone	361	DNS リバース検索のあとに正引き検索を実行しません。
forwardchecktag	361	リバース DNS 検索を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します。一致しなければ名前に「*」を付けます。
identnone	362	IDENT 検索を実行しません。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ます。
identnonelimited	362	IDENT 検索を実行しません。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ます。
identnonenumeric	362	IDENT 検索および IP からホスト名への変換を実行しません。
identnonesymbolic	362	IDENT 検索を実行しません。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ます。
identtcp	362	着信 SMTP 接続での IDENT 検索および 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ます
identtcplimited	362	着信 SMTP 接続での IDENT 検索と IP からホスト名への変換を実行します (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 アドレスを含めます。
indenttcpnumeric	362	着信 SMTP 接続で IDENT 検索を実行します。IP からホスト名への変換を実行しません。
identtcpsymbolic	362	着信 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ます。
interfaceaddress	360	指定された TCP/IP インタフェースアドレスにバインドします。
lastresort	364	最後のホストを指定します。
mailfromdnsverify	354	`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ます。
mx	363	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します。
nameservers	364	TCP/IP スタックが選択したネームサーバーの代わりに照合するネームサーバーのリストを指定します。`nameservers` には、空白文字で区切られたネームサーバーの IP アドレスのリストが必要です。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
nocache	361	接続情報をキャッシュしません。
nomailfromdnsverify	354	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ません。
nomx	363	TCP/IP ネットワークが MX 検索をサポートしません。
nonrandomemx	363	MX 検索を実行しますが、返されたエントリを同等の優先度でランダム化しません。
port	360	SMTP 接続用のデフォルトポート番号を指定します。標準ポートは 25 です。
randommx	363	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します。
single	365	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します。
single_sys	365	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。
threaddepth	379	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。
その他		
deferralrejectlimit	419	不正な RCPT TO: アドレス数を制限します
dispositionchannel	412	配信ステータス通知 (DSN) を最初にキューに入れるチャネルとしてプロセスチャネルよりも優先されます。
destinationfilter	413	一般的な MTA チャネルで使用して、チャネルレベルのフィルタを指定して送信メッセージに適用します。
filter	413	フィルタファイルの場所を示す URL を必要な引数としてとります
nodestinationfilter	413	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。
nosourcefilter	413	どのチャネルメールボックスフィルタもソースチャネルに対して無効になります。
nofilter	413	デフォルトで、ユーザーメールボックスフィルタがチャネルに対して有効ではないことを示します。
notificationchannel	412	MDN (Message Disposition Notification) を最初にキューに入れるチャネルとしてプロセスチャネルよりも優先されます。
sourcefilter	413	一般的な MTA チャネルで使用して、チャネルレベルのフィルタを指定して着信メッセージに適用します
submit	413	チャネルを送信専用のチャネルに指定します。

表 12-2　機能別チャネルキーワード (続き)

キーワード	ページ	定義
user	413	pipe チャネルでどのユーザー名で実行するかを示すのに使用されます。

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る)。

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、書き換えルール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 サーバーが含まれます。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 12-3 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します。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ります。
`tcp_intranet`	イントラネット内のメールを送受信します。
`tcp_auth`	tcp_local のスイッチチャネルとして使用されます。認証されたユーザーは、リレーブロックの制約を回避するため tcp_auth チャネルに移されます。
`tcp_submit`	送信されたメッセージ (通常の場合はユーザー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 で受け入れます (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または削除することで、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したりできます。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 (*msg_svr_base*/`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。詳細は、『Sun Java System Messaging Server Administration Reference』を参照してください。

この節には、以下の項があります。

- 348 ページの「SMTP チャネルオプションを設定する」
- 348 ページの「SMTP コマンドとプロトコルのサポート」
- 358 ページの「TCP/IP 接続と DNS 検索のサポート」
- 366 ページの「SMTP 認証、SASL、TLS」

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。チャネルオプションファイルは MTA 設定ディレクトリに保存し、*x*_option という名前を付けてください。*x* はチャネル名です。たとえば、/*msg_svr_base*/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、サーバーのメーリングリストの展開を無効にするには、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構文の詳細については、『Messaging Server Reference Manual』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、EXPN、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。

- 353 ページの「VRFY コマンドのサポート」
- 354 ページの「DNS ドメイン確認」
- 355 ページの「文字セットのラベルと 8 ビットデータ」
- 357 ページの「プロトコルストリーミング」

表 12-4 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 12-4　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します。**
smtp	SMTP プロトコルをサポートします。キーワード smtp は、すべての SMTP チャネルで必須です (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ません。これがデフォルトです。
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ます。
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改行記号として認識されます。
smtp_lf	キャリッジリターン (CR) なしの、ラインフィーダ (LF) のみを使用できます。
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す。
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します**
ehlo	初期接続に SMTP EHLO コマンドを使用します。
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します。
noehlo	EHLO コマンドを使用しません。
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) の処理方法を指定します**
allowetrn	ETRN コマンドを処理します。
blocketrn	ETRN コマンドをブロックします。
domainetrn	ドメインを指定する ETRN コマンドだけを処理します。
silentetrn	チャネル情報をエコーせずに ETRN コマンドを処理します。
sendetrn	ETRN コマンドを送信します。
nosendetrn	ETRN コマンドを送信しません。

表 12-4　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
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します**
domainvrfy	完全なアドレスを使用して VRFY コマンドを発行します。
localvrfy	ローカルアドレスを使用して VRFY コマンドを発行します。
novrfy	VRFY コマンドを発行しません。
vrfyallow	VRFY コマンドに対して詳細な情報を提供する応答を出します。
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します。
vrfyhide	SMTP VRFY コマンドに対してあいまいな応答を出します。
EXPN キーワード	**チャネルによる EXPN キーワードの処理方法を指定します**
expnallow	DISABLE_EXPAND SMTP チャネルオプションによって SMTP サーバーレベルで EXPN が無効にされている場合でも、EXPN を許可します。
expndisable	EXPN を無条件で無効にします。
expndefault	SMTP サーバーの設定で EXPN が許可されていれば EXPN を許可します (デフォルト)。
DNS ドメイン検査	**チャネルが DNS ドメイン確認を行うかどうかを指定します**
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します。
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ません。
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します (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します。
eightnegotiate	チャネルが 8 ビット転送の使用をネゴシエートします (可能な場合)。
eightstrict	ヘッダーに不正な 8 ビットデータが含まれている場合は、チャネルがそのメッセージを拒否するように指定します。
sevenbit	チャネルは 8 ビット文字をサポートしません。8 ビット文字はエンコードされる必要があります。

表 12-4　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します
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します。

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 キーワードでは、LF なしの CR のみを使用できます。これらのオプションは、着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、smtp_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 サーバー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ー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、サーバー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、サーバーが 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ーもあります。

EHLO コマンドを受け取ったサーバー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ー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 サーバー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、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 サーバー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 サーバーに要求できます。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ー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 コマンド行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 サーバーが ETRN コマンドをサポートする場合、サーバーは指定のシステムに別途接続し、そのシステム宛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 はドメインと一致し、MTA が実行しようとする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 サーバーで、ETRN はサポートされているコマンドとして通知されません。

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 サーバーが 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 キーワードのあとには、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ー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 サーバー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。

サーバーは、ユーザー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ー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。サーバーの応答には、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、サーバーの中には、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バウンスし、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 サーバー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 サーバーを介して処理されるすべての着信 TCP/IP チャネルに適用されます。

EXPN サポート

キーワード : expnallow、expndisable、expndefault

expnallow は、DISABLE_EXPAND SMTP チャネルオプションによって SMTP サーバーレベルで EXPN が無効にされている場合でも、EXPN を許可します。expndisable は、EXPN を無条件で無効にします。expndefault は、SMTP サーバーの設定で EXPN が許可されていれば EXPN を許可します (デフォルト)。リストごとに展開を無効にすることができますが、サーバーレベルで無効にされている場合は、リストごとの設定は意味を持ちません。

DNS ドメイン確認

キーワード : mailfromdnsverify、nomailfromdnsverify

mailfromdnsverify を着信 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のメール (UBE) が送られる場合は、この確認を行なった方がよい場合もあります。

COM および ORG トップレベルドメインの DNS ワイルドカードエントリの導入によって mailfromdnsverify が有用でなくなったため、mailfromdnsverify コードは変更されました。DNS が 1 つまたは複数の A レコードを返すと、これらの値と新しい MTA オプション BLOCKED_MAIL_FROM_IPS によって指定されたドメインリテラルとが比較されます。一致する値が見つかると、ドメインは無効とみなされます。正しい動作を復元するための、現在の正しい設定は以下のとおりです。

```
BLOCKED_MAIL_FROM_IPS=[64.94.110.11]
```

このオプションの値はデフォルトでは空の文字列です。

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仕組みを提供します。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ー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仕組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)。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付けされます。

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、MIME-version: ヘッダー行がない場合は、そのヘッダー行が追加されます。

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 サーバーの中には、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 サーバーが 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の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、ヘッダーに不正な 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。デフォルトは 0 です。

TCP/IP 接続と DNS 検索のサポート

サーバーによる 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。

表 12-5 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一覧を示します。

表 12-5　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します**
port	SMTP 接続用のデフォルトポート番号を指定します。標準ポートは 25 です。
interfaceaddress	指定された TCP/IP インタフェースアドレスにバインドします。
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します**
cacheeverything	すべての接続情報をキャッシュします。
cachefailures	接続失敗に関する情報だけをキャッシュします。
cachesuccesses	接続成功に関する情報だけをキャッシュします。
nocache	接続情報をキャッシュしません。
リバース DNS 検索	**着信 SMTP 接続に対するリバース DNS 検索の処理方法を指定します**
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します。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ます。
forwardchecknone	DNS リバース検索のあとに正引き検索を実行しません。

表 12-5　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
forwardchecktag	リバース DNS 検索を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します。一致しなければ名前に「*」を付けます。
IDENT 検索 /DNS リバース検索	**着信 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します**
identnone	IDENT 検索を実行しません。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ます。
identnonelimited	IDENT 検索を実行しません。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ます。
identnonenumeric	IDENT 検索および IP からホスト名への変換を実行しません。
identnonesymbolic	IDENT 検索を実行しません。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ます。
identtcp	着信 SMTP 接続での IDENT 検索および 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ます
identtcplimited	着信 SMTP 接続での IDENT 検索と IP からホスト名への変換を実行します (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 アドレスを含めます。
indenttcpnumeric	着信 SMTP 接続で IDENT 検索を実行します。IP からホスト名への変換を実行しません。
identtcpsymbolic	着信 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ます。
MX レコードのサポートと TCP/IP ネームサーバー	**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します**
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します。
nomx	TCP/IP ネットワークが MX 検索をサポートしません。
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します。
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します。
nonrandomemx	MX 検索を実行しますが、返されたエントリを同等の優先度でランダム化しません。
nameservers	TCP/IP スタックが選択したネームサーバーの代わりに照合するネームサーバーのリストを指定します。nameservers には、空白文字で区切られたネームサーバーの IP アドレスのリストが必要です。
defaultnameservers	TCP/IP スタックが選択したネームサーバーを照合します。

表 12-5　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
lastresort	最後のホストを指定します。
switch キーワード	**メールを着信する代替チャネルのリストを制御します**
allowswitchchannel	switchchannel チャネルからこのチャネルへの切り替えを許可します
noswitchchannel	サーバー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ません。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ません。
switchchannel	サーバー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ます。
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ます。
saslswitchchannel	SASL 認証が成功した場合にほかのチャネルへ切り替えます。
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します。**
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します。
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します。
single_sys	各宛先システム用にメッセージのコピーを 1 つずつ作成します。

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 が待機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 チャネルが待機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る。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ない)。

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、検出された番号がオリジナル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実行し、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。この場合、MTA はオリジナルの IP アドレスを使用します。

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、正引きの結果がオリジナルの 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ーの 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 は着信した 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 は着信した 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) を挿入します。IP 番号そのものは Received: ヘッダーに挿入されません。
- identtcpnumeric は着信した 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 サーバー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 分ほどかかる)。

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、よりユーザー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 番号とホスト名の両方が含まれます。

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: ヘッダーにユーザーフレンドリーではない情報を使用するため、パフォーマンスの向上につながる可能性もあります。これがデフォルトです。

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、identtcplimited キーワードは identtcp と、identnonelimited キーワードは identnone とそれぞれ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 (メールの転送) 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。

ネームサーバー検索

キーワード : nameservers、defaultnameservers

ネームサーバー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ーの代わりに nameservers チャネルキーワードを使ってネームサーバー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ーの 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ーが使用されます。

UNIX でネームサーバー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。次に例を示します。

```
tcp_local single_sys smtp mx lastresort mailhub.siroe.comTCP-DAEMON
```

着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。366 ページの saslswitchchannel および 369 ページの tlsswitchchannel も参照してください。

次の各キーワードは、着信メール用代替チャネルの選択を制御するものです。switchchannel、allowswitchchannel、noswitchchannel。

MTA がリモートシステムから着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 を介する着信 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、この限りではありません。この場合、着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。サーバー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着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、サーバー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、チャネルがファイア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。正規ホストは、チャネルに関連付けられている完全修飾ホスト名です。これは、3 行からなるチャネルブロックの 2 行目に指定できます。

```
tcp_firewall smtp mx daemon router
firewall.acme.com
TCP-DAEMON
```

正規ホストを 2 行からなるチャネルブロックの TCP-DAEMON のあとに指定して、送信接続がそれぞれの接続を特定のホストとして識別するようにもできます。

```
tcp_firewall smtp mx daemon router
TCP-DAEMON firewall.acme.com
```

ファイア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 Security Layer) を使用した SMTP サーバー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 19 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, switchchannel、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 サーバーは、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 サーバーは、クライアントが SASL 認証を使用することを要求します。SMTP サーバーは、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着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

authrewrite チャネルキーワードと関連の AUTH_REWRITE マッピングテーブルを使用すると、認証動作で取得したアドレス情報に基づいてヘッダーとエンベロープアドレスを変更することができます。特に、SASL 認証で、認証された電子メールアドレスが提供されるように設定することができます。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 12-6 にあるように、authrewrite キーワードは必須のビット値をとります。

表 12-6　authrewrite のビット値

ビット	値	説明
0	1	何も変更しないでください (デフォルト)
1	2	認証動作によって提供されたアドレスを含む、Sender: ヘッダーフィールドを追加します。ほかの Resent- フィールドが存在する場合は、Resent-variant が使用されます。
2	4	認証動作によって提供されたアドレスを含む、Sender: ヘッダーフィールドを追加します。

表 12-6　authrewrite のビット値 (続き)

ビット	値	説明
3	8	以下のような形式の AUTH_REWRITE というマッピングテーブルプローブを作成します。 *mail-from* | *sender* | *from* | *auth-sender* *mail-from* はエンベロープ From: アドレス、*sender* は Sender: または Resent-sender: ヘッダーフィールドのアドレス、*from* は From: または Resent-From: ヘッダーフィールドのアドレス、*auth-sender* は認証動作によって提供されたアドレス。 この結果は、AUTH_REWRITE マッピングを使用して実行されます。マッピングでは、縦棒文字 (|) で区切られた項目の一覧が返されます。項目は、次のフラグ設定に基づいて順番に消費されます。 $J $K　メッセージのエンベロープ From: アドレスを置き換えます $Y $T　適切な Sender: または Resent-sender: ヘッダーフィールドを追加します。 $N　メッセージを拒否します。エラーメッセージのテキストはマッピングの結果によって指定されます。テキストがなにも指定されていない場合は、invalid originator address used (無効な差出人アドレス) というエラーメッセージが表示されます。 $Z 適切な From: または Resent-from: ヘッダーフィールドを追加します (**注** : 一般に、From: フィールドを無効にするべきではない)。 ほかの Resent- フィールドがヘッダー内に存在する場合、Resent-variants が使用されます。
4	16	認証によって認証済みアドレスが提供されていない場合でも、AUTH_REWRITE マッピングを適用します。このビットがクリアされている場合は、認証済みアドレスが利用可能なときだけマッピングが適用されます。
5	32	AUTH_REWRITE マッピングプローブの先頭にソースチャネルを含めます。ほかの情報とは | で区切られています。このビットがクリアされている場合、チャネルは含められません。

警告　エンベロープおよびヘッダーアドレスの変更はほとんどの場合に正しく行われないため、$Z フラグは厳しく制限する必要があります。

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 に対応した (maysaslserver キーワード、または mustsaslserver キーワードを使用する) 着信 TCP/IP チャネルに配置されると、MTA の SMTP サーバーが、「誤った」形式 (オリジナルの ESMTP_AUTH 仕様に基づくもの。この仕様は、新しく適切な AUTH 仕様ではなく、適切な ESMTP 形式とは互換性がない) の AUTH を通知することになります。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 コマンドを通知 (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 サーバーは着信接続時に TLS の使用を許可しない。SMTP サーバーもコマンド自体も 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 サーバー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 サーバーが STARTTLS 拡張をサポートすることを通知し、メッセージを着信する際に TLS を使用できる) を包括しています。

TLS が機能するためには、次の条件が整っている必要があります。

- mailsrv アカウントでファイルにアクセスできるように、証明書の保護と所有権が設定されている。
- 証明書が保存されているディレクトリに、mailsrv アカウントでその中のファイルにアクセスできるような保護と所有権が設定されている。

musttls キーワードを指定すると、MTA は送着信接続に必ず 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 サーバー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 サーバーが 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、着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

サーバー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。

メッセージの処理と配信の詳細については、199 ページの「ジョブコントローラ」および 257 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 12-7 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 12-7　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します。**
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します。
チャネルの方向性	**チャネルを処理するプログラムの種類を指定します**
bidirectional	チャネルは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ます。
master	チャネルはマスタープログラム (master) によって処理されます。
slave	チャネルはスレーブプログラム (slave) によって処理されます。
遅延配信	**遅延ジョブの配信に関する設定を定義します。**
backoff	遅延メッセージの配信試行頻度を指定します。normalbackoff、nonurgentbackoff、urgentbackoff で置き換え可能です。
deferred	Deferred-delivery: ヘッダー行の認識と処理を行います。
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します。
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します。**
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) に設定します。該当するメッセージは次の定期ジョブまで処理されません。
normalblocklimit	指定値以上のサイズを持つメッセージの優先度を「低」に設定します。
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」に設定します。

表 12-7　メッセージの処理と配信のキーワード

キーワード	定義
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します。**
pool	チャネルが動作するプールを指定します。
after	チャネルが動作するまでの遅延時間を指定します。
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の最大数を指定します。**
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します。
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します。
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します**
expandlimit	アドレスの数がこの制限を超えた場合、着信メッセージを「オフライン」で処理します。
expandchannel	expandlimit の適用による遅延拡張を実行するチャネルを指定します。
holdlimit	アドレスの数がこの制限を越えた場合、着信メッセージを保留します。
トランザクションの制限	**接続トランザクションの制限を指定します**
transactionlimit	1 つの接続について許されるメッセージの数を制限します。
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します。**
notices	通知を送り、メッセージを返すまでの時間を指定します。
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します。
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します。
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します。

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、immnonurgent

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。

immnonurgent キーワードは、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します。

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。

緊急 : 30, 60, 60, 120, 120, 120, 240
標準 : 60, 120, 120, 240, 240, 240, 480
緊急ではない : 120, 240, 240, 480, 480, 480, 960

優先度が「緊急」のメッセージの場合、最初の配信失敗から 30 分後に再度の配信を試み、再配信から 60 分後に次の再配信、その 60 分後に次の再配信、さらに 120 分後に次の再配信が続きます。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、283 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 分後に再度の配信を試み、再配信から 1 時間後 (最初の配信失敗から 1 時間半後) に 2 回目の再配信、その 2 時間後に 3 回目、その 3 時間後に 4 回目、その 4 時間後に 5 回目、その 5 時間後に 6 回目、その 8 時間後に 7 回目、その 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。配信が失敗すると、配信失敗の通知が生成され、差出人にメッセージが返されます。間隔の構文は ISO 8601P に記述されており、『Sun Java System Messaging Server Administration Reference』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 分後に再度の配信を試み、その 1 時間後に 2 回目の再配信、その 8 時間後に 3 回目、その 1 日後に 4 回目、その 2 日後に 5 回目、その 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。配信が失敗すると、配信失敗の通知が生成され、差出人にメッセージが返されます。

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 分後に再度の配信を試み、その 120 分後に 2 回目の再配信、その 16 時間後に 3 回目、その 36 時間後に 4 回目、その 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。配信が失敗すると、配信失敗の通知が生成され、差出人にメッセージが返されます。

チャネル実行ジョブの処理プール

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 (378 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、257 ページの「ジョブコントローラファイル」、199 ページの「ジョブコントローラ」、および 376 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、257 ページの「ジョブコントローラファイル」、375 ページの「チャネル実行ジョブの処理プール」、および 199 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。

379 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 1 つまたは複数の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。

接続トランザクションの制限を設定する

キーワード : transactionlimit

transactionlimit は、1 つの接続について許されるメッセージの数を制限します。これを使用して、次のように攻撃を阻止できます。

攻撃者が SMTP を介して接続し、正しい電子メールアドレスを推測しようとして多数の RCPT TO コマンドを送信するとします。このような攻撃は、1 つのトランザクションで許可される無効な RCPT TO の回数を制限することによって阻止できます。これに対抗して攻撃者が複数のトランザクションを使用したとしても、1 つの SMTP セッションで許可されるトランザクション数を transactionlimit で制限できます。攻撃者は複数のセッションを使用することはできますが、多大な負担がかかることになります。接続抑制を使用してさまざまな方法でセッション数を制限することで、ほとんどの場合負担を非常に大きくできます。

ただし、阻止する側にも負担はかかります。受取人制限やトランザクション制限、あるいはその両方にうまく対応できない SMTP クライアントもあります。このようなクライアントのために例外をもうける必要があります。ただし、TCP チャネルオプションは無条件で SMTP サーバーに適用されます。これを解決するには、チャネルキーワードと switchchannel を使って、問題になるエージェントをより制限値の高いチャネルにルーティングします。

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。このキーワードのデフォルト値は 10 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。

基本的に、threaddepth はジョブがどのくらい集中してスケジューリングされるかを制御します。次の 2 つの状況を考えてみましょう。

(1) 標準 (送信) SMTP チャネル

(2) スマートホスト宛転送の SMTP チャネル

ジョブコントローラは、宛先ホストによって特定のチャネルに送信されるメッセージを分類し、それらの宛先ホストのバックログに基づいてメッセージが処理されるようにジョブをスケジューリングします。

(1) の場合は、多数の宛先ホストが存在し、それらのほとんどのバックログは少量になります。多数のスレッドが動作していて、aol、yahoo、hotmail などの大量のトラフィックが存在する宛先ホストを除いて、すべてが良好に機能している必要があります。スレッドの深さが 128 の場合、バックログが 128 に達すると yahoo への配信を行う 2 番目のスレッドだけが取得できます。これは適切な状態とはいえません。

(2) の場合は、1 つの宛先ホストだけが存在し、多数のスレッドがそのホストへの配信を行なっていて、理想的な状態です。あえていうなら、10 というデフォルト値は小さすぎます。

threaddepth キーワードは、チャネルの接続先の SMTP サーバー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 サーバーに接続する 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。

複数アドレスの拡張

キーワード : expandlimit、expandchannel、holdlimit

ほとんどのチャネルでは、複数の受取人アドレスを持つメッセージの転送がサポートされています。ただし、1 つのメッセージに多くの受取人アドレスが指定されていると、メッセージ転送処理に遅延 (オンライン遅延) が生じる場合があります。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、そのチャネルで着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。

アドレスのタイプとルール

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。転送レイヤー (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ルール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、パーセント記号のルールを使用して、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 アドレスルールを使用すると、SMTP エンベロープの転送レイヤーのアドレスでもこれらのルール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 を使用しないでください。

uucp (bangstyle)

bang-style 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 bbang-style アドレスルール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 形式のヘッダードレスルール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、パーセント記号のルールを使用して、書き換える必要があります。

注	メッセージヘッダーで 733 アドレスルールを使用すると、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でください。

header_uucp

UUCP または bang-style のヘッダーアドレス。このキーワードの使用は推奨しません。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ルール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 に指示するものです。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 はデフォルトであり、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、書き換えルール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 書き換えルール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ルール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 がアドレスのすべての明示的ルーティングの短絡化を試行するようにします。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。これがデフォルトです。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 サーバーには、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 つ目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、着信するユーザー 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) を追加する必要があります。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 (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。デフォルトは nodefaulthost です。

switchchannel キーワードは、前出の項目の、364 ページの「着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、着信 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ー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 1 (無効なメッセージを変更せずに通過させる) が使用されます。

表 12-8 missingrecipientpolicy の値

値	動作
0	To: ヘッダー行をエンベロープ To: 受取人で置き換えます。
1	無効なメッセージを変更せずに通過させます。
2	To: ヘッダー行をエンベロープ To: 受取人で置き換えます。
3	単一の Bcc: ヘッダー行をすべてのエンベロープ To: 受取人で置き換えます。
4	To: ヘッダー行にグループのコンストラクタ (たとえば「;」) を生成します。「To: Recipients not specified: ;」のようになります。
5	空白の Bcc: ヘッダー行を生成します。
6	メッセージを拒否します。

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 により着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 マッピングのどちらか存在する方に対して照合し、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。詳細は、267 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ルール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。そのチャネルの着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しないように 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ません (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ため)。

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、着信メッセージの Received: ヘッダー行を作成する際に元のエンベロープの From: アドレスを含めるように MTA に指示します。デフォルトは receivedfrom 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パース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に 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有用ではなく、お勧めもしません。commentstrip キーワードは、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を削除するように、MTA 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。sourcecommenttotal 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有用ではなく、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構文は、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パース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) が抽出されるが、これはヘッダー行を再構築するときに変更したり除外することもできます。

この動作は、personalinc、personalmap、personalomit、および personalstrip キーワードを使用して制御されます。personalinc キーワードは、ヘッダーの個人名を維持するように 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。personalomit キーワードは、すべての個人名を削除するように MTA に指示します。personalstrip キーワードは、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、MTA に指示します。personalmap キーワードは、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、MTA 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、ヘッダーの個人名を維持するように 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。sourcepersonalomit キーワードは、すべての個人名を削除するように MTA に指示します。最後に、sourcepersonalstrip キーワードは、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、MTA に指示します。sourcepersonalmap キーワードは、ソースチャネルを通じて個人名を実行するように 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。

PERSONAL_NAMES マッピングテーブルプローブの構文は、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後、「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後、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、ユーザー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ルール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ルールのチェックを強制するように指示します。これがデフォルトです。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なった後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後で、本当にチャネルと一致したとみなされるよう、アドレスもエイリアスから展開する必要があります。

viaaliasrequired キーワードは、たとえば、ローカルチャネルで、任意のアカウント (UNIX システム上の任意のネイティブ Berkeley メールボックスなど) への配信を防ぐために使用できます。

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ルール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、メッセージをパースして、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。ヘッダーオプションファイルの形式については、『Sun Java System Messaging Server Administration Reference』の MTA の章を参照してください。

headertrim キーワードは、元のメッセージヘッダーが処理されたあとで、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、その宛先チャネルの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。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 channel 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 (*instance_root*/imta/config/) に保存されます。

X-Envelope-to: ヘッダー行を生成するまたは削除する

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 行に表示できるアドレスの数を制御します。maxheaderchars キーワードは、1 行に表示できる文字数を制御します。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 の折り返しルール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、ユーザー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ー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 つの整数引数を伴います。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る）

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バウンスされます。

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に MTA に指示します。

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み立て

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み立て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部分メッセージはすべて、代わりに再組み立て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。

再組み立てチャネルの保持時間

再組み立てチャネルのキューにあるメッセージは、一定の時間だけ保持されます。最初の非配信通知が送信されるまでの時間の半分が経過すると、メッセージの各部が再組み立てされないまま送信されます。この時間の値を選択すると、再組み立てチャネルのキューにあるメッセージについて非配信通知が送信されなくなります。

notices チャネルキーワードは、非配信通知を送信するまでの時間を指定します。したがって、メッセージを断片のまま送信するまでの保持時間も指定します。notices キーワードの値は、再組み立てのためにメッセージを保持する期間の 2 倍に設定してください。たとえば、notices の値を 4 にすると、メッセージの断片は 2 日間保持されます。

```
defragment notices 4
DEFRAGMENT-DAEMON
```

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み立てされるように固有 ID の引数が付け加えられます。

maxblocks と 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 の積をヘッダーのサイズとしてとります (実際のヘッダーサイズと maxblocks * MAX_HEADER_BLOCK_USE のどちらか小さい方がヘッダーサイズとみなされる)。

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仕組みを提供します。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によって転送用にソフト改行が実行されます。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。

メッセージの制限、制限容量、受取人、認証の試行

この節では、メッセージのサイズ制限、ユーザー制限容量、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。

認証の試行失敗回数の制限

キーワード : `disconnectbadauthlimit`

このキーワードを使用すると、1 つのセッションで許可される認証の試行失敗の回数を制限できます。この回数に達するとセッションの接続は切断されます。このオプションのデフォルト値は 3 です。

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、着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。

ソースブロック制限を、差出人ごとに指定することもできます。MTA オプション LDAP_SOURCEBLOCKLIMIT を使用してユーザー LDAP 属性を指定し、この属性を差出人の LDAP エントリに追加します。ソースブロック制限は差出人のドメイン単位でも指定できます。この場合は、MTA オプション LDAP_DOMAIN_ATTR_SOURCEBLOCKLIMIT を使用してドメイン LDAP 属性を指定し、この属性を差出人のドメイン LDAP エントリに追加します。上記のどちらの場合にも、デフォルト値は設定されていません。

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。blocklimit キーワードと linelimit キーワードは、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 サーバーは、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。

サイズまたは受取人数の制限を超えるメッセージを再ターゲット化する

キーワード: alternatechannel、alternateblocklimit、alternatelinelimit、alternaterecipientlimit

MTA では、受取人数、サイズ、または行数の指定制限を超えるメッセージを別の宛先チャネルに再ターゲットできます。これは alternatechannel、alternateblocklimit、alternatelinelimit、および alternaterecipientlimit のチャネルキーワードのセットで実装されます。これらのキーワードは、任意の宛先チャネルに指定できます。alternatechannel キーワードは、使用する代替チャネルの名前を指定する単一の引数をとります。これ以外のキーワードはそれぞれ、対応するしきい値を指定する整数の引数を受け入れます。これらのしきい値のいずれかを超えるメッセージは、元の宛先チャネルではなく、代替チャネルのキューに保管されます。

次のチャネルブロックの例では、tcp_local チャネルからインターネットに送信されるはずだった 5,000 ブロックを超える大きなメッセージが tcp_big チャネルから送信されています。

```
tcp_local smtp ... rest of keywords ...   alternatechannel tcp_big
alternateblocklimit 5
tcp-daemon


tcp_big smtp ...rest of keywords...
tcp-big-daemon
```

次に、alternate* チャネルキーワードの使用例を示します。

- 大きなメッセージを後でまたは時間外に配信する場合は、alternatechannel (たとえば tcp_big) を実行する時間が指定できます。

  その方法の 1 つは、imsimta qm ユーティリティの STOP channel_name コマンドおよび START channel_name コマンドを使用することです。ジョブコントローラによって実行されるカスタムな定期的ジョブまたは cron ジョブを介して、これらのコマンドを定期的に実行します。

- ジョブコントローラで大きなメッセージや受取人の多いメッセージを専用のプールで処理する場合は、alternatechannel も使用できます。

  小さなメッセージや受取人の少ないメッセージは、大きなメッセージや受取人の多いメッセージと分離できます (後者はリモート SMTP サーバーでの処理と受け入れに時間がかかることがあるため)。大きなメッセージのせいで小さなメッセージの配信が遅れるのを避けたい場合は分離します。

  ジョブコントローラによる通常のメッセージスケジュールおよびスレッドやプロセスへのメッセージの割り当ては、ほとんどの構成で受け入れられます。

- 大きなメッセージや受取人の多いメッセージに対して TCP/IP チャネルのタイムアウト値を特別に指定する場合は、alternatechannel を使用できます。

  特に、TCP/IP チャネルのタイムアウト値を設定すると、大きなメッセージや受取人の多いメッセージを受信するのに非常に長い時間を要するリモートホストにメッセージを送信する場合に役立ちます。

  ただし、ほとんどの構成にはデフォルトの自動のタイムアウト調整で十分です。デフォルト値を調整することはあっても、特別なチャネルを使用する必要はありません。詳細については、『Messaging Server Reference Manual』で STATUS_DATA_RECV_PER_ADDR_TIME および STATUS_DATA_RECV_PER_BLOCK_TIME の各チャネルオプションを参照してください。

- 特に大きなメッセージに対して MIME メッセージの断片化を特別に設定する場合は、alternatechannel および alternateblocklimit チャネルキーワードを maxblocks チャネルキーワードとともに使用できます。

  指定したサイズを超えるメッセージを断片化する場合は、通常、指定したい maxblocks サイズを通常使用する送信 TCP/IP チャネルに設定します。maxblocks チャネルキーワードは、通常、断片化が実行されるしきい値でもあり、各断片のサイズでもあります。

  しかし、しきい値トリガーを大きくし、実際の断片を小さくする場合は、送信 TCP/IP チャネルに対して alternatechannel および alternateblocklimit を使用できます。その後、代替チャネルに対して maxblock サイズを使用し、指定サイズを超えたメッセージを断片化できます。

- alternatechannel を特別なフィルタ処理とともに使用することができます。たとえば、受取人が多いメッセージがスパムである可能性に備えてより慎重な検査が必要な場合です。送信チャネルに基づいて、別のフィルタ処理を行うことができます (『Sun Java System Messaging Server Administration Reference』の destinationfilter チャネルキーワードを参照)。

  変換チャネルを介して比較的多くのリソースを必要とするスキャン (ウィルスフィルタ処理など) を実行している場合、非常に大きなメッセージによってリソース問題が生じる可能性があります。この場合は、代替変換チャネルを使用できます。または、送信チャネルに基づいて、通常の変換チャネル内で特別な変換処理を行います。

- 大きな送信メッセージを専用のチャネルから送信する場合は、alternatechannel を使用できます。これを使用すると、mail.log* ファイルやカウンタ表示を分析したときに、大きな送信メッセージが見つけやすくなります。

  さらに、大きなメッセージを専用のチャネルで処理すると、配信統計を慎重に分析する場合に役立ちます。リモート SMTP ホストに送信された大きなメッセージや受取人の多いメッセージは処理に時間がかかるため、標準メッセージとは別の配信統計が作成されるからです。

制限容量超過ユーザー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ー (UNIX) 宛のメッセージの処理を制御します。ここでいうユーザーとは、ディスク制限容量を超過していて、ネイティブチャネルのユーザー ID に配信されたユーザーを指します。

noexquota は MTA に、制限容量を超過したユーザー宛のメッセージを、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ー宛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。

1000 文字を超える行を含む SMTP メールを処理する

キーワード : rejectsmtplonglines、wrapsmtplonglines、truncatesmtplonglines

rejectsmtplonglines は、SMTP で許可されている 1000 文字 (CRLF を含む) を超える行を含むメッセージを拒否するオプションを追加します。この領域のほかのオプションは wrapsmtplonglines と truncatesmtplonglines です。wrapsmtplonglines は長すぎる行を折り返し、truncatesmtplonglines は長すぎる行を切り捨てます。デフォルトは truncatesmtplonglines です。これらのキーワードは両方とも、送信に最初に使用されたチャネル (tcp_local など) に適用される必要があります。その後切り替えられるチャネルには影響はありません。

General Content-type、Filename Content-type、および Content-disposition パラメータの長さを制御する

キーワード : parameterlengthlimit および nameparameterlengthlimit

parameterlengthlimit は、general content-type および content-disposition パラメータが切り捨てられる位置を制御します。デフォルト値は 1024 です。nameparameterlengthlimit は、name content-type および filename content-disposition パラメータが切り捨てられる位置を制御します。デフォルト値は 128 です。メッセージ上で MIME 処理が実行されていない場合、もっとも外側のメッセージヘッダーのみが処理されるので注意してください。MIME 処理は、inner キーワードや文字セット変換の使用、そのほかさまざまな方法で有効化できます。

メッセージの受取人を制限する

キーワード : recipientlimit および recipientcutoff

recipientlimit は、メッセージが受け付ける受取人の合計アドレス数を指定します。recipientcutoff は、MTA に対して提示された受取人の合計数を指定した値と比較します。値が制限を超えた場合、配信のためのメッセージは受け付けられません。これらのキーワードは両方とも、1 つの整数引数を受け入れます。対応するチャネルキーワードが設定されていない場合、これらのデフォルトは無限大です。

受取人の制限を、差出人または差出人のドメインに設定することもできます。このためには、適切な MTA オプション (LDAP_RECIPIENTLIMIT、LDAP_RECIPIENTCUTOFF、LDAP_DOMAIN_ATTR_RECIPIENTLIMIT、および LDAP_DOMAIN_ATTR_RECIPIENTCUTOFF) を使用して、ユーザー LDAP またはドメイン LDAP 属性を指定し、その属性を差出人のユーザーエントリまたはドメインエントリに追加します。

ヘッダーのサイズを制限する

キーワード : headerlimit

プライマリ (もっとも外側の) メッセージヘッダーの最大サイズを制限します。この制限に達すると、プライマリメッセージヘッダーはそのまま切り捨てられます。グローバル MTA オプション HEADER_LIMIT が設定されている場合、このチャネルレベルの制限よりも優先されます。デフォルトでは制限なしです。

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。

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 サーバーの中には、一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。

キーワード `multiple`、`addrsperfile`、`single`、`single_sys` 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を制限しないことを意味する。ただし SMTP チャネルのデフォルトは 99)。

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。次に例を示します。

```
tcp_local single_sys smtp subdirs 10
```

セッションの制限を設定する

キーワード : disconnectbadcommandlimit、disconnectrecipientlimit、disconnectrejectlimit、disconnecttransactionlimit

4 つの新しいチャネルキーワードが提供する機能によって、SMTP サーバーは、いくつかのエラーが検出されたあとにクライアントとの接続を切断されます。

disconnectrecipientlimit - セッションの受取人数を制限する。

disconnectrejectlimit - 拒否される受取人数を制限する。

disconnecttransactionlimit - トランザクション数を制限する。

disconnectbadcommandlimit - 不正なコマンド数を制限する。

これらはすべて、セッションの制限です。disconnectbadcommandlimit 以外のすべての制限は、MAIL FROM または RSET コマンドの発行時にチェックされます。制限のどれかを超えた場合、サーバーは 4xy エラーを発行して接続を切断します。disconnectbadcommandlimit は、不正なコマンドの発行時にチェックされるという点だけが異なります。

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`、`logheader`

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。

`logheader` は、チャネル単位で MTA オプション `LOG_HEADER` よりも優先されます。値が 0 の場合 (デフォルト)、メッセージヘッダーのログ記録が無効になります。詳細は、『Sun Java System Messaging Server Administration Reference』を参照してください。

ログ記録については、第 21 章「ログの管理」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。デフォルトでは両方のデバッグ出力とも無効になっています (`nomaster_debug` および `noslave_debug` に設定)。

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。通常、ログファイルはログディレクトリに保存されます。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。この *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、ユーザーが 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 サーバーでは XLOOP 拡張がアドバタイズされます。

XLOOP をサポートする SMTP サーバーと通信する場合、MTA の SMTP クライアントにより、通知された文字列と MTA の値が比較され、クライアントが SMTP サーバーと通信している場合は、メッセージがただちにバウンスされます。

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。

プロセスチャネルのオーバーライド

キーワード : notificationchannel、dispositionchannel

これらのキーワードは、配信ステータス通知 (DSN) および Message Disposition Notification (MDN) をそれぞれ最初にキューに入れるチャネルとして、プロセスチャネルよりも優先されます。指定された名前のチャネルが存在しない場合、Messaging Server はプロセスチャネルの使用を再開します。

notificationchannel は、配信ステータス通知 (DSN) を最初にキューに入れるチャネルとしてプロセスチャネルよりも優先されます。指定された名前のチャネルが存在しない場合、Messaging Server はプロセスチャネルの使用を再開します。

dispositionchannel は、MDN (Message Disposition Notification) を最初にキューに入れるチャネルとしてプロセスチャネルよりも優先されます。指定された名前のチャネルが存在しない場合、Messaging Server はプロセスチャネルの使用を再開します。

チャネル動作のタイプ

キーワード : submit

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 サーバーなどの 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ー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ー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、ユーザー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 チャネルおよび LMTP 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。

LMTP チャネルの場合、通常の使用方法は以下のとおりです。

```
fileinto @$4O:$U+$S@$D
```

この $4O は、数字の 4 と大文字の O です。数字の 0 ではありません。

スパムフィルタのキーワード

キーワード : destinationspamfilterXoptin、sourcespamfilterXoptin

destinationspamfilterXoptin は、このチャネル宛のすべてのメッセージがフィルタリングソフトウェア X (フィルタリングソフトウェア X は option.dat の spamfilterX_library で定義される) を介して実行されることを指定します。キーワードのあとにはフィルタパラメータが続き、使用できるパラメータはフィルタ処理プログラムによって異なります。

sourcespamfilterXoptin は、このチャネルから発信されるすべてのメッセージがフィルタリングソフトウェア X (フィルタリングソフトウェア X は option.dat の spamfilterX_library で定義される) を介して実行されることを指定します。キーワードのあとにはシステム全体のデフォルトパラメータが続き、使用できるパラメータはフィルタ処理プログラムによって異なります。switchchannel が有効な場合、このキーワードは switched-to チャネルに入れられます。

これらのキーワードの使用方法の詳細については、466 ページの「チャネルレベルのフィルタ処理を指定するには」を参照してください。

アドレス検証の後、かつアドレス拡張の前のルーティング

キーワード : aliasdetourhost

aliasdetourhost を使用すると、ホストしているユーザーの mailHost 属性値にソースチャネル固有の優先順位を与えられます。特に、aliasdetourhost は、ある種の処理のために別々のホストに送られるローカル (このシステムでホストされている) ユーザー宛てのメッセージのルーティングにおいて「迂回経路」を確保するために一般的に使用されています。メッセージは元のホストで検証されて (アドレスが正しいローカルアドレスであるかどうか)、迂回経路で処理ホストに送信され、再び元のホストに返送されて拡張および配信されます。

aliasdetourhost は、よりよい設定を実現し、「中間フィルタ」タイプのチャネルおよびサードパーティのフィルタリングホストを使用可能にします。aliasdetourhost は通常、代替変換チャネルに追加して使用されます。aliasdetourhost は、ローカル (このシステムでホストされている) ユーザー宛てのルーティングに適用されますが、代替変換チャネルはリモートの受取人宛てのルーティングに適用されます。

aliasdetourhost の引数は、ホスト名またはドメイン名か、ホストおよびドメインの仕様です (書き換えルールでは、ホスト名、IP リテラルアドレス、およびチャネルタグの処理が可能で、これらは黙示的にホスト名とみなされることに注意)。このキーワードをソースチャネルで指定すると、LDAP に保存されたアドレスのエイリアス展開が、メールホスト情報がチェックされるよりも前に (変換タグ情報の処理のあと) 停止します。このとき、メッセージは aliasdetourhost の値に送信されてアドレスの処理は正常に完了していますが、エイリアス展開は実行されておらず、なおかつアドレス検証は完了済みです。

変換チャネルのフィルタ処理に関するさまざまな問題を回避するための aliasdetourhost の使用例を、以下に示します。システムは、フロントエンド MTA とバックエンドメールストアから構成されているとします。ユーザーは、配信オプションを forward および mailbox に設定しています。MTA は、ウィルス / スパム防止システム用に代替変換チャネルを使用します。このユーザー宛てにメッセージが到着すると、MTA エイリアスは展開され、ローカルとリモートに受取人を 1 人ずつ作成します。リモートの受取人のコピーは直接送信されます。一方、ローカルの受取人のコピーは、変換チャネルに送信されてスキャンされ、返送されます。次に、2 回目のエイリアス展開が行われて、リモートの受取人に対して 2 つ目のコピーが作成され、ローカルの受取人のコピーは通常どおり配信されます。最終結果 : リモートの受取人には 2 つのコピー、ローカルの受取人には 1 つのコピーが送信されます。

ローカルにホストされているユーザーに代替変換チャネルを使用せずに (ただし、場合によっては他の受取人に対して代替変換チャネルを使用して)、aliasdetourhost を使用するチャネルを使うと、次のことが可能になります。

- メッセージの受け入れ
- 外部のスパム / ウィルスフィルタへのメッセージのルーティング
- アドレス拡張および配信のためのメッセージの再受け付け

例 1:

MTA とは別のホストでサードパーティ製のスキャナが動作していると想定します。次の例では、偽の重複エントリを作成しなくてもユーザーエントリの転送が可能であり、なおかつメッセージを受け入れる前に受取人アドレスの検証を実行する機能は保持されています。

1. 新しいチャネル tcp_scanner を作成します。

   作成したチャネルに daemon キーワードを設定し、使用するフィルタ処理システムを指定します。さらに、チャネルに enqueue_removeroute を追加します。tcp_scanner チャネルは、imta.cnf 内で次のようになります。

   ```
   tcp_scanner smtp mx single_sys subdirs 20 noreverse maxjobs 7
   pool
   SMTP_POOL daemon my_a-v_filter.siroe.com enqueue_removeroute
   tcp_scanner-daemon
   ```

2. スキャンするすべてのソース tcp チャネルの tcp_local に aliasDetourHost tcp_scanner-daemon を追加します。スキャンの対象となる tcp チャネルには、tcp_local、tcp_submit、tcp_intranet、tcp_auth などがあります。以下に、tcp_local と tcp_submit. の例を示します。

   ```
   ! tcp_local
   tcp_local smtp mx single_sys remotehost inner switchchannel
   identnonenumeric subdirs 20 maxjobs 7 pool SMTP_POOL maytlsserver
   maysaslserver saslswitchchannel tcp_auth missingrecipientpolicy 0
   aliasdetourhost tcp_scanner-daemon
   tcp-daemon

   ! tcp_submit
   tcp_submit submit smtp mx single_sys mustsaslserver maytlsserver
   missingrecipientpolicy 4 aliasdetourhost tcp_scanner-daemon
   tcp_submit-daemon
   ```

   aliasdetourhost の引数 (tcp_scanner-daemon) は、新しいチャネル tcp_scanner の正規のホスト名です。

3. スキャニングシステムから tcp_scanner チャネルを介して返送されるメッセージを受け入れるように、書き換えルールを作成します。

   ```
   [1.2.3.4] $E$R$U[1.2.3.4]@tcp_scanner-daemon
   ```

   1.2.3.4 は、スキャナシステムの IP アドレスです。

この書き換えルールを使用しない場合、メッセージはほかのいずれかの tcp* ソースチャネルを介して送信されます。すべてのメッセージには aliasdetourhost が含まれるため、メッセージは再びスキャンされて、ループが発生します。

4. 設定をコンパイルしなおし、ディスパッチャを再起動します。

```
#imsimta cnbuild
#imsimta restart dispatcher
```

例 2:

サードパーティ製のスキャナは MTA と同じホストで動作しているが、異なるポートで待機していると想定します。ポート 10024 でメールが受け入れられ、10025 に返送されるとします。

1. 新しいチャネル tcp_scanner を作成します。

```
! tcp_scanner
tcp_scanner smtp nomx single_sys identnonenumeric subdirs 20 maxjobs 7 pool
SCAN_POOL daemon 127.0.0.1 port 10024 enqueue_removeroute
tcp_scanner-daemon
```

2. スキャンするすべてのソース tcp チャネルの tcp_local に aliasDetourHost tcp_scanner-daemon を追加します。スキャンの対象となる tcp チャネルには、tcp_local、tcp_submit、tcp_intranet などがあります。以下に、tcp_local と tcp_submit の例を示します。

```
! tcp_local
tcp_local smtp mx single_sys remotehost inner switchchannel
identnonenumeric subdirs 20 maxjobs 7 pool SMTP_POOL maytlsserver
maysaslserver saslswitchchannel tcp_auth missingrecipientpolicy 0
aliasdetourhost tcp_scanner-daemon
tcp-daemon

! tcp_submit
tcp_submit submit smtp mx single_sys mustsaslserver maytlsserver
missingrecipientpolicy 4 aliasdetourhost tcp_scanner-daemon
tcp_submit-daemon
```

3. mappings ファイルに以下のように追加して、tcp_scanner チャネルを経由して送信メールを再ルーティングします。

```
CONVERSIONS

in-chan=tcp_scanner;out-chan=*;CONVERT          No
in-chan=tcp_*;out-chan=tcp_local;CONVERT    Yes,Channel=tcp_scanner
```

4. SMTP_POOL の下の job_controller.cnf に、同時スキャン数の制限を追加します。

   スキャニングソフトウェアにも制限が必要ですが、同時スキャン数と同じに設定して、メッセージングサーバーがメッセージを受け入れない場合にスキャナへのメール送信を試みないようにすることをお勧めします。

   ```
   !
   [POOL=SCAN_POOL]
   job_limit=2
   !
   ```

5. dispatcher.cnf に新しいサービスを追加して、特別なポートのスキャナから返されたメールを受け入れ、再びスキャンしないように tcp_scan を経由させて送信します。

   ```
   !
   [SERVICE=SMTP_SCANNING]
   INTERFACE_ADDRESS=127.0.0.1
   PORT=10025
   IMAGE=IMTA_BIN:tcp_smtp_server
   LOGFILE=IMTA_LOG:tcp_smtp_server.log
   STACKSIZE=2048000
   PARAMETER=CHANNEL=tcp_scanner
   !
   ```

6. 設定をコンパイルしなおし、ディスパッチャを再起動します。

   ```
   #imsimta cnbuild
   # imsimta restart job_controller
   # imsimta restart dispatcher
   ```

非請求の SMTP 拡張のサポート

キーワード : sourcenosolicit、destinationnosolicit

Internet Draft の draft-malamud-no-soliciting-07.txt に記述されている NO-SOLICIT SMTP 拡張は、プロポーズドスタンダードとして Messaging Server に実装されています。この機能を制御するためには、次のチャネルキーワードが使用できます。

sourcenosolicit は、このチャネルから送信されたメール内でブロックされる請求フィールドの値のコンマ区切りの一覧を指定します。値の一覧は、NO-SOLICIT EHLO 応答に表示されます。値にはグローバルな形式のワイルドカードを使用できますが、ワイルドカードを含む値は EHLO 通知には表示されません。

destinationnosolicit は、このチャネルのキューに入れられたメール内で受け入れられない請求フィールドの値のコンマ区切りの一覧を指定します。

不正な RCPT TO: アドレスに制限を設定する

キーワード: deferralrejectlimit

1 つのセッション中に許可される不正な RCPT TO: アドレス数を制限します。指定した数の To: アドレスが拒否されると、続くすべての受取人は、正しいものも不正なものも 4xx エラーとして拒否されます。ALLOW_REJECTIONS_BEFORE_DEFERRAL SMTP チャネルキーワードと同じ機能が提供されますが、この場合はチャネル単位です。

第 13 章

定義済みチャネルを使用する

チャネルによっては 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 13-1 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 10 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ルールを設定する方法については、第 11 章「書き換えルールの設定」を参照してください。

この章には、以下の節があります。

- 423 ページの「Pipe 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」
- 425 ページの「ネイティブ (/var/mail) チャネルを設定するには」
- 426 ページの「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」
- 427 ページの「変換チャネル」
- 448 ページの「文字セット変換とメッセージの再フォーマット」

defaults チャネルについては、346 ページの「チャネルのデフォルトを設定する」を参照してください。

表 13-1 定義済みチャネル

チャネル	定義
defaults	各種チャネルにデフォルトのキーワードを指定するために使用します。346 ページの「チャネルのデフォルトを設定する」を参照してください。
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します。
ims-ms	メールをローカルストアに配信します。

表 13-1　定義済みチャネル (続き)

チャネル	定義
native	UNIX 専用。/var/mail にメールを配信します (Messaging Server では、/var/mail アクセスはサポートされない。ユーザーが /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)。
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します。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します。
reprocess 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ません。process チャネルは、ほかの MTA チャネルと同様、公にされます。
defragment	断片化された MIME メッセージの修復方法を提供します。
conversion	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します。
bitbucket	破棄するメッセージに使用します。
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ーへのメッセージの処理に使用します。通常、受信したメッセージをバウンスし、カスタムバウンスメッセージを差出人に送り返します。
hold	ユーザーへのメッセージを保留します。ユーザーがあるメールサーバーから別のサーバーに移行された場合などに使用します。
sms	SMS ゲートウェイへの片方向電子メールをサポートします。

表 13-1　定義済みチャネル (続き)

チャネル	定義
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 サーバーが含まれます。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。 tcp_local はリモート SMTP ホストからのメールを受信します。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ります。 tcp_intranet はイントラネット内のメールを送受信します。 tcp_auth は tcp_local のスイッチチャネルとして使用されます。認証されたユーザーは、リレーブロックの制約を回避するため tcp_auth チャネルに移されます。 tcp_submit は、送信されたメッセージ (通常の場合はユーザー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 で受け入れます (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す。

Pipe 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着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、着信メールをメールソートプログラム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ー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 チャネルで呼び出すことができ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 チャネルで呼び出すことができるように登録する各コマンドに一意の名前が設定されます。これによってエンドユーザーが mailprogramdeliveryinfo LDAP 属性の値としてメソッド名を指定できるようになります。

たとえば、UNIX の myprocmail コマンドをユーザー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ーとして実行する 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 *msg_svr_base*/data/site-programs に実行ファイルが存在することを確認してください。また、実行権限が「others」に設定されていることも確認してください。

ユーザーがプログラムにアクセスするためには、そのユーザーの 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 ユーティリティの詳細については、『Messaging Server Reference Manual』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、次の終了コードおよびコマンド行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。

コマンド行の引数 : 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、ユーザーが実行するプログラムの場合はユーザー名を、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ー名 + ドメイン名を示します。たとえば、次のコマンド行は 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特徴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、*msg_svr_base*/config/native_option)。

オプションファイルは複数の行で構成されています。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
```

値は、オプションの要件に基づき、文字列または整数のいずれかとなります。

表 13-2 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」構文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)	ユーザーの .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ー 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ー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
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、ユーザー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。 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。

表 13-2　ローカルチャネルのオプション (続き)

オプション	説明
SHELL_TIMEOUT (整数)	.forward を完成するために、チャネルがユーザー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ー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。次に例を示します。 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、ユーザー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。その他の理由によってメッセージが一時保留される可能性もあります。

メッセージが保留される場合、メッセージは、reprocess チャネルに送られる場合と同じ方法で hold チャネル (*msg_svr_base* /queue/hold ディレクトリ内) に送られます。この方法により、エンベロープ To: アドレスは変更されません。メッセージは hold チャネルキュー (*msg-server*/queue/hold ディレクトリ) に *ZZxxx*.HELD ファイルとして書き込まれます。これによって、メッセージはジョブコントローラから見えなくなるため、「保留」されることになります。imsimta qm dir -held コマンドを使用すると、.HELD ファイルの一覧を表示できます。保留メッセージは、imsimta qm -release コマンドを使用して選択および解除できます。解除すると、メッセージ名は *ZZxxx*.00 に変更され、ジョブコントローラに通知が行われます。その後メッセージは hold チャネルと関連付けられているマスタープログラム (reprocess.exe) で処理されます。したがって、メッセージ (および To: アドレス) は、通常の書き換え機能を使用して処理されます。

imsimta qm コマンドの詳細については、『Sun Java System Messaging Server Administration Reference』を参照してください。

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異なる。メッセージには複数の本文部分が含まれることがある。たとえば添付ファイルにも本文部分がある。また、本文部分は MIME ヘッダーによって指定および描写される)。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 (194 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、Messaging Server マニュアルの Web サイトの下部にある Messaging Server のテクニカルノート (http://docs.sun.com/db/coll/S1_MsgTechNotes) 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。

この節には、以下の項があります。

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"
```

注 MIME ヘッダー行は、一般の MIME 以外のヘッダー行 (`To:`、`Subject:`、`From:` など) とは異なります。基本的に、変換チャネルの場合、MIME ヘッダー行は `Content-` という文字列で始まっています。

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 書き換えルール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。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る)。

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 ファイルによって制御されます。このファイルは imta_tailor ファイル (デフォルトの場合 : *msg_svr_base*/conversions) の IMTA_CONVERSION_FILE オプションで指定します。conversions ファイルを構成するエントリはそれぞれ別の行に記述され、1) どの形式の本文部分を 2) 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 ルールに一致しています。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、一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 13-1 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 13-6 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"
```

コード例 13-1 の conversion ファイルで、3 つの指定パラメータに続く 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 13-1) 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または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 13-1 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行引数です。INPUT_FILE および OUTPUT_FILE は、元の本文部分を持つファイルと、プログラムで変換後の本文部分を保存するファイルとを指定する変換チャネル環境パラメータ (433 ページの「変換チャネル環境変数の使い方」を参照) です。

注 エンベロープ発信元および受取人の情報は、外側のメッセージヘッダーを含むファイルが通常の変換エントリによって要求されると、x-envelope-from および x-envelope-to フィールドとして提供されるようになりました。

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。

注	conversions ファイルを変更した場合は、必ず設定をコンパイルしなおしてください (『Sun Java System Messaging Server Administration Reference』の imsimta refresh コマンドを参照)。

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 13-3) 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 (438 ページの表 13-4 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着信ヘッダーでは、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=NAME; parameter-copy-0=*;
  dparameter-symbol-0=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'"
```

着信ヘッダー：

```
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=NAME は、最初の Content-type パラメータの値 (この例では Draft1.doc) が 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=FILENAME は、最初の Content-disposition パラメータの値 (この例では Draft1.doc) が 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 (もっとも外側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 ORIGINAL_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 MIME ヘッダー行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一時ファイルです。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 MIME ヘッダー行が保存されます。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 MIME ヘッダー行が読み取られます。変更できるのは MIME ヘッダー行のみです。MIME 以外の一般のヘッダー行は、変換チャネルで変更できません。

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。437 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="msg_svr_base/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 13-3　変換チャネル環境変数

環境変数	説明
ATTACHMENT_NUMBER	現在の部分の添付ファイル番号。これは、ATTACHMENT-NUMBER 変換の照合パラメータと同じ形式です。
CONVERSION_TAG	アクティブな変換タグの現在のリスト。これは、TAG 変換の照合パラメータに対応します。
INPUT_CHANNEL	メッセージを変換チャネルのキューに入れたチャネル。これは、IN-CHANNEL 変換の照合パラメータに対応します。
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ります。
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ります。
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ります。
OUTPUT_CHANNEL	メッセージの送信先のチャネル。これは、OUT-CHANNEL 変換の照合パラメータに対応します。

表 13-3　変換チャネル環境変数 (続き)

環境変数	説明
OUTPUT_FILE	サイト提供のプログラムがその出力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みます。
OUTPUT_HEADERS	サイト提供のプログラムが封入する部分の 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みます。ファイルには、option=value 行ではなく実際の MIME ヘッダー行が含まれ、最後の行は空白行となります。また、変更できるのは MIME ヘッダー行のみです。MIME 以外の一般のヘッダー行は、変換チャネルで変更できません。
OUTPUT_OPTIONS	サイト提供のプログラムで変換チャネルオプションを読み取るファイル名です。437 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。
PART_NUMBER	現在の部分の番号。これは、PART-NUMBER 変換の照合パラメータと同じ形式です。
PART_SIZE	処理されている部分のバイト数。

メール変換タグ

メール変換タグは、特定の受取人または差出人に関連付けられている特別なタグです。メッセージの配信時には、このタグは変換チャネルプログラムから見えるので、特別な処理に使用することができます。変換タグは LDAP ディレクトリに保存されます。

メール変換タグは、次のように使用できます。管理者は、選択したユーザーに対してメール変換タグの値を harmonica に設定します。次に、管理者は、そのメールを処理する際にタグおよび harmonica の値が存在するかどうかを検出する変換チャネルを設定します。この場合、プログラムはいくつかの任意の機能を実行します。

メール変換タグは、ユーザー単位またはドメイン単位に設定できます。ドメインレベルの受取人の LDAP 属性は MailDomainConversionTag です (MTA オプション LDAP_DOMAIN_ATTR_CONVERSION_TAG によって変更可能)。ユーザーレベルの受取人の LDAP 属性は MailConversionTag です (MTA オプション LDAP_CONVERSION_TAG によって変更可能)。どちらの属性にも、それぞれの値に異なるタグを指定して複数の値を設定することができます。特定の受取人に関連付けられているタグのセットは累積されます。すなわち、ドメインレベルのタグセットとユーザーレベルのタグセットが結合されます。

差出人ベースの変換タグは、MTA オプション LDAP_SOURCE_CONVERSION_TAG および LDAP_DOMAIN_ATTR_SOURCE_CONVERSION_TAG によって設定できます。これらは、ソースアドレスに関連付けられたそれぞれの変換タグごとにユーザーおよびドメインレベルの LDAP 属性を指定します。上のどちらのオプションでも、デフォルトの属性は設定されていません。

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 13-4) 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、このファイルは実行時の一時ファイルとして、情報を渡すために使用されます。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 一時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 *msg_svr_base*/include/deprecated/pmdf_err.h にある pmdf_err.h ファイルごとの PMDF__FORCERETURN ステータスです。このステータスコードは、差出人にメッセージをバウンス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は、440 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 13-4　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 コンテンツの説明。
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ます。
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み取る際に使用するモードです。
STATUS	コンバータの終了ステータス。通常は、変換チャネルの何らかの動作を開始する特殊な指示です。すべての指示のリストは *msg_svr_base*/include/deprecated/pmdf_err.h を参照してください。

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。封入する 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 ORIGINAL_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 ORIGINAL_HEADERS 環境変数で表したファイルの内容を読み取り、封入する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構文は次のとおりです。

'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。次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。

すべての指示のリストは *msg_svr_base*/include/deprecated/pmdf_err.h を参照してください。以下に、変換チャネルでよく使用される指示を示します。

表 13-5　変換チャネルで一般的に使用される特殊な指示

名前	16 進数値	10 進数値
PMDF__FORCEHOLD	0x0A9C86AA	178030250
PMDF__FORCERETURN	0x0A9C857A	178029946
PMDF__FORCEDELETE	0x0A9C8662	178030178

これらの指示の機能について、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 マッピングと変換ルールのセットを使うと、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ルールも含まれています。

```
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'"
```

アラビア語文字セットの自動検出

アラビア語文字セットを自動的に検出するために、新しい *auto_ef* プログラムが追加されました。

変換チャネルから auto_ef プログラムを呼び出して、アラビア語文字セットのテキストメッセージのうち、ラベルのないメッセージまたは不正なラベルのメッセージのほとんどを自動的に検出し、それらにラベルを付けることができます。ラベルのないメッセージまたは間違ったラベルが付いているメッセージは、通常はアラビア語の Yahoo または Hotmail から送信されたものです。

文字セットのラベルが正しく付いていないメッセージは、多くのメールクライアントで正しく表示されません。

auto_ef プログラムは、メッセージに MIME Content-type ヘッダーが含まれている場合には、text/plain コンテンツタイプのメッセージのみを検査して処理します。メッセージに MIME Content-type ヘッダーが含まれていない場合、auto_ef は強制的に text/plain コンテンツタイプを追加します。

このプログラムを有効にするには、次の操作を行う必要があります。

1. *msg_svr_base*/config ディレクトリのマッピングファイルを編集して、選択したソースチャネルと宛先チャネルの変換チャネルを有効にします。インターネットからローカルユーザーに着信するすべてのメールの変換チャネルを有効にするには、次のようなセクションをマッピングファイルに追加します。

   ```
   CONVERSIONS

        IN-CHAN=tcp*;OUT-CHAN=ims-ms;CONVERT    YES
   ```

   IN チャネルと OUT チャネルは、設定によって異なります。リレー MTA に配備する場合は、これらのチャネルを設定に合わせて変更する必要があります。たとえば、次のように指定します。

   ```
   IN-CHAN=tcp*;OUT-CHAN=tcp*;CONVERT YES
   ```

   すべてのチャネルについて有効にする場合は、次のように指定します。

   ```
   IN-CHAN=*;OUT-CHAN=*;CONVERT YES
   ```

2. 次の内容を含む変換ファイルを作成します。Messaging Server ユーザーが所有し、読み取り権限を持つ *msg_svr_base*/config ディレクトリに作成してください。

   ```
   !
   in-channel=*; out-channel=*;
     in-type=text; in-subtype=*;
     parameter-copy-0=*; dparameter-copy-0=*;
     original-header-file=1; override-header-file=1;
     command="msg_svr_base/lib/arabicdetect.sh"
   !
   ```

3. 次のコマンドを使用して MTA 設定をコンパイルします。

   msg_svr_base/sbin/imsimta cnbuild

4. 次のコマンドを使用して再起動します。

   msg_svr_base/sbin/imsimta restart

表 13-6　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します (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ます。
IN-CHAN, IN-CHANNEL	変換用に照合するチャネルを入力します (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ます。
IN-TYPE	変換用に照合する MIME タイプを入力します (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ます。
IN-SUBTYPE	変換用に照合する MIME サブタイプを入力します (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ます。
IN-PARAMETER-NAME-*n*	変換用に照合する MIME Content-Type パラメータ名を入力します。*n*= 0, 1, 2, ...。このパラメータを IN-PARAMETER-VALUE-*n* とともに使用すると、名前と値からパラメータを特定できます。
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します。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ます。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します。本文部分に IN-PARAMETER-VALUE-*n* が指定されていない場合に、このパラメータのテストのデフォルト値として使用されます。
IN-DISPOSITION	変換用に照合する MIME Content-Disposition を入力します。
IN-DPARAMETER-NAME-*n*	変換用に照合する MIME Content-Disposition パラメータ名を入力します。*n* = 0, 1, 2, ...。このパラメータを IN-DPARAMETER-VALUE-*n* とともに使用すると、名前と値からパラメータを特定できます。

表 13-6　変換パラメータ (続き)

パラメータ	説明
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します。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ます。ワイルドカード使用可。
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、このパラメータのテストのデフォルト値として使用されます。
IN-DESCRIPTION	変換用に照合する MIME Content-Description を入力します。
IN-SUBJECT	封入する MESSAGE/RFC822 部分から Subject (件名) を入力します。
TAG	メーリングリスト CONVERSION_TAG パラメータで設定されているタグを入力します。
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	MIME タイプが入力 MIME タイプと異なる場合に、MIME タイプを出力します。
OUT-SUBTYPE	MIME サブタイプが入力サブタイプと異なる場合に、MIME サブタイプを出力します。
OUT-PARAMETER-NAME-*n*	MIME Content-Type パラメータ名を出力します。*n* = 0, 1, 2, ...。
OUT-PARAMETER-VALUE-*n*	OUT-PARAMETER-NAME-*n* に対応する MIME Content-Type パラメータの値を出力します。
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してコピーします。
OUT-DISPOSITION	MIME Content-Description が入力 MIME Content-Disposition と異なる場合に、MIME Content-Disposition を出力します。
OUT-DPARAMETER-NAME-*n*	MIME Content-Disposition パラメータ名を出力します。*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します。

表 13-6　変換パラメータ (続き)

パラメータ	説明
DPARAMETER-COPY-*n*	本文入力部分の Content-Disposition パラメータリストから本文出力部分の Content-Disposition: パラメータリストにコピーする Content-Disposition: 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します。引数にはワイルドカードを使用できます。特に、* という引数を使用すると、元の Content-Disposition: パラメータはすべてコピーされます。
OUT-DESCRIPTION	MIME Content-Description が入力 MIME Content-Description と異なる場合に、MIME Content-Description を出力します。
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 があります。
OUT-ENCODING	メッセージが再組み立てされるときに、変換ファイルに適用するエンコード。
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)	
COMMAND	変換を実行するためのコマンドで、このパラメータは必須です。変換を実行するためのコマンドで、このパラメータは必須です。コマンドが指定されていない場合、このエントリは無視されます。パスの指定には「¥」ではなく「/」を使用します。例 : command="D:/tmp/mybat.bat"
DELETE	0 または 1 に設定します。このフラグが設定されている場合は、メッセージ部分が削除されます (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 ラベルに変更されます。Relabel=0 では何も変更されません。通常、ラベルの変更は間違ったラベルが付いている部分に対して行います (例 : Content-type:application/octet-stream から Content-type:application/msword)。これによってユーザーは、その部分をファイルに保存してプログラムで開かなくても、「ダブルクリック」で開くことができます。
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します。また、ほかの CHARSET-CONVERSION 操作や変換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います。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ます。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ます。パスの指定には「¥」ではなく「/」を使用します。例 : command="D:/tmp/mybat.bat"

表 13-6　変換パラメータ (続き)

パラメータ	説明
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, ...。各 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します。
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, ...。各 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します。IN-PARAMETER-NAME-*n* 句で一致した MIME パラメータ名を変換する変数名を引数とします。
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みます。書き込まない場合もあります。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ます。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ます。
ORIGINAL-HEADER-FILE	0 または 1 に設定します。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 ORIGINAL_HEADERS で表されるファイルに書き込まれます。
OVERRIDE-HEADER-FILE	0 または 1 に設定します。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します。
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ります。
PART-NUMBER	ドット文字を伴った整数で *a. b. c...* のように表示されます。MIME 本文部分の番号を示します。

文字セット変換とメッセージの再フォーマット

この節では、MTA によって内部的に実行される文字セット、書式設定、およびラベル変換について説明します。ただし、この節の例の一部には、DEC VMS や d チャネルなどの古いまたは廃止された技術が使用されています。それらの技術は古いまたは廃止されたものですが、その例は DEC や d チャネルに限定されたものではありません。このような例も、変換技術のしくみを説明するうえでは有効です。今後のリリースでは例を更新する予定です。

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。このテーブルの名前を `CHARSET-CONVERSION` と言います。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。たとえば、日本語版 OpenVMS を実行しているサイトでは、現在インターネットで使用されている ISO-2022 Kanji と DEC Kanji との変換が必要になります。あるいは、多国語文字が非常に多く使用されているために、MIME で使用するよう指定されている ISO-8859-1 文字セットと DEC Multinational Character Set (DEC-MCS) との多少の相違が問題になるような場合にも、これら 2 つの文字セットの間で実際の変換が必要になります。

`CHARSET-CONVERSION` マッピングテーブルは、メッセージのフォーマットを変更するためにも使われます。MIME 以外のいくつかのフォーマットを MIME に変換する機能が提供されます。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ない)。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) の名前、*out-channel* は宛先チャネル (メッセージの送信先) の名前です。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、その結果がコ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 13-7 にキーワードの一覧を示します。

表 13-7　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、変換を実行します。
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します。
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します。
BASE64	MIME エンコードを BASE64 に切り替えます。このキーワードはすでにエンコードされたメッセージ部分のみに適用されます。Content-transfer-encoding によるメッセージ、7BIT または 8bit は、特別なエンコードは不要であるため、この BASE64 オプションによる影響を受けません。
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します。
Block	MacMIME フォーマット部分からデータフォークのみを抽出します。
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」します。
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します。
Level	重複するマルチパートレベルをメッセージから削除します。
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します。
No	変換を無効にします。
QUOTED-PRINTABLE	MIME エンコードを QUOTED-PRINTABLE に切り替えます。
Record,Text	テキスト部分を 80 バイトのところで折り返します。
Record,Text= n	テキスト部分を n バイトのところで折り返します。
RFC1154	メッセージを RFC 1154 フォーマットに変換します。
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とに「フラット化」します。
UUENCODE	MIME エンコードを X-UUENCODE に切り替えます。
Yes	変換を有効にします。

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る)。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、in-char-set は *out-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。

文字セット変換マッピングの機能は拡張されて、以下の機能が新しく追加されました。

- マッピングエントリの出力テンプレートに IN-CHARSET オプションを指定できます。このオプションがある場合、エンコードされた単語で指定されている文字セットよりも優先されます。
- RELABEL-ONLY オプションを指定できます。このオプションは、整数 0 または 1 を受け入れます。このオプションの値が 1 の場合、OUT-CHARSET は IN-CHARSET に置き換わるだけで、ラベルの変更は行いません。
- IN-CHARSET オプションを使用して入力文字セットを * に設定すると、文字セットから適切なラベルを察知できるようになります。

例 : ISO-8859-1 から UTF-8 への変換およびその逆の変換

ローカルで使用している ISO-8859-1 を、インターネットで使用するために UTF-8 に変換する必要があるとします。さらに、インターネットへの接続は tcp_local を通して行われ、内部メッセージの発信と配信は tcp_internal と ims-ms で行われるとします。次に示す CHARSET-CONVERSION テーブルは、このような変換を実現します。

```
CHARSET-CONVERSION

  IN-CHAN=tcp_internal;OUT-CHAN=tcp_local;CONVERT                 Yes
  IN-CHAN=tcp_local;OUT-CHAN=tcp_internal;CONVERT                 Yes
  IN-CHAN=tcp_local;OUT-CHAN=ims-ms;CONVERT                       Yes
  IN-CHAN=*;OUT-CHAN=*;CONVERT                                    No
  IN-CHAN=tcp_internal;OUT-CHAN=tcp_local;IN-CHARSET=ISO-8859-1  OUT-CHARSET=UTF-8
  IN-CHAN=tcp_local;OUT-CHAN=tcp_internal;IN-CHARSET=UTF-8       OUT-CHARSET=ISO-8859-1
  IN-CHAN=tcp_local;OUT-CHAN=ims-ms;IN-CHARSET=UTF-8           OUT-CHARSET=ISO-8859-1
```

例 : EUC-JP から ISO-2022-JP への変換およびその逆の変換

次に示す CHARSET-CONVERSION テーブルは、ローカルで使用している EUC-JP と ISO 2022 ベースの JP コードとの変換を指定しています。

```
CHARSET-CONVERSION

  IN-CHAN=ims-ms;OUT-CHAN=ims-ms;CONVERT                      No
  IN-CHAN=tcp_internal;OUT-CHAN=ims-ms;CONVERT                No
  IN-CHAN=tcp_internal;OUT-CHAN=tcp_internal;CONVERT          No
  IN-CHAN=tcp_internal;OUT-CHAN=*;CONVERT                     Yes
  IN-CHAN=*;OUT-CHAN=ims-ms;CONVERT                           Yes
  IN-CHAN=*;OUT-CHAN=tcp_internal;CONVERT                     Yes
  IN-CHAN=tcp_internal;OUT-CHAN=*;IN-CHARSET=EUC-JP           OUT-CHARSET=ISO-2022-JP
  IN-CHAN=*;OUT-CHAN=ims-ms;IN-CHARSET=ISO-2022-JP            OUT-CHARSET=EUC-JP
  IN-CHAN=*;OUT-CHAN=tcp_internal;IN-CHARSET=ISO-2022-JP    OUT-CHARSET=EUC-JP
```

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着信します。以下の例は、ローカルユーザー宛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
```

ほかのローカルメールシステムに対するチャネルのエントリを追加することもできます。たとえば、tcp_internal チャネルのエントリは次のようになります。

```
CHARSET-CONVERSION

    IN-CHAN=tcp_local;OUT-CHAN=l;CONVERT                  Yes
    IN-CHAN=tcp_local;OUT-CHAN=tcp_internal;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

ユーザー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。conversions ファイルのエントリについては、431 ページの「変換処理を制御するには」を参照してください。

たとえば、次のような CHARSET-CONVERSION テーブルがあるとします。

```
CHARSET-CONVERSION

   IN-CHAN=tcp_local;OUT-CHAN=tcp_internal;CONVERT
            Yes
```

また、次のような MTA conversion ファイルエントリがあるとします。

```
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
```

これらを組み合わせた結果、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 (このラベル付けはより正確であり、元のユーザー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)。このようなラベル変更が特に役

立つのは、MIME-CONTENT-TYPES-TO-MR マッピングテーブルと組み合わせる場合です。このテーブルは、結果として得られた MIME タイプを変換して適切な MRTYPE タグに戻すために使用されます。この処理が最適に動作するためには、正確な MIME ラベル付けが必要です。すべてのコンテンツタイプに application/octet-stream とだけラベル付けされている場合、MIME-CONTENT-TYPES-TO-MR マッピングテーブルでは、これらすべてを 1 種類の MRTYPE に変換することしかできません。

上記の例と、次のような MIME-CONTENT-TYPES-TO-MR マッピングテーブルエントリについて考えます。

```
APPLICATION/POSTSCRIPT           PS
APPLICATION/MSWORD                  MW
```

たとえば、受信時に次のようなラベルが付いているとします。

```
Content-type: application/octet-stream; name=stuff.ps
```

このラベルは次のように変更されます。

```
Content-type: application/postscript
```

その後、PostScript であることをメッセージルーターに知らせる PS という MRTYPE タグに変換されます。

逆方向のラベル変換が役立つ場合もあります。逆方向のラベル変換とは、具体的な MIME 添付ファイルラベルを、一般的なバイナリデータを表す application/octet-stream というラベルに「ダウングレード」することを意味します。特に、具体的な MIME ラベルの「ダウングレード」は、convert_octet_stream チャネルキーワードとの組み合わせで mime_to_x400 チャネル (PMDF-X400) または xapi_local チャネル (PMDF-MB400) で使用されることがよくあります。その目的は、すべての MIME バイナリ添付ファイルを X.400 bodypart 14 形式に強制的に変換することです。

たとえば、次のような CHARSET-CONVERSION マッピングテーブルがあるとします。

```
CHARSET-CONVERSION

    IN-CHAN=*;OUT-CHAN=mime_to_x400*;CONVERT       Yes
```

また、次のような PMDF conversions ファイルエントリがあるとします。

```
out-chan=mime_to_x400*; in-type=application; in-subtype=*;
   out-type=application; out-subtype=octet-stream; relabel=1

out-chan=mime_to_x400*; in-type=audio; in-subtype=*;
   out-type=application; out-subtype=octet-stream; relabel=1

out-chan=mime_to_x400*; in-type=image; in-subtype=*;
```

```
   out-type=application; out-subtype=octet-stream; relabel=1

out-chan=mime_to_x400*; in-type=video; in-subtype=*;
   out-type=application; out-subtype=octet-stream; relabel=1
```

これらを組み合わせた結果、mime_to_x400* チャネルに送られるすべてのメッセージについて、さまざまな特定の MIME 添付ファイルラベルが一般的な application/octet-stream ラベルにダウングレードされます (convert_octet_stream が適用されるようにするため)。

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する)。

たとえば、次の CHARSET-CONVERSION テーブルは、VMS MAIL メールボックスまたは GroupWise ポストオフィスに配信するときは Appledouble フォーマットに変換することと、メッセージルーターチャネルに配信するときは Macbinary フォーマットに変換することを MTA に指示します。

```
CHARSET-CONVERSION

   IN-CHAN=*;OUT-CHAN=l;CONVERT             Appledouble
   IN-CHAN=*;OUT-CHAN=wpo_local;CONVERT     Appledouble
   IN-CHAN=*;OUT-CHAN=tcp_internal;CONVERT     Macbinary
```

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。Macbinary フォーマットへの変換は、すでにいずれかの MacMIME フォーマットになっている部分に適用されます。また、MacMIME フォーマットでない部分には、MIME Content-type: ヘッダーに X-MAC-TYPE パラメータと X-MAC-CREATOR パラメータが含まれている場合だけ適用されます。

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。

たとえば、次のような CHARSET-CONVERSION テーブルがあるとします。

```
CHARSET-CONVERSION

IN-CHAN=bsout_*;OUT-CHAN=*;CONVERT      Yes
```

また、UNIX で次のような MTA conversions ファイルエントリがあるとします。

```
in-chan=bsout_*; in-type=*; in-subtype=*;
service-command="/pmdf/bin/compress.sh compress $INPUT_FILE
$OUTPUT_FILE"
```

これらを組み合わせた結果、BSOUT チャネルから届くすべてのメッセージが圧縮されます。

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渡すためには、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行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。たとえば、INPUT_FILE と OUTPUT_FILE の値が a.in と a.out である場合に、UNIX で次のように宣言したとします。

```
in-chan=bsout_*; in-type=*; in-subtype=*;
    service-command="/pmdf/bin/convert.sh $INPUT_FILE $OUTPUT_FILE"
```

結果として、次のコマンドが実行されます。

```
/pmdf/bin/convert.sh a.in a.out
```

第 14 章

スパムとウィルスのフィルタ処理プログラムを Messaging Server に統合する

この章では、Messaging Server を使用して、スパムおよびウィルスのフィルタ処理ソフトウェアを統合および設定する方法について説明します。この章で説明するスパムおよびウィルスのフィルタ処理技術は、変換チャネルによって提供される技術よりも強力です (427 ページの「変換チャネル」を参照)。Messaging Server では、Symantec Brightmail AntiSpam、SpamAssassin、および Internet Content Adaptation Protocol (ICAP、RFC 3507) がサポートするスパム防止 / ウィルス防止プログラム (Symantec AntiVirus Scan Engine など) がサポートされています。

注 この章のスパム防止またはスパムのフィルタ処理機能についての参照は、該当する場合には、ウィルス防止またはウィルスのフィルタ処理機能にも当てはまります。製品によって、両方の機能を提供するものもあれば (Brightmail)、スパムのフィルタ処理機能のみ (SpamAssassin)、またはウィルスのフィルタ処理機能のみ (Symantec AntiVirus Scan Engine) を提供するものもあります。また、`spam` は設定パラメータで一般的に使用されます。

この章には、以下の節があります。

スパムのフィルタ処理プログラムを Messaging Server に統合する - 動作方式

Messaging Server から見ると、スパム防止ソリューションはどれも同じように動作します。

1. Messaging Server は、スパムのフィルタ処理ソフトウェアにメッセージのコピーを送信します。
2. スパムのフィルタ処理ソフトウェアは、メッセージを分析し、スパムかどうかの判定を返します。プログラムによっては、SpamAssassin のようにスパムスコアを含む判定も返します。スパムスコアは、メッセージがスパムである確率を数値で示すものです。
3. Messaging Server はこの判定を読み取り、メッセージに対して Sieve アクションを実行します (469 ページの「スパムメッセージに対して実行するアクションを指定する」を参照)。

スパムのフィルタ処理プログラムは、プロトコルを介して MTA と対話します。Symantec AntiVirus Scan Engine などの ICAP ベースのプログラムで使用するような標準プロトコル、Brightmail で使用するような独自のプロトコル、または、SpamAssassin で使用するような非標準のプロトコルが使用可能です。各プロトコルには、MTA とのインタフェースのためのソフトウェアのフックが必要です。Brightmail および Spam Assassin は、メッセージングサーバーと最初に統合できるスパムのフィルタ処理プログラムです。MTA は現在、ICAP を使用するプログラムをサポートしています。

サードパーティのスパムのフィルタ処理プログラムを配備および設定する

Messaging Server にサードパーティのフィルタ処理ソフトウェアを配備するには、5 つのアクションが必要です。

- **配備するスパムのフィルタ処理プログラムと、配備先のサーバー数を決定します。** Messaging Server では、最大 4 種類のスパムおよびウィルス防止プログラムを使用して着信メッセージをフィルタ処理できます。これらのプログラムは、個々のシステム上、単一のシステム配備の Messaging Server と同じシステム上、または 2 層配備の MTA と同じシステム上で実行できます。必要なサーバー数は、メッセージ負荷、ハードウェアのパフォーマンス、およびほかの要因によって異なります。サイトでのハードウェア要件を決定するガイドラインについては、スパムのフィルタ処理ソフトウェアのマニュアルを参照するか、販売代理店にお問い合わせください。

- **スパムのフィルタ処理ソフトウェアをインストールして構成します。**この情報については、スパムのフィルタ処理ソフトウェアのマニュアルを参照するか、販売代理店にお問い合わせください。
- **フィルタ処理のクライアントライブラリをロードおよび構成します。**このためには、MTA `option.dat` ファイルにクライアントライブラリと設定ファイルを指定して、さらにフィルタ処理ソフトウェアの設定ファイルに必要なオプションを設定する必要があります。461 ページの「スパムのフィルタ処理ソフトウェアのクライアントライブラリをロードおよび構成する」
- **フィルタ処理を実行するメッセージを指定します。**ユーザー、ドメイン、またはチャネルごとにメッセージをフィルタ処理できます。463 ページの「フィルタ処理を行うメッセージを指定する」
- **スパムに対して行う処置を指定します。**スパムは、破棄したり、フォルダにファイリングしたり、件名にタグ付けしたりできます。469 ページの「スパムメッセージに対して実行するアクションを指定する」

注 以前のバージョンの Messaging Server でサポートされていたのは、Brightmail フィルタ処理技術のみでした。このため、キーワードとオプションには、`sourcebrightmail` または `Brightmail_config_file` のような名前が付けられていました。これらのキーワードとオプションは、`sourcespamfilter` または `spamfilter_config_file` などの、より一般的な名前に変更されています。以前の Brightmail の名前は、互換性のために保持されています。

スパムのフィルタ処理ソフトウェアのクライアントライブラリをロードおよび構成する

それぞれのスパムフィルタ処理プログラムは、Messaging Server のクライアントライブラリファイルと設定ファイルを提供します。クライアントライブラリをロードおよび構成するには、次の 2 つのことが必要です。

- `option.dat` ファイルに、スパムのフィルタ処理ソフトウェアのライブラリパス (`spamfilterX_library`) と設定ファイル (`spamfilterX_config_file`) を指定します。これら以外にも、スパムのフィルタ処理の LDAP 属性やスパムメッセージに実行する Sieve アクションを指定するためのいくつかのオプションがあります。

- スパムのフィルタ処理ソフトウェアの設定ファイルに、必要なオプションを指定します。それぞれのスパムフィルタ処理プログラムには、個々の設定ファイルおよび設定オプションがあります。これらについては、フィルタ処理ソフトウェアの節およびスパムのフィルタ処理ソフトウェアのマニュアルで説明しています。475 ページの「Symantec Brightmail AntiSpam を使用する」または 480 ページの「SpamAssassin を使用する」、および 494 ページの「Symantec Anti-Virus Scanning Engine (SAVSE) を使用する」を参照してください。

スパムのフィルタ処理ソフトウェアのライブラリパスを指定する

Messaging Server は、メッセージに対して最大 4 種類のフィルタ処理システムを呼び出せます。たとえば、Symantec AntiVirus Scan Engine と SpamAssassin の両方を使用してメッセージを実行できます。それぞれのフィルタ処理ソフトウェアは、1 から 4 までの番号で識別されます。これらの番号は、各種のスパムフィルタオプション、LDAP 属性、およびチャネルキーワードの一部として表示され、フィルタの ID 番号として *X* が使用されます。たとえば、`sourcespamfilterXoptin` または `spamfilterX_config_file` のようになります。キーワードまたはオプション名から ID 番号を省略した場合、デフォルトは 1 です。

次の `option.dat` 設定では、Messaging Server が Symantec AntiVirus Scan Engine と SpamAssassin の両方を使用してメッセージをフィルタ処理するように指定しています。

```
spamfilter1_library=Symantec_Library_File
spamfilter1_config_file=Symantec_Config_File
spamfilter2_library=SpamAssassin_Library_File
spamfilter2_config_file=SpamAssassin_Config_File
```

ほかのオプションまたはキーワードを使用してシステムを設定する場合は、オプションまたはキーワードの最後に対応する番号を指定してください。たとえば、`sourcespamfilter2optin` は SpamAssassin を参照します。`sourcespamfilter1optin` は Symantec AntiVirus Scan Engine を参照します。連続した番号を使用する必要はありません。たとえば、Symantec AntiVirus Scan Engine を一時的に無効にするには、`spamfilter1_library` 設定ファイルをコメントアウトするだけすみます。

フィルタ処理を行うメッセージを指定する

スパムのフィルタ処理ソフトウェアのインストールが完了して Messaging Server で使用可能になったら、フィルタ処理を行うメッセージを指定する必要があります。Messaging Server では、ユーザー、ドメイン、またはチャネルごとにフィルタするよう設定できます。それぞれの場合について、次の節で説明します。

- 463 ページの「ユーザーレベルのフィルタ処理を指定するには」
- 465 ページの「ドメインレベルのフィルタ処理を指定するには」
- 466 ページの「チャネルレベルのフィルタ処理を指定するには」

注 *optin* という表現は、ユーザー、ドメイン、またはチャネルのいずれにメールのフィルタ処理を受信させるかを選択することを意味します。

ユーザーレベルのフィルタ処理を指定するには

ユーザーごとにフィルタ処理を指定したほうがよい場合があります。たとえば、スパムまたはウィルスのフィルタ処理を ISP の顧客に対するプレミアムサービスとして提供する場合に、サービスを受けるユーザーと受けないユーザーを指定できます。ユーザーごとのフィルタ処理を行うための一般的な手順は、次のとおりです。

1. スパムのフィルタ処理ソフトウェアを起動するユーザー LDAP 属性を指定します。

   option.dat に LDAP_OPTIN*X* オプションを設定します。次に例を示します。

   ```
   LDAP_OPTIN1=SymantecAV
   LDAP_OPTIN2=SpamAssassin
   ```

2. スパムのフィルタ処理の対象となるユーザーエントリにフィルタ属性を設定します。

   フィルタ属性は複数値を持ち、サーバーによって異なります。手順 1 の例を使用した場合のエントリを以下に示します。

   ```
   SymantecAV:virus
   SpamAssassin:spam
   ```

   ウィルスとスパムの両方をフィルタ処理できる Brightmail のようなプログラムの場合、有効な値は spam および virus です。複数値を持つ属性として使用する場合、それぞれの値に個別の属性を入力する必要があります。たとえば、Brightmail のフィルタ属性が Brightmail に設定されている場合のエントリは以下のとおりです。

   ```
   Brightmail:spam
   Brightmail:virus
   ```

ユーザーレベルのフィルタ処理の例

この例では、Brightmail が使用されていると仮定しています。また、option.dat ファイルで LDAP_OPTIN1 が Brightmail に設定されているとします。Otis Fanning というユーザーには、そのユーザーエントリの Brightmail 属性が spam および virus に設定されています。このユーザーのメールには、Brightmail によってスパムとウィルスのフィルタ処理が行われます。コード例 14-1 は、Otis Fanning に対する Brightmail のユーザーエントリを示しています。

コード例 14-1 Brightmail 用の LDAP ユーザーエントリの例

```
dn:uid=fanning,ou=people,o=sesta.com,o=ISP
objectClass:person
objectClass:organizationalPerson
objectClass:inetOrgPerson
objectClass:inetUser
objectClass:ipUser
objectClass:inetMailUser
objectClass:inetLocalMailRecipient
objectClass:nsManagedPerson
objectClass:userPresenceProfile
cn:Otis Fanning
sn:fanning
initials:OTF
givenName:Otis
pabURI:ldap://ldap.siroe.com:389/ou=fanning,ou=people,o=sesta.com,o=isp,o=pab
mail:Otis.Fanning@sesta.com
mailAlternateAddress:ofanning@sesta.com
mailDeliveryOption:mailbox
mailHost:manatee.siroe.com
uid:fanning
dataSource:iMS 5.0 @(#)ims50users.sh 1.5a 02/3/00
userPassword:password
inetUserStatus:active
mailUserStatus:active
mailQuota: -1
mailMsgQuota: 100
Brightmail:virus
Brightmail:spam
```

Symantec AntiVirus Scan Engine および SpamAssassin が使用された場合、エントリは次のようになります。

```
SymantecAV:virus
SpamAssassin:spam
```

このほかの例と詳細については、「Symantec Brightmail AntiSpam を使用する」、「SpamAssassin を使用する」、または「Symantec Anti-Virus Scanning Engine (SAVSE) を使用する」を参照してください。

ドメインレベルのフィルタ処理を指定するには

フィルタ処理の対象となるドメインを指定できます。この機能の使用例は、スパム防止またはウィルス防止のフィルタ処理を ISP ドメインの顧客に対するプレミアムサービスとして提供する場合です。ドメインのフィルタ処理を指定するための一般的な手順は、次のとおりです。

1. フィルタ処理ソフトウェアを起動するドメイン LDAP 属性を指定します。

   option.dat に LDAP_DOMAIN_ATTR_OPTIN*X* オプションを設定します。次に例を示します。

   ```
   LDAP_DOMAIN_ATTR_OPTIN1=SymantecAV
   LDAP_DOMAIN_ATTR_OPTIN2=SpamAssassin
   ```

2. フィルタ属性をスパムのフィルタ処理の対象となるドメインエントリに設定します。

   フィルタ属性は複数値を持ち、サーバーによって異なります。手順 1 の例を使用した場合のエントリを以下に示します。

   ```
   SymantecAV:virus
   SpamAssassin:spam
   ```

   ウィルスとスパムの両方をフィルタ処理できる Brightmail のようなプログラムの場合、有効な値は spam および virus です。複数値を持つ属性として使用する場合、それぞれの値に個別の属性値を入力する必要があります。たとえば、LDAP_DOMAIN_ATTR_OPTIN1 が Brightmail に設定されている場合のエントリは以下のとおりです。

   ```
   Brightmail:spam
   Brightmail:virus
   ```

ドメインレベルのフィルタ処理の例

この例では、Brightmail が使用されていると仮定しています。また、option.dat ファイルで LDAP_DOMAIN_ATTR_OPTIN1 が Brightmail に設定されているとします。Sun LDAP スキーマ 1 の DC ツリーの sesta.com ドメインエントリでは、Brightmail 属性は spam および virus に設定されています。Sun LDAP スキーマ 2 の場合、ドメインエントリに Brightmail を設定して、スパムのフィルタ処理の対象にします。

sesta.com に送信されるすべてのメールは、Brightmail によってスパムおよびウィルス用にフィルタ処理されます。コード例 14-2 に、ドメインエントリを示します。

コード例 14-2 Brightmail 用の LDAP ドメインエントリの例

```
dn:dc=sesta,dc=com,o=internet
objectClass:domain
objectClass:inetDomain
objectClass:mailDomain
objectClass:nsManagedDomain
```

コード例 14-2　Brightmail 用の LDAP ドメインエントリの例 (続き)

```
objectClass:icsCalendarDomain
description:DC node for sesta.com hosted domain
dc:sesta
inetDomainBaseDN:o=sesta.com,o=isp
inetDomainStatus:active
mailDomainStatus:active
mailDomainAllowedServiceAccess:+imap, pop3, http:*
mailRoutingHosts:manatee.siroe.com
preferredMailHost:manatee.siroe.com
mailDomainDiskQuota: 100000000
mailDomainMsgQuota: -1
mailClientAttachmentQuota: 5
Brightmail:spam
Brightmail:virus
```

Symantec AntiVirus Scan Engine および SpamAssassin が使用された場合、エントリは次のようになります。

```
SymantecAV:virus
SpamAssassin:spam
```

このほかの例と詳細については、「Symantec Brightmail AntiSpam を使用する」、「SpamAssassin を使用する」、または「Symantec Anti-Virus Scanning Engine (SAVSE) を使用する」を参照してください。

チャネルレベルのフィルタ処理を指定するには

ソースチャネルまたは宛先チャネルによるフィルタ処理を行うと、スパムのフィルタ処理の柔軟性と精度が大幅に向上します。たとえば、次のようなフィルタ処理を実行するとします。

- 特定の MTA リレーからバックエンドメッセージストアへのメッセージだけをフィルタ処理します。
- 特定の MTA からのすべての着信メールをフィルタ処理します。
- 特定の MTA からのすべての送信メールをフィルタ処理します。
- 特定の MTA からのすべての送信および着信メールをフィルタ処理します。

Messaging Server により、ソースチャネルまたは宛先チャネルによるフィルタ処理を指定できます。このためのメカニズムは、表 14-1 で説明するチャネルキーワードです。次の例は、チャネルレベルのフィルタ処理の設定方法を示しています。

1. メッセージをバックエンドメッセージストアのホストに送信するすべての SMTP サーバーの `imta.cnf` ファイルに書き換えルールを追加します。次に例を示します。

   ```
   msg_store1.siroe.com    $U@msg_store1.siroe.com
   ```

2. その書き換えルールと対応するチャネルを destinationspamfilterXoptin キーワードを使用して追加します。次に例を示します。

```
tcp_msg_store1 smtp subdirs 20 backoff "pt5m" "pt10" "pt30"
"pt1h" ¥
"pt2h" "pt4h" maxjobs 1 pool IMS_POOL fileinto $U+$S@$D ¥
destinationspamfilter1optin spam
msg_store1.siroe.com
```

表 14-1 スパムフィルタ用の MTA チャネルキーワード

チャネルキーワード	説明
destinationspamfilter*X*optin	これらのサービスがユーザーまたはドメインによって LDAP_OPTIN LDAP 属性で指定されていない場合でも、このチャネル宛のすべてのメッセージがスパム防止ソフトウェア *X* によってフィルタ処理されることを指定します (ソフトウェア X のフィルタ処理は option.dat の spamfilter*X*_library によって定義される)。フィルタパラメータはフィルタ処理プログラムによって異なり、キーワードのあとに続きます。たとえば、Brightmail のパラメータは通常、spam または virus または spam,virus です。SpamAssassin のパラメータは spam です。 この例では、メッセージストア宛のメールはすべて、スパムでないかスキャンされます。 ims-ms destinationspamfilter1optin spam,virus. . .
sourcespamfilter*X*optin	これらのサービスがユーザーまたはドメインによって LDAP_OPTIN LDAP 属性で指定されていない場合でも、このチャネルから発信されるすべてのメッセージがスパム防止ソフトウェア *X* によってフィルタ処理されることを指定します。キーワードのあとにはシステム全体のデフォルトパラメータが続きます。使用できるパラメータはフィルタ処理プログラムによって異なります。たとえば、Brightmail の場合のパラメータは、spam または virus または spam,virus です。SpamAssassin の場合、パラメータは spam です。switchchannel が有効な場合、このキーワードは switched-to チャネルに入れられます。

チャネルレベルのフィルタ処理の例

以下の例では、番号 1 によって指定されるフィルタ処理プログラムを想定しています。

例 1: MTA リレーから msg_store1.siroe.com というバックエンドメッセージストアへのすべてのメールを、スパムおよびウィルス用にフィルタ処理します。

1. メッセージをバックエンドメッセージストアのホストに送信する imta.cnf ファイルに、書き換えルールを追加します。次に例を示します。

msg_store1.siroe.com　$U@msg_store1.siroe.com

2. その書き換えルールと対応するチャネルを destinationspamfilterXoptin キーワードを使用して追加します。次に例を示します。

```
tcp_msg_store1 smtp subdirs 20 backoff "pt5m" "pt10" "pt30" "pt1h" ¥
"pt2h" "pt4h" maxjobs 1 pool IMS_POOL fileinto $U+$S@$D ¥
destinationspamfilter1optin spam,virus
msg_store1.siroe.com
```

例 2: MTA を通過するすべての着信メールをスパム用にフィルタ処理します (通常、すべての着信メッセージは tcp_local チャネルを通過する)。

```
tcp_local smtp mx single_sys remotehost inner switchchannel ¥
identnonelimited subdirs 20 maxjobs 7 pool SMTP_POOL ¥
maytlsserver maysaslserver saslswitchchannel tcp_auth ¥
sourcespamfilter1optin spam
tcp-daemon
```

例 3: MTA を通過するすべてのインターネットへの発信メールをフィルタ処理します (通常、インターネットへのすべての発信メッセージは tcp_local チャネルを通過する)。

```
tcp_local smtp mx single_sys remotehost inner switchchannel ¥
identnonelimited subdirs 20 maxjobs 7 pool SMTP_POOL ¥
maytlsserver maysaslserver saslswitchchannel tcp_auth ¥
destinationspamfilter1optin spam
tcp-daemon
```

例 4: MTA を通過するすべての着信および発信メールをフィルタ処理します。

```
tcp_local smtp mx single_sys remotehost inner switchchannel ¥
identnonelimited subdirs 20 maxjobs 7 pool SMTP_POOL ¥
maytlsserver maysaslserver saslswitchchannel tcp_auth ¥
sourcespamfilter1optin spam destinationspamfilter1optin spam
tcp-daemon
```

例 5: ユーザーの optin を使わずに、2 層システムにあるローカルメッセージストア宛のすべてのメールをフィルタ処理します。

```
ims-ms smtp mx single_sys remotehost inner switchchannel ¥
identnonelimited subdirs 20 maxjobs 7 pool SMTP_POOL ¥
maytlsserver maysaslserver saslswitchchannel tcp_auth ¥
destinationspamfilter1optin spam
tcp-daemon
```

例 6: すべての着信および発信メールをスパムおよびウィルス用にフィルタ処理します (使用するソフトウェアがスパムとウィルスの両方をフィルタ処理することを前提とする)。

```
tcp_local smtp mx single_sys remotehost inner switchchannel ¥
identnonelimited subdirs 20 maxjobs 7 pool SMTP_POOL ¥
maytlsserver maysaslserver saslswitchchannel tcp_auth ¥
destinationspamfilter1optin spam,virus sourcespamfilter1optin ¥
spam,virus
tcp-daemon
```

スパムメッセージに対して実行するアクションを指定する

スパムのフィルタ処理プログラムは、メッセージを分析し、スパムかどうかの判定を Messaging Server に返します。そのあと Messaging Server は、メッセージに応じたアクションを実行します。アクションは、Sieve メールフィルタ処理言語を使って指定します。指定できるアクションは、メッセージの破棄、フォルダへのファイリング、ヘッダーの追加、件名へのタグの追加などです。if-then-else ステートメントを含む複雑な Sieve スクリプトも指定できます。

注 複雑な Sieve 構文については、Sieve 仕様 3028 を参照してください。http://www.cyrusoft.com/sieve/ も参照してください。

Sieve スクリプトは、表 14-2 で説明する MTA スパムフィルタオプション (option.dat) で指定します。主なスパムフィルタアクションのオプションは NULL 値がスパム判定値として返された時に実行される Sieve ルールを指定する SpamfilterX_null_action と、文字列がスパム判定として返された時に実行される Sieve ルールを指定する SpamfilterX_string_action です。

スパムのフィルタ処理プログラムは、通常、文字列または NULL 値を MTA に返して、メッセージがスパムであることを示します。プログラムによってはスパムスコアも返します。スパムスコアは、スパムであるメッセージまたはスパムではないメッセージの確率を示す数値です。このスコアは、アクションシーケンスの一部として使用できます。次の例は、フィルタ処理を行うメッセージに対するアクションの指定方法を示しています。それぞれの例では、番号 1 によって指定されるフィルタ処理プログラムを想定しています。

例 1: NULL 判定値を含むスパムメッセージをファイル SPAM_CAN にファイリングします。

```
spamfilter1_null_action=data:,require "fileinto"; fileinto
"SPAM_CAN";
```

同じアクションを、次の文字列を返すスパムメッセージで実行できます。

```
spamfilter1_string_action=data:,require "fileinto"; fileinto
"SPAM_CAN";
```

例 2: 返された判定文字列を含むスパムメッセージを、その判定文字列のあとに指定されているファイル内にファイリングします。$U がこれを実行します。つまり、返される判定文字列が spam の場合、メッセージは spam というファイルに保存されます。

```
spamfilter1_null_action=data:,require "fileinto"; fileinto "$U";
```

例 3: 文字列判定値を含むスパムメッセージを破棄します。

```
spamfilter1_string_action=data:,discard
```

同じアクションを、NULL 値を返すスパムメッセージで実行できます。

```
spamfilter1_null_action=data:,require "fileinto"; fileinto
"SPAM_CAN";
```

例 4: この行では、文字列判定値によってスパムであると判断された各メッセージに、ヘッダー Spam-test: FAIL を追加しています。

```
spamfilter1_string_action=data:,require ["addheader"];addheader
"Spam-test:FAIL";
```

例 5: この行では、文字列を返すスパムメッセージの件名に、文字列 [PROBABLE SPAM] を追加しています。

```
spamfilter1_string_action=data:,addtag "[PROBABLE SPAM]";
```

例 6: この行では、ヘッダーに resent-from および User-1 がある場合は、文字列判定値が仮定され、スパムメッセージがメールボックス testspam にファイリングされます。メッセージにこのヘッダーがない場合、メッセージは spam にファイリングされます。

```
spamfilter1_string_action=data:,require "fileinto";¥
  if header :contains ["resent-from"] ["User-1"] {¥
  fileinto "testspam";¥
  } else {¥
  fileinto "spam";};
```

判定文字列はほとんどのスパムフィルタソフトウェアで設定できるため、返される文字列に応じてさまざまなアクションを指定できます。これは、対応するペアの spamfilter*X*_verdict_*n* オプションと spamfilter*X*_action_*n* オプションを使って実行できます。

例 7: これらの対応するペアのオプションは、判定文字列 remove が返されたスパムメッセージを破棄します。

```
spamfilter1_verdict_0=remove
spamfilter1_action_0=data:,discard
```

スパム判定文字列の指定方法については、該当のスパムフィルタ処理ソフトウェアの節を参照してください。

表 14-2 MTA スパムフィルタオプション (option.dat)

SpamAssassin 用の MTA オプション	説明
SpamfilterX_config_file	フィルタ処理ソフトウェア X 設定ファイルのフルパスとファイル名を指定します。 デフォルト : なし
SpamfilterX_library	フィルタ処理ソフトウェア X 共有ライブラリのフルパスとファイル名を指定します。 デフォルト : なし
SpamfilterX_optional	フィルタ処理ライブラリ X によって報告された障害を、一時的な処理の失敗として処理するか無視するかを制御します。デフォルト値 0 は、スパムのフィルタ処理の問題により一時的な処理の障害が発生すると指定します。この値を 1 に変更すると、スパムフィルタの処理は場合によってはライブラリ障害のフィルタ処理をスキップしますが、すべてではありません。特に、システムがライブラリコードを返さずにスタックすると、一部の MTA もスタックすることがあります。-2 と 2 も設定できます。これらは、それぞれ 0 および 1 と同じですが、スパムフィルタプラグインによって問題が報告された場合に syslog メッセージが送信される点で異なります。 デフォルト : 0
LDAP_optinX	ユーザー単位でフィルタ処理ソフトウェア X を有効にするために使用する LDAP 属性の名前を指定します。これは、inetMailUser オブジェクトクラス内の属性である必要があります。 属性自体には複数の値を指定でき、大文字小文字は区別されます。SpamAssassin の場合、値は小文字の spam にする必要があります。 デフォルト : なし
LDAP_domain_attr_optinX	ドメイン単位でフィルタ処理ソフトウェア X を有効にするために使用する LDAP 属性の名前を指定します。宛先ドメインに適用されます。オブジェクトクラス mailDomain 内にある必要がある以外は、LDAP_optin と同様です。 デフォルト : なし

表 14-2　MTA スパムフィルタオプション (option.dat) (続き)

SpamAssassin 用の MTA オプション	説明
SpamfilterX_null_optin	LDAP_optinX または LDAP_domain_attr_optinX で定義される属性の値として見つかった場合に、属性がそこにはないかのように MTA を機能させる文字列を指定します。つまり、そのエントリに対するフィルタ処理を無効にします。詳細は、463 ページの「フィルタ処理を行うメッセージを指定する」を参照してください。 デフォルト : 空の文字列。空の optin 属性は、デフォルトでは無視されます。これは、iPlanet Messaging Server 5.2 からの変更です。5.2 では、空の optin 属性は空の optin リストでフィルタ処理をトリガしました。5.2 の動作は、spamfilterX_null_optin を実際には発生することのない文字列に設定することによって復元できます。
SpamfilterX_null_action	フィルタ処理ソフトウェア X の判定が NULL で返された場合にメッセージの処理を指定する Sieve ルールを定義します。Sieve 式は、ファイル URL を使って外部的に保存できます。例 : file:///var/opt/SUNWmsgsr/config/null_action.sieve また、アドレスがスパムの送信に使用されていた、関係のない相手に非配信通知を配信する傾向があるため、スパムを拒否するときに Sieve 拒否アクションを使用しないでください。 デフォルト : data:,discard;
SpamfilterX_string_action	判定が文字列で返された場合にメッセージの処理を指定する Sieve ルールを定義します。Sieve 式は、ファイル URL を使って外部的に保存できます。例 : file:///var/opt/SUNWmsgsr/config/null_action.sieve また、サーバーがスパムの送信に使用されていた、関係のない相手に非配信通知を配信する傾向があるため、スパムを拒否するときに Sieve 拒否アクション (reject) を使用しないでください。 デフォルト : data:,require "fileinto"; fileinto "$U; ここで、$U は verdict が返した文字列です。

表 14-2　MTA スパムフィルタオプション (option.dat) (続き)

SpamAssassin 用の MTA オプション	説明
spamfilter*X*_verdict_*n*	spamfilter*X*_verdict_*n* および spamfilter*X*_action_*n* は対応するペアであり、*n* は 0 から 9 の数字です。これらのオプションにより、任意の判定文字列に Sieve フィルタを指定できます。これは、spamfilter*X*_verdict_*n* および spamfilter*X*_action_*n* を、それぞれ判定文字列および Sieve フィルタに設定することによって行われます。ここで、*n* は 0 から 9 の整数です。たとえば、サイトで「reject」判定によって Sieve 拒否アクションを発生させるには、次のように指定します。 spamfilter1_verdict_*0*=reject spamfilter1_action_*0*=data:,require "reject"; reject "Rejected by spam filter"; spamfilter*X*_verdict_*n* オプションと対応するアクションのオプションすべてのデフォルト値は、空の文字列です。 デフォルト : なし
spamfilter*X*_action_*n*	spamfilter*X*_verdict_*n* を参照してください。デフォルト : なし
spamfilter*X*_final	一部のフィルタ処理ライブラリには、受取人アドレスに基づいて一連のアクションを実行する機能があります。spamfilter*X*_final は、フィルタ処理ライブラリに渡される受取人アドレスを指定します。値が 0 の場合はいわゆる中間アドレスを使用し、1 の場合は最終形式の受取人アドレスを送信します。 デフォルト : 0

表 14-2　MTA スパムフィルタオプション (option.dat) (続き)

SpamAssassin 用の MTA オプション	説明
optin_user_carryover	転送はスパムフィルタ処理に対するチャレンジです。forward 配信オプションを指定し、別のユーザーの転送アドレスを指定するユーザーエントリを考慮します。さらに、ユーザーエントリは、一部の固有のフィルタ処理に optin するように設定します。フィルタ処理を転送されるメッセージに適用すべきでしょうか。これに対し、1 人の特定のユーザーにとって正しいフィルタ処理を選択しても、ほかのユーザーにとっては正しい選択ではない場合もあります。また、サイトのセキュリティポリシーに違反する手段としてフィルタ処理操作が省かれることもあります。 すべてのケースに対応できる 1 つの答えはないため、OPTIN_USER_CARRYOVER は、スパムのフィルタ処理の optin リストが転送されるときに、あるユーザーエントリまたはエイリアスエントリから別のエントリに転送される方法を制御します。これはビットエンコード値です。ビット値には、次のような意味があります。 ビット 0 (値 1): 各 LDAP ユーザーエントリは、以前アクティブだったユーザーまたはドメインの optin よりも無条件で優先されます。 ビット 1 (値 2): ユーザーのドメインに optin 属性がある場合、以前アクティブだったユーザー、ドメイン、エイリアスよりも優先されます。 ビット 2 (値 4): ユーザーに optin 属性がある場合、以前アクティブだったユーザー、ドメイン、エイリアスよりも優先されます。 ビット 3 (値 8): [optin] 非定位置パラメータで指定した optin は、以前アクティブだったユーザー、ドメイン、エイリアスよりも優先されます。 デフォルト : 0 (ユーザーが別のユーザーに転送する配信オプションを持つ場合、optin が累積します。このデフォルトにより、転送時にサイトのセキュリティポリシーが有効であることが保証されます。ほかの設定では保証されない場合があります)。

Symantec Brightmail AntiSpam を使用する

Brightmail ソリューションは、Brightmail サーバーおよびスパムとウィルスを防止するルールで構成され、ルールのリアルタイムの更新版は電子メールサーバーにダウンロードされます。

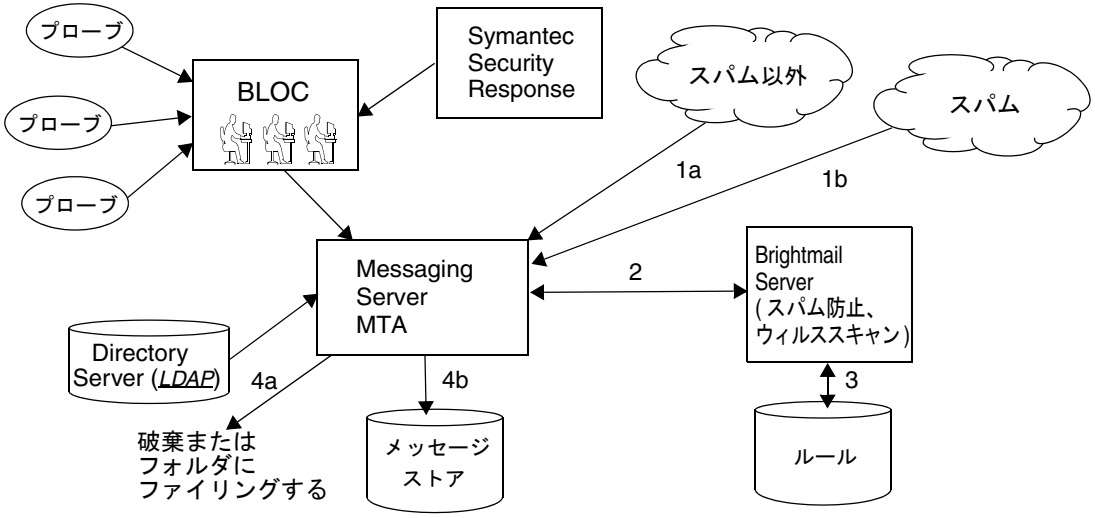
Brightmail の機能

Brightmail サーバーは顧客のサイトに配備されます。Brightmail では、電子メールプローブがインターネット周辺に配置され、新しいスパムを検出します。Brightmail の技術者はリアルタイムでこのスパムを阻止するカスタムルールを作成します。ルールは Brightmail サーバーにダウンロードされます。これもリアルタイムで行われます。Brightmail のデータベースは更新され、Brightmail サーバーは特定のユーザーまたはドメインの電子メールに対してこのデータベースフィルタを使用します。

Brightmail のアーキテクチャ

図 14-1 に、Brightmail のアーキテクチャを示します。

図 14-1 Brightmail と Messaging Server のアーキテクチャ



Brightmail Logistics and Operations Center (BLOC) が電子メールプローブからスパムを受信すると、オペレータがただちに適切なスパムフィルタ処理ルールを作成します。作成されたルールは、Brightmail の顧客のマシンにダウンロードされます。同様に、Symantec Security Response のリアルタイムのウィルスルールが Brightmail から送信されます。これらのルールは顧客の Brightmail サーバーでスパムやウィルスを検出するために使用されます。

MTA は Brightmail SDK を使用して Brightmail サーバーと通信します。MTA は Brightmail からの応答に基づいてメッセージを送信します。MTA はメール (1a) または (1b) を受信すると、Brightmail サーバー (2) にメッセージを送信します。Brightmail サーバーはルールとデータを使用してメッセージがスパムやウィルスであるかどうか判断し (3)、判定を MTA に返します。その判定に基づいて、MTA はメッセージを破棄するか、フォルダに保存するか (4a)、通常どおり宛先に配信するかのいずれかを実行します。

Brightmail SDK はサードパーティのソフトウェアなので、Sun のインストールキットには含まれていません。Brightmail SDK およびサーバーソフトウェアは、Brightmail Inc. から入手する必要があります。MTA には、Brightmail を統合するために Brightmail SDK をロードするかどうか、どこにロードするかを指定する構成設定があります。

SDK がロードされると、Brightmail のメッセージ処理は複数の係数と細分度 (アクティブな処理が「optin」であることを示す、Brightmail で使用される用語) によって決定されます。これは、次の基準に基づいて示されます。

- ソースチャネルまたは宛先チャネルは Brightmail に対して有効になっているかどうか (`imta.cnf`)
- optin されたサービス用のチャネルのデフォルトはあるかどうか (`imta.cnf`)
- ドメイン単位の optin があるかどうか (LDAP)
- ユーザー単位の optin があるかどうか (LDAP)

各メッセージ受取人にとっては、上記の optin とデフォルトは組み合わされています。つまり、チャネルのデフォルトがすでにスパムとウィルスの両方に対して指定されていれば、ユーザー単位の optin は不要になります。言い換えると、システム管理者が全員に対してスパムとウィルスのフィルタ処理を行うことを決定すれば、ユーザーにスパムやウィルスの対策を選択させる必要はないということです。システムまたはドメインオプションによってすでにユーザーが optin している場合、処理を optin 解除する (そのサービスを不要とする) ことはできません。また、サービスを optin していて、別のアドレスにメールを転送した場合、そのアドレスはフィルタ処理が実行されたあとにメールを受信します。

提供されるサービスは、ウィルス検出またはスパム検出の 2 つのみです。Brightmail では、「content-filtering」サービスも提供されますが、この機能は Sieve を使用して提供されるため、Brightmail で Sieve フィルタ処理を実行した場合の付加価値はありません。

メッセージにウィルスが含まれていると判明した場合は、ウィルスを除去するように Brightmail サーバーを設定でき、これによって除去済みのメッセージが MTA に再送信されます。ウィルス除去済みのメッセージが再送信されると、元のメッセージから情報が失われることによって生じる副次的な悪影響があるため、MTA に除去済みのメッセージを再送信しないように Brightmail を設定することをお勧めします。メッセージがスパムである場合、Brightmail からその設定とともに返された判定に基づいて、MTA はメッセージの処理を決定できます。メッセージは、破棄したり、フォルダにファイリングしたり、件名にスパムまたはウィルスとしてタグ付けしたり、Sieve ルールに渡したり、通常どおり `INBOX` に配信したりできます。

Brightmail サーバーは、MTA と同一システム上に配置することも、別のシステムに配置することもできます。任意の数の MTA を実行する Brightmail サーバーのファームを構築することもできます。Brightmail SDK では、Brightmail 設定ファイルによって使用する Brightmail サーバーが決定されます。

Brightmail の要件とパフォーマンスの考慮

- Brightmail サーバーは Solaris オペレーティングシステムで実行する必要があります。
- Brightmail がスパムとウィルスの両方のチェックを実行する場合、MTA のメッセージスループットは 50% ほど低下する可能性があります。MTA のスループットを維持するには、各 MTA につき 2 台の Brightmail サーバーが必要です。
- SpamAssassin にはユーザー単位で別の種類のフィルタ処理を実行する機能がありますが、同じメッセージに 2 つの別のフィルタ処理基準を同時に適用することはできません。このため、SpamAssassin ではシステム全体のフィルタ処理のみ可能です。個々のユーザーに対するフィルタ処理はできません。

Brightmail を配備する

以下の手順を実行して、Brightmail を配備します。

- **Brightmail をインストールして構成します。**インストールおよび設定情報については、Brightmail ソフトウェアマニュアルを参照するか、販売代理店にお問い合わせください。一部の Brightmail 設定ファイルオプションについては 478 ページの「Brightmail 設定オプション」に示しますが、完全な最新の情報は Brightmail のマニュアルを参照してください。
- **Brightmail クライアントライブラリをロードおよび構成します。**このためには、Brightmail クライアントライブラリ `libbmiclient.so` と設定ファイル `config` を MTA に対して指定する必要があります。461 ページの「スパムのフィルタ処理ソフトウェアのクライアントライブラリをロードおよび構成する」を参照してください。
- **スパム用にフィルタ処理するメッセージを指定します。**ユーザー、ドメイン、またはチャネルごとにメッセージをフィルタ処理できます。463 ページの「フィルタ処理を行うメッセージを指定する」を参照してください。
- **スパムメッセージに対して実行するアクションを指定します。**スパムは、破棄したり、フォルダにファイリングしたり、件名にタグ付けしたりできます。469 ページの「スパムメッセージに対して実行するアクションを指定する」を参照してください。
- **必要に応じてほかの MTA フィルタ設定パラメータを設定します。**471 ページの「MTA スパムフィルタオプション (`option.dat`)」を参照してください。

Brightmail 設定オプション

一部の Brightmail 設定ファイルオプションについては、表 14-3 に示します。Brightmail 設定ファイルオプションの完全なリストは、Brightmail から入手できます。オプションおよび値は大文字と小文字が区別されません。

表 14-3　Brightmail 設定ファイルオプション (一部)

Brightmail オプション	説明
`blSWPrecedence`	1 つのメッセージが複数の判定を受けることがあります。その場合、このオプションで順序を指定します。このオプションを `virus-spam` と指定した場合、メッセージに対して先にウィルス処理、次にスパム処理が行われます。判定はハイフン (-) で区切られます。これは Sun Java System Messaging Server で Brightmail を使用する場合に推奨される設定です。

表 14-3　Brightmail 設定ファイルオプション (一部) (続き)

Brightmail オプション	説明
blSWClientDestinationDefault	スパムでもウィルスでもなく、したがって判定を受けない通常のメッセージの配信方法を指定します。このようなメールを通常に配信するには、値として inbox を指定します。デフォルトはありません。
blSWLocalDomain	この属性ではローカルとみなされるドメインを指定します。いくつかのドメインがすべてローカルとみなされ、それを指定する場合は、この属性の行は複数になることがあります。ローカルドメインと外部ドメインを使用して、判定のための 2 種類の処理を指定します。 次の blSWClientDestinationLocal と blSWClientDestinationForeign を参照してください。たとえば、次のように指定します。 blSWLocalDomain=siroe.com
blSWClientDestinationLocal	このオプションではローカルドメイン用に判定とアクションのペアを指定します。この指定は通常 2 行で行われ、1 行はスパム用、もう 1 行はウィルス用です。値は verdict|action という形式をとります。次に例を示します。 blSWClientDestinationLocal=spam|spambox blSWClientDestinationLocal=virus| 「null」アクション (| の右側に指定なし) に対するデフォルトの Brightmail 解釈は、メッセージを破棄することです。したがって、上記の例では判定が virus であるメッセージは破棄されます。また、判定が spam である場合、上記の例では spambox というフォルダにメッセージが保存されます。メッセージがスパムでもウィルスでもない場合、判定は一致せず、前出の blSWClientDestinationDefault の設定内容に基づいてメールは通常どおり配信されます。 Brightmail サーバーを MTA と別のマシンで使用している場合には、各 MTA によって実行されるアクションをカスタマイズできます。それには次の MTA オプションを使用して Brightmail サーバーから返されるアクションや判定を無効にします。Brightmail_verdict_n/Brightmail_action_n/Brightmail_null_action/Brightmail_string_action。この例では、MTA で別の Brightmail_null_action を使用してウィルスアクション (MTA のアクションを無効にするアクション) を無効にできます。または Brightmail_verdict_0=spambox と Brightmail_action_0=data:,require "fileinto";fileinto "Junk"; を使用して、spambox の代わりに Junk というフォルダに保存できます。

表 14-3　Brightmail 設定ファイルオプション (一部) (続き)

Brightmail オプション	説明
`blSWClientDesintationForeign`	上記の `blSWClientDestinationLocal` と同じ形式と内容です。ただし、ローカル以外のドメインのユーザーに適用されます。
`blSWUseClientOptin`	Sun Java System Messaging Server で使用する場合は、常に TRUE に設定してください。
`blswcServerAddress`	`ip:port[,ip:port,...]` という形式で Brightmail サーバーの IP アドレスとポート番号を指定します

SpamAssassin を使用する

この節には、以下の項があります。

SpamAssassin の概要

Messaging Server では、SpamAssassin の使用がサポートされています。SpamAssassin はフリーウェアのメールフィルタで、スパムの特定に使用されます。SpamAssassin は Perl で記述されたライブラリ、アプリケーションのセット、および SpamAssassin のメッセージングシステムへの統合に使用するユーティリティで構成されています。

SpamAssassin では、すべてのメッセージのスコアが計算されます。スコアは、メッセージヘッダーや本文の情報に対して一連のテストを実行することによって計算されます。各テストに成功するか失敗するかによって、True (スパム) または False (スパムではない) がレンダリングされます。スコアは正または負の実数です。スコアが指定したしきい値 (通常 5.0) を超えると、スパムであるとみなされます。SpamAssassin の結果文字列の例を次に示します。

True ; 18.3 / 5.0

True は、メッセージがスパムであることを示します。18.3 は SpamAssassin スコアです。5.0 はしきい値です。

SpamAssassin には高い設定性があります。テストはいつでも追加したり削除したりでき、既存テストのスコアは調整されます。これらはすべてさまざまな設定ファイルを通じて実行されます。SpamAssassin の詳細については、SpamAssassin の Web サイトを参照してください。

Brightmail のスパムおよびウィルススキャンライブラリを呼び出す場合と同じ方法で SpamAssassin spamd サーバーに接続できます。Messaging Server で提供するモジュールの名前は libspamass.so です。

SpamAssassin/Messaging Server の動作方式

spamd は SpamAssassin のデーモンバージョンであり、MTA から呼び出すことができます。spamd は、要求のソケットを待機し、子プロセスを生成してメッセージがテストされるようにします。子プロセスは、メッセージが処理され結果が戻されたあと、破棄されます。コード自体は子プロセス間で共有されるため、理論上、フォークは効率的なプロセスであるはずです。

SpamAssassin インストールによるクライアント部分、spamc は、使用されません。代わりに、Messaging Server の一部である libspamass.so という共有ライブラリによってこの機能が実行されます。libspamass.so は、Brightmail SDK の読み込みと同じように読み込まれます。

MTA から見ると、SpamAssassin とスパムのフィルタ処理用の Brightmail は、ほぼ透過的に切り替えることができます。同じ機能を持っているわけではないので、完全に透過的というわけではありません。たとえば、Brightmail ではウィルス用のフィルタ処理も行えますが、SpamAssassin はスパム用のフィルタ処理にしか使われません。この 2 つのソフトウェアパッケージによって返される結果、つまり判定も、異なります。SpamAssassin がスコアを提供するのに対し、Brightmail は判定名のみを提供します。このため、設定にもいくつかの違いがあります。

MTA と一緒に SpamAssassin を使用すると、SpamAssassin からはスコアと判定のみが返されます。メッセージ自体は変更されません。つまり、ヘッダーの追加や件名の変更のようなオプションは、Sieve スクリプトによって行う必要があります。また、mode オプションを使用すると、判定を示すために返す文字列を指定できます。文字列の選択肢は、NULL、デフォルト、SpamAssassin の結果文字列、または verdict 文字列です。詳細は、491 ページの表 14-4 を参照してください。

SpamAssassin の要件と使用法の考慮

- SpamAssassin はフリーウェアです。ソフトウェアとマニュアルについては、http://www.spamassassin.org を参照してください。
- SpamAssassin は、スパムを非常に正確に検出できるよう調整し設定することができます。SpamAssassin の調整はユーザーと SpamAssassin コミュニティが行うもので、Messaging Server は SpamAssassin で実行できる内容を提供したり拡張したりはしません。
- 特定のメンバーが利用可能でない場合、SpamAssassin によるスループットの低下は Brightmail の場合より大きいと思われます。
- MTA と統合された SpamAssassin は、ユーザー、ドメイン、またはチャネルに対して有効にできます。
- SpamAssassin は、Vipul Razor または Distributed Checksum Clearinghouse (DCC) のようなそのほかのオンラインデータベースを使って設定できます。
- Messaging Server は Secure Socket Layer (SSL) バージョンの `libspamass.so` を提供しません。ただし、OpenSSL を使うように SpamAssassin を構築できます。
- Perl 5.6 以降が必要です。

SpamAssassin を実行する場所

SpamAssassin は、それ自体が置かれている個々のシステム上、単一のシステム配備の Messaging Server と同じシステム上、または 2 層配備の MTA と同じシステム上で実行できます。MTA とメッセージストア間で Local Mail Transfer Protocol (LMTP) が使われている場合、フィルタ処理は MTA から呼び出す必要があります。メッセージストアから呼び出すことはできません。MTA とメッセージストア間で SMTP が使われている場合は、いずれか一方から呼び出すことができ、いずれかのシステムか 3 つ目の別のシステムで実行できます。

SpamAssassin を実行するサーバーのファームを使用する場合は、それらの前でロードバランサを使用する必要があります。MTA は、SpamAssassin サーバー用に 1 つのアドレスのみを持つよう設定されます。

SpamAssassin を配備する

以下の手順を実行して、SpamAssassin を配備します。

- **SpamAssassin をインストールして構成します。**インストールおよび設定情報については、SpamAssassin ソフトウェアマニュアルを参照してください。491 ページの「SpamAssassin オプション」も参照してください。
- **SpamAssassin のクライアントライブラリをロードおよび構成します。**このためには、クライアントライブラリ `libspamass.so` と設定ファイルを MTA に対して指定する必要があります (ファイルは作成する必要がある)。461 ページの「スパムのフィルタ処理ソフトウェアのクライアントライブラリをロードおよび構成する」を参照してください。
- **スパム用にフィルタ処理するメッセージを指定します。**ユーザー、ドメイン、またはチャネルごとにメッセージをフィルタ処理できます。463 ページの「フィルタ処理を行うメッセージを指定する」を参照してください。
- **スパムメッセージに対して実行するアクションを指定します。**スパムは、破棄したり、フォルダにファイリングしたり、件名にタグ付けしたりできます。469 ページの「スパムメッセージに対して実行するアクションを指定する」を参照してください。
- **必要に応じてほかのフィルタ設定パラメータを設定します。**471 ページの「MTA スパムフィルタオプション (`option.dat`)」を参照してください。

SpamAssassin 設定の例

この節では、一般的な SpamAssassin 設定の例について説明します。

- 483 ページの「スパムを個別のフォルダにファイリングするには」
- 485 ページの「スパムメッセージに SpamAssassin スコアを含むヘッダーを追加する」
- 487 ページの「件名行に SpamAssassin の結果文字列を追加するには」

注 これらの例では、多数のオプションとキーワードを使用しています。詳細は、467 ページの「スパムフィルタ用の MTA チャネルキーワード」および 471 ページの「MTA スパムフィルタオプション (`option.dat`)」を参照してください。

スパムを個別のフォルダにファイリングするには

この例では、ローカルメッセージストアに届くメッセージをテストし、`spam` というフォルダ内にスパムをファイリングします。最初の 3 つのステップは任意の順序で実行できます。

1. SpamAssassin の設定ファイルを作成します。

   このファイルの名前と場所は、手順 2 で指定したものです。適切な名前は spamassassin.opt です。このファイルには以下の行が含まれます。

```
host=127.0.0.1
port=2000
mode=0
verdict=spam
debug=1
```

   host および port は、spamd が実行されているシステムの名前と、spamd が着信要求を待機するポートの名前です。mode=0 は、メッセージがスパムとして認識された場合に、verdict によって指定された文字列が返されることを指定します。debug=1 は、SpamAssassin ライブラリでデバッグをオンにします。SpamAssassin 設定パラメータについては、491 ページの表 14-4 を参照してください。

2. option.dat ファイルに以下の行を追加します。

```
! for Spamassassin
spamfilter1_config_file1=/opt/SUNWmsgsr/config/spamassassin.
opt
spamfilter1_library1=/opt/SUNWmsgsr/lib/libspamass.so
spamfilter1_optional=1
spamfilter1_string_action=data:,require "fileinto"; fileinto
"$U;
```

   spamfilter1_config_files は、SpamAssassin 設定ファイルを指定します。

   spamfilter1_library は、SpamAssassin 共有ライブラリを指定します。

   spamfilter1_optional=1 は、spamd による失敗があった場合に MTA が操作を続行することを指定します。

   spamfilter1_string_action は、スパムメッセージに対して実行する Sieve アクションを指定します。

   この例では、spamfilter1_string_action は必須ではありません。これは、デフォルト値があらかじめ data:,require "fileinto"; fileinto "$U; であるためです。この行では、スパムメッセージがフォルダに送られることを指定しています。フォルダの名前は、SpamAssassin によって返されるスパム判定値です。SpamAssassin によって返される値は、spamassassin.opt の verdict オプションによって指定されます (手順 1 を参照)。この例では、フォルダ名は spam です。

3. フィルタ処理するメッセージを指定します。

   ローカルメッセージストアに届くすべてのメッセージをフィルタ処理するには、ims-ms チャネルの destinationspamfilterXoptin spam を追加して imta.cnf ファイルを変更します。

```
!
! ims-ms
ims-ms defragment subdirs 20 notices 1 7 14 21 28 backoff
"pt5m" "pt10m" "pt30m" "pt1h"  "pt2h" "pt4h" maxjobs 4 pool
IMS_POOL fileinto
$U+$S@$D destinationspamfilter1optin spam
ims-ms-daemon
```

4. 設定をコンパイルしなおし、サーバーを再起動します。MTA を再起動するだけでかまいません。stop-msg を実行する必要はありません。

```
# imsimta cnbuild
# imsimta restart
```

5. spamd デーモンを起動します。次の形式のコマンドを使用してこれを実行します。

   ```
   spamd -d
   ```

   spamd は、デフォルトではローカルシステムからの接続を受け入れるだけです。SpamAssassin と Messaging Server が別のシステム上で実行されている場合は、この構文が必要です。

   ```
   spamd -d -i listen_ip_address -A allowed_hosts
   ```

   listen_ip_address は待機対象のアドレスであり、*allowed_hosts* はこの spamd インスタンスに接続できる、認証されたホストまたはネットワークのリスト (IP アドレスを使用) です。

注 -i *listen_ip_address* で 0.0.0.0 を設定すると、spamd にすべてのアドレスを待機させることができます。spamfilter*X*_verdict_*n* を使用すると、システムの IP アドレスを変更したときにコマンドスクリプトを変更しなくても済むため、すべてのアドレスを待機することをお勧めします。

スパムメッセージに SpamAssassin スコアを含むヘッダーを追加する

この例では、SpamAssassin によってスパムであると判断された各メッセージに、ヘッダー「Spam-test: 結果文字列」を追加しています。ヘッダーの例は次のとおりです。

```
Spam-test:True ; 7.3 / 5.0
```

ここで、「Spam-test:」はリテラルで、その後ろはすべて結果文字列です。True はそれがスパムであることを意味し、False はスパムではないことを意味します。7.3 は SpamAssassin スコアです。5.0 はしきい値です。この結果は、一定のスコアを超えたメールや一定のスコア間のメールをファイリングまたは破棄できる、Sieve フィルタを設定する場合に便利です。

また、USE_CHECK を 0 に設定すると、判定文字列とともに一致する SpamAssassin テストのリストが返されます。491 ページの表 14-4 の「USE_CHECK」を参照してください。

1. フィルタ処理するメッセージを指定します。詳細は、483 ページの「スパムを個別のフォルダにファイリングするには」の手順 3 を参照してください。
2. SpamAssassin の設定ファイルを作成します。

   このファイルの名前と場所は、spamfilter_config*X*_file で指定したものです (次の手順を参照)。このファイルには、以下の行が含まれます。

```
host=127.0.0.1
port=2000
mode=1
field=
debug=1
```

   host および port は、spamd が実行中のシステムの名前と、spamd が着信要求を待機するポートの名前です。mode=1 は、メッセージがスパムであると判明した場合に SpamAssassin の結果文字列が返されることを指定します。field= は、SpamAssasin の結果文字列のプレフィックスを指定します。この例では、プレフィックスは Sieve スクリプトで指定するため、必要ありません。debug=1 は、SpamAssassin ライブラリでデバッグをオンにします。

3. option.dat ファイルに以下の行を追加します。

```
!for Spamassassin
spamfilte1r_config_file=/opt/SUNWmsgsr/config/spamassassin.opt
spamfilter1_library=/opt/SUNWmsgsr/lib/libspamass.so
spamfilter1_optional=1
spamfilter1_string_action=data:,require ["addheader"];addheader
"Spam-test:$U";
```

前に示した例と同様、最初の 3 つのオプションは、SpamAssassin 設定ファイル、共有ライブラリ、および、共有ライブラリで失敗があった場合に MTA 操作を続行することを指定します。

```
spamfilter1_string_action=data:,require ["addheader"];addheader
"Spam-test:$U";
```

この行は、スパムメッセージにヘッダーを追加することを指定します。ヘッダーには、リテラルプレフィックス Spam-text: と、その後に SpamAssassin が返す文字列が含まれます。手順 2 で mode=1 を指定したため、SpamAssassin 結果文字列が返されます。例 : True ; 7.3 / 5.0

4. 設定をコンパイルしなおし、サーバーを再起動して、spamd デーモンを起動します。

483 ページの「スパムを個別のフォルダにファイリングするには」を参照してください。

件名行に SpamAssassin の結果文字列を追加するには

SpamAssassin の結果文字列を件名行に追加することによって、ユーザーが SpamAssassin スコアを含むメッセージを読むかどうかを判断できます。次に例を示します。

Subject:[SPAM True ; 99.3 / 5.0] Free Money At Home with Prescription Xanirex!

USE_CHECK を 0 に設定すると、判定文字列とともに一致する SpamAssassin テストのリストが返されます (491 ページの表 14-4 の「USE_CHECK」を参照)。このリストは非常に長くなることがあるため、USE_CHECK は 1 に設定することをお勧めします。

1. フィルタ処理するメッセージを指定します。483 ページの「スパムを個別のフォルダにファイリングするには」の手順 3 を参照してください。
2. SpamAssassin の設定ファイルを作成します。

   この手順の詳細は、483 ページの「スパムを個別のフォルダにファイリングするには」を参照してください。mode=1 は、メッセージがスパムであると判明した場合に SpamAssassin の結果文字列が返されることを指定します。

```
host=127.0.0.1
port=2000
mode=1
debug=1
```

host および port は、spamd が実行中のシステムの名前と、spamd が着信要求を待機するポートの名前です。mode=1 は、メッセージがスパムであると判明した場合に SpamAssassin の結果文字列が返されることを指定します。debug=1 は、SpamAssassin ライブラリでデバッグをオンにします。

3. option.dat ファイルに以下の行を追加します。

```
!for Spamassassin
spamfilter1_config_file=/opt/SUNWmsgsr/config/spamassassin.opt
spamfilter1_library=/opt/SUNWmsgsr/lib/libspamass.so
spamfilter1_optional=1
spamfilter1_string_action=data:,addtag "[SPAM detected:$U]";
```

前に示した例と同様、最初の 3 つのオプションは、SpamAssassin 設定ファイル、共有ライブラリ、および、共有ライブラリで失敗があった場合に MTA 操作を続行することを指定します。

```
spamfilter1_string_action=data:,addtag "[SPAM detected $U]";
```

この行は、Subject: 行にタグを追加することを指定します。これには、リテラルプレフィックス SPAM detected が含まれ、そのあとに field 文字列 (デフォルトは Spam-Test)、そのあとに SpamAssassin によって返される [結果文字列] が含まれます。手順 2 で mode=1 を指定したため、SpamAssassin 結果文字列が返されます。したがって、件名行は次のようになります。

```
Subject:[SPAM detected Spam-Test:True ; 11.3 / 5.0] Make Money!
```

addheader と addtag を併用することもできます。

```
spamfilter1_string_action=data:,require ["addheader"];addtag
"[SPAM detected $U]";addheader "Spamscore:$U";
```

メッセージは次のようになります。

```
Subject:[SPAM detected Spam-Test:True ; 12.3 / 5.0] Vigaro Now!
Spamscore:Spam-Test:True ; 12.3 / 5.0
```

spamassassin.opt で field= を設定して、Spam-Test のデフォルト値を削除します。よりすっきりとしたメッセージが返されます。

```
Subject:[SPAM True ; 91.3 / 5.0] Vigaro Now!
Spamscore:True ; 91.3 / 5.0
```

4. 設定をコンパイルしなおし、サーバーを再起動して、spamd デーモンを起動します。

483 ページの「スパムを個別のフォルダにファイリングするには」を参照してください。

SpamAssassin をテストする

SpamAssassin をテストするには、最初に spamassassion.opt ファイルに debug=1 を設定します。imta.cnf にあるチャネル固有の master_debug または slave_debug をオンにする必要はありません。そのあと、テストユーザーにテストメッセージを送信します。*msg_svr_base*/data/tcp_local_slave.log* ファイルには次のような行があるはずです。

```
15:15:45.44: SpamAssassin callout debugging enabled; config
/opt/SUNWmsgsr/config/spamassassin.opt
     15:15:45.44: IP address 127.0.0.1 specified
     15:15:45.44: Port 2000 selected
     15:15:45.44: Mode 0 selected
     15:15:45.44: Field "Spam-Test:" selected
     15:15:45.44: Verdict "spam" selected
     15:15:45.44: Using CHECK rather than SYMBOLS
     15:15:45.44: Initializing SpamAssassin message context
     ...
     15:15:51.42: Creating socket to connect to SpamAssassin
     15:15:51.42: Binding SpamAssassin socket
     15:15:51.42: Connecting to SpamAssassin
     15:15:51.42: Sending SpamAssassin announcement
     15:15:51.42: Sending SpamAssassin the message
     15:15:51.42: Performing SpamAssassin half close
     15:15:51.42: Reading SpamAssassin status
     15:15:51.67: Status line:SPAMD/1.1 0 EX_OK
     15:15:51.67: Reading SpamAssassin result
     15:15:51.67: Result line:Spam:False ; 1.3 / 5.0
     15:15:51.67: Verdict line:Spam-Test:False ; 1.3 / 5.0
     15:15:51.67: Closing connection to SpamAssassin
     15:15:51.73: Freeing SpamAssassin message context
```

ログファイルにこのような行が含まれていない場合、または spamd が実行されていない場合は、SMTP サーバーに最後のピリオド (.) が送信されたあと、SMTP ダイアログに次のようなエラーメッセージが返されます。

```
452 4.4.5 Error writing message temporaries - Temporary scan
failure:End message status = -1
```

また、option.dat に spamfilter1_optional=1 が設定 (強く推奨) されている場合、メッセージは受け入れられますがフィルタ処理は行われません。これはスパムのフィルタ処理が有効になっていないのと同じで、tcp_local_slave.log* に次の行が表示されます。

```
15:35:15.69: Creating socket to connect to SpamAssassin
15:35:15.69: Binding SpamAssassin socket
15:35:15.69: Connecting to SpamAssassin
15:35:15.69: Error connecting socket:Connection refused
15:35:15.72: Freeing SpamAssassin message context
```

SpamAssassin に対する呼び出しは、SMTP サーバーがメッセージ全体を受信したあと、つまり、最後の「.」が SMTP サーバーに送信されたあとで、SMTP サーバーが受け取ったメッセージの差出人に認識される前に行われます。

もう 1 つのテストは、たとえば Mail-SpamAssassin-2.60 ディレクトリから、sample-spam.txt を使ってサンプルのスパムメッセージを送信することです。このメッセージには、次のような特殊なテキスト文字列が含まれます。

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

対応する tcp_local_slave.log* には、次のような内容が含まれます。

```
16:00:08.15: Creating socket to connect to SpamAssassin
16:00:08.15: Binding SpamAssassin socket
16:00:08.15: Connecting to SpamAssassin
16:00:08.15: Sending SpamAssassin announcement
16:00:08.15: Sending SpamAssassin the message
16:00:08.15: Performing SpamAssassin half close
16:00:08.15: Reading SpamAssassin status
16:00:08.43: Status line:SPAMD/1.1 0 EX_OK
16:00:08.43: Reading SpamAssassin result
16:00:08.43: Result line:Spam:True ; 1002.9 / 5.0
16:00:08.43: Verdict line:Spam-Test:True ; 1002.9 / 5.0
16:00:08.43: Closing connection to SpamAssassin
16:00:08.43: Mode 0 verdict of spam
16:00:08.43: Mode 0 verdict of spam
16:00:08.47: Freeing SpamAssassin message context
```

mail.log_current ファイルの対応するエントリは、次のようになります。宛先アドレスの +spam 部分は、メッセージが spam というフォルダにファイリングされることを示します。

```
15-Dec-2003 15:32:17.44 tcp_intranet ims-ms E 1 morchia@siroe.com
rfc822;morchia
morchia+spam@ims-ms-daemon
15-Dec-2003 15:32:18.53 ims-ms D 1 morchia@siroe.com rfc822;morchia
morchia+spam@ims-ms-daemon
```

SpamAssassin オプション

この節では、SpamAssassin オプションの表を示します。

表 14-4 SpamAssassin オプション (spamassassin.opt)

オプション	説明	デフォルト
debug	libspamass.so でデバッグをオンにするかどうかを指定します。spamd 自体のデバッグは、spamd を呼び出すコマンド行で制御されます。0 または 1 に設定します。	0
field	SpamAssasin の結果文字列のプレフィックスを指定します。SpamAssassin の結果は次のようになります。 Spam-Test:False ; 0.0 / 5.0 Spam-Test:True ; 27.7 / 5.0 field オプションは、結果の Spam-Test 部分の変更方法を提供します。空の field 値が指定されると「:」が削除されることに注意してください。 USE_CHECK が 0 に設定されている場合、結果の文字列は次のようになります。 Spam-test:False ; 0.3 / 4.5 ; HTML_MESSAGE,NO_REAL_NAME Spam-test:True ; 8.8 / 4.5 ; NIGERIAN_BODY, NO_REAL_NAME,PLING_PLING,RCVD_IN_SBL,SUBJ_ALL_CAPS	"Spam-test"
host	spamd が実行されているシステムの名前。	localhost

表 14-4　SpamAssassin オプション (spamassassin.opt) (続き)

オプション	説明	デフォルト
mode	SpamAssassin フィルタ結果から判定情報への変換を制御します。つまり、メッセージ処理後にどの判定情報を返すかを指定します。次の 4 種類のモードが使用可能です。詳細は、493 ページの「SpamAssassin の mode オプション」を参照してください。 0 - メッセージがスパムメッセージの場合は、判定文字列 (verdict オプションで指定される文字列) を返します。MTA オプション spamfilterX_string_action を使用して、verdict 文字列が返された場合の処理を指定できます。以下で定義する verdict オプションが空である場合や指定されていない場合、メッセージがスパムメッセージのときは、NULL 判定を返します。MTA オプション spamfilterX_null_action を使用して、NULL 判定が返された場合の処理を指定できます。 スパムではない場合は、SpamAssassin のデフォルトの結果文字列を返します。デフォルトの判定とは、何のアクションもとらず通常どおり配信することを意味します。 1 - メッセージがスパムであると判明した場合は SpamAssassin の結果文字列を返します。スパムではない場合は、SpamAssassin のデフォルトの結果文字列を返します。デフォルトの判定とは、何のアクションもとらず通常どおり配信することを意味します。SpamAssassin の結果文字列は、次のように表示されます。True ; 6.5 / 7.3 2 - モード 1 と同様。ただし、メッセージがスパムかどうかにかかわらず SpamAssassin の結果文字列が返される点で異なります。デフォルト判定または NULL 判定が返されることはなく、verdict オプションが使用されることはありません。 3 - メッセージがスパムであると判明した場合に SpamAssassin の結果文字列を返します。スパムではない場合は verdict オプションで指定された verdict 文字列を返します。SpamAssassin 結果文字列に対するアクションを制御するには、spamfilterX_verdict_*n* オプションと spamfilterX_action_*n* オプションの一致ペアを使用します。verdict 文字列に対するアクションを制御するには、spamfilterX_string_action を使用します。	0
port	spamd が着信要求を待機するポート番号を指定します。	783
USE_CHECK	1 - spamd CHECK コマンドを使って SpamAssassin スコアを返します。 0 - SYMBOLS コマンドを使って一致する SpamAssassin テストのスコアとリストを返すことができます。2.55 以前のバージョンの SpamAssassin でこのオプションを使うと、システムがハングアップしたりその他の問題が発生することがあります。前述の「field」を参照してください。	
SOCKS_HOST	文字列。中間にある SOCKS サーバーの名前を指定します。このオプションが指定されている場合、指定された SOCKS サーバーを介して ICAP 接続が確立され、直接には接続されません。	""

表 14-4　SpamAssassin オプション (spamassassin.opt) (続き)

オプション	説明	デフォルト
SOCKS_PORT	中間にある SOCKS サーバーが動作しているポートを指定します。	1080
SOCKS_PASSWORD	SOCKS サーバーを介した接続を確立するために使用するパスワード (文字列) を指定します。ユーザー名およびパスワードが必要かどうかは、SOCKS サーバーの設定によって異なります。	""
SOCKS_USERNAME	SOCKS サーバーを介した接続を確立するために使用するユーザー名 (文字列) を指定します。	""
verdict	MODE 0 で使用される判定文字列を指定します。	""

SpamAssassin の mode オプション

メッセージの処理後、SpamAssassin はメッセージがスパムかどうかを判定します。mode オプションを使用すると、判定を示すために返す文字列を指定できます。文字列の選択肢は、NULL、デフォルト、SpamAssassin の結果文字列、または verdict オプションで指定された verdict 文字列です。デフォルトの文字列とは、NULL、SpamAssassin の結果文字列、または verdict で指定された文字列ではなく、その他の設定不可能な結果文字列です。mode の動作の概要を次の表で説明します。

表 14-5　SpamAssassin の mode オプションに対応して返される文字列

verdict の設定	スパムかどうか	**mode=0**	**mode=1**	**mode=2**	**mode=3**
verdict="" (設定なし)	はい	NULL	SpamAssassin の結果	SpamAssassin の結果	SpamAssassin の結果
	いいえ	デフォルト	デフォルト	SpamAssassin の結果	デフォルト
verdict= 文字列	はい	verdict 文字列	SpamAssassin の結果	SpamAssassin の結果	SpamAssassin の結果
	いいえ	デフォルト	デフォルト	SpamAssassin の結果	verdict 文字列

1 列目は、verdict オプションが設定されているかどうかを示します。2 列目は、メッセージがスパムかどうかを示します。mode の列は、各モードに対応して返される文字列を示します。たとえば、verdict が設定されておらず mode が 0 に設定されている場合、メッセージがスパムでなければデフォルト文字列が返されます。verdict が YO SPAM! に設定されていて mode が 0 に設定されている場合、メッセージがスパムであれば YO SPAM! という文字列が返されます。

Symantec Anti-Virus Scanning Engine (SAVSE) を使用する

SAVSE の配備方法以外にも、この節には、ほかの ICAP 対応のスパム防止 / ウィルス防止プログラムを配備する際に有効な情報が含まれています。この節には、以下の項があります。

SAVSE の概要

SAVSE は、ウィルススキャンサービスを提供する TCP/IP サーバーアプリケーションおよび通信用のアプリケーションプログラミングインタフェース (API) です。ネットワークインフラストラクチャ機器を通して提供されたり、これらの機器に保存されるトラフィックを保護するために設計されています。モバイルコードや圧縮ファイル形式も含め、すべての主要なファイル形式でウィルス、ワーム、およびトロイの木馬を検出し、これらから保護します。詳細については、Symantec の Web サイトを参照してください。

注 Messaging Server では、SAVSE のスキャン機能だけがサポートされています。修復機能や削除機能はサポートされていません。

SAVSE の要件と使用法の考慮

これは Symantec から単独でライセンス許可された製品です。

SAVSE の構成では、スキャンモードだけがサポートされています。スキャンと修復、スキャンと削除のモードはサポートされていません。

SAVSE を実行する場所

SAVSE、または ICAP をサポートするほかのサーバーは、それ自体が置かれている個々のシステム上、単一のシステム配備の Messaging Server と同じシステム上、または 2 層配備の MTA と同じシステム上で実行できます。MTA とメッセージストア間で Local Mail Transfer Protocol (LMTP) が使われている場合、フィルタ処理は MTA から呼び出す必要があります。メッセージストアから呼び出すことはできません。MTA とメッセージストア間で SMTP が使われている場合は、いずれか一方から呼び出すことができ、いずれかのシステムか 3 つ目の別のシステムで実行できます。

SAVSE を実行するサーバーのファームを使用する場合は、それらの前でロードバランサを使用する必要があります。MTA は、SpamAssassin サーバー用に 1 つのアドレスのみを持つよう設定されます。

SAVSE を配備する

以下の手順を実行して、SAVSE を配備します。

- **SAVSE をインストールして構成します。**インストールおよび設定情報については、Symantec ソフトウェアマニュアルを参照してください。498 ページの「SAVSE オプション」も参照してください。
- **SAVSE のクライアントライブラリをロードおよび構成します。**このためには、クライアントライブラリ `libicap.so` と設定ファイルを MTA に対して指定する必要があります (ファイルは作成する必要がある)。461 ページの「スパムのフィルタ処理ソフトウェアのクライアントライブラリをロードおよび構成する」を参照してください。
- **ウィルス用にフィルタ処理するメッセージを指定します。**ユーザー、ドメイン、またはチャネルごとにメッセージをフィルタ処理できます。463 ページの「フィルタ処理を行うメッセージを指定する」を参照してください。
- **ウィルスメッセージに対して実行するアクションを指定します。**ウィルスは、破棄したり、フォルダにファイリングしたり、件名にタグ付けしたりできます。469 ページの「スパムメッセージに対して実行するアクションを指定する」を参照してください。
- **必要に応じてほかのフィルタ設定パラメータを設定します。**471 ページの「MTA スパムフィルタオプション (`option.dat`)」を参照してください。

SAVSE の設定例

次の例では、ローカルメッセージストアに届くメッセージをテストして、ウィルスを含むメッセージを破棄します。最初の 3 つのステップは任意の順序で実行できます。

1. SAVSE の設定ファイルを作成します。

   このファイルの名前と場所は、手順 2 で指定したものです。ここでは SAVSE.opt という名前を使用します。このファイルの設定例を次に示します。

```
host=127.0.0.1
port=1344
mode=0
verdict=virus
debug=1
```

   host および port は、SAVSE プログラムが実行されているシステムの名前と、SAVSE プログラムが受信要求を待機するポート (SAVSE のデフォルトは 1344) です。mode=0 は、メッセージがウィルスを含んでいると認識された場合に、verdict によって指定された文字列 (この例の場合は virus という単語) が返されることを指定します。debug=1 はデバッグをオンにします。ICAP 設定パラメータについては、498 ページの表 14-6 を参照してください。

2. option.dat ファイルを作成します。次に例を示します。

```
! for Symantex Anti-virus Scan Engine
spamfilter1_config_file=/opt/SUNWmsgsr/config/SAVSE.opt
spamfilter1_library=/opt/SUNWmsgsr/lib/libicap.so
spamfilter1_optional=1
spamfilter1_string_action=data:,discard
```

   spamfilter1_config_files は、SAVSE 設定ファイルを指定します。

   spamfilter1_library は、SAVSE 共有ライブラリの場所を指定します。

   spamfilter1_optional=1 は、SAVSE プログラムによる失敗があった場合に MTA が操作を続行することを指定します。

   spamfilter1_string_action は、スパムメッセージに対して実行する Sieve アクションを指定します。この値は、ウィルスを含むメッセージが破棄されるように指定します。これはデフォルト値なので、値を変更する場合を除き、指定する必要はありません。

3. フィルタ処理するメッセージを指定します。

   ローカルメッセージストアに届くすべてのメッセージをフィルタ処理するには、ims-ms チャネルの destinationspamfilter1optin spam を追加して imta.cnf ファイルを変更します。

```
!
! ims-ms
ims-ms defragment subdirs 20 notices 1 7 14 21 28 backoff
"pt5m" "pt10m" "pt30m" "pt1h"  "pt2h" "pt4h" maxjobs 4 pool
IMS_POOL fileinto
$U+$S@$D destinationspamfilter1optin virus
ims-ms-daemon
```

4. 設定をコンパイルしなおし、サーバーを再起動します。MTA を再起動するだけでかまいません。stop-msg を実行する必要はありません。

```
# imsimta cnbuild
# imsimta restart
```

5. SAVSE が起動していることを確認します。

   自動的に起動しているはずですが、起動していない場合は、次のような起動コマンドを使用します。**/etc/init.d/symcscna start**

ほかの使用可能な設定

mode を 0 に設定して spamfilterX_null_option とともに使用すると、メッセージがスパムであると判明した場合に特定のフォルダにメッセージをファイリングするなど、ほかのアクションを実行できます。次に例を示します。

```
spamfilter1_null_option=data:,require "fileinto"; fileinto "VIRUS";
```

感染したメッセージをフォルダにファイリングすることは、ほとんどの場合不適切です。

mode を 1 に設定すると、アクションを起動できます。たとえば、mode を 1 に設定して、MTA の spamfilterX_string_action オプションを次のように設定すると、スパムの結果を拒否メッセージに含めることができます。

```
spamfilter1_string_action=data:,require "reject"; reject "Message
contained a virus [$U]";
```

reject アクションはウィルスを差出人に送り返すため、fileinto と同様、このアクションを使用してウィルスを処理する方法は不適切です。

option.dat ファイルに行を追加して、スパムメッセージのヘッダーにタグを追加することもできます。次に例を示します。

```
spamfilter1_string_action=data:,addtag "[SPAM detected!]";
```

mode を 2 に設定すると、メッセージにウィルスが含まれると判定されるかどうかにかかわらず、アクションを実行できます。あとでテストできるヘッダーフィールドが追加されているところは、明らかに mode 2 のアプリケーションです。

```
spamfilterX_string_action=data:,require ["addheader"];addheader "$U"
```

SAVSE オプション

SAVSE オプションファイルは実際に、かなり一般的な ICAP オプションファイルです。オプションファイルの名前および場所は、option.dat の spamfilterX_config_file で設定されています。SAVSE オプションファイルは、option=value という形式の行から構成されています。HOST というオプションが 1 つ必要です。このオプションは、ICAP フィルタ処理サーバーが動作しているシステムの名前に設定する必要があります。ICAP サーバーがローカルホストで動作していても、このオプションを設定する必要があります。SAVSE オプションファイルを以下に示します。

表 14-6 ICAP オプション

オプション	説明	デフォルト
debug	ICAP インタフェースモジュールからのデバッグ出力を有効または無効にします。0 または 1。	0
field	ICAP の結果のプレフィックスを指定します。SAVSE の結果文字列は次のようになります。 Virus-Test:False Virus-Test:True; W32.Mydoom.A@mm.enc このオプションは、結果の Virus-Test: 部分の変更方法を提供します。空の field 値が指定されると「:」が削除されることに注意してください。	Virus-Test:
host	ICAP フィルタ処理サーバーが動作しているシステムの名前	localhost

表 14-6　ICAP オプション (続き)

オプション	説明	デフォルト
mode	ICAP フィルタ結果から判定情報への変換を制御します。つまり、メッセージ処理後に返す文字列情報を指定します。次の 4 種類のモードが使用可能です。詳細は、500 ページの「ICAP の mode オプション」を参照してください。 0 - メッセージにウィルスが含まれる場合は、判定文字列 (verdict オプションで指定される文字列) を返します。MTA オプション spamfilterX_string_action を使用して、verdict 文字列が返された場合の処理を指定できます。verdict オプションが空である場合や設定されていない場合は、NULL 判定を返します。MTA オプション spamfilterX_null_action を使用して、NULL 判定が返された場合に、メッセージを破棄するデフォルトアクションを置き換えるときの処理を指定できます。 メッセージにウィルスが含まれていない場合は、デフォルトの文字列が返されます。デフォルトの文字列とは、設定不可能な文字列で、何のアクションもとらず通常どおり配信することを意味します。 1 - メッセージにウィルスが含まれていると判明した場合は ICAP の結果文字列を返します。メッセージにウィルスが含まれていない場合は、デフォルトの文字列が返されます。デフォルトの文字列とは、何のアクションもとらず通常どおり配信することを意味します。以下に、ICAP 結果文字列の例を 2 つ示します。 VIRUS TEST: FALSE VIRUS-TEST: TRUE; W32.Mydoom.A@mm.enc 2 - ICAP 結果文字列を無条件に返します。デフォルト判定または NULL 判定が返されることはなく、verdict オプションが使用されることはありません。この設定は、メッセージにウィルスが含まれると判定されるかどうかにかかわらず、アクションを実行する必要がある場合を対象にしています。あとでテストできるヘッダーフィールドが追加されているところは、明らかに mode 2 のアプリケーションです。 spamfilterX_string_action=data:,require ["addheader"];addheader "$U" 3 - メッセージにウィルスが含まれていると判明した場合は ICAP の結果文字列を返します。ウィルスが含まれていない場合は verdict オプションで指定された verdict 文字列を返します。この設定は、ウィルスが発見されたらある 1 つのアクションを実行し、発見されなかったら別のアクションを実行する必要がある場合を対象としています。ICAP 結果文字列に対応するアクションを制御するには、spamfilterX_verdict_*n* オプションと spamfilterX_action_*n* オプションの一致ペアを使用します。verdict 文字列に対するアクションを制御するには、spamfilterX_string_action を使用します。	0
port	ICAP サーバーが動作しているポート番号を指定します。	1344

表 14-6　ICAP オプション (続き)

オプション	説明	デフォルト
SOCKS_HOST	文字列。中間にある SOCKS サーバーの名前を指定します。このオプションが指定されている場合、指定された SOCKS サーバーを介して ICAP 接続が確立され、直接には接続されません。	""
SOCKS_PORT	整数。中間にある SOCKS サーバーが動作しているポートを指定します。	1080
SOCKS_PASSWORD	文字列。SOCKS サーバーを介した接続を確立するために使用するパスワードを指定します。ユーザー名およびパスワードが必要かどうかは、SOCKS サーバーの設定によって異なります。	""
SOCKS_USERNAME	文字列。SOCKS サーバーを介した接続を確立するために使用するユーザー名を指定します。	""
verdict	MODE 0 および 3 で使用される判定文字列を指定します。	""

ICAP の mode オプション

メッセージの処理後、ICAP ウィルス防止プログラムは SASVE と同様に、メッセージにウィルスが含まれているかどうかを判定します。mode オプションを使用すると、この判定を示すために ICAP プログラムから返す文字列を指定できます。文字列の選択肢は、NULL、デフォルト、ICAP の結果文字列、または verdict オプションで指定された verdict 文字列です。デフォルトの文字列とは、NULL、ICAP の結果文字列、または verdict で指定された文字列ではなく、プログラムによって返されるその他の設定不可能な文字列です。mode の動作の概要を次の表で説明します。

表 14-7　ICAP の mode オプションに対応して返される判定文字列

verdict の設定	ウィルスかどうか	**mode=0**	**mode=1**	**mode=2**	**mode=3**
verdict="" (設定なし)	はい	NULL	ICAP の結果	ICAP の結果	ICAP の結果
	いいえ	デフォルト	デフォルト	ICAP の結果	デフォルト
verdict= 文字列	はい	verdict 文字列	ICAP の結果	ICAP の結果	ICAP の結果
	いいえ	デフォルト	デフォルト	ICAP の結果	verdict 文字列

1 列目は、verdict オプションが設定されているかどうかを示します。2 列目は、メッセージにウィルスが含まれているかどうかを示します。mode の列は、各モードに対応して返される文字列を示します。たとえば、verdict が設定されておらず mode が 0 に設定されている場合、メッセージにウィルスが含まれていなければ、ICAP プログラムからデフォルト文字列が返されます。verdict が WARNING VIRUS! に設定されていて mode が 0 に設定されている場合、メッセージにウィルスが含まれていれば、ICAP プログラムは WARNING VIRUS! という文字列が返されます。

Sieve 拡張のサポート

標準の Sieve 機能以外にも、Messaging Server は多くの拡張機能を提供します。拡張機能には addheader、addtag、spamtest、spamadjust などがあります。addheader と addtag については、485 ページの「スパムメッセージに SpamAssassin スコアを含むヘッダーを追加する」および 487 ページの「件名行に SpamAssassin の結果文字列を追加するには」を参照してください。ここでは、spamtest と spamadjust について説明します。

これらの拡張は、管理者に、別のしきい値を設定する機能と、SpamAssassin の判定を無効にするホワイトリストを設定する機能を提供します。この 2 つの拡張を組み合わせて、特定のメッセージの差出人に応じて別のしきい値を持つようにすることもできます。spamadjust は非標準のアクションです。spamtest については ftp://ftp.isi.edu/in-notes/rfc3685.txt を参照してください。

spamtest を使用して、Sieve [RELATIONAL] 拡張を使った特定の値に対する SpamAssassin スコアを、"i;ascii-numeric" 比較子と比較することができます。SpamAssassin スコアは通常は実数ですが、spamtest は、最初にスコアを近似整数に丸めることで、スコアを 0 から 10 の整数値にします。0 未満の値は強制的に 0 になり、10 を超える値は強制的に 10 になります。最後に、Messaging Server が維持するテキスト文字列が付加されて、spamtest テストが示すテスト文字列が生成されます。

spamadjust は、現在のスパムスコアの調整に使われます。このアクションは、実際の数値に対してスキャンされる単一の文字列引数をとります。この値が現在のスパムスコアの調整に使われます。また、文字列全体も現在のスコアテキスト文字列に付加されます。以下の例では、この文字列は "undisclosed recipients" です。

複数の spamadjust アクションを指定でき、それぞれが現在のスコアに追加されます。また、スコア値は常に 0 で始まります。符号付き数値を指定でき、現在のスコアを小さくしたり大きくしたりできます。spamadjust に対する require 句はなく、代わりに spamtest 拡張を列挙する必要があります。

たとえば、SpamAssassin の MODE を 2 に設定した spamadjust を使用すると、次のようになります。

```
spamfilterX_string_action=data:,require ["addheader"];addheader "$U"
```

システムレベルの Sieve フィルタは、特定のタイプのヘッダーをチェックし、それが見つかった場合は SpamAssassin スコアに 5 を追加することによって、SpamAssassin を変更します。

```
spamfilter1_string_action=require "spamtest";                                ¥
if header :contains ["to", "cc", "bcc", "resent-to", "resent-cc", "resent-bcc"]  ¥
                    ["<undisclosed recipients>", "undisclosed.recipients"]      ¥
{spamadjust "+5 undisclosed recipients";}
```

最後に、ユーザーレベルの Sieve スクリプトは、結果を示す値のテスト、スパムであることが確実なメッセージの破棄、スパムと思われるメッセージのファイリング、ローカルドメインのアドレスからのメッセージ通過の許可を行うことができます。

```
spamfilter1_string_action=require ["spamtest", "relational",  ¥
"comparator-i;ascii-numeric", "fileinto"];                    ¥
if anyof (address :matches "from" ["*@siroe.com",             ¥
                                   "*@*.siroe.com"])          ¥
    {keep;}                                                   ¥
elsif spamtest :value "ge" :comparator "i;ascii-numeric" "8" ¥
    {discard;}                                                ¥
elsif spamtest :value "ge" :comparator "i;ascii-numeric" "5" ¥
    {fileinfo "spam-likely";}                                 ¥
else                                                          ¥
   {keep;}
```

第 15 章

LMTP 配信

Sun Java System Messaging Server の MTA では、LMTP (Local Mail Transfer Protocol、RFC 2033 で定義) を使用して、複数層のメッセージングサーバーが展開されている環境でメッセージストアに配信できます。受信リレーとバックエンドメッセージストアが使用されるこのような環境では、メーリングリストの拡大などのアドレス拡張と自動返信や転送などの配信方法に関してリレーが重要な役割を果たします。バックエンドストアへの配信はこれまで SMTP 上で行われてきました。SMTP では、バックエンドシステムで LDAP ディレクトリの受取人アドレスを再度調べる必要があるため、MTA の全機能が使用されます。速度と効率性を向上するために、MTA では SMTP ではなく LMTP を使用してバックエンドストアにメッセージを配信できます。Sun Java System Messaging Server の LMTP サーバーは、汎用 LMTP サーバーとしてではなく、リレーとバックエンドメッセージストア間のプライベートプロトコルとして機能します。説明をわかりやすくするために、2 層展開を例にとっています。

注	LMTP は多層展開での使用を目的として設計されています。LMTP を単一システム展開で使用することはできません。また、Messaging Server の LMTP サービスは、そのままでは LMTP サーバーまたはほかの LMTP クライアントで動作するように設計されていません。

この章には、以下の節があります。

LMTP 配信の特徴

MTA の LMTP サーバーがバックエンドメッセージストアへの配信に関して効率性が高い理由は次のとおりです。

- バックエンドストアにかかる負荷が減少する。

  リレーは水平方向に拡張可能ですが、バックエンドストアはそうでないため、可能な限り多くの処理をリレーに割り当てることをお勧めします。

- LDAP にかかる負荷が減少する。

  大規模なメッセージング展開では、LDAP インフラストラクチャは制限要因であることがよくあります。

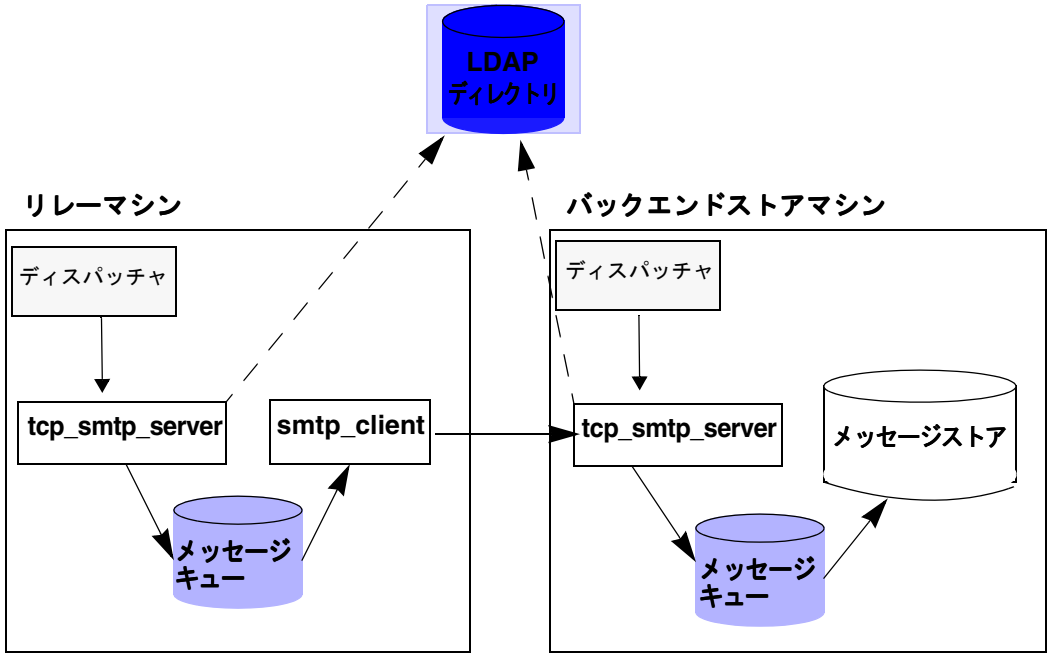
- メッセージキューの数が減少する。

  リレーとバックエンドストアの両方にキューがあると、メッセージング展開の管理者が不着メールを見つける作業はいっそう困難になります。

LMTP を使用しない 2 層展開でのメッセージ処理

図 15-1 に、LMTP を使用しない 2 層展開でのメッセージ処理を示します。

図 15-1　LMTP を使用しない 2 層展開



LMTP を使用しない場合で、ストアシステムの前にリレーを配備した 2 層展開では、受信メッセージの処理は、リレーマシンのディスパッチャによってピックアップされ、`tcp_smtp_server` プロセスにハンドオフされた SMTP ポートの接続から始まります。このプロセスでは、受信メッセージに対して次のような処理が行われます。

- ディレクトリ内のユーザーを検索する
- ユーザーがこの電子メール展開でホストされるドメインに属しているかどうかを判断する
- ユーザーがそのドメインで有効なユーザーであるかどうかを判断する
- エンベロープアドレスを `@mailhost:user@domain` という形式に書き換える
- メールホストに配信するためにメッセージをキューに入れる

次に、メールメッセージはキューから `smtp_client` プロセスに引き継がれ、メールホストに送信されます。メールホスト上では、非常によく似た処理が行われます。SMTP ポートへの接続がディスパッチャによってピックアップされ、`tcp_smtp_server` プロセスにハンドオフされます。このプロセスでは、メッセージに対して次のような処理が行われます。

- ディレクトリ内のユーザーを検索する

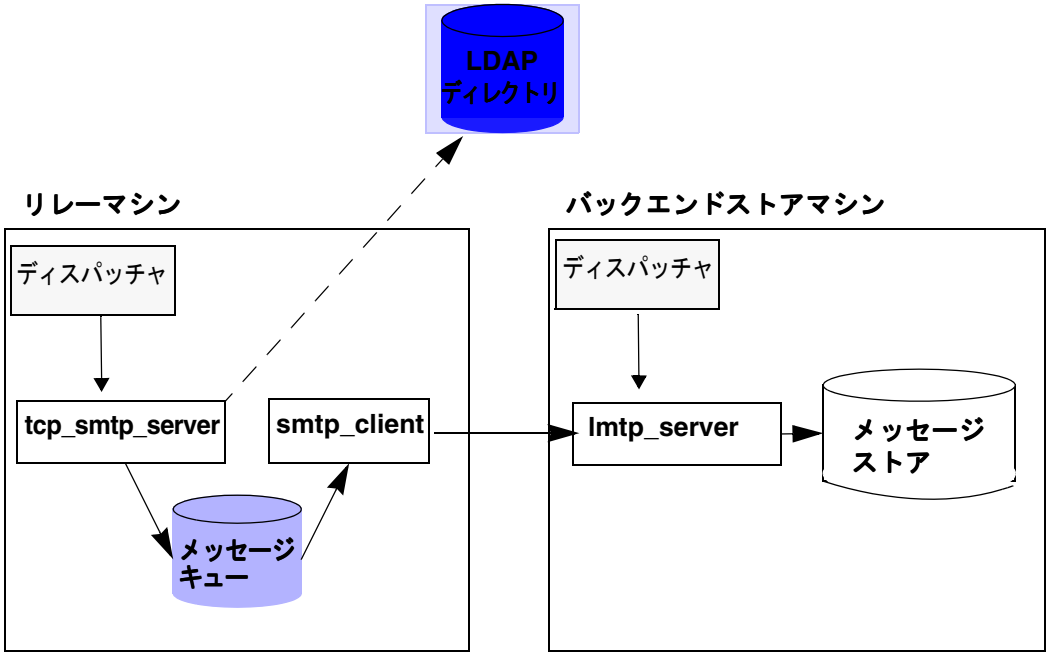
- ユーザーがこの電子メール展開でホストされるドメインに属しているかどうかを判断する
- ユーザーがそのドメインで有効なユーザーであるかどうかを判断する
- メッセージを ims_ms チャネルに送信するためにエンベロープアドレスを書き換える
- ストアに配信するためにメッセージをキューに入れる

次に、メールメッセージは ims_ms プロセスに引き継がれ、ストアへの配信が試行されます。ここでは、キューに入れる処理が 2 回実行されています。また各 MTA はそれぞれ LDAP 検索を実行しています。

LMTP を使用する 2 層展開でのメッセージ処理

図 15-2 に、LMTP を使用する 2 層展開でのメッセージ処理を示します。

図 15-2　LMTP を使用する 2 層展開



LMTP が配備されている場合、リレーマシンの SMTP ポートへの接続がディスパッチャによってピックアップされ、tcp_smtp_server プロセスにハンドオフされます。このプロセスでは、受信メッセージに対して次のような処理が行われます。

- ディレクトリ内のユーザーを検索する
- ユーザーがこの電子メール展開でホストされるドメインに属しているかどうかを判断する
- ユーザーがそのドメインで有効なユーザーであるかどうかを判断する
- ユーザーのメールボックスをホストしているバックエンドメッセージストアのマシンを特定する
- @mailhost:uid@domain.LMTP または @mailhost:uid@domain.LMTPNATIVE という形式にアドレスを書き換える
- メールホストに配信するためにメッセージをキューに入れる

user@domain.LMTP および user@domain.LMTPNATIVE という形式のアドレスは、前者は tcp_lmtp チャネル、後者は tcp_lmtpnative チャネルを介してメッセージストアシステムにルーティングされます。これらのチャネルは、SMTP ではなく LMTP を使用してバックエンドメッセージストアと通信します。ストアマシンでは、LMTP ポートへの接続がディスパッチャに受信され、lmtp_server プロセスにハンドオフされます。次に、LMTP サーバーによってメッセージがユーザーのメールボックスまたは UNIX のネイティブメールボックスに挿入されます。メッセージの配信が成功すると、そのメッセージはリレーマシン上でキューから取り出されます。成功しなかった場合は、メッセージはリレーマシンに残ります。メッセージストアの LMTP プロセスでは、アドレスやメッセージの処理に MTA の機能は使用されません。

LMTP の概要

ほとんどの場合、MTA 自体は基本的にバックエンドサーバーで使用されることはありません。必要な MTA コンポーネントは次のとおりです。

- ディスパッチャ
- `libimta`
- LMTP サーバー
- `imta.cnf` ファイル
- `mappings` ファイル
- `imta.tailor` ファイル

ディスパッチャには MTA 設定ファイルが必要ですが、ファイルは非常に短くすることができます。ディスパッチャの下で実行する LMTP サーバーを起動できるようにするため、ディスパッチャはバックエンドサーバーで実行する必要があります。ディスパッチャと LMTP サーバーは `libimta` のさまざまな機能を使用するので、これもバックエンドサーバーに存在する必要があります。

LMTP サーバーでは、通常ならば実行される MTA のキューの出し入れ機能、ヘッダー処理、またはアドレス変換が実行されません。メッセージの内容とアドレスについての処理は、すべてリレーシステムで実行されます。処理後は、メッセージストアに送信される正確な形式で、メッセージストアが必要とする形式の配信アドレスがすでに付けられたメッセージが LMTP サーバーに提示されます。ユーザーの制限容量など、メッセージがストアに配信される際に通常使用可能な追加受取人情報は、受取人アドレスとともに LMTP パラメータとして提示されます。配信試行が失敗した場合は、メッセージはリレーシステムの LMTP キューに残ります。

LMTP 配信の設定

LMTP 配信メカニズムを設定するには、リレーマシンの両方とバックエンドストア上での設定が必要です。リレー上では、`DELIVERY_OPTIONS` MTA オプション (`option.dat` にある) を変更して、ストアに配信されるメッセージが LMTP チャネルに渡されるようにする必要があります。バックエンドストアはディスパッチャを使って設定する必要がありますが、ジョブコントローラは必要ありません。LMTP サーバーを実行するには、このディスパッチャを設定する必要があります。

通常の多層配備では、ユーザーはさまざまなバックエンドメッセージストアのマシン上にプロビジョニングされます。これらのバックエンドマシンの 1 つ以上で LMTP が有効でない場合もあるため、フロントエンドリレーは、どのストアマシンが LMTP を認識するかに注意を払う必要があります。これを行うには、汎用データベース機能を使って、LMTP 配信を受け入れるように設定されているメッセージストアの名前を明示的に指定します。

LMTP を使って受信 MTA リレーを設定するには、次の手順に従います。

LMTP を使うように受信 MTA リレーを設定するには、次の手順に従います。

1. option.dat に次の行を追加して、テキストデータベースを有効にします。

   ```
   USE_TEXT_DATABASES=1
   ```

   この手順では、汎用データベースの平文テキストファイルの使用が MTA で有効になっています。すでに汎用データベースを使用している場合は、この手順を飛ばしてもかまいません。

2. 汎用データベースのテキストファイルを作成または変更します。

```
# cd /opt/SUNWmsgsr/config/
# vi general.txt

LMTP_CS|msg-store.siroe.com   lmtpcs-daemon
LMTP_CS|name-1-lmtp-store.siroe.com  lmtpcs-daemon
LMTP_CS|name-2-lmtp-store.siroe.com  lmtpcs-daemon
..
..
LMTP_CN|Zmar.Talek@siroe.com  lmtpcn-daemon
..
LMTP_CN|Fred.Bloggs@siroe.com  lmtpcn-daemon

# chown mailsrv general.txt
```

2 つのタイプのエントリがあります。1 つは lmtpnative チャネルに対するユーザー固有の配信を処理するためのものであり、もう 1 つは、tcp_lmtpcs チャネル経由の配信のストア全体での設定を処理するためのものです。

3. options.dat ファイルで、DELIVERY_OPTIONS 変数を作成または変更します。

   DELIVERY_OPTIONS の値を変更する必要があります。配信オプションの現在のデフォルトは、次のようになっています。

```
DELIVERY_OPTIONS=¥
     *mailbox=$M%$¥$2I$_+$2S@ims-ms-daemon,¥
     &members=*, ¥
     *native=$M@native-daemon, ¥
     *unix=$M@native-daemon, ¥
     /hold=$L%$D@hold-daemon,¥
     &file=+$F@native-daemon, ¥
     &@members_offline=*,¥
     program=$M%$P@pipe-daemon,¥
     #forward=**, ¥
     *^!autoreply=$M+$D@bitbucket
```

   これを次のように変更します。

```
DELIVERY_OPTIONS=¥
    #*mailbox=@$X:$M$_+$2S%$¥$2I@ims-ms-daemon,¥
    #&members=*,¥
    #*native=@$X:$M,¥
    #*unix=@$X:$M,¥
    #/hold=$L%$D@hold,¥
    #*file=@$X:+$F,¥
    #&@members_offline=*,¥
    #program=$M%$P@pipe-daemon,¥
    #forward=**, ¥
    #*^!autoreply=$M+$D@bitbucket
```

   メールボックス配信オプションのパターンが変更されたことと、リレーマシンでのアクションを強制するために自動返信の配信オプションの先頭に # 文字が付加されたことに注意してください。$X の置換により、ユーザーの mailhost 属性の値が挿入されます。これによってソースルートアドレスが生成されます。

   また、ネイティブ、UNIX、ファイル、およびプログラムの各配信方法を利用するには、ターゲットマシンで MTA が稼働している必要があります。

4. LMTP 書き換えルールを imta.cnf ファイルに追加します。

```
# cd /opt/SUNWmsgsr/config/
# cp imta.cnf imta.cnf.orig
# vi imta.cnf

!
! pipe
.pipe-daemon $U%$H.pipe-daemon@pipe-daemon
!
! tcp_local
! トップレベルインターネットドメインのルール
<IMTA_TABLE:internet.rules
!
! マッピング検索を内部 IP アドレスに対して実行する
[] $E$R${INTERNAL_IP,$L}$U%[$L]@tcp_intranet-daemon
!
! general.txt の検索を LMTP ホストに対して実行する
.domain-name.com $S$U%$H$D@$(LMTP_CN|$U@$H$D)
.domain-name.com $S$U%$H$D@$(LMTP_CS|$H$D)
!
! tcp_intranet
! マッピング検索を内部 IP アドレスに対して実行する
[] $E$R${INTERNAL_IP,$L}$U%[$L]@tcp_intranet-daemon
.domain-name.com $U%$H.domain-name.com@tcp_intranet-daemon
```

この手順では、1 組の書き換えルールにより、アドレスの発信元経路指定部分が LMTP 配信のためのエントリと一致するかどうかを確認するため、汎用データベースのタグ付けのプローブが実行されます。手順 2 で作成した general.txt ファイルには、適切なチャネルを経由してバックエンドメッセージストアに配信を指定する、タグ付きエントリがあります。ここで、書き換えルールにある $S は、アドレスに発信元経路指定が含まれているときのみ適用されることを意味します。汎用データベース内のエントリに一致するものがあれば、書き換えルールは成功し、メッセージは、LMTP 経由で配信を行う tcp_lmtp*X* チャネル経由で、発信元経路指定のバックエンドホストに送信されます。

一致するものがなければ、その他の書き換えルールで一致するものが見つかるまで、書き換えプロセスが継続します。ほとんどの場合、汎用データベースのプローブを介して一致するものが見つからない場合、メッセージは SMTP 経由で配信を行う tcp_intranet チャネルを経由して経路指定されます。

5. imta.cnf に新しいチャネルブロックを追加します。

   また、imta.cnf ファイルのチャネル定義セクションにある lmtp および lmtpn チャネルにチャネル定義を組み込む必要もあります。次に例を示します。

```
! tcp_lmtpcs (LMTP クライアント - ストア)
tcp_lmtpcs defragment lmtp port 225 nomx single_sys subdirs 20
maxjobs 7 pool SMTP_POOL dequeue_removeroute
lmtpcs-daemon
!
! tcp_lmtpcn (LMTP クライアント - ネイティブ)
tcp_lmtpcn defragment lmtp port 226 nomx single_sys subdirs 20
maxjobs 7 pool SMTP_POOL dequeue_removeroute
!lmtpcn-daemon
```

6. 設定の変更をコミットします。

```
# cd /opt/SUNWmsgsr/bin
# ./imsimta refresh

Compiled configuration done

Killing Dispatcher : 23021

Dispatcher startup requested

Job Controller shutdown requested

Job Controller startup requested
```

注 必ず、LMTP チャネル上の lmtp チャネルキーワードを使用してください。LMTP チャネル上の smtp および lmtp チャネルキーワードを両方とも使用しないようにしてください。また、デフォルトでは LMTP チャネル定義はコメント行とされていることにも注意してください。LMTP を機能させたい場合は、コメント行を解除する必要があります。

Sun Java System Messaging Server のユーザーと Sun Java System Messaging Server 以外のユーザーが混在している組織では、Sun Java System Messaging Server 以外のマシンのユーザーを示すことができる必要があります。このようなユーザーの delivery オプションを mailbox に設定することはできません。このようなユーザーの delivery オプションは forward に設定する必要があります。転送先アドレスは、ソースルートされた形式であることが必要です。使用できるアドレスの例を次に示します。

```
mailForwardingAddress: @nonSJSMSHost.siroe.com:first.last@siroe.com
```

または

```
mailForwardingAddress: @nonSJSMSHost.siroe.com:login@siroe.com
```

または

```
mailForwardingAddress: @nonSJSMSHost.siroe.com:first.last@nonSJSMSHost.siroe.com
```

または

```
mailForwardingAddress: @nonSJSMSHost.siroe.com:login@nonSJSMSHost.siroe.com
```

つまり、次のようになります。

```
@nonSJSMSHost.siroe.com:nonSJSMSHost に認識されるアドレス
```

MTA を使用せずに LMTP を使用するバックエンドストアを設定する

バックエンドストアは、LMTP を使用してメッセージを受信する場合、MTA は不要です。つまり、バックエンドストアは、ジョブコントローラも MTA に関連するアドレス書き換え機能も持ちません。ただし、ディスパッチャと簡単な MTA 設定は必要です。具体的には、dispatcher.cnf ファイルと mappings ファイルが必要です。これらは、MTA 設定の重要な部分のみを含んでいます。

dispatcher.cnf ファイルには、次の内容が含まれている必要があります。

```
! rfc 2033 LMTP サーバー - ストア
!
[SERVICE=LMTPSS]
PORT=225
IMAGE=IMTA_BIN:tcp_lmtp_server
LOGFILE=IMTA_LOG:tcp_lmtpss_server.log
PARAMETER=CHANNEL=tcp_lmtpss
STACKSIZE=2048000
! 次の行のコメントを解除し、ディスパッチャが特定のインタフェース (HA 環境など)
! で待機する必要がある場合は、INTERFACE_ADDRESS を適切な
! ホスト IP ( ドットで 4 つに区切られた形式 ) に設定する。
!INTERFACE_ADDRESS=
!
! rfc 2033 LMTP サーバー - ネイティブ
!
[SERVICE=LMTPSN]
PORT=226
IMAGE=IMTA_BIN:tcp_lmtpn_server
LOGFILE=IMTA_LOG:tcp_lmtpsn_server.log
PARAMETER=CHANNEL=tcp_lmtpsn
STACKSIZE=2048000
! 次の行のコメントを解除し、ディスパッチャが特定のインタフェース (HA 環境など)
! で待機する必要がある場合は、INTERFACE_ADDRESS を適切な
! ホスト IP ( ドットで 4 つに区切られた形式 ) に設定する。
!INTERFACE_ADDRESS=
```

デフォルトでは、dispatcher.cnf ファイルの LMTP サービスはコメントアウトされています。LMTP を使用するには、それらのコメントを解除する必要があります。

通常のディスパッチャオプションである MAX_CONNS、MAX_PROCS、MAX_LIFE_CONNS、および MAX_LIFE_TIME も設定できますが、使用しているハードウェアに適した設定にする必要があります。

PORT_ACCESS マッピングは重要です。バックエンドサーバーへの LMTP の実装は、Sun Java System Messaging Server リレーとバックエンドストア間のプライベートプロトコルとして機能します。PORT_ACCESS マッピングを使用してこのようなリレーがこれらのサービスに確実に接続できるようにする必要があります。マッピングファイルは次のようになります。

```
PORT_ACCESS

  TCP|*|225|1.2.3.4|* $Y
  TCP|*|226|1.2.3.4|* $Y
  TCP|*|225|1.2.3.5|* $Y
  TCP|*|226|1.2.3.5|* $Y
  TCP|*|*|*|*   $N500$ Do$ not$ connect$ to$ this$ machine
```

PORT_ACCESS マッピングテーブルで指定されているサンプルの IP アドレスは、バックエンドストアに接続しているネットワーク上にあるリレーの IP アドレスに置き換える必要があります。

imta.cnf ファイルが存在する必要がありますが、このファイルは設定を完全なものにするためにのみ存在します。もっとも小さい imta.cnf ファイルは、次のチャネル定義で構成されています。

```
! tcp_lmtpss (LMTP サーバー - ストア)
tcp_lmtpss lmtp
tcp_lmtpss-daemon

!
! tcp_lmtpsn (LMTP サーバー - ネイティブ)
tcp_lmtpsn lmtp
tcp_lmtpsn-daemon
```

デフォルトでは、LMTP チャネル定義はコメントアウトされています。LMTP を機能させたい場合は、コメント行を解除する必要があります。

LMTP を使用してメッセージをメッセージストアと完全な MTA のあるバックエンドシステムに送信するためのリレーを設定する

バックエンドストアに全機能を備えた MTA を配備しながら、LMTP を使用して負荷を抑えたい場合があります。たとえば、バックエンドストアでプログラム配信を行う場合です。この場合、509 ページの「LMTP を使って受信 MTA リレーを設定するには、次の手順に従います。」の説明に従って、LMTP を使用してバックエンドストアに配信するリレーを設定する必要があります。

完全な MTA を備えたバックエンドメッセージストアシステムに LMTP を設定する

バックエンドストアのメッセージングシステムの設定から LMTP によるストアへの直接配信の設定に変更する場合、必要なのは `dispatcher.cnf` ファイルの最後に次の行を追加することだけです。

```
! rfc 2033 LMTP サーバー - ストア
!
[SERVICE=LMTPSS]
PORT=225
IMAGE=IMTA_BIN:tcp_lmtp_server
LOGFILE=IMTA_LOG:tcp_lmtpss_server.log
PARAMETER=CHANNEL=tcp_lmtpss
STACKSIZE=2048000
! 次の行のコメントを解除し、ディスパッチャが特定のインタフェース (HA 環境など )
! で待機する必要がある場合は、INTERFACE_ADDRESS を適切な
! ホスト IP ( ドットで 4 つに区切られた形式 ) に設定する。
!INTERFACE_ADDRESS=
!
! rfc 2033 LMTP サーバー - ネイティブ
!
[SERVICE=LMTPSN]
PORT=226
IMAGE=IMTA_BIN:tcp_lmtpn_server
LOGFILE=IMTA_LOG:tcp_lmtpsn_server.log
PARAMETER=CHANNEL=tcp_lmtpsn
STACKSIZE=2048000
! 次の行のコメントを解除し、ディスパッチャが特定のインタフェース (HA 環境など )
! で待機する必要がある場合は、INTERFACE_ADDRESS を適切な
! ホスト IP ( ドットで 4 つに区切られた形式 ) に設定する。
!INTERFACE_ADDRESS=
!
```

デフォルトでは、dispatcher.cnf ファイルの LMTP サービスはコメントアウトされています。LMTP を使用するには、それらのコメントを解除する必要があります。また、LMTP ポート番号は単なる例であり、任意の番号を選択できます。

これは、LMTP のみに関してバックエンドストアを設定する場合について前述した際の dispatcher.cnf ファイル全体と同じです。LMTP のみのバックエンドストアで説明したように、このマッピングファイルには、PORT_ACCESS マッピングも必要です。

LMTP プロトコルの実装例

この節では、LMTP ダイアログのサンプルを使用して、そこで示される内容を説明します。リレー上の LMTP クライアントでは、標準の LMTP プロトコルを使用してバックエンドストア上の LMTP サーバーと交信します。ただし、このプロトコルは特定の方法で使用されます。次に例を示します。

```
---> LHLO
<---250 OK
```

LHLO メッセージに対してアクションは実行されません。返信は常に 250 OK です。

```
---> MAIL FROM:address size=messageSizeInBytes
<---250 OK
```

差出人のアドレスに対するチェックまたは変換は行われません。size= パラメータにより配信されるメッセージのサイズがバイト単位で指定されます。これは、プロトコルで記述されているサイズと同じサイズです。必ずしもメッセージの正確なサイズではありませんが、実際のサイズがこれを超えることはありません。LMTP サーバーによって、メッセージの受信に必要な、このサイズのメモリバッファが割り当てられます。

```
---> RCPT TO:uid+folder@domain xquota=size,number xdflg=xxx
<---250 OK
```

受信される際に受取人のアドレスのチェックは行われませんが、受取人の一覧が後で使用するために作成されます。プライマリドメインの uids では、アドレスの @domain の部分は省略されます。また、+folder の部分はオプションです。これは MTA のメッセージストアチャネルで使用されるものと同じアドレス形式です。

xquota= パラメータでは、最大合計サイズと最大メッセージ数で構成されるユーザーのメッセージ制限容量が指定されます。この情報は、アドレス変換を実行するためにユーザーについての LDAP 検索を実行している間に取得されたもので、MTA によって提供されます。また、この情報は、ディレクトリと同期化されたメッセージストアで制限容量の情報を保持するために使用されます。制限容量の情報を取得しても、パフォーマンスヒットが追加で発生することはありません。

xdflg= パラメータではビットフィールドとして解釈される数値が指定されます。このビットによってメッセージの配信方法が制御されます。たとえば、値が 2 であるビットが設定されている場合、ユーザーが制限容量を超えていてもメッセージの配信が保証されます。xdflg は内部パラメータであり、それに含まれるビットは予告なしに変更または追加されることがあるので注意してください。サーバーでこの拡張機能を使用してほかのクライアントをサポートしたり、ほかのサーバーでこのパラメータとともにクライアントを使用することはできません。

この対話は何度も繰り返されます (受取人ごとに 1 回実行)。

```
--->DATA
---> < メッセージテキスト >
--->.
```

次に、SMTP の場合と同じように、LMTP クライアントからメッセージ全体がドット付きで送信されます。行にある単独のドット (.) でメッセージは終わります。メッセージサイズが超過している場合、LMTP サーバーは次の内容を送信します。

```
<--- 500 message too big
```

その後接続を終了します。

メッセージが正しく受信された場合、LMTP サーバーは RCPT TO: 行で指定されている各受取人のステータスを LMTP クライアントに返します。たとえば、メッセージの配信が成功した場合の応答は次のようになります。

```
<--- 250 2.5.0 address OK
```

この address は RCPT TO: 行に表示されたアドレスです。

交信は別の MAIL FROM: 行と繰り返されるか、あるいは次の対話で終了します。

```
---> quit
<---221 OK
```

表 15-1 に、各受取人のステータスコードを示します。この表には 3 つの列があり、最初の列にショートコード、2 番目の列にそれと同義のロングコード、3 番目の列にステータステキストを示します。2.x.x ステータスコードは成功コード、4.x.x コードは再試行可能なエラー、5.x.x コードは再試行不能なエラーです。

表 15-1 受取人の LMTP ステータスコード

ショートコード	ロングコード	ステータステキスト
250	2.5.0	OK
420	4.2.0	Mailbox Locked
422	4.2.2	Quota Exceeded
420	4.2.0	Mailbox Bad Formats
420	4.2.0	Mailbox not supported
430	4.3.0	IMAP IOERROR
522	5.2.2	Persistent Quota Exceeded
523	5.2.3	Message too large
511	5.1.1	mailbox nonexistent
560	5.6.0	message contains null
560	5.6.0	message contains nl
560	5.6.0	message has bad header
560	5.6.0	message has no blank line

これ以外の場合は、メッセージストアのメールボックス、ネイティブ (したがって UNIX も)、およびファイルの配信オプションに変更があります。これらのルールの目的は、メッセージが適切な LMTP チャネルを介してバックエンドサーバーに送信されるアドレスを生成することです。生成されたアドレスは、次の形式のソースルートされたアドレスになります。

`@sourceroute:`*localpart@domain*

第 16 章

不在メッセージの自動返信

電子メールへの応答として自動的に生成される返信 (自動返信)、特に不在メッセージを処理するために、MTA では MDN (Message Disposition Notification) および Sieve スクリプト言語が使用されます。MDN は、MTA によって差出人またはポストマスター (あるいはその両方) に送信される電子メールメッセージであり、メッセージの配信状態について報告するものです。MDN は、開封確認、確認通知、受信通知、配信確認とも呼ばれます。Sieve は、メールフィルタの作成に使用される簡単なスクリプト言語です。

この章では、不在返信メッセージの自動返信のメカニズムについて説明します。ほとんどの場合、デフォルト設定を変更する必要はありませんが、不在処理がバックエンドメッセージストアではなく MTA リレーマシンで実行されるようにご使用のシステムを設定することもできます。

この章には、以下の節があります。

不在返信メッセージの自動返信の概要

不在処理の Sieve スクリプトは、さまざまな LDAP Vacation 属性から自動的に生成されます (528 ページの「不在返信メッセージの自動返信の属性」を参照)。Sieve スクリプトを明示的に指定して柔軟性を高めることもできます。不在メッセージ追跡の基本手段は、目的の受取人ごとに 1 つあるファイルの集合です。このファイルには、各差出人に返信が送信された時間が記録されます。

デフォルトでは、MTA はバックエンドストアシステムで不在メッセージを評価します。MTA リレーはバックエンドストアほど多くの処理を実行しないため、パフォーマンスを考慮して、バックエンドストアではなくメールリレーマシンで MTA が不在メッセージの評価するように設定することもできます。ただし、この設定を行うと、さまざまなリレーがさまざまなメッセージを処理するため、不在メッセージが意図したよりも頻繁に送信される可能性があります。意図したよりも頻繁に不在メッセージが送信されることを防ぐには、リレー間でファイルの記録を共有します。この方法も容認できない場合は、常にバックエンドストアシステムで不在メッセージを評価してください。

自動返信を設定する

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 `option.dat` ファイルで定義されます。`option.dat` ファイルにある `DELIVERY_OPTIONS` のデフォルトの自動返信ルールは、次のとおりです。

```
*^!autoreply=$M+$D@bitbucket
```

MTA は、自動返信 `DELIVERY_OPTION` MTA オプションにある「^」を認識します。これによって、MTA は不在の日付をチェックします。現在の日付が不在期間内である場合、処理は続行され、MTA は自動返信 `DELIVERY_OPTION` にある「!」を認識します。次に MTA は、ユーザーエントリのさまざまな自動返信 LDAP 属性に基づいて自動返信 Sieve スクリプトを作成します。自動返信ルールには、プレフィックス文字「!」、「#」、「^」、および「*」を付けることができます。

たとえば、メールボックス配信オプションに「!」フラグを付けることができます。このフラグによって、自動返信スクリプトの生成が無条件で有効になります。ただし、自動返信機能を別の配信オプションで有効化し、自動返信機能がさらに「^」フラグによって制御されるようにするのは理にかなっています。この段階で日付をチェックしたほうが Sieve のロジックを使用するよりも効率的です。

表 16-1 に、自動返信ルールで使用されるプレフィックス文字 (1 列目) とその定義 (2 列目) を示します。

表 16-1　DELIVERY_OPTIONS の自動返信ルールで使用されるプレフィックス文字

プレフィックス文字	定義
!	自動返信 Sieve スクリプトの生成を有効にします。
#	処理がリレーで実行されることを許可します。

表 16-1　DELIVERY_OPTIONS の自動返信ルールで使用されるプレフィックス文字 (続き)

プレフィックス文字	定義
^	評価する必要があると不在の日付から判明した場合にのみ、オプションを評価します。
*	ルールは、ユーザーにのみ適用可能です。

自動返信ルール自体は、bitbucket チャネル宛のアドレスを指定します。自動返信が生成されると、メールはこのメソッドによって配信されると見なされますが、MTA 機能には配信アドレスが必要です。bitbucket チャネルに配信された内容はすべて破棄されます。

バックエンドストアシステムで自動返信を設定する

DELIVERY_OPTIONS のデフォルトの自動返信ルールにより、自動返信はユーザーが使用するメールサーバー上で実行されます。バックエンドストアシステムで不在メッセージの評価を実行する場合は、設定を変更する必要はありません。これがデフォルトの動作です。

リレーでの自動返信を設定する

パフォーマンスを向上するために、バックエンドストアシステムではなくリレーで不在メッセージの評価を実行する場合は、option.dat ファイルを編集し、文字 # を DELIVERY_OPTIONS の自動返信ルールの先頭に追加します。次に例を示します。

1. エディタを使用して option.dat ファイルを開きます。
2. 自動返信ルールが次に示すようになるように、DELIVERY_OPTIONS オプションに追加または変更を行います。

   ```
   #*^!autoreply=$M+$D@bitbucket
   ```

   デフォルトの DELIVERY_OPTIONS オプションは次のようになっています。

   ```
   DELIVERY_OPTIONS=*mailbox=$M%$¥$2I$_+$2S@ims-ms-daemon, ¥
    &members=*, ¥
    *native=$M@native-daemon, ¥
    /hold=@hold-daemon:$A, ¥
    *unix=$M@native-daemon, ¥
    &file=+$F@native-daemon, ¥
   ```

```
&@members_offline=* ¥
,program=$M%$P@pipe-daemon, ¥
#forward=**, ¥
*^!autoreply=$M+$D@bitbucket
```

これによって、処理がリレーで実行されるようになります。リレーで MTA による自動返信を実行する場合、特定の人物が不在メッセージを最近送信しているかどうかを各リレーで個々に記録するか、その情報をリレー間で共有するかのいずれかとなります。送信された不在メッセージの数が非常に多くても問題ではない場合は特に、前者のほうが簡単です。不在メッセージの頻度ルールを厳しく適用する場合は、情報をリレー間で共有する必要があります。リレー間で情報を共有するには、ファイルは NFS 上にある必要があります。

これらのファイルの場所はオプション VACATION_TEMPLATE で制御されます。このオプション (option.dat 内) は、/<path>/%A に設定する必要があります。ここで、<path> は、各リレーマシン間で共有されるディレクトリへのパスです。テンプレートは file:URL である必要があります。ユーザーの名前を置換するには、$U を使用します。デフォルト設定は次のとおりです。

VACATION_TEMPLATE=file:///opt/SUNWmsgsr/data/vacation/$3I/$1U/$2U/$U.vac

メタキャラクタについては、219 ページの表 9-6 を参照してください。

注 不在ファイルのテンプレートによる UID へのアクセスが可能になったため、ユーザーの UID に基づいて不在ファイルのパスを構築できるようになりました。また、不在ファイルのパスを決定する際のアドレスとして、以前は現在の受取人アドレスが使用されていましたが、今はユーザーのメール属性に格納されているアドレスが使用されます。

不在返信メッセージの自動返信の動作方式

不在処理は、起動されると次のように機能します。

1. Sun Java System Messaging Server は、不在処理がシステムレベルではなくユーザーレベルの Sieve スクリプトで実行されたことを確認します。不在処理にシステムレベルのスクリプトが使用されている場合は、エラーが発生します。
2. 内部 MTA フラグの「no vacation notice」がチェックされます。このフラグが設定されている場合、処理は終了し、不在通知は送信されません。
3. メッセージの返信用アドレスがチェックされます。返信用アドレスが空白の場合、処理は終了し、不在通知は送信されません。

4. MTA は、現在のメッセージの To:、Cc:、Resent-to:、または Resent-cc: の各ヘッダーフィールドにある :addresses タグ付き引数にユーザーのアドレスまたはその他のアドレスが指定されているかどうかをチェックします。いずれのヘッダーフィールドでもアドレスが見つからない場合、処理は終了し、不在通知は送信されません。
5. Messaging Server は、:subject 引数と理由文字列のハッシュを作成します。この文字列は現在のメッセージの返信用アドレスとともに、ユーザーごとの不在応答の履歴に照らしてチェックされます。応答が :days 引数で許可されている時間内にすでに送信されている場合、処理は終了し、応答は送信されません。
6. Messaging Server は、:subject 引数、理由文字列、および :mime 引数から不在通知を作成します。この応答メッセージには、次の 2 つの基本的な形式があります。
   - 最初の部分に理由テキストがある、RFC 2298 で指定されている形式の MDN。
   - 単一パートのテキスト返信 (この形式は、「reply」自動返信モードの属性の設定をサポートするためにのみ使用される)。

不在メッセージが Messenger Express を使用して設定された場合、mailautoreplymode は自動的に reply に設定されます。

MTA フラグの「no vacation notice」は、デフォルトでは設定解除されています。このフラグは、システムレベルの Sieve スクリプトで標準外の novacation アクションを使用して設定できます。novacation Sieve アクションは、システムレベルの Sieve スクリプトでのみ許可されます。ユーザーレベルのスクリプトで使用された場合は、エラーが発生します。このアクションを使用して、不在返信に関してサイト全体に適用する制約を実装できます。たとえば、サブ文字列「MAILER-DAEMON」を含んでいるアドレスへの返信をブロックするなどです。

ユーザーごと、応答ごとの情報は、一連のフラットテキストファイルに保存されます。ファイルは、ローカルユーザーごとに 1 つあります。これらのファイルの場所およびネーミング方式は、VACATION_TEMPLATE MTA オプションで指定します。このオプションは file:URL に設定する必要があります。

これらのファイルの保守は自動的に行われ、VACATION_CLEANUP MTA オプションの設定 (整数) によって制御されます。これらのファイルのいずれかが開かれるたびに、この値を使用して現在時刻の値 (秒単位) が計算されます。結果がゼロである場合、ファイルがスキャンされ、有効期限切れのエントリはすべて削除されます。このオプションのデフォルト値は 200 です。これは、200 分の 1 の確率でクリーンアップパスが実行されることを意味します。

これらのフラットテキストファイルの読み取りと書き込みに使用される機能は、NFS 上で正しく動作するように設計されています。これによって、複数の MTA が単一のファイルセットを共通のファイルシステム上で共有することが可能になっています。

不在返信メッセージの自動返信の属性

不在処理で使用される LDAP ユーザーディレクトリ属性は、次のとおりです。

- LDAP_PERSONAL_NAME で定義されている属性

  エイリアス処理は、この属性で指定された個人名情報を記録し、この情報を使用して、生成された MDN または不在返信の `From:` フィールドを作成します。個人情報を公開しないように注意して使用してください。

- `vacationStartDate`

  休暇開始日時。値の形式は、`YYYYMMDDHHMMSSZ` です。この値は GMT を標準にしています。自動返信は、現在時刻がこの属性で指定されている時刻よりもあとの場合にのみ生成される必要があります。この属性がない場合、開始日は適用されません。`LDAP_START_DATE` MTA オプションを別の属性名に設定すると、この情報を別の属性で参照するように MTA に指示することができます。

  この属性は、Sieve スクリプトを生成したコードによって読み取られ、チェックされます。現在時刻が不在開始日より前である場合、不在処理は中止されます。現時点では、Sieve には日付 / 時刻テスト機能および比較機能がないため、スクリプト自体ではこの属性を操作できません。

- `vacationEndDate`

  休暇終了日時。値の形式は、`YYYYMMDDHHMMSSZ` です。この値は GMT を標準にしています。自動返信は、現在時刻がこの属性で指定されている時刻よりも前の場合にのみ生成される必要があります。この属性がない場合、終了日は適用されません。`LDAP_END_DATE` MTA オプションを別の属性名に設定すると、この情報を別の属性で参照するように MTA に指示することができます。

  この属性は、Sieve スクリプトを生成したコードによって読み取られ、チェックされます。現在時刻が不在終了日よりあとである場合、不在処理は中止されます。現時点では、Sieve には日付 / 時刻テスト機能および比較機能がないため、スクリプト自体ではこの属性を操作できません。

- `mailAutoReplyMode`

  ユーザーのメールアカウントに自動返信モードを指定します。この属性の有効な値は、次のとおりです。

  - `echo` - 追加された `mailAutoReplyText` テキストまたは `mailAutoReplyTextInternal` テキストに加えて、元のメッセージテキストをエコー出力するマルチパートを作成します。
  - `reply` - `mailAutoReplyText` または `mailAutoReplyTextInternal` のいずれかで指定されているシングルパートの返信を元の差出人に送信します。

これらのモードは、不在処理に渡される標準外の :echo 引数および :reply 引数として Sieve スクリプト内にあります。echo では、返信内容として元のメッセージが含まれた「処理済」の MDN が生成されます。reply では、返信テキストのみの返信が生成されます。不正な値は不在処理に渡される引数として示されません。したがって、元のメッセージのヘッダーのみの MDN が生成されます。自動返信モードとして echo を選択すると、前回の返信が送信された時期にかかわらず、すべてのメッセージに対して自動返信が送信されることにも注意してください。

LDAP_AUTOREPLY_MODE MTA オプションを別の属性名に設定すると、この情報を別の属性で参照するように MTA に指示することができます。

- mailAutoReplySubject

  自動返信応答で使用する Subject フィールドの内容を指定します。これは UTF-8 文字列である必要があります。この値は、:subject 引数として不在処理に渡されます。LDAP_AUTOREPLY_SUBJECT MTA オプションを別の属性名に設定すると、この情報を別の属性で参照するように MTA に指示することができます。

  現時点では、Sieve には特定の置換を実行する機能がないため、$SUBJECT を使用して元のメッセージをヘッダーに挿入することはできません。

- mailAutoReplyText

  受取人のドメイン内のユーザーを除くすべての差出人に送信する自動返信のテキスト。これが指定されていない場合、外部ユーザーは不在メッセージを受信しません。LDAP_AUTOREPLY_TEXT MTA オプションを別の属性名に設定すると、この情報を別の属性で参照するように MTA に指示することができます。

- mailAutoReplyTextInternal

  受取人のドメインから送信者に送られる自動返信のテキスト。これが指定されていない場合、内部ユーザーがメールの自動返信テキストのメッセージを受け取ります。LDAP_AUTOREPLY_TEXT_INT MTA オプションを別の属性名に設定すると、この情報を別の属性で参照するように MTA に指示することができます。

  MTA は、mailAutoReplyText または mailAutoReplyTextInternal のいずれかの属性値を理由文字列として不在処理に渡します。

- mailAutoReplyTimeOut

  任意の差出人への自動返信の応答に成功するまでの時間間隔 (時間単位)。mailAutoReplyMode=reply の場合にのみ使用されます。値が 0 の場合は、メッセージ受信のたびに応答が送り返されます。この値は、不在処理に渡される標準外の :hours 引数に変換されます (通常、Sieve 不在処理では、この目的のために :days 引数のみがサポートされている。また、0 の値は許可されていない)。

  この属性がユーザーエントリにない場合、AUTOREPLY_TIMEOUT_DEFAULT MTA オプションからデフォルトのタイムアウトが取得されます。LDAP_AUTOREPLY_TIMEOUT MTA オプションを設定すると、この情報を別の属性で参照するように MTA に指示することができます。

第 17 章

メールのフィルタリングとアクセス制御

この章では、メールをソース (差出人、IP アドレスなど) やヘッダー文字列に基づいてフィルタリングする方法について説明します。メールフィルタリングには、MTA へのアクセスを制御するため、マッピングテーブルを使う方法と、Sieve サーバー側ルール (SSR) を使う方法の 2 つがあります。

マッピングテーブルを使って MTA へのアクセスを制限すると、From: アドレスと To: アドレス、IP アドレス、ポート番号、およびソースまたは宛先チャネルに基づいてメッセージをフィルタリングできます。マッピングテーブルを使うと、SMTP リレーの有効または無効を切り替えることができます。Sieve はメールフィルタリングスクリプトであり、これを使うと、ヘッダーで見つかった文字列に基づいてメッセージをフィルタリングできます。これは、メッセージ本文に対しては機能しません。

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します。ヘッダーベースの制御が望ましい場合には、Sieve サーバー側ルールを使います。

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 : 管理者は、特定のマッピングテーブルを設定することによって MTA サービスへのアクセスを制御できます。管理者は、Messaging Server によるメールの送信または受信をどのユーザーに許可するか、あるいは許可しないかを制御できます。

第 2 部 メールボックスフィルタ : ユーザーと管理者は、メッセージをフィルタリングし、メッセージヘッダーで見つかった文字列に基づいて、フィルタ済みのメッセージに対するアクションを指定できます。Sieve フィルタ言語を使用します。フィルタリングは、MTA レベルまたはユーザーレベルのチャネルで実行できます。

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

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。これらのマッピングテーブルを使用すると、メールの送信、受信、またはその両方をどのユーザーに許可するか、あるいは許可しないかを制御できます。表 17-1 に、この節で説明するマッピングテーブルのリストを示します。FROM_ACCESS、MAIL_ACCESS、ORIG_MAIL_ACCESS の各マッピングテーブルに与えられているアプリケーション情報の文字列には、HELO/EHLO SMTP コマンドで要求されるシステム名が含まれます。この名前は文字列の最後に表示されて、文字列 (通常は「SMTP」) のほかの部分とはスラッシュで区切られています。要求されるシステム名は、ある種のワームやウィルスのブロックに役立つ場合があります。

アクセス制御マッピングテーブル - 操作

アクセス制御マッピングテーブルの形式は、ほかのマッピングテーブルと同じ一般的なものです (237 ページの「マッピングファイル」を参照)。マッピングテーブル名の後ろに改行が入り、そのあとに 1 つまたは複数のマッピングエントリが続きます。マッピングエントリは、左側の「検索パターン」と右側の「テンプレート」から構成されています。検索パターンは特定のメッセージをフィルタリングし、テンプレートはメッセージに対して実行するアクションを指定します。次に例を示します。

```
SEND_ACCESS

 *|Elvis1@sesta.com|*|*       $Y
 *|Nelson7@sesta.com|*|*     $Y
 *|AkiraK@sesta.com|*|*       $Y
 *|*@sesta.com|*|*              $NMail$ Blocked
```

この例では、ドメイン Elvis1、Nelson、および AkiraK を除き、ドメイン sesta.com からのすべての電子メールをブロックしています。

アクセス制御マッピングエントリの検索パターンは多数の検索条件から構成され、個々の検索条件は縦棒 (|) で区切られています。検索条件の順序はアクセスマッピングテーブルによって決定されます。これについてはあとの節で説明します。例として、SEND_ACCESS マッピングテーブルの検索形式を次に示します。

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。

注 mappings ファイルを変更した場合は、必ず設定をコンパイルしなおしてください (『Sun Java System Messaging Server Administration Reference』の imsimta refresh コマンドを参照)。

表 17-1 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 (537 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、着信接続をブロックする場合に使用します。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 アドレスが調べられます。
ORIG_SEND_ACCESS (537 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、着信接続をブロックする場合に使用します。書き換え後、エイリアス展開の前に To アドレスが調べられます。
MAIL_ACCESS (539 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着信接続をブロックする場合に使用します。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ります。
ORIG_MAIL_ACCESS (539 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着信接続をブロックする場合に使用します。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ります。

表 17-1　アクセス制御マッピングテーブル (続き)

マッピングテーブル	説明
FROM_ACCESS (542 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場合に使用します。このテーブルは、To アドレスが不適切な場合に使用します。
PORT_ACCESS (544 ページを参照)。	IP 番号に基づいて着信接続をブロックする場合に使用します。

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。

アクセス制御マッピングテーブルのフラグ

表 17-2 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 17-3 を参照)。

引数を伴うフラグの場合、引数がテーブルに表示される読み取り順に並んでいる必要があります。次に例を示します。

ORIG_SEND_ACCESS

```
  tcp_local|*|tcp_local|*     $N$D30|Relaying$ not$ allowed
```

この場合、遅延期間の後ろに拒否文字列が来るのが正しい順序です。フラグ自体は、どのような順序で指定してもかまいません。したがって、次のエントリはいずれも同じ結果になります。

```
30|Relaying$ not$ allowed$D$N
$N30|Relaying$ not$ allowed$D
30|$N$DRelaying$ not$ allowed
```

表 17-2　アクセスマッピングフラグ

フラグ	説明
$A	SASL が使用されている場合に設定されます。248 ページの「特殊なフラグの確認」を参照してください。
$B	ビットバケットにメッセージをリダイレクトします。

表 17-2　アクセスマッピングフラグ (続き)

フラグ	説明
$D	配信遅延の確認が要求された場合に設定されます (FROM_ACCESS では指定できない)。248 ページの「特殊なフラグの確認」を参照してください。
$F	配信失敗の確認が要求された場合に設定されます (FROM_ACCESS では指定できない)。248 ページの「特殊なフラグの確認」を参照してください。
$H	.HELD ファイルとしてメッセージを保留します。
$S	配信成功の確認が要求された場合に設定されます (FROM_ACCESS では指定できない)。248 ページの「特殊なフラグの確認」を参照してください。
$T	TLS が使用されている場合に設定されます。248 ページの「特殊なフラグの確認」を参照してください。
$U	ORIG_SEND_ACCESS、SEND_ACCESS、ORIG_MAIL_ACCESS、および MAIL_ACCESS で使用された場合、マッピング開始時から 1 つの整数引数をとり、MM_DEBUG の値をそれに応じて設定します。また、このフラグが設定されていると、チャネルレベルのデバッグも有効になります。その結果、ソース IP アドレス、元のアドレス、受取人アドレスなどに基づいてデバッグを有効化できるようになります。
$Y	アクセスを許可します。
$V	すべての受取人について強制破棄が実行されるようにします。
$Z	すべての受取人について強制破棄が実行されるようにします。
フラグと引数、引数の読み取り順序 + (このリストはアルファベット順にしないこと)	
$U*integer*	マッピング開始時から 1 つの整数引数をとり、MM_DEBUG をそれに応じて設定します。また、このフラグが設定されていると、チャネルレベルのデバッグも有効になります。その結果、ソース IP アドレス、元のアドレス、受取人アドレスなどに基づいてデバッグを有効化できるようになります。
$J*address*	* 元のエンベロープの From: アドレスを指定の *address* に置換します。
$K*address*	* ++ 元の Sender: アドレスを指定の *address* に置換します。
$I*user*|*identifier*	特定のユーザーのグループ ID を調べます。
$<*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) に送ります。

表 17-2　アクセスマッピングフラグ (続き)

フラグ	説明
$>*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) に送ります。
$D*delay*	応答を *delay* (100 分の 1 秒単位) だけ遅らせます。正の値の場合、トランザクションでの各コマンド時にこの遅延が適用され、負の値の場合、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) にのみこの遅延が適用されます。
$T*tag*	*tag* を前に付けます。
$A*header*	メッセージにヘッダー行 *header* を追加します。
$G*conversion_tag*	ORIG_SEND_ACCESS、SEND_ACCESS、ORIG_MAIL_ACCESS、および MAIL_ACCESS で使用された場合、マッピングの結果の値を読み込んで、現在の受取人に適用される変換タグの集合として処理します。FROM_ACCESS とともに使用された場合、変換タグはすべての受取人に適用されます。マッピングから読み取られる一連の引数の中で、$G は $A (ヘッダーアドレス) のあとに配置されます。436 ページの「メール変換タグ」を参照してください。
$S*x,y,z*	* マッピングの結果から、追加および別々の引数がマッピング結果から読み取られるようにします。この引数は、カンマで区切られた 1 個 ～ 3 個の整数値から構成されます。最初の値は、トランザクション用の新しい最小の blocklimit を設定します。2 番目の値は新しい最小の recipientlimit を設定し、3 番目の値は新しい最小の recipientcutoff を設定します。いずれかの取得引数が読み込まれると、マッピングの結果から引数が読み込まれます。詳細は、404 ページの「絶対的なメッセージサイズ制限を指定する」を参照してください。
$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します。
$,*spamadjust_arg*	アクセスマッピングテーブルから Sieve spamadjust 処理を実行可能にします。引数は、spamadjust 引数と同じ形式をとります。また、上のマッピングの一部は受取人単位で適用されます。実行されるすべての spamadjust 処理はすべての受取人に適用されます。
$N*string*	アクセスを拒否し、オプションのエラーテキスト *string* を送ります。

表 17-2　アクセスマッピングフラグ (続き)

フラグ	説明
$F*string*	*$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* を送ります。

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。

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、メッセージのエンベロープ From: アドレスおよびエンベロープ To: アドレス、メッセージがどのチャネルから入ってきたか、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 From: アドレスとエンベロープ 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 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キューに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」、「$F」、または「$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、534 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。

MTA オプション ACCESS_ORCPT を 1 に設定すると、SEND_ACCESS、ORIG_SEND_ACCESS、MAIL_ACCESS、および ORIG_MAIL_ACCESS マッピングテーブルに渡されるプローブ値に縦棒で区切られたフィールドが付加されます。このフィールドには、元の受取人 (ORCPT) アドレスが入ります。メッセージに ORCPT アドレスが含まれない場合は、変更されていない元の RCPT TO: アドレスが代わりに使用されます。デフォルトは 0 であり、プローブ値で終わります。

src-channel|*from-address*|*dst-channel*|*to-address*|*ORCPT_address*

次の例は、mail や Pine などの UNIX ユーザー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ーは、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、次の例に示す 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。

コード例 17-1　　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
```

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、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。

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ー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。着信 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*

ここで、*port-access-probe-info* は、SMTP 着信メッセージの場合は 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構成され、それ以外の場合は空白になります。*app-info* には、HELO/EHLO SMTP コマンドで要求されるシステム名が含まれます。この名前は文字列の最後に表示されて、文字列 (通常は「SMTP」) のほかの部分とはスラッシュで区切られています。要求されるシステム名は、ある種のワームやウィルスのブロックに役立つ場合があります。*submit-type* は、メッセージが Messaging Server にどのように送信されたかに応じて、MAIL、SEND、SAML、または SOML のいずれかになります。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 サーバーに送信され

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。

MTA オプション ACCESS_ORCPT を 1 に設定すると、SEND_ACCESS、ORIG_SEND_ACCESS、MAIL_ACCESS、および ORIG_MAIL_ACCESS マッピングテーブルに渡されるプローブ値に縦棒で区切られたフィールドが付加されます。このフィールドには、元の受取人 (ORCPT) アドレスが入ります。メッセージに ORCPT アドレスが含まれない場合は、変更されていない元の RCPT TO: アドレスが代わりに使用されます。デフォルトは 0 であり、プローブ値で終わります。次に例を示します。

port-access-probe-info|*app-info*|*submit-type*|*send_access-probe-info*|*ORCPT_address*

着信 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、ユーザーに対し POP および IMAP クライアントの From: アドレス設定を正しく行なったりするように奨励する効果もあります。たとえば、IP アドレス 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 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 From: アドレスに siroe.com を表示するようなサイトでは、次の例に示す MAIL_ACCESS マッピングテーブルを使用します。

コード例 17-2　　MAIL_ACCESS マッピングテーブル

```
MAIL_ACCESS

! vip の 2 つのシステムのエントリ
!
  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
!
! ほかのシステムのアドレスから vip の From: アドレスを使用することを
! 許可しない
!
  TCP|*|25|*|*|SMTP|MAIL|tcp_*|vip@siroe.com|*|*  ¥
      $N500$ Not$ authorized$ to$ use$ this$ From:$ address
!
! siroe.com の From: アドレスを持つサブネット内からの送信を
! ブロックする
!
  TCP|*|25|1.2.*.*|*|SMTP|MAIL|tcp_*|*@siroe.com|*|*  $Y
!
! 通知を許可する
!
  TCP|*|25|1.2.*.*|*|SMTP|MAIL|tcp_*||*|*  $Y
!
! non-siroe.com アドレスを持つサブネット内からの送信を
! ブロックする
!
  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、またはその両方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*

ここで、*port-access-probe-info* は、SMTP 着信メッセージの場合は 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構成され、それ以外の場合は空白になります。*app-info* には、HELO/EHLO SMTP コマンドで要求されるシステム名が含まれます。この名前は文字列の最後に表示されて、文字列 (通常は「SMTP」) のほかの部分とはスラッシュで区切られています。要求されるシステム名は、ある種のワームやウィルスのブロックに役立つ場合があります。*submit-type* は、メッセージが MTA にどのように送信されたかに応じて、MAIL、SEND、SAML、または SOML のいずれかになります。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 サーバー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は、その特定の To: アドレスに対しメッセージをキューに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」、「$F」、または「$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、534 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。

コード例 17-3 FROM_ACCESS マッピングテーブル

```
FROM_ACCESS

  *|SMTP|*|tcp_auth|*|        $Y
  *|SMTP|*|tcp_auth|*|*       $Y$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_auth|*|      $Y
  *|SMTP|*|tcp_auth|*|*     $Y$K$3
```

ただし、FROM_ACCESS の本来の目的は、次の例に示すように、より複雑で微妙な変更を行うことにあります。着信メッセージに 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、着信メッセージに 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ない)。さらに、エンベロープの From: にオプションのサブアドレス情報が含まれているというだけでは、SMTP AUTH およびエンベロープの From: アドレスが異なるとみなさないとします。

```
FROM_ACCESS

! 認証済みのアドレスが使用できない場合、何もしない
  *|SMTP|*|tcp_auth|*|                $Y
! 認証済みのアドレスがエンベロープの From: に一致する場合は、何もしない
  *|SMTP|*|tcp_auth|*|$2*             $Y
! 認証済みのアドレスがエンベロープの From:sans
! サブアドレスに一致する場合は、何もしない
  *|SMTP|*|tcp_auth|*+*@*|$2*@$4*     $Y
! ただし、認証済みアドレスが存在しているが
! 一致しない場合は、
! Sender: ヘッダー
  *|SMTP|*|tcp_auth|*|*               $Y$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、着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。

注 MMP は PORT_ACCESS マッピングテーブルを使用しません。MMP を使用している場合、特定の IP アドレスからの SMTP 接続を拒否するには、TCPAccess オプションを使用する必要があります。175 ページの「MMP を使ったメールアクセスを設定するには」を参照してください。マッピングテーブルを使って SMTP 接続を制御する場合は、INTERNAL_IP マッピングテーブルを使用します (552 ページの「外部サイトの SMTP リレーを許可する」を参照)。

$< フラグにオプションの文字列が続いており、マッピングプローブが一致しなかった場合は、Messaging Server が文字列を syslog (UNIX) またはイベントログ (NT) に送ります。$> 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 17-3 に使用可能なフラグを表示します。

表 17-3 PORT_ACCESS マッピングフラグ

フラグ	説明
$Y	アクセスを許可します。
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) に送ります。
$> 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) に送ります。
$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ります。
$F 文字列	「$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ります。
$T テキスト	LOG_CONNECTION MTA オプションのビット 1 が設定されており、かつ「$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ります。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。	

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コンパイルしなおしてください。

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 への指定 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。

$[*msg_svr_base*/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|*|*  ¥
$C$[msg_svr_base/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。

$[*msg_svr_base*/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 はサーバーの 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、-sender、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。詳細は、『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) を参照してください。次の例で、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。

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

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、サーバーがあるホスト以外のシステムです。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 Messaging Server システムの SMTP サーバー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 *msg_svr_baset*`/config/mappings` にあります。

たとえば、IP アドレスが 123.45.67.89 の 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
```

この例の最初のエントリでは、`$(IP-pattern/signicant-prefix-bits)` 構文を使用して、32 ビットの 123.45.67.89 すべてに一致する IP アドレスが内部として認識されるように指定しています。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ように指定しています。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 アドレスやサブネットを指定して、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、IP アドレスまたはサブネット (サブネットの指定には $(.../...) 構文を使用) を左側に、$Y を右側に指定する必要があります。また、既存の $(.../...) エントリを変更して、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 リレーサーバー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

! IP アドレスを 123.45.67.80 ～ 123.45.67.95 の範囲に一致させる
   $(123.45.67.80/28)   $Y
! IP アドレスを 123.45.67.96 ～ 123.45.67.99 の範囲に一致させる
   $(123.45.67.96/30)   $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、/imsimta test -match ユーティリティが便利です。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も便利です。

INTERNAL_IP マッピングテーブルを編集したら、必ず imsimta restart コマンド (コンパイルされた設定で実行していない場合) または imsimta refresh コマンド (コンパイルされた設定で実行している場合) 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 コマンド行ユーティリティについては、『Messaging Server Reference Manual』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。お使いのシステムまたはサイトで 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、現在の書き換えルールのすぐあとに新しい書き換えルールを追加します。現在の書き換えルールは次のようになっています。

```
! マッピング検索を内部 IP アドレスに対して実行する
[]     $E$R${INTERNAL_IP,$L}$U%[$L]@tcp_intranet-daemon
```

次の新しい書き換えルールを追加します。

```
! マッピング検索を外部 IP アドレスに対して実行する []
$E$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、前の項で説明した構文を使用します。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、ユーザー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 ユーザーおよび IMAP ユーザー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ーによるリレーを許可するには、2 つのクラスのユーザーを識別するように 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ーの場合、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
- 555 ページの「認証ユーザーのメールを識別する」
- 556 ページの「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 サーバーが着信 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ルール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、サーバー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。

```
! 最終的な書き換えルール
defaults noswitchchannel
! ローカルストア
ims-ms ...
```

- 着信 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
  ```

- 着信 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 サーバーにメッセージが入ってくると、サーバーが着信接続のソース IP アドレスを調べるようになります。サーバーは、着信接続のリテラル 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ルール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。

認証ユーザー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ーが存在することがあります。これらのユーザー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。ユーザーが 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、認証ユーザーによる送信を許可し、認証されていないユーザー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。

認証ユーザー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 が明示的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ーが送信した 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

この例では、承認されていないユーザーがシステムを介して SMTP メールのリレーを行えないようにします。まず、ローカルユーザーによる 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 ユーザーおよび IMAP ユーザーは、メールの送信に Messaging Server を使います。ローカルユーザーには、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ーのほか、ローカルユーザーとして認証され得るリモートユーザーも含まれます。

サーバー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、これらのユーザー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 サーバー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)。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ー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ーにメールを送信したり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

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 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 のルールを使った有効な 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|*|*@*|*|* ¥
$[msg_svr_base/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構文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、着信接続を 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
! . . 通常のオプションの残りの部分 . . .
DNS_VERIFY_DOMAIN=rbl.maps.vix.com
DNS_VERIFY_DOMAIN=dul.maps.vix.com
!. . . など . . .
```

この場合、メッセージは SMTP レベルで拒否されます。つまり、メッセージは SMTP ダイアログの間に拒否されることになり、MTA に送信されることはありません。この方法の短所は、内部ユーザーからのメッセージを含む、通常の SMTP 着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

! 内部接続を無条件で許可する
  *|*|*|*|* $C$|INTERNAL_IP;$3|$Y$E
! RBL リストに対するほかの接続をチェックする
  TCP|*|25|*|*  ¥
$C$[msg_svr_base/lib/dns_verify.so, ¥
dns_verify_domain_port,$1,rbl.maps.vix.com.]EXTERNAL$E


ORIG_MAIL_ACCESS

  TCP|*|25|*|*|SMTP|*|tcp_local|*@*|*|* ¥
$C$[msg_svr_base/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

dns_verify プログラムは DNS ベースのデータベースをサポートします。このデータベースは、不特定多数宛のメールを送る可能性のある着信 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、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|*|*  ¥
$C$[<msg_svr_base/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 内のクエリーで見つかっても、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ー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、次に示す ORIG_SEND_ACCESS マッピングテーブルの場合を考えてみます。

```
ORIG_SEND_ACCESS

! ユーザーはインターネットへの送信を許可されている
!
  *|adam@siroe.com|tcp_local|*    $Y
  *|betty@siroe.com|tcp_local|*   $Y
!... など ...
!
! ユーザーはインターネットへの送信を許可されていない
!
  *|norman@siroe.com|tcp_local|*  $NInternet$ access$ not$ permitted
  *|opal@siroe.com|tcp_local|*    $NInternet$ access$ not$ permitted
!... など ...
!
! ユーザーはインターネットからの受信を許可されている
!
  tcp_*|*|*|adam@siroe.com        $Y
  tcp_*|*|*|betty@siroe.com       $Y
!... など ...
!
! ユーザーはインターネットからの受信を許可されていない
!
  tcp_*|*|*|norman@siroe.com      $NInternet$ e-mail$ not$ accepted
  tcp_*|*|*|opal@siroe.com        $NInternet$ e-mail$ not$ accepted
!... など ...
```

このように、ユーザー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、何千件ものユーザーを効率的に処理できる設定) を次の例で示します。この例では、一般データベースのソーステキストファイルのサンプルおよび ORIG_SEND_ACCESS マッピングテーブルのサンプルを示します。このソースファイルをデータベースのフォーマットにコンパイルするには、imsimta crdb コマンドを実行します。

```
% imsimta crdb input-file-spec output-database-spec
```

imsimta crdb ユーティリティの詳細については、『Sun Java System Messaging Server Administration Reference』を参照してください。

データベースエントリ

```
SEND|adam@domain.com     $Y
SEND|betty@domain.com    $Y
!... など ...
SEND|norman@domain.com   $NInternet$ access$ not$ permitted
SEND|opal@domain.com     $NInternet$ access$ not$ permitted
!... など ...
RECV|adam@domain.com     $Y
RECV|betty@domain.com    $Y
!... など ...
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

! インターネットに送信する場合はチェックする
!
  *|*|*|tcp_local        $C${SEND|$1}$E
!
! インターネットから受信する場合はチェックする
!
  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「SEND|」および「RECV|」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。

第 2 部 メールボックスフィルタ

メールボックスフィルタは、Sieve フィルタとも呼ばれ、メッセージヘッダー内に指定の文字列を含んだメッセージをフィルタし、これらのメッセージに指定のアクションを適用します。管理者は、チャネルや MTA を介して、ユーザーに送信されるメールストリームをフィルタすることができます。Messaging Server のフィルタはサーバー上に保存されてサーバーによって評価されるため、サーバー側ルール (SSR) と呼ばれることがあります。

第 2 部には、以下の項目があります。

Sieve フィルタのサポート

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。Sieve の構文およびセマンティクスの詳細については、RFC3028 を参照してください。また、Messaging Server では次の Sieve 拡張機能もサポートしています。

- **`jettison`**: メッセージが速やかに破棄されるという点で `discard` と似ていますが、`discard` は黙示的な保存をキャンセルするだけなのに対し、`jettison` は強制的に破棄 (`discard`) を実行します。動作上の相違点は、複数の Sieve フィルタが必要な場合にだけ考慮します。たとえば、システムレベルの破棄は、明示的に `keep` を指定するユーザー Sieve フィルタで置き換え可能です。システムレベルの `jettison` は、ユーザー Sieve によって行われるすべての処理よりも優先されます。
- Head-of-household Sieve フィルタ：あるユーザーが別のユーザーのために Sieve フィルタを指定する手段を提供します。MTA オプションによって制御される 2 つの LDAP 属性をユーザーエントリで使用します。
  - `LDAP_PARENTAL_CONTROLS` - 「`Yes`」または「`No`」のどちらかの文字列値を含む属性を指定します。「`Yes`」は head of household Sieve がこのエントリに適用されることを意味し、「`No`」はそのような Sieve が適用されないことを意味します。デフォルトはありません。
  - `LDAP_FILTER_REFERENCE` - head of household Sieve が格納されているディレクトリエントリを指す DN が含まれている属性を指定します。デフォルトはありません。

head of household Sieve を含むエントリには、次の MTP オプションによって指定される 2 つの属性が含まれている必要があります。

- LDAP_HOH_FILTER - head of household Sieve を含む属性を指定します。このオプションの値は、デフォルトで mailSieveRuleSource になります。
- LDAP_HOH_OWNER - head of household Sieve の所有者の電子メールアドレスを含む属性を指定します。このオプションの値は、デフォルトで mail になります。

head of household Sieve が動作するためには、どちらの属性も必要です。

Sieve フィルタリングの概要

Sieve フィルタは、メールメッセージに適用される 1 つまたは複数の (メッセージヘッダーにある文字列によって異なる) 条件付きアクションで構成されています。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。ユーザーは Messenger Express を使用して、自分のメールボックスにユーザー単位のフィルタを作成できます。この具体的な手順については、Messenger Express のオンラインヘルプを参照してください。

サーバーは、次の優先順位に従ってフィルタを適用します。

1. ユーザーレベルのフィルタ

   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、またはユーザー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。ユーザー単位のフィルタが設定されます。

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、それぞれのユーザーはメールボックスフィルタを所有していません。ユーザーが Messenger Express 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。

ユーザーレベルのフィルタを作成するには

ユーザー単位のフィルタは、特定ユーザー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。ユーザー単位のメールフィルタは、Messenger Express のみで作成できます。

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。

   `../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. 設定をコンパイルしなおし、ディスパッチャを再起動します。

   注 : フィルタファイルへの変更を有効にするのに、コンパイルしなおし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構文は以下のとおりです。

`destinationfilter` *URL-pattern*

`sourcefilter` チャネルキーワードの構文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構文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。565 ページの表 17-4 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 17-4 filter チャネルキーワードの *URL-pattern* の置換タグ (大文字と小文字の区別なし)

タグ	意味
*	グループの拡張を実行します。
**	mailForwardingAddress 属性を拡張します。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ます。
$$	$ 文字に置き換えます
$¥	後続のテキストを小文字にします
$^	後続のテキストを大文字にします
$_	後続のテキストで大文字と小文字を変換しません
$~	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ます
$1S	$S と同じですが、サブアドレスがない場合は何も行いません
$2S	$S と同じですが、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します
$3S	$S と同じですが、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します
$A	アドレス (ローカル部分 @ ホスト . ドメイン) に置き換えます

表 17-4　filter チャネルキーワードの *URL-pattern* の置換タグ (大文字と小文字の区別なし) (続き)

タグ	意味
$D	ホスト . ドメインに置き換えます
$E	第 2 スペア属性の値 LDAP_SPARE_1 を挿入します
$F	配信ファイル名 (mailDeliveryFileURL 属性) を挿入します
$G	第 2 スペア属性の値 LDAP_SPARE_2 を挿入します
$H	ホストに置き換えます
$I	ホストしているドメインを挿入します (domainUidSeparator で指定した区切り文字の右側に UID の一部を挿入)。ホストしているドメインがないと失敗します
$1I	$I と同じですが、ホストしているドメインがない場合は何も挿入しません
$2I	$I と同じですが、ホストしているドメインがない場合は何も挿入せず前の文字を削除します
$3I	$I と同じですが、ホストしているドメインがない場合は何も挿入せず後続の文字を無視します
$L	ローカル部分に置き換えます
$M	ホストしているドメイン部分を除いた UID を挿入します
$P	メソッド名を挿入します (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します。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 +) に続くユーザー部分の該当する箇所です。ただし、MTA オプションの SUBADDRESS_CHAR で指定することもできます。サブアドレスを指定しないと失敗します
$U	現在のアドレスのメールボックス部分を挿入します。@ 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ます。

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

   ```
   ../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. 設定をコンパイルしなおし、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、ユーザー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、ユーザー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。

Messaging Server 6 2004Q2 より前の Messaging Server では、jettison Sieve アクションによる filter_discard チャネルの使用は、MTA オプション FILTER_DISCARD によって制御されていました。これは現在では、FILTER_JETTISON オプションによって制御されるようになり、このオプションは FILTER_DISCARD の設定からデフォルト値を取得します。また、FILTER_DISCARD のデフォルト値は 1 です (破棄されたメッセージは bitbucket チャネルに送られる)。

ユーザーレベルのフィルタをデバッグするには

ユーザーから Sieve フィルタが期待通りに動作しないという苦情が寄せられた場合、フィルタをデバッグするために実行できるいくつかの手順があります。これらの手順を以下に示します。

1. fileinto フィルタを動作させるには、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。

   ```
   fileinto $u+$s@$d
   ```

2. ユーザーの LDAP エントリからユーザーレベルのフィルタを取得します。

   ユーザーレベルのフィルタは、MailSieveRuleSource 属性の下のそれぞれの LDAP エントリに保存されています。ldapsearch コマンドを使用してこのフィルタを検索する際には、これらが base64 でエンコードされているため -Bo スイッチで出力をデコードする必要があることに注意してください。

   ```
   ./ldapsearch -D "cn=directory manager" -w password -b
   "o=alcatraz.sesta.com,o=isp" -Bo uid=test
   ```

   以下で説明している imsimta test -rewrite コマンドも、自動的にフィルタをデコードします。

3. ユーザーのフィルタが MTA から見えることを確認します。

   次のコマンドを発行します。

   ```
   # imsimta test -rewrite -filter -debug user@sesta.com
   ```

   これにより、前の手順で取得したユーザーの Sieve フィルタが出力されます。フィルタが見つからない場合は、LDAP エントリがそれらを返さない理由を調べる必要があります。imsimta test -rewrite の出力にフィルタが表示されていたら、ユーザーのフィルタが MTA から見えているということです。次に必要な処理は、imsimta test -expression コマンドを使用してフィルタの解釈をテストすることです。

4. imsimta test -exp を使用して、ユーザーのフィルタをデバッグします。以下の情報が必要です。

   a. ユーザーの mailSieveRuleSource 属性からの Sieve 言語ステートメント。前述の手順を参照してください。

b. フィルタをトリガするはずだった rfc2822 メッセージ。

c. フィルタがメッセージに対してどのような処理を行うべきだったかという説明。

5. ユーザーの mailSieveRuleSource の値に基づいて、Sieve 言語ステートメントを含むテキストファイル (temp.filter など) を作成します。次に例を示します。

```
require "fileinto";
if anyof(header :contains
["To","Cc","Bcc","Resent-to","Resent-cc",
   "Resent-bcc"] "commsqa"){
   fileinto "QMSG";
}
```

期待される結果 : commsqa がこのメッセージの受取人である場合は、メッセージを QMSG というフォルダにファイリングします。

6. ユーザーによって指定される rfc2822 メッセージファイルの内容を含む、test.msg という名前のテキストファイルを作成します。

ユーザーのメッセージストア領域から .msg ファイルを使用するか、あるいはユーザーによって指定される rfc2822 メッセージファイルの内容を含む test_rfc2822.msg というテキストファイルを作成することができます。

7. imsimta test -exp コマンドを使用します。

```
# imsimta test -exp -mm -block -input=temp.filter
-message=test_rfc2822.msg
```

8. 出力を確認します。

imsimta test -exp コマンドの最後の行は、Sieve 解釈の結果を示します。結果は次のようになります。

```
Sieve Result: []
```
または
```
Sieve Result:[action]
```

action は、Sieve フィルタを適用した結果としてこのメッセージに実行されたアクションです。

フィルタの条件に一致した場合、結果としていくつかのアクションが表示されます。一致するものがない場合、Sieve 結果は空白となり、Sieve フィルタに論理上のエラーがあるか、または .msg ファイルに一致する情報が含まれていないかのどちらかです。ほかのエラーが発生している場合は、Sieve スクリプトファイルに構文エラーがあるので、デバッグする必要があります。

出力の詳細については、570 ページの「imsimta test -exp の出力」を参照してください。

9. フィルタが構文的に有効で結果が正しい場合、次に必要な処理は tcp_local_slave.log デバッグログファイルを調べることです。

テストしたメッセージファイルと送信されているメッセージファイルが同一でない可能性があります。受信しているメッセージを確認する唯一の方法は、tcp_local_slave.log ファイルを調べることです。このログには、MTA に送信されている正確なメッセージと、メッセージに対してフィルタがどのように適用されているかが表示されます。

tcp_local_slave.log デバッグファイルの入手の詳細については、411 ページの「デバッグのキーワード」の slave_debug キーワードを参照してください。

imsimta test -exp の出力

imsimta test -exp の完全なコマンドは、次のとおりです。

```
# imsimta test -exp -mm -block -input=temp.filter
-message=rfc2822.msg
```

出力の例を以下に示します。

コード例 17-4 imsimta test -exp の出力

```
# imsimta test -exp -mm -block -input tmp.filter -message=rfc2822.msg
Expression:if header :contains ["to"] ["pamw"]          (1)
Expression: {
Expression:redirect "usr3@sesta.com";
Expression:keep;
Expression: }
Expression:
Expression:Dump:header:2000114;0  3  1  :contains  1  "to"  1
"pamw"  if  8  ;
Dump:redirect:2000121;0  1  1  "usr3@sesta.com"  ;  keep:2000117;0  (2)
Dump: 0
Result: 0
Filter result:[ redirect "usr3@sesta.com" keep ]   (3)
```

1) Expression: 出力行は、tmp.filter テキストファイルから読み取られ、解析されるフィルタを示します。これらは、スクリプトのデバッグにはあまり関係がありません。

2) Dump: 出力行は、Sieve ステートメントを解釈しているコンピュータの結果を示します。ここにエラーが表示されないようにしてください。出力は入力と一致しているように見えなければなりません。たとえば、ダンプには、フィルタファイル内の行 redirect "usr3@sesta.com"; とよく似た redirect,usr3@sesta.com という語が表示されます。

このように一致するテキストが表示されない場合は調べてみる必要があります。表示された場合、これらもスクリプトのデバッグとはあまり関係がありません。

3) 出力の一番下の行に、`Filter result:` ステートメントが表示されます。前述したように、次の 2 とおりの結果が考えられます。

`Sieve Result:[]` または `Sieve Result:[`*action*`]`

action は、Sieve スクリプトによって実行されるアクションです。結果が空白になる場合もあるので注意してください。たとえば、破棄フィルタの場合、そのフィルタが常に、テストの対象となるすべての `.msg` ファイルを破棄しているのではないことをテストする必要があります。角括弧の間に何らかのアクションが含まれる場合、次のようになります。

```
Filter result:[ fileinto "QMSG" keep]
```

これは、`rfc2822.msg` ファイルのテキストがフィルタ条件に一致していることを示しています。この例では、フィルタはメールを `QMSG` フォルダにファイリングして、コピーを受信箱に保存します。この場合、結果として実行されるアクションは、`fileinto` および `keep` です。

フィルタをテストする際には、両方の結果について、各種の `.msg` ファイルをテストする必要があります。常に、使用するフィルタに一致するメッセージがフィルタ処理されていること、また、一致させたくないメッセージがフィルタ処理されていないことをテストする必要があります。

ワイルドカードとの照合の場合は、`:contains` テストではなく `:matches` テストを行う必要があることを理解しておいてください。たとえば、`from=*@sesta.com` と一致させたい場合は `:matches` を使用してください。そうしないと、テストの条件が満たされないためテストは失敗します。

imsimta test -exp の構文

`imsimta test -exp` は、指定した RFC2822 メッセージに対して Sieve 言語ステートメントをテストし、フィルタの結果を標準出力に送ります。

構文は次のとおりです。

```
imsimta test -exp -mm -block -input=Sieve_language_scriptfile
-message=rfc2822_message_file
```

ここで、

`-block` は、単一の Sieve スクリプトとして完全な入力を示します。デフォルトでは、各行は別々のスクリプトとして処理され、別々に評価されます。Sieve は、ファイルの終わりに到達したときだけ評価されます。

`-input=`*Sieve_file* は、Sieve スクリプトを含むファイルです。デフォルトでは、stdin からテストスクリプト行またはスクリプトブロックが読み込まれます。

-message=*message_file* は、Sieve スクリプトのテストを実行したい RFC 2822 メッセージを含むテキストファイルです。ここには、RFC 2822 メッセージのみが存在する必要があります。キューファイル (zz*.00 ファイル) ではありえません。

このコマンドを有効にすると、スクリプト情報を読み取り、それをテストメッセージと関連させて評価し、結果として出力します。結果には、実行すべきアクションと、スクリプトの最終ステートメントの評価結果が示されます。

このほかに、次のような修飾子が有効です。

-from=*address* は、エンベロープのテストに使用されるエンベロープ from: アドレスを指定します。デフォルトでは、MTA オプション RETURN_ADDRESS によって指定された値が使用されます。

-output=*file* は、結果をファイルに書き込みます。デフォルトでは、スクリプトの評価結果が stdout に書き込まれます。

第 18 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そ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ー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。

システムにユーザー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。サーバーがサポートするユーザー数によって、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、別のメッセージストアパーティションを追加することです (619 ページの「メッセージストアのパーティションを構成する」を参照)。

また、複数のホストしているドメインをサポートしている場合は、1 つのサーバー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。

Messaging Server では、メッセージストアの管理のために、Sun Java System Console インタフェースに加えてコマンド行ユーティリティのセットを提供しています。表 18-1 では、このコマンド行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、623 ページの「メッセージストアの保守手順を実行する」および『Messaging Server Reference Manual』を参照してください。

表 18-1 メッセージストアのコマンド行ユーティリティ

ユーティリティ	説明
`configutil`	ストアの設定パラメータを設定および変更します。
`deliver`	メールを、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直接配信します。
`hashdir`	特定のユーザー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。
imsconnutil	メッセージストアユーザーへのユーザーアクセスを監視します。
`imexpire`	存続期間など、管理者が指定した条件に基づいて、メッセージストアからメッセージを自動的に削除します。
`iminitquota`	LDAP ディレクトリから容量制限を再初期化し、使用中のディスクスペースを再計算します。

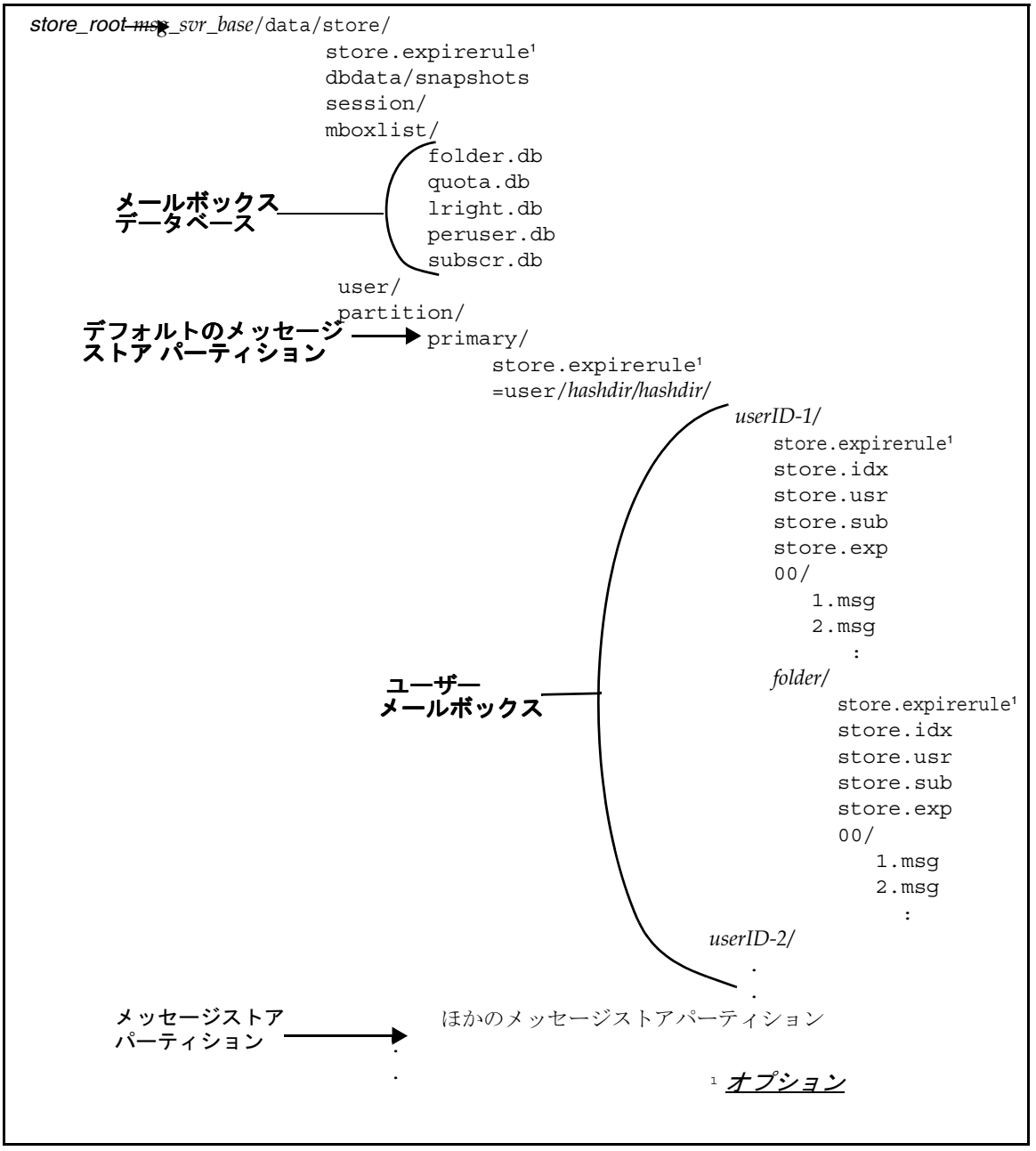
表 18-1　メッセージストアのコマンド行ユーティリティ (続き)

ユーティリティ	説明
imsasm	ユーザーメールボックスの保存と回復を行います。
imsbackup	保存したメッセージのバックアップを作成します。
imsexport	Certificate Management System のメールボックスを UNIX /var/mail 形式のフォルダ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復元します。
imscripter	IMAP サーバーのプロトコルスクリプティングツール。単独、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	バックアップディレクトリを作成、またはメッセージストア内の情報に合わせて同期化します。
MoveUser	ユーザーのアカウントを、別の Messaging Server に移動します。
imquotacheck	メッセージストア内の各ユーザーのメールボックスサイズの合計を計算し、制限容量と比較します。imquotacheck 通知のローカライズ版では、% 記号および $ 記号の変換が正しく行われません。エンコーディングを修正するには、メッセージファイル内のすべての $ を ¥24 で置き換え、すべての % を ¥25 で置き換えます。
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 18-1 は、サーバー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 18-2 を参照してください。

図 18-1　メッセージストアのディレクトリレイアウト



メッセージストアは、いくつかのメールボックスデータベースとユーザーメールボックスで構成されています。メールボックスデータベースは、ユーザー、メールボックス、パーティション、制限容量、およびその他のメッセージストア関連のデータで構成されています。ユーザーメールボックスには、ユーザーのメッセージとフォルダがあります。メールボックスは「メッセージストアパーティション」に格納されます。メッセージストアパーティションとは、「ディスク」パーティション上の、メッセージストアを格納するための専用エリアです。詳細は、619 ページの「メッセージストアのパーティションを構成する」を参照してください。メッセージストアパーティションはディスクパーティションと同じではありませんが、管理の便宜をはかるために、各メッセージストアパーティション用に 1 つのディスクパーティションを使用することをお勧めします。

INBOX などのメールボックスは、*store_root* にあります。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。

store_root/partition/primary/=user/53/53/=mack1

次の表で、メッセージストアディレクトリについて説明します。

表 18-2　メッセージストアのディレクトリの説明

場所	内容 / 説明
msg_svr_base	デフォルト : /opt/SUNWmsgsr サーバープログラム、設定、管理、および情報についてのファイルの格納に使用される、Messaging Server マシン上のディレクトリ。
store_root	*msg_svr_base*/data/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 サブディレクトリが格納されています。
./store.expirerule	メッセージを自動的に削除するルール (有効期限ルール) が格納されています。このオプションのファイルは別の場所に置くこともできます。605 ページの「自動メッセージ削除 (有効期限およびパージ) 機能を設定するには」を参照してください。
store_root/dbdata/snapshots	メッセージストアデータベースのバックアップスナップショット。

表 18-2　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_root/mboxlist/	メールボックスデータベース (Berkeley DB) が格納されています。このデータベースには、メールボックスや制限容量についての情報が保存されています。 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ます。 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.db には、ユーザーごとに 1 つのエントリが存在します。 lright.db - acl 検索権限別のフォルダのインデックス。 peruser.db には、ユーザー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、特定のユーザーがメッセージを開封したかどうか、または削除したかどうかを示します。 subscr.db には、ユーザーの購読に関する情報が格納されています。
store_root/session/	アクティブなメッセージストアプロセスについての情報が格納されています。
store_root/user/	使用されていません。
store_root/partition/	メッセージストアパーティションが格納されています。デフォルトで primary パーティションが作成されています。このディレクトリには、ほかのパーティションを定義して格納することもできます。
store_root/partition/primary/=user/	パーティションのサブディレクトリにある全ユーザー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。特定のユーザー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 を持つユーザー用のトップレベルのメールフォルダ。これがそのユーザーの INBOX です。デフォルトドメインでは、*userid* は *uid* となります。ホストしているドメインでは、*userid* は *uid@domain* となります。着信メッセージはこのメールフォルダに配信されます。
. . . /*userid*/*folder*	メッセージサーバー上のユーザー定義のメールボックス。

表 18-2　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	/*userid*/ ディレクトリに保存されたメールについての次の情報を提供するインデックス。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。さらにこのインデックスには、各ユーザーに関する mboxlist 情報のバックアップコピーや、各ユーザーに関する制限容量情報のバックアップコピーも含まれます。
.../*userid*/store.usr	フォルダにアクセスしたユーザーのリストが格納されています。リストされた各ユーザーについて、そのユーザーが最後にフォルダにアクセスした時間、ユーザーが表示したメッセージのリスト、ユーザー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.sub	ユーザーの購読に関する情報が格納されています。
.../*userid*/store.exp	削除されたものの、ディスクからは削除さ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、削除されたメッセージが存在する場合のみ表示されます。
.../*userid*/*nn*/ または .../*userid/folder/nn*/	*nn* は *message_id*.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あり、*nn* には 00 ～ 99 までの数字が入ります。*message_id* も数字です。たとえば、メッセージ 1 ～ 99 は .../00 ディレクトリに保存されます。最初のメッセージは 1.msg、2 番目のメッセージは 2.msg、3 番目のメッセージは 3.msg となり、以降同様に続きます。メッセージ 100 ～ 199 は 01 ディレクトリに保存され、メッセージ 9990 ～ 9999 は 99 ディレクトリに保存され、メッセージ 10000 ～ 10099 は 00 ディレクトリに保存されます。以降同様に続きます。

メッセージストアによるメッセージの削除方法

メッセージは、次の 3 段階の手順でメッセージストアから削除されます。

1. **削除** : クライアントがメッセージフラグを「削除」に設定します。この時点では、メッセージには削除のマークが付けられますが、クライアントは削除フラグを外せばメッセージを復元できます。第 2 のクライアントが存在する場合、そのクライアントからは削除されたフラグがただちには認識できない可能性があります。`configutil` パラメータの `local.imap.immediateflagupdate` を設定すると、フラグの更新がただちに行われるようになります。

2. **消去** : メッセージはメールボックスから削除されます。厳密には、メッセージはメッセージストアのインデックスファイルである `store.idx` から削除されます。メッセージ自体はディスクに残っていますが、メッセージの消去後、クライアントはメッセージを復元できなくなります。

   期限切れは、消去の特殊なケースです。メッセージのサイズや存続期間など、管理者が定義した一連の削除条件に適合するメッセージが消去されます。605 ページの「自動メッセージ削除 (有効期限およびパージ) 機能を設定するには」を参照してください。

3. **パージ** : `stored` ユーティリティにより、消去されたメッセージをすべてディスクからパージします。デフォルトの場合は、毎日午後 11 時に実行されます。この設定は、メッセージのパージスケジュールを制御する `local.schedule.purge` およびパージまでの猶予期間 (メッセージがパージされずに保持される期間) を制御する `store.cleanup` を使用して変更できます。

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、ユーザーのメールボックスを表示して監視したり、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、任意のユーザー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ー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	ほかのユーザー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、ほかの管理者がこれらの権限を持っている場合があります。

次の項で説明する管理者の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. 構成を行う Messaging Server をコンソールから開きます。
2. 「設定」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. 「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。
4. 「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。
5. 追加する管理者のユーザー 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ー ID は、Sun Java System Directory Server に認識されるものでなければなりません。
6. 「了解」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。
7. 「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。

コマンド行：コマンド行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。また、管理者は、サービス管理者グループのメンバーである必要があります (LDAP ユーザーエントリ：`memberOf:cn=Service Administrators,ou=Groups,o=usergroup`)。

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. 「了解」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。

コマンド行：コマンド行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。

コマンド行：コマンド行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。

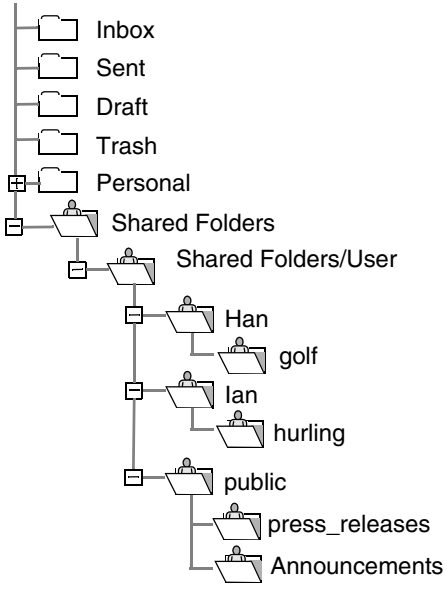
```
configutil -o store.admins -v "adminlist"
```

共有フォルダについて

「共有フォルダ」は、ユーザーのグループによってアクセスおよび読み取りが可能なフォルダです。言い換えると、共有フォルダへのアクセス権は、複数のユーザーに付与されます。たとえば、golf というフォルダを作成し、ほかのユーザーがそのフォルダの内容を表示することを許可できます。

デフォルトでは、Messaging Server によって電子メールアカウントに Shared Folders/Users というフォルダが作成されます。ユーザーはこのフォルダ内に共有フォルダを作成しアクセスします。図 18-2 に、クライアントで共有フォルダが表示される例を示します。この例については、589 ページの「分散共有フォルダを設定するには」でもさらに説明します。

図 18-2　Ed のクライアント共有メールフォルダリストの例



ユーザーは専用の共有フォルダを作成し、電子メールクライアントを使用してそのフォルダに対するアクセス権を付与できます。ただしその電子メールクライアントは共有フォルダをサポートしている必要があります。これらの共有フォルダは、アクセス権を持つほかのユーザーの Shared Folders に表示されます。

共有フォルダは、ある話題についてのリアルタイムの会話を開始、共有、アーカイブする場合に便利です。たとえば、ソフトウェア開発者のグループは、プロジェクトの進行状況について話し合うための共有フォルダを作成できます。メッセージが共有フォルダに送信されると、その共有フォルダを購読しているユーザーは誰でもそのメールボックスを開いてメッセージを読むことができます (購読者は個々のアドレスでもグループアドレスでも追加できる)。

共有フォルダには 2 種類あります。

- **非公開** - 非公開共有フォルダは、特定のユーザーが所有する共有フォルダです。フォルダの所有者がほかのユーザーにアクセスを付与します。
- **公開** - 公開共有フォルダには所有者がいません。管理者がパブリックユーザーアカウントを作成します。このアカウントを使用して公開フォルダをホストできます。公開フォルダの電子メールアドレスは次のようになります。

  `public+`*`foldername`*`@`*`domain`*

  たとえば、社内の何らかの会についての情報を送信するために `public+software_dev@siroe.com` などのフォルダを作成します。興味のある従業員に対して、この公開フォルダへのアクセス権を付与します。

通常、共有フォルダは特定のメッセージストア上のユーザーのみが使用できます。ただし、Messaging Server では、複数のメッセージストアからアクセスできる特殊な共有フォルダが作成できます。このようなフォルダは、「分散共有フォルダ」と呼ばれます。詳細は、589 ページの「分散共有フォルダを設定するには」を参照してください。

共有フォルダへのアクセス権

アクセス権は、`folder.db` に保存されているアクセス制御リスト (ACL) で保守されます。アクセス権は ACL を設定することで付与できます。ACL を設定するには、`readership` コマンド行ユーティリティで IMAP `SETACL` コマンド (`-s` オプション) を使用するか (588 ページの「公開フォルダのアクセス制御権を変更するには」を参照)、Messenger Express インタフェースを使用します。

ACL の識別子

各 ACL エントリには、エントリが適用されるユーザーまたはユーザーのグループを特定する識別子があります。ダッシュ記号 - で始まる識別子は、ユーザーまたはユーザーのグループに付与されていない権限です。

`anyone` は特別な識別子です。`anyone` のアクセス権は、すべてのユーザーに適用されます。同様に、`anyone@`*`domain`* のアクセス権は、同一ドメイン内のすべてのユーザーに適用されます。

グループの識別子は `group=` で始まります。

ACL 権限を示す文字

各 ACL エントリには、文字列で示される権限セットがあります。この文字列は RFC 2086 で定義されています。ユーザーの権限セットを計算するために、サーバーはユーザーとユーザーが属しているグループすべてに付与されているすべての権限を加算してから、ユーザーとユーザーが属しているグループに認められていないすべての権限を減算します。

次の表は、Messaging Server によって認識される文字の一覧です。文字の名前を示すとともに、各文字についての簡単な説明、および権限を持つユーザーが発行できる IMAP コマンドを示します。

表 18-3　ACL 権限を示す文字

文字	説明
l	lookup - ユーザーは共有フォルダを表示および購読できます (使用できる IMAP コマンド : LIST および LSUB)。
r	read - ユーザーは共有フォルダを読み取ることができます (使用できる IMAP コマンド : SELECT、CHECK、FETCH、PARTIAL、SEARCH、フォルダからの COPY)。
s	seen - セッション全体にわたって、開封済みの情報を保持するようにシステムに指示します (IMAP STORE SEEN フラグを設定すること)。
w	write - ユーザーは開封済みのマークを付けることができ、メッセージを削除できます (IMAP STORE フラグを SEEN および DELETED 以外に設定すること)。
i	insert - ユーザーは電子メールをあるフォルダから別のフォルダにコピーおよび移動できます (使用できる IMAP コマンド : フォルダへの APPEND、COPY)。
p	post - ユーザーは共有フォルダ電子メールアドレスにメールを送信できます (IMAP コマンドは不要)。
c	create - ユーザーは新規のサブフォルダを作成できます (使用できる IMAP コマンド : CREATE)。
d	delete - ユーザーは共有フォルダからエントリを削除できます (使用できる IMAP コマンド : EXPUNGE、STORE DELETED フラグをセットすること)。
a	administer - ユーザーは管理者権限を持ちます (使用できる IMAP コマンド : SETACL)。

グループ ACL

ACL エントリの識別子で、グループ名を指定できます。このエントリのアクセス権は、グループのすべてのメンバーに適用されます。グループのメンバーは、inetMailUser オブジェクトクラスの aclGroupAddr 属性を使用してサーバーによって決定されます。aclGroupAddr 属性のフィルタを介して、グループはダイナミックなメーリングリストに示されます。次に、グループを定義する LDIF レコードの例を示します。これには aclGroupAddr 属性も含まれます。

```
dn:cn=lee-staff,ou=Groups, o=sesta.com
cn:lee-staff
mailHost:mail.sesta.com
inetMailGroupStatus:active
mgrpErrorsTo:lee.jones@sesta.com
description:Dynamic Group of Lee's staff
objectClass:top
objectClass:groupofuniquenames
objectClass:inetmailgroup
objectClass:inetmailgroupmanagement
objectClass:inetlocalmailrecipient
objectClass:groupofurls
mail:lee-staff@sesta.com
memberURL:ldap:///o=sesta.com??sub?
(&(aclGroupAddr=lee-staff@sesta.com)(objectclass=inetmailuser))
```

フォルダの ACL で使用されるグループ電子メールアドレスは、必ずしもグループ用に作成されるわけではありません。実際には、グループにメンバーを追加する際に、このようなダイナミックグループを作成し、ユーザーエントリに対して aclGroupAddr 属性を設定することがあります。このようなグループが作成されると、スタティックな外部メンバーを mgrpRfc822MailMember 属性にある電子メールアドレスを使用して追加できるようになります。メンバーの追加には uniqueMember 属性を使用したり、memberURL 属性に値を追加したりしないでください。これを行うと、MTA がメーリングリストのメンバーとして認識している内容と IMAP サーバーがグループメンバーとして認識している内容が切断されます。

ユーザーが IMAP サーバーにログインしたり、Messenger Express などの HTTP アクセスサービスクライアントを使用してログインすると、サーバーは aclGroupAddr 属性をほかのメッセージストア関連の属性とともに取り込み、グループ名をメモリにキャッシュします。サーバーはこの情報を使用して、クライアントがアクセス権の確認が必要なコマンド (LIST、SELECT など) を発行するたびにユーザーのアクセス権を判断します。

共有フォルダに関するタスク

この節では、共有フォルダ管理者のタスクについて説明します。

公開フォルダを作成するには

公開フォルダの場合は、LDAP データベースおよび `readership` コマンドへのアクセスが必要であるため、システム管理者が作成する必要があります。

1. すべての公開フォルダのコンテナとして機能する LDAP ユーザーエントリを追加します。たとえば、`public` というエントリを追加します。

```
dn:cn=public,ou=people,o=sesta.com,o=ISP
objectClass:person
objectClass:organizationalPerson
objectClass:inetOrgPerson
objectClass:inetUser
objectClass:ipUser
objectClass:inetMailUser
objectClass:inetLocalMailRecipient
objectClass:nsManagedPerson
objectClass:userPresenceProfile
cn:public
mail:public@sesta.com
mailDeliveryOption:mailbox
mailHost:manatee.siroe.com
uid:public
inetUserStatus:active
mailUserStatus:active
mailQuota: -1
mailMsgQuota: 100
```

2. `mboxutil` コマンド行ユーティリティを使用して、パブリックアカウント内にフォルダを作成します。次に例を示します。

```
mboxutil -c user/public/golftournament
```

3. readership コマンド行ユーティリティを使用して、このフォルダに適した ACL を設定します。

   このフォルダを公開するためには、アクセスできるユーザーグループをフォルダに割り当てる必要があります。そのためには、readership コマンドを使用して ACL を設定します。ACL の設定方法については、このあとに続く 588 ページの「公開フォルダのアクセス制御権を変更するには」を参照してください。

公開フォルダのアクセス制御権を変更するには

公開フォルダのアクセス制御を変更したり、新規に作成した公開フォルダのアクセス制御を設定したりする必要が生じることがあります。

これを実行するには、readership コマンド行ユーティリティを使用します。このコマンドの形式は、次のとおりです。

```
readership -s foldername identifier rights_chars
```

foldername は権限設定対象の公開フォルダの名前、*userid* は権限割り当て先の個人またはグループ、*rights_chars* は割り当てる権限です (これらは、RFC 2086 準拠のアクセス制御文字)。各文字の意味については、585 ページの「ACL 権限を示す文字」を参照してください。公開フォルダのアクセス制御は、Messenger Express インタフェースを使用しても変更できます。

例

たとえば、sesta ドメインの全員に公開フォルダ golftournament の検索、読み取り、電子メールのマーク付けができる (ただし、送信は除く) アクセス権を付与する場合は、次のようにコマンドを発行します。

```
readership -s User/public/golftournament anyone@sesta lwr
```

検索、読み取り、電子メールのマーク付け、および送信する権限をグループに割り当てる場合は、次のようにコマンドを発行します。

```
readership -s User/public/golftournament group=golfinterest lwrp
```

このフォルダの管理者権限と送信権限を jdoe という個人に割り当てる場合は、次のようにコマンドを発行します。

```
readership -s User/public/golftournament jdoe lwrpa
```

公開フォルダへの個人またはグループのアクセスを拒否するには、ダッシュを userid の前に付けます。たとえば、jsmith の検索、読み取り、書き込み権限を拒否するには、次のようにコマンドを発行します。

```
readership -s User/public/golftournament -jsmith lwr
```

共有フォルダの一覧表示を有効化または無効化するには

設定オプション local.store.sharedfolders の設定によって、サーバーは LIST コマンドに対する応答として共有フォルダを返す場合と返さない場合があります。このオプションを off に設定すると、オプションは無効になります。デフォルトでは、この設定は有効 (on) になっています。

SELECT および LSUB コマンドは、このオプションの影響を受けません。LSUB コマンドは、すべての購読されているフォルダを返します。これには共有フォルダが含まれます。ユーザーは SELECT を使用して所有するフォルダや購読しているフォルダを選択できます。

分散共有フォルダを設定するには

通常、共有フォルダは特定のメッセージストア上のユーザーのみが使用できます。ただし、Messaging Server では、複数のメッセージストアからアクセスできる「分散共有フォルダ」が作成できます。つまり、分散共有フォルダへのアクセス権は、メッセージストアグループ内の任意のユーザーに付与できます。ただし、Web メールクライアント (Messenger Express などの HTTP アクセスクライアント) では、リモートでの共有フォルダアクセスがサポートされていないことに注意してください。ユーザーは共有フォルダを一覧表示および購読できますが、内容を表示したり変更したりすることはできません。

分散共有フォルダでは、次の条件を満たす必要があります。

- メッセージストア userid は、メッセージストアグループ全体で一意である。
- 配備全体で、ディレクトリデータが同一である。

リモートのメッセージストア (つまり、共有フォルダを保持していないメッセージストア) は、589 ページの表 18-4 に示されている設定変数を使用してプロキシサーバーとして設定されている必要があります。

表 18-4　分散共有フォルダの設定に使用する変数

名前	値	データ形式
local.service.proxy.serverlist	メッセージストアのサーバーリスト	スペースで区切られた文字列
local.service.proxy.admin	デフォルトのストア管理者ログイン名	文字列
local.service.proxy.adminpass	デフォルトのストア管理者パスワード	文字列

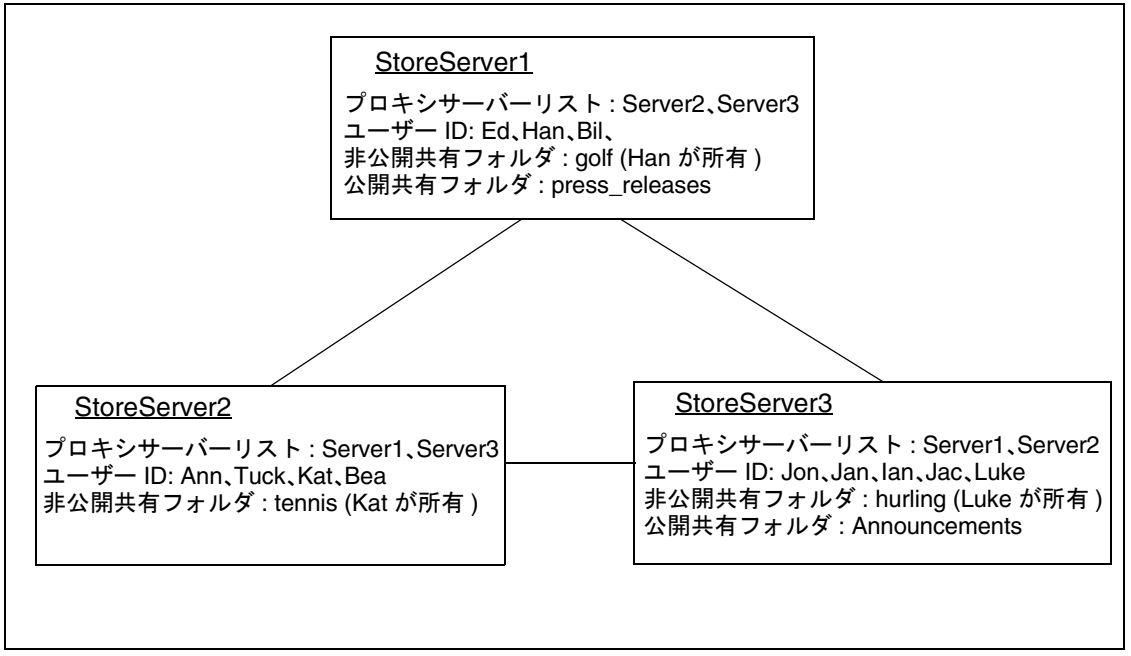
表 18-4　分散共有フォルダの設定に使用する変数 (続き)

名前	値	データ形式
`local.service.proxy.admin.`*hostname*	特定ホスト用のストア管理者ログイン名	文字列
`local.service.proxy.adminpass.`*hostname*	特定ホスト用のストア管理者パスワード	文字列

分散共有フォルダの設定例

図 18-3 に、StoreServer1、StoreServer2、および StoreServer3 という 3 つのメッセージストアサーバーで共有する分散フォルダの例を示します。

図 18-3　分散共有フォルダの例



これらのサーバーは、表 18-4 で示されている変数を設定することによって、ピアプロキシメッセージストアとして互いに接続しています。各サーバーには、*golf* (Han が所有)、*tennis* (Kat が所有)、および *hurling* (Luke が所有) という非公開共有フォルダがあります。さらに、*press_releases* および *Announcements* という 2 つの公開共有フォル

ダもあります。これら 3 つのサーバーのいずれかに存在するユーザーは、これら 3 つの共有フォルダすべてにアクセスできます。583 ページの図 18-2 に、Ed の共有フォルダリストが示されています。次に、この構成における各サーバーの ACL の例を示します。

```
$ StoreServer1 :> readership -l
Ed: user/Han/golf
Ian: user/Han/golf
anyone: user/public/press_releases
```

```
$ StoreServer2 :> readership -l
Jan: user/Kat/tennis
Ann: user/Kat/tennis
anyone: user/public+Announcements user/public+press_releases
```

```
$ StoreServer3 :> readership -l
Tuck: user/Ian/hurling
Ed: user/Ian/hurling
Jac: user/Ian/hurling
anyone: user/public/Announcements
```

共有ファイルデータを監視および保守するには

readership コマンド行ユーティリティを使用すると、folder.db、peruser.db、および lright.db の各ファイルに保存されている共有フォルダデータを監視および保守できます。folder.db には、ACL のコピーが格納された各フォルダの記録があります。peruser.db には、ユーザーおよびメールボックスごとのエントリがあり、このエントリには、各種フラグ設定およびユーザーが任意のフォルダに前回アクセスした日付が示されています。lright.db には、全ユーザーの一覧があり、ユーザーが検索権限を持つ共有フォルダも示されています。

readership コマンド行ユーティリティでは次のオプションが使用できます。

表 18-5 readership オプション

オプション	説明
-d days	指定した日数以内にフォルダを選択したユーザーの数を共有フォルダごとに示すレポートを返します。
-p months	指定した月数以内に共有フォルダを選択していなかったユーザーの peruser.db からデータを削除します。
-l	lright.db. 内のデータをリスト表示します。
-s *folder_identifier_rights*	指定したフォルダにアクセス権を設定します。これによって、lright.db および folder.db が更新されます。

さまざまなオプションを使用して、次の機能を実行できます。

- 592 ページの「共有フォルダ使用状況を監視するには」
- 593 ページの「ユーザーとその共有フォルダを一覧表示するには」
- 593 ページの「非アクティブなユーザーを削除するには」
- 593 ページの「アクセス権を設定するには」

共有フォルダ使用状況を監視するには

共有フォルダに積極的にアクセスしているユーザーの数を調べるには、次のコマンドを発行します。

readership -d *days*

days は、チェック対象とする日数です。このオプションではアクティブなユーザーの数が返されるのであって、アクティブなユーザーの一覧が返されるのではないことに注意してください。

例 : 過去 30 日以内に共有フォルダを選択したユーザーの数を調べるには、次のようにコマンドを発行します。

```
readership -d 30
```

ユーザーとその共有フォルダを一覧表示するには

ユーザーおよびユーザーがアクセスした共有フォルダを一覧表示するには、次のコマンドを発行します。

```
readership -l
```

出力例 :

```
$ readership -l
group=lee-staff@siroe.com: user/user2/lee-staff
richb: user/golf user/user10/Drafts user/user2/lee-staff
user/user10/Trash
han1: user/public+hurling@siroe.com user/golf
gregk: user/public+hurling@siroe.com user/heaving user/tennis
```

非アクティブなユーザーを削除するには

非アクティブなユーザー (指定の期間内に共有フォルダにアクセスしなかったユーザー) を削除するには、次のコマンドを発行します。

```
readership -p months
```

months は、チェックに使用する月数です。

例 : 過去 6 か月間に共有フォルダにアクセスしなかったユーザーを削除します。

```
readership -p 6
```

アクセス権を設定するには

新規の公開フォルダにアクセス権を割り当てたり、現在の公開フォルダのアクセス権を変更したりできます。

このコマンドを使用したアクセス権の設定方法の例については、588 ページの「公開フォルダのアクセス制御権を変更するには」を参照してください。

メッセージストアの制限容量について

メッセージストアの制限容量は、ユーザーまたはドメインが使用できるディスク容量またはメッセージ数の「制限容量」を設定するしくみです。この節では、以下の情報について説明します。

- 594 ページの「ユーザーの制限容量」
- 595 ページの「ドメインの制限容量」
- 595 ページの「Telephony Application Server に関する例外」

詳細は、630 ページの「制限容量を監視するには」を参照してください。

ユーザーの制限容量

ユーザーの制限容量は、ディスク容量またはメッセージの数によって指定できます。ディスク制限容量は、各ユーザーに割り当てられるディスク容量をバイト単位で指定します。ディスク制限容量は、ユーザーのメールフォルダの数に関係なくユーザーのメッセージの合計サイズに適用されるか、ユーザーメッセージの合計数に適用されます。メッセージ制限容量は、ユーザー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 (表 18-6) および `configutil` 変数 (表 18-7) に保存されます。最新の詳細情報については、『Sun Java System Communications Services Schema Reference Manual』(`http://docs.sun.com/doc/819-0113`) を参照してください。Messaging Server では、制限容量を設定する以外にも、以下の機能を制御できます。

- **制限容量の通知**は、「ディスク制限容量のしきい値」に達したときに、ユーザーに警告メッセージを送信します。
- **制限容量の適用**は、制限容量を超過したあとはメッセージストアへのメッセージの配信を停止します。制限容量を超過した場合でもメッセージを配信できるようにすることもできます。

  制限容量を超過したためにメッセージ配信が停止された場合、以下のどちらかの状態になるまで、着信メッセージは MTA キューに残ったままとなります。

  - ユーザー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 によってメッセージが配信されます。
  - 未配信のメッセージの MTA キューに残留している期間が、メッセージが差出人に返されるよう指定された猶予期間を超えてしまったとき (604 ページの「猶予期間を設定するには」を参照)。

  ディスク容量は、ユーザー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ーがメッセージを削除したときに使用可能になります。

- **デフォルトの制限容量**は、すべてのユーザーに対してデフォルトの制限容量を設定するか、ユーザーごとに異なる制限容量を設定します。ユーザー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、まず個々のユーザー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 はすべてのユーザー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。

ドメインの制限容量

ユーザーの場合と同様、制限容量はドメインに対しても、バイト数またはメッセージ数のどちらかを設定できます。この制限容量は、特定のドメイン内のすべてのユーザーの、累積されたバイトまたはメッセージすべてに対するものです。

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 12 章「チャネル定義を設定する」を参照してください。

表 18-6 に、制限容量の LDAP 属性を示します。最新の詳細情報については、『Sun Java System Communications Services Schema Reference Manual』(http://docs.sun.com/doc/819-0113) を参照してください。

表 18-6 メッセージストアの制限容量の属性

属性	説明
mailQuota	ユーザーのメールボックスに指定できるディスク容量のバイト。固有の値は次のとおりです。 0 - ユーザーのメールボックスに領域を割り当てません。 -1 - 領域の容量を制限しません。 -2 - システムのデフォルトの容量を使用します (`configutil` パラメータ `store.defaultmailboxquota`)。
mailMsgQuota	ユーザーに許可された最大メッセージ数。ストア内のすべてのフォルダの累積カウント。固有の値は次のとおりです。 0 - ユーザーのメールボックスにメッセージを割り当てません。 -1 - 許可するメッセージ数を制限しません。 -2 - システムのデフォルトの容量を使用します (`configutil` パラメータ `store.defaultmessage.quota`)。

表 18-6　メッセージストアの制限容量の属性 (続き)

属性	説明
mailUserStatus	メールユーザーのステータス。次の値のいずれかとなります。 `active` - メールは通常どおり処理されます。デフォルトは `active`。 `inactive` - ユーザーのメールアカウントが非アクティブ。一時的な失敗が返されます。 `deleted` - 削除済みでパージ可能というマークが付いたアカウント。永久的な失敗が返されます。メールボックスへのアクセスがブロックされます。 `hold` - 保留キューに送信され、メールボックスにアクセスするメールが拒否されます `overquota` - このステータスの場合、MTA はメールをメールボックスに配信しません。これは、`configutil` パラメータの `store.overquotastatus` が `on` の場合に設定されるステータスです。
mailDomainDiskQuota	ドメイン内のすべてのメールボックスの累積カウントに指定できるディスク容量のバイト。-1 の値は、領域の容量を制限しないことを示します (デフォルト)。ドメインのディスク制限容量を適用するには、コマンド `imquotacheck -f -d` *domain* を実行します。
mailDomainMsgQuota	ドメインに許可された最大メッセージ数。つまり、ストア内のすべてのメールボックスの総数。-1 の値は、制限がないことを示します (デフォルト)。ドメインのメッセージ制限容量を適用するには、コマンド `imquotacheck -f -d` *domain* を実行します。
mailDomainStatus	メールドメインのステータス。値とデフォルトは「mailUserStatus」と同じ。

表 18-7　メッセージストアの `configutil` パラメータ

パラメータ	説明
store.quotaenforcement	制限容量の適用を有効にします。オフの場合も、制限容量データベースは更新されますが、メッセージは常に配信されます。 デフォルト : On
store.quotanotification	制限容量の通知を有効にします。デフォルト : On
store.defaultmailboxquota	デフォルトの制限容量をバイト数によって保存します。 デフォルト : -1 (無制限)
store.defaultmessagequota	デフォルトの制限容量をメッセージ数によって保存します。数値。 デフォルト : -1 (無制限)
store.quotaexceededmsg	制限容量の警告メッセージ。指定しない場合、通知は送信されません。 デフォルト : なし。

表 18-7　メッセージストアの configutil パラメータ (続き)

パラメータ	説明
store.quotaexceededmsginterval	制限容量の超過の通知を送信する間隔 (日単位)。デフォルト : 7
store.quotagraceperiod	メールボックスへのメッセージが差出人に戻されるときの、メールボックスが制限容量を超過した時間。時間数。デフォルト : 120
store.quotawarn	制限容量の警告のしきい値。クライアントに制限容量の超過の警告が送信されるときの、制限容量を超えるパーセント。デフォルト : 90
local.store.quotaoverdraft	Netscape Messaging Server から移行したシステムとの互換性を提供するために使用されます。ON のとき、ディスク容量が制限容量を超過するメッセージを 1 つ配信できます。制限容量を超過すると、メッセージが遅延またはバウンスされ、制限容量の警告メッセージが送信され、制限容量の猶予期間のタイマーが開始されます。デフォルトでは、メッセージストアがしきい値に達したときに、制限容量の警告メッセージが送信されます。デフォルト : Off。ただし、store.overquotastatus が設定されている場合は on とみなされます。そうでない場合、ユーザーは制限容量を超過することはできず、overquotastatus が使用されることはありません。
local.store.overquotastatus	メッセージが MTA のキューに入れられる前に、制限容量の適用を有効にします。これによって、MTA キューがいっぱいになりません。このパラメータが設定されている場合、ユーザーがまだ制限容量を超過していないが、着信メッセージが制限容量を超過すると、メッセージが配信され、MTA がそれ以降はメッセージを受け入れないように mailuserstatus LDAP 属性に overquota が設定されます。デフォルト : off

メッセージストアの制限容量を設定する

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ー制限容量を指定するには

個別の制限容量が設定されていないユーザーに適用するデフォルトの制限容量を設定するには、以下の手順を実行します。

コンソール：コンソールでデフォルトのユーザー制限を指定するには、以下の手順に従います。

1. 「設定」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。
2. 「制限容量」タブをクリックします。
3. デフォルトのユーザーディスク制限容量を指定するには、「デフォルトのユーザーディスク制限容量」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。

   サイズ制限：このオプションは、デフォルトのユーザー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「K バイト」または「M バイト」を選択します。
4. 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ーメッセージ制限容量」ボックスに数字を入力します。
5. 「保存」をクリックします。
6. デフォルトのメッセージストアの制限容量を使用する場合は、ユーザーエントリで、「M バイト」属性を -1 に設定します。表 18-6 を参照してください。

コマンド行：コマンド行でデフォルトのユーザー制限を指定するには、以下の手順に従います。

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ー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ー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。

デフォルトのメッセージストアの制限容量を使用する場合は、ユーザーエントリで、`mailQuota` 属性を `-2` に設定します。表 18-6 を参照してください。

個々のユーザー制限容量を指定するには

各ユーザーは、制限容量を個別に設定できます。ユーザー固有の制限容量を設定するには、ユーザーの LDAP エントリで `mailQuota` または `mailmsgquota` 属性を設定します (表 18-6 を参照)。制限容量を適用するには、`configutil store.quotaenforcement` を `on` に設定します。

ドメイン制限容量を指定するには

ディスク容量の制限容量またはメッセージの制限容量は、特定のドメインに対して設定できます。これらの制限容量は、特定のドメイン内のすべてのユーザーの、累積されたバイトまたはメッセージすべてに対するものです。ドメインの制限容量を設定するには、ユーザーの LDAP エントリで `mailDomainDiskQuota` または `mailDomainMsgQuota` 属性を設定し (表 18-6 を参照)、`imquotacheck -f` を実行します。

制限容量の通知を配備するには

制限容量の通知とは、制限容量に近づいたときにユーザーに警告メッセージを送信する処理のことです。この機能を使用するには、3 つの手順が必要です。

- 600 ページの「制限容量の通知を有効にする」
- 600 ページの「制限容量の警告メッセージを定義する」
- 601 ページの「制限容量のしきい値の指定」

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。制限容量の通知を無効にするには、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。600 ページの「制限容量の警告メッセージを定義する」を参照してください。
4. 「保存」をクリックします。

コマンド行 : コマンド行で制限容量の通知を有効または無効にするには、以下のようにします。

```
configutil -o store.quotanotification -v [ yes | no ]
```

メッセージに何も設定されなかった場合、ユーザー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形式については、次の節を参照してください。

制限容量の警告メッセージを定義する

ディスク制限容量を超えそうなユーザーに送信するメッセージは、以下の手順で定義します。メッセージはユーザー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。

コマンド行 : コマンド行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。メッセージには少なくとも件名行を含むヘッダーがあり、$$、メッセージ本文がそのあとに続いている必要があります。'$' は、新しい行を表します。使用しているシェルによっては、$ の前に ¥ を追加して、$ が持つ特殊な意味をエスケープする必要があることもあります (ほとんどの場合、$ はシェルのエスケープ文字)。次に例を示します。

```
configutil -o store.quotaexceededmsg -v 'Subject: WARNING: User quota
exceeded$$User quota threshold exceeded - reduce space used.'
```

さらに、次の変数がサポートされます。

[ID] - ユーザー ID

[DISKUSAGE] - ディスク使用量

[NUMMSG] - メッセージの数

[PERCENT] - store.quotawarn パーセンテージ

[QUOTA] - mailquota 属性

[MSGQUOTA] - mailmsgquota 属性

次にこれらの変数の使用例を示します。

```
configutil -o store.quotaexceededmsg -v 'Subject:Overquota
Warning$$[ID],$$Your mailbox size has exceeded [PERCENT] of its
alloted quota.$Disk Usage:[DISKUSAGE]$Number of
Messages:[NUMMSG]$Mailquota:[QUOTA]$Message Quota:
[MSGQUOTA]$$-Postmaster'
```

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.quotaexceededmsginterval -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は、クライアントに警告が送信されるときの、制限容量を超えた割合です。ユーザーのディスク使用量が指定したしきい値を超えたら、サーバーからユーザーに警告メッセージが送信されます。

注 local.store.quotaoverdraft=on 電子メール通知は、store.quotawarn で設定されたしきい値に関係なくユーザーのディスク使用量が制限容量の 100% を超えるまでトリガされません。

クライアントが IMAP ALERT 機能をサポートしている IMAP ユーザーの場合は、ユーザー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、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、ユーザー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。

コマンド行 : コマンド行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。

制限容量の適用を有効または無効にするには

デフォルトでは、制限容量を超えてもユーザーまたはドメインには何の影響もなく、制限容量超過の通知が設定されている場合に通知を受信するだけです。制限容量を適用すると、ディスク容量が制限容量レベルを下回るまで、それ以上メッセージを受信しないようメールボックスをロックします。

制限容量の適用をユーザーレベルで有効にする

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。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。

コマンド行 : 制限容量の適用を有効または無効にするには、以下のようにします。

```
configutil -o store.quotaenforcement -v [ on | off]
```

制限容量を超過すると、メッセージは MTA キューに保存され、メッセージが配信されなかったが、あとで再配信が試行されることを示す通知が差出人へ送信されます 配信の再試行は、猶予期間の期限が切れて、すべてのメッセージが差出人に戻されるまで、またはディスク使用量が制限容量を下回り、メッセージを MTA キューから取り出してメッセージストアに配信できるようになるまで続行されます。制限容量を超えた場合に、メッセージをキューに入れる前にメッセージを返す場合は、次のコマンド行を使用します。

```
configutil -o store.overquotastatus -v on
```

制限容量の適用をドメインレベルで有効にする

特定のドメインの制限容量を適用するには、次のコマンドを使用します。

```
imquotacheck -f -d domain
```

すべてのドメインについて有効にするには、-d オプションを除外します。ドメインがその制限容量を超過すると、maildomainstatus 属性が overquota に設定され、このドメインへの全配信が停止します。ドメインが overquota でない場合、値は active に設定されます。

制限容量の適用を無効にする

ユーザーの制限容量が適用されているようである場合は、制限容量を無効にした場合でも、次のパラメータを確認してください。

次の configutil パラメータは、オフにするか設定しないようにする必要があります。

- store.quotaenforcement
- local.store.overquotastatus
- local.store.quotaoverdraft

store.overquotastatus が on の場合、常に store.quotaoverdraft が on であるとみなし、そうでない場合はユーザーは制限容量を超過して拒否を引き起こすことはありません。また、store.quotaoverdraft が on の場合、ユーザーは制限容量よりも小さいメッセージを 1 つだけ許可されます。すなわち、ユーザーの制限容量よりも大きいメッセージは受け入れません。

これらのパラメータを変更したあとは、必ずメッセージングサービスを再起動してください。

次のメッセージストア属性はアクティブにする必要があります。

- maildomainstatus
- mailuserstatus

メッセージがメールボックスの制限容量よりも大きい場合は、制限容量の適用設定とは無関係にバウンスされます。

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。
- ユーザー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。この時点でサーバーが、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。この時間制限は、`quotagraceperiod configutil` パラメータによって制御されます。
- メッセージがメッセージキューの最大時間を過ぎてもメッセージキューに残っているとき。これは、`notices` MTA チャネルキーワードによって制御されます (283 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)。

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、メッセージ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、メッセージは差出人に戻されます。

注	猶予期間とは、メッセージがメッセージキュー内に保持される期間ではなく、メッセージキュー内に含まれているすべての着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。猶予期間は、ユーザーが制限容量のしきい値に達し、警告を受けたときに開始します。601 ページの「制限容量のしきい値の指定」を参照し、注意してください。

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「日」または「時間」を指定します。
4. 「保存」をクリックします。

コマンド行 : コマンド行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。

Netscape Messaging Server の制限容量の互換性モード

Netscape Messaging Server のディスク使用量が制限容量を超過すると、サーバーはメッセージの配信を延期またはバウンスし、制限容量超過通知を送信して、猶予期間を開始します。Messaging Server には、この動作を保持するパラメータ `local.store.quotaoverdraft` があります。

`ON` に設定すると、メッセージはディスクの使用量が制限容量を超過するまで配信されます。超過時には、メッセージが延期され (メッセージは MTA メッセージキューに保持されるが、メッセージストアに配信されない)、制限容量超過警告メッセージがユーザーに送信されて、猶予期間が開始されます。猶予期間は、制限容量超過メッセージがバウンスされるまで、メールボックスが制限容量超過である期間を決定します。デフォルトでは、メッセージストアがしきい値に達したときに、制限容量の警告メッセージが送信されます。このパラメータのデフォルトは、`Off` です。

自動メッセージ削除 (有効期限およびパージ) 機能を設定するには

自動メッセージ削除機能 (有効期限切れおよびパージとも呼ばれる) を使用すると、管理者が定義した一連の条件に基づいて、メッセージストアからメッセージが自動的に削除されます。この機能によって、古いメッセージやサイズの大きいメッセージ、開封済みまたは削除済みメッセージ、特定の Subject: 行を持つメッセージなどを自動的に削除できます。次の削除条件が設定できます。

- フォルダ (メールボックス) 別、ユーザー別、ドメイン別、メッセージストア全体、または特定のパーティション
- 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
- メールボックスでのメッセージの存続期間 (日数)
- メッセージのサイズと猶予期間 (サイズ超過のメッセージをパージする前にメッセージストアに残しておく日数)
- メッセージのフラグが *seen* か *deleted* かどうか
- ヘッダー文字列

この機能は、メッセージの消去やパージを行う `imexpire` ユーティリティを使用して実行します。メッセージ削除プロセスの詳細については、580 ページの「メッセージストアによるメッセージの削除方法」を参照してください。

注	サーバーによってメッセージは警告なしに削除されます。したがって、自動メッセージ削除ポリシーについてユーザーに知らせておくことは重要です。メッセージが突然削除されると、ユーザーや管理者は大変驚くことになるからです。

imexpire の動作方式

imexpire は、コマンド行から呼び出すか、imsched デーモンを使用して自動的に実行されるようにスケジュールします。管理者は、コンソールまたは configutil コマンド行ユーティリティを使用して、グローバル有効期限ルール (メッセージストア全体に適用されるルール) を設定します。ローカル有効期限ルール (フォルダまたはユーザーに適用されるルール) は、有効期限ルールファイル (store.expire) をメッセージストアパーティション、ユーザーまたはメールボックスディレクトリに作成することで設定できます。

imexpire は、起動時にすべての有効期限ルールをロードします。デフォルトでは、imexpire はパーティションごとに 1 つのスレッドを作成します。各スレッドは割り当てられたパーティションの下にあるユーザーフォルダのリストを通過し、その間にローカル有効期限ルールをロードします。この有効期限機能により、各フォルダは有効期限ルールに照らしてチェックされ、メッセージは必要に応じて消去されます。メールボックスディレクトリの内に store.exp ファイルが存在し、store.cleanupage 設定パラメータで指定した期間を過ぎていたために消去されたり期限切れになっていたりするメッセージがある場合は、パージ機能によってメッセージハッシュディレクトリ内にあるメッセージファイルが完全に削除され、store.exp ファイルから UID のレコードが削除されます。

msg_svr_base/config/ 内の expire_exclude_list と呼ばれるファイルで、1 行に 1 つずつユーザー ID を追加して、指定したユーザーを有効期限ルールから除外することもできます。

自動メッセージ削除機能を配備するには

自動メッセージ削除は、コマンド行を使用するか、コンソールの GUI を使用して配備できます。この処理には 3 つの手順が必要です。

1. メッセージの自動削除ポリシーを定義します。自動削除するメッセージ、自動削除するメッセージを所有しているユーザー、ドメイン、パーティション、およびサイズ、メッセージ存続期間、ヘッダーについて特定して削除条件を定義します。607 ページの「自動メッセージ削除ポリシーを定義するには」を参照してください。

2. imexpire ルールを指定してこのポリシーを実装します。608 ページの「自動メッセージ削除ポリシーを実装するルールを設定するには」を参照してください。
3. imexpire スケジュールを指定します。616 ページの「自動メッセージ削除とログレベルをスケジュールするには」を参照してください。

自動メッセージ削除ポリシーを定義するには

削除条件を指定して独自の自動メッセージ削除ポリシーを定義します。Imexpire を使用すると、次の条件を使用する削除が可能になります。

メッセージの存続期間 : X 日間より存続期間が長いメッセージを自動的に削除します。属性 : messagedays。

メッセージの件数 : X 件を超えたフォルダ内のメッセージを自動的に削除します。属性 : messagecount。

サイズ超過メッセージの存続期間 : X バイトを超えるメッセージを Y 日間の猶予期間後に自動的に削除します。属性 : messagesize および messagesizedays。

開封済みおよび削除済みメッセージフラグ :「開封済み」または「削除済み」フラグが付いているメッセージを自動的に削除します。これらの条件には、「and」または「or」が設定できます。or に設定した場合、メッセージに開封済みまたは削除済みフラグが付いていると、ほかの条件にかかわらず自動削除されます。and に設定した場合、メッセージに付いている開封済みまたは削除済みフラグは、指定したほかの条件すべてを満たした場合に設定されます。属性 : seen および deleted。

メッセージのヘッダーフィールド : メッセージを削除する条件としてヘッダーおよび文字列を指定できます。たとえば、「Subject: Work from Home!」というヘッダーがあるメッセージをすべて削除できます。

メッセージのフォルダ : メッセージを削除するフォルダを指定できます。属性 : folderpattern

注	imexpire を使用して、メッセージが開封されてからの期間に基づいてメッセージを削除または保存することはできません。たとえば、200 日経過しても読まれていないメッセージを削除するという指定はできません。

自動メッセージ削除ポリシーの例

例 1: 1,000 件を超えるメッセージが存在するフォルダ内の、存続期間が 365 日のメッセージをすべて削除します。

例 2: ドメイン siroe.com 内の、存続期間が 180 日を超えるメッセージを削除します。

例 3:「削除済み」のマークが付いているメッセージをすべて削除します。

例 4: sesta.com 内の 1,000 件を超えるメッセージが存在するフォルダから、「開封済み」マークが付いていて、存続期間が 30 日より長く、サイズが 100K バイトより大きく、X-spam というヘッダーが付いたメッセージを削除します。

自動メッセージ削除ポリシーを実装するルールを設定するには

前の節で定義した自動メッセージ削除ポリシーを実装するには、imexpire ルールを設定する必要があります。次のようにルールを定めます。

- GUI を使用する (614 ページの図 18-4 を参照)
- store.expirerule ファイルにルールを追加する。2 つの store.expirerule ルールの例を次に示します。

```
Rule1.folderpatter:user/.*/trash
Rule1.messagedays: 2
Rule2.folderpattern:user/.*
Rule2.messagedays: 14
```

この例では、Rule 1 でごみ箱フォルダ内のすべてのメッセージが 2 日後に削除されることを指定しています。Rule 2 ではメッセージストアのすべてのメッセージが 14 日後に削除されることを指定しています。

この節には、以下の項があります。

- 608 ページの「有効期間ルールのガイドライン」
- 611 ページの「imexpire ルールをテキストモードで設定する」
- 612 ページの「imexpire フォルダパターンを設定する」
- 613 ページの「コンソールを使用して自動メッセージ削除ルールを設定するには」

有効期間ルールのガイドライン

この節では、store.expirerule ファイルルールのガイドラインを示します。

注 以前のリリースの Messaging Server では、有効期限ルールは、configutil パラメータの store.expirerule.*attribute* で設定できました (Sun Java System Messaging Server Administration Reference を参照)。このリリースでもこれは引き続き可能ですが、ヘッダーの制約を使用した有効期限ルール (特定の件名でメッセージの有効期限が切れるなど) はサポートされません。このため、コンソールの GUI または store.expirerule ファイルルールを使用するほうが良い場合があります。

- ルールは、store.expirerule というファイルに指定されます。

- 同一のルールで複数の有効期限条件が指定できます (上記の例を参照)。
- ルールはメッセージストア全体に適用でき (グローバルルール)、パーティション、ユーザー、フォルダごとにも適用できます。グローバルルール以外は、store.expirerule ルールを使用してのみ作成できます。
  - グローバルルールは、configutil パラメータの store.expirerule.*rulename.attribute* を使用するか、*msg_svr_base*/config/store.expirerule にルールを指定して作成します
  - パーティションルールは、*store_root*/partition/*partition_name*/store.expirerule にルールを指定して作成できます。
  - ユーザールールは、*store_root*/partition/*partition_name*/*userid*/store.expirerule にルールを指定するか、folderpattern ルールを user/*userid*/.* となるように指定して作成できます
  - フォルダルールは、*store_root*/partition/*partition_name*/*userid/folder*/store.expirerule にルールを指定するか、folderpattern ルールを user/*userid*/*folder* となるように指定して作成できます

注 ユーザールールとフォルダルールも、folderpattern 属性を指定することによって、グローバル有効期限ファイル (*msg_svr_base*/config/store.expirerule) に置くことができます。

- 複数の有効期限ルールが同時に 1 つのメールボックスに適用できます。メールボックスに対する有効期限ポリシーは、グローバルルールとローカルルールで構成されます。ローカルルールは同一ディレクトリのメールボックスおよびそのサブフォルダのすべてに適用されます。
- imexpire によって、メールボックスに排他的なルールが指定されていないかぎり、そのメールボックスに適用されているすべての有効期限ルールが結合されます (表 18-8 を参照)。その結果、ルールセットには、すべての適用可能なルールの中からもっとも制約度の高い有効期限ポリシーが採用されます。たとえば、メッセージの最長存続期間がルール X によって 10 日間、ルール Y によって 5 日間と指定されている場合、結合結果は 5 日間となります。

表 18-8　imexpire 属性

属性	説明 (属性値)
exclusive	ルールが排他的であるかどうかを指定します。exclusive として指定すると、指定したメールボックスにこのルールのみが適用され、これ以外のルールはすべて無視されます。複数の排他的なルールが存在する場合、最後にロードされた排他的なルールが使用されます。たとえば、グローバルな排他的ルールおよびローカルな排他的ルールが指定された場合、ローカルルールが使用されます。グローバルな排他的ルールが複数存在する場合、configutil によって最後にリストされたグローバルルールが使用されます (yes/no)。
folderpattern	このルールによって影響を受けるフォルダを指定します。形式は user/ で始まる必要があり、これはディレクトリ *store_root*/partition/*/ を表します。614 ページの図 18-4 および 612 ページの表 18-9 を参照。(POSIX 正規表現)
messagecount	フォルダ内の最大メッセージ数。この数を超える新しいメッセージが配信されると、もっとも古いメッセージが消去されます (整数)。
foldersize	新しいメッセージが配信されたときにもっとも古いメッセージが消去される前のフォルダの最大サイズ (バイトを表す整数)。
messagedays	メッセージが消去されるまでの存続日数 (整数)。
messagesize	消去のマークが付けられる前のメッセージの最大サイズ (バイトを表す整数)。
messagesizedays	猶予期間。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 (整数)。
メッセージのヘッダーフィールド	メッセージに削除のマークを付けるためのヘッダーフィールドと文字列を指定します。値は大文字と小文字が区別されず、正規表現は認識されません。 例 : Rule1.Subject: Get Rich Now! *Expires* ヘッダーや *Expiry-Date* ヘッダーについては、これらのヘッダーフィールドで指定された日付の値が messagedays 属性よりも古い場合、imexpire によってそのメッセージは削除されます。有効期限に関するヘッダーフィールドが複数指定された場合、もっとも古い有効期限が使用されます (文字列)。
regexp	UNIX 正規表現をルール作成において有効にします。(1 または 0)。この属性を指定しないと、IMAP 表現が使用されます。
seen	seen はメッセージのステータスフラグの 1 つであり、ユーザーがメッセージを開いたときにシステムによって設定されます。seen 属性が and に設定されている場合、メッセージが開封済みであり、**かつ**、ほかの条件が満たされていればルールは適用されます。seen 属性が or に設定されている場合、メッセージが開封済みであるか、**または**、もう 1 つの条件が満たされていればルールは適用されます (and/or)。

表 18-8　imexpire 属性 (続き)

属性	説明 (属性値)
deleted	deleted はメッセージのステータスフラグの 1 つであり、ユーザーがメッセージを削除したときにシステムによって設定されます。属性 deleted が and に設定されている場合、メッセージが削除済みであり、**かつ**、もう 1 つの条件が満たされていればルールは適用されます。deleted 属性が or に設定されている場合、メッセージが開封済みであるか、**または**、もう 1 つの条件が満たされていればルールは適用されます **(and/or)**。

imexpire ルールをテキストモードで設定する

自動メッセージ削除ルールは、store.expirerule ファイルにルールを指定することによって設定します。store.expirerule ファイルは、1 行につき 1 つの有効期限条件を含みます。グローバルルール設定ファイル (*msg_svr_base*/data/store/store.expirerule) の有効期限条件は、次の形式になっています。

rule_name.*attribute*：*value*

コード例 18-1 に、*msg_svr_base*/config/store.expirerule の一連の有効期限ルールを示します。

Rule 1 では、グローバル有効期限ポリシー (すべてのメッセージに適用されるポリシー) を設定しています。設定内容は次のとおりです。

- UNIX 正規表現をルール作成において有効にします。
- 100,000 バイトよりも大きいメッセージを 3 日後に削除します。
- ユーザーによって削除済みとされたメッセージを削除します。
- 「Viagra Now!」または「XXX Porn!」という文字列が Subject: ヘッダーにあるメッセージをすべて削除します。
- すべてのフォルダのメッセージ数を 1,000 件までに制限します。1,000 件を超えた場合、フォルダ内でもっとも古いメッセージを削除して合計件数を 1,000 以内に維持します。
- 存続期間が 365 日よりも長いメッセージをすべて削除します。

Rule 2 では、ホストしているドメインが siroe.com のユーザーに対して自動メッセージ削除ポリシーを設定しています。メールボックスサイズを 1M バイトに制限し、削除済みメッセージを削除し、存続期間が 14 日より長いメッセージを削除します。

Rule 3 では、ユーザー f.dostoevski の inbox フォルダに対して自動メッセージ削除ポリシーを設定しています。「On-line Casino」という件名行のあるメッセージを削除します。

コード例 18-1 imexpire ルールの例

```
Rule1.regexp: 1
Rule1.folderpattern:user/.*
Rule1.messagesize: 100000
Rule1.messagesizedays: 3
Rule1.deleted:or
Rule1.Subject: Vigara Now!
Rule1.Subject: XXX Porn!
Rule1.messagecount: 1000
Rule1.messagedays: 365
Rule2.regexp: 1
Rule2.folderpattern: user/.*@siroe.com/.*
Rule2.exclusive:yes
Rule2.deleted:or
Rule2.messagedays: 14
Rule2.messagecount: 1000
Rule3.folderpattern: user/f.dostoevski/inbox
Rule3.Subject: *On-line Casino*
```

imexpire フォルダパターンを設定する

フォルダパターンは POSIX 正規表現を使用して指定できます。このためには、imexpire 属性の regex を 1 に設定します。この属性を指定しないと、IMAP 表現が使用されます。この形式は user/ で始まり、そのあとにパターンが続きます。表 18-9 に、各種フォルダのフォルダパターンを示します。

表 18-9 正規表現を使用した imexpire フォルダパターン

フォルダパターン	適用範囲
user/*userid*/.*	*userid* のすべてのフォルダ内にあるすべてのメッセージに規則を適用します。
user/*userid*/Sent	*userid* のフォルダ Sent: 内のメッセージに規則を適用します。
user/.*	メッセージストア全体に規則を適用します。
user/.*/trash	すべてのユーザーの trash フォルダに規則を適用します。
user/.*@siroe.com/.*	ホストされたドメイン siroe.com: 内のフォルダに規則を適用します。

表 18-9　正規表現を使用した imexpire フォルダパターン (続き)

フォルダパターン	適用範囲
user/[^@]*/.*	デフォルトドメイン内のフォルダに規則を適用します。

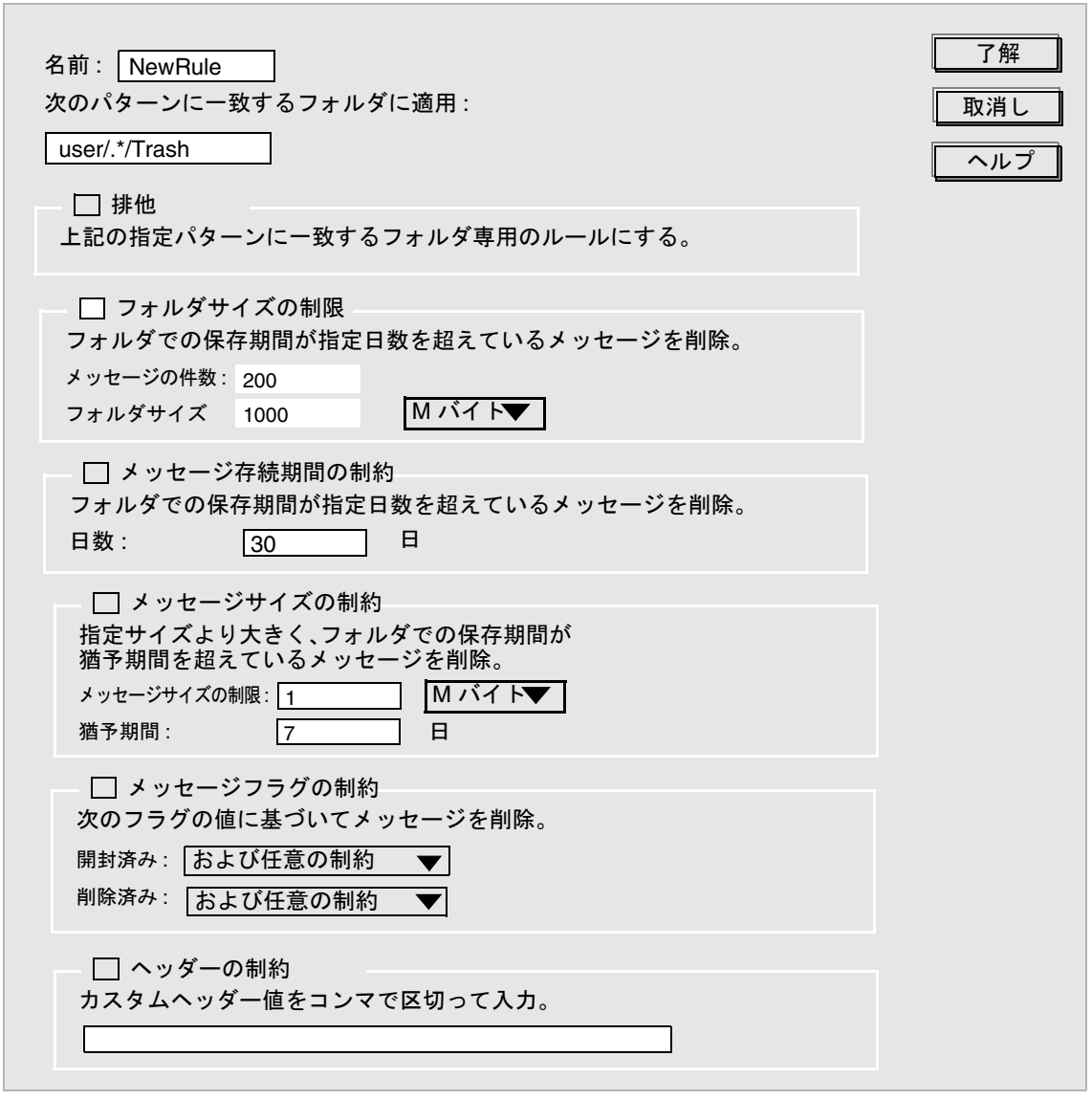
コンソールを使用して自動メッセージ削除ルールを設定するには

1. 次の操作で自動メッセージ削除の GUI を呼び出します。

   メインコンソール -> サーバーグループ -> Messaging Server (開く) -> Messaging Server コンソール -> 設定タブ -> メッセージストア -> 有効期限 / パージ -> 追加

   この GUI の略図を図 18-4 に示します。

図 18-4　自動メッセージ削除 (有効期限またはパージ) GUI - 略図



2. 新しいルールの名前を入力します。
3. メッセージを自動的に削除するフォルダを入力します。

前述の 612 ページの「imexpire フォルダパターンを設定する」を参照してください。

4. このルールが指定した条件と一致するフォルダに対する排他的なルールである場合は、「排他」ボックスをクリックします。

   このボックスにチェックマークを付けると、このルールが、指定したパターンに一致するほかのすべてのルールに優先します。「排他」チェックボックスの詳細については、610 ページの表 18-8 を参照してください。

5. フォルダサイズに基づいてルールを作成するには、以下を実行します。

   - 「フォルダサイズの制限」チェックボックスにチェックマークを入れます。「メッセージの件数」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、もっとも古いメッセージが削除されるまで保持されるフォルダの最大サイズをバイト単位で指定します。

6. メッセージの存続期間に基づいてルールを作成するには、「メッセージ存続期間の制約」チェックボックスにチェックマークを付けます。

   「日数」フィールドで、メッセージがフォルダに保持される期間を日数で指定します。

7. メッセージサイズに基づいてルールを作成するには、以下を実行します。

   - 「メッセージサイズの制約」チェックボックスにチェックマークを入れます。「メッセージサイズの制限」フィールドに、フォルダで許可されるメッセージの最大サイズを入力します。「猶予期間」フィールドに、サイズを超過したメッセージがフォルダ内に保持される (削除されるまでの) 期間を入力します。

8. 「開封済み」または「削除済み」メッセージフラグが設定されているかどうかに基づいてルールを作成するには、以下を実行します。

   - 「メッセージフラグの制約」チェックボックスにチェックマークを入れます。
   - 「開封済み :」フィールドでは、「および」を選択すると、メッセージが開封済みであり、**かつ**、もう 1 つの条件を満たしている場合にルールを適用することを指定できます。「または」を選択すると、メッセージが開封済みであるか、**または**、もう 1 つの条件を満たしている場合にルールを適用することを指定できます。
   - 「削除済み :」フィールドでは、「および」を選択すると、メッセージが削除済みであり、**かつ**、もう 1 つの条件を満たしている場合にルールを適用することを指定できます。「または」を選択すると、メッセージが削除済みであるか、**または**、もう 1 つの条件を満たしている場合にルールを適用することを指定できます。

9. ヘッダーフィールドとその値に基づいてルールを作成するには、以下を実行します。

   - 「ヘッダーの制約」チェックボックスにチェックマークを入れます。

- ヘッダーと値のリストを次の形式でコンマで区切って入力します。

  header1：*value1*, *header2*：*value2*

  例：`Subject: Work at Home!,From: virus@sesta.com`

Expires ヘッダーや *Expiry-Date* ヘッダーで、日付の値が「メッセージの存続期間の制約」よりも古い場合、メッセージは削除されます。複数の有効期限ヘッダーフィールドが指定されている場合、もっとも早い有効期限日が使用されます (文字列)。

10. 「了解」をクリックすると、新しいルールが自動メッセージ削除リストに追加されます。

自動メッセージ削除とログレベルをスケジュールするには

自動メッセージ削除は、`imsched` スケジューリングデーモンによってアクティブになります。デフォルトでは、`imsched` は毎日 23:00 に `imexpire` を呼び出し、メッセージは消去およびパージされます。このスケジュールは、`configutil` パラメータの `local.schedule.expire`、`local.schedule.purge`、および `store.cleanupage` を設定することによってカスタマイズできます。表 18-10 を参照してください。

有効期限およびパージは、大きなメッセージストアでは完了するまでに時間のかかることがあるので、これらのプロセスの実行頻度は実験して決定することをお勧めします。たとえば、有効期限およびパージの 1 サイクルに 10 時間かかる場合、有効期限およびパージのデフォルトスケジュールを 1 日に 1 回とするわけにはいきません。有効期限とパージをスケジュールするには、`local.schedule.purge` を使用してパージのスケジュールを個別に指定します。`local.schedule.purge` が設定されていない場合、`imexpire` は有効期限を実行したあとにパージを実行します。

表 18-10 有効期限およびパージ configutil ログおよびスケジュールパラメータ

パラメータ	説明

local.schedule.expire	imexpire を実行する間隔。UNIX の crontab の書式を使用します。
	分 時 日付 月 曜日
	
	値は空白文字またはタブ文字で区切られ、値の範囲は、分は 0 ～ 59、時は 0 ～ 23、日付は 1 ～ 31、月は 1 ～ 12、曜日は 0 ～ 6 (0 = 日曜日) となります。各時間フィールドには、アスタリスク (すべての取りうる値)、コンマ区切りの値のリスト、またはハイフンで区切られた 2 つの値による範囲を使用することもできます。日は、「日」と「曜日」の両方で指定できることに注意してください。ただし、このような発生回数は非常に少ないので、通常、両方で指定することはありません。日と曜日の両方で指定した場合、その両方が必須条件になります。たとえば、17 日と火曜日を設定すると、両方の値が真であることが求められます。
	
	実行間隔の例
	
	1) imexpire を 12:30am、8:30am、4:30pm に実行する場合
	`30 0,8,16 * * *  /opt/SUNWmsgsr/lib/imexpire`
	
	2) imexpire を平日の朝 3:15am に実行する場合
	`15 3 * * 1-5  /opt/SUNWmsgsr/lib/imexpire`
	
	3) imexpire を毎週月曜日だけ実行する場合
	`0 0 * * 1  /opt/SUNWmsgsr/lib/imexpire`
	
	デフォルト : `0 23 * * *  /opt/SUNWmsgsr/lib/imexpire`

local.schedule.purge	purge を実行する間隔。UNIX の crontab の書式を使用します。 分 時 日付 月 曜日。 デフォルト : 0 0,4,8,12,16,20 * * * /opt/SUNWmsgsr/lib/purge -num=5 (4 時間ごと)
store.cleanupage	purge で完全に削除するまでの、有効期限が切れた、または消去されたメッセージの存続期間 (時間数)。 デフォルト : なし
local.store.expire.loglevel	ログのレベルを指定します。 1 = expire セッション全体の要約をログに記録します。 2 = 有効期限が切れたメールボックスごとに 1 メッセージをログに記録します。 3 = 有効期限が切れたメッセージごとに 1 メッセージをログに記録します。 デフォルト : 1

コンソールを使用した場合の imexpire スケジュール

次の操作で自動メッセージ削除の GUI を呼び出します。

メインコンソール -> サーバーグループ -> Messaging Server (開く) -> Messaging Server コンソール -> 構成タブ -> メッセージストア -> 有効期限またはパージ

このコンソールページでは、有効期限ルールが上部に、有効期限およびパージスケジュールが下部に一覧表示されます。有効期限およびパージをスケジュールするには、「有効期限 / パージスケジュール」のプルダウンメニューを使用して、有効期限とパージの両方の月、日、曜日 (0 = 日曜日)、時、分を設定します。

注 日の値は、「日」と「曜日」の両方で設定できます。両方で設定した場合、両方が評価されます。水曜日と 17 日を設定した場合、パージおよび有効期限は、17 日が水曜日にあたった場合にのみ実行されます。

imexpire ログレベルを設定する

`imexpire` が完了すると、デフォルトのログファイルに要約が記録されます。有効期限をコマンド行から呼び出す場合は、-v (詳細) および -d (デバッグ) の各オプションを使用して、詳細ステータスまたはデバッグメッセージを stderr に記録するように `imexpire` に指示できます。`imsched` を使用して `imexpire` を呼び出す場合は、`configutil` パラメータの `local.store.expire.loglevel` を 1、2、または 3 に設定して各ログレベルを選択できます。Loglevel 1 はデフォルトで、有効期限セッション全体の要約が記録されます。Loglevel 2 では、有効期限切れのメールボックスごとに 1 つのメッセージが記録されます。Loglevel 3 では、有効期限切れのメッセージごとに 1 つのメッセージが記録されます。

メッセージの自動削除から指定されたユーザーを除外する

msg_svr_base`/config/` 内の `expire_exclude_list` ファイルで、1 行に 1 つずつユーザー ID を追加して、指定したユーザーを有効期限ルールから除外できます。

メッセージストアのパーティションを構成する

メッセージストアパーティションとは、ディスクパーティション上の、メッセージストアを格納するための専用エリアです。メッセージストアパーティションはディスクパーティションと同じではありませんが、管理の便宜をはかるために、各メッセージストアパーティション用に 1 つのディスクパーティションと 1 つのファイルシステムを使用することをお勧めします。メッセージストアパーティションは、メッセージストアとして特に指定されたディレクトリです。

デフォルトでは、ユーザーメールボックスは *store_root*/partition/ ディレクトリに保存されています (576 ページの図 18-1 を参照)。partition ディレクトリは、単一または複数の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。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を追加できます。たとえば、ユーザー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。

store_root/partition/mkting/
store_root/partition/eng/
store_root/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ー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、ユーザー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ー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。ユーザーアカウントの設定時やユーザー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、サーバーの実行に使用するユーザー 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ー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. 構成を行う Messaging Server をコンソールから開きます。
2. 「設定」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メッセージストア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。

注 デフォルトのパーティションとは、ユーザーが作成されたときに、ユーザーエントリに `mailMessageStore` LDAP 属性が指定されなかった場合に使用されるパーティションのことです。デフォルトのパーティションが必要にならないように、すべてのユーザーエントリに `mailMessageStore` LDAP 属性を指定する必要があります。

8. 「了解」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。

コマンド行 : コマンド行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。

- ユーザー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 (620 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、この方法がユーザー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、ユーザー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、ユーザーに通知を出して、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、ユーザー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 (『Sun Java System Communications Services Schema Reference Manual』を参照)。

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、ユーザー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. ユーザー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*>

   次に例を示します。

   mboxutil -r user/ofanning/INBOX user/ofanning/INBOX secondary

3. 移動したユーザーの 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. ユーザー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。

デフォルトのメッセージストアパーティション定義の変更

デフォルトのパーティションとは、ユーザーが作成されたときに、ユーザーエントリに mailMessageStore LDAP 属性が指定されなかった場合に使用されるパーティションのことです。デフォルトのパーティションが必要にならないように、すべてのユーザーエントリにユーザーのメッセージストアパーティションを指定する mailMessageStore LDAP 属性を指定する必要があります。さらに、デフォルトのパーティションを負荷分散やその他の理由で変更しては**なりません**。デフォルトのパーティションの定義に依存するユーザーが存在する間に、デフォルトのパーティションを変更するのは無効であり危険です。

デフォルトのパーティションをどうしても変更する必要がある場合は、configutil パラメータの store.defaultpartition でデフォルトの定義を変更する前に、古いデフォルトのパーティションのすべてのユーザー (そのままになっていたユーザー) の mailMessageStore 属性が現在のパーティション (これはデフォルトでなくなる) に設定されているようにしてください。

メッセージストアの保守手順を実行する

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。サーバー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、サーバーの実行状況に関する情報を記録したログファイルを監視する必要もあります。ログファイルについては、第 21 章「ログの管理」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

この節では、メールボックスの管理および監視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

メールボックスの一般的な保守タスクを実行する場合は、mboxutil コマンドを使用します。mboxutil で実行できるタスクは次のとおりです。

- メールボックスの一覧表示
- 孤立メールボックスおよび非アクティブメールボックスの一覧表示と削除
- メールボックスの作成
- 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
- 孤立したメールボックスおよび非アクティブのメールボックスの削除
- 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、630 ページの「制限容量を監視するには」を参照してください。

注 mboxutil プロセスを実行途中で強制終了しないでください。SIGKILL (kill -9) で強制終了すると、各サーバーを再起動し、回復処理を行わなければならないことがあります。

表 18-11 に mboxutil コマンドの一覧を示します。構文や使用要件の詳細については、『Messaging Server Reference Manual』を参照してください。

表 18-11　mboxutil オプション

オプション	説明
-a	廃止。すべてのユーザーの制限容量に関する情報の表示に使用します。imquotacheck を使用します。
-c *mailbox*	指定したメールボックスを作成します。-f とともに使用できます。 メールボックスが 1 つ存在していないと、次のメールボックスを作成できません。
-d *mailbox*	指定したメールボックスを削除します。 ユーザーをメッセージストアから削除するには、-d *mailbox* に次の値を使用します。 user/*userid*/INBOX たとえば、ユーザー john をメッセージストアから削除するには、-d user/john/INBOX を使用します。ユーザー john のメールボックスの mm フォルダを削除するには、-d user/john/mm を使用します。 Delegated Administrator ユーティリティの commadmin user delete コマンドまたは Delegated Administrator コンソールを使用して LDAP ディレクトリでユーザーステータスを削除済みとマークすることによってユーザーを削除する方法を推奨します。次に、commadmin user purge コマンドを使用して、指定された日数よりも長い期間、削除済みとしてマークされていたユーザーをパージします。 前の段落の説明のように Delegated Administrator ユーティリティを使用した場合は、メールボックスを削除するために mboxutil -d コマンドを使用する必要はありません。
-e	メッセージストア内のすべての削除済みのメッセージを消去します。*pattern* に一致するすべての削除済みメールボックスを消去するために、このオプションを -p *pattern* オプションと使用することもできます。
-f *file*	メールボックス名を保存するファイルを指定します。-f オプションを -c 、-d または -r オプションと使用できます。 ファイルには、mboxutil コマンドの実行対象になるメールボックスのリストが含まれます。次にデータファイルのエントリの例を示します。 user/daphne/INBOX user/daphne/projx user/daphne/mm
-k *mailbox cmd*	廃止。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。

表 18-11　mboxutil オプション (続き)

オプション	説明
-l	サーバーのすべてのメールボックスを一覧表示します。 異なる言語ロケール用にマルチバイトフォルダを作成する場合は、*msg_svr_base*/bin/msg/bundles/encbylang.properties を編集して、適切な文字セットを LANG 環境変数に関連付けます。
-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 ホスト内の Inbox で、対応するエントリが LDAP にないものを検索します。たとえば、-o オプションは、所有者が LDAP から削除された、または別のサーバー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、mboxutil 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 -w が指定された場合は、書き込みません
-p *MUTF7_IMAP_pattern*	-l オプションとともに使用した場合、名前が *MUTF7_IMAP_pattern* と一致するメールボックスのみが一覧表示されます。 名前が *MUTF7_IMAP_pattern* に一致するメールボックスを削除または消去するために、このオプションを -d または -e オプションとともに使用することもできます。 IMAP ワイルドカードを使用できます。このオプションは、IMAP M-UTF-7 形式のパターンを受け入れます。ascii でないメールボックスの検索にはこの方法を推奨しません。ascii でないメールボックスの検索には、-P オプションを使用します。
-P *regexp*	指定された POSIX 正規表現に一致する名前のメールボックスのみが一覧表示されます。このオプションはローカル言語の *regexp* を受け入れます
-q *domain*	廃止。imquotacheck -d *domain* を使用します
-r *oldname newname* [*partition*]	メールボックスの名前を *oldname* (現在の名前) から *newname* (新規の名前) 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。ファイルを使用する場合は、-f フラグとともに使用できます。 このオプションを使用してユーザー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は、この操作は失敗します。

表 18-11　mboxutil オプション (続き)

オプション	説明
-R *mailbox*	まだパージされていない削除済みメッセージを復元します。 メールボックスが消去または有効期限切れになると、削除済みメッセージの uid が store.exp ファイルに保存されます。cleanupage を過ぎると、メッセージは imexpire によって物理的に削除されます。expunge または expire を誤って発行した場合、このオプションを使用して、imexpire でパージされていない削除済みメッセージを元のメールボックスに復元できます。
-s	-l オプションとともに使用すると、メールボックス名のみを表示します。その他のデータは表示されません。
-t *num*	指定された日数 (*num*) 内にアクセスされていないメールボックスを一覧表示します。-t オプションは、孤立メールボックスを識別する -o オプションとともに使用する必要があります。 したがって、-t オプションは、非アクティブメールボックス (前回アクセスした日付に基づいて) を孤立メールボックス (LDAP ディレクトリに対応するユーザーエントリがないメールボックス) とともに識別します。 孤立メールボックスおよび非アクティブメールボックスを識別 (一覧表示) するには、mboxutil -o -w *file* -t *num* を使用します。 それらの孤立メールボックスおよび非アクティブメールボックスを削除のためにマークするには、mboxutil -d -f *file* を使用します。この *file* は、前の -w *file* で使用したファイルと同じファイルにします。 この機能を使用するには、config 変数の local.enablelastaccess を少なくとも -t オプションで指定した日数を有効にする必要があります。
-u *user*	廃止。すべてのユーザー情報の表示に使用します。imquotacheck -u *user* を使用します
-w *file*	-o オプションとともに使用します。-o オプション (孤立アカウントを識別する) によって生成されたメールボックス名をファイルに書き込みます。
-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。

注　mboxutil コマンドで POSIX 正規表現を使用できます。

メールボックスのネーミングルール

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ー、*mailbox* はメールボックスの名前を表します。ホストしているドメインでは、*userid* は *uid@domain* です。

たとえば次のコマンドでは、ユーザー ID が crowe であるユーザーの、INBOX という名前のメールボックスが作成されます。INBOX は、ユーザー 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、各ユーザー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名です。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

全ユーザー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
```

ユーザー 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
```

ユーザー 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
```

ユーザー 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、次のように入力します。

```
mboxutil -d user/druscilla/INBOX
```

ユーザー 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
```

ユーザー 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。

ユーザー 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*

孤立アカウントを削除するには

孤立アカウント (対応するエントリが LDAP にないメールボックス) を検索するには、次のコマンドを使用します。

mboxutil -o

コマンド出力が以下のように続きます。

```
mboxutil: Start checking for orphaned mailboxes
user/annie/INBOX
user/oliver/INBOX
mboxutil: Found 2 orphaned mailbox(es)
mboxutil: Done checking for orphaned mailboxes
```

次のコマンドで作成したファイルは、孤立メールボックスを削除するスクリプトファイルにすることができます。この例では、ファイル名は orphans.cmd です。

mboxutil -o -w orphans.cmd

コマンド出力は次のとおりです。

```
mboxutil: Start checking for orphaned mailboxes
mboxutil: Found 2 orphaned mailbox(es)
mboxutil: Done checking for orphaned mailboxes
```

次のコマンドを使用して孤立したファイルを削除します。

mboxutil -d -f orphans.cmd

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。したがって、特定のユーザー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 (d1/a7/ など) をレポートします。このパスは、ユーザー 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、ユーザー 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、何人のユーザーが共有 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ーに与えることができます。ほかのユーザー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ー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ー数とメールボックスの名前を表示させます。ユーザー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ーとは、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ーのことです。自分の個人用メールボックスを読んだユーザーは、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報告されません。

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ーをすべてカウントします。

```
readership -d 15
```

制限容量を監視するには

imquotacheck ユーティリティを使って、制限容量の使用状況と制限を監視します。このユーティリティは、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、K バイトでレポートされます。また、このユーティリティでメールボックスのサイズとユーザーに制限容量を比較することもできます。オプションとして、制限容量に対し一定の割合を超えたユーザーに対し、電子メールによる通知を送信することができます。

注 imquotacheck のいくつかの機能が変更されました (Messaging Server 6.x では、quotacheck ユーティリティが imquotacheck ユーティリティに替わりまし)。Messaging Server 5.x では、ユーザーのリストを取得するために quotacheck ユーティリティを使用すると、quotacheck はローカル mboxlist データベースを検索しました。この機能は、mboxutil ユーティリティの検索機能と重複します。

Messaging Server 6.x では、この重複機能が imquotacheck ユーティリティから削除されました。imquotacheck でユーザーの検索を実行する場合、検索はローカルの mboxlist データベースではなく、LDAP ディレクトリに対して行われます。ローカル mboxlist データベースからユーザーのリストを取得するには、mboxutil ユーティリティを使用します。

制限容量がルールファイルの最小しきい値を超えるすべてのユーザーの使用状況を一覧表示するには、次のように入力します。

```
imquotacheck
```

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。

```
imquotacheck -d siroe.com
```

デフォルトのルールファイルにしたがって、すべてのユーザーに通知を送信するには、次のように入力します。

```
imquotacheck -n
```

指定したルールファイル *myrulefile* と指定したメールテンプレートファイル *mytemplate.file* にしたがって、すべてのユーザーに通知を送信するには、次のように入力します (詳細については、『Sun Java System Messaging Server Administration Reference』を参照)。

```
imquotacheck -n -r myrulefile -t mytemplate.file
```

ルールファイルを無視して、すべてのユーザーの使用状況を一覧表示するには、次のように入力します。

```
imquotacheck -i
```

user1 のフォルダ使用状況別に一覧表示するには (ルールファイルを無視)、次のように入力します。

imquotacheck -u user1 -e

ディスク容量を監視するには

システムがディスク容量やパーティションの使用状況を監視する頻度と、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。詳細は、838 ページの「ディスク容量を監視する」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ー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。
- サーバーの状態、ディスク容量、サービスへの応答時間などの定期的監視 (850 ページの「stored」を参照)。
- 必要に応じた警告の生成。
- 必要に応じたデータベース回復 (658 ページの「メッセージストアの起動と回復」を参照)。

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 18-12 に stored オプションの一部を示します。一般的な使用例についてはこの表に従ってください。構文や使用要件の詳細については、『Messaging Server Reference Manual』を参照してください。

表 18-12　stored オプション

オプション	説明
-d	廃止。stored を起動するには、start-msg store を使用します。start-msg store は、デーモンとして実行され、システムチェックを実行し、アラーム、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。
-t	stored のステータスをチェックします。このコマンドのリターンコードが状態を表します。
-v	詳細に出力します。
-v -v	さらに詳細に出力します。

状態を出力するには、次のように入力します。

```
stored -t -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、次のように 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ります (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)。UNIX 上で stored を再起動するには、コマンド行で次のコマンドを使用します。

msg_svr_base/sbin/stop-msg store
msg_svr_base/sbin/start-msg store

サーバー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。

同一メッセージのストレージが重複するためメッセージストアのサイズを小さくする

1 つのメッセージが複数の受取人に送信されると、そのメッセージは各受取人のメールボックスに格納されます。一部のメッセージングシステムでは、同じメッセージのそれぞれのコピーを各受取人のメールボックスに格納します。それに対して、Sun Java System Messaging Server では、そのメッセージが格納されているメールボックスの数に関係なく、メッセージのコピーを 1 つだけ保持するよう努めます。このために、そのメッセージが含まれているメールボックス内にそのメッセージへのハードリンクを作成します。

ほかのメッセージングシステムが Sun Java Messaging Server に移行されると、移行プロセスによってメッセージの複数のメッセージコピーが作成されることがあります。メッセージストアが大きい場合、大量のメッセージが不必要に重複することになります。また、IMAP `append` 操作やほかのソースからなど、通常のサーバー操作で同じメッセージの複数のコピーが蓄積される可能性もあります。

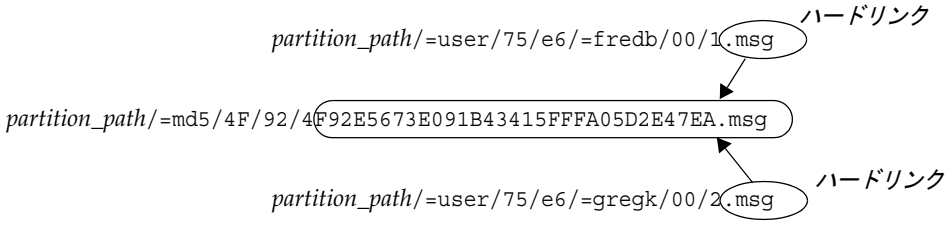
Messaging Server には、余分のメッセージコピーを削除し、それらを 1 つのコピーへのハードリンクに置き換える `relinker` という新しいコマンドが用意されています。

Relinker の動作方式

再リンク機能は、コマンド行モードでもリアルタイムモードでも実行できます。`relinker` コマンドを実行すると、メッセージストアのパーティション全体がスキャンされ、MD5 メッセージダイジェストリポジトリがハードリンクとして作成または更新され、余分のメッセージファイルが削除されて、必要なハードリンクが作成されます。

ダイジェストリポジトリは、メッセージストアに格納されているメッセージへのハードリンクから構成されます。このリポジトリは、ディレクトリ階層 *partition_path*`/=md5` に格納されます。このディレクトリは、ユーザーメールボックスの階層 *partition_path*`/=user` に対応しています (576 ページの図 18-1 を参照)。ダイジェストリポジトリのメッセージは、その MD5 ダイジェストによって一意に識別されます。たとえば、`fredb/00/1.msg` のダイジェストが `4F92E5673E091B43415FFFA05D2E47` である場合、*partition*`/=user/`*hashdir*`/`*hashdir*`/=fredb/00/1.msg` は *partition*`/=md5/`*hashdir*`/`*hashdir*`/4F92E5673E091B43415FFFA05D2E47EA.msg` にリンクされます。別のメールボックスに同じメッセージ (*partition_path*`/=user/`*hashdir*`/`*hashdir*`/gregk/00/17.msg` など) があると、そのメッセージもまた *partition_path*`/=md5/4F/92/4F92E5673E091B43415FFFA05D2E47EA.msg` にハードリンクされます。634 ページの図 18-5 はこのことを示しています。

図 18-5　メッセージストアのダイジェストリポジトリ



このメッセージの場合、リンクカウントは 3 になります。両方のメッセージが fredb および gregk というメールボックスから削除されると、リンクカウントは 1 になり、メッセージをパージできます。

relinker プロセスは、リアルタイムモードで実行しても同じように機能します。詳細は、635 ページの「リアルタイムモードで relinker を使用する」を参照してください。

コマンド行モードで relinker を使用する

relinker は、メッセージストアのパーティション全体をスキャンし、MD5 メッセージリポジトリをハードリンクとして作成または更新して、余分のメッセージファイルを削除します。relinker がストアパーティションをスキャンし終わると、再リンク前後の一意のメッセージ数とパーティションのサイズに関する統計情報が出力されます。すでにハッシュされているストアを迅速に処理するために、relinker は =md5 にまだ存在しないメッセージのダイジェストを計算します。また、ダイジェストリポジトリ全体を消去することもできます (ユーザーメールボックスに影響しない場合)。

コマンドの構文は次のとおりです。

relinker [-p *partitionname*] [-d]

ここで、*partitionname* は処理されるパーティション (デフォルト : すべてのパーティション) を示し、-d はダイジェストリポジトリが削除されること示します。出力例を次に示します。

```
# relinker

Processing partition:primary
Scanning digest repository...
Processing user directories.............................
-----------------------------------------------------------
Partition statistics                Before        After
-----------------------------------------------------------
Total messages                     4531898      4531898
```

```
Unique messages                        4327531      3847029
Message digests in repository                0      3847029
Space used                                99210Mb      90481Mb
Space savings from single-copy             3911Mb      12640Mb
-------------------------------------------------------------


# relinker -d

Processing partition:primary
Purging digest repository...
-------------------------------------------------------------
Partition statistics                    Before        After
-------------------------------------------------------------
Message digests in repository      3847029             0
-------------------------------------------------------------
```

relinker は、特にリポジトリにメッセージがまったく存在しない場合の最初の実行は、時間がかかることがあります。これは、すべてのメッセージのダイジェストを計算しなければならないためです (relinker 条件がすべてのメッセージを含めるように設定されている場合)。relinker 条件の設定については、636 ページの「relinker を設定する」を参照してください。たとえば、100G バイトのメッセージストアを処理するのに 6 時間を要するとします。ただし、実行時再リンクが有効になっている (635 ページの「リアルタイムモードで relinker を使用する」を参照) 場合は、relinker コマンドを実行する必要はありません。

relinker コマンド行モードが排他的に使用され、実行時オプションが有効でない場合は、ダイジェストリポジトリ (=md5) をパージする必要があります。そうしないと、ストア (=user) 内にパージされたメッセージがダイジェストリポジトリ内にリンクを持ち続ける (孤立する) ため、それらのディスク領域が解放されません。移行後などにストアの最適化を 1 回だけ行なっている場合は、relinker を一度実行してから、relinker -d でリポジトリ全体を削除できます。移行中にパージを何度も行なった場合は、実行するたびに期限切れや孤立したメッセージがリポジトリからパージされるので、relinker コマンドを繰り返し実行するだけで十分です。

異なるパーティションの各処理と並行して、relinker の複数のインスタンスを実行しておくと安全です (-p オプションを使用)。メッセージは同じパーティション内でのみ再リンクされます。

リアルタイムモードで relinker を使用する

リアルタイムモードで relinker 機能を有効にするには、configutil パラメータの local.store.relinker.enabled を yes に設定します。リアルタイムモードで relinker を使用すると、設定された relinker 条件 (636 ページの「relinker を設定する」を参照) に一致する、配信された (または復元された、IMAP によって追加された、など) すべ

てメッセージのダイジェストが計算され、そのダイジェストがリポジトリにすでに存在するかどうかが確認されます。ダイジェストが存在する場合は、メッセージの新しいコピーを作成する代わりに、そのダイジェストへのリンクが宛先メールボックスに作成されます。ダイジェストが存在しない場合は、メッセージが作成され、あとでそのメッセージへのリンクがリポジトリに追加されます。

stored は、各パーティションのダイジェストリポジトリをスキャンし、リンクカウントが 1 か、relinker 条件に一致しないメッセージをパージします。スキャンは、設定可能な期間内に一度に 1 つのディレクトリで実行されます。これは、入出力負荷が均等に分散され、ほかのサーバー操作に著しい影響を及ぼさないようにするためです。デフォルトでは、パージサイクルは 24 時間です。これは、メッセージがストアから削除されるか、設定可能な最長保存期間を過ぎたあとも、24 時間まは存在するということを意味します。このタスクは、relinker のリアルタイムモードが有効になっている場合にのみ使用できます。

relinker を設定する

表 18-13 に、relinker 条件を設定するためのパラメータを示します。

表 18-13 relinker の configutil パラメータ

パラメータ	説明
local.store.relinker.enabled	append コードと stored パージでのリアルタイムによるメッセージの再リンクを有効にします。このオプションがオフになっている場合でも relinker コマンド行ツールを実行できますが、stored によってリポジトリがパージされないため、relinker -d でこのタスクを実行する必要があります。このオプションをオンにすると、ディスク容量の節約と引き換えにメッセージ配信のパフォーマンスが低下します。 デフォルト : no
local.store.relinker.maxage	メッセージをリポジトリに保存しておく最長保存期間、または relinker コマンド行によって考慮されるメッセージの最長保存期間 (時間数)。-1 は保存期間に制限がない、つまり孤立したメッセージのみがリポジトリからパージされることを意味します。relinker の場合は、保存期間に関係なく既存のメッセージが処理されることを意味します。この値を小さくすると、リポジトリが小さい状態で保たれるため、relinker または stored によるパージの実行速度が上がり、ディスク領域を速く回復できます。この値を大きくすると、長期間にわたって重複するメッセージの再リンクを実行できます (ユーザーが数日違いで同じメッセージをストアにコピーしたり、数日間または数週間にわたって移行を行なったりする場合など)。 デフォルト : 24

表 18-13 relinker の configutil パラメータ (続き)

パラメータ	説明
local.store.relinker.minsize	実行時またはコマンド行の relinker によって考慮されるメッセージの最小サイズ (K バイト)。ゼロ以外の値に設定すると、小さいリポジトリと引き換えに小さいメッセージがもたらす relinker のさまざまなメリットが受けられません。 デフォルト : 0
local.store.relinker.purgecycle	stored によるパージサイクル全体のおおよその期間 (時間数)。実際の期間は、リポジトリ内の各ディレクトリのスキャンにかかる時間によって異なります。この値が小さいと、入出力操作が増し、この値が大きいと、ディスク領域の回復が遅くなります。0 に設定すると、ディレクトリ間で途切れることなく、連続的にパージが実行されます。-1 に設定すると、stored によるパージが実行されません (relinker -d コマンドを使用してパージを実行する必要がある)。 デフォルト : 24

メッセージストアのバックアップと復元を行う

メッセージストアのバックアップと復元は、もっとも一般的で重要な管理タスクです。メッセージストアのすべてのメッセージとフォルダのバックアップを行います。メッセージストアにバックアップと復元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。

- システムのクラッシュ
- ハードウェア障害
- 過失によるメッセージまたはメールボックスの削除
- 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)
- ユーザーの移行

コマンド行ユーティリティの imsbackup と imsrestore を使用するか、Legato Networker® が採用された統合ソリューションを使用してメッセージストアのバックアップと復元を行うことができます。

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ー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 つ目のメッセージコピーは、1 つ目のメッセージファイルの名前へのリンクとしてバックアップされます。以降も同様です。imsbackup 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保守します。ただし、この方法を採用する場合はデータの復元時に注意が必要です。詳細は、643 ページの「部分的な復元に関する考察」を参照してください。

注 メッセージストアのバックアップと復元は、すべてのメッセージファイルとディレクトリをバックアップする方法でも行うことができます。652 ページの「メッセージストアの災害時のバックアップと復元」を参照してください。

この節では、以下の項目に分けて説明しています。

メールボックス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。システムのバックアップによってピーク時のシステム負荷が減少することがあるからです。たとえば、バックアップは午前 2 時など早朝 (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 (642 ページの「増分バックアップ」を参照) 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。増分バックアップを毎日の保守手順の中で実行し、フルバックアップを週に 1 度実行することをお勧めします。

同時バックアップと順次バックアップ

ユーザー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ー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、たとえばサーバー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

バックアップグループは、正規表現で定義されたユーザーメールボックスの任意の集まりです。ユーザーメールボックスをバックアップグループに組織化することで、より柔軟なバックアップ管理を定義することができます。

たとえば、3 つのバックアップグループを作成し、第 1 のグループには A ～ L で始まるユーザー ID を、第 2 のグループには M ～ Z で始まるユーザー ID のユーザーを、第 3 のグループには ID が数字で始まるユーザーを含めます。管理者はこれらのバックアップグループを使用してメールボックスを同時にバックアップできます。または、ある日に一部のグループのみバックアップし、別の日にほかのグループを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について注意すべき事項がいくつかあります。

1. バックアップグループはメールユーザーの任意の「仮想」グループです。これらは見かけとは異なり、メッセージストアディレクトリに正確にはマッピングされません (576 ページの図 18-1)。

2. バックアップグループは、UNIX 正規表現を使用して管理者によって定義されます。
3. 正規表現は、次の設定ファイルで定義されています。
   msg_svr_base/config/backup-groups.conf
4. バックアップグループが imsbackup および imsrestore で参照された場合、次のパス形式が使用されます。*partition_name/backup_group*。

backup-groups.conf の形式は次のとおりです。

```
group_name=definition
group_name=definition
.
.
.
```

上記の例を使用して、次の 3 つの定義によるバックアップグループを作成するとします。

```
groupA=[a-l].*
groupB=[m,-z].*
groupC=[0-9].*
```

これで imsbackup および imsrestore をいくつかのレベルでスコープすることができます。次のバックアップコマンドを使用してメッセージストア全体をバックアップおよび復元することができます。

imsbackup -f *device* /

groupA の全ユーザー全メールボックスをバックアップするには、次のコマンドを使用します。

imsbackup -f *device* /*partition*/groupA

デフォルトのパーティションは primary です。

事前定義のバックアップグループ

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 user という名前のグループで、ここにはすべてのユーザーが含まれています。たとえば、次のコマンドで primary パーティションのすべてのユーザーがバックアップされます。

```
imsbackup -f backupfile /primary/user
```

Messaging Server のバックアップと復元のユーティリティ

データのバックアップと復元のために、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。ただし、imsbackup および imsrestore 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、これら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するサポートは非常に限定されています。また、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ません。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については、646 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。

次の例では、メッセージストア全体を /dev/rmt/0 にバックアップします。

```
imsbackup -f /dev/rmt/0 /
```

次の例では、ユーザー ID joe のメールボックスが /dev/rmt/0 にバックアップされます。

```
imsbackup -f /dev/rmt/0 /primary/user/joe
```

次の例では、バックアップグループ groupA に定義された全ユーザーの全メールボックスが backupfile にバックアップされます (639 ページの「バックアップグループを作成するには」を参照)。

```
imsbackup -f- /primary/groupA > backupfile
```

増分バックアップ

次の例は、2004 年 5 月 1 日 の午後 1 時 10 分から現在までに保存されたメッセージをバックアップします。デフォルトでは、日付に無関係にすべてのメッセージをバックアップします。

```
imsbackup -d 20040501:13100
```

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 コマンドの完全な構文に関する説明は、『Messaging Server Reference Manual』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復元するには、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 のメッセージを復元します。

```
imsrestore -f backupfile /primary/user1
```

imsbackup コマンドの完全な構文に関する説明は、『Messaging Server Reference Manual』を参照してください。

バックアップ実行時の多数宛メールの除外

バックアップ操作の実行時には、バックアップから除外するメールボックスを指定できます。重要でないメッセージが多数発生する多数宛またはごみのメールボックスを除外して、バックアップセッションを合理化し、操作完了までの時間を短縮し、バックアップデータの格納に必要なディスク容量を最小限にできます。

メールボックスを除外するには、configutil パラメータの local.store.backup.exclude の値を指定します。

1 つのメールボックス、または「%」文字で区切ったメールボックスのリストを指定できます。「%」文字は、メールボックス名には使用できません。たとえば、次の値を指定できます。

```
Trash
```

```
Trash%Bulk Mail%Third Class Mail
```

最初の例では、フォルダ Trash が除外されます。2 番目の例では、フォルダ Trash、Bulk Mail、および Third Class Mail が除外されます。

バックアップユーティリティは、local.store.backup.exclude パラメータに指定されたフォルダ以外のユーザーメールボックスのすべてのフォルダをバックアップします。

この機能は、Messaging Server バックアップユーティリティ、Legato Networker、およびサードパーティ製のバックアップソフトウェアと使用できます。

local.store.backup.exclude の設定を無効にし、バックアップ操作時に完全な論理名を指定して除外したメールボックスをバックアップできます。ごみ箱フォルダが除外されたとします。たとえば、次のように指定することにより、引き続きごみ箱をバックアップできます。

```
/primary/user/user1/trash
```

ただし、次のように指定すると、ごみ箱フォルダは除外されます。

```
/primary/user/user1
```

部分的な復元に関する考察

部分的な復元は、メッセージストアの一部を復元するときにのみ行われます。完全な復元は、メッセージストア全体を復元するときに行われます。メッセージストアでは単一コピーによるメッセージシステムが使用されています。つまり、メッセージの 1 つのコピーのみが 1 つのファイルとしてストアに保存されます。コピーされたメッセージのほかのインスタンス (メッセージが複数のメールボックスに送信される場合など) は、コピーへのリンクとして保存されます。このため、メッセージを復元する場合には注意が必要です。次に例を示します。

- **完全な復元** : 完全な復元では、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。
- **部分的なバックアップおよび復元** : 部分的なバックアップおよび部分的な復元では、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。

次の例では、部分的な復元が実行された場合に、複数のユーザーによって使用されるメッセージに発生する事柄を示します。以下のように、3 人のユーザー A、B、C に属する、まったく同じ 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、システムは部分的なバックアップと完全な復元を以下のように実行します。

1. ユーザー B および C のメールボックスをバックアップします。
2. ユーザー B および C のメールボックスを削除します。
3. 手順 1 のバックアップデータを復元します。

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 件だけ復元されます。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的な復元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. ユーザー A のメールボックスを削除します。
3. ユーザー A のメールボックスを復元します。

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、複数回の部分的な復元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. ユーザー A と B のメールボックスを削除します。
3. ユーザー B のメールボックスを復元します。

   復元ユーティリティが、最初に A/INBOX を復元するよう管理者に要求します。
4. ユーザー A と B のメールボックスを復元します。
5. ユーザー A のメールボックスを削除します (任意)。

注

すべてのメッセージを部分的な復元で復元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。

ドライブやテープなど、バックアップデバイスが検索可能である場合、imsrestore は A/INBOX/1 が格納されている位置を検索し、B/INBOX/1 として復元します。UNIX パイプなど、バックアップデバイスが検索不能である場合、imsrestore はオブジェクト ID とリンクされているオブジェクトの ID をファイルに記録します。管理者は -r オプションを使用して imsrestore を再び呼び出し、欠落しているメッセージ参照を復元する必要があります。

増分バックアップされたメールボックスからのメッセージを復元するには

増分バックアップされたメールボックスからメッセージを復元する際に、そのメールボックスがメッセージを復元するサーバーに存在する場合は、imesrestore を実行するだけでメッセージを復元できます。ただし、増分バックアップされたメールボックスからメッセージを復元する際に、そのメールボックスがすでに存在しない場合は、別の復元手順に従う必要があります。

メッセージストアサーバーに存在しないメールボックスを復元するには、次の手順のいずれかを使用します。

- 復元操作時に、ユーザーへのメッセージの配信を停止します。このためには、LDAP 属性の mailDeliveryOption に hold を設定します。
- mboxutil -c コマンドでメールボックスを作成してから、imesrestore を使用します。

増分バックアップを復元するためにこれらの手順に従う必要がある理由は、次のとおりです。メールボックスが削除または移行された場合、imsrestore ユーティリティはバックアップアーカイブに保存されたメールボックスの一意の ID 有効期間およびメッセージの一意の ID (UID) を使用してメールボックスを再作成します。

以前は、imsrestore は削除または移行されたメールボックスを再作成するときに、新しい UID 有効期間をメールボックスに、新しい UID をメッセージに割り当てました。その場合、キャッシュにメッセージが入っているクライアントはメールボックスの UID 有効期間およびメッセージの UID の同期をとりなおす必要がありました。クライアントは、新しいデータを再びダウンロードする必要があり、その結果、サーバーの作業負荷が増大しました。

新しい imsrestore の処理では、クライアントのキャッシュの同期は維持され、復元処理は透過的に実行され、パフォーマンスに悪影響を及ぼすことはありません。

メールボックスが存在する場合、imsrestore は新しい UID を復元されたメッセージに割り当てるので、新しい UID は既存のメッセージにすでに割り当てられている UID と矛盾しません。UID の一貫性を保証するために、復元操作時に imsrestore でメールボックスをロックします。ただし、imsrestore は、新しい UID 値を割り当てる代わりに、バックアップアーカイブからのメールボックスの UID 有効期間とメッセージ UID を使用するようになったため、増分バックアップおよび復元を実行すると UID の一貫性が保てなくなる場合があります。

imsbackup ユーティリティの -d 日付オプションを使用して増分バックアップを実行する場合、復元操作を完了するために imsrestore を複数回呼び出す必要がある場合があります。増分バックアップを実行する場合、最後のフルバックアップとその後のすべての増分バックアップを復元する必要があります。

復元操作間に新しいメッセージをメールボックスに配信できますが、この場合、メッセージの UID に矛盾が生じることがあります。UID の矛盾が生じないようにするには、前述の手順のいずれかを実行する必要があります。

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復元を行います。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 `msg_srv_base/lib/msg/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/bin/nsr`

   これは、古いバージョンの Networker (5.x) を使用している場合にのみ必須です。Networker 6.0 以上では、`nsrfile` は自動的に `/usr/bin/nsr` の下にインストールされます。
3. ユーザー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. 639 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。

mkbackupdir.sh によって作成されたディレクトリ構造を確認してください。その構造は、表 18-4 に示されているようになっている必要があります。

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ーに対して、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。表 18-4 に示した例を参照してください。

注

Legato Networker の旧バージョンでは、保存設定の名前には最高 64 文字まで使用できます。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 /primary/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 : inetuser)。

図 18-6 には、バックアップグループのディレクトリ構造の例が示されています。

図 18-6 バックアップグループのディレクトリ構造

```
/backup/primary/groupA/amy
                       /bob
                       /carly
                /groupB/mary
                       /nancy
                       /zelda
                /groupC/123go
                       /1bill
                       /354hut
```

次の例に、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。

```
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、/backup を使用します。

      同時バックアップには、/backup/*server*/*group* を使用します。

      639 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。

649 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/primary/groupA
   /backup/secondary/groupB
   /backup/tertiary/groupC
          .
          .
Backup Command:savepnpc
Parallelism: 4
```

Legato Networker を使用したデータの復元

データの回復は、Legato Networker の nwrecover インタフェースまたは recover コマンド行ユーティリティを使用して実行できます。以下の例では、ユーザー 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

Messaging Server では、コマンド行 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 を単体で実行すると、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る)。

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 Messaging Server に統合します。

1. ユーザーをグループに分割し (639 ページの「バックアップグループを作成するには」を参照)、*msg_svr_base*/config/ ディレクトリの下に backup-groups.conf ファイルを作成します。

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、ユーザー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。次に例を示します。

   ```
   # imsbackup -f- /primary/groupA > /bkdata/groupA &
   # imsbackup -f- /primary/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. ユーザーを復元するには、ユーザー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復元し、imsrestore を使用してデータファイルからユーザーを復元します。

   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 バイトより大きい場合は、次のコマンドを使用します。

   ```
   # cat /bkdata/groupA | imsrestore -f- /primary/groupA/andy
   ```

バックアップおよび復元の問題のトラブルシューティング

この節では、一般的なバックアップと復元の問題、およびその問題の解決策について説明します。

- **問題** : imsrestore または imsasm を使用してフォルダまたは INBOX を復元すると、そのフォルダ内のすべてのメッセージが現在のフォルダに追加されます。その結果、そのフォルダにメッセージの複数のコピーが作成されます。

  解決策 : imsasm スクリプトで、imsrestore の -i フラグが設定されていないことを確認してください。

- **問題** : メールフォルダに追加された新しいメッセージのみの増分バックアップを行おうとしたが、フォルダ全体がバックアップされてしまいます。新しいメッセージのみをバックアップする方法はありますか。

  解決策 : imsbackup の -d *datetime* フラグを設定します。このようにすると、指定された日時から現在までに保存されたメッセージがバックアップされます。デフォルトでは、日付に無関係にすべてのメッセージがバックアップされます。

メッセージストアの災害時のバックアップと復元

災害とは、メッセージストア全体に壊滅的な障害の発生を意味します。つまり、メッセージストアサーバー上のすべてのデータが失われた状態のことです。メッセージストアの災害時の完全な復元では、失われた次のデータが復元されます。

- すべてのメッセージストアデータ。このデータは、637 ページの「メッセージストアのバックアップと復元を行う」に説明されている手順を使ってバックアップできます。ファイルシステムバックアップ方法を使用する場合は、次のデータを必ずバックアップするようにします。
  - すべてのメッセージストアパーティション
  - msg_svr_base/data/store/mboxlist にあるメッセージストアデータベースファイル
- msg_svr_base/data/store/dbdata/snapshots にあるメッセージストアデータベースのスナップショット。メッセージストアデータベースのスナップショットファイルの場所はパラメータ local.store.snapshotpath で設定できます。ファイルシステムバックアップを使用する場合は、これらのデータが復元されたあとで reconstruct -m を実行します。
- すべての設定データ。次のデータがあります。
  - msg_svr_base/data/config にあるローカル設定ファイル
  - LDAP Directory Server にある Messaging Server の設定データ

ユーザーアクセスを監視する

Messaging Server では、imsconnutil コマンドが提供されます。このコマンドを使用して、ユーザーの IMAP、POP、および HTTP を介したメッセージストアアクセスを監視できます。また、ユーザーの最新のログインおよびログアウトを確認できます。このコマンドは、メッセージストア単位で機能するものであり、メッセージストア全体に対しては機能しません。

注 適用される法律または条例に違反する使用、または顧客自身のポリシーまたは契約に違反する使用の場合、この機能またはその他の Messaging Server の機能を使ってユーザーのメールを監視、読み取り、もしくはアクセスすると、不利益を引き起こす可能性があります。

このコマンドを使用するにはシステムユーザー (デフォルト : inetuser) によるルートアクセスが必要です。また、設定変数の local.imap.enableuserlist、local.http.enableuserlist、local.enablelastaccess を 1 に設定する必要があります。

IMAP または Web メールクライアントを介して現在ログインしているユーザーを一覧表示するには、次のコマンドを使用します。

```
# imsconnutil -c
```

メッセージストアのユーザーごとの最新の IMAP、POP、または Messenger Express によるアクセス (ログインおよびログアウト) を一覧表示するには、次のコマンドを使用します。

```
# imsconnutil -a
```

次のコマンドは 2 つの処理を実行します。1) 指定したユーザーが現在 IMAP、Messenger Express、または mshttp を介して接続しているクライアントからログインしているかどうか判別する (一般に、POP ユーザーの場合は常時接続でない場合があるので、この処理は POP には機能しないことに注意)。2) ユーザーが最後にログインおよびログアウトした時刻を一覧表示する。

```
# imsconnutil -c -a -u user_ID
```

ユーザーの一覧は、次のコマンドを使用して 1 行につき 1 ユーザーずつファイルで入力できます。

```
# imsconnutil -c -a -f filename
```

`-s` フラグを使用して、特定のサービス (`imap` または `http`) を指定することもできます。たとえば、特定のユーザー ID が IMAP にログインしたかどうかを一覧表示するには、次のコマンドを使用します。

```
# imsconnutil -c -s imap -u user_ID
```

`imsconnutil` の構文の詳細については、『Sun Java System Messaging Server Administration Reference』を参照してください。

次に出力例を示します。

```
$ ./imsconnutil -a -u soroork
UID IMAP last access HTTP last access POP last access
=====================================================================
soroork 08/Jul/2003:10:49:05 10/Jul/2003:14:55:52 ----NOT-RECORDED----

$ ./imsconnutil -c
IMAP
UID     TIME                    AUTH              TO                    FROM
=============================================================================
ed      17/Jun/2003:11:24:03  plain     172.58.73.45:193  129.157.12.73:2631
bill    17/Jun/2003:04:28:43  plain     172.58.73.45:193  129.158.16.34:2340
mia     17/Jun/2003:09:36:54  plain     172.58.73.45:193  192.18.184.103:3744
jay     17/Jun/2003:05:38:46  plain     172.58.73.45:193  129.159.18.123:3687
paul    17/Jun/2003:12:23:28  plaintext 172.58.73.45:193  192.18.194.83:2943
tony    17/Jun/2003:05:38:46  plain     172.58.73.45:193  129.152.18.123:3688
anil    17/Jun/2003:12:26:40  plaintext 172.58.73.45:193  192.18.164.17:1767
anil    17/Jun/2003:12:25:17  plaintext 172.58.73.45:193  129.150.17.34:3117
jack    17/Jun/2003:12:26:32  plaintext 172.58.73.45:193  129.150.17.34:3119
toni    17/Jun/2003:12:25:32  plaintext 172.58.73.45:193  192.18.148.17:1764
=============================================================================
10 users were logged in to imap.
Feature is not enabled for http.
-----------------------------------------------------------------------------
```

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、662 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、この章のこれまでの部分と、『Sun Java System Messaging Server Administration Reference』のコマンド行ユーティリティおよび `configutil` に関する章を再度読まれますよう、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。

- 655 ページの「標準的なメッセージストアの監視手順」
- 667 ページの「一般的な問題と解決策」
- 658 ページの「メッセージストアの起動と回復」

- 662 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」

標準的なメッセージストアの監視手順

ここでは、メッセージストアの監視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、メッセージストアの全般的なチェック、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、848 ページの「メッセージストアを監視する」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。

ディスクの空き容量の不足は、メールサーバーで発生する問題や故障のうち、特に頻繁におきる原因の 1 つです。メッセージストアへ書き込むとき、そのための容量が不足していると、メールサーバーに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。

ディスク容量の監視の詳細については、631 ページの「ディスク容量を監視するには」および848 ページの「メッセージストアを監視する」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。ログファイルはコンソールを使用して表示するか、*msg_svr_base*`/log/` ディレクトリで表示できます。ログファイルは定期的に監視する必要があります。

ログ記録はサーバー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、サーバー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。サーバーのログポリシーの定義の詳細については、第 21 章「ログの管理」を参照してください。

ユーザーの IMAP/POP セッションをチェックする

Messaging Server では、テレメトリと呼ばれる機能が提供されており、ユーザーの IMAP または POP セッション全体をファイルに取得できます。この機能は、クライアント問題をデバッグするのに便利です。たとえば、メッセージアクセスクライアントが期待どおりに機能しないとユーザーが訴えた場合、この機能を使用してアクセスクライアントと Messaging Server 間の対話を記録することができます。

セッションの記録をとるには、次のディレクトリを作成します。

msg_svr_base/data/telemetry/*pop_or_imap*/*userid*

Messaging Server によって、セッションにつき 1 ファイルがそのディレクトリに作成されます。出力例を次に示します。

```
LOGIN redb 2003/11/26 13:03:21
>0.017>1 OK User logged in
<0.047<2 XSERVERINFO MANAGEACCOUNTURL MANAGELISTSURL MANAGEFILTERSURL
>0.003>* XSERVERINFO MANAGEACCOUNTURL {67}
http://redb@cuisine.blue.planet.com:800/bin/user/admin/bin/enduser
MANAGELISTSURL NIL MANAGEFILTERSURL NIL
2 OK Completed
<0.046<3 select "INBOX"
>0.236>* FLAGS (¥Answered flagged draft deleted ¥Seen $MDNSent Junk)
* OK [PERMANENTFLAGS (¥Answered flagged draft deleted ¥Seen $MDNSent Junk ¥*)]
* 1538 EXISTS
* 0 RECENT
* OK [UNSEEN 23]
* OK [UIDVALIDITY 1046219200]
* OK [UIDNEXT 1968]
3 OK [READ-WRITE] Completed
<0.045<4 UID fetch 1:* (FLAGS)
>0.117>* 1 FETCH (FLAGS (¥Seen) UID 330)
* 2 FETCH (FLAGS (¥Seen) UID 331)
* 3 FETCH (FLAGS (¥Seen) UID 332)
* 4 FETCH (FLAGS (¥Seen) UID 333)
* 5 FETCH (FLAGS (¥Seen) UID 334)
<etc>
```

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。stored -t -v を実行します。
- *store_root*/mboxlist 内に作成されたログファイルをチェックします。
- デフォルトログファイルの *msg_svr_base*/log/default/default 内の stored メッセージをチェックします。
- 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場合は、必ず以下のファイル (*msg_svr_base*/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 18-14 stored 操作

stored 操作	説明
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ます。約 1 分ごとにスタンプが付けられます。
stored.lcu	データベースログのクリーンアップごとに押されます。約 5 分ごとにタイムスタンプが付けられます。
stored.per	ユーザー単位のデータベース書き込み時に押されます。タイムスタンプは 1 時間ごとに付けられます。

stored プロセスの詳細については、631 ページの「stored ユーティリティを使用する」および『Messaging Server Reference Manual』の Messaging Server コマンド行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 機能の監視の詳細については、848 ページの「メッセージストアを監視する」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 (*store_root*/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。

ユーザーフォルダのチェック

ユーザー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 no fix)。これにより、ユーザーフォルダおよびレポートのエラーを確認します。`reconstruct` コマンドの詳細については、662 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。Solaris の場合は、`coreadm` を使用して `core` ファイルの場所を設定します。

メッセージストアの起動と回復

メッセージストアのデータはメッセージ、インデックスデータ、およびメッセージストアデータベースで構成されています。このデータは堅固ですが、ごくまれにメッセージストアのデータに関する問題がシステムに存在することがあります。このような問題はデフォルトのログファイルに示され、ほとんどの場合は透過的に修正されます。ただし、`reconstruct` ユーティリティを実行する必要があることを示すエラーメッセージがログファイルに表示される場合がまれにあります。また、最終手段として、メッセージは 637 ページの「メッセージストアのバックアップと復元を行う」で説明されているバックアップと復元のプロセスによって保護されます。この節では、`stored` の自動起動および回復プロセスについて説明します。

メッセージストアでは、以前は管理者の職責であった多くの回復処理が自動化されています。これらの処理はメッセージストアデーモンの `stored` によって起動時に実行され、必要に応じてデータベーススナップショットおよび自動高速復元が含まれます。`stored` によってメッセージストアのデータベースが徹底的にチェックされ、問題が検出された場合は自動的に修復されます。

また、`stored` は、デフォルトのログにステータスメッセージを出力することで、データベースのステータスの総合的な分析を提供し、メッセージストアに行われた修復およびメッセージストアを回復するために行われた自動試行について報告します。

自動起動と自動回復 - 動作方式

`stored` デーモンは、ほかのメッセージストアプロセスが起動する前に起動します。このデーモンによってメッセージストアデータベースは初期化され、必要に応じて回復処理が行われます。メッセージストアデータベースは、フォルダ、容量制限、購読、およびメッセージフラグの情報を保持します。データベースはログ用とトランザクション用であるので、回復はすでに組み込まれています。また、一部のデータベース情報は、各フォルダのインデックス領域に予備でコピーされています。

データベースは非常に堅固ですが、まれに壊れたとしても、ほとんどの場合は stored によって透過的に回復および修復されます。ただし、stored が再起動された場合は毎回、デフォルトのログファイルをチェックして、ほかに管理操作が必要ないことを確認してください。データベースをさらに修復する必要がある場合は、ログファイルのステータスメッセージで reconstruct を実行するように示されます。

メッセージストアデータベースを開く前に、stored はデータベースの完全性を分析し、ステータスメッセージを *warning* のカテゴリの下にあるデフォルトログに出力します。メッセージには管理者にとって有用なものも、内部分析に使用されるコード化されたデータで構成されるものもあります。stored によって問題が検出された場合は、データベースの修復が試行され、再起動が試行されます。

データベースが開くと、stored は、ほかのサービスが起動することを合図します。自動修復が失敗した場合、デフォルトログのメッセージで実行するべきアクションが示されます。詳細は、659 ページの「reconstruct -m が必要であることを示すエラーメッセージ」を参照してください。

以前のリリースでは、stored は非常に長い時間がかかる回復プロセスを開始することがあり、stored が「スタック」したかのように見えることがありました。このタイプの長い回復プロセスは取り除かれ、stored は最終的な状態を 1 分以内に判断するようになりました。ただし、stored がスナップショットからの回復などの回復手段を採用する必要がある場合、プロセスは数分かかる場合があります。

ほとんどの回復プロセスでは通常、終了後のデータベースは最新の状態になっていて、ほかに必要な操作はありません。ただし、一部の回復プロセスでは、reconstruct -m を実行してメッセージストアの冗長データを同期させる必要がある場合もあります。これもデフォルトログに示されます。したがって、起動後にデフォルトログを監視することは重要です。メッセージストアが通常どおり起動し、機能しているように見える場合でも、reconstruct など、要求されている操作がある場合は実行することが重要です。

ログファイルを読むもう 1 つの理由は、データベースに障害を引き起こした原因を確認することです。stored は、システムでの問題の種類にかかわらずメッセージストアを回復するように設計されていますが、データベース障害はより大きな問題が潜んでいることの徴候である可能性があるので、原因を解明することをお勧めします。

reconstruct -m が必要であることを示すエラーメッセージ

ここでは、reconstruct -m の実行が必要なエラーメッセージのタイプについて説明します。

エラーメッセージでメールボックスエラーが示された場合は、reconstruct <mailbox> を実行します。次に例を示します。

```
"Invalid cache data for msg 102 in mailbox user/joe/INBOX. Needs
reconstruct"

"Mailbox corrupted, missing fixed headers: user/joe/INBOX"
```

"Mailbox corrupted, start_offset beyond EOF: user/joe/INBOX"

エラーメッセージでデータベースエラーが示された場合は、reconstruct -m を実行します。次に例を示します。

"Removing extra database logs. Run reconstruct -m soon after startup to resync redundant data"

"Recovering database from snapshot. Run reconstruct -m soon after startup to resync redundant data"

データベーススナップショット

スナップショットは、データベースのホットバックアップであり、壊れたデータベースを数分で透過的に回復するために stored で使用されます。これは、ほかの領域に保存された冗長情報に依存する reconstruct を使用するよりもはるかに速い方法です。

メッセージストアのデータベーススナップショット - 動作方式

データベースのスナップショット (mboxlist ディレクトリ内) は自動的に作成されます。デフォルトでは、24 時間ごとに作成されます。デフォルトでは、スナップショットは store ディレクトリのサブディレクトリにコピーされます。デフォルトでは、常時 5 つのスナップショットが保存されています。ライブデータベースが 1 つ、スナップショットが 3 つ、データベース / 削除済みコピーが 1 つです。データベース / 削除済みコピーはより新しいものであり、mboxlist データベースディレクトリの removed サブディレクトリに入れられたデータベースの非常時用のコピーです。

現在のデータベースに障害があるために回復プロセスで削除することが決定されると、stored がデータベースを removed ディレクトリに移動します (可能な場合)。したがって、必要に応じてそのデータベースを分析できるようになっています。

データの移動は、1 週間に 1 度だけ実行されます。データベースのコピーがすでに移動先に存在する場合、stored はストアが起動するたびごとにはコピーを置き換えません。removed ディレクトリのデータが 1 週間よりも古い場合にのみ置き換えます。これは、元のデータベースが一連の起動によってあまりにも早く置き換えられないようにするためです。

メッセージストアのデータベーススナップショットの間隔と場所を指定するには

データベースとスナップショットを組み合わせるには、5 倍の容量が必要です。スナップショットが別のディスク上で実行されるように再設定し、システムの要件に合わせることを強くお勧めします。

stored によって起動時にデータベースに関する問題が検出された場合は、最善のスナップショットが自動的に回復します。3 つのスナップショット変数を使用して設定できるパラメータは、次のとおりです。スナップショットファイルの場所、スナップショットの作成間隔、保存されるスナップショットの数。表 18-15 に、これらの configutil パラメータを示します。

スナップショットの間隔が短すぎると、システムに頻繁に負荷がかかるとともに、データベースの問題がスナップショットとしてコピーされる可能性が高くなります。スナップショットの間隔が長すぎると、データベースはスナップショットが作成された過去の時点の状態を維持することになります。

スナップショット間隔は 1 日にすることをお勧めします。1 週間またはそれより長い間隔のスナップショットは、システムで問題が数日間解消されない場合に、問題が存在する前の状態に戻すのに便利です。

stored はデータベースの監視を実行し、データベースが完全でない可能性がある場合は最新のスナップショットを拒否する高度な機能があります。代わりに最新のもっとも信頼性の高いスナップショットを取り出します。1 日前のスナップショットが取り出される可能性があることにもかかわらず、システムはより新しい冗長データがある場合はそのデータを使用して古いスナップショットデータを無効にします。

つまり、スナップショットの根本的な役割は、システムを最新の状態に近づけることと、進行中のデータを再構築しようとするシステムのほかの部分の負担を軽減することです。

表 18-15 メッセージストアデータベーススナップショットのパラメータ

パラメータ	説明
local.store.snapshotpath	メッセージストアのデータベーススナップショットファイルの場所。既存の絶対パスまたは store ディレクトリを基準とする相対パスのいずれかになります。 デフォルト : dbdata/snapshots
local.store.snapshotinterval	スナップショット間隔 (単位 : 分)。有効な値 : 1 ～ 46080 デフォルト : 1440 (1440 分 = 1 日)
local.store.snapshotdirs	保存される異なるスナップショットの数。有効な値 : 2 ～ 367 デフォルト : 3

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。659 ページの「reconstruct -m が必要であることを示すエラーメッセージ」を参照してください。

注 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は起動時に自動的に実行されます。

表 18-16 では、reconstruct オプションを一覧表示しています。構文や使用要件の詳細については、『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) を参照してください。

表 18-16 reconstruct オプション

オプション	説明
-e	再構築の前に、store.exp ファイルを削除します。これは、ストア処理によって消去されなかった削除済みメッセージの内部ストアレコードを除去します。-i または -e を使用するときに -f オプションも使用すると役立ちます。これは、これらのオプションはフォルダが実際に再構築された場合にのみ機能するからです。同様に、-n オプション (再構築ではなくチェックを実行する) を使用する場合は、-i および -e オプションは機能しません。 reconstruct が破損を検出しない場合は、reconstruct -e は削除済みメッセージを復元しません。-f は、再構築を強要します。
-i	再構築の前に、store.idx ファイル長を 0 に設定します。-i または -e を使用するときに -f オプションも使用すると役立ちます。これは、これらのオプションはフォルダが実際に再構築された場合にのみ機能するからです。同様に、-n オプション (再構築ではなくチェックを実行する) を使用する場合は、-i および -e オプションは機能しません。
-f	reconstruct に 1 つまたは複数のメールボックスで修復を行うように強制します。
-l	lright.db を再構築します。

表 18-16 reconstruct オプション (続き)

オプション	説明
-m	整合性チェックを行い、必要に応じてメールボックスデータベースを修復します。このオプションを使用すると、スプールエリアで見つかった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のデータベースエントリの追加または削除が行われます。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。つまり、folder.db、quota.db、および lright.db を修復します

-n	メールボックスの修復を実行せずに、メッセージストアだけをチェックします。メールボックス名を指定せずに、-n オプションを単独で使用することはできません。メールボックス名を指定しない場合、-n オプションは -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	廃止。mboxutil -o を参照してください。
-o -d *filename*	廃止。mboxutil -o を参照してください。
-p *partition*	-p オプションを -m オプションとともに使用し、再構築の範囲を指定されたパーティションに制限します。-p オプションを指定しない場合、reconstruct はデフォルトですべてのパーティションを再構築します。つまり、folder.db および quota.db を修復しますが、lright.db は修復しません。これは lright.db を修復すると、メッセージストア内のすべてのユーザーに対して ACL のスキャンを実行する必要があるためです。これをすべてのパーティションに対して実行するのは効率的でありません。lright.db を修復するには、reconstruct -l を実行します。 パーティション名を指定します。完全なパス名は使用しないでください。
-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、ほかのサーバープロセスの実行中に実行できます。
-r [*mailbox*]	指定した 1 つまたは複数のメールボックスのパーティションエリアを修復し、整合性をチェックします。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は、ユーザー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エリアが修復されます。

表 18-16 reconstruct オプション (続き)

オプション	説明
-u *user*	-u オプションを -m オプションとともに使用し、再構築の範囲を指定されたユーザーに制限します。-u オプションは、-p オプションとともに使用する必要があります。-u オプションを指定しない場合、reconstruct はすべてのパーティションまたは -p オプションで指定されたパーティションを再構築します。 ユーザー名を指定します。完全なパス名は使用しないでください。

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、「システム I/O エラーが発生しました」または「メールボックスのフォーマットが不正です」のどちらかのエラーが返された場合。
- メールボックスにアクセスしたらサーバーがクラッシュした場合。
- 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場合。

reconstruct -r は、最初に整合性チェックを行います。問題が検出された場合にのみ、すべての整合性を報告し、再構築を行います。したがって、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。

ユーザー 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 (666 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
```

コマンド行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ーフォルダ 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、次のように入力します。

```
reconstruct -m
```

整合性チェックを行い、プライマリパーティションを修復するには次のように入力します。

```
reconstruct -p primary -m
```

注 -p および -m フラグを指定して reconstruct を実行しても、lright.db は修復されません。これは lright.db を修復すると、メッセージストア内のすべてのユーザーに対して ACL のスキャンを実行する必要があるためです。これをすべてのパーティションに対して実行するのは効率的でありません。lright.db を修復するには、reconstruct -l を実行します。

整合性チェックを行い、john という名前のユーザーのメールボックスを修復するには次のように入力します。

```
reconstruct -p primary -u john -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復元されたため、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。

  - すべてのサーバーを停止します。
  - *store_root*/mboxlist 内のすべてのファイルを削除します。
  - サーバー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。

reconstruct のパフォーマンス

reconstruct が処理を実行するのにかかる時間は、次に示すいくつかの要素によって異なります。

- 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プロセス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。

ユーザー数が約 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ーで実行されているシステムでは、次のパフォーマンスが得られました。

- reconstruct -m に要した時間は約 1 時間
- reconstruct -r -f に要した時間は約 18 時間

注 reconstruct の操作にかかる時間は、サーバーで 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。

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。

Messenger Express または Communications Express がメールページを読み込まない

ユーザーが Messenger Express ページまたは Communications Express メールページを読み込めない場合は、圧縮後にデータが破損している可能性があります。これは、古いプロキシサーバーがシステムに配備されている場合に発生することがあります。この問題を解決するには、`local.service.http.gzip.static` および `local.service.http.gzip.dynamic` を 0 に設定して、データ圧縮を無効にしてください。これで問題が解決したら、プロキシサーバーを更新することができます。

ワイルドカードパターンを使用したコマンドが機能しない

UNIX シェルには、ワイルドカードパターンを引用符で囲む必要があるものとその必要のないものがあります。たとえば、C シェルはワイルドカード (*、?) を含む引数をファイルとして展開しようとするため、一致するものがない場合は失敗します。これらのパターンマッチング引数は、`mboxutil` のようなコマンドに渡すためには引用符で囲む必要があります。

次に例を示します。

```
mboxutil -l -p user/usr44*
```

これは Bourne シェルで機能しますが、tsch や C シェルでは失敗します。これらのシェルには次のコマンドが必要です。

```
mboxutil -l -p "user/usr44*"
```

ワイルドカードパターンを使用したコマンドが機能しない場合は、そのシェルではワイルドカードを引用符で囲む必要があるかどうかを確認してください。

不明または無効なパーティション

ユーザーのメールボックスが作成したばかりの新しいパーティションに移動され、Messaging Server が更新または再起動されていない場合、Messenger Express で「パーティションが不明または無効です」というメッセージが表示されることがあります。この問題は新しいパーティションでのみ発生します。この新しいパーティションにユーザーメールボックスを新しく追加する場合、Messaging Server の更新または再起動を行う必要はありません。

ユーザーメールボックスディレクトリに関する問題

ユーザー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ーに限られていて、システム全体に対する損傷がないときです。ユーザー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、またはユーザー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*/mboxlist/ ユーザーディレクトリ全体を、メッセージストア外部の別の場所にコピーします。
3. 問題の原因になっている可能性のあるユーザーフォルダを見つけるには、`reconstruct -r -n` コマンドを実行します。`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` の詳細については、628 ページの「hashdir ユーティリティ」および、『Messaging Server Reference Manual』の Messaging Server コマンド行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 (*store_root*/mboxlist) 内にはなく、`partition` ディレクトリ (*store_root*/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 (*store_root*/partition/ ディレクトリ) 上にあっても、明らかにデータベース (*store_root*/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 コマンドの詳細については、662 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。

store デーモンが起動しない

stored が起動せずに次のエラーメッセージが表示される場合があります。

```
# msg_svr_base/sbin/start-msg

msg_svr_base: Starting STORE daemon ...Fatal error: Cannot find group in name service
```

上記のメッセージは、local.servergid に設定された UNIX グループが見つからないことを示しています。Stored などは、gid をグループに設定する必要があります。local.servergid によって定義されたグループが誤って削除されることがあります。この場合は、削除されたグループを作成し、inetuser をグループに追加し、*instance_root* の所有権とそのファイルを inetuser とグループに変更します。

第 19 章

セキュリティとアクセス制御を設定する

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、侵入者がユーザー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ー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、Sun Java System サーバー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、この章だけでなくほかキュメントも参照する必要があります。特に、『Sun ONE Server Console 5.2 Server Management Guide』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。

サーバーのセキュリティについて

サーバー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、承認されたユーザーだけがサーバー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、ほかのユーザー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。

サーバー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、アクセス制御の設定に重点を置きます。この章で説明する Messaging Server のセキュリティ関連の内容は、次のとおりです。

- **ユーザー ID とパスワードログイン** : ユーザーは、IMAP、POP、HTTP、または SMTP にログインするためにユーザー 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ーを設定します。
- **管理者によるアクセス制御** : コンソール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、サーバーの 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** : サーバー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なったり、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 18 章「メッセージストアを管理する」を参照してください。
- **エンドユーザーアカウントの設定** : エンドユーザーアカウント情報は、主に Delegated Administrator 製品を使って管理できます (Sun LDAP Schema 1 の場合のみ有効)。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ーアカウントを管理することもできます。
- **不特定多数宛メール (UBE) のフィルタリング** : 第 17 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。

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があります。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、文書 Web サイト (http://docs.sun.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、ユーザー ID とパスワード認証、クライアント証明書認証、および Access Manager をサポートしています。ただし、クライアントとサーバー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ー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、サーバー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、サーバーは自動的に 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 が、特定の IP アドレスにバインドされる。
- 各セッション ID に、タイムアウト値が関連付けられる。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ーが管理する。このため、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 5 章「POP、IMAP、および HTTP サービスの設定」を参照してください。

Access Manager の詳細については、143 ページの第 6 章「シングルサインオン (SSO) の有効化」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ーに提示する方法の 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については、679 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、ユーザー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、679 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 / 応答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 / 応答型の認証メカニズムですが、ほか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 / 応答型の認証メカニズム。RFC 1939 に定義されています。
- LOGIN - PLAIN と同等。SMTP 認証の標準化前の実装との互換性を保つためにのみ存在します。デフォルトでは、このメカニズムは SMTP で使用される場合にのみ有効になります。

チャレンジ / 応答型の認証メカニズムでは、サーバー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ー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ー自体のハッシュと一致すると、ユーザーが認証されます。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ー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。

表 19-1 に、SASL および SASL 関連の `configutil` パラメータの一部を示します。`configutil` パラメータがもっとも多く挙げられている最新のリストは、『Sun Java System Messaging Server Administration Reference』を参照してください。

表 19-1 SASL および SASL 関連の configutil パラメータの一部

パラメータ	説明
sasl.default.ldap.has_plain_passwords	ブール代数値。ディレクトリがプレーンテキストパスワードを格納していることを示します。この値により APOP、CRAM-MD5、および DIGEST-MD5 が有効になります。 デフォルト : False
sasl.default.transition_criteria	サポートされず使用されません。sasl.default.auto_transition を参照してください。
sasl.default.auto_transition	ブール型。設定し、ユーザーがプレーンテキストのパスワードを入力した場合、パスワード保存形式がディレクトリサーバーのデフォルトのパスワード保存形式に移行されます。プレーンテキストのパスワードから、APOP、CRAM-MD5 または DIGEST-MD5 への移行に使用することができます。 デフォルト : False
service.imap.allowanonymouslogin	IMAP による使用のため、SASL ANONYMOUS メカニズムを有効にします。 デフォルト : False
service.{imap|pop|http}.plaintextmincipher	1 より大きな値に設定すると、セキュリティレイヤ (SSL または TLS) が有効でない限り、プレーンテキストのパスワードの使用を無効にします。これによりユーザーは、ログインする自分のクライアントで SSL または TLS を強制的に有効にすることになり、自分のパスワードがネットワーク上で漏洩することを防ぎます。MMP には同等のオプション「RestrictPlainPasswords」があります。 **注** : Messaging Server の 5.2 リリースでは、SSL または TLS が使用する暗号の強度の数値が調べられます。この機能はオプションの簡潔化のため、また一般的な使用法に合わせるため削除されました。 デフォルト : 0
sasl.default.mech_list	有効にする SASL メカニズムの、スペース区切りのリスト。空でない場合、この設定は sasl.default.ldap.has_plain_passwords オプションおよび service.imap.allowanonymouslogin オプションよりも優先します。このオプションは、すべてのプロトコル (imap、pop、smtp) に適用されます。 デフォルト : False
sasl.default.ldap.searchfilter	ユーザーがドメインの inetDomainSearchFilter に指定されていない場合に、ユーザーの検索に使用されるデフォルトの検索フィルタ。構文は inetDomainSearchFilter と同じ (スキーマガイドを参照)。 デフォルト : (&(uid=%U)(objectclass=inetmailuser))

表 19-1　SASL および SASL 関連の configutil パラメータの一部 (続き)

パラメータ	説明
sasl.default.ldap.searchfordomain	デフォルトでは、認証システムは LDAP 内のドメインをドメイン検索のルールに従って検索し (参照は必要)、その後ユーザーを検索します。ただし、このオプションがデフォルトの「1」ではなく「0」に設定されている場合、ドメイン検索は行われず、sasl.default.ldap.searchfilter を使用したユーザーの検索が local.ugldapbasedn で指定した LDAP ツリーの直下で行われます。旧バージョンの単一ドメインスキーマとの互換性のために提供されていますが、小さな企業であっても合併や名称変更により複数ドメインのサポートが必要になる可能性があるため、新しい配備のための使用にはお勧めしません。

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、ユーザー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. 「設定」タブをクリックします。
3. 左のペインで「Data」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「cleartext」を選択します。

注　この変更は、将来作成するユーザーにのみ影響を与えます。既存のユーザーは、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 に設定すると、これらのチャレンジ / 応答型の SASL メカニズムを無効にすることができます。

注	既存のユーザーは、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「ユーザーを移行するには」を参照)。

MMP には同等のオプション「CRAM」があります。

ユーザーを移行するには

`configutil` を使用して、移行するユーザーに関する情報を指定できます。たとえば、ユーザーパスワードを変更する場合や、適切なユーザー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.auto_transition -v value
```

value には、次のいずれかを指定できます。

- `no` または `0` - パスワードを移行しない。これがデフォルトです。
- `yes` または `1` - パスワードを移行する。

ユーザーを正常に移行するには、Messaging Server がユーザー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. ユーザー / グループツリーのベースサフィックスを選択します。
4. 「オブジェクト」メニューから「アクセス権を設定」を選択します。
5. 「Messaging Server End User Administrator Write Access Rights (Messaging Server エンドユーザー管理者書き込みアクセス権」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. 「了解」をクリックします。

sasl.default.mech_list を使用して SASL メカニズムのリストを有効にできます。空でない場合、この設定は sasl.default.ldap.has_plain_passwords オプションおよび service.imap.allowanonymouslogin オプションよりも優先します。このオプションは、すべてのプロトコル (imap、pop、smtp) に適用されます。

ユーザー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、ユーザー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、内部ユーザーは、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については、130 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。

ユーザーパスワードは、クリアテキストまたは暗号文の形式で、ユーザーのクライアントソフトウェアからサーバーに転送できます。クライアントとサーバーの両方が、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 (687 ページの「SSL を有効化し暗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。

ユーザー ID とパスワードは、LDAP ユーザー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。689 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 / 応答型の 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、ユーザー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」とは、クライアントがサーバー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のことです。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、ほかのユーザーが悪用できるオープンリレーの発生を防ぎながら、ローカルユーザーが移動先から (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ー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、366 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

この節では、以下の項目に分けて説明しています。

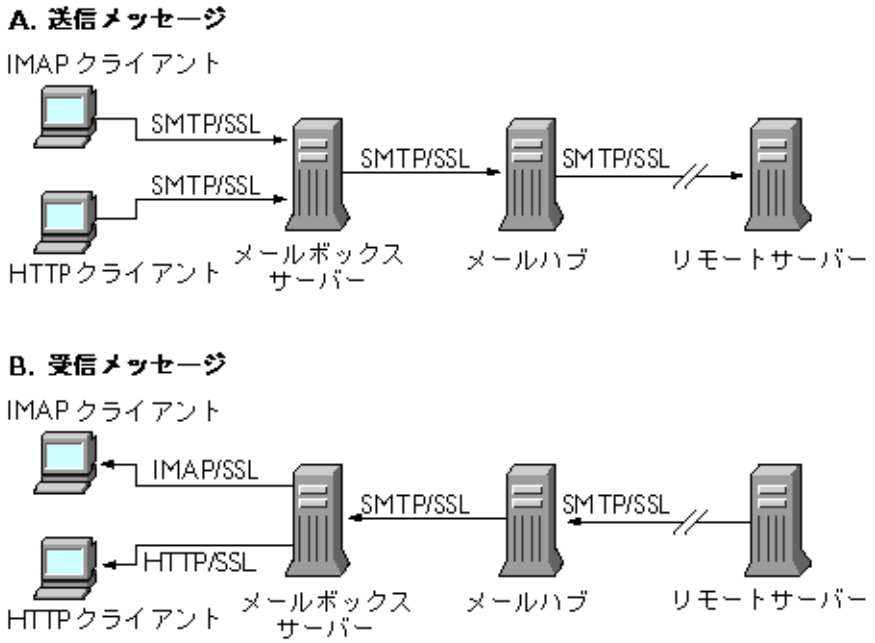
Messaging Server では、クライアントとサーバー間で、暗号化された通信および証明書に基づく認証を行うために TLS (Transport Layer Security) プロトコルを使用します。TLS プロトコルは、SSL (Secure Sockets Layer) プロトコルとも呼ばれます。Messaging Server は、SSL バージョン 3.0 および 3.1 をサポートします。TLS には、SSL との完全な互換性があり、必要な SSL 機能がすべて含まれています。

SSL に関する背景情報については、『Managing Servers with iPlanet Console』の付録の「Introduction to SSL」を参照してください。SSL は、公開キー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 iPlanet 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 とほかのサーバー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、POP3 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、POP と POP/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 19-1 に示すように、SSL は、送信メッセージと着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 19-1 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ー上ではメッセージは暗号化されません。

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、369 ページの「Transport Layer Security」および『Messaging Server Reference Manual』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、サーバー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ー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 Sun Java System サーバー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コンソールインタフェースは多くのサーバー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、『Managing Servers with iPlanet 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。

管理コンソールからの証明書の入手

SSL の用途が暗号化か認証かにかかわらず、Messaging Server 用のサーバー証明書を入手する必要があります。この証明書は、使用するサーバーの識別情報をクライアントやほかのサーバーに提供します。管理コンソールから証明書を入手する場合は、この節の手順に従います。コマンド行モードで自己署名済み証明書を作成する場合は、685 ページの「自己署名済み証明書を作成するには」を参照してください。

内部モジュールと外部モジュールを管理するには

サーバー証明書によって、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。サーバーの証明書とキーのペアは、そのサーバーの識別情報を示します。これらは、サーバー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。

Sun Java System サーバーは、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、サーバー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ーを設定する場合は、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、サーバー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、『Managing Servers with iPlanet 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、サーバー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。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。

サーバー証明書を要求するには

サーバー証明書を要求するには、コンソールでサーバーを開き、「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、『Managing Servers with iPlanet 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。
3. 証明書のニックネームを `server-cert` から `Server-Cert` に変更します。

   証明書のニックネームを変更したくない場合は、`configutil` パラメータ `encryption.rsa.nssslpersonalityssl` を設定して、使用したい証明書ニックネームを変更することができます。

詳細な手順については、『Managing Servers with iPlanet 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。サーバー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。サーバーは、クライアントやほかのサーバー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を設定するには」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、ほか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、『Managing Servers with iPlanet Console』の「Introduction to Public-Key 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。ほかの商用認証局の CA 証明書を追加する場合や、社内使用のために (Sun Java System 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、684 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、『Managing Servers with iPlanet 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

サーバーには、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ーを開き、「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、『Managing Servers with iPlanet 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 Sun Java System サーバー上で、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 Sun Java System サーバーでは、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 つのサーバープロセスで入力が必要)、さらに無人でサーバー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` という名前で、ディレクトリ *msg_svr_base*`/config/` に保存されています。ファイル内の各エントリは、次の形式で 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 管理者はサーバー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、およびサーバー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。

自己署名済み証明書を作成するには

コマンド行モードで自己署名済み証明書を作成する場合は、この節の手順に従います。証明書ウィザードで証明書を作成するには、681 ページの「管理コンソールからの証明書の入手」を参照してください。

1. スーパーユーザー (root) としてログインします。
2. /opt/SUNWmsgsr/config/sslpassword に certutil の証明書データベースのパスワードを指定します。次に例を示します。

   ```
   # echo "password" > /opt/SUNWmsgsr/config/sslpassword
   ```

   password は、ユーザー固有のパスワードです。
3. sbin ディレクトリに移動し、証明書データベース (cert8.db) とキーのデータベース (key3.db) を生成します。次に例を示します。

   ```
   # cd /opt/SUNWmsg/sbin
   # ./certutil -N -d /opt/SUNWmsgsr/config -f
   /opt/SUNWmsgsr/config/sslpassword
   ```
4. デフォルトの自己署名済みルート認証局証明書を生成します。次に例を示します。

```
# ./certutil -S -n SampleRootCA -x -t "CTu,CTu,CTu"
-s "CN=My Sample Root CA, O=sesta.com" -m 25000
-o /opt/SUNWmsgsr/config/SampleRootCA.crt
-d /opt/SUNWmsgsr/config
-f /opt/SUNWmsgsr/config/sslpassword   -z /etc/passwd
```

5. ホストの証明書を生成します。次に例を示します。

```
../certutil -S -n Server-Cert -c SampleRootCA -t "u,u,u"
-s "CN=hostname.sesta.com, o=sesta.com" -m 25001
-o /opt/SUNWmsgsr/config/SampleSSLServer.crt
-d /opt/SUNWmsgsr/config -f /opt/SUNWmsgsr/config/sslpassword
-z /etc/passwd
```

hostname.sesta.com は、サーバーホスト名です。

6. 証明書の妥当性を検査します。次に例を示します。

```
# ./certutil -V -u V -n  SampleRootCA -d /opt/SUNWmsgsr/config
# ./certutil -V -u V -n  Server-Cert -d /opt/SUNWmsgsr/config
```

7. 証明書を列挙します。次に例を示します。

```
# ./certutil -L -d /opt/SUNWmsgsr/config
# ./certutil -L -n Server-Cert -d /opt/SUNWmsgsr/config
```

8. modutil を使って、使用可能なセキュリティモジュール (secmod.db) を列挙します。次に例を示します。

```
# ./modutil -list -dbdir /opt/SUNWmsgsr/config
```

9. 次の例が示すように、証明書データベースファイルの所有者をメールサーバーのユーザーおよびグループに変更します。

```
chown mail:mailserv /opt/SUNWmsgsr/config/cert8.db
chown mail:mailserv /opt/SUNWmsgsr/config/key3.db
```

10. メッセージングサービスを再起動して、SSL を有効にします。

注 以前は、証明書とキーファイルは常に Messaging Server の設定ディレクトリにありました。現在は、local.ssldbpath (証明書とキーファイルの場所を指定する) および local.ssldbprefix (証明書とキーファイルのプレフィックスを指定する) を使用して、これらのファイルの場所を指定できます。

SSL を有効化し暗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暗号化方式を選択できます。

暗号化方式について

「暗号化方式」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。各暗号化方式によって強度が異なります。つまり、強度の高い暗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ーによる解読が困難になります。

暗号化方式では、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、暗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、サーバーに対して、希望する暗号化用の暗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暗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暗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、サーバーは柔軟な暗号化サポートを提供する必要があります。Messaging Server では、最大 6 つの暗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暗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、『Managing Servers with iPlanet Console』の「Introduction to SSL」を参照してください。

表 19-2 Messaging Server の SSL 暗号化方式

暗号化方式	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 が提供する暗号化方式で、もっとも高速で、もっとも強度の高い暗号化方式と暗号化キーの組み合わせを提供します。
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暗号化方式で、低速で、強度の高い暗号化方式と暗号化キーの組み合わせを提供します。
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暗号化方式で、低速で、中程度の強度の暗号化方式と暗号化キーの組み合わせを提供します。
40 ビットの暗号化と MD5 メッセージ認証を使用した RC4	RSA が提供する暗号化方式で、もっとも高速で、強度の低い暗号化方式と暗号化キーの組み合わせを提供します。

表 19-2　Messaging Server の SSL 暗号化方式 (続き)

暗号化方式	説明
40 ビットの暗号化と MD5 メッセージ認証を使用した RC2	RSA が提供する暗号化方式で、低速で、強度の低い暗号化方式と暗号化キーの組み合わせを提供します。
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を使用します。

特定の暗号化方式を使わないようにする妥当な理由がないかぎり、すべての暗号化方式を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 ビットの暗号化方式では、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。

SSL を有効にし、暗号化方式を選択するには、次のコマンド行を実行します。

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 暗号化方式を有効化 / 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 暗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。

優先する暗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暗号化方式のリストです。

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、369 ページの「Transport Layer Security」および『Messaging Server Reference Manual』を参照してください。

証明書に基づくログインを設定するには

Sun Java System サーバーでは、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ー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ーとの SSL セッションを確立し、ユーザーの証明書をサーバーに提出します。その後、サーバーが、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、そのユーザー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ー用の証明書を入手します (詳細は、681 ページの「管理コンソールからの証明書の入手」を参照)。
2. 証明書セットアップウィザードを実行して、サーバーが認証するユーザー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、683 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)。

   サーバー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、サーバー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、687 ページの「SSL を有効化し暗号化方式を選択するには」を参照)。
4. サーバーが提出された証明書の情報に基づいて LDAP ユーザーディレクトリを適切に検索するように、サーバーの `certmap.conf` ファイルを編集します (省略可)。

   ユーザーの証明書内の電子メールアドレスと、ユーザー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ー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` の形式と変更可能な部分の詳細については、『Managing Servers with iPlanet 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、ユーザーが、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ーの証明書が要求されます。サーバー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ーディレクトリ内のエントリと一致する場合、そのユーザーは、認証され、ユーザーに適用されるアクセス制御ルール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。

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 台のサーバー上で、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ーで 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、707 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。

ネットワークセキュリティサービスツール

ネットワークセキュリティサービスとは、オープン規格に基づいたインターネットセキュリティのためのアプリケーションを実装および配備するのに使用するオープンソースのライブラリおよびツールのセットのことです。セキュリティツールは、診断の実行、証明書、キーおよび暗号化モジュールの管理、また SSL および TLS ベースのアプリケーションのデバッグに役立ちます。これらのツールは、`/usr/sfw/bin` にあります。

証明書とキーの管理

ここで説明するツールは、暗号化および識別に利用するキーおよび証明書を保存、取得、および保護します。

certutil

証明書データベースツールの `certutil` は、`cert8.db` および `key3.db` データベースファイルを作成および変更できるコマンド行ユーティティです。キーおよび証明書管理プロセスは、一般にキーをキーのデータベースに作成することから始まり、それから証明書を証明書データベースに生成して管理します。`certutil` については、次の URL を参照してください。

http://www.mozilla.org/projects/security/pki/nss/tools/certutil.html

cmsutil

cmsutil コマンド行ユーティリティは、S/MIME ツールキットを使用して、暗号化メッセージ構文 (Cryptographic Message Syntax、CMS) メッセージに対する暗号化や復号化などの基本的な操作を実行します。このユーティリティは、メッセージの暗号化、復号化、および署名などの基本的な証明書管理操作を実行します。cmsutil については、次の URL を参照してください。

http://www.mozilla.org/projects/security/pki/nss/tools/cmsutil.html

modutil

セキュリティモジュールデータベースツールの modutil は、PKCS #11 モジュール (secmod.db ファイル) のデータベースの管理用のコマンド行ユーティリティです。このツールを使用して、PKCS #11 モジュールを追加および削除し、パスワードを変更し、デフォルトを設定し、モジュールの内容を表示し、スロットを使用可または使用不可にし、FIPS-140-1 準拠を有効または無効にし、暗号化操作にデフォルトのプロバイダを割り当てることができます。modutil については、次の URL を参照してください。

http://www.mozilla.org/projects/security/pki/nss/tools/modutil.html

pk12util

pk12util コマンド行ユーティリティは、PKCS #12 規格で定義された、キーと証明書の両方を、対応するデータベースに対して、また対応するファイル形式でインポートおよびエクスポートします。pk12util については、次の URL を参照してください。

http://www.mozilla.org/projects/security/pki/nss/tools/pk12util.html

ssltap

SSL デバッグツールの ssltap は、SSL 対応のコマンド行プロキシです。このツールは、SSL サーバーに対して要求を代理処理し、クライアントとサーバー間で交換されるメッセージの内容を表示できます。このツールは、TCP 接続を監視し、通過するデータを表示します。接続が SSL の場合は、表示されるデータには解釈された SSL レコードとハンドシェーキングが含まれます。詳細は、次の URL を参照してください。

http://www.mozilla.org/projects/security/pki/nss/tools/ssltap.html

Messaging Server への管理者アクセスを構成する

この節では主に Sun Java System LDAP スキーマ v. 1 について説明します。次の項があります。

この節では、サーバー管理者による 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、委任サーバー管理を行うときに発生します。

「委任サーバー管理」とは、ほとんどの Sun Java System サーバーが持っている機能で、管理者が、ほかの管理者に対して、個々のサーバーやサーバー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、委任されたサーバーのタスクについて簡単に説明します。詳細は、『Managing Servers with iPlanet Console』のサーバー管理の委任に関する章を参照してください。

委任管理の階層

ネットワーク上に最初の Sun Java System サーバー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 ユーザー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ー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 (Sun Java System LDAP スキーマ v. 1 を使用している場合) を実装することができます。

1. **構成管理者** : Sun Java System サーバーネットワークの「スーパーユーザー」。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. **サーバー管理者** : ドメイン管理者は、各タイプのサーバー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 (ほかのサーバーにはアクセス不可)。

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ー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ータスク (サーバーの起動または停止、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ます。次の節の「サーバー全体に対するアクセス権を与えるには」を参照してください。
- 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します。694 ページの「特定タスクへのアクセスを限定するには」を参照してください。

サーバー全体に対するアクセス権を与えるには

ユーザーまたはグループに 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ーを選択します。

   「コンソール」ウィンドウのメニューから「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. そのサーバーへのアクセス権を持つユーザー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、『Managing Servers with iPlanet Console』のサーバー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ー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、特定のサーバータスクをリスト内の特定のユーザー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、管理者はサーバーに接続して 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、そのサーバー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。サーバーは、接続しているユーザー (サーバー全体に対するアクセス権をすでに持っているユーザー) 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、そのユーザー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、ほかのユーザー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ー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. サーバーを開き、そのサーバー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、アクセスルールのリストに対する追加や編集を行い、ユーザー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、ほか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、『Managing Servers with iPlanet Console』のサーバー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については、『Managing Servers with iPlanet Console』のサーバー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

この節では、以下の項目に分けて説明しています。

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ー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ー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。不特定多数宛メールを制御するための具体的な方法については、第 17 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、702 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ー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、704 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

フィルタ文には、サービス情報とクライアント情報の両方が含まれています。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、ユーザー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については、699 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ー (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ー名 (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 のユーザー 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ー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、700 ページの「サーバーホストの指定」および 701 ページの「クライアントのユーザー名の指定」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ほかのパターンの使用方法については、699 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。

次の構文も使用できます。

「+」または「-」*serviceList*:*$*next_rule*

+ (許可フィルタ) は、デーモンリストサービスがクライアントリストに付与されることを意味します。

- (拒否フィルタ) は、クライアントリストに対してサービスが拒否されることを意味します。

* (ワイルドカードフィルタ) は、すべてのクライアントにこれらのサービスの使用を許可します。

$ は、ルールの区切りです。

次の例では、すべてのクライアントで複数のサービスを有効にしています。

```
+imap,pop,http:*
```

次の例では、複数のルールを単純化して各ルールが単一のサーバー名を所有し、クライアントリスト用のワイルドカードを使用するようにします。これは、LDIF ファイルでアクセス制御を指定するためのもっとも一般的な方法です。

```
+imap:ALL$+pop:ALL$+http:ALL
```

次の例では、あるユーザーに対してすべてのサービスを許可しない方法を示します。

```
-imap:*$-pop:*$-http:*
```

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、またはユーザー名を表すことができます。

表 19-3 サービスフィルタのワイルドカード名

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します。
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) に一致します。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ません。
UNKNOWN	名前が不明なすべてのユーザー、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します。 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ります。 一時的な DNS サーバーの問題により、ホスト名が使用できなくなる場合があります。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します。 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ません。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します。
KNOWN	名前が認識されているすべてのユーザー、または名前およびアドレスが認識されているすべてのホストに一致します。 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ります。 一時的な DNS サーバーの問題により、ホスト名が使用できなくなる場合があります。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ません。 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ません。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ません。
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します。

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*)

サーバーホストの指定

serviceSpec エントリにサーバー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ルールを持つ複数のドメインを 1 つのサーバー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ー名の指定

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ー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ー名 (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ー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ー名を信頼することができません。一般的に、特定のユーザー名を使用せずに、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 Sun Java System にお知らせください。
- ユーザー名の検索は時間がかかるので、すべてのユーザー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。ユーザー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ーは、ユーザー名の検索を実行せずに一致します。ただし、ほかのすべてのシステムについては、ユーザー名の検索が実行されます。

ユーザー名検索の機能は、クライアントホスト上の承認されていないユーザー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、ユーザー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
```

最初のルールは、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 番目のルールでは、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 つのサーバー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。ほか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 17 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. 「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。

   a. 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。「許可フィルタ」ウィンドウまたは「拒否フィルタ」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、「了解」をクリックします。

   b. 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。「許可フィルタ」ウィンドウまたは「拒否フィルタ」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、「了解」をクリックします。

   c. 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、697 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、702 ページの「フィルタの例」を参照してください。

コマンド行 : 次のように、コマンド行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、697 ページの「フィルタの構文」で説明した構文ルール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、697 ページの「フィルタの構文」で説明した構文ルール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については、580 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ー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、ユーザー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ほかのサービスが、ユーザーを認証して、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、エンドユーザー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ー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ーを設定する場合は、Sun Java System 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。ログインサーバーの設定方法や 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ーの設定方法や、認証 SDK の使用方法については、Sun Java System 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. 「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。

   「許可フィルタ」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、「了解」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。

   「許可フィルタ」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、「了解」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、697 ページの「フィルタの構文」を参照してください。

コマンド行 : 次のように、コマンド行を使用して、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、697 ページの「フィルタの構文」で説明した構文ルールに従って指定します。

POP before SMTP を有効にする

SMTP リレーサーバー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、認証済みのユーザーだけに 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ーに指示します。POP before SMTP をサイトに導入すると、ユーザー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、エンドユーザー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。

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のメッセージを許可する必要があることを SMTP サーバーに通知します。

SMTP プロキシをインストールするには

1. 『Sun Java Enterprise System インストールガイド』の説明に従って、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

   msg_svr_base/lib/SmtpProxyAService@25|587

   msg_svr_base/config/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 4 つの新しいファイルが作成されます。そのファイルを次に示します。

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 リレーサーバー上で、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 サーバーに接続する際に、実際の IP アドレスとその他の接続情報を SMTP サーバーに通知する必要があります。この情報により、SMTP サーバーは、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 サーバー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. MMP が SMTP サーバーに接続するために使用する IP アドレスが INTERNAL_IP マッピングテーブルによって「internal」として扱われていないことを確認します。

   INTERNAL_IP マッピングファイルについては、第 17 章「メールのフィルタリングとアクセス制御」の 550 ページの「SMTP リレーを追加するには」を参照してください。

5. 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. *msg_svr_base*/config/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。157 ページの第 7 章「マルチプレクササービスの設定および管理」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、『Sun Java System Messaging Server Administration Reference』(http://docs.sun.com/doc/819-0106) の「Encryption (SSL) Option」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 リレーサーバーホスト名 (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ーは、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、デフォルトはありません。
      例 : default:SmtpRelays manatee:485 gonzo mothra

      SmtpProxyPassword。SMTP リレーサーバー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 サーバー上の 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。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。デフォルトは空で、POP before SMTP を有効にするユーザーに対する通常の設定は `tcp_intranet` です。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 (690 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 リレーサーバーで検索を行う必要がなくなります。このオプションは、ホストしている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 (690 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 認証後にユーザー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 17 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。

SSL を使用したユーザーまたはグループディレクトリの検索

MTA、MMP、および IMAP/POP/HTTP の各サービスに対して、SSL によるユーザーディレクトリまたはグループディレクトリの検索を行うことができます。この機能を使用するためには、Messaging Server を SSL モードで構成しておく必要があります。この機能を有効にするには、configutil パラメータの `local.service.pab.ldapport` を `636` に設定し、`local.ugldapport` を `636` に設定し、`local.ugldapusessl` を `1` に設定します。

第 20 章

Communications Express メールでの S/MIME の管理

Secure/Multipurpose Internet Mail Extension (S/MIME) は、Sun Java System Communications Express メールで利用できます。S/MIME を使用するように設定された Communications Express メールユーザーは、Communications Express メール、Microsoft Outlook Express、および Mozilla メールシステムのほかのユーザーと署名付きまたは暗号化されたメッセージを交換できます。

Communications Express メールでの S/MIME の使用方法についての情報は、オンラインヘルプにあります。S/MIME を使用するための情報について、この章で説明します。この章には、以下の節があります。

S/MIME とは

S/MIME は、Communications Express メールユーザーに次の機能を提供します。

- 送信するメールメッセージのデジタル署名を作成して、メッセージが改ざんされておらず、送信者からのものであることを受信者に保証する
- 送信するメールメッセージを暗号化して、メッセージが受信者のメールボックスに届く前に、表示、変更、またはメッセージの内容が使用されることがないように防止する
- 証明書失効リスト (Certificate Revocation List、CRL) を使用する処理で、着信する署名付きメッセージのデジタル署名を確認する
- 受信者がメッセージの内容を読むことができるように、着信した暗号化されたメッセージを自動的に復号化する
- Communications Express メールおよび Mozilla メールシステムなど、S/MIME に準拠したクライアントのほかのユーザーと署名付きまたは暗号化されたメッセージを交換する

理解する必要がある概念

S/MIME を適切に管理するには、次の概念を理解している必要があります。

- プラットフォームの基本的な管理手順
- Lightweight Directory Access Protocol (LDAP) ディレクトリの構造および使用
- LDAP ディレクトリのエントリの追加または変更
- Sun Java System Directory Server の設定プロセス
- 次の概念と目的
  - 通信回線の保護のための Secure Socket Layer (SSL)
  - デジタル署名された電子メールメッセージ
  - 暗号化された電子メールメッセージ
  - ブラウザのローカルキーストア
  - スマートカード、スマートカードを使用するためのスマートカードソフトウェアとハードウェア
  - 非公開キーと公開キーのペアとそれらの証明書
  - 認証局 (CA)
  - キーと証明書の確認

- 証明書失効リスト (Certificate revocation list、CRL)。証明書失効リストについては、741 ページの「証明書が CRL でチェックされるタイミング」参照してください。

必要なソフトウェアおよびハードウェアコンポーネント

この節では、Communications Express メールで S/MIME を使用するために必要なハードウェアおよびソフトウェアについて説明します。すべての正しいバージョンのソフトウェアをサーバーおよびクライアントマシンにインストールしてから、S/MIME の設定をするようにしてください。

表 20-1 は、Communications Express メールがアクセスされるクライアントマシンに必要なソフトウェアおよびハードウェアを示しています。

表 20-1 クライアントマシンに必要なハードウェアおよびソフトウェア

コンポーネント	説明
オペレーティングシステム	• Microsoft Windows 98、2000、または XP
ブラウザ	• Microsoft Windows で動作する Microsoft Internet Explorer、バージョン 6 SP2 • Microsoft Windows 2000 および Microsoft Windows 98 で動作する Microsoft Internet Explorer、バージョン 6 SP1 (2004 年 12 月 1 日現在の最新のパッチを適用済みのもの)
Sun ソフトウェア	Sun Java 2 Runtime Environment、Standard Edition、バージョン 1.4.2_03 以降 (ただし 1.5 を除く)
証明書付きの非公開キーと公開キー	証明書付きの非公開キーと公開キーの 1 つ以上のペア。証明書は必須であり、標準の X.509 v3 形式である必要があります。S/MIME の機能を使用する各 Communications Express メールユーザー用のキーおよび証明書を CA から入手します。キーとその証明書はクライアントマシンまたはスマートカードに保存されます。公開キーと証明書は、Directory Server がアクセスできる LDAP ディレクトリにも保存されます。 キーが有効であることをさらに保証するためにキーの証明書を証明書失効リスト (CRL) で確認する場合、CA によって管理される CRL は、使用しているシステムに組み込まれている必要があります。741 ページの「証明書が CRL でチェックされるタイミング」を参照してください。
スマートカードソフトウェア (キーおよび証明書がスマートカードに保存される場合にのみ必要)	• ActivCard Gold、バージョン 2.1 または 3.0 • NetSign、バージョン 3.1

表 20-1　クライアントマシンに必要なハードウェアおよびソフトウェア (続き)

コンポーネント	説明
スマートカードリーダー	クライアントマシンおよびスマートカードソフトウェアがサポートする任意のモデルのスマートカード読み取り装置

表 20-2 は、サーバーマシンに必要な Sun Microsystems ソフトウェアを示しています。

表 20-2　サーバーマシンに必要なソフトウェア

Sun コンポーネント	説明
メールサーバー	Solaris、バージョン 8 または 9、および Sun SPARC マシン上の Sun Java System Messaging Server 6 2005Q1 リリース
LDAP サーバー	Sun Java System Directory Server 5 2004Q2 以降
Java	Java 2 Runtime Environment、Standard Edition、バージョン 1.4.2 以降
Access Manager	(Schema 2 での配備の場合) - Sun Java System Access Manager 6 2005Q1 および Communications Express - Sun Java System Communications Express 6 2005Q1

S/MIME を使用するための要件

Messaging Server のインストール後、Communications Express メールユーザーは署名および暗号化機能をただちに使用できるわけではありません。ユーザーが S/MIME を利用できるようになる前に、この節で説明する要件を満たしている必要があります。

非公開キーと公開キー

標準の X.509 v3 形式の証明書を含む、非公開キーと公開キーのペアを少なくとも 1 つ、S/MIME を使用する各 Communications Express メールユーザーに発行する必要があります。確認プロセスで使用される証明書は、キーが本当に使用者のものであることをほかのメールユーザーに対して保証します。1 人のユーザーに、複数のキーペアと関連する証明書を割り当てることができます。

キーおよびキーの証明書は、ユーザーの組織で発行するか、またはサードパーティベンダーから購入できます。キーおよび証明書の発行方法に無関係に、発行する組織は認証局 (CA) と呼ばれます。

キーのペアとキーの証明書は、次の 2 とおりの方法で保存されます。

- スマートカードと呼ばれる、Common Access Card (CAC) に保存

スマートカードは、商用クレジットカードに似ていて、クレジットカードと同様にメールユーザーが使用および管理する必要があります。スマートカードには、非公開キー情報を読むための、メールユーザーのコンピュータ (クライアントマシン) に接続された特別なカードリーダーが必要になります。詳細については、715 ページの「スマートカードに保存されたキー」を参照してください。

- メールユーザーのコンピュータ (クライアントマシン) のローカルキーストアに保存

  メールユーザーのブラウザがキーストアを提供します。ブラウザは、キーのペアおよび証明書をキーストアにダウンロードするためのコマンドも提供します。詳細については、715 ページの「クライアントマシンに保存されたキー」を参照してください。

スマートカードに保存されたキー

非公開キーと公開キーのペアが証明書とともに、スマートカードに保存される場合は、メールユーザーのコンピュータにカードリーダーが適切に接続されている必要があります。カード読み取り装置にはソフトウェアも必要であり、装置およびそのソフトウェアは装置の購入先のベンダーから供給されます。

適切にインストールすると、メールユーザーは、送信するメッセージ用のデジタル署名を作成するときに、スマートカードを読み取り装置に挿入します。スマートカードのパスワードの確認後、Communications Express メールがメッセージに署名するために非公開キーにアクセスできるようになります。サポートされているスマートカードと読み取り装置については、713 ページの「必要なソフトウェアおよびハードウェアコンポーネント」を参照してください。

スマートカードのベンダーからのライブラリが、ユーザーのコンピュータに必要です。詳細については、738 ページの「クライアントマシン用のキーアクセスライブラリ」を参照してください。

クライアントマシンに保存されたキー

キーのペアおよび証明書をスマートカードに保存しない場合は、メールユーザーのコンピュータ (クライアントマシン) のローカルキーストアに保存する必要があります。クライアントマシンのブラウザがキーストアを提供し、キーのペアおよび証明書をキーストアにダウンロードするためのコマンドも提供します。キーストアはパスワードで保護される場合がありますが、これはブラウザによって異なります。

ローカルキーストアをサポートするには、ユーザーのコンピュータにはブラウザのベンダーからのライブラリが必要です。詳細については、738 ページの「クライアントマシン用のキーアクセスライブラリ」を参照してください。

LDAP ディレクトリでの公開キーの公開

すべての公開キーおよび証明書は、Sun Java System Directory Server によってアクセス可能な、LDAP ディレクトリに保存する必要もあります。この行為は、S/MIME メッセージを作成するほかのメールユーザーが利用できるようにするための、公開キーの公開と呼ばれます。

送信者および受信者の公開キーは、暗号化されたメッセージの暗号化および復号化の処理で使用されます。公開キーの証明書は、デジタル署名に使用された非公開キーの検証に使用されます。

公開キーおよび証明書を公開するために `ldapmodify` を使用する方法については、748 ページの「証明書の管理」を参照してください。

メールユーザーに S/MIME の使用を許可する

署名付きまたは暗号化されたメッセージを作成するには、正当な Communications Express メールユーザーが作成許可を持っている必要があります。このためには、ユーザーの LDAP エントリに `mailAllowedServiceAccess` または `mailDomainAllowedServiceAccess` LDAP 属性を使用します。これらの属性を使用して、個人またはドメインベースでメールユーザーに S/MIME の使用を許可または拒否できます。

詳細については、747 ページの「S/MIME 機能の使用を許可する」を参照してください。

複数言語のサポート

メールメッセージに英語のみを使用する Communications Express メールユーザーは、中国語など Latin 以外の言語の文字を含む S/MIME メッセージを読めない場合があります。この理由の 1 つは、そのユーザーのマシンにインストールされた Java 2 Runtime Environment (JRE) には `/lib` ディレクトリに `charsets.jar` ファイルがないためです。

デフォルトの JRE インストールプロセスを使用して英語版の JRE をダウンロードした場合、`charsets.jar` ファイルはインストールされません。ただし、デフォルトインストールのすべてのその他の言語を選択した場合は、`charsets.jar` がインストールされます。

charsets.jar ファイルが /lib ディレクトリにインストールされるようにするには、カスタムインストールを使用して JRE の英語版をインストールするようにユーザーに警告します。インストールプロセス時に、ユーザーはその他の言語のサポートオプションを選択する必要があります。

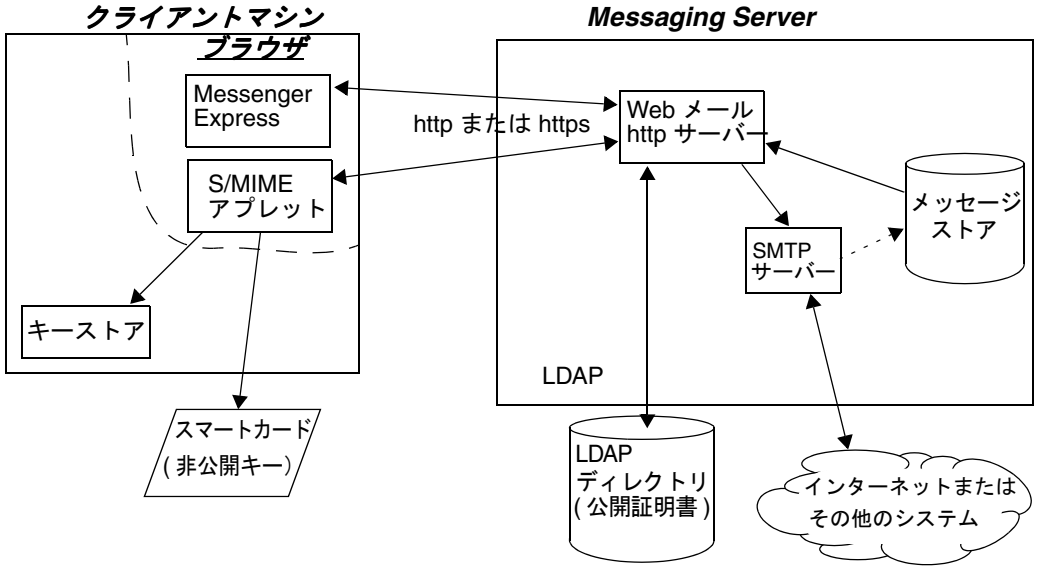
Messaging Server のインストール後の作業

この節では、S/MIME アプレットについて説明し、Communications Express メール用に S/MIME を設定するための基本設定手順を示します。設定手順では、S/MIME アプレットのパラメータと Messaging Server のオプションを設定します。

S/MIME アプレット

メッセージの署名、メッセージの暗号化、またはメッセージの復号化の処理は、非公開キーおよび公開キーの確認のためのさまざまな処理とともに、S/MIME アプレットと呼ばれる特別なアプレットで処理されます。S/MIME 機能の設定は、smime.conf ファイルに含まれるパラメータと Messaging Server のオプションを使って行います。図 20-1 に、S/MIME アプレットとほかのシステムコンポーネントとの関係を示します。

図 20-1　S/MIME アプレット



初めてのログイン

S/MIME の使用を許可された Communications Express メールユーザーが初めて Messaging Server にログインするときには、S/MIME アプレットについての一連の特別なプロンプトが表示されます。プロンプトに対して「はい」または「常に」で答えると、S/MIME アプレットがコンピュータにダウンロードされます。アプレットは、ユーザーが Communications Express メールからログアウトするまでマシンに残ります。

詳細は、748 ページの「証明書の管理」を参照してください。

S/MIME アプレットのダウンロード

ユーザーのマシンで Java 2 Runtime Environment (JRE) に対してキャッシングが有効になっていない限り、ユーザーが Communications Express メールにログインするたびに S/MIME アプレットがダウンロードされます。キャッシングが有効な場合、初期のダウンロード後にユーザーのマシンに S/MIME アプレットのコピーが保存されるので、ユーザーがログインするたびに S/MIME アプレットをダウンロードしなくてすむようになります。

キャッシングはパフォーマンスを向上させるので、Java 2 Runtime Environment、バージョン 1.4.*x* のキャッシングを有効にするため、次の手順を実行するようにユーザーに指示できます。

1. Microsoft Windows の「コントロールパネル」に移動します。
2. Java Plug-in アイコン (Java 2 Runtime Environment) をダブルクリックします。
3. 「キャッシュ」タブをクリックします。
4. 「キャッシュを有効」チェックボックスにチェックマークを付けます。
5. 「適用」をクリックします。

ダウンロード後は、ユーザーは S/MIME アプレットを意識することはありません。メッセージの署名、暗号化、または復号化は、Communications Express メールによって行われるようにみえます。エラーメッセージがポップアップ表示されないかぎり、ユーザーは非公開キーまたは公開キーを確認するプロセスにも気がつきません。詳細は、739 ページの「非公開キーと公開キーの確認」を参照してください。

基本的な S/MIME の設定

S/MIME の設定ファイルの `smime.conf` には、各パラメータの説明と例が含まれています。`smime.conf` ファイルは Messaging Server とともに組み込まれ、*msg-svr-base*`/config/` ディレクトリに存在します。この *msg-svr-base* は、Messaging Server がインストールされるディレクトリです。

次の手順は、S/MIME の機能を設定するために最小限必要な手順です。

1. Messaging Server のインストール後、Communications Express メールの基本的な機能が有効であることを確認します。
2. まだの場合は、S/MIME の機能を使用することを許可されているすべてのメールユーザーのために、標準の X.509 v3 形式の証明書付きの非公開鍵と公開鍵のペアを作成または入手します。
3. スマートカードをキーおよび証明書に使用する場合は、次のようにします。
   a. スマートカードをメールユーザーに配布します。
   b. Communications Express メールがアクセスされる、各クライアントマシンにスマートカードの読み取り装置とソフトウェアが適切にインストールされているようにします。
4. ブラウザのローカルキーストアを使用してキーおよび証明書を保存する場合は、キーのペアと証明書をローカルキーストアにダウンロードする方法をメールユーザーに指示します。
5. スマートカードまたはローカルキーストアをサポートするための正しいライブラリがクライアントマシンに存在するようにします。詳細については、738 ページの「クライアントマシン用のキーアクセスライブラリ」を参照してください。
6. S/MIME をサポートするように LDAP ディレクトリを設定します。

a. Directory Server によってアクセス可能な LDAP ディレクトリに CA のすべての証明書を、認証局の識別名で保存します。それらの証明書の LDAP 属性は、cacertificate;binary です。証明書を保存するディレクトリの情報を書き留めます。この手順のあとのほうでこの情報が必要になります。

   LDAP ディレクトリ情報の指定例については 734 ページの「trustedurl」を、LDAP ディレクトリの検索については 748 ページの「証明書の管理」を参照してください。

b. Directory Server によってアクセス可能な LDAP ディレクトリに公開キーと証明書を保存します。公開キーと証明書の LDAP 属性は、usercertificate;binary です。公開キーと証明書を保存するディレクトリの情報を書き留めます。この手順のあとのほうでこの情報が必要になります。

   LDAP ディレクトリ情報の指定例については 727 ページの「certurl」を、LDAP ディレクトリの検索については 748 ページの「証明書の管理」を参照してください。

c. S/MIME メッセージを送受信するすべてのユーザーは、ユーザーエントリで LDAP フィルタにより S/MIME を使用することを許可されるようにします。mailAllowedServiceAccess または mailDomainAllowedServiceAccess LDAP 属性でフィルタを定義します。

   注 : デフォルトでは、mailAllowedServiceAccess または mailDomainAllowedServiceAccess を使用しない場合、smime を含むすべてのサービスが利用可能です。これらの属性でサービスを明示的に指定する場合は、サービスの http および smtp、また smime を指定して、メールユーザーに S/MIME 機能を使用する許可を与える必要があります。

   詳細については、747 ページの「S/MIME 機能の使用を許可する」を参照してください。

7. 使用可能なテキストエディタで smime.conf ファイルを編集します。パラメータの構文についてはファイルの始めにあるコメントを参照してください。

   smime.conf 内のすべてのテキストおよび例のパラメータの前には、コメント文字 (#) がついています。必要なパラメータを smime.conf に追加するか、パラメータの例をファイルの別の部分にコピーしてその値を変更できます。例をコピーして編集する場合は、その行の先頭の # 文字を必ず削除してください。

   次のパラメータをファイルに追加する場合、各パラメータを別々の行に追加します。

   a. trustedurl -- CA の証明書の場所を特定するための LDAP ディレクトリ情報を設定します。手順 6 の手順 a で書き留めておいた情報を使用します。

   b. certurl -- 公開キーと証明書の場所を特定するための LDAP ディレクトリ情報を設定します。手順 6 の手順 b で書き留めておいた情報を使用します。

c. usercertfilter -- smime.conf ファイルに含まれる例の値を設定します。例の値は、ほとんどの場合、必要なフィルタです。例をコピーし、行の先頭の # 文字を削除します。

このパラメータは、Communications Express メールユーザーのプライマリ、代替、および同等の電子メールアドレスのフィルタ定義を指定し、キーのペアが異なるメールアドレスに割り当てられるときにユーザーの非公開キーと公開キーのペアのすべてが見つかるようにします。

d. sslrootcacertsurl -- S/MIME アプレットと Messaging Server 間の通信リンクに SSL を使用する場合、sslrootcacertsurl に Messaging Server の SSL 証明書の確認に使用する CA の証明書の場所を特定するための LDAP ディレクトリ情報を設定します。詳細については、736 ページの「SSL でインターネットリンクを保護する」を参照してください。

checkoverssl -- S/MIME アプレットと Messaging Server 間の通信リンクに SSL を使用しない場合は、0 に設定します。

e. crlenable -- CRL チェックを行うと、smime.conf ファイルにほかのパラメータを追加する必要がある場合があるため、一時的に CRL チェックを無効にするには 0 に設定します。

f. logindn および loginpw -- 公開キーと CA 証明書が含まれる LDAP ディレクトリにアクセスするための認証が必要な場合は、これらのパラメータには読み取り権限を持つ LDAP エントリの識別名とパスワードを設定します。

注 : certurl、crlmappingurl、sslrootcacertsurl、または trustedurl パラメータによって指定された LDAP 情報で LDAP ディレクトリがアクセスされるたびに、logindn および loginpw の値が使用されます。詳細については、723 ページの「公開キー、CA 証明書、および CRL にアクセスするための、資格情報を使用した LDAP へのアクセス」を参照してください。

認証で LDAP ディレクトリにアクセスする必要がない場合は、logindn および loginpw を設定してはなりません。

8. 次のように configutil で Messaging Server オプションを設定します。

a. local.webmail.smime.enable -- 1 に設定します。

b. local.webmail.cert.enable -- 証明書を CRL でチェックする場合は、1 に設定します。

詳細については、734 ページの「Messaging Server オプション」を参照してください。

9. これで Communications Express メールが S/MIME 機能対応に設定されました。次の手順に従って S/MIME 機能が有効であることを確認します。

a. Messaging Server を再起動します。

b. S/MIME に関連する診断メッセージを、Messaging Server ログファイルの *msg-svr-base*/log/http で確認します。

c. S/MIME の問題が検出された場合は、設定パラメータでどのように問題を修正すべきか判断するのに診断メッセージが役立ちます。

d. 必要な設定パラメータを訂正します。

e. Messaging Server のログファイルに S/MIME の診断メッセージがなくなるまで手順 a. ～ d. を繰り返します。

f. 次の手順に従って S/MIME 機能が有効であることを確認します。

I. クライアントマシンから Messaging Server にログインします。S/MIME アプレット用の特別なプロンプトに対して「はい」または「常に」で答えます。詳細については、748 ページの「証明書の管理」を参照してください。

II. 自分宛の短いメッセージを作成します。

III. 「作成」ウィンドウの下部の「暗号化」チェックボックスのチェックマークがまだ付いていない場合は付けて、メッセージを暗号化します。

IV. 「送信」をクリックして、暗号化したメッセージを自分自身に送信します。これは、キーおよび証明書のほとんどのメカニズムを実際に使用します。

V. 暗号化されたメッセージの問題が検出された場合、もっともよくある原因は smime.conf ファイル内の LDAP ディレクトリに使用した値や LDAP ディレクトリへのキーおよび証明書の保存方法に関するものです。診断メッセージの詳細については、Messaging Server ログを確認してください。

表 20-3 に要約された残りの S/MIME パラメータは、S/MIME 環境をさらに設定するために使用できる多くのオプションを提供します。パラメータについては、725 ページの「smime.conf ファイルのパラメータ」を参照してください。

表 20-3 smime.conf パラメータの要約

S/MIME の必須パラメータ	スマートカードおよびローカルキーストアのパラメータ	CRL チェックのパラメータ	初期設定とセキュリティ保護されたリンクのパラメータ
certurl*	platformwin	checkoverssl	alwaysencrypt
logindn		crlaccessfail	alwayssign
loginpw		crldir	sslrootcacertsurl
trustedurl*		crlenable	
usercertfilter*		crlmappingurl	
		crlurllogindn	

表 20-3　smime.conf パラメータの要約 (続き)

S/MIME の必須パラメータ	スマートカードおよびローカルキーストアのパラメータ	CRL チェックのパラメータ	初期設定とセキュリティ保護されたリンクのパラメータ
		crlurlloginpw	
		crlusepastnextupdate	
		readsigncert	
		revocationunknown	
		sendencryptcert	
		sendencryptcertrevoked	
		readsigncert	
		sendsigncertrevoked	
		timestampdelta	

* これらのパラメータにはデフォルト値がないので、値を指定する必要があります。

公開キー、CA 証明書、および CRL にアクセスするための、資格情報を使用した LDAP へのアクセス

S/MIME に必要な公開キー、CA 証明書、および CRL は、LDAP ディレクトリに保存されます (前の節を参照)。キー、証明書、および CRL には、LDAP の 1 つの URL または複数の URL からアクセスできます。たとえば、CRL を 1 つの URL に、公開キーと証明書を別の URL に保存できます。Messaging Server では、必要な CRL または証明書情報をどの URL に含めるか、またそれらの URL へアクセスできるエントリの DN およびパスワードも指定できます。それらの DN/ パスワードの資格情報はオプションです。いずれも指定しない場合、LDAP はまず HTTP サーバーの資格情報でアクセスを試み、それが失敗した場合は、anonymous でのアクセスを試みます。

次の smime.conf の資格情報パラメータの 2 つのペアを設定して、必要な URL にアクセスできます。logindn と loginpw、および crlurllogindn と crlurlloginpw。

logindn と loginpw は、smime.conf に含まれるすべての URL に使用される資格情報です。certurl および trustedurl パラメータによって指定された公開キー、公開キーの証明書、および CA 証明書の読み取り権限がある LDAP エントリの DN およびパスワードを指定します。

crlurllogindn および crlurlloginpw は、マッピングテーブルから得られる URL に対する読み取り権限がある LDAP エントリの DN およびパスワードを指定します (詳細は、742 ページの「CRL へのアクセス」を参照)。それらの資格情報が受け入れられない場合、LDAP アクセスは拒否され、その他の資格情報での再試行は行われません。パラメータは両方とも指定するか、または両方とも空である必要があります。これらのパラメータは、証明書から直接得られる URL には適用されません。

特定の URL のパスワードの設定

Messaging Server では、次の smime.conf URL にアクセスするための DN とパスワードのペアを具体的に定義することができます。certUrl、trustedUrl, crlmappingUrl、sslrootcacertsUrl。

構文は次のとおりです。

url_type　　*URL*[|*URL_DN* | *URL_password*]

次に例を示します。

```
trustedurl==ldap://mail.siroe.com:389/cn=Directory Manager,
ou=people,
o=siroe.com,o=ugroot?cacertificate?sub?(objectclass=certificationau
thority) | cn=Directory manager | boomshakalaka
```

LDAP 資格情報の使用の要約

この節では、LDAP 資格情報の使用について簡単に説明します。

- すべての LDAP 資格情報はオプションです。いずれも指定しない場合、LDAP はまず HTTP サーバーの資格情報でアクセスを試み、それが失敗した場合は、anonymous でのアクセスを試みます。

  次のように、指定できる 2 組の URL に対する資格情報として smime.conf パラメータの 2 つのペアが使用されます。

  logindn & loginpw - smime.conf 内のすべての URL

  crlurllogindn & crlurlloginpw - マッピングテーブルからのすべての URL

  これらは、デフォルトの LDAP 資格情報ペアと呼ばれます。

- smime.conf に、または対応する CRL URL を介して指定された URL には、オプションのローカル LDAP 資格情報ペアが指定されます。
- 資格情報は、それぞれ次のように指定された順序でチェックされます。

  1) ローカル LDAP 資格情報ペア - 指定された場合、1 つのみが試みられます

  2) デフォルトの LDAP 資格情報ペア - 指定された際に、ローカル LDAP 資格情報ペアがない場合は、1 つのみが試みられます

3) サーバー - ローカル LDAP 資格情報ペアもデフォルトの LDAP 資格情報ペアも指定されない場合、最初に試みられます

4) anonymous - サーバーに障害が発生するか、またはいずれも指定されない場合にのみ、最後に試みられます

- URL にローカル LDAP 資格情報のペアが指定された場合、それがまず使用され、アクセスが失敗した場合は、アクセスが拒否されます。
- URL にローカル LDAP 資格情報ペアが指定されない場合、対応するデフォルトの LDAP 資格情報ペアが使用されます。アクセスが失敗した場合は、アクセスが拒否されます。

smime.conf ファイルのパラメータ

smime.conf ファイルは Messaging Server とともに組み込まれ、*msg-svr-base*/config/ ディレクトリに存在します。この *msg-svr-base* は、Messaging Server がインストールされるディレクトリです。ファイルに含まれるすべてのテキストおよびパラメータの例の前には、コメント文字 (#) がついています。

値を指定したパラメータを smime.conf ファイルに追加するか、またはパラメータの例を編集できます。例を使用する場合、例をファイルの別の部分にコピーし、パラメータの値を編集し、行の先頭の # 文字を削除します。

Messaging Server をインストールしたあとに、使用可能なテキストエディタで smime.conf を編集します。表 20-4 に説明があるパラメータには、大文字と小文字の区別がなく、特に指定されていないかぎり、設定する必要はありません。

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ

パラメータ	目的
alwaysencrypt	S/MIME を使用する許可が与えられているすべての Communications Express メールユーザーのためにすべての送信メッセージを自動的に暗号化するかどうかの初期設定を制御します。各 Communications Express メールユーザーは、755 ページの表 20-6 に説明があるチェックボックスを使用して、各自のメッセージに対してこのパラメータの値を無効にできます。 次の値のいずれかを選択します。 0 - メッセージを暗号化しません。Communications Express メールでは暗号化チェックボックスのチェックマークが付いていない状態で表示されます。これがデフォルトです。 1 - メッセージを常に暗号化します。Communications Express メールでは暗号化チェックボックスのチェックマークが付いた状態で表示されます。 例 : alwaysencrypt==1
alwayssign	S/MIME を使用する許可が与えられているすべての Communications Express メールユーザーのためにすべての送信メッセージに自動的に署名を付けるかどうかの初期設定を制御します。各 Communications Express メールユーザーは、755 ページの表 20-6 に説明があるチェックボックスを使用して、各自のメッセージに対してこのパラメータの値を無効にできます。 次の値のいずれかを選択します。 0 - メッセージに署名しません。Communications Express メールでは署名チェックボックスのチェックマークが付いていない状態で表示されます。これがデフォルトです。 1 - メッセージに常に署名します。Communications Express メールでは署名チェックボックスのチェックマークが付いた状態で表示されます。 例 : alwaysensign==1

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ (続き)

パラメータ	目的
certurl	Communications Express メールユーザーの公開キーおよび証明書の場所を特定するための LDAP ディレクトリ情報を指定します。公開キーの LDAP 属性は、usercertificate;binary です。証明書については、748 ページの「証明書の管理」を参照してください。 このパラメータは、Messaging Server がサービスを提供するすべてのユーザーが含まれる LDAP ディレクトリ情報ツリーのユーザー / グループの最上位のノードを指す必要があります。これは、複数のドメインがあるサイトでは特に重要になります。識別名は、1 つのドメインのユーザーが含まれるサブツリーではなく、ユーザー / グループツリーのルートの識別名である必要があります。 このパラメータの設定は必須です。 例 : certurl==ldap://mail.siroe.com:389/ou=people, o=siroe.com, o=ugroot
checkoverssl	キーの証明書を CRL でチェックするときに SSL 通信リンクを使用するかどうかを制御します。詳細については、736 ページの「SSL でインターネットリンクを保護する」を参照してください。 次の値のいずれかを選択します。 0 - SSL 通信リンクを使用しません。 1 - SSL 通信リンクを使用します。これがデフォルトです。 CRL チェックが有効な場合に、プロキシサーバーを使用すると、問題が発生する場合があります。詳細については、743 ページの「プロキシサーバーと CRL チェック」を参照してください。

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ (続き)

パラメータ	目的
crlaccessfail	CRL へのアクセスを複数回試行し、アクセスに失敗したあとに、Messaging Server が再びアクセスを試みるまでの待機時間を指定します。このパラメータにはデフォルト値がありません。 **構文 :** crlaccessfail==*number_of_failures*:*time_period_for_failures*:*wait_time_before_retry* ここで、 *number_of_failures* は、*time_period_for_failures* によって指定された時間間隔の間に Messaging Server が CRL へのアクセスを失敗できる回数です。値は、0 よりも大きくする必要があります。 *time_period_for_failures* は、Messaging Server が失敗した CRL へのアクセス試行を数える秒単位の期間です。値は、0 よりも大きくする必要があります。 *wait_time_before_retry* は、指定された時間間隔に、失敗した試行回数が限度に達したことを検出してから、Messaging Server が再度 CRL へのアクセスを試みる前に、待つ秒数です。値は、0 よりも大きくする必要があります。 例 : crlaccessfail==10:60:300 この例では、Messaging Server は CRL へのアクセスを 1 分以内に 10 回失敗します。その後、5 分待ってから、再び CRL へのアクセスを試みます。詳細については、746 ページの「CRL へのアクセスの問題」を参照してください。
crldir	Messaging Server が CRL をダウンロードするディスク上のディレクトリ情報を指定します。デフォルトは、*msg-svr-base*/data/store/mboxlist であり、ここの *msg-svr-base* は Messaging Server がインストールされたディレクトリです。詳細については、744 ページの「古い CRL の使用」を参照してください。

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ (続き)

パラメータ	目的
crlenable	証明書を CRL でチェックするかどうかを制御します。一致するものがある場合は、証明書は失効したとみなされます。smime.conf ファイルの send*revoked パラメータの値は、証明書が失効したキーを Communications Express メールが拒否するか、または使用するかどうかを決定します。詳細については、739 ページの「非公開キーと公開キーの確認」を参照してください。 次の値のいずれかを選択します。 0- 各証明書を CRL でチェックしません。 1- 各証明書を CRL でチェックします。これがデフォルトです。Messaging Server の local.webmail.cert.enable オプションを 1 に設定するようにします。そのようにしないと、crlenable が 1 に設定されても CRL チェックは行われません。
crlmappingurl	CRL マッピング定義の場所を特定するための LDAP ディレクトリ情報を指定します。このパラメータは、マッピング定義がある場合にのみ必要です。詳細については、742 ページの「CRL へのアクセス」を参照してください。このパラメータにはデフォルト値がありません。URL へアクセスするための DN およびパスワードを追加することもできます。 構文 : crlmappingurl　　*URL* **[** | *URL_DN* | *URL_password***]** 例 : crlmappingurl==ldap://mail.siroe.com:389/cn=XYZ Messaging, ou=people, o=mail.siroe.com,o=isp?msgCRLMappingRecord?sub?(objectclass=msgCRLMappingTable)|cn=Directory Manager | pAsSwOrD
crlurllogindn	CRL マッピング定義に対する読み取り権限がある LDAP エントリの識別名を指定します (エントリが証明書から直接のものでない場合の詳細は、742 ページの「CRL へのアクセス」を参照)。 crllogindn および crlloginpw の値を指定しない場合は、Messaging Server は LDAP ディレクトリにアクセスするために HTTP サーバーに対するログイン値を使用します。それが失敗した場合、Messaging Server は匿名で LDAP ディレクトリへのアクセスを試みます。 例 : crllogindn==cn=Directory Manager

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ (続き)

パラメータ	目的
crlurlloginpw	crllogindn パラメータの識別名に対するパスワードを ASCII テキストで指定します。 crllogindn および crlloginpw の値を指定しない場合、Messaging Server は LDAP ディレクトリにアクセスするために HTTP サーバーに対するログイン値を使用します。それが失敗した場合、Messaging Server は匿名で LDAP ディレクトリへのアクセスを試みます。 例 : crlloginpw==zippy
crlusepastnextupdate	現在の日付が CRL の次の更新日フィールドに指定された日付を過ぎたときに CRL を使用するかどうかを制御します。詳細については、744 ページの「古い CRL の使用」を参照してください。 次の値のいずれかを選択します。 0 - 古い CRL を使用しません。 1 - 古い CRL を使用します。これがデフォルトです。
logindn	certurl および trustedurl パラメータによって指定された LDAP ディレクトリにある公開キーおよび公開キーの証明書、また CA 証明書の読み取り権限がある LDAP エントリの識別名を指定します。 logindn および loginpw の値が指定されない場合は、Messaging Server は LDAP ディレクトリにアクセスするために HTTP サーバーに対するログイン値を使用します。それが失敗した場合、Messaging Server は匿名で LDAP ディレクトリへのアクセスを試みます。 例 : logindn==cn=Directory Manager
loginpw	logindn パラメータの識別名に対するパスワードを ASCII テキストで指定します。 logindn および loginpw の値が指定されない場合は、Messaging Server は LDAP ディレクトリにアクセスするために HTTP サーバーに対するログイン値を使用します。それが失敗した場合、Messaging Server は匿名で LDAP ディレクトリへのアクセスを試みます。 例 : loginpw==SkyKing

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ (続き)

パラメータ	目的
platformwin	Microsoft Windows プラットフォームでスマートカードまたはローカルキーストアを使用するときに必要なライブラリ名を 1 つ以上指定します。クライアントマシンでデフォルト値ではうまくいかない場合にのみこのパラメータを変更します。デフォルトは次のとおりです。 platformwin==CAPI:library=capibridge.dll; 詳細については、738 ページの「クライアントマシン用のキーアクセスライブラリ」を参照してください。
readsigncert	メッセージを読むときに S/MIME デジタル署名を確認するために、公開キーの証明書を CRL でチェックするかどうかを制御します。非公開キーを使用してメッセージのデジタル署名を作成しますが、非公開キーを CRL でチェックすることはできないので、非公開キーに関連付けられた公開キーの証明書が CRL でチェックされます。詳細については、739 ページの「非公開キーと公開キーの確認」を参照してください。 次の値のいずれかを選択します。 0- 証明書を CRL でチェックしません。 1- 証明書を CRL でチェックします。これがデフォルトです。
revocationunknown	証明書を CRL でチェックしたときに、あいまいなステータスが返された場合に実行するアクションを決定します。この場合、証明書のステータスが有効または失効のどちらであるかは確かでありません。詳細については、739 ページの「非公開キーと公開キーの確認」を参照してください。 次の値のいずれかを選択します。 ok - 証明書が有効であるとして扱います。 revoked - 証明書が失効しているとして扱います。これがデフォルトです。
sendencryptcert	送信メッセージの暗号化に使用する公開キーの証明書を使用する前に CRL でチェックするかどうかを制御します。詳細については、739 ページの「非公開キーと公開キーの確認」を参照してください。 次の値のいずれかを選択します。 0- 証明書を CRL でチェックしません。 1- 証明書を CRL でチェックします。これがデフォルトです。

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ (続き)

パラメータ	目的
sendencryptcertrevoked	送信メッセージの暗号化に使用した公開キーの証明書が失効した場合に実行するアクションを決定します。詳細については、739 ページの「非公開キーと公開キーの確認」を参照してください。 次の値のいずれかを選択します。 allow - 公開キーを使用します。 disallow - 公開キーを使用しません。これがデフォルトです。
sendsigncert	送信メッセージのデジタル署名の作成に非公開キーを使用できるかどうかを判断するために、公開キーの証明書を CRL でチェックするかどうかを制御します。デジタル署名に非公開キー使用をしますが、非公開キーを CRL でチェックすることはできないので、非公開キーに関連付けられた公開キーの証明書が CRL でチェックされます。詳細については、739 ページの「非公開キーと公開キーの確認」を参照してください。 次の値のいずれかを選択します。 0- 証明書を CRL でチェックしません。 1- 証明書を CRL でチェックします。これがデフォルトです。
sendsigncertrevoked	非公開キーのステータスが失効であると判断されたときに実行するアクションを決定します。非公開キーを使用してメッセージのデジタル署名を作成しますが、非公開キーを CRL でチェックすることはできないので、非公開キーに関連付けられた公開キーの証明書が CRL でチェックされます。公開キーの証明書が失効すると、対応する非公開キーも失効します。詳細については、739 ページの「非公開キーと公開キーの確認」を参照してください。 次の値のいずれかを選択します。 allow - 失効ステータスの非公開キーを使用します。 disallow - 失効ステータスの非公開キーを使用しません。これがデフォルトです。

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ (続き)

パラメータ	目的
sslrootcacertsurl	Messaging Server の SSL 証明書を確認するために使用される有効な CA の証明書の場所を特定するための識別名と LDAP ディレクトリ情報を指定します。Messaging Server で SSL が有効である場合には、これは必須のパラメータです。詳細については、736 ページの「SSL でインターネットリンクを保護する」を参照してください。 クライアントアプリケーションからすべての要求を受信するプロキシサーバーに対する SSL 証明書を持っている場合、それらの SSL 証明書の CA 証明書もこのパラメータが指す LDAP ディレクトリに存在する必要があります。 URL へアクセスするための DN およびパスワードを追加することもできます。 構文 : `crlmappingurl` *URL* **[** | *URL_DN* | *URL_password***]** 例 : `sslrootcacertsurl==ldap://mail.siroe.com:389/cn=SSL Root CA Certs, ou=people, o=siroe.com, o=isp?cacertificate;binary?base?(objectclass=certificationauthority)`|cn=Directory Manager | pAsSwOrD
timestampdelta	公開キーの証明書を CRL でチェックするときにメッセージの送信時間または受信時間のどちらを使用するかを決定するために使用する秒単位の時間間隔を指定します。 このパラメータのデフォルト値の 0 は、常に受信時間を使用するように Communications Express メールに指示します。詳細については、745 ページの「使用するメッセージ時刻の判断」を参照してください。 例 : `timestampdelta==360`

表 20-4　smime.conf ファイルの S/MIME 設定パラメータ (続き)

パラメータ	目的
trustedurl	有効な CA の証明書の場所を特定するための識別名と LDAP ディレクトリ情報を指定します。これは必須のパラメータです。 URL へアクセスするための DN およびパスワードを追加することもできます。 構文 : crlmappingurl　　*URL* **[** | *URL_DN* | *URL_password***]** 例 : trustedurl==ldap://mail.siroe.com:389/cn=Directory Manager, ou=people, o=siroe.com,o=ugroot?cacertificate?sub?(objectclass=certificationauthority)|cn=Directory Manager | pAsSwOrD
usercertfilter	Communications Express メールユーザーのプライマリ、代替、および同等の電子メールアドレスのフィルタ定義を指定し、キーのペアが異なるメールアドレスに割り当てられるときにユーザーの非公開キーと公開キーのペアのすべてが見つかるようにします。 このパラメータは必須であり、デフォルト値がありません。

Messaging Server オプション

S/MIME に適用する 3 つの Messaging Server オプションを設定するには、Messaging Server がインストールされたマシンで以下を実行します。

1. root としてログインします。次に、次のように入力します。

   # cd *msg-svr-base*/sbin

   msg-svr-base は Messaging Server がインストールされたディレクトリです。

2. 次の表に説明がある Messaging Server オプションをシステムの要件に応じて設定します。次のオプションを設定するには、configutil ユーティリティを使用します。別途記載がないかぎり、オプションを設定する必要はありません。

パラメータ	目的
local.webmail.cert.enable	CRL チェックを処理するプロセスが CRL チェックを実行すべきかどうかを制御します。 0- プロセスは証明書を CRL でチェックしません。これがデフォルトです。 1- プロセスは証明書を CRL でチェックします。1 に設定するときには、smime.conf ファイルの crlenable パラメータも 1 に設定します。
local.webmail.cert.port	CRL 通信に使用するために Messaging Server が実行されるマシンのポート番号を指定します。このポートは、そのマシンのみにローカルに使用されます。値は、1024 よりも大きくする必要があります。デフォルトは 55443 です。 デフォルトのポート番号がすでに使用されている場合は、これは必須のオプションです。
local.webmail.smime.enable	Communications Express メールユーザーが S/MIME 機能を使用できるかどうかを制御します。次の値のいずれかを選択します。 0 - システムが適切なソフトウェアおよびハードウェアコンポーネントで構成されている場合でも、Communications Express メールユーザーは S/MIME 機能を使用できません。これがデフォルトです。 1 - S/MIME 機能の使用を許可されている Communications Express メールユーザーが S/MIME 機能を使用できます。 例 : configutil -o local.webmail.smime.enable -v 1

SSL でインターネットリンクを保護する

次の表に要約されているように、Messaging Server では Communications Express メールに影響するインターネットリンクのために Secure Socket Layer (SSL) をサポートしています。

リンク	説明
Messaging Server と Communications Express メール間	このリンクを SSL で保護するには、Messaging Server の管理作業が必要です。Communications Express メールユーザーは、ブラウザで Messaging Server に対する URL 情報を入力するときに、HTTP ではなく、HTTPS プロトコルを使用する必要があります。 詳細については、736 ページの「Messaging Server と Communications Express メール間のリンクを保護する」を参照してください。
Messaging Server と S/MIME アプレットの間	公開キーの証明書を CRL でチェックするときには、S/MIME アプレットは Messaging Server と直接通信する必要があります。SSL でリンクを保護する場合は、`smime.conf` ファイルの `sslrootcacertsurl` および `checkoverssl` を設定することに加えて、Messaging Server の管理作業が必要になります。 詳細については、737 ページの「Messaging Server と S/MIME アプレット間のリンクを保護する」を参照してください。

Messaging Server と Communications Express メール間のリンクを保護する

Messaging Server は Messaging Server と Communications Express メール間のインターネットリンクに対する Secure Socket Layer (SSL) の使用をサポートします。Messaging Server を SSL 用に設定したら、Communications Express を SSL 用に設定します。詳細は、『Communications Express 管理ガイド』(`http://docs.sun.com/doc/819-1065?l=ja`) を参照してください。Communications Express メールユーザーは、次のようにブラウザに HTTP プロトコル (`HTTP://`*hostname*`.`*domain*`:`*unsecure_port*) ではなく、HTTPS プロトコルで Communications Express URL を指定します。

`HTTPS://`*hostname*`.`*domain*`:`*secured_port*

Communications Express のログインウィンドウが表示されると、リンクが保護されていることを示すロックアイコンがウィンドウの下部のロックされた位置に表示されます。

Messaging Server の SSL 設定情報については、679 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。また、『Communications Express 管理ガイド』(http://docs.sun.com/doc/819-1065?l=ja) も参照してください。

Messaging Server と S/MIME アプレット間のリンクを保護する

公開キーの証明書を CRL でチェックするときには、S/MIME アプレットは Messaging Server と直接通信する必要があります。SSL でこの通信リンクを保護するには、次の手順を実行します。

1. SSL に対応するように Messaging Server を設定する管理作業を実行します。679 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。
2. ルートの SSL CA 証明書の場所を特定するための情報を指定するために `smime.conf` ファイルの `sslrootcacertsurl` パラメータを設定します。それらの CA 証明書は、Messaging Server と S/MIME アプレット間に SSL リンクが確立されたときに Messaging Server の SSL 証明書を確認するために使用されます。
3. `smime.conf` ファイルの `checkoverssl` パラメータに `1` を設定します。この Messaging Server オプションは、Messaging Server と S/MIME アプレット間のリンクに SSL を使用するかどうかを決定します。Communications Express メールユーザーが Messenger Server の URL をどのように指定しようと (`HTTP` または `HTTPS`)、`checkoverssl` が `1` に設定されると、Messaging Server と S/MIME アプレット間のリンクは SSL で保護されます。

注 Messaging Server と、Communications Express メールなどのクライアントアプリケーション間にはプロキシサーバーを使用できます。保護されたまたは保護されていない通信リンクとプロキシサーバーの使用方法については、743 ページの「プロキシサーバーと CRL チェック」を参照してください。

クライアントマシン用のキーアクセスライブラリ

メールユーザーが非公開キーと公開キーのペアおよび証明書をスマートカードまたはブラウザのローカルキーストアのどちらに保存しようと、その保存方法をサポートするために、クライアントマシンにキーアクセスライブラリが存在する必要があります。

ライブラリは、スマートカードおよびブラウザのベンダーが提供します。適切なライブラリがクライアントマシンに存在するようにし、`smime.conf` ファイルに適切なプラットフォームパラメータでライブラリ名を指定します。選択できるパラメータを次に示します。

- PC で稼働する Microsoft Windows の場合は `platformwin`

クライアントマシンにインストールされていることがわかっているライブラリのみを指定できます。何がインストールされているかわからない場合は、特定のプラットフォームおよびベンダー用のすべてのライブラリ名を指定できます。S/MIME アプレットが指定されたライブラリ名に必要なライブラリを見つけることができない場合は、S/MIME は機能しません。

1 つまたは複数のライブラリファイル名を指定するための構文は、次のとおりです。

`platform_parameter``==`*`vendor`*`:library=`*`library_name`*`;...`

ここで、

platorm_parameter は、Communications Express メールにアクセスするクライアントマシンのプラットフォーム用のパラメータ名です。次の名前が選択可能です。
`platformwin`

vendor は、スマートカードまたはブラウザのベンダーを指定します。次の文字のいずれかを選択します。

`cac` (ActivCard または NetSign スマートカードの場合)
`capi` (CAPI を使用する Internet Explorer の場合)
`mozilla` (ネットワークセキュリティサービスを使用する Mozilla の場合)

library_name は、ライブラリファイル名を指定します。使用しているベンダーおよびオペレーティングシステムのライブラリ名については、表 20-5 を参照してください。

表 20-5　クライアントマシン用の特別なライブラリ

スマートカードまたはブラウザのベンダー	オペレーティングシステム	ライブラリファイル名
Cryptographic Application Programming Interface (CAPI) を使用する Internet Explorer	Microsoft Windows	`acpkcs211.dll`
	Microsoft Windows	`capibridge.dll`
	Microsoft Windows	`softokn3.dll`
	Microsoft Windows	`core32.dll`

例

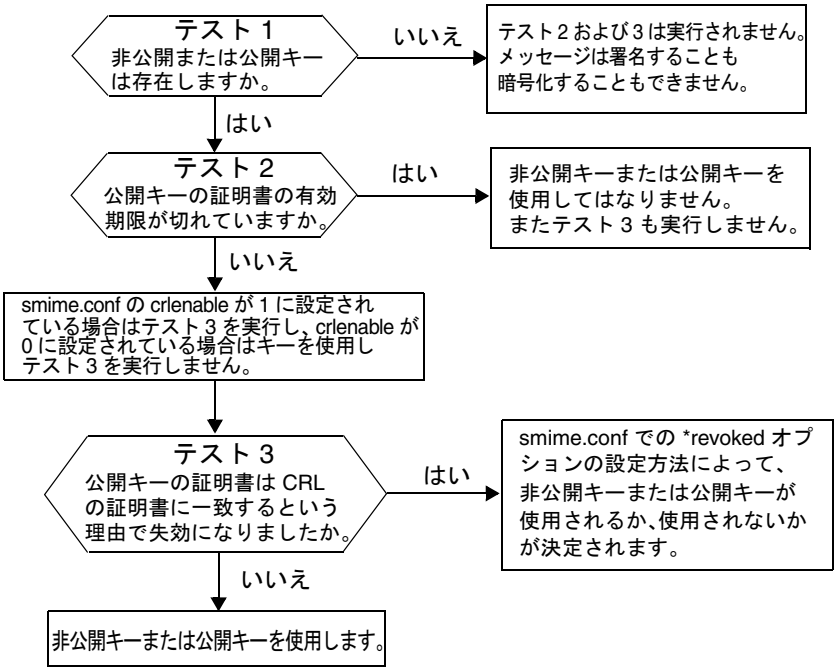
次の例では、Microsoft Windows プラットフォーム用のスマートカードライブラリを 1 つ、Internet Explorer ライブラリを 1 つ、および Mozilla ライブラリを 1 つ指定しています。

```
platformwin==CAC:library=acpkcs211.dll;CAPI:library=capibridge.dll;
MOZILLA:library=softokn3.dll;
```

非公開キーと公開キーの確認

Communications Express メールは、図 20-2 に示された確認テストに合格してからでないと、非公開キーも公開キーも使用できません。この節の以下の部分では、公開キーの証明書を CRL でチェックする方法の詳細について説明します。

図 20-2　非公開キーと公開キーの確認



ユーザーの非公開キーまたは公開キーを見つける

1 人の Communications Express メールユーザーが複数の非公開および公開キーのペアおよび複数の電子メールアドレス (プライマリ、代替、またはエイリアスアドレス) を持つ場合は、それぞれのキーがそれぞれのアドレスに関連付けられている可能性があります。その場合、S/MIME アプレットが確認のためにすべてのキーを見つけることができることが重要です。`smime.conf` ファイルで `usercertfilter` パラメータを使用して、公開キーの証明書を CRL でチェックする時にキーの所有者のメールアドレスのリストを作成するフィルタを定義します。詳細については、734 ページの「`usercertfilter`」を参照してください。

証明書が CRL でチェックされるタイミング

証明書失効リスト、すなわち CRL は、キーのペアおよび証明書を発行する CA が管理する失効した証明書のリストのことです。CRL チェックを有効にすると、証明書が要求されるたびに、その証明書が失効しているかどうかを確認するためにシステムが CRL をチェックします。

`smime.conf` ファイルで `crlenable` に `1` を設定すると、有効期限が切れていないキーが見つかると CRL テストが実行されます。公開キーの証明書は、CRL でチェックされます。各 CA の CRL は 1 つのみですが、同じ CRL が異なる場所に存在することは可能です。

S/MIME アプレットが Messaging Server にチェックの要求を送信すると、Messaging Server で CRL による証明書のチェックが実行されます。公開キーの証明書は、公開キーの検証に使用されます。非公開キーは公開されず、そのキーの所有者のみが使用するため、非公開キーを CRL で直接チェックすることはできません。非公開キーが有効であるかどうかを確認するには、キーのペアの公開キーの証明書が使用されます。公開キーの証明書が CRL テストに合格すると、関連する非公開キーもテストに合格します。

証明書の所有者が所属組織の一員でなくなったため、またはスマートカードを失くしたためなど、証明書の失効はさまざまな理由で発生します。

証明書を CRL でチェックする必要がある状況には次の 3 つがあります。

- 送信メッセージが署名されているとき

  S/MIME アプレットは、`sendsigncert` に `0` が、または `crlenable` に `0` が設定されていないかぎり、常にこのチェックを実行します。

- 着信署名付きメッセージを読むとき

  S/MIME アプレットは、`readsigncert` に `0` が、または `crlenable` に `0` が設定されていないかぎり、常にこのチェックを実行します。

- 送信メッセージが暗号化されているとき

  S/MIME アプレットは、`sendencryptcert` に `0` が、または `crlenable` に `0` が設定されていないかぎり、常にこのチェックを実行します。

CRL へのアクセス

証明書には、配布点と呼ばれる 0 個以上の URL が含まれています。Messaging Server は、配布点を使用して CRL を検索します。証明書に CRL URL が含まれていない場合、証明書を CRL でチェックすることはできず、本当のステータスがわからない状態で非公開キーまたは公開キーがメッセージの署名や暗号化に使用されます。

Messaging Server が利用できるすべての URL を試行しても CRL を見つけることもアクセスすることもできない場合は、証明書のステータスは不明になります。不明ステータスの非公開キーまたは公開キーを使用するかどうかは、`revocationunknown` の設定によって決定されます。

各 CA には CRL が 1 つのみサポートされますが、同じ CRL の複数のコピーは、ユーザーの公開キーの証明書の異なる URL で示される異なる場所に存在できます。Messaging Server は、CRL にアクセスできるようになるまで、証明書のすべての URL ロケーションを試行します。

アクセスを最適化するために、最新の CRL を CA から必要な場所に定期的にダウンロードして、複数の CRL のコピーを管理できます。証明書に埋め込まれた URL を変更することはできませんが、CRL 情報が含まれる新しい URL へ証明書内の URL をマッピングして、Messaging Server が新しい CRL の場所を使用するようにリダイレクトできます。次の構文を使用して LDAP ディレクトリ上に 1 つ以上のマッピング定義のリストを作成します (「crlmappingurl」を参照)。

`msgCRLMappingRecord=`*`url_in_certificate`*`==`*`new_url`*`[|`*`url_login_DN`*`|`*`url_login_password`*`]`

`url_in_certificate` は、CRL を検索するための古い情報が含まれる、証明書内の URL です。`new_url` は、新しい CRL 情報が含まれる新しい URL です。*url_login_DN* および *url_login_password* は new_url へのアクセスを許可されたエントリの DN およびパスワードです。両方ともオプションであり、指定した場合、新しい URL のアクセスにのみ使用されます。

DN およびパスワードが受け入れられない場合、LDAP アクセスは拒否され、その他の資格情報での再試行は行われません。これらのログイン資格情報は、LDAP URL に対してのみ有効です。`smmime.conf` で `crlurllogindn` および `crlurlloginpw` を使用する場合は、マッピングレコードにログインの DN およびパスワードを指定する必要はありません。723 ページの「公開キー、CA 証明書、および CRL にアクセスするための、資格情報を使用した LDAP へのアクセス」を参照してください。

マッピングには 1 層のみが許されます。証明書内の異なる URL を同一の新しい URL にマッピングできますが、証明書の 1 つの URL に複数の新しい URL を割り当てることはできません。たとえば、次のマッピングリストは無効です。

```
msgCRLMappingRecord=URL12==URL45
msgCRLMappingRecord=URL12==URL66
msgCRLMappingRecord=URL12==URL88
msgCRLMappingRecord=URL20==URL90
msgCRLMappingRecord=URL20==URL93
```

次の例は、正しいマッピングリストです。

```
msgCRLMappingRecord=URL12==URL45
msgCRLMappingRecord=URL14==URL66
msgCRLMappingRecord=URL88==URL66
msgCRLMappingRecord=URL201==URL90
msgCRLMappingRecord=URL202==URL93
```

LDAP ディレクトリにマッピング定義を作成したら、`smime.conf` ファイルで `crlmappingurl` を使用してマッピング定義の場所を特定するためのディレクトリ情報を指定します。729 ページの「`crlmappingurl`」を参照してください。

プロキシサーバーと CRL チェック

システムでクライアントアプリケーションと Messaging Server 間にプロキシサーバーを使用する場合、CRL チェックを実行するように適切に S/MIME アプレットを設定しても CRL チェックが妨げられることがあります。この問題が発生すると、Communications Express メールのユーザーは、有効なキーの証明書に対する失効または不明ステータスを警告するエラーメッセージを受信します。

次の状態は、問題が生じる原因になります。

- 次の設定値で CRL チェックが要求される
  - `smime.conf` ファイルで `crlenable` パラメータが `1` に設定されている
  - Messaging Server の `local.webmail.cert.enable` オプションが `1` に設定されている
- S/MIME アプレットとプロキシサーバー間の通信リンクが SSL で保護されていないが、`smime.conf` ファイルで `checkoverssl` パラメータが `1` に設定されているので、S/MIME アプレットが保護されたリンクを要求する

この問題を解決するには、以下を実行します。

1. クライアントマシンとプロキシサーバー間の通信リンクを SSL で保護されたリンクに設定し、設定値はすべてそのままにします。

   または、

2. 通信リンクを保護しない状態で、`checkoverssl` に `0` を設定します。

詳細は、736 ページの「SSL でインターネットリンクを保護する」を参照してください。

古い CRL の使用

S/MIME アプレットが Messaging Server にチェックの要求を送信すると、Messaging Server で CRL による証明書のチェックが実行されます。証明書をチェックするたびに CRL をメモリにダウンロードするのではなく、Messaging Server は CRL のコピーをディスクにダウンロードして、そのコピーを証明書のチェックに使用します。すべての CRL には、日付を指定する次の更新日フィールドがあります。この日付以降は、最新のバージョンの CRL を使用する必要があります。次の更新日は、CRL の期限切れ日または使用の時間制限とみなすことができます。次の更新日を過ぎた CRL は、古いものとみなされ、Messaging Server が証明書を次回チェックするときに最新バージョンの CRL をダウンロードするようにします。

S/MIME アプレットが証明書を CRL でチェックするように要求するたびに、Messaging Server は次のことを実行します。

1. 現在の日付を CRL の次の更新日と比較します。
2. CRL が古くなった場合は、Messaging Server は最新バージョンの CRL をダウンロードしてディスク上の古い CRL を置き換え、チェックを続けます。ただし、最新の CRL が見つからない、またはダウンロードできない場合は、`smime.conf` ファイルの `crlusepastnextupdate` の値によって何を行うべきかを判断します。
3. `crlusepastnextupdate` が `0` に設定されている場合、古くなった CRL は使用されず、問題の証明書のステータスはあいまいなになります。S/MIME アプレットは、`smime.conf` の `revocationunknown` の値によって次に実行すべきことを判断します。

   a. `revocationunknown` に `ok` が設定されている場合は、証明書を有効とみなし、非公開キーまたは公開キーを使用してメッセージの署名や暗号化が行われます。

   b. `revocationunknown` に `revoked` が設定されている場合は、証明書を無効とみなし、メッセージの署名や暗号化には非公開キーも公開キーも使用せず、キーを使用できないことをポップアップエラーメッセージでメールユーザーに警告します。

   `crlusepastnextupdate` に `1` が設定されている場合は、S/MIME アプレットは古い CRL を使用し続け、これによって Communications Express メール内では処理が中断されることはありませんが、この状況を警告するメッセージが Messaging Server ログファイルに書き込まれます。

証明書が CRL でチェックされるたびに、この一連の手順が実行されます。Messaging Server が新しいバージョンの CRL を適時ダウンロードできる限り、また `smime.conf` ファイルの設定によっては、メールの処理は中断することなく進行します。古くなった CRL が使用されていることを示すメッセージが繰り返し表示されていないかどうか、Messaging Server ログを定期的に確認してください。新しい CRL がダウンロードできない場合は、アクセスできない理由を調査する必要があります。

使用するメッセージ時刻の判断

timestampdelta パラメータは、主に次の目的で使用されます。

1. 送信先に到着するまでに時間がかかるメッセージの状況に対処するため。この場合、送信者のキーが、メッセージが送信されたときに有効であったにもかかわらず無効として処理された可能性があります。
2. 送信時間が偽造されている可能性があるので、メッセージの送信時間の信頼性を限定するため。

どのメッセージにも次の 2 つの時間が関連付けられています。

- メッセージが送信された時間。メッセージヘッダーの詳細の Date 行に示されます
- メッセージが送信先に到着した時間。メッセージヘッダーの詳細の最後の Received 行に示されます

注 メッセージヘッダーの詳細は、メッセージの From フィールドの右側にある三角のアイコンをクリックして表示できます。

メッセージの送信時に有効であった証明書は、メッセージが送信先に到着するまでに失効したり有効期限が切れたりする場合があります。このようになった場合、証明書の妥当性のチェックを行う際に送信時間と受信時間のどちらの時間を使用すべきでしょうか。送信時間を使用すると、メッセージが送信されたときに証明書が有効であったことが確認されます。しかし、常に送信時間を使用することは、メッセージが送信先に到着するのに長い時間がかかることがあるということを考慮していません。そのような場合は、受信時間を使用するほうがよいこともあります。

smime.conf ファイルで timestampdelta パラメータを使用して CRL チェックに使用する時間に影響を与えることができます。このパラメータに秒を表す正の整数を設定します。受信時間から timestampdelta の値を差し引いた時間が送信時間より前の時間である場合は、送信時間が使用されます。そうでない場合は、受信時間が使用されます。timestampdelta の値が小さいほど、受信時間が使用される頻度が高まります。timestampdelta が設定されていない場合は、常に受信時間が使用されます。733 ページの「timestampdelta」を参照してください。

CRL へのアクセスの問題

ネットワークやサーバーの問題などさまざまな理由で、Messaging Server が証明書を CRL でチェックしようとしたときに CRL が利用できないことがあります。Messaging Server に絶えず CRL へのアクセスを試行させるのではなく、`smime.conf` ファイルで `crlaccessfail` パラメータを使用して、CRL へのアクセスを試みる頻度を管理し、ほかのタスクのために Messaging Server を解放できます。

`crlaccessfail` で次のことを定義します。

- 失敗した試行をカウントする回数 (試行が失敗するたびに、Messaging Server ログにエラーメッセージが書き込まれる)
- 失敗した試行をカウントする期間
- CRL へのアクセスの新しいサイクルを試みるまで待つ期間

パラメータの構文および例については、722 ページの「crlaccessfail」を参照してください。

証明書が失効した場合

公開キーの証明書が CRL のエントリに一致しない場合、非公開キーまたは公開キーを使用して、送信するメッセージの署名や暗号化を行います。証明書が CRL のエントリに一致する場合、または証明書のステータスが不明な場合は、非公開キーまたは公開キーは失効したとみなされます。デフォルトでは、Communications Express メールは送信するメッセージの署名や暗号化に、証明書が失効しているキーは使用しません。受信者が署名付きのメッセージを読む時点でそのメッセージの非公開キーが失効している場合は、受信者は署名が信頼できないことを示す警告メッセージを受け取ります。

必要に応じて、`smime.conf` ファイルに次のパラメータを指定して、すべての失効した証明書に対する各種デフォルトポリシーを変更できます。

- 公開キーの証明書が失効したために失効したとみなされた非公開キーを使用して送信するメッセージに署名するには、`sendsigncertrevoked` に `allow` を設定します
- 証明書が失効した公開キーで送信するメッセージを暗号化するには、`sendencryptcertrevoked` に `allow` を設定します
- ステータスが不明の証明書を有効とみなすには、`revocationunknown` に `ok` を設定します。送信するメッセージの署名または暗号化には非公開キーまたは公開キーが使用されます

S/MIME 機能の使用を許可する

Communications Express メールを介して利用できるさまざまなメールサービスを使用する許可は、LDAP フィルタで付与または拒否できます。mailAllowedServiceAccess または mailDomainAllowedServiceAccess LDAP 属性でフィルタを定義します。一般に、フィルタは次の 3 通りのいずれか 1 つの役目を果たします。

- フィルタを使用しない場合にすべてのユーザーにすべてのサービスを使用する許可を与える
- ユーザーのリストに対して指定されたサービス名を使用する許可を明示的に与える (プラス記号 (+) がサービス名リストの前にくる)
- ユーザーのリストに対して指定されたサービス名を使用する許可を明示的に拒否する (マイナス記号 (-) がサービス名リストの前にくる)

S/MIME の必須のメールサービス名は、http、smime、および smtp です。Communications Express メールユーザー間で S/MIME の使用を制限する必要がある場合は、適切な LDAP 属性構文およびサービス名を使用してフィルタを作成します。属性は、LDAP コマンドで作成または変更します。

S/MIME の許可の例

1. 次の例は、1 人の Communications Express メールユーザーの S/MIME 機能へのアクセスを阻止します。

```
mailAllowedServiceAccess:-smime:*$+imap,pop,http,smtp:*
```

または

```
mailAllowedServiceAccess:+imap,pop,http,smtp:*
```

2. 次の例は、1 つのドメイン内のすべての Communications Express メールユーザーの S/MIME 機能へのアクセスを阻止します。

```
mailDomainAllowedServiceAccess:-smime:*$+imap:*$+pop:*$+smtp:*$+http:*
```

または

```
mailDomainAllowedServiceAccess:+imap:*$+pop:*$+smtp:*$+http:*
```

詳細については、697 ページの「フィルタの構文」を参照してください。

証明書の管理

次のほとんどの例では、ldapsearch および ldapmodify コマンドを使用して、LDAP ディレクトリでユーザーキーおよび証明書を検索しています。それらのコマンドは、Directory Server が提供します。それらのコマンドについては、『Sun ONE Directory Server Resource Kit Tools Reference』のリリース 5.2 を参照してください。

LDAP ディレクトリに含まれる CA 証明書

この例では、認証局の証明書を LDAP ディレクトリに追加します。それらの証明書のディレクトリ構造はすでに存在しています。証明書および証明書が属す LDAP エントリを add-root-CA-cert.ldif とうい名前の .ldif ファイルに入力します。証明書情報以外のすべてのテキストは ASCII テキストとしてファイルに入力します。証明書情報は Base64 でエンコーディングされたテキストとして入力する必要があります。

```
dn: cn=SMIME Admin,ou=people,o=demo.siroe.com,o=demo
objectClass:top
objectClass:person
objectClass:organizationalPerson
objectClass:inetOrgPerson
objectClass: certificationAuthority
cn: RootCACerts
sn: CA
authorityRevocationList: novalue
certificateRevocationList: novalue
cacertificate;binary:: MFU01JTUUEjAQBgNVBAsTCU1zZ1NlcnZlcjcMBoGA1UEAxMTydG
QGEwJVUzEOMAwGA1UEMFUJTUUxEjAQBgNVBAsTCU1zZ1NlcnZlcjEMBoGA1UEAxMTQ2VydG
aFw0wNjAxMwODAwMDBaM267hgbX9FExCzAJByrjgNVBAk9STklBMQwCgYDVQQVHR8EgaQwg
YTAlVMRMQYDVQQIEwpDQUxJRk9STklBMQwwCgYDVQQKEwww3ltgYz11lzAdBgNVBpYSE9Vc
5yZWQaddWlm899XBsYW5ldC5jb20wgZ8wDQYJoGBAK1mUTy8vvnOFg4mlHjkghytQUR1k8l
5mvWRf77ntm5mGXRD3XMU4OciUq6zUfIg3ngvxlLyERTIqjUS8HQU4R5pvj+rrVgsAGjggE
+FNAJmtOV2A3wMyghqkVPNDP3Aqq2fkcn4va3C5nRNAYxNNVE84JJ0H3jyPDXhMBlQU6vQn
weMBAAjggEXMIIBEzARBglghkgBhCAQEEBApqlSai4mfuvjh02SQkoPMNDAgTwMB8GA1UdI
QYMBaAEd38IK05AHreiU9OYc6vNMOwZMIGsBgNVHR8EgaQwgaEwb6BtoGuGaWxkYXA6Lyht
bmcucmVkLmlbGFuZXQuY29tL1VJD1DXJ0aWZpY2F0ZSBNYW5hZ2VyLE9VPVBlb3BsZSxPPW
aWxxYT9jZXJ0aZpY2jdu2medXRllkghytQURYFNrkuoCygKoYoaHR0cDovL3Bla2kghytQU
Zy5yZWQuaXBsYW5lC5jb20vcGVranLmNybDAeBgNVHREEFzAVgRNwb3J0aWEuc2hhb0BzdW
4uY29tMA0GCxLm78freCxS3Pp078jyTaDci1AudBL8+RrRUQvxsMJfZeFED+Uuf10Ilt6kw
Tc6W5UekbirfEZGAVQIzlt6DQJfgpifGLvtQ60Kw==
```

CA の証明書は、ldapmodify コマンドを使用して LDAP ディレクトリに追加されます。

```
# ldapmodify -a -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd -v
-f add-root-CA-cert.ldif
```

smime.conf の trustedurl パラメータの値は、LDAP ディレクトリ内の CA 証明書の位置を指定します。例 1 では、trustedurl は次のように設定されます。

```
trustedurl==ldap://demo.siroe.com:389/cn=SMIME Admin, ou=people,
o=demo.siroe.com,o=demo?cacertificate;binary?sub?(objectclass=certi
ficationAuthority)
```

LDAP ディレクトリに含まれる公開キーおよび証明書

この例は、メールユーザーの公開キーおよび証明書を LDAP ディレクトリに追加する方法を示しています。この例では、メールユーザーがすでに LDAP ディレクトリに存在していると仮定しています。キーおよび証明書、またそれらが属す LDAP エントリを add-public-cert.ldif とうい名前の .ldif ファイルに入力します。キーおよび証明書情報以外のすべてのテキストは ASCII テキストとしてファイルに入力します。キーおよび証明書情報は Base64 でエンコーディングされたテキストとして入力する必要があります。

```
dn: uid=JohnDoe,ou=People, o=demo.siroe.com,o=demo
changetype:modify
replace: usercertificate
usercertificate;binary:: MFU01JTUUxEjAQBgNVBAsT1zZ1NlcnZlcjMBoGA1UEAxMTydG
QGEwJVUzEAwGA1hMFU01JTUUxEjAQBgNVBAsTCU1zZ1NlcnZlcjEcMBoGA1UEAxMTQ2VydG
aFw0wNjAxMTODAwaM267hgbX9FExCzAJBgwyrjgNVBAk9STklBMQwwCgYDVQQVHR8EgaQwg
AlVzMRMwEQYDVQQIDQUxJRk9STklBMQwwCgYDVQQKEwww3ltgoOYz11lzAdBgNVBpYSE9Vc
5yZWaddiiWlm899XBsYW5ldb20wgZ8wDQYJoGBAK1mUTy8vvO2nOFg4mlHjkghytQUR1k8l
5mvgcWL77ntm5mGXRD3XMU4OcizUfIg3ngvxlLKLyERTIqjUS8HQU4R5pvj+rrVgsAGjggE
+FG9NAqtOV2A3wMyghqkVPNDP3Aqq2BYfkcn4va3RNAYxNNVE84JJ0H3jyPDXhMBlQU6vQn
1NAgMBGjggEXMIIBEzARBglghkgBhvhCAQEEBApqlSai4mfuvjh02SQMNDAgTwMB8GA1UdI
QYMBaEd38IK05AHreiU9OYc6v+ENMOwZMIGsBgNVHR8EgaQwgaEwb6BuGaWxkYXA6Lyht74
tpbmcmVkLmlwbGFuZXQuY29tL1VJRD1DZXJ0aWZpY2F0ZSBNYW5hZ2V9VPVBlb3BsZSxPPW
1haWxT9jZXJ0aWZpY2jdu2medXRllHjkghytQURYFNrkuoCygKoYoaHDovL3Bla2kghytQU
luZy5WQuaXBsYW5ldC5jb20vcGVraW5nLmNybDAeBgNVHREEFzAVgRNw0aWEuc2hhb0BzdW
4uY29A0GCxLm78UfreCxS3Pp078jyTaDv2ci1AudBL8+RrRUQvxsMJfZD+Uuf10Ilt6kwhm
Tc6W5UekbirfEZGAVQIzlt6DQJfgpifGLvtQ60Kw==
```

次のように ldapmodify コマンドを使用して、公開キーおよび証明書を LDAP ディレクトリに追加します。

```
# ldapmodify -a -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd -v
-f add-public-cert.ldif
```

smime.conf の certurl パラメータの値は、LDAP ディレクトリ内の公開キーおよび証明書の位置を指定します。例 2 では、certurl は次のように設定されます。

```
certurl==ldap://demo.siroe.com:389/ou=people, o=demo.siroe.com,
o=demo?userCertificate;binary?sub?
```

キーおよび証明書が LDAP ディレクトリに存在することを確認する

次の例では、LDAP ディレクトリで CA 証明書、また公開キーとその証明書を検索する方法を示しています。

1 つの CA 証明書の検索

次の例では、-b オプション、cn=SMIME admin, ou=people,o=demo.siroe.com,o=demo objectclass=* で定義されたベース DN は LDAP ディレクトリに存在する 1 つの CA 証明書を示します。ディレクトリに見つかると、ldapsearch は証明書についての情報を ca-cert.lidf ファイルに返します。

```
#  ldapsearch -L -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd
-b "cn=SMIME admin, ou=people,o=demo.siroe.com,o=demo"
"objectclass=*" > ca-cert.ldif
```

次の例は、ca-cert.ldif ファイルに出力された検索結果を示しています。ファイルの内容の形式は、ldapsearch の -L オプションを使用した結果です。

```
# more ca-cert.ldif
dn: cn=SMIME admin,ou=people,o=demo.siroe.com,o=demo
objectClass:top
objectClass:person
objectClass:organizationalPerson
objectClass:inetOrgPerson
objectClass: certificationAuthority
cn: RootCACerts
cn: SMIME admin
sn: CA
authorityRevocationList: novalue
certificateRevocationList: novalue
cacertificate;binary:: MFU01JTUUxEjAQBgNVBAsTCU1zZNlcnZlcjcMBoGA1UEAxMTydG
QGEwJVEOMAwGA1UEChMFU0UUxEjAQBgNVBAsTCU1zZ1NlcnZlcjEcMBoGA1UEAxMTQ2VydG
aFw0jAxMTIwODAwMDBaM267X9FExCzAJBgwyrjgNVBAk9STklBMQwwCgYDVQQVHR8EgaQwg
YlVzMRMwEQYDVQQIEwpDQUx9STklBMQwwCgYDVQQKEwww3ltgoOYz11lzAdBgNVBpYSE9Vc
5yQuaddiiWlm899XBsYW5ljb20wgZ8wDQYJoGBAK1mUTy8vvO2nOFg4mlHjkghytQUR1k8l
5mcWRfL77ntm5mGXRD3XMciUq6zUfIg3ngvxlLKLyERTIqjUS8HQU4R5pvj+rrVgsAGjggE
+FNAJmqtOV2A3wMyghqkDP3Aqq2BYfkcn4va3C5nRNAYxNNVE84JJ0H3jyPDXhMBlQU6vQn
1NABAAGjggEXMIIBEzglghkgBhvhCAQEEBApqlSai4mfuvjh02SQkoPMNDAgTwMB8GA1UdI
QYMAFEd38IK05AHreOYc6v+ENMOwZMIGsBgNVHR8EgaQwgaEwb6BtoGuGaWxkYXA6Lyht74
tpbucmVkLmlwbGFuZY29tL1VJRD1DZXJ0aWZpY2F0ZSBNYW5hZ2VyLE9VPVBlb3BsZSxPPW
```

```
1haWYT9jZXJ0aWZpdu2medXRllHjkghytQURYFNrkuoCygKoYoaHR0cDovL3Bla2kghytQU
luZyZWQuaXBsYW5ldb20vcGVraW5nLmNybDAeBgNVHREEFzAVgRNwb3J0aWEuc2hhb0BzdW
4uYtMA0GCxLm78Ufre3Pp078jyTaDv2ci1AudBL8+RrRUQvxsMJfZeFED+Uuf10Ilt6kwhm
Tc6W5UekbirfEZGAVQIzlt6DQJfgpifGLvtQ60Kw==
```

複数の公開キーの検索

次の例では、-b オプション、o=demo.siroe.com,o=demo objectclass=* で定義されたベース DN は、LDAP ディレクトリ内でそのベース DN 以下にあるすべての公開キーおよび証明書を usergroup.ldif ファイルに返します。

```
# ldapsearch -L -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd
-b "o=demo.siroe.com,o=demo" "objectclass=*" > usergroup.ldif
```

1 つの公開キーの検索

次の例では、-b オプション、uid=JohnDoe, ou=people,o=demo.siroe.com,o=demo objectclass=* で定義されたベース DN は LDAP ディレクトリに存在する 1 つの公開キーとその証明書を示します。

```
# ldapsearch -L -h demo.siroe.com -D "cn=Directory Manager" -w mypasswd
-b "uid=JohnDoe, ou=people,o=demo.siroe.com,o=demo" "objectclass=*" >
public-key.ldif
```

次の例は、public-key.ldif ファイルに出力された検索結果を示しています。ファイルの内容の形式は、ldapsearch の -L オプションを使用した結果です。

```
# more public-key.ldif
dn: uid=sdemo1, ou=people, o=demo.siroe.com, o=demo
objectClass:top
objectClass:person
objectClass:organizationalPerson
objectClass: siroe-am-managed-person
objectClass:inetOrgPerson
objectClass:inetUser
objectClass:ipUser
objectClass:userPresenceProfile
objectClass:inetMailUser
objectClass:inetLocalMailRecipient
objectClass: icsCalendarUser
objectClass: sunUCPreferences
mail: JohnDoe@demo.siroe.com
mailHost: demo.siroe.com
.

.

uid: JohnDoe
.
```

```
.
mailUserStatus:active
inetUserStatus:active
.
.
usercertificate;binary:: MFU01JTUUxEjAQBgNBAsTCU1zZ1NlcnZjcMBoGA1UEAxMTydG
QGEwJEOwGA1UEChMFU01JTUUxEjAQBgNVBAsTCU1zZ1NlcnZlcjEcMBoGA1UEAxMTQ2VydG
aFw0MTIwODAwMDBaM267hgbX9FExCzAJBgwyrjgNVBAk9STklBMQwwCgYDVQQVHR8EgaQwg
YTAlVEQYDVQQIEwpDQUxJRk9STklBMQwwCgYDVQQKEwww3ltgoOYz11lzAdBgNVBpYSE9Vc
5yZWQdWlm899XBsYW5ldC5jb20wgZ8wDQYJoGBAK1mUTy8vvO2nOFg4mlHjkghytQUR1k8l
5mvgc7ntm5mGXRD3XMU4OciUq6zUfIg3ngvxlLKLyERTIqjUS8HQU4R5pvj+rrVgsAGjggE
+FG9NmV2A3wMyghqkVPNDP3Aqq2BYfkcn4va3C5nRNAYxNNVE84JJ0H3jyPDXhMBlQU6vQn
1NAgMAgEXMIIBEzARBglghkgBhvhCAQEEBApqlSai4mfuvjh02SQkoPMNDAgTwMB8GA1UdI
QYMBaEdK05AHreiU9OYc6v+ENMOwZMIGsBgNVHR8EgaQwgaEwb6BtoGuGaWxkYXA6Lyht74
tpbucmVkwbGFuZXQuY29tL1VJRD1DZXJ0aWZpY2F0ZSBNYW5hZ2VyLE9VPVBlb3BsZSxPPW
1haxYT9jZaWZpY2jdu2medXRllHjkghytQURYFNrkuoCygKoYoaHR0cDovL3Bla2kghytQU
luZyZWQuaYW5ldC5jb20vcGVraW5nLmNybDAeBgNVHREEFzAVgRNwb3J0aWEuc2hhb0BzdW
4u9tMA0GC78UfreCxS3Pp078jyTaDv2ci1AudBL8+RrRUQvxsMJfZeFED+Uuf10Ilt6kwhm
Tc6W5UekbirfEZGAVQIzlt6DQJfgpifGLvtQ60Kw==
.
.
```

ネットワークセキュリティサービスの証明書

ネットワークセキュリティサービス (NSS) に使用される各種証明書は、LDAP データベースではない、それぞれのデータベースに保存されます。Messaging Server には、証明書および関連する CRL をデータベースに格納するための 2 つのユーティリティ certutil および crlutil があります。これらのユーティリティを使用して、データベースを検索することもできます。

certutil の詳細については、『Sun Java System Directory Server 管理ガイド』(http://docs.sun.com/doc/819-2011?l=ja) を参照してください。crlutil ユーティリティの詳細は、このユーティリティに付属するヘルプテキストを参照してください。どちらのユーティリティのオンラインヘルプも、それらのユーティリティを引数なしで実行することによって表示できます。

Communications Express S/MIME エンドユーザー情報

この節には、エンドユーザー用の情報が含まれています。この節には、以下の項があります。

初めてのログイン

メールユーザーが Communications Express メールに初めてログインする場合、S/MIME アプレットに関連する特別なプロンプトが表示されます。

Microsoft Windows の場合のプロンプト

Microsoft Windows 98、2000、または XP で初めて Communications Express メールにログインするときには、次のプロンプトが表示されます。

1. 使用しているコンピュータ (クライアントマシン) に Java 2 Runtime Environment (JRE) がインストールされていない場合は、次のようなプロンプトが表示されます。

   ```
   Do you want to install and run "Java Plug-in 1.4.2_03 signed on
   11/20/03 and distributed by Sun Microsystems, Inc."?
   Publisher authenticity verified by: VeriSign Class 3 Code Signing
   2001 CA
   ```

   「はい」をクリックして、続くプロンプトに従って JRE をインストールします。

注

英語のサポートを希望し、かつ中国語などの Latin 以外の文字が含まれる着信 S/MIME メッセージを読む必要がある場合は、使用しているコンピュータの `/lib` ディレクトリに `charsets.jar` ファイルが存在する必要があります。

`charsets.jar` ファイルが `/lib` ディレクトリにインストールされるようにするには、カスタムインストールを使用して JRE の英語版をインストールするようにします。インストールプロセス時に、「その他の言語のサポート」オプションを選択します。

詳細については、716 ページの「複数言語のサポート」を参照してください。

最後のインストールプロンプトに対して、「完了」をクリックします。コンピュータを再起動して、再度 Communications Express メールにログインします。

2. 次のプロンプトが表示されます。

   ```
   Do you want to trust the signed applet distributed by "Sun Microsystems, Inc."?
   Publisher authenticity verified by: Thawte Consulting cc
   ```

   次のいずれかの応答をクリックします。

   - 「はい」。この Communications Express メールセッションに S/MIME アプレットを受け入れる場合。ログインするたびにプロンプトが表示されます。
   - 「いいえ」。S/MIME アプレットを拒否する場合。S/MIME 機能を使用できません。
   - 「常に」。この Communications Express メールセッションおよびそれ以降のすべての Communications Express メールセッションに S/MIME アプレットを受け入れる場合。再度プロンプトが表示されることはありません。

3. 次のプロンプトが表示されます。

   ```
   Do you want to trust the signed applet distributed by "sun microsystems, inc."?
   Publisher authenticity verified by: VeriSign, Inc.
   ```

   次のいずれかの応答をクリックします。

   - 「はい」。この Communications Express メールセッションに S/MIME アプレットを受け入れる場合。ログインするたびにプロンプトが表示されます。
   - 「いいえ」。S/MIME アプレットを拒否する場合。S/MIME 機能を使用できません。
   - 「常に」。この Communications Express メールセッションおよびそれ以降のすべての Communications Express メールセッションに S/MIME アプレットを受け入れる場合。再度プロンプトが表示されることはありません。

署名と暗号化の設定

これらの設定は、すべてのユーザーの送信メッセージを次のいずれの方法で処理するかを制御する、初期の署名および暗号化の設定です。

- 自動的に署名する、または
- 自動的に暗号化する、または
- 自動的に署名し暗号化する

また、初期の設定は、Communications Express の「メール」ウィンドウおよび「オプション」-「設定」ウィンドウの下部にある署名および暗号化のチェックボックスにチェックマークを付ける (機能がオン)、または付けない (機能がオフ) のいずれかで表示するかも制御します。`smime.conf` ファイルで `alwaysencrypt` および `alwayssign` パラメータを使用して、初期設定を指定します。

メールメッセージの初期の設定を変更できることをメールユーザーに知らせます。Communications Express メールにログインしたあと、ユーザーは、一時的に 1 つのメッセージの設定を無効にしたり、途中ですべてのメッセージの設定を無効にしたりできます。

表 20-6 には、チェックボックスの使用が要約されています。

表 20-6 Communications Express メールの署名および暗号化チェックボックス

チェックボックスのテキスト	場所	Communications Express メールユーザーが実行すること
メッセージに署名する	Communications Express メールウィンドウの下部にあり、メッセージの作成、転送、または返信に使用されます。	• 現在のメッセージに署名するにはこのボックスにチェックマークを付けます。 • 現在のメッセージに署名しない場合はこのボックスにチェックマークを付けません。
メッセージを暗号化する	Communications Express メールウィンドウの下部にあり、メッセージの作成、転送、または返信に使用されます。	• 現在のメッセージを暗号化するにはこのボックスにチェックマークを付けます。 • 現在のメッセージを暗号化しない場合はこのボックスにチェックマークを付けません。
すべての送信メッセージに署名する	Communications Express メールの「オプション」-「設定」ウィンドウの「セキュアなメッセージ送信」オプション下にあります。	• すべてのメッセージを自動的に署名するには、このボックスにチェックマークを付けます。 • すべてのメッセージに自動的に署名しない場合は、このボックスにチェックマークを付けません。 **注 :**「すべての送信メッセージに署名する」の設定は、「メッセージに署名」チェックボックスでメッセージごとに無効にできます。

表 20-6　Communications Express メールの署名および暗号化チェックボックス (続き)

チェックボックスのテキスト	場所	Communications Express メールユーザーが実行すること
すべての送信メッセージを暗号化する	Communications Express メールの「オプション」-「設定」ウィンドウの「セキュアなメッセージ送信」オプション下にあります。	• すべてのメッセージを自動的に暗号化するには、このボックスにチェックマークを付けます • すべてのメッセージを自動的に暗号化しない場合は、このボックスにチェックマークを付けません。 **注 :**「すべての送信メッセージを暗号化する」の設定は、「メッセージを暗号化」チェックボックスでメッセージごとに無効にできます。

Java コンソールを有効にする

Communications Express メールユーザーが署名および暗号化されたメッセージを処理するときに、S/MIME アプレットが Java コンソールにさまざまなオペレーティングメッセージを出力できます。メールユーザーが報告する問題のトラブルシューティングに、Java コンソールのメッセージが役立ちます。ただし、オペレーティングメッセージは、LDAP エントリの `inetMailUser` オブジェクトクラスに `nswmExtendedUserPrefs` 属性を追加することによって、Java コンソールをユーザーが使用できるようにした場合にだけ生成されます。次に例を示します。

```
nswmExtendedUserPrefs: meSMIMEDebug=on
```

すべてのメールユーザーが Java コンソールを常に使用できるようにしてはなりません。そのようにすると、Communications Express メールのパフォーマンスが著しく低下します。

第 21 章

ログの管理

この章では、Messaging Server MTA、メッセージストア、およびサービスに対するログ機能の概要を示します。この章では、ログ機能の管理方法の手順も示します。

この章には、以下の節があります。

ログの概要

ログは、システムのサービスについてのタイムスタンプおよびラベルをシステムが提供する手段です。ログは、システムの現在のスナップショットおよび履歴ビューの両方を提供します。

Messaging Server のログファイルを理解して使用すると、次のことが可能です。

- メッセージのサイズ、メッセージの配信の頻度、MTA を通るメッセージの数などの、メッセージの統計情報の収集
- 傾向の見極め
- 容量計画の関連付け
- 問題のトラブルシューティング

たとえば、サイトでユーザー数の増加に伴いディスク容量を追加する必要がある場合は、Messaging Server のログファイルを使用してシステムの要求がどの程度増加したかを確認し、必要な新しいディスクストレージの容量の計画を立てることができます。

また、Messaging Server のログを使用して、1 日にわたるメッセージングのパターンがどのようなものであるかを理解することができます。日々の負荷のピークがいつ発生するかを理解すると、容量計画を立てるのに役立ちます。

ログは、ユーザーの問題のトラブルシューティングにも役立ちます。たとえば、ユーザーが予期していたメールメッセージを受信しない場合、Messaging Server のログ機能を使用してユーザーのメールメッセージを追跡できます。そのようにすると、メッセージが自動的にフィルタ処理され、SPAM フォルダに送信されたために受信できなかったことがわかる場合があります。

ログデータのタイプ

一般に、ログは次の 2 つの情報を提供します。

- 処理データ
- エラー状態、イベントログとも呼ばれる

ほとんどの場合、Messaging Server のログは処理データを提供します。この処理データには、メッセージがシステムに到着した日時、メッセージの送信者および受信者、メッセージがディスクに書き込まれたとき、その後、メッセージがディスクから削除され、ユーザーのメールボックスに入れられたときなどの情報が含まれます。

ただし、Messaging Server のログはイベントログデータも一部提供します。イベントログデータを入手するには、いくつかのログファイルから複数の項目を収集する必要があります。次に、メッセージ ID などの一意の定数を使用して、システムの 1 地点から次の地点へ通過するメッセージのライフサイクルを検索して関連付けを行うことができます。

Messaging Server のログファイルのタイプ

Messaging Server のログには、次の 3 種類のログファイルがあります。

1. **MTA ログ** : これらのログは Message Transfer Agent で説明した処理データを提供します。
2. **エラーログ** : これらのログは MTA デバッグログおよび MTA サブコンポーネントログ (ジョブコントローラやディスパッチャなど) です。
3. **メッセージストアおよびサービスのログ** : これらのログは、http サーバー、mshttpd、imap、および pop サービス、また管理サービスからのメッセージを提供します。これらの種類のログの形式は、最初の 2 種類のログとは異なります。

次の表は、異なる種類のログファイルを示しています。デフォルトでは、ログファイルは *msg_svr_base*/data/log ディレクトリにあります。各種類のログファイルを個別にカスタマイズしたり表示したりすることができます。

表 21-1　Messaging Server のログファイル

ログファイルの種類	ログファイルの説明	デフォルト名
Message Transfer Agent	日時情報、キューの出入り情報などの、MTA を通るメッセージトラフィックについての情報を表示します。	mail.log、mail.log_current、mail.log_yesterday
接続	電子メールを送信するためにこのシステムに接続するリモートマシン (MTA) が含まれます。	connection.log
カウンタ	チャネルごとに送受信されたメッセージの傾向の情報が含まれます。	counters
ジョブコントローラ	マスター、ジョブコントローラ、送信者、およびキューからの取り出しチャネルプログラムのデータが含まれます。	job_controller.log
ディスパッチャ	ディスパッチャに関連するエラーが含まれます。ディスパッチャのデバッグをオンにすると、情報が増えます。	dispatcher.log
チャネル	チャネルに関連するエラーが記録されます。キーワードの master_debug と slave_debug は、チャネルのデバッグをオンにし、そのようにすると、チャネルのログファイルの情報がさらに詳細になります。情報のレベルおよび種類は、option.dat の各種 *_DEBUG MTA オプションで制御されます。	*channelname*_master.log* (例 : tcp_local_master.log*) *channelname*_slave.log* (例 : tcp_local_slave.log*)
管理	管理サーバーを介したコンソールと 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	admin、admin.*sequenceNum.timeStamp*
IMAP	サーバーの 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	imap、imap.*sequenceNum.timeStamp*
POP	サーバーの 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	pop、pop.*sequenceNum.timeStamp*
HTTP	サーバーの 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	http、http.*sequenceNum.timeStamp*
デフォルト	サーバーのその他のアクティビティ (コマンド行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	default、default.*sequenceNum.timeStamp*
msgtrace	メッセージストアについての追跡情報が含まれます。ファイルが短時間に非常に大きくなることがあります。それに合わせて監視します。	msgtrace

表 21-1　Messaging Server のログファイル (続き)

ログファイルの種類	ログファイルの説明	デフォルト名
watcher	プロセスの失敗や応答しないサービスを監視し (113 ページの表 4-4 を参照)、特定の障害を示すエラーメッセージを記録します。	watcher

各項目の説明

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を指定します。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、サーバーのインストール時に始まり、そのサーバー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を指定します。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。詳細は、791 ページの「サービスログオプションを定義、設定する」を参照してください。

各種ログファイルのメッセージの追跡

次の節では、システム内でのメッセージの流れについて、またどの時点で情報が各種ログファイルに書き込まれるかについて説明します。この説明は、問題のトラブルシューティングおよび解決のために Message Server のログファイルを使用する方法を理解するのに役立ちます。参考のために 188 ページの表 8-2 を参照してください。

1. リモートホストはメッセージングホストの TCP ソケットに接続し、SMTP サービスを要求します。
2. MTA ディスパッチャは要求に応答し、接続をメッセージジングホストの SMTP サービスに渡します。

   MTA はモジュール方式になっているため、ジョブコントローラや SMTP サービスディスパッチャを含む、1 組のプロセスから構成されています。ディスパッチャは、着信 TCP 接続を受けて、SMTP サービスへ送信します。SMTP サービスは、メッセージをディスクのチャネル領域に書き込みます。SMTP サービスは、送信者や受信者などの、メッセージのエンベロープパラメータを認識します。システムの設定エントリは、メッセージがどのチャネル宛てかを示します。

3. ディスパッチャは、スレッドをフォークし、そのスレッドを特定の IP アドレスからの着信接続に利用可能にしたことを dispatcher.log ファイルに書き込みます。
4. SMTP サーバーは、リモートホストが接続し、メッセージを送信した場合に行われた対話を記録し、tcp_smtp_server.log ファイルに書き込みます。このログファイルは、ディスパッチャがホストの IP 上にある SMTP サーバーに接続を渡すときに作成されます。
5. SMTP サーバーは、tcp_intranet などのチャネルプログラム用のディスク上のキュー領域にメッセージを書き込み、ジョブコントローラに通知します。
6. ジョブコントローラは、チャネルプログラムにアクセスします。
7. チャネルプログラムはメッセージを配信します。

   各チャネルには、それぞれのログファイルがあります。ただし、それらのログは通常チャネルの開始および停止を示します。さらに情報を得るには、そのチャネルのデバッグレベルを有効にする必要があります。ただし、そのようにするとシステムの処理速度が低下し、問題がさらに不明瞭になるので、実際に問題が発生している場合にのみデバッグレベルを有効にします。

注 効率性を向上させるため、すでに存在するプロセス用にチャネルが実行中である場合は、新しいメッセージが着信しても、システムは新しいチャネルプロセスを生成しません。現在実行されているプロセスが新しいメッセージを引き受けます。

8. メッセージは、別のホスト、別の TCP 接続などの、次のホップへ配信されます。この情報は connection.log ファイルに書き込まれます。

   SMTP サーバーがディスク上のキュー領域にメッセージを書き込むと同時に、メッセージを管理するチャネルもレコードを mail.log_current、または mail.log ファイルに書き込みます。レコードは、メッセージがキューに入れられた日時、送信者、受信者などの情報を示します。詳細については、770 ページの「MTA メッセージログの例」を参照してください。メッセージの追跡にもっとも役立つファイルは、mail.log_current ファイルです。

ログの管理用のツール

コンソールおよび configutil コマンドを使用して Messaging Server ログファイルの作成および管理のためのポリシーをカスタマイズできます。

メッセージストアログの場合、コンソール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 は別のログ機能を使用しているため、コンソールを使って MTA ログサービスを設定したり、ログを表示したり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。

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。

MTA メッセージおよび接続のログの管理

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。

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。

詳細については、767 ページの「MTA ログを有効にする」を参照してください。

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 *msg_svr_base*/data/log/mail* ファイルにエントリが書き込まれます。そのようなログエントリは、MTA (または特定のチャネル) を通過するメッセージの数についての統計情報を収集するのに役立ちます。それらのログエントリを使用して、メッセージが送信または配信されたかどうか、またいつ送信または配信されたかなどのその他の問題も調査できます。

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。また、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 *msg_svr_base*/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。

警告 ログが有効になっていると、mail.log ファイル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監視し、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 ログエントリの形式について

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、次に示すように、各ログファイルエントリには 8 個または 9 個のフィールドがあります。

```
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 (上の例では 19-Jan-1998 19:16:57.64)。
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 サーバーの受信を示し、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 (上の例では E)。764 ページの表 21-2 を参照。
5. メッセージのサイズ (上の例では 1)。デフォルトでは K 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 (上の例では 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です。
7. エンベロープ To: アドレスの元の形式 (上の例では 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 (上の例では marlowe@siroe.com)。
9. 配信ステータス (SMTP チャネルのみ)。

次の表には、ログエントリコードの説明があります。

表 21-2　ログエントリのコード

エントリ	説明
B	不良コマンドが SMTP サーバーに送信されました。受取人アドレスフィールドには拒否されたコマンドが、診断フィールドには SMTP サーバーからの応答が含まれます。MTA チャネルオプションの MAX_B_ENTRIES は、特定のセッションでログされる不良コマンドの数を制御します。デフォルトは 10 です。
BA	トランザクションの前の方で認証が成功したあとの不良コマンド。
BS	TLS の起動に成功したあとの不良コマンド。
BSA	TLS や AUTH での不良コマンド。
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
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	キューに入れる
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)

表 21-2　ログエントリのコード (続き)

エントリ	説明
K	受取メッセージの拒否。差出人が NOTIFY=NEVER DSN フラグの設定を要求した場合、メッセージがタイムアウトになった場合、またはメッセージが手動で返された場合。たとえば、imsimta qm "delete" コマンドは常に各受取人の「K」レコードを生成するが、qm "return" コマンドが「R」レコードではなく「K」レコードを生成する場合。これは、差出人自身の要求によって、通知が差出人に送信されなかったことを示します。 これは、同じ種類の拒否またはタイムアウトである「R」レコードと比較できますが、「R」レコードでは、この失敗したメッセージに関して、元の差出人に戻される新しい通知メッセージも生成されます。
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)、失敗またはバウンスメッセージの生成
W	メッセージはまだ配信されていないが、キューに残っていて再配信が試行されていることを元の差出人に通知するために送信された警告メッセージ。
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。診断フィールドがあとに続きます。connection.log_current (ログファイルが 1 つしか使用されない場合は mail.log_current) に書き込まれます。接続が終了した理由を記録するために使用されます。特に、なんらかのセッション切断制限に達したために接続が終了した場合は、その情報が診断フィールドに表示されます。
O	接続開始
U	ログの SMTP 認証の成功および失敗。形式はほかの O エントリや C エントリと同じです。特に、同じアプリケーションや転送情報のフィールドが同じ順序で表示されます。ユーザー名が明らかな場合は、ユーザー名フィールドに記録されます。このエントリは、LOG_CONNECTION MTA オプションのビット 7 (値は 128) によって制御されます。
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 MTA オプションファイルで有効になっている場合、形式は次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。

```
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) 文字と数。これがマルチスレッドチャネルエントリ (つまり、tcp_* チャネルエントリ) であった場合、プロセス ID と数の間にスレッド ID も存在します。この例では、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) します。
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 サーバー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) になります。
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 (着信 SMTP メッセージの場合) や、チャネルの正規のホスト名 (ほか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。362 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 ログを有効にする

少数の特定の MTA チャネルの統計情報を収集するには、対象となる MTA チャネルのみのログチャネルキーワードを有効にします。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。

➤ 特定のチャネルの MTA のログを有効にする

1. `imta.cnf` ファイルを編集します。

   このファイルは `/opt/SUNWmsgsr/config` ディレクトリにあります。

2. 特定のチャネルに対してログの作成を有効にするには、チャネル定義で `logging` キーワードを追加します。次に例を示します。

   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。787 ページの「サービスログの管理」を参照してください。

➤ すべてのチャネルの MTA ログを有効にする

1. `imta.cnf` ファイルを編集します。

   このファイルは `/opt/SUNWmsgsr/config` ディレクトリにあります。

2. MTA 設定ファイルのチャネルブロックのセクションの `defaults` チャネル (346 ページの「チャネルのデフォルトを設定する」を参照) に `logging` キーワードを追加します。次に例を示します。

```
defaults logging notices 1 2 4 7 copywarnpost copysendpost
postheadonly noswitchchannel immnonurgent maxjobs 7 defaulthost
siroe.com

l defragment charset7 us-ascii charset8 iso-8859-01
siroe.com
```

その他の MTA ログオプションの指定

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。IMTA テイラーファイル (*msg_svr_base*/config/imta_tailor) の IMTA_OPTION_FILE オプションで指定されたファイルで、MTA オプションファイルを指定します。デフォルトでは、これは *msg_svr_base*/config/option.dat ファイルです。

MTA オプションファイルの詳細については、『Sun Java System Messaging Server Reference』(http://docs.sun.com/doc/819-0106) を参照してください。

➤ MTA ログを syslog へ送信する

1. MTA オプションファイルを編集します。
2. LOG_MESSAGES_SYSLOG オプションを 1 に設定します。

   0 の値はデフォルトであり、syslog (イベントログ) のログが実行されなかったことを示します。

➤ ログメッセージエントリを関連付ける

1. MTA オプションファイルを編集します。
2. LOG_MESSAGES_ID オプションを 1 に設定します。

   0 の値はデフォルトであり、メッセージ ID が mail.log ファイルに保存されなかったことを示します。

➤ メッセージの配信再試行を確認する

1. MTA オプションファイルを編集します。
2. LOG_FILENAME オプションを 1 に設定します。

   このオプションを使用すると、特定のメッセージファイルの配信が何回再試行されたかを即座に簡単に確認できます。このオプションは、MTA が複数の受信者へのメッセージをディスク上で別々のメッセージファイルコピーに分割する場合としない場合を把握するのにも役立ちます。

➤ TCP/IP 接続のログを記録する

1. MTA オプションファイルを編集します。
2. LOG_CONNECTION オプションを設定します。

このオプションを使用すると、MTA は TCP/IP 接続とメッセージトラフィックのログを記録します。接続ログエントリは、デフォルトで mail.log* ファイルに書き込まれます。さらに、接続ログエントリを connection.log* ファイルに書き込むことも可能です。詳細は、SEPARATE_CONNECTION_LOG オプションを参照してください。

➤ connection.log ファイルにエントリを書き込む

1. MTA オプションファイルを編集します。
2. SEPARATE_CONNECTION_LOG オプションを 1 に設定します。

   このオプションを使用して、ログエントリを connection.log ファイルに代わりに書き込むことを指定できます。デフォルト値の 0 は、接続ログを MTA ログファイルに格納します。

➤ プロセス ID でログメッセージを関連付ける

1. MTA オプションファイルを編集します。
2. LOG_PROCESS オプションを設定します。

   LOG_CONNECTION とともに使用すると、このオプションは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。

➤ メールを mail.log ファイルのキューに入れるプロセスに関連付けられたユーザー名を保存する

1. MTA オプションファイルを編集します。
2. LOG_USERNAME オプションを設定します。

   このオプションは、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ー名を 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、ユーザー名フィールドが認証ユーザー名 (プレフィックスとしてアスタリスクが付いたもの) になります。

MTA メッセージログの例

MTA メッセージファイルに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、768 ページの「その他の MTA ログオプションの指定」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。

MTA ログの例 : ユーザーがメッセージを送信する場合

ローカルユーザーが送信 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次に示します。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。

コード例 21-1 ログ : ローカルユーザー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、1 ブロック (1) メッセージをチャネル `1` からチャネル `tcp_local` のキューに入れた (E) ときの日付と時刻を示します。
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。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. 1 ブロック (1) メッセージを `tcp_local` チャネルのキューから取り出した (D) ときの日付と時刻を示しています。つまり、`tcp_local` チャネルがリモートの SMTP サーバー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 サーバーの 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 サーバーは拡張 SMTP ステータスコードと追加テキストで応答しています。

MTA ログの例 : オプションのログフィールド

コード例 21-2 はコード例 21-3 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。

コード例 21-2 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.
```

MTA ログの例 : リストに送信する場合

773 ページのコード例 21-3 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、ユーザー 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。

コード例 21-3 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.
```

MTA ログの例 : 存在しないドメインに送信する場合

775 ページのコード例 21-4 は、存在しないドメイン (ここでは very.bogus.com) への送信の試行を示しています。つまり、存在しないことが MTA の書き換えルール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 は元の差出人へのバウンス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、バウンス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 が生成したバウンス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ないが、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る)。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)。

コード例 21-4 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>
```

MTA ログの例 : 存在しないリモートユーザーに送信する場合

777 ページのコード例 21-5 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。775 ページのコード例 21-4 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 サーバーが発行した 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。

コード例 21-5 ログ : 存在しないリモートユーザーに送信する場合

```
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>
```

MTA ログの例 : リモート側のメッセージ送信試行が拒否される場合

コード例 21-6 に、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 (553 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

! ... 多数のエントリを省略 ...
!
   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、リモートユーザー harold@varrius.com が MTA システムを介してリモートユーザー alan@very.bogus.com にリレーしようとしても、拒否されます。

コード例 21-6 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。MTA チャネルが拒否されるメッセージを送信しようとすると、コード例 21-4 とコード例 21-5 で示されているように R レコードで示されます。

注 ログに書き込まれた最後の J レコードは、特定のセッションの最後のレコードであることを示します。また、Messaging Server の現在のバージョンは J レコードの数を制限しません。

---

2. 試行されたエンベロープ From: および 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。

MTA ログの例 : 配信試行が複数回行われた場合

コード例 21-7 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。

コード例 21-7 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。775 ページのコード例 21-4 とは異なり、DNS は宛先ドメイン名 some.org を否定していません。「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 サーバー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 サーバー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 サーバーをまだ起動していない、またはその SMTP サーバー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。

MTA ログの例 : 変換チャネルを通過する着信 SMTP メッセージ

781 ページのコード例 21-8 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。

コード例 21-8　　ログ : 変換チャネルを通過する着信 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. 外部ユーザー 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。

MTA ログの例 : 送信接続ログ

782 ページのコード例 21-9 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、ユーザー 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 (同様のチャネルが大抵そうであるように) single_sys チャネルキーワードで示された tcp_local 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。

コード例 21-9 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、これらの個別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 サーバー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。

MTA ログの例 : 受信接続ログ

コード例 21-10 に、LOG_CONNECTION=3 によって接続ログが有効になっているときの着信 SMTP メッセージのログ出力を示します。

コード例 21-10 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、このエントリが着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、着信 TCP/IP チャネル (tcp_local) からチャネル l 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 12 章「チャネル定義を設定する」を参照してください。

4. 着信接続が閉じています。c 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、このエントリが着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を有効にする

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。ディスパッチャの設定情報は、*msg_svr_base*/imta/dispatcher.cnf ファイルに指定されます。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は、dispatcher.cnf ファイルを編集することができます

➤ ディスパッチャのエラーデバッグ出力を有効にする

1. dispatcher.cnf ファイルを編集します。
2. DEBUG オプションを -1 に設定します。

   論理または環境変数の IMTA_DISPATCHER_DEBUG (UNIX) を設定することもできます。この変数は、32 ビットのデバッグマスクに 16 進数の FFFFFFFF の値を定義します。次の表には、各ビットの意味の説明があります。

表 21-3 ディスパッチャデバッグビット

ビット	16 進数値	10 進数値	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。

表 21-3　ディスパッチャデバッグビット (続き)

ビット	16 進数値	10 進数値	使用目的
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)。
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 拒否の dispatcher.log ファイルへのログ。

➤ ディスパッチャパラメータを設定する (Solaris)

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。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。

1. ヒープサイズ (すなわち、デフォルトの datasize) を表示するには、次のいずれかを使用します。

   csh コマンド :

   ```
   # limit
   ```

   ksh コマンド

   ```
   # ulimit -a
   ```

   Solaris ユーティリティ

   ```
   # sysdef
   ```

2. 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。

サービスログの管理

この節では、メッセージストア (POP、IMAP、および HTTP)、管理、およびデフォルトの各サービスのログについて説明します (759 ページの表 21-1 を参照)。

これらのサービスの場合、コンソール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。

この節では、以下の項目に分けて説明しています。

- サービスログの特性について
- サービスログファイルの形式について
- サービスログオプションを定義、設定する
- サービスログを検索、表示する
- メッセージストアのログを使用したメッセージの追跡
- メッセージストアのログの例

サービスログの特性について

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ルール、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、管理、およびデフォルトのサービスごとに個別に設定できます (791 ページの「サービス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 21-4 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 21-4　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ーがアクセスできない場合など、サーバー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ほかのサーバー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	サーバー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	ユーザーがログインに失敗したり、セッションが終了したりした場合など、通知 (正常だが重要な状況) が発生したときに、イベントがログに記録されます。これがデフォルトのログレベルです。
Information	ユーザーがログオンやログオフを行なったり、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にのみ役立ちます。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、791 ページの「サービス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 21-5 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 21-5　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	ユーザーアカウントに関連するアクション (ユーザー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	サーバー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 (読み取りまたは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、794 ページの「サービスログを検索、表示する」を参照してください。

サービス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、およびサーバー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。791 ページの「サービスログオプションを定義、設定する」を参照してください。

サービスログファイルの形式について

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 21-6 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 21-6 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、`02/Jan/1999:13:08:21 -0700`
hostName	サーバー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` (789 ページの表 21-5 を参照)。
logLevel	イベントのログレベル。たとえば、`Notice` (788 ページの表 21-4 を参照)。
eventMessage	イベント固有の説明メッセージで、長さは任意です。たとえば、`Log created (894305624)`。

以下に、コンソール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。上記の例では次の 3 つの数です。
0 115 0。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ー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、794 ページの「サービスログを検索、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は、記録されるイベントのタイプに固有の形式です。つまり、各サービスのイベントメッセージに表示される内容は、各サービスによって定義されています。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。

サービス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

ログファイルのネーミングの形式 (*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切り分けることができます。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、サーバーを停止したり、新しいログファイルの作成をサーバー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。この場合、24 時間以内に 3.5M バイト以上のデータが記録されると、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、サーバー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かなり異なる場合があります。

ログオプションについて

メッセージストアのログ設定を制御するオプションは、コンソールまたはコマンド行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ーでは、ログレベルが低い場合 (Notice など)、週に何百 M 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蓄積された場合は、いくつかのログデータが失われる可能性があります。
- サーバー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ー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、サーバー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、サーバー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。

サービスログを検索、表示する

コンソールには、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。サーバー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 (787 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、サーバー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 (788 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、サーバー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 (790 ページの「サービスログファイルの形式について」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* 内の任意の文字 (例 : `[aeiou]`)
  [^*nnn*] *nnn* 内にない任意の文字 (例 : `[^aeiou]`)
  [*n-m*] *n-m* の範囲内の任意の文字 (例 : `[A-Z]`)
  [^*n-m*] *n-m* の範囲内にない任意の文字 (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 機能領域 (およびすべてのログレベル) を指定します。
- サーバーの機能に関する基本的な問題を調べるには、すべての機能領域 (および Critical ログレベル) を指定します。

サービスログの使用

この節では、`configutil` コマンドを使用して、またログの検索および表示のためのコンソールを使用して、サービスログを使用する方法について説明します。

➤ サービスログを syslog へ送信する

- 次のように `syslogfacility` オプションを指定して `configutil` コマンドを実行します。

  ```
  configutil -o logfile.service.syslogfacility -v value
  ```

  ここの *service* は `admin`、`pop`、`imap`、`imta`、または `http` に、*value* は `user`、`mail`、`daemon`、`local0` から `local7`、または none になります。

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。

➤ コンソールを使用してログオプションを設定する

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. 「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. 「新規ログエントリの作成」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。

➤ HTTP ログを無効にする

使用しているシステムが HTTP メッセージアクセス (Web メール) をサポートしていない場合は、次の変数を設定して HTTP のログを無効にできます。システムに Web メールサポート (たとえば、Messenger Express) が必要な場合は、これらの変数を設定しないでください。

- 次のように configutil コマンドを実行します。

```
configutil -o service.http.enable -v no
configutil -o service.http.enablesslport -v no
```

➤ サーバーログレベルを設定する

- 次のように configutil コマンドを実行します。

```
configutil -o logfile.service.loglevel -v level
```

ここの *service* は admin、pop、imap、imta、または http に、*loglevel* は Nolog、Critical、Error、Warning、Notice、Information、または Debug になります。

➤ ログファイルのディレクトリパスを指定する

- 次のように configutil コマンドを実行します。

```
configutil -o logfile.service.logdir -v dirpath
```

➤ 各ログの最大ファイルサイズを指定する

- 次のように configutil コマンドを実行します。

```
configutil -o logfile.service.maxlogfilesize -v size
```

size にはバイト数を指定します。

➤ サービスログローテーションのスケジュールを指定する

- 次のように configutil コマンドを実行します。

```
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サービスログファイル数を指定する

- 次のように configutil コマンドを実行します。

```
configutil -o logfile.service.maxlogfiles -v number
```

➤ 保存容量の上限を指定する

- 次のように configutil コマンドを実行します。

```
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を指定する

- 次のように configutil コマンドを実行します。

```
configutil -o logfile.service.minfreediskspace -v number
```

number にはバイト数を指定します。

➤ ログの保存期間を指定する

```
configutil -o logfile.service.expirytime -v number
```

number には秒数を指定します。

➤ 検索対象を指定し、結果を表示する

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. コンソールで、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”部分は、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。
   - 「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「目次」タブをクリックします。
3. ログに記録されたサービスの「目次」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。

メッセージストアのログを使用したメッセージの追跡

MTA がメッセージを追跡する方法と同様に、メッセージ ID によってメッセージを追跡するためにメッセージストアのログを使用できます。この方法でメッセージを追跡すると、メッセージのライフサイクルの重要なイベントを追跡できます。

メッセージストアログのメッセージを追跡するには、通常のログ設定に加えてメッセージの追跡も設定する必要があります。デフォルトでは、メッセージの追跡は有効になっていません。

注 メッセージの追跡は、ディスク領域を大量に使用します。十分なディスク容量がない限り、この機能を有効にしないでください。

メッセージストアのログは、次の操作を追跡できます。

- append (付加) - メッセージストアライブラリがメッセージをフォルダに追加する主な方法。append の追跡は、メッセージストアに入るメッセージを示します。
- fetch (フェッチ) - エンドユーザーのためにメッセージまたはメッセージの一部を取得する IMAP コマンド。メッセージの追跡のために、この意味はエンドユーザーが読むためにサービスがメッセージを取得する場合にまで拡大されます。

  メッセージの追跡では、メッセージの本体の部分が取得された場合にのみ本体の取り込みとみなされるように、メッセージのヘッダーが読み取られたときの追跡を避ける必要がある場合があります。
- expunge (消去) - IMAP の用語であり、この場合この意味は任意のサービスがユーザーのフォルダからメッセージを削除する場合にまで拡大されます。

➤ メッセージの追跡を有効にする

- 次のように `configutil` コマンドを実行します。

  ```
  configutil -o local.msgrace.active -v "yes"
  ```

  メッセージの追跡情報は、プロセスごとにデフォルトのログに書き込まれます。IMAP fetch は、`imap` ログファイルに書き込まれます。ims_master append は、`ims_master` チャネルログファイルに書き込まれます。

➤ メッセージの追跡を単一のログファイルにリダイレクトする

- メッセージの追跡ログを単一の「msgtrace」ログファイルにリダイレクトするには、configutil コマンドを使用してログファイルのパラメータを設定する必要があります。msgtrace ログファイルは、その他のログファイルとは異なり、ローカルに設定されます。次に例を示します。

```
configutil -o "local.logfile.msgtrace.buffersize" -v "0"
configutil -o "local.logfile.msgtrace.expirytime" -v "604800"
configutil -o "local.logfile.msgtrace.flushinterval" -v "60"
configutil -o "local.logfile.msgtrace.logdir" -v
"/opt/SUNWmsgsr/data/log"
configutil -o "local.logfile.msgtrace.loglevel" -v "Information"
configutil -o "local.logfile.msgtrace.logtype" -v "NscpLog"
configutil -o "local.logfile.msgtrace.maxlogfiles" -v "10"
configutil -o "local.logfile.msgtrace.maxlogfilesize" -v "2097152"
configutil -o "local.logfile.msgtrace.maxlogsize" -v "20971520"
configutil -o "local.logfile.msgtrace.minfreediskspace" -v "5242880"
configutil -o "local.logfile.msgtrace.rollovertime" -v "86400"
```

➤ メッセージ追跡ログの設定を解除する

- msgtrace ログファイルの設定を解除するには、configutil コマンドを使用してその設定へのすべての参照を削除します。次に例を示します。

```
configutil -o "local.logfile.msgtrace.buffersize" -v ""
configutil -o "local.logfile.msgtrace.expirytime" -v ""
configutil -o "local.logfile.msgtrace.flushinterval" -v ""
configutil -o "local.logfile.msgtrace.logdir" -v ""
configutil -o "local.logfile.msgtrace.loglevel" -v ""
configutil -o "local.logfile.msgtrace.logtype" -v ""
configutil -o "local.logfile.msgtrace.maxlogfiles" -v ""
configutil -o "local.logfile.msgtrace.maxlogfilesize" -v ""
configutil -o "local.logfile.msgtrace.maxlogsize" -v ""
configutil -o "local.logfile.msgtrace.minfreediskspace" -v ""
configutil -o "local.logfile.msgtrace.rollovertime" -v ""
```

➤ LMTP ログを設定する

- LMTP を使用する場合に、単一の「msgtrace」ログファイルを使用しない場合は、ローカルに tcp_lmtp_server ログファイルも設定する必要があります。LMTP を使用しない場合、またはメッセージの追跡を使用しない場合、または「msgtrace」ログファイルのメッセージ追跡を使用する場合、LMTP メッセージストアサイドログを初期化する必要はありません。LMTP はすでに MTA 情報を分けてログに記録しています。次に例を示します。

```
configutil -o "local.logfile.tcp_lmtp_server.buffersize" -v "0"
configutil -o "local.logfile.tcp_lmtp_server.expirytime" -v "604800"
configutil -o "local.logfile.tcp_lmtp_server.flushinterval" -v "60"
configutil -o "local.logfile.tcp_lmtp_server.logdir" -v "/opt/SUNWmsgsr/data/log"
configutil -o "local.logfile.tcp_lmtp_server.loglevel" -v "Information"
configutil -o "local.logfile.tcp_lmtp_server.logtype" -v "NscpLog"
configutil -o "local.logfile.tcp_lmtp_server.maxlogfiles" -v "10"
configutil -o "local.logfile.tcp_lmtp_server.maxlogfilesize" -v "2097152"
configutil -o "local.logfile.tcp_lmtp_server.maxlogsize" -v "20971520"
configutil -o "local.logfile.tcp_lmtp_server.minfreediskspace" -v "5242880"
configutil -o "local.logfile.tcp_lmtp_server.rollovertime" -v "86400"
```

メッセージストアのログの例

メッセージストアログ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。

メッセージストアのログの例 : 不良パスワード

ユーザーが無効なパスワードを入力すると、「ユーザーが見つからない」というメッセージではなく「認証」の失敗がログに記録されます。セキュリティ上の理由から、クライアントには「ユーザーが見つからない」というメッセージが渡されますが、ログには本当の理由 (無効なパスワード) が記録されます。

コード例 21-11 メッセージストアのログ : 無効なパスワード

```
[30/Aug/2004:16:53:05 -0700] vadar imapd[13027]:Account Notice:badlogin:
[192.18.126.64:40718] plaintext user1 authentication failure
```

メッセージストアのログの例 : 無効になったアカウント

次の例は、無効になったアカウントが原因でユーザーがログインできない理由を示しています。さらに、無効になったアカウントは「(inactive)」または「(hold)」として明示されます。

コード例 21-12 メッセージストアのログ : 無効になったアカウント

```
[30/Aug/2004:16:53:31 -0700] vadar imapd[13027]:Account Notice:badlogin:
[192.18.126.64:40720] plaintext user3 account disabled (hold)
```

メッセージストアのログの例 : 付加されたメッセージ

次の例は、メッセージがフォルダに追加されるたびに発生する付加メッセージを示しています。メッセージストアのログは、`ims_master` および `lmtp` チャネルを介してメッセージストアに入るすべてのメッセージを記録します。ユーザー ID、フォルダ、メッセージのサイズ、およびメッセージ ID の「append」を記録します。

コード例 21-13 メッセージストアのログ : 付加

```
[31/Aug/2004:16:33:14 -0700] vadar ims_master[13822]:Store Information:
append:user1:user/user1:659:<Roam.SIMC.2.0.6.1093995286.11265.user1@vadar.siroe.com>
```

メッセージストアのログの例 : クライアントが取得するメッセージ

メッセージストアのログは、クライアントがメッセージを取得すると、「fetch」メッセージを書き込みます。メッセージストアのログは、少なくとも 1 つの本体部分のすべてのクライアントのフェッチを記録します。ユーザー ID、フォルダ、およびメッセージ ID の「fetch」を記録します。

コード例 21-14 メッセージストアのログ : クライアントが取得するメッセージ

```
[31/Aug/2004:15:55:26 -0700] vadar imapd[13729]:Store
Information:fetch:user1:user/user1:<Roam.SIMC.2.0.6.1093051161.3655.user1@vadar.siroe.com>
```

メッセージストアのログの例 : フォルダから削除されるメッセージ

メッセージストアは、IMAP または POP メッセージがフォルダから削除される (ただし、システムからは削除されない) と、「expunge」メッセージを書き込みます。メッセージがユーザーまたはユーティリティのどちらによって消去されたかがログに記録されます。フォルダおよびメッセージ ID の「expunge」を記録します。

コード例 21-15 メッセージストアのログ : フォルダから削除されるメッセージ

```
31/Aug/2004:16:57:36 -0700] vadar imexpire[13923]:Store
Information:expunge:user/user1:<Roam.SIMC.2.0.6.1090458838.2929.user1@vadar.siroe.com>
```

メッセージストアのログの例 : 重複したログインメッセージ

1 つの msgtrace ログファイルに対するメッセージの追跡を設定する場合、imap および pop ログファイルに記録される通常の「login」メッセージは、msgtrace ファイルに重複して記録されます。

コード例 21-16 メッセージストアのログ : ログイン

```
[30/Aug/2004:16:53:13 -0700] vadar imapd[13027]:Account Information:login
[192.18.126.64:40718] user1 plaintext
```

第 22 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。

監視手順に関連する項目は、第 23 章「Messaging Server を監視する」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 5 章から第 10 章と、『Sun Java System Messaging Server Administration Reference』の MTA 設定およびコマンド行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。

- 804 ページの「MTA 設定をチェックする」
- 805 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」
- 805 ページの「危険なファイルの所有権をチェックする」
- 806 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」
- 807 ページの「ログファイルをチェックする」
- 808 ページの「チャネルプログラムを手動で実行する」
- 809 ページの「個々のチャネルを起動および停止する」
- 810 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。詳細は、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行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 設定の構文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

MTA メッセージキューディレクトリ (通常、*msg_svr_base*/data/queue/) にメッセージがあるかどうかをチェックします。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 のようなコマンド行ユーティリティを使用します。imsimta qm の詳細については、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行ユーティリティの章と 857 ページの「imsimta qm counters」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 ログの詳細については、762 ページの「MTA メッセージおよび接続のログの管理」を参照)。次に、ディレクトリ /*msg_svr_base*/log/ 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

Messaging Server をインストールしたときに、メールサーバーのユーザーアカウント (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。

/*msg_svr_base*/data/queue/
/*msg_svr_base*/log/
/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 /opt/SUNWmsgsr/data/queue
drwx------ 6 inetuser bin 512 Feb 7 09:32 /opt/SUNWmsgsr/data/queue

ls -l -p -d /opt/SUNWmsgsr/log/imta
drwx------ 2 inetuser bin 1536 Mar 10 09:00 /opt/SUNWmsgsr/log/imta

ls -l -p -d /opt/SUNWmsgsr/imta/tmp
drwx------ 2 inetuser bin 512 Feb 7 10:00 /opt/SUNWmsgsr/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、/*msg_svr_base*/data/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 /opt/SUNWmsgsr/data/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、着信メッセージを処理する常駐サーバープロセスを含むものもあります。このようなサーバーは、チャネルのスレーブ (着信) 方向を扱います。MTA ディスパッチャは、そのような MTA サーバーの作成を行います。ディスパッチャの設定オプションは、サーバーの可用性、作成されたサーバーの数、各サーバー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 サーバー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。次に例を示します。

```
imsimta process

USER       PID  S     VSZ     RSS    STIME      TIME     COMMAND
inetuser   9567  S   18416    9368   02:00:02   0:00   /opt/SUNWmsgsr/lib/tcp_smtp_server

inetuser   6573  S   18112    5720    Jul_13    0:00    /opt/SUNWmsgsr/lib/job_controller

inetuser   9568  S   18416    9432    02:00:02  0:00   /opt/SUNWmsgsr/lib/tcp_smtp_server

inetuser   6574  S   17848    5328    Jul_13      0:00    /opt/SUNWmsgsr/lib/dispatcher
```

ジョブコントローラがない場合、/*msg_svr_base*/data/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 の詳細については、『Sun Java System Messaging Server Administration Reference』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、/*msg_svr_base*/data/log 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、msg-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、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行ユーティリティの章を参照してください。

注 実行が必要なプログラムを実行する (exec()) 前に子プロセスをフォークしている (fork()) 場合を除き、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を表示しないようにしてください。ただし、このような重複が発生しているのは非常に短い期間です。

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 /*msg_svr_base*/log に作成されます。表 22-1 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 22-1 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 (通常はサーバー) 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 ms-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ー固有の情報であり、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	各ログファイルの作成時には、同一のチャネルが作成した過去のログが上書きされないよう、ファイル名に固有の ID (*uniqueid*) が付加されています。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、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行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 (*/msg_svr_base*`/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、『Sun Java System Messaging Server Administration Reference』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。

`msimta 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 22-1 で示すように、`imsimta submit` でディレクトリ */msg_svr_base*`/log` 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。

`msimta submit` コマンドと `imsimta run` コマンドの構文、オプション、パラメータ、例の詳細については、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行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` コマンドの詳細については、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行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。532 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 */msg_svr_base*/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ルールを追加します。

```
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 */msg_svr_base*/config/mappings) にある PORT_ACCESS マッピングテーブルに以下のアクセスルールを追加します。

```
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、必ず上記のルール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 (804 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)。以下の手順で、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 */msg_svr_base*/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、メッセージ ID が mail.log_current ファイルにあるヘッダー行に表示されます。

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 サーバーを再起動します。
4. エンドユーザー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、/*msg_svr_base*/log 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。

   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については、763 ページの「MTA ログエントリの形式について」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したりする方法を説明します。詳細については、809 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 813 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 /*msg_svr_base*/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 /*msg_svr_base*/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、810 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 サーバーを再起動します。

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および停止します。これらのキーワードの使用の詳細については、809 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ために、エンドユーザー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、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、/*msg_svr_base*/data/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* 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. /*msg_svr_base*/log 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、手順 5 ～ 7 を繰り返します。

   手順 7 でコピーしたログファイルは、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 */msg_svr_base*/config 内の 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 */msg_svr_base*/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 サーバーを再起動します。

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

   すべてのファイルには、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値とメッセージ ID 値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 サーバー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り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り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りした場合は、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ー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。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 (810 ページの「メッセージパスにあるチャネルを識別する」、811 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ー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。

TLS の問題

SMTP ダイアログの間に、`STARTTLS` コマンドが次のエラーを返した場合で、

```
454 4.7.1 TLS library initialization failure
```

かつ証明書をインストール済みで pop/imap アクセスを試みている場合は、次のことを確認します。

- 証明書の保護と所有権が、`mailsrv` アカウントでファイルにアクセスできるように設定されている
- 証明書が保存されているディレクトリに、`mailsrv` アカウントでその中のファイルにアクセスできるような保護と所有権が設定されている。

保護を変更し、証明書をインストールしたら、次のコマンドを実行します。

```
stop-msg dispatcher
start-msg dispatcher
```

再起動でも問題は解決するはずですが、完全にシャットダウンして証明書をインストールしてから操作をやり直すことをお勧めします。

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、以下の手順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、着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ーではなく 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 サーバーから MTA の SMTP サーバーへの置換は、Messaging Server のインストールの際に実行されます。詳細は、『Sun Java Enterprise System 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 サーバーの場合、SMTP サーバー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 サーバー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る)。`imsimta process` コマンドを使用して、SMTP サーバー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。詳細は、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行ユーティリティの章を参照してください。

ディスパッチャ (SMTP サーバー) が起動しない

ディスパッチャが起動しない場合は、まず dispatcher.log-* に関連するエラーメッセージがあるかどうか確認します。/tmp/.SUNWmsgsr.dispatcher.socket ファイルの作成やアクセスに関する問題がログで示されている場合は、/tmp 保護が 1777 に設定されていることを確認します。これは、権限で次のように表示されます。

```
drwxrwxrwt    8 root    sys           734  Sep 17 12:14      tmp/
```

また、.SUNWmsgsr.dispatcher.socket ファイルの ls -l を実行して、適切な所有権を確認します。たとえば、このファイルが root によって作成された場合は、inetmail でアクセスすることはできません。

SUNWmsgsr.dispatcher.file を削除しないでください。また、存在しない場合は作成しないでください。このファイルはディスパッチャによって作成されます。保護が 1777 に設定されていないと、ディスパッチャはソケットファイルを作成およびアクセスできないため、起動または再起動しません。また、Messaging Server とは関連のないほかの問題が発生している可能性もあります。

着信 SMTP 接続時のタイムアウト

着信 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、着信 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. 同時に許可する着信 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、ユーザーは 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。次に例を示します。

```
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 (Sun Java System Messaging
Server 6.1 (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) が応答を不正としない場合は、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。

3. 220 個の見出しの応答が遅い場合や、SMTP サーバーで 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
```

変更後は次のようになります。

```
hosts:dns nis [NOTFOUND=return] files
```

/etc/nsswitch.conf ファイルでこのような変更を行うと、複数の SMTP サーバーで不要な検索を実行する必要がなく、少数の SMTP サーバーでメッセージを処理するため、パフォーマンスを向上させることができます。

4. 着信 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着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記述されていることもあります。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)。363 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 サーバーのポート 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 Messaging Server が稼働している場合は、手順 1 と手順 2 を省略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、メモリ内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 */msg_svr_base*/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 ジョブコントローラ設定ファイル
!
! グローバルデフォルト
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については、199 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。823 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、あるいはユーザー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ます。

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バウンスのレポート) 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バウンスするときに、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ーまたはチャネル間でバウンスされていることを MTA が検出すると、配信は停止され、メッセージは */msg_svr_base*/data/queue/*channel* にある、サフィックスが .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、各サーバーまたはチャネルがメッセージの配信をほかのサーバーやチャネルが担当するとみなしたときです。

たとえば、エンドユーザーが、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、サーバー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。

imsimta qm release を実行するか、または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。

これらの手順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。次に例を示します。

```
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、『Sun Java System Messaging Server Administration Reference』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 サーバー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 サーバー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 サーバー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 サーバー (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。フィルタはサーバー上に保存されて評価されるため、SSR (Server-Side Rules) と呼ばれることがよくあります。

この節では、SSR に関する以下の情報について説明します。

- 825 ページの「SSR ルールをテストする」
- 826 ページの「一般的な構文の問題」

568 ページの「ユーザーレベルのフィルタをデバッグするには」も参照してください。

SSR ルールをテストする

- 以下のコマンドを使用して、MTA のユーザー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 */msg_svr_base*/config にある option.dat ファイルに mm_debug=5 を追加します。

一般的な構文の問題

- フィルタに構文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
```

アドレスのローカル部分または受信フィールド内のアスタリスク

MTA は、アドレスのローカル部分および作成する受信フィールド内の ASCII 文字のみではなく 8 ビット文字をチェックし、アスタリスクで置き換えるようになりました。

一般的なエラーメッセージ

MTA が起動に失敗すると、コマンド行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。804 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については、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行ユーティリティの章と、第 5 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。

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、サーバー、またはブラウザ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。

bad equivalence for alias. . .

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。

cannot open alias include file. . .

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。

duplicate aliases found. . .

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。

duplicate host in channel table. . .

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`) の書き換えルール (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ルール (左側) が複数あると、MTA はそれらの書き換えルールを、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 (書き換えルールの部分) 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。

duplicate mapping name found. . .

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。

mapping name is too long. . .

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。

error initializing ch_ facility: compiled character set version mismatch

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、『Sun Java System Messaging Server Administration Reference』を参照してください。

error initializing ch_ facility: no room in. . .

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` の詳細については、『Sun Java System Messaging Server Administration Reference』の MTA コマンド行ユーティリティの章を参照してください。

local host alias or proper name too long for system. . .

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構文エラー (書き換えルール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構文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ルール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。

no equivalence addresses for alias. . .

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。

no official host name for channel. . .

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については、『Sun Java System Messaging Server Administration Reference』の MTA の設定およびコマンド行ユーティリティの章と、第 12 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ルール部分には入れることはできません。

official host name is too long

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 128 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、書き換えルール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。次に例を示します。

Original l Channel:

```
! ローカル /var/mail ストアへの配信チャネル
l subdirs 20 viaaliasrequired maxjobs 7 pool LOCAL_POOL
walleroo.pocofronitas.thisnameismuchtoolongandreallymakesnosensebutiti
sanexample.monkey.gorilla.orangutan.antidisestablismentarianism.newt.s
alamander.lizard.gecko.komododragon.com
```

Create Place Holder:

```
! ローカル /var/mail ストアへの配信チャネル
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。使用法と構文の詳細については、『Sun Java System Messaging Server Administration Reference』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構文エラー (書き換えルール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、書き換えルール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構文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ルール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 サーバー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 サーバー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

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ー、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。『Sun Java System Messaging Server Administration Reference』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。805 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、805 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。

不正なホストまたはドメインエラー

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ルールがありません。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るが、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ー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、358 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 23 章

Messaging Server を監視する

一般的に、十分に計画され的確に設定されたサーバーは、管理者の手を煩わすことなく動作を続けます。したがって、管理者の役割は、サーバーが問題の兆候を示していないか、監視することです。この章では、Messaging Server の監視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 22 章「MTA のトラブルシューティング」を参照してください。

自動監視と自動再起動

Messaging Server には、サービスを透過的に監視する方法と、サービスに障害が発生したり、応答しなくなったりした場合 (サービスがハングアップまたはフリーズした場合) にサービスを自動的に再起動する機能が用意されています。この機能ですべてのメッセージストア、MTA、および MMP サービス (IMAP、POP、HTTP、ジョブコントローラ、ディスパッチャ、MMP サーバーなど) を監視できます。この機能は、SMS サーバーや TCP/SNMP サーバーなどのほかのサービスは監視しません (TCP/SNMP はジョブコントローラで監視される)。詳細は、113 ページの「障害が発生したサービスや応答がないサービスの自動再起動」および 859 ページの「msprobe および watcher 関数を使用した監視」を参照してください。

毎日の監視作業

毎日の実施を必要とする作業のうち、特に重要なものは、ポストマスターメールのチェック、ログファイルの監視、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ーは、ポストマスター宛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ルールは 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ーまたはメーリングリスト宛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ー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、ユーザーは、ポストマスターアドレス宛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、ローカルユーザーからはサーバー応答時間に関するメールを受信し、ほかのサーバー管理者からはサーバー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ー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ー応答の不良) を送ることもできます。

ログファイルを監視および管理する

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、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、*msg_svr_base*/data/log にあります。ログファイルは定期的に監視する必要があり、サーバーに問題がある場合は特に必要です。

ログ記録はサーバー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、サーバー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。サーバーのログポリシーの定義の詳細については、第 21 章「ログの管理」を参照してください。

msprobe ユーティリティを設定する

msprobe ユーティリティは、監視関数や再起動関数を自動的に実行します。詳細は、859 ページの「msprobe および watcher 関数を使用した監視」を参照してください。

システムのパフォーマンスを監視する

この章では、Messaging Server の監視機能に注目しています。ただし、サーバーが動作しているシステムも、同時に監視することが必要です。適切に設定されたサーバーであっても、設定が適切ではないシステム上では、本来の性能を発揮しないことがあるからです。また、サーバーエラーの発生は、ハードウェアの処理能力がメールシステムの動作には十分ではない場合もあります。この章では、システムパフォーマンスの監視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。パフォーマンスを監視する手順を以下に示します。

- 837 ページの「終端間メッセージ配信時間を監視する」
- 838 ページの「ディスク容量を監視する」
- 840 ページの「CPU 使用状況を監視する」

終端間メッセージ配信時間を監視する

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、サーバー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監視するには

- メッセージを送信および受信する機能を使用します。サーバー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。850 ページの「immonitor-access」を参照してください。

ディスク容量を監視する

ディスクの空き容量の不足は、メールサーバーで発生する問題や故障のうち、特に頻繁におきる原因の 1 つです。MTA キューやメッセージストアへ書き込むとき、そのための容量が不足していると、メールサーバーにエラーが発生します。さらに、ログファイルを監視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。

メッセージストアパーティションは、新しいメッセージがメールボックスに配信されるたびに大きくなります。たとえば、メッセージストアに制限容量を課さない場合、メッセージストアがパーティションに利用できるディスク容量より大きくなることがあります。ディスク容量が不足するもう 1 つの原因は、MTA メッセージキューが大きくなりすぎることです。3 番目の原因としては、ログファイル監視機能に問題が発生し、ログファイルが制御できないほど大きくなってしまう場合が考えられます。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ることに注意してください。

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したりすることがあります。

メッセージストアパーティションが一杯になると、メッセージアクセスデーモンが失敗したり、メッセージストアデータが壊れたりすることがあります。`imexpire` や `reconstruct` などのメッセージストア保守ユーティリティは、破損を修復したり、ディスク使用量を削減したりすることができます。ただし、それらのユーティリティはさらにディスク容量を必要とし、ディスク全体を占有したパーティションの修復はダウン時間の発生原因になります。

ディスク容量を監視するには

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監視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監視する必要があり、その容量を監視する方法は異なることがあります。

Messaging Server は、メッセージストアディスクの使用量を監視し、パーティションがすべての利用可能なディスク容量を使い果たすのを防止する手段を提供します。

次の手順で、メッセージストアのディスク容量の使用状況を監視できます。

- メッセージストアのディスク使用量を監視するためのパラメータを設定します。

- ディスク使用量のしきい値に達したら、メッセージストアパーティションをロックします。

詳細は、次の節の「メッセージストアを監視する」と「メッセージストアのパーティションを監視する」を参照してください。

メッセージストアを監視する

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監視するには、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
- alarm.diskavail.msgalarmdescription

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監視する頻度と、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、システムがディスク容量を 600 秒毎に監視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
```

これらのパラメータの詳細については、862 ページの表 23-6 を参照してください。

メッセージストアのパーティションを監視する

利用可能なディスク容量の指定された割合をパーティションが超過したら、メッセージをメッセージストアパーティションに配信するのを停止できます。このためには、2 つの configutil パラメータを使って、その機能を有効にし、ディスク使用量のしきい値を指定します。

この機能を使用して、メッセージストアデーモンはパーティションのディスク使用量を監視します。ディスクの使用量が増加するにつれ、ストアデーモンは動的にパーティションをチェックする頻度 (100 分に 1 回～ 1 分に 1 回) を増やします。

ディスク使用量が指定されたしきい値を超えた場合、ストアデーモンは次のようにします。

- パーティションをロックします。着信メッセージは MTA メッセージキューに保持されますが、メッセージストアパーティション内のメールボックスには配信されません。
- メッセージをデフォルトのログファイルに記録します。

- ポストマスターに電子メール通知を送信します。configutil パラメータの alarm.msgalarmnoticercpt を設定して、電子メールの受信者を変更できます。

ディスクの使用量がしきい値を下回ると、パーティションのロックが解除され、メッセージはまたストアに配信されるようになります。

次のような configutil パラメータがあります。

- local.store.checkdiskusage は、パーティション監視機能を有効にします。

  許容可能な値 : yes、no

  デフォルト値 : yes

- local.store.diskusagethreshold は、ディスク使用量のしきい値を指定します。local.store.diskusagethreshold の値は、1 ～ 99 のパーセンテージです。

  デフォルト値 : 99

ディスク使用量のしきい値には、ローカルメッセージストアを再パーティションしたり、さらに多くのディスク容量を割り当てたりするための時間が確保できる低いパーセンテージに設定する必要があります。

たとえば、パーティションが 1 時間に 2 パーセントの割合でディスク容量を使用していき、またローカルメッセージストアにさらにディスク容量を割り当てるのに 1 時間かかるとします。この場合は、ディスク使用量のしきい値を 98 パーセント未満の値に設定する必要があります。

MTA キューとログ領域を監視する

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監視する必要があります。

ログ領域の管理については、第 21 章「ログの管理」を参照してください。mail.log ファイルの監視方法については、762 ページの「MTA メッセージおよび接続のログの管理」を参照してください。

CPU 使用状況を監視する

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、ユーザーのログインに時間がかかり、配信速度が遅くなります。

CPU 使用状況を監視するには

CPU 使用状況の監視は、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 を監視する

この節には、以下の項があります。

メッセージキューのサイズを監視する

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については、196 ページの「チャネルメッセージキュー」、819 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 820 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。
- ユーザー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監視するには

メッセージキューを監視する最良の方法は、`imsimta qm` を使用することです。857 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 (*msg_svr_base*`/data/queue/`) 内のファイルの数を監視することもできます。ファイルの数はサイト固有であるので、「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監視する

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 サーバーの故障や、接続への応答時のリモートサーバー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` レコードが `mail.log_current` に表示されます。

配信エラーの頻度を監視するには

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_svr_base*`/data/log/mail.log_current` ファイル内のレコードを確認します。次に例を示します。

```
mail.log:06-Oct-2003 00:24:03.66 501d.0b.9 ims-ms    Q  5
durai.balusamy@Sun.COM rfc822;durai.balusamy@Sun.COM durai@ims-ms-daemon
<00ce01c38bda$c7e2b240$6501a8c0@guindy> Mailbox is busy
```

受信 SMTP 接続を監視する

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。

- 外部ユーザーがメールをリレーしようとしている。
- 外部ユーザー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

- **外部ユーザーによるメールのリレー** - 表面的には問題発生の兆候はありません。
- **サービス拒否攻撃** - 外部のメッセージ要求により SMTP サーバーを過負荷にしようとする試みです。

受信用 SMTP 接続を監視するには

- **外部ユーザー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_svr_base*`/log/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 サーバーに接続しているユーザーとその人数を調べるには、netstat コマンドを実行し、SMTP ポートの接続数 (デフォルトは 25) を確認します。次に例を示します。

```
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たりすることがあります。さらに、SMTP サーバー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、サーバー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 サーバー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監視する

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監視するには

`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。806 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。

LDAP Directory Server を監視する

この節には、以下の項目があります。

slapd を監視する

LDAP ディレクトリサーバー (`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

- クライアントの 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 を監視するには

- `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。
- ユーザー検索時の ns-slapd 応答時間をチェックします。
- コンソールを表示して slapd を監視します。
- 850 ページの「immonitor-access」も参照してください。

メッセージアクセスを監視する

この節には、以下の項があります。

imapd、popd、および httpd を監視する

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、監視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

接続が拒否されるか、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
```

クライアントに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。

```
Client 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 を監視するには

- watcher と msprobe によって監視することができます。113 ページの「障害が発生したサービスや応答がないサービスの自動再起動」および 859 ページの「msprobe および watcher 関数を使用した監視」を参照してください。
- SNMP によって監視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監視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。サーバー情報は、Network Services Monitoring MIB にあります。

- ログファイルをチェックします。

  msg_svr_base/log/*service* ディレクトリ (*service* は http、IMAP、POP のいずれか) を確認します。このディレクトリで、ログファイルの数を確認します。ファイル名の 1 つは、*service* の名前 (imap、pop、http) で、残りのファイル名はサービスの名前にシーケンス番号および日付が連結されたものです。次に例を示します。

  ```
  imap  imap.29.1010221593 imap.31.1010394412  imap.33.1010567224
  ```

  サービス名だけのファイルは、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、シーケンス番号の一番大きいファイルが次に新しいファイルです (第 21 章「ログの管理」を参照)。

  サーバーが停止した場合は、以下のように表示されることがあります。

  imap.12.1065431243:[07/Oct/2003:01:15:43 -0700] gotmail-2 imapd[20525]: General Warning: Sun Java System Messaging Server IMAP4 6.1 (built Sep 24 2003) shutting down

- `counterutil` を使ってチェックできます。850 ページの「counterutil」および『Sun Java System Messaging Server Administration Reference』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。
- 861 ページの「警告メッセージ」に記載されているサーバー応答設定パラメータを設定することによって、指定したサーバー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。
- 850 ページの「immonitor-access」を参照してください。

stored を監視する

`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` の詳細については、『Sun Java System Messaging Server Administration Reference』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 を監視するには

- 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_svr_base*/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。次に例を示します。

  ```
  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
  inetuser 28250     1   0    Jan 05 ?        8:44 /opt/SUNWmsgsr/lib/stored -d
  ```

- *msg_svr_base*/store/mboxlist 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_svr_base*/config 内の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ます。1 分ごとにタイムスタンプが付けられます。
  stored.lcu - データベースログのクリーンアップごとに押されます。5 分ごとにタイムスタンプが付けられます。
  stored.per - ユーザー単位のデータベース書き込み時に押されます。60 分ごとにタイムスタンプが付けられます。

- デフォルトログファイルの *msg_svr_base*/log/default/default 内の stored メッセージをチェックします。

- watcher と msprobe によって監視することができます。113 ページの「障害が発生したサービスや応答がないサービスの自動再起動」および 859 ページの「msprobe および watcher 関数を使用した監視」を参照してください。

メッセージストアを監視する

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ーの分散、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。以下の項目で、これらの問題について説明します。

- 838 ページの「ディスク容量を監視するには」
- 848 ページの「メッセージストアデータベースのロック状態を監視する」
- 849 ページの「mboxlist ディレクトリ内のデータベースログファイルの数を監視する」
- 630 ページの「制限容量を監視するには」

メッセージストアデータベースのロック状態を監視する

データベースロックの状態は、さまざまなサーバー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。したがって、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監視するには

`counterutil -o db_lock` コマンドを使用します。

mboxlist ディレクトリ内のデータベースログファイルの数を監視する

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_svr_base*/store/mboxlist ディレクトリ内) 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ー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監視するには

msg_svr_base/store/mboxlist ディレクトリを調べて、2 つまたは 3 つのファイルしかないことを確認してください。

監視用のユーティリティとツール

監視には、以下のツールを利用できます。

immonitor-access

immonitor-access は、Messaging Server のコンポーネントやプロセスのステータスを監視します。対象となるコンポーネントやプロセスには、メール配信 (SMTP サーバー)、メッセージアクセスとストア (POP サーバーおよび IMAP サーバー)、ディレクトリサービス (LDAP サーバー)、および HTTP サーバーがあります。このユーティリティでは、さまざまなサービスの応答時間と、メッセージの送受信にかかるラウンドトリップの総時間を測定します。ディレクトリサービスは、指定のユーザーをディレクトリ内で検索し、応答時間を測定することで監視します。メール配信はメッセージを送信することで (SMTP) 監視し、メッセージアクセスおよびストアはそのメッセージを受信することで監視します。HTTP サーバーの監視は、起動して実行中であるかどうかを調べることだけに制限されます。

手順については、『Sun Java System Messaging Server Administration Reference』を参照してください。

stored

stored ユーティリティはサーバー上で保守タスクを実行し、監視も実行できます。ただし、監視タスクは msprobe で行うことをお勧めします。859 ページの「msprobe および watcher 関数を使用した監視」を参照してください。

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。

```
# /opt/SUNWmsgsr/sbin/counterutil -l
Listing registry (/opt/SUNWmsgsr/data/counter/counter)
numobjects = 11
refcount = 1
created = 25/Sep/2003:02:04:55 -0700
modified = 02/Oct/2003:22:48:55 -0700
       entry = alarm
       entry = diskusage
       entry = serverresponse
       entry = db_lock
       entry = db_log
       entry = db_mpool
       entry = db_txn
```

```
entry = imapstat
entry = httpstat
entry = popstat
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 コマンドの使用法については、『Sun Java System Messaging Server Administration Reference』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 23-1　counterutil alarm 統計

サフィックス	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	現在の監視値。
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 サフィックスの意味を表 23-2 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 23-2　counterutil imapstat 統計

サフィックス	説明
currentStartTime	現在の IMAP サーバー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 サーバーが処理した同時接続の最大数。
numConnections	現在の IMAP サーバー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 サーバーが処理した失敗した接続の数。
numFailedLogins	現在の IMAP サーバーが処理した失敗したログインの数。
numGoodLogins	現在の IMAP サーバー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 23-3　counterutil diskstat 統計

サフィックス	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。

サーバー応答の統計

コマンド counterutil -o serverresponse は以下の情報を生成します。この情報は、サーバーが稼働中かどうかと、サーバーの応答速度をチェックする際に便利です。

表 23-4　counterutil serverresponse 統計

サフィックス	説明
http.laststattime	最後に http サーバー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 サーバー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 サーバー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 サーバー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 サーバー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。

ログ記録はサーバーのパフォーマンスに影響を与えることがあるため、サーバー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。詳細は、第 21 章「ログの管理」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 21 章「ログの管理」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。

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 サーバー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 サーバー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)。

`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 `inetuser` 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 854 ページの「imsimta counters」に記載されているものと同じです。使用法については、『Sun Java System Messaging Server Administration Reference』を参照してください。

次に例を示します。

```
# imsimta counters -create
# imsimta qm counters show
Channel                 Messages    Recipients     Blocks
----------------------  ----------  ----------    ----------
tcp_intranet
   Received               13077        13859         264616
   Stored                    92           91           -362
   Delivered              12985        13768         264978
   Submitted               2594         2594           3641
...
```

MTA を再起動するたびに、**`# imsimta counters -create`** を実行する必要があります。

SNMP を使用した MTA の監視

Messaging Server では、SNMP (Simple Network Management Protocol) を利用したシステム監視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、Messaging Server の特定の部分を監視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 imquotacheck

imquotacheck 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監視することができます。imquotacheck 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。

たとえば次のコマンドでは、全ユーザーの制限容量に関する情報を一覧表示します。

```
% imquotacheck
--------------------------------------------------------------------------
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、ユーザー sorook の制限容量の使用状況を示します。

```
% imquotacheck -u sorook
--------------------------------------------------------------------------
quota usage for user sorook
--------------------------------------------------------------------------
diskquota      size(K)    %use    msgquota      msgs    %use    user

no quota       1487               no quota      305             sorook
```

msprobe および watcher 関数を使用した監視

Messaging Server には、各種のシステムサービスを監視するために、watcher と msprobe という 2 つのプロセスが用意されています。watcher は、サーバーのクラッシュを監視し、必要に応じて再起動を行います。msprobe は、サーバーのハングアップ (応答なし) を監視します。特に、msprobe は次の状態を監視します。

- **サーバー応答時間** : msprobe は有効になっているサーバーにそのプロトコルコマンドを使って接続し、応答時間を測定します。応答時間が警告のしきい値を超えると、警告メッセージが送信されます (861 ページの「警告メッセージ」を参照)。autorestart が有効になっている場合、msprobe がサーバーに接続できないか、サーバーの応答時間が指定のタイムアウト期間を超えると、サーバーは再起動されます。サーバーの応答時間はカウンタデータベースに記録され、デフォルトのログファイルに記録されます。サーバーの応答時間の統計を表示するには、counterutil を使用します (850 ページの「counterutil」を参照)。

  msprobe によって監視されるサーバーは、imap、pop、http、cert、job_controller、smtp、lmtp、mmp、および ens です。smtp または lmtp が応答しないときは、ディスパッチャが再起動されます。ens は自動的に再起動できません。

- **ディスク使用量** : msprobe はメッセージストアパーティションごとのディスクの利用度と使用量をチェックします。特に、メッセージストアの mboxlist データベースディレクトリと MTA キューディレクトリをチェックします。ディスク使用量が設定したしきい値を超えると、警告メッセージが送信されます。ディスクのサイズと使用量はカウンタデータベースに記録され、デフォルトのログファイルに記録されます。管理者は、counterutil ユーティリティ (850 ページの「counterutil」を参照) を使用してディスク使用量の統計を表示できます。

- **メッセージストアの mboxlist データベースログファイルの累積** : ログファイルの累積は、mboxlist データベースのエラーを示しています。msprobe はアクティブなログファイルの数をカウントし、その数がしきい値よりも大きい場合は、重大エラーメッセージを default ログファイルに記録して、管理者にサーバーを再起動することを通知します。autorestart が有効になっている (local.autorestart が yes に設定されている) 場合は、ストアデーモンが再起動されます。

watcher と msprobe は、表 23-5 に示す configutil オプションによって制御されます。詳細については、113 ページの「障害が発生したサービスや応答がないサービスの自動再起動」を参照してください。

表 23-5　msprobe および watcher の configutil オプション

オプション	説明
local.watcher.enable	サービスの障害を監視する watcher を有効にします。対象となるサービスは、IMAP、POP、HTTP、ジョブコントローラ、ディスパッチャ、メッセージストア (stored)、imsched、および MMP です。LMTP/SMTP サーバーはディスパッチャによって監視され、LMTP/SMTP クライアントは job_controller によって監視されます。それぞれの障害について、エラーメッセージをデフォルトのログファイルに記録します。デフォルト : on
local.autorestart	サーバーの自動再起動を有効にします。障害の発生したサービスまたはハングアップしたサービスを自動的に再起動します。デフォルト : no
local.autorestart.timeout	再試行失敗のタイムアウト。ここに指定した時間内でサーバーに 3 回以上障害が発生すると、システムはサーバーの再起動を試行しなくなります。値 (秒単位で指定) は、msprobe の間隔 (local.schedule.msprobe) よりも長い時間に設定する必要があります。デフォルト : 600 秒
local.schedule.msprobe	msprobe の実行スケジュール。crontab 形式でスケジュールを示す文字列 (617 ページの表 18-10 を参照)。デフォルトは 600 秒です。
service.readtimeout	サーバーが再起動されるまでのデフォルトのタイムアウト。 デフォルト : smtp/lmtp の場合は 120 秒、その他のプロトコルの場合は 30 秒
local.probe.service.timeout	特定のサーバーが再起動されるまでのタイムアウト。*service* は、imap、pop、http、cert、job_controller、smtp、lmtp、mmp または ens のいずれかになります。 デフォルト : service.readtimeout の値を使用する
local.probe.warningthreshold	警告メッセージが default ログファイルに記録されるまでのサーバーの無応答時間 (秒)。 デフォルト : 5 秒
local.probe.service.warningthreshold	警告メッセージが default ログファイルに記録されるまでの特定のサーバーの無応答時間 (秒)。*service* は、imap、pop、http、cert、job_controller、smtp、lmtp、mmp または ens のいずれかになります。 デフォルト : local.probe.warningthreshold の値を使用する

表 23-5　msprobe および watcher の configutil オプション

オプション	説明
local.queuedir	キューサイズが alarm.diskavail.msgalarmthreshold によって定義されたしきい値を超えているかどうかを確認するための MTA キューディレクトリ。 デフォルト : なし
local.schedule.msprobe	msprobe の実行スケジュール。この値は crontab 形式でスケジュールを示す文字列です。 デフォルト : 600 秒
service.readtimeout	サーバーを再起動するまでのサーバーの無応答時間。local.schedule.msprobe を参照してください。 デフォルト : 10 秒

警告メッセージ

msprobe は、電子メールメッセージの形式で警告をポストマスター (846 ページを参照) に送信して、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送信され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: /opt/SUNWmsgsr
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
```

msprobe でディスクおよびサーバーのパフォーマンスを監視する頻度と、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。このためには、configutil コマンドを使用して警告パラメータを設定します。表 23-6 に、有用な警告パラメータとそのデフォルト設定を示します。完全なリストについては、『Sun Java System Messaging Server Administration Reference』を参照してください。

表 23-6 有用な警告メッセージの configutil パラメータ

パラメータ	説明 (括弧内はデフォルト)
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	(利用可能なメールパーティションのディスク容量のパーセンテージ) ディスク利用度の警告についての説明フィールドのテキスト。
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 に設定します。
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ます。
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します。
alarm.diskavail.msgalarmwarninginterval	(24). (24) ディスク利用度の警告が繰り返される間隔 (時)。
alarm.serverresponse.msgalarmdescription	(サーバーの応答時間を表す秒数)。サーバーの応答警告についての説明フィールドのテキスト。
alarm.serverresponse.msgalarmstatinterval	(600) サーバー応答のチェックの間隔 (秒)。サーバーの応答を確認しない場合は、0 に設定します。
alarm.serverresponse.msgalarmthreshold	(10) サーバー応答時間 (秒) がこの値を超えると、警告が発行されます。
alarm.serverresponse.msgalarmthresholddirection	(1) サーバー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します。
alarm.serverresponse.msgalarmwarninginterval	(24) サーバー応答警告が繰り返される間隔 (時)。

付録 A

SNMP サポート

Messaging Server では、SNMP (Simple Network Management Protocol) を利用したシステム監視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、Messaging Server の特定の部分を監視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。Messaging Server の監視の詳細については、第 23 章「Messaging Server を監視する」を参照してください。

この付録では、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。ただし、この付録では、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 の詳細については、RFC 2788 および RFC 2789 を参照してください。

この付録には、以下の節があります。

SNMP の実装

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 などのサーバーのネットワークサービスを監視するためのものです。Mail Monitoring MIB は MTA を監視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、その履歴を監視することができます。アクティブ状態の監視では、現在キューに入っているメッセージと開いている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、開いているネットワーク接続のソース IP アドレスなどです。一方、履歴の監視からは、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 監視機能の詳細については、RFC 2788 および RFC 2789 を参照してください。

SNMP は、Solaris 8 および 9 と、Java Enterprise System が対応しているすべてのバージョンの Microsoft Windows が稼働しているプラットフォームでサポートされています。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 のネイティブ 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。

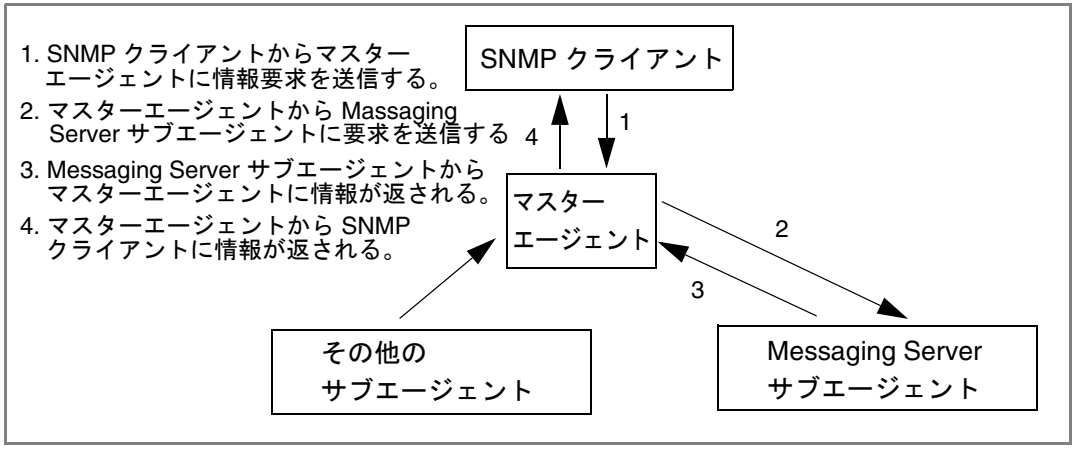
Messaging Server SNMP サポートには、次のような制限があります。

- SNMP を通じて監視できる 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、監視用のみである。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る)。

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ネイティブ 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 Messaging Server 用の SNMP サポートを設定する

SNMP 監視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 のネイティブ 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 のネイティブ 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
```

SNMP クライアントから監視する

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、*msg_svr_base*/lib/config-templates 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ほかの Sun Java System 製品との共存

SNMP サポートが提供されているほかの Netscape 製品または Sun Java System 製品では、プラットフォームのネイティブ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 Sun Java System 製品を Messaging Server と同じホストで実行し、両者を SNMP で監視する場合は、『Managing Servers with iPlanet Console』(http://docs.sun.com/source/816-5572-10/11_snmp.htm) の第 11 章の説明に従って Sun Java System 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 (ネイティブ SNMP エージェント) がほかの Sun Java System 製品のネイティブではない Sun Java System 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監視できるプラットフォームでは、applTable に複数の MTA とサーバーのセット、またほか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、再起動後にゼロにリセットされます。

各サイトには、監視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、サーバー情報があります。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 送信サーバーが有効の場合は、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 = Sun Java System Messaging Server 6.1
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 などのサフィックスは行番号 (applIndex) です。applIndex の値は、MTA に対しては値 1、HTTP サーバーに対しては値 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 番目のサフィックスがある行は HTTP サーバーのデータを提供しています。
2. 監視している 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 送信サーバーの動作ステータスは、それぞれのサーバーに設定された TCP ポートを通じて実際にこれらのサーバーに接続し、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、各サーバーのステータス (up (1)、down (2)、または congested (4)) が決定されます。

   これらの試みは、サーバーに対する通常の受信接続として認識され、各サーバーの 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 サーバーの場合、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

各サーバーを監視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ーステータス (applOperStatus) を監視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、監視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。ほかのサーバー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 という形式のサフィックスでは、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 ポートを表すサフィックスで、.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ーエージェント (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 というサフィックスは、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) を監視することは、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] = tcp_intranet[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 という形式のサフィックスでは、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 tcp_intranet 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 という形式のサフィックスでは、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開かれたか、チャネルから開かれた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` という形式のサフィックスでは、`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 tcp_intranet 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。

Messaging Server の Event Notification Service (ENS) を管理する

この付録では、Event Notification Service Publisher (ENS Publisher) を有効にし、Messaging Server の 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。

- Messaging Server に ENS Publisher をロードする
- Event Notification Service (ENS) のサンプルプログラムを実行する
- Event Notification Service (ENS) を管理する

ENS および ENS API の詳細は、Sun Java System Calendar Server のマニュアルの Web ページ (http://docs.sun.com/app/docs/coll/CalendarServer_05q1?l=ja) および Messaging Server のマニュアルの Web ページ (http://docs.sun.com/app/docs/coll/MessagingServer_05q1?l=ja) にある『Sun Java System Communications Services Event Notification Service Guide』を参照してください。

Messaging Server に ENS Publisher をロードする

Event Notification Service (ENS) (ENS) は、基礎となる発行および購読サービスです。ENS は、Sun Java System 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 (Messaging Server の ENS Publisher) は、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 (「Messaging Server に ENS Publisher をロードするには」を参照)。

Messaging Server で通知を購読するには、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。

Messaging Server に ENS Publisher をロードするには

コマンド行から以下の手順を実行します。以下の手順では、Messaging Server のインストールディレクトリの位置は *msg_svr_base* で、Messaging Server ユーザーは *inetuser* です。これらの変数の一般的な値は、前者は `/opt/SUNWmsgsr`、後者は `inetuser` です。

1. `inetuser` 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` *msg_svr_base*

   `./configutil -o "local.store.notifyplugin" -v "`*msg_svr_base*`/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` *msg_svr_base*`/sbin`

   `./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。詳細は、「Event Notification Service (ENS) 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 (ENS) のサンプルプログラムを実行する

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、*msg_svr_base*/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. *msg_svr_base*/examples ディレクトリに変更します。
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。*msg_svr_base*/examples 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 Event Notification Service for Messaging and Collaboration Manual』を参照してください。

注 *msg_svr_base*/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ー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 (ENS) 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 サーバー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。

  msg_svr_base`/sbin/start-msg ens`

- ENS を停止するには、次のコマンドを実行します。

  msg_svr_base`/sbin/stop-msg ens`

iPlanet Event Notification Service 設定パラメータ

iBiff の動作は、いくつかの設定パラメータによって制御されます。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 B-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します。デフォルトは 0 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します。デフォルトは 0 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します。`ReadMsg`、`NewMsg` などのさまざまな *eventTypes* については、『Messaging Server for Messaging and Collaboration Manual』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` 通知が無効になります。

表 B-1　iBiff 設定パラメータ (続き)

パラメータ	説明
local.store.notifyplugin.ensHost	ENS サーバーのホスト名を指定します。デフォルトは 127.0.0.1。
local.store.notifyplugin.ensPort	ENS サーバーの TCP ポートを指定します。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します。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ません。これは単に、ENS が使用する一意の識別子です。 このキーは、このキーに一致するイベントの通知を受けるために、加入者が登録するものです。

コンソールインタフェースを使用してメールユーザーとメーリングリストを管理する (推奨しない)

この付録は参考用としてのみ使用してください。この付録では説明されていますが、ユーザーとメーリングリストの作成および管理にはコンソールインタフェースを使用しないでください。ユーザー管理ユーティリティなど、ほかの推奨されたプロビジョニングツールを使用してください。

警告 コンソールインタフェースを使用してユーザーやグループを作成すると、さまざまな問題が発生します。Delegated Administrator など、ほかの推奨されたプロビジョニングツールを使用してください。『Sun Java System Messaging Server 管理ガイド』(http://docs.sun.com/doc/819-1054?l=ja) を参照してください。

この付録は参考用としてのみ使用してください。コンソールインタフェースを使ってユーザー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ことは**お勧めしません**。

メールユーザーを管理する

メールユーザーにアクセスするには

この項では、ユーザー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、ユーザーエントリの属性として企業の中央 LDAP ユーザーディレクトリに保存されています。そのため、メールアカウントを管理するには、そのディレクトリ内のユーザーエントリを変更する必要があります。

新規ユーザーを作成するには

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ーを作成します。新規ユーザー用のメールアカウントをインストールする必要もあります。メールアカウントをインストールしないと、ユーザーはコンソールのメール管理部分が使用できません。ユーザーを作成したり、その他のユーザー情報を指定したりする全プロセスについては、『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章に詳しく説明されています。

新規メールユーザーを作成するには、次の手順に従います。

1. コンソールのメインウィンドウで「ユーザーおよびグループ」タブをクリックします。
2. ドロップダウンリストから「新規ユーザー」を選択し、「作成」をクリックします。
3. ユーザーが属する組織単位を選択し、「了解」をクリックします。「ユーザーの作成」ウィンドウが開きます。
4. 『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章の説明に従って、ユーザーについての情報を入力します。
5. 「ユーザーの作成」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。このユーザー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。「ユーザーの作成」ウィンドウに「メール」タブが表示されます。
7. 「ユーザーの作成」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、「ユーザーの作成」ウィンドの下部にある「了解」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「了解」をクリックしてください。

既存のユーザー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、既存のユーザーにメール機能を与える場合は、ユーザーディレクトリ内でそのユーザーにアクセスし、メールアカウントの属性を追加または変更します。

既存のユーザー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. コンソールのメインウィンドウで「ユーザーおよびグループ」タブをクリックします。
2. 「ユーザーおよびグループ」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 (ユーザーの姓など) を入力し、ユーザーディレクトリを検索します。
4. 「ユーザーおよびグループ」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ーを選択して「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「了解」をクリックします。

ユーザーの電子メールアドレスを指定するには

メールがユーザーに正しく配信されるようにするには、まずユーザー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、ユーザー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。

ユーザー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. コンソールから「ユーザーの作成」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、884 ページの「メールユーザー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、ユーザー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) ユーザー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、ユーザーのアドレスとして公開される電子メールアドレスです。ユーザー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ー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、本質的にはユーザー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、ユーザー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る。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、ユーザー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、各ユーザー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. 「了解」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 (別のアドレスを入力する場合は、もう一度「追加」をクリックして「代替アドレス」ウィンドウを開く)。

7. ユーザー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「了解」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、各ユーザー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。

iPlanet Delegated Administrator for Messaging を使用している場合も、エンドユーザー向けの HTML インタフェースが提供されているので、エンドユーザー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。コンソールインタフェースと iPlant Delegated Administrator 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ーであるかにかかわらず、最新の設定が表示されます。

注 Delegated Administrator for Messaging では、Sun Java System LDAP スキーマ v. 1 のみがサポートされ、v.2 はサポートされません。

ユーザー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. コンソールから「ユーザーの作成」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、884 ページの「メールユーザー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. このユーザー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、888 ページの「POP/IMAP 配信を指定するには」を参照。
   - プログラム配信を指定する場合は、889 ページの「プログラム配信を指定するには」を参照。
   - UNIX 配信を指定する場合は、889 ページの「UNIX 配信を指定するには」を参照。
5. ユーザー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「了解」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 配信を指定するには

このオプションを選択すると、ユーザーの標準 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、573 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) ユーザーに割り当てる保存領域の上限 (ディスク制限容量) を入力します。制限は、デフォルト設定 (598 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 (K バイト /M バイト) にすることができます。
5. (省略可) ユーザーの保存可能なメッセージ数の上限を入力します。制限はデフォルト設定 (598 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には

このオプションを指定すると、メールがユーザー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、ユーザー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、ユーザー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。

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. ユーザー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. 「了解」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、ユーザーのメール配信方法が UNIX 配信に設定されます。つまり、UNIX 配信機能により、メッセージがユーザー指定の 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、ユーザーの 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 ユーザーが 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、ユーザー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、Delegated Administrator for Messaging にはエンドユーザー向けの HTML インタフェースがあり、ユーザー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。コンソールインタフェースと 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ーであるかにかかわらず、最新の設定が表示されます。

注 Delegated Administrator for Messaging では、Sun Java System LDAP スキーマ v. 1 のみがサポートされ、v.2 はサポートされません。

ユーザー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. コンソールから「ユーザーの作成」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、884 ページの「メールユーザー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。

   ユーザー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. 「了解」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. ユーザー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「了解」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ーアカウント間では、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

Messaging Server の自動返信機能を使用すると、着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、Vacation モード、自動返信モードの 2 種類を指定できます。

また、Delegated Administrator for Messaging にもエンドユーザー向けの HTML インタフェースがあり、エンドユーザー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。コンソールインタフェースと 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ーであるかにかかわらず、最新の設定が表示されます。

注 Delegated Administrator for Messaging では、Sun Java System LDAP スキーマ v. 1 のみがサポートされ、v.2 はサポートされません。

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. コンソールから「ユーザーの作成」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、884 ページの「メールユーザー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」: このユーザーの自動返信機能を無効にします。

   「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が生成されません。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. 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. ユーザー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「了解」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

ユーザー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. コンソールから「ユーザーの作成」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、884 ページの「メールユーザー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. 「了解」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 ユーザー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ます。新規グループ用のメールアカウントをインストールする必要もあります。メールアカウントをインストールしないと、グループはコンソールのメール管理部分が使用できません。ディレクトリグループを作成したり、その他のグループ情報を指定したりする全プロセスについては、『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章に詳しく説明されています。

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. コンソールのメインウィンドウで「ユーザーおよびグループ」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、「了解」をクリックします。
4. 『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章を参照して、「グループの作成」ウィンドウで、グループエントリの作成に必要な情報を入力します。

   メーリングリストの作成だけを目的とする場合は、「ユーザー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」タブを使用して追加できます。

   - 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 (スタティックまたはダイナミック) 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、「メール」タブを使用して追加します。

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「了解」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「了解」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「了解」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、ユーザー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. コンソールのメインウィンドウで「ユーザーおよびグループ」タブをクリックします。
2. 「ユーザーおよびグループ」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、ユーザーディレクトリを検索します。
4. 「ユーザーおよびグループ」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「了解」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. コンソールから「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、893 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る。

   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. 「了解」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 (別のアドレスを入力する場合は、もう一度「追加」をクリックして「代替電子メールアドレス」ウィンドウを開く)。

6. (省略可)「Errors-to」フィールドに、メーリングリスト宛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。

   ユーザー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る)。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ーで処理するように設定すれば、ほかのサーバーの負荷を軽減できます。

   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` コマンド行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、ユーザーの追加や削除、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`)、「了解」をクリックします。
   - 「検索」をクリックして、「ユーザー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 : 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 の構文が表示されます。「ユーザーおよびグループを検索」ウィンドウの詳細については、『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章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「了解」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、ユーザーディレクトリに適用するダイナミック検索条件を定義します。

ここでは、コンソールの「ユーザーおよびグループ」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。正規メンバーの追加は、インタフェースの別の場所で実行します。その手順については、『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章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、『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章を参照してください。

メンバーのダイナミック検索条件を定義するには

ダイナミック検索条件は、ユーザー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。グループ宛にメッセージが届くと、このメカニズムによって、名前のスタティックなリストではなく、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ーが決まります。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 の構文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、『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章を参照してください。Sun Java System Directory Server マニュアルおよび RFC 1959 も参照してください。

LDAP URL の構文は、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 C-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 サーバーのポート番号。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 ポートが使用されます。
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ります。
attributes	検索結果として返される属性。これらの属性は、Messaging Server によって返されます。
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ります。 「`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ります (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ります。
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=*`) が使用されます。

以下に、「Sunnyvale」をメールホストとするユーザー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) のユーザー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。

コンソール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ー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ー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。コマンド行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、コマンド行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、コンソール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. コンソールから「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、893 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、「電子メール専用メンバーのダイナミック検索条件」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。
   - 「了解」をクリックして「電子メール専用メンバーのダイナミック検索条件」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、897 ページの「メンバーのダイナミック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。
- 「了解」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「了解」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、特定のユーザー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. コンソールから「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、893 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ー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、「ユーザー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。「了解」をクリックして「許可された差出人」フィールドにユーザーを追加し、「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。

「ユーザーおよびグループを検索」ウィンドウの詳細については、『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章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、「了解」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「了解」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。

モデレータは、転送メッセージを受信すると、その処理方法を決定します (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)。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. コンソールから「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、893 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「ユーザー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。

   「ユーザーおよびグループを検索」ウィンドウの詳細については、『Sun ONE Server Console 5.2 Server Management Guide』の「User and Group Administration」の章を参照してください。
6. 「了解」をクリックしてモデレータを「モデレータのリスト」リストに追加し、「モデレータの追加」ウィンドウを閉じます (別のアドレスを入力する場合は、もう一度「追加」をクリックして「モデレータの追加」ウィンドウを開く)。
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「了解」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。

ショートメッセージサービス (SMS)

この付録では、Sun™ ONE Messaging Server 上にショートメッセージサービス (SMS) を実装する方法について説明します。この付録には、以下の項目があります。

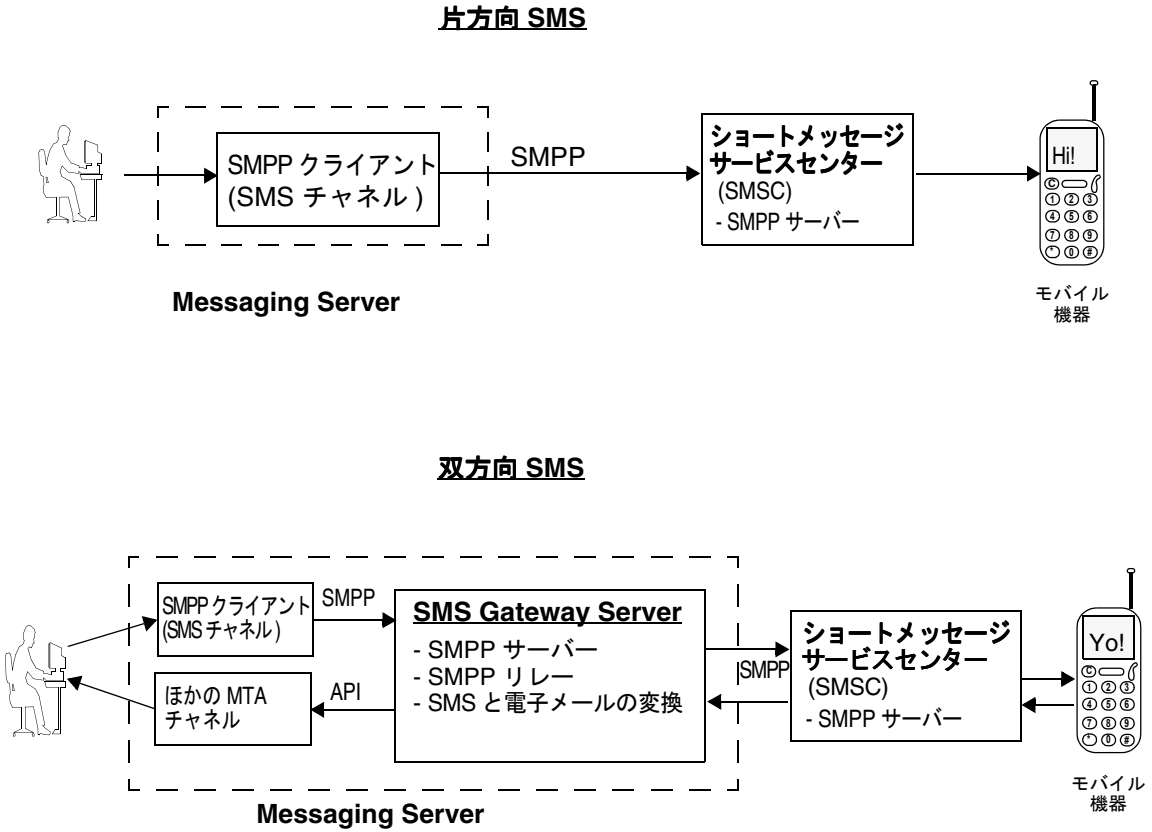
はじめに

Sun Java System Messaging Server では、ショートメッセージサービス (SMS) によって電子メールからモバイル、モバイルから電子メールへのメッセージングが実装されます。SMS は、片方向 (電子メールからモバイルのみ)、または双方向 (電子メールからモバイル、モバイルから電子メールの両方) のどちらかに設定できます。片方向のみのサービスを有効にするには、SMS チャネルを追加および設定する必要があります。双方向のサービスを有効にするには、SMS チャネルを追加および設定し、さらに SMS Gateway Server を設定する必要があります。

片方向と双方向のどちらの場合でも、生成された SMS メッセージは、Short Message Peer to Peer (SMPP) プロトコルを介してショートメッセージサービスセンター (SMSC) に送信されます。具体的には、SMSC では TCP/IP をサポートする V3.4 以上の SMPP サーバーが提供されている必要があります。

図 D-1 に、片方向 SMS の場合と双方向 SMS の場合のメッセージの論理フローを示します。

図 D-1　　片方向 SMS と双方向 SMS の論理フロー



片方向 SMS

片方向サービスを有効にするために、Messaging Server はリモート SMSC と通信する SMPP クライアント (MTA SMS チャネル) を使用します。SMS チャネルは、908 ページの「電子メールから SMS への変換プロセス」で説明されているように、キューに入れられた電子メールメッセージを SMS メッセージに変換します。この変換には、マルチパート MIME メッセージや、文字セットの変換問題の処理が含まれます。

このような処理を実行する SMS チャネルは、SMPP の外部ショートメッセージエンティティ (ESME) として機能します。

双方向 SMS

双方向 SMS では、メールサーバーは電子メールをリモート機器に送信するだけでなく、リモート機器から返信を受信したり、リモート機器の電子メール作成に対応したりできます。

双方向の SMS を有効にするには、前項目で説明されている MTA SMS チャネル (SMPP クライアント) に加えて、SMS Gateway Server が必要です。SMS Gateway Server は、Sun Java System Messaging Server の一般的なインストールプロセスの一環でインストールされますが、インストール後に設定する必要があります。SMS Gateway Server では、以下の 2 つの機能を実行します。

- SMPP リレー

  SMS Gateway Server は、MTA SMS チャネルと SMSC 間の透過的な SMPP クライアントとして機能します。リレーとして機能することに加え、SMS Gateway Server はリレーするメッセージ用に一意の SMS ソースアドレスを生成します。また、リモート SMSC から返されたメッセージ ID をあとで SMS 通知メッセージとの関連で使用するために保存します。

- SMPP サーバー

  SMS Gateway Server は SMPP サーバーとして機能し、モバイルを起点とする SMS メッセージ、電子メールに対する返信、および SMS 通知を受信します。SMS Gateway Server は、SMS メッセージから宛先電子メールアドレスを抽出します。抽出には変換プロセスが定義されているプロファイルを使用します。プロファイルには、電子メールからモバイルに送信されたメッセージに応答してリモート SMSC が返した通知メッセージの処理方法も定義されています。

注　Sun Java System Messaging Server は、Microsoft Windows プラットフォーム上での双方向 SMS をサポートしていません。

条件

このマニュアルでは、LogicaCMG の SMPP 仕様および使用している SMSC の SMPP マニュアルを読み終えていることを前提にしています。

SMS を実装するには、次の要件を満たす必要があります。

- Sun Java System Messaging Server 6 以上 (片方向 SMS は、iPlanet Messaging Server 5.2 でも実装される)
- SMSC は、TCP/IP 対応の SMPP V3.4 以上をサポートしている必要があり、Messaging Server を実行するホストと SMSC の間で TCP/IP 接続が可能である必要があります。

SMS Gateway Server のストレージ計画については、983 ページの「SMS Gateway Server のストレージ要件」を参照してください。

SMS チャネルの動作方式

SMS チャネルは、キューに入れられた電子メールメッセージを SMS メッセージに変換して、配信を担当する SMSC に渡すマルチスレッドチャネルです。

この節には、チャネル動作についての以下の項目があります。

電子メールをチャネルに送信する

923 ページの「SMS チャネルの設定」に従って SMS チャネルを設定すると、チャネルに 1 つまたは複数のホスト名が関連付けられます。説明のため、ここでは `sms.siroe.com` というホスト名がチャネルに関連付けられたホスト名であると仮定します。この場合、電子メールは次の形式のアドレスでチャネルに送信されます。

```
local-part@sms.siroe.com
```

`local-part` は、SMS 宛先アドレス (携帯電話番号、ポケットベル ID など) または次の形式の属性と値のペアのリストのどちらかです。

/attribute1=value1/attribute2=value2/.../@sms.siroe.com

表 D-1 に、認識される属性名とその使用法を示します。これらの属性を使用して、一部のチャネルオプションで受取人単位の制御が行えます。

表 D-1　SMS 属性

属性名	属性値と使用法
ID	SMS メッセージの送信先である SMS 宛先アドレス (携帯電話番号、ポケットベル ID など)。この属性とその値は必須です。
FROM	SMS ソースアドレス。オプションが USE_HEADER_FROM=0 である場合は無視されます。
FROM_NPI	NPI。指定した NPI 値を使用します。オプションが USE_HEADER_FROM=0 である場合は無視されます。
FROM_TON	TON。指定した TON 値を使用します。オプションが USE_HEADER_FROM=0 である場合は無視されます。
MAXLEN	生成された SMS メッセージまたはこの受取人宛のメッセージに含める最大合計バイト数 (8 ビットバイト)。MAXLEN の値と MAX_MESSAGE_SIZE チャネルオプションで指定されている値のうち、低いほうの値が使用されます。
MAXPAGES	この受取人用に電子メールを分割して生成される SMS メッセージの最大数。MAXPAGES の値と MAX_PAGES_PER_MESSAGE チャネルオプションで指定されている値のうち、低いほうの値が使用されます。
NPI	ID 属性で指定されている宛先 SMS アドレスの番号計画識別子 (NPI) の値を指定します。この属性で受け入れられる値については、DEFAULT_DESTINATION_NPI チャネルオプションを参照してください。この属性が使用されると、その値は DEFAULT_DESTINATION_NPI チャネルオプションで指定されている値より優先されます。
PAGELEN	この受取人宛の単一の SMS メッセージに含める最大バイト数。この値と MAX_PAGE_SIZE チャネルオプションで指定されている値のうち、小さいほうの値が使用されます。
TO	ID と同義。
TO_NPI	NPI と同義。
TO_TON	TON と同義。
TON	ID 属性で指定されている宛先 SMS アドレスの番号種別 (TON) の値を指定します。この属性で受け入れられる値については、DEFAULT_DESTINATION_TON チャネルオプションを参照してください。この属性が使用されると、その値は DEFAULT_DESTINATION_TON チャネルオプションで指定されている値より優先されます。

次にアドレスの例を示します。

```
123456@sms.siroe.com
/id=123456/@sms.siroe.com
/id=123456/maxlen=100/@sms.siroe.com
/id=123456/maxpages=1/@sms.siroe.com
```

電子メールアドレスの SMS 宛先アドレス部分に対する変換、妥当性チェック、およびその他の処理については、917 ページの「サイト定義のアドレス妥当性チェックと変換」を参照してください。

電子メールから SMS への変換プロセス

電子メールをリモートサイトに送信するには、電子メールをリモート SMSC によって認識される SMS メッセージに変換する必要があります。この節では、SMS チャネルのキューに入れられた電子メールメッセージを 1 つまたは複数の SMS メッセージに変換するプロセスについて説明します。以下で説明されているように、生成される SMS メッセージの最大数、SMS メッセージの合計の長さの最大値、および 1 つの SMS メッセージの最大サイズはオプションで制御します。電子メールメッセージのテキスト部分 (MIME のテキストコンテントタイプ) のみが使用され、変換される部分の最大数も制御できます。

電子メールメッセージのヘッダー行とテキスト部分で使用される文字セットは、すべて Unicode に変換されてから、適切な SMS 文字セットに変換されます。

`SMS_TEXT` マッピングテーブル (918 ページの「サイト定義のテキスト変換」を参照) がない場合は、SMS チャネルのキューに入れられた電子メールメッセージに対して図 D-2 で示す処理が実行されます。

図 D-2　　SMS チャネルの電子メール処理

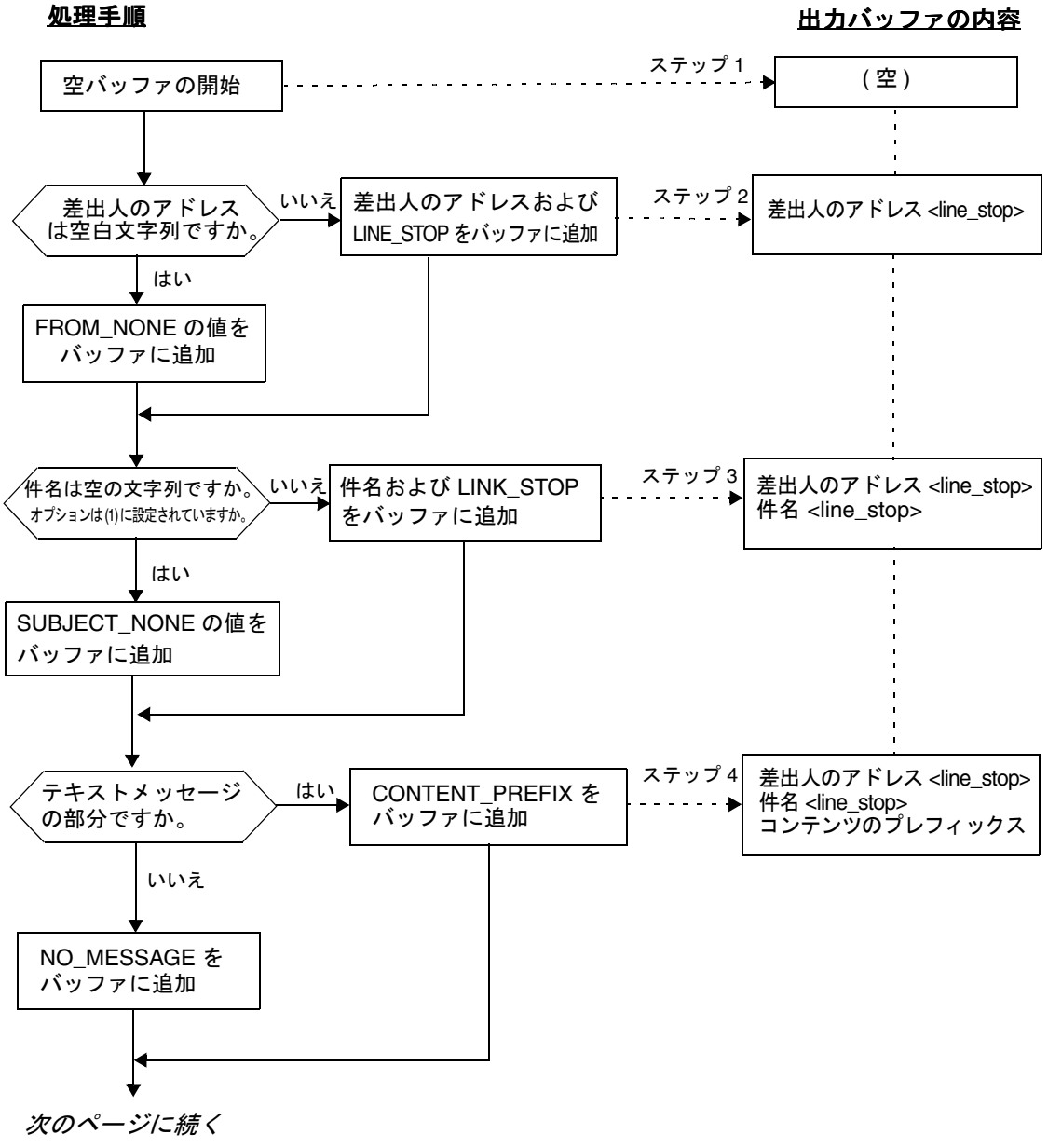
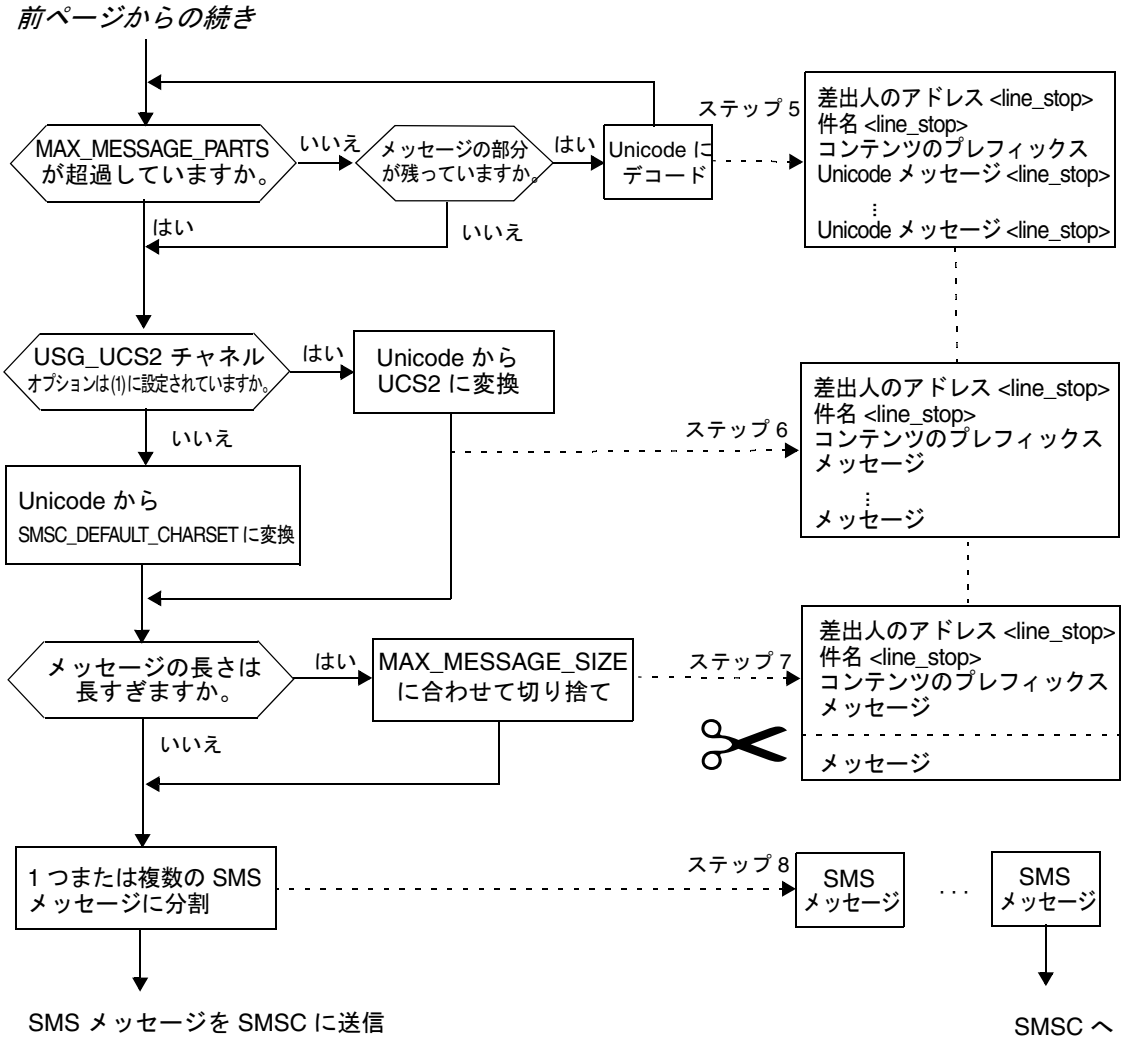


図 D-3　　SMS チャネルの電子メール処理 (続き)



以下の手順は、図 D-2 で示されている番号と対応します。

1. 空の出力バッファが開始されます。バッファに使用される文字セットは Unicode です。

2. 電子メールメッセージの差出人のアドレスは、以下の 5 つのソースから 1 つ取り出されます。ソースは優先度の高いものから低いものの順で表示されています。

   1. Resent-from:
   2. From:
   3. Resent-sender:
   4. Sender:
   5. Envelope From:

   差出人のアドレスが空の文字列である場合は、差出人のアドレスの代わりに FROM_NONE チャネルオプションの値がバッファに追加されます。

   差出人のアドレスが空の文字列ではない場合は、FROM_FORMAT チャネルオプションを処理した結果および LINE_STOP オプションの値が出力バッファに追加されます。

   Resent-from: および Resent-sender: ヘッダー行は、USE_HEADER_RESENT オプションの値が 1 である場合にのみ考慮されることに注意してください。それ以外の場合は、Resent- ヘッダー行は無視されます。

3. Subject: ヘッダー行が存在しない場合または空の場合は、SUBJECT_NONE オプションの値が出力バッファに追加されます。

   それ以外の場合は、SUBJECT_FORMAT オプションを処理した結果および LINE_STOP チャネルオプションの値が出力バッファに追加されます。

4. テキストメッセージ部分がない場合は、NO_MESSAGE チャネルオプションの値が出力バッファに追加されます。

   テキストメッセージ部分がある場合は、CONTENT_PREFIX チャネルオプションの値が出力バッファに追加されます。

   テキスト以外のメッセージ部分は破棄されます。

5. 各テキスト部分に関しては、MAX_MESSAGE_PARTS の制限に達していない場合、テキスト部分は Unicode にデコードされ、LINE_STOP チャネルオプションの値とともにバッファに追加されます。

6. 結果の出力バッファは、Unicode から SMSC のデフォルトの文字セットまたは UCS2 (UTF-16) のどちらかに変換されます。SMSC のデフォルトの文字セットは、SMSC_DEFAULT_CHARSET オプションを使用して指定します。

7. 変換後は、MAX_MESSAGE_SIZE のバイト数を超えないように切り捨てられます。

8. 手順 6 で変換された文字列は、1 つまたは複数の SMS メッセージに分割されます。各 SMS メッセージは、MAX_PAGE_SIZE のバイト数以内の長さになります。最大で MAX_PAGES_PER_MESSAGE の SMS メッセージが生成されます。

注	電子メールメッセージは複数の受取人を持つ場合もあるので、手順 6 ～ 手順 8 は受取人のアドレスごとに実行される必要があります。このとき、906 ページの「電子メールをチャネルに送信する」で説明されている MAXLEN、MAXPAGES または PAGELEN 属性が使用されます。

電子メールメッセージ処理の例

たとえば、チャネルのデフォルトの設定で次のような電子メールメッセージを処理するとします。

```
From: John Doe
To:1234567@sms.siroe.com
Subject: Today's meeting
Date: Fri, 26 March 2001 08:17

The staff meeting is at 14:30 today in the big conference room.
```

この電子メールメッセージは次のように SMS メッセージに変換されます。

```
jdoe@siroe.com (Today's meeting) The staff meeting is at 14:30 today
in the big conference room.
```

別の一連のオプション設定での処理を次に示します。

```
CONTENT_PREFIX=Msg:
FROM_FORMAT=From:${pa}
SUBJECT_FORMAT=Subj:$s
```

この設定では以下の結果になります。

```
From:John Doe Subj:Today's meeting Msg:The staff meeting is at 14:30
today in the big conference room.
```

SMS メッセージの送信プロセス

電子メールメッセージが 1 つまたは複数の SMS メッセージ (通常は各受取人用に異なるセットがある) に変換されると、SMS メッセージは宛先 SMSC に送信されます。送信処理は、TCP/IP 対応の SMPP V3.4 を使用して有効にします。SMPP サーバーのホスト名 (SMPP_SERVER) は、SMS チャネルに関連付けられた正式なホスト名として採用されます。使用する TCP ポート (SMPP_PORT) は、port チャネルキーワードで指定します。

処理するメッセージがある場合、チャネルが起動します。チャネルは 942 ページの「SMPP オプション」で説明されているように、ESME_ チャネルオプションで指定されている証明書を提示して SMPP サーバーにトランスミッタとしてバインドします。表 D-2 に、BIND_TRANSMITTER PDU (プロトコルデータユニット) で設定するフィールドの一覧と各フィールドの値を示します。

表 D-2 生成された BIND_TRANSMITTER PDU のフィールド

フィールド	値
system_id	ESME_SYSTEM_ID チャネルオプション。デフォルト値は空の文字列
password	ESME_PASSWORD チャネルオプション。デフォルト値は空の文字列
system_type	ESME_SYSTEM_TYPE チャネルオプション。デフォルト値は空の文字列
interface_version	0x34 は SMPP V3.4 を示します
addr_ton	ESME_ADDRESS_TON。デフォルト値は 0x00 で、これは不明な TON を示します
addr_npi	ESME_ADDRESS_NPI。デフォルト値は 0x00 で、これは不明な NPI を示します
addr_range	ESME_IP_ADDRESS チャネルオプション。デフォルト値は空の文字列

チャネルはマルチスレッドです。送信するメールの数に応じて、チャネルは複数のデキュースレッドを実行します。複数のチャネルプロセスが実行されていることさえもあります。各スレッドは BIND_TRANSMITTER を実行して TCP/IP 接続上で送信する必要のあるすべての SMS メッセージを送信し、その後 UNBIND を送信して接続を終了します。再び使用する可能性をふまえてアイドル時間に接続を開いたままにしておく試

行は行われません。リモート SMPP サーバーがスロットルエラーを返してきた場合は、UNBIND が発行されて TCP/IP 接続は終了し、新しい接続と BIND が確立されます。SMS メッセージの送信が終了する前に SMPP サーバーが UNBIND を返してきた場合も同様に動作します。

その後、SMS メッセージは SMPP SUBMIT_SM PDU を使用して送信されます。永久的なエラーが返された場合 (たとえば、ESME_RINVDSTADR)、電子メールメッセージは配信されずに戻ってきます。一時的なエラーが返された場合は、電子メールメッセージはあとで配信が試行されるように再びキューに入れられます。正確には、永久的なエラーとは、エラーが原因で発生した状態がいつまでも続く可能性があるもので、配信試行の繰り返しに前向きな効果がないものです。たとえば、無効な SMS 宛先アドレスなどです。これとは異なり、一時的なエラーとは、エラーが原因で発生した状態が近い将来に存在しなくなる可能性のあるものです。たとえば、サーバーダウンやサーバーが混み合っている状態です。

USE_HEADER_FROM オプションの値が 1 である場合、送信される SMS メッセージのソースアドレスが設定されます。使用される値は、元の電子メールメッセージから生成され、返信の送信先 (電子メール) アドレスとしてもっとも可能性の高いものに選ばれます。したがって、ソースアドレスは以下の 7 つのソースから作成されます。ソースは優先度の高いものから低いものの順で表示されています。

1. Resent-reply-to:
2. Resent-from:
3. Reply-to:
4. From:
5. Resent-sender:
6. Sender:
7. Envelope From:

Resent-reply-to: および Reply-to: ヘッダー行は、USE_HEADER_REPLY_TO オプションの値が 1 である場合にのみ考慮されることに注意してください。また、Resent-reply-to:、Resent-from:、および Resent-sender: ヘッダー行は、USE_HEADER_RESENT オプションの値が 1 である場合にのみ考慮されることにも注意してください。つまり、Resent-reply-to: ヘッダー行が考慮されるには、これらのオプションの両方の値が 1 である必要があります。これらのオプションは両方とも、デフォルト値は 0 です。したがって、デフォルトの設定では 項目 4、6、および 7 のみが考慮されます。さらに、SMS メッセージのソースアドレスは 20 バイトに制限されているので、選択されるソースアドレスは、その制限を超えている場合は切り捨てられることに注意してください。

表 D-3 に、SUBMIT_SM PDU に設定する必須フィールドを示します。

表 D-3 生成された SUBMIT_SM PDU の必須フィールド

フィールド	値
service_type	DEFAULT_SERVICE_TYPE チャネルオプション。デフォルト値は空の文字列。
source_addr_ton	DEFAULT_SOURCE_TON チャネルオプション。USE_HEADER_FROM=1 の場合、このフィールドの値は英数字の TON を示す 0x05 になります。これ以外の場合は、デフォルト値の国際 TON を示す 0x01 になります。
source_addr_npi	DEFAULT_SOURCE_NPI チャネルオプション。デフォルト値は 0x00。
source_addr	DEFAULT_SOURCE_ADDRESS チャネルオプション。USE_HEADER_FROM=0 以外の場合は、電子メールメッセージの差出人を示す英数字の文字列。
dest_addr_ton	TON アドレス指定属性または DEFAULT_DESTINATION_TON チャネルオプション。デフォルト値は国際 TON を示す 0x01 です。
dest_addr_npi	NPI アドレス指定属性または DEFAULT_SOURCE_NPI チャネルオプション。デフォルト値は不明の NPI を示す 0x00 です。
dest_addr	電子メールエンベロープ To: アドレスのローカル部分を基に生成された宛先 SMS アドレス。906 ページの「電子メールをチャネルに送信する」を参照。
esm_class	片方向 SMS の場合は 0x03 に設定し、ストアアンドフォワードモード、デフォルトの SMSC メッセージタイプ、および返信パスを設定しないことを示します。双方向 MSM メッセージの場合は 0x83 に設定します。
protocol_id	0x00 は CDMA および TDMA には使用されません。GSM の場合に 0x00 を指定すると、インターネットプロトコルを使用せず、SME 対 SME のプロトコルを使用することを示します。
priority_flag	GSM と CDMA の場合は 0x00、TDMA の場合は 0x01。どちらも標準レベルの優先度を示します。DEFAULT_PRIORITY チャネルオプションの説明を参照。
schedule_delivery_time	空の文字列は即時配信を示します。
validity_period	DEFAULT_VALIDITY_PERIOD チャネルオプション。デフォルト値は空の文字列で、これは SMSC のデフォルトを使用することを示します。
registered_delivery	0x00 は登録された配信がないことを示します。
replace_if_present_flag	0x00 は過去の SMS メッセージを置き換えないことを示します。
data_coding	0x00 は SMSC のデフォルトの文字セットを示します。0x08 は UCS2 文字セットを示します。
sm_default_msg_id	0x00 はあらかじめ定義されているメッセージを使用しないことを示します。

表 D-3　生成された SUBMIT_SM PDU の必須フィールド (続き)

フィールド	値
sm_length	SMS メッセージの長さと内容。詳細は 908 ページの「電子メールから SMS への変換プロセス」を参照。
short_message	SMS メッセージの長さと内容。詳細は 908 ページの「電子メールから SMS への変換プロセス」を参照。

表 D-4 に、SUBMIT_SM PDU に設定するオプションのフィールドを示します。

表 D-4　生成された SUBMIT_SM PDU のオプションのフィールド

フィールド	値
privacy	DEFAULT_PRIVACY チャネルキーワードの説明を参照。デフォルトでは、電子メールメッセージに Sensitivity: ヘッダー行がない場合、このフィールドは提供されません
sar_refnum	USE_SAR チャネルキーワードの説明を参照。デフォルトでは、このフィールドは提供されません
sar_total	前述の sar_refnum を参照。
sar_seqnum	前述の sar_refnum を参照。

チャネルは、送信する SMS メッセージがなくなるまで (メッセージキューが空になるまで)、または MAX_PAGES_PER_BIND を超過するまで、SMPP サーバーにバインドしたままです。後者の場合で送信する SMS メッセージがまだ残っている場合は、新しい接続が確立され、バインドが実行されます。

SMS チャネルはマルチスレッドです。チャネルの処理スレッドは、それぞれが SMPP サーバーとの専用の TCP 接続を保持します。たとえば、3 つの処理スレッドがあり、それぞれが送信対象の SMS メッセージを処理する場合、チャネルは SMPP サーバーとの 3 つの開いた TCP 接続を持ちます。各接続はトランスミッタとして SMPP サーバーにバインドします。また、どの処理スレッドにも、処理中の SMS 送信は 1 度に 1 つしかありません。つまり、スレッドは SMS メッセージを送信すると、送信応答 (SUBMIT_SM_RESP PDU) があるまで待機し、それまで別の SMS メッセージを送信しません。

サイト定義のアドレス妥当性チェックと変換

サイトで妥当性チェックを実行したり、受取人の電子メールアドレスでエンコードされた SMS 宛先アドレスを変換する必要がある場合もあります (906 ページの「電子メールをチャネルに送信する」を参照)。たとえば、サイトで実行する処理には以下のようなものがあります。

- 非数値文字を取り除く (例 : 800.555.1212 を 8005551212 に変換する)
- プレフィックスを前に付ける (例 : 8005551212 を +18005551212 に変換する)
- 妥当性を検証する (例 : 123 は短すぎる)

上記の最初の 2 つのタスクは、DESTINATION_ADDRESS_NUMERIC および DESTINATION_ADDRESS_PREFIX チャネルオプションを使用して実行できます。一般的に、上記の 3 つのタスクおよびその他のタスクは、マッピングテーブルを使用して実装できます。書き換えルールからマッピングテーブルを呼び出す方法、または FORWARD マッピングテーブルによる方法のどちらかを使用します。書き換えルールからマッピングテーブル呼び出す方法を使用した場合は、柔軟性がもっとも高く、サイト定義のエラー応答が付いたアドレスを拒否することもできます。この節の以下の部分では、このようなアプローチ、つまり書き換えルールからマッピングテーブルを呼び出す方法を使用する場合について説明します。

宛先アドレスは数字のみで、10 または 11 桁の長さを持ち、文字列「+1」が先頭に付いている必要があると仮定します。これは、以下の書き換えルールを使用して実現できます。

```
sms.siroe.com      ${X-REWRITE-SMS-ADDRESS,$U}@sms.siroe.com
sms.siroe.com      $?Invalid SMS address
```

上記の最初の書き換えルールは、X-REWRITE-SMS-ADDRESS という名前のサイト定義のマッピングテーブルを呼び出しています。このマッピングテーブルは、検査のために電子メールアドレスのローカル部分に渡されます。マッピングプロセスで、そのローカル部分が受け入れ可能と判断された場合は、アドレスは受け入れられ、SMS チャネルに書き換えられます。マッピングプロセスで、そのローカル部分が受け入れ不可と判断された場合は、次の書き換えルールが適用されます。次の書き換えルールは「$?」書き換えルールであるので、アドレスは拒否され、「Invalid SMS address」というエラーテキストが表示されます。

以下に X-REWRITE-SMS-ADDRESS マッピングテーブルを示します。このマッピングテーブルによって、属性と値のペアのリスト形式または SMS 宛先アドレスの行のどちらかであるローカル部分に対して必要な検証ステップが実行されます。

```
X-VALIDATE-SMS-ADDRESS

! 数値以外の文字を取り除く
    $_*$[$ -/:-~]%*  $0$2$R
! アドレス形式が 1nnnnnnnnnn または nnnnnnnnnn の場合は受け入れる
! 受け入れる場合、出力は +1nnnnnnnnnn であることを確認する
```

```
   1%%%%%%%%%%       +1$0$1$2$3$4$5$6$7$8$9$Y
   %%%%%%%%%%       +1$0$1$2$3$4$5$6$7$8$9$Y
! このアドレスは受け入れられなかったため、無効となる
   *                 $N

X-REWRITE-SMS-ADDRESS

    */id=$_*/*       $C$0/id=$|X-VALIDATE-SMS-ADDRESS;$1|/$2$Y$E
    */id=$_*/*       $N
    *                $C$|X-VALIDATE-SMS-ADDRESS;$0|$Y$E
    *                $N
```

上記の設定の場合、DESTINATION_ADDRESS_NUMERIC オプションの値は必ず 0 (デフォルト) にしてください。それ以外の値では、SMS 宛先アドレスから「+」が取り除かれます。

サイト定義のテキスト変換

変換ルールのテーブルを使用して、908 ページの「電子メールから SMS への変換プロセス」に示した手順 1 ～ 6 をサイトでカスタマイズできます。これらのルールは、MTA のマッピングファイル内のマッピングテーブルを使用して指定します。

マッピングテーブルの名前は、*SMS_Channel*_TEXT とし、*SMS_Channel* には SMS チャネルの名前を指定します。たとえば、チャネルの名前が sms である場合は SMS_TEXT、チャネルの名前が sms_mway である場合は SMS_MWAY_TEXT とします。

このマッピングテーブルには、2 種類のエントリが入ります。ただし、これらのエントリの形式についての説明を始める前に、エントリの作成および使用方法を理解するため、マッピングファイルの使用方法を理解しておくことが不可欠です。これら 2 種類のエントリの説明の後、マッピングテーブルの例を示します。

2 種類のエントリを以下に示します。

- メッセージヘッダーエントリ
- メッセージ本文エントリ

メッセージヘッダーエントリ

メッセージヘッダーエントリは、SMS メッセージに含めるヘッダー行を指定したり、ヘッダー行の略記方法または略記以外の場合に行われる変換方法を指定したりします。メッセージヘッダーエントリの 1 つによって、ヘッダー行がゼロでない長さの文字列に正常にマッピングされた場合にのみ、ヘッダー行は生成される SMS メッセージに含められます。各エントリには次のような形式があります。

H|*pattern*　*replacement-text*

メッセージのヘッダー行がこのパターンに一致すると、ヘッダー行は置換テキスト「replacement-text」で置き換えられます。このときマッピングファイルのパターン一致機能および文字列置換機能が使用されます。その後、メタキャラクタ「$Y」が置換テキストに指定されていれば、ヘッダー行のマッピングによる最終的な結果は SMS メッセージに含められます。ヘッダー行がどのパターン文字列とも一致しない場合、ヘッダー行が長さゼロの文字列にマッピングされた場合、またはメタキャラクタ「$Y」が置換テキストに指定されていない場合は、ヘッダー行は SMS メッセージに含められません。以下に 2 つのエントリを示します。

```
H|From:*  F:$0$Y
H|Subject:*   S:$0$Y
```

これらのエントリによって、`From:` および `Subject:` ヘッダー行は SMS メッセージに含められます。このとき `From:` および `Subject:` は `F:` と `S:` として略記されます。以下のエントリの場合、

```
H|Date:*  H|D:$0$R$Y
H|D:*,*%19%*:*:*  H|D:$0$ $5:$6$R$Y
```

`Date:` ヘッダー行が受け入れられ、次のヘッダー行のようにマップされます。

```
Date: Wed, 16 Dec 1992 16:13:27 -0700 (PDT)
```

これは次のように変換されます。

```
D: Wed 16:13
```

非常に複雑で反復的なマッピングが作成される場合もあります。サイトにカスタムフィルタを設定する場合は、最初にマッピングファイルの動作方法を理解しておく必要があります。エントリの右側の H| は、必要に応じて省略できます。H| は、一連の反復的なマッピングで必要とされるテーブルエントリの数を削減するために右側に置かれています。

メッセージ本文エントリ

メッセージ本文エントリは、マッピングを確立してメッセージ本文の各行に適用されます。メッセージ本文の各行は、確立されたマッピングが適用されてから、生成中の SMS メッセージに組み込まれます。メッセージ本文エントリは、次の形式をとります。

`B|`*pattern*　　`B|`*replacement-text*

メッセージ本文の行が *pattern* パターンと一致すると、置換テキスト *replacement-text* で置き換えられます。ここでも、この機能を使用して非常に複雑で反復的なマッピングが作成される場合があります。エントリの右側の B| は、必要に応じて省略できます。

SMS マッピングテーブルの例

コード例 D-1 に、SMS_TEXT マッピングテーブルの例を示します。各行の終わりにある括弧の中の数字は、このテーブルのあとに示す「説明テキスト」というタイトルのセクションでの項目番号と対応しています。

コード例 D-1　　SMS_TEXT マッピングテーブルの例

```
SMS_TEXT

   H|From:*          H|F:$0$R$Y  (1.)
   H|Subject:*       H|S:$0$R$Y  (1.)
   H|F:*<*>*         H|F:$1$R$Y  ()
   H|F:*(*)*         H|F:$0$2$R$Y  (2.)
   H|F:*"*"*         H|F:$0$2$R$Y  (3.)
   H|F:*@*           H|F:$0$R$Y  (4.)
   H|%:$ *           H|$0:$1$R$Y  (5.)
   H|%:*$            H|$0:$1$R$Y  (5.)
   H|%:*$ $ *        H|$0:$1$ $2$R$Y  (6.)
   B|*--*            B|$0-$1$R  (7.)
   B|*..*            B|$0.$1$R  (7.)
   B|*!!*            B|$0!$1$R  (7.)
   B|*??*            B|$0?$1$R  (7.)
   B|*$ $ *          B|$0$ $1$R  (6.)
   B|$ *             B|$0$R  (5.)
   B|*$              B|$0$R  (5.)
```

説明テキスト

上記の例の SMS_TEXT マッピングテーブルのエントリの説明を以下に示します。

上記の例では、マッピングの反復的適用の実装と制御にメタキャラクタ「$R」が使用されています。これらのマッピングを反復することによって、強力なフィルタリングが実行されます。たとえば、前後に付いている単一のスペースを削除する (6)、または 2 つのスペースを 1 つに削減する (7) という単純なマッピングは、全体として採用された場合に、前後に付いているすべてのスペースを削除し、連続する複数のスペースを単一のスペースに削減するフィルタとなります。このようなフィルタリングによって、各 SMS メッセージのサイズを小さくできます。

1. これらの 2 つのエントリによって、From: および Subject: ヘッダー行が SMS メッセージに含められます。From: および Subject: は、それぞれ F: と S: として略記されます。これら以外のエントリにも、From: および Subject: ヘッダー行にさらに影響を与えるものがあります。

   このエントリは、<...> パターンを含む From: ヘッダー行を角括弧内のテキストのみにします。次に例を示します。

   ```
   F: "John C. Doe" <jdoe@siroe.com> (Hello)
   ```

   これは、次のように置き換えられます。

   ```
   F: jdoe@siroe.com
   ```

2. このエントリは、From: ヘッダー行の (...) パターン内のすべてを包括的に削除します。次に例を示します。

   ```
   F: "John C. Doe" <jdoe@siroe.com> (Hello)
   ```

   これは、次のように置き換えられます。

   ```
   F: "John C. Doe" <jdoe@siroe.com>
   ```

3. このエントリは、From: ヘッダー行の "..." パターン内のすべてを包括的に削除します。次に例を示します。

   ```
   F: "John C. Doe" <jdoe@siroe.com> (Hello)
   ```

   これは、次のように置き換えられます。

   ```
   F: <jdoe@siroe.com> (Hello)
   ```

4. このエントリは、From: ヘッダー行のアットマーク (@) の左側にあるものをすべて包括的に削除します。次に例を示します。

   ```
   F: "John C. Doe" <jdoe@siroe.com> (Hello)
   ```

   これは、次のように置き換えられます。

   ```
   F: "John C. Doe" <jdoe@
   ```

5. これらの 4 つのエントリは、メッセージヘッダーと本文の行から前後に付いているスペースを削除します。

6. これら 2 つのエントリは、メッセージのヘッダーと本文の行の 2 つのスペースを 1 つのスペースに削減します。

7. これらの 4 つのエントリは、二重になっているダッシュ、ピリオド、感嘆符、および疑問符を一致する文字の単一の不定発生に削減します。これによっても、SMS メッセージのバイト数を節約できます。

エントリの順序は非常に重要です。たとえば、所定の順序で、次のようなメッセージの From: ヘッダー行から始めます。

```
From: "John C. Doe"  (Hello)
```

これは次のように短縮されます。

```
jdoe
```

この結果までの手順は次のとおりです。

1. 次の From: ヘッダー行から始めます。

   ```
   From: "John C. Doe"  (Hello)
   ```

   最初のマッピングエントリのパターンがこれと一致し、次の結果になります。

   ```
   F: "John C. Doe"  (Hello)
   ```

   結果文字列の「$R」メタキャラクタによって、結果文字列は再度マッピングされます。

2. 直前の手順の結果文字列にマッピングが適用されます。これによって、次の結果になります。

   ```
   F: jdoe@siroe.com
   ```

   マッピングの「$R」によって、一連のマッピング全体がこの手順の結果に再び適用されます。

3. 次に、マッピングが適用され、次の結果になります。

   ```
   F: jdoe
   ```

   マッピングの「$R」によって、一連のマッピング全体がこの手順の結果に再び適用されます。

4. 次に、マッピングが適用され、次の結果になります。

   ```
   F:jdoe
   ```

   マッピングの「$R」によって、一連のマッピング全体がこの手順の結果に再び適用されます。

5. ほかのエントリは一致しないため、最終的な結果文字列は次のとおりになります。

   ```
   F:jdoe
   ```

これが SMS メッセージに取り込まれます。

注

imsimta test-mapping ユーティリティを使用してマッピングテーブルをテストすることができます。次に例を示します。

```
# imsimta test -mapping -noimage_file
-mapping_file=test.txt
Enter table name:SMS_TEXT
Input string: H|From: "John C. Doe"  (Hello)
Output string: H|F:jdoe
Output flags: [0,1,2,89]
Input string: ^D
#
```

imsimta ユーティリティの詳細については、『Sun Java System Messaging Server 管理ガイド』(http://docs.sun.com/doc/819-1054?l=ja) を参照してください。

SMS チャネルの設定

この節では、片方向 (電子メールからモバイル) および双方向 (電子メールからモバイル、モバイルから電子メール) の両方の機能に必要な SMS チャネルの設定方法について説明します。片方向の場合も双方向の場合も SMS チャネルの設定は同じです。ただし、954 ページの「双方向 SMS 用に SMS チャネルを設定する」の項目で示されている例外を除きます。

この節には、以下の項目があります。

SMS チャネルを追加する

SMS チャネルを Messaging Server の設定に追加するには、次の 2 つの手順を実行する必要があります。

1. 924 ページの「チャネル定義と書き換えルールを追加する」
2. 927 ページの「SMS チャネルオプションファイルを作成する」

すべての状況で設定が必須とされるチャネルオプションはありませんが、次に示すオプションのうち、1 つまたは複数は設定する必要があります。ESME_PASSWORD、ESME_SYSTEM_ID、MAX_PAGE_SIZE、DEFAULT_SOURCE_TON、および DEFAULT_DESTINATION_TON。また、説明されているように、imta.cnf ファイルまたはチャネルオプションファイルのどちらかのチャネル定義を介して、SMPP サーバーのホスト名または IP アドレスと TCP ポートを設定する必要があります。

複数の SMS チャネルを設定し、それぞれに異なる特徴を持たせることもできます。複数の SMS チャネルの使用の詳細については、950 ページの「SMS チャネルをさらに追加する」を参照してください。

次のことに注意してください。imta.cnf ファイルを変更した場合はコンパイルしなおす必要があります。チャネルオプションファイルのみを変更した場合はコンパイルしなおす必要はありません。

また、チャネルの変更が反映されるまでの時間は、変更内容によって異なることがあります。チャネルオプションの変更の多くは、変更が行われてから起動したすべてのチャネルに反映されます。ジョブコントローラが頻繁に新しいチャネルを起動しているので、この場合はほとんど瞬時に反映されたように見えます。一部の変更は、コンパイルしなおし、SMTP サーバーを再起動するまで反映されません。これらのオプションは、チャネル自体が作動したときではなく、メッセージがチャネルのキューに入れられるときに処理されます。

チャネル定義と書き換えルールを追加する

チャネル定義と書き換えルールを追加するには、次の手順を実行します。

1. SMS チャネルを MTA の設定に追加する前に、そのチャネルの名前を決める必要があります。チャネルの名前は、sms または sms_x のどちらかにします。x は大文字と小文字が区別される文字列であり、長さは 1 ～ 36 バイトです (例 : sms_mway)。

2. チャネル定義を追加するには、*installation-directory*/config/ ディレクトリにある imta.cnf ファイルを編集します。ファイルの最後に空白行を追加し、次の 2 行を追加します。

   ```
   channel-name port p threaddepth t ¥
     backoff pt2m pt5m pt10m pt30m notices 1
   smpp-host-name
   ```

   channel-name はこのチャネル用に選んだ名前、*p* は SMPP サーバーが待機する TCP ポート、*t* は 1 つの配信プロセスで SMPP サーバーに同時に接続できる最大数、および *smpp-host-name* は SMPP サーバーを実行しているシステムのホスト名です。

   たとえば、チャネル定義は次のように指定します。

   ```
   sms_mway port 55555 threaddepth 20 ¥
     backoff pt2m pt5m pt10m pt30m notices 1
   smpp.siroe.com
   ```

   threaddepth の計算方法については、926 ページの「同時接続の数を制御する」を参照してください。

   backoff および notices チャネルキーワードについては、951 ページの「配信再試行の間隔を調整する」を参照してください。

   ホスト名の代わりに IP アドレスを *smpp-host-name* に指定する場合は、ドメインリテラルを指定します。たとえば、IP アドレスが 127.0.0.1 である場合は、*smpp-host-name* に [127.0.0.1] と指定します。または、SMPP_SERVER チャネルオプションを使用することもできます。

注 Sun Java System Messaging Server 6.1 では、master チャネルキーワードの使用は推奨されていません。存在する場合は無視されます。

3. チャネル定義の追加が終了したら、ファイルの前半に移動し、次の形式で書き換えルールを追加します。

   ```
   smpp-host-name $u@smpp-host-name
   ```

   たとえば、次のように指定します。

   ```
   smpp.siroe.com $u@smpp.siroe.com
   ```

4. imta.cnf ファイルを保存します。
5. imsimta cnbuild コマンドを使用して設定をコンパイルしなおします。
6. imsimta restart dispatcher コマンドを使用して SMTP サーバーを再起動します。

7. 上記の設定では、電子メールメッセージは id@smpp-host-name (例 : 123456@smpp.siroe.com) にアドレス指定することによってチャネルに送信されます。アドレス指定の詳細については、908 ページの「電子メールから SMS への変換プロセス」を参照してください。

8. (任意) SMPP サーバーのホスト名をユーザーに対して非表示にする場合、またはほかのホスト名を同一のチャネルに関連付ける場合は、書き換えルールをさらに追加します。たとえば、host-name-1 と host-name-2 をチャネルに関連付けるには、次の書き換えルールを追加します。

   ```
   host-name-1 $U%host-name-1@smpp-host-name
   host-name-2 $U%host-name-2@smpp-host-name
   ```

   たとえば、SMPP サーバーのホスト名は smpp.siroe.com だが、ユーザーには id@sms.sesta.com 宛に電子メールを送らせたい場合は、次のような書き換えルールを追加します。

   ```
   sms.sesta.com $U%sms.sesta.com@smpp.siroe.com
   ```

SMPP_SERVER および SMPP_PORT チャネルオプションは、チャネルの正式なホスト名および port チャネルキーワードの設定よりも優先されることに注意してください。SMPP_PORT オプションを使用した場合、port キーワードを併せて使用する必要はありません。これら 2 つのオプションを使用する利点は、設定をコンパイルしなおさずに実行でき、その後変更できることです。SMPP_SERVER オプションのその他の使用法については、950 ページの「SMS チャネルをさらに追加する」を参照してください。

同時接続の数を制御する

threaddepth チャネルキーワードは、1 つの配信プロセス内の各配信スレッドに割り当てられるメッセージの数を制御します。許可される同時接続の総数を計算するには、SMPP_MAX_CONNECTIONS および job_limit の 2 つのオプションの値を乗算します (SMPP_MAX_CONNECTIONS * job_limit)。SMPP_MAX_CONNECTIONS オプションは、1 つの配信プロセスでの配信スレッドの最大数を制御します。チャネルが実行されているジョブコントローラ処理プールの job_limit オプションは、同時に実行される配信プロセスの最大数を制御します。

同時接続の総数を制限するには、制限内容に応じてこれらのオプションのどちらかまたは両方を調節する必要があります。たとえば、リモート SMPP サーバーが 1 つの接続しか許可しない場合、SMPP_MAX_CONNECTIONS および job_limit の両方を 1 に設定する必要があります。値を調整するときは、job_limit を 1 よりも大きくすることをお勧めします。

SMS チャネルオプションファイルを作成する

一般的に、チャネルオプションファイルには、チャネルの動作に必要なサイト固有のパラメータが格納されます。SMS にはチャネルオプションファイルは必須ではありません。チャネルオプションファイルが自分のインストールに必要であると判断した場合は、テキストファイル形式で *installation-directory*/config/ ディレクトリに保存します。ほかのチャネルオプションファイルと同じように、ファイル名は次の形式をとります。

*channel_name*_option

たとえば、チャネルの名前が sms_mway である場合、チャネルオプションファイルは次のようになります。

installation-directory/config/sms_mway_option

各オプションは、次の形式を使用して、ファイルの各行に記述します。

option_name=*option_value*

たとえば、次のように指定します。

```
PROFILE=GSM
SMSC_DEFAULT_CHARSET=iso-8859-1
USE_UCS2=1
```

使用可能な SMS チャネルオプションの一覧と各オプションの説明は、後述の「使用可能なオプション」を参照してください。

使用可能なオプション

SMS チャネルには多くのオプションが含まれており、次のように大きく 6 つのカテゴリに分類されます。

- 電子メールから SMS への変換 : 電子メールから SMS への変換プロセスを制御するオプション。
- SMS Gateway Server オプション : ゲートウェイプロファイルのオプション。
- SMS フィールド : 生成された SMS メッセージの SMS 固有のフィールドを制御するオプション。
- SMPP プロトコル : TCP/IP 対応の SMPP プロトコルの使用に関するオプション。
- ローカライズ : SMS メッセージに挿入されたテキストフィールドのローカライズを可能にするオプション。
- その他 : デバッグオプション。

これらのオプションについては、次の表で要約を示し、その後に続く節で詳細を説明します。

表 D-5　SMS チャネルオプション

オプション (ページ番号)	説明	デフォルト
電子メールから SMS への変換オプション		
GATEWAY_NOTIFICATIONS	電子メール通知メッセージを SMS メッセージに変換するかどうかを指定します。	0
MAX_MESSAGE_PARTS	1 つの電子メールメッセージから抽出するメッセージの最大部分数	2
MAX_MESSAGE_SIZE	1 つの電子メールメッセージから抽出する最大バイト数	960
MAX_PAGE_SIZE	単一の SMS メッセージに含める最大バイト数	160
MAX_PAGES_PER_MESSAGE	1 つの電子メールメッセージを分割して生成される最大 SMS メッセージ数	6
ROUTE_TO	指定した IP ホスト名に SMS メッセージをルーティングします。	
SMSC_DEFAULT_CHARSET	SMSC が使用するデフォルトの文字セット	US-ASCII
USE_HEADER_FROM	SMS ソースアドレスを設定します	0
USE_HEADER_PRIORITY	電子メールメッセージのヘッダーにある優先順位情報の使用を制御します	1
USE_HEADER_REPLY_TO	SMS ソースアドレスを生成する際の `Reply-to:` ヘッダー行の使用を制御します	0
USE_HEADER_RESENT	差出人情報を生成する際の `Resent-*:` ヘッダー行の使用を制御します	0
USE_HEADER_SENSITIVITY	電子メールメッセージのヘッダーからのプライバシー情報の使用を制御します	1
USE_UCS2	可能な場合に SMS メッセージで UCS2 文字セットを使用します	1
SMS Gateway Server オプション		
GATEWAY_PROFILE	SMS Gateway Server の設定ファイル (`sms_gateway.cnf`) で設定されたゲートウェイプロファイル名と照合します	なし
SMS フィールドオプション		
DEFAULT_DESTINATION_NPI	SMS 宛先アドレスのデフォルトの NPI	0x00

表 D-5　SMS チャネルオプション (続き)

DEFAULT_DESTINATION_TON	SMS 宛先アドレスのデフォルトの TON	0x01
DEFAULT_PRIORITY	SMS メッセージのデフォルトの優先順位設定	0=GSM、CDMA 1=TDMA
DEFAULT_PRIVACY	SMS メッセージのデフォルトのプライバシー値フラグ	-1
DEFAULT_SERVICE_TYPE	送信された SMS メッセージに関連付けられた SMS アプリケーションサービス	なし
DEFAULT_SOURCE_ADDRESS	デフォルトの SMS ソースアドレス	0
DEFAULT_SOURCE_NPI	SMS ソースアドレスのデフォルトの NPI	0x00
DEFAULT_SOURCE_TON	SMS ソースアドレスのデフォルトの TON	0x01
DEFAULT_VALIDITY_PERIOD	SMS メッセージのデフォルトの有効期間	なし
DESTINATION_ADDRESS_NUMERIC	宛先 SMS アドレスを 0 ～ 9 文字に減らします	0
DESTINATION_ADDRESS_PREFIX	宛先 SMS アドレスの先頭に付けるテキスト文字列	なし
PROFILE	使用する SMS プロファイル	GSM
USE_SAR	sar_ フィールドを使用して、複数の SMS メッセージを配列します	0
SMPP プロトコルオプション		
ESME_ADDRESS_NPI	SMPP サーバーにバインドする際に指定する ESME NPI	0x00
ESME_ADDRESS_TON	SMPP サーバーにバインドする際に指定する ESME TON	0x00
ESME_IP_ADDRESS	Sun Java System Messaging Server を実行しているホストの IP アドレス	なし
ESME_PASSWORD	SMPP サーバーにバインドする際に提示するパスワード	なし
ESME_SYSTEM_ID	バインドする際に SMSC に提示するシステム ID	なし
ESME_SYSTEM_TYPE	バインドする際に SMSC に提示するシステムタイプ	なし
MAX_PAGES_PER_BIND	SMPP サーバーとの 1 回のセッションで送信する最大 SMS メッセージ数	1024
REVERSE_ORDER	マルチパート SMS メッセージの送信シーケンス	0
SMPP_MAX_CONNECTIONS	SMPP サーバーとの最大同時接続数	20

表 D-5　SMS チャネルオプション (続き)

SMPP_PORT	片方向 SMS の場合、SMPP サーバーの待機先 TCP ポート。双方向 SMS の場合、SMPP リレーの LISTEN_PORT に使用されるものと同じ TCP ポート。	なし
SMPP_SERVER	片方向 SMS の場合、接続先の SMPP サーバーのホスト名。 双方向 SMS の場合、SMS Gateway Server のホスト名または IP アドレスをポイントするように設定します。SMPP リレーの LISTEN_INTERFACE_ADDRESS オプションを使用している場合は、指定したネットワークインタフェースアドレスに関連付けられているホスト名または IP アドレスを必ず使用してください。	なし
TIMEOUT	SMPP サーバーでの読み取りおよび書き込み完了までのタイムアウト	30

ローカライズオプション		
CONTENT_PREFIX	電子メールメッセージの内容を導入するためのテキスト	Msg:
DSN_DELAYED_FORMAT	配信遅延通知用の書式設定文字列	空の文字列
DSN_FAILED_FORMAT	配信失敗通知用の書式設定文字列	説明を参照
DSN_RELAYED_FORMAT	リレー通知用の書式設定文字列。	説明を参照
DSN_SUCCESS_FORMAT	配信成功通知用の書式設定文字列。	説明を参照
FROM_FORMAT	電子メールメッセージの差出人を示す場合に表示されるテキスト	$a
FROM_NONE	差出人が存在しない場合に表示されるテキスト	なし
LANGUAGE	(i-default) 言語グループ。この中からテキストフィールドを選択します	i-default
LINE_STOP	電子メールメッセージから抽出された各行の終わりに置くテキスト	スペース文字
NO_MESSAGE	メッセージに内容がないことを示すテキスト	]no message]
SUBJECT_FORMAT	電子メールメッセージの件名を示す場合に表示されるテキスト	$s
SUBJECT_NONE	電子メールメッセージに件名がない場合に表示されるテキスト	なし

表 D-5　SMS チャネルオプション (続き)

その他のオプション		
DEBUG	詳細なデバッグ出力を有効にします	-1

電子メールから SMS への変換オプション

以下のオプションは、電子メールメッセージから SMS メッセージへの変換を制御します。オプションの値の範囲は括弧内に示されています。一般に、1 通の電子メールメッセージは、1 つ以上の SMS メッセージに変換されます。この変換プロセスについては、908 ページの「電子メールから SMS への変換プロセス」を参照してください。

GATEWAY_NOTIFICATIONS

(0 または 1) 電子メール通知を SMS 通知に変換するかどうかを指定します。電子メール通知メッセージは、RFC 1892、1893、1894 に準拠している必要があります。デフォルト値は 0 です。

GATEWAY_NOTIFICATIONS=0 の場合、このような通知は破棄され、SMS 通知に変換されません。

通知の SMS 通知への変換を有効にするには、GATEWAY_NOTIFICATIONS=1 に設定します。このオプションが 1 に設定されている場合、ローカライズオプション (DSN_*_FORMAT) によって、SMS メッセージに変換されてゲートウェイから送信される通知のタイプ (成功、失敗、遅延、リレー) が制御されます。通知タイプの値が空の文字列である場合、そのタイプの通知は SMS メッセージに変換されません。

MAX_MESSAGE_PARTS

(整数) マルチパート電子メールメッセージを SMS メッセージに変換する場合、テキスト部分のうち最初の MAX_MESSAGE_PARTS 数のみが変換されます。残りの部分はページされます。デフォルトでは、MAX_MESSAGE_PARTS は 2 です。メッセージ部分を無制限に許可するには、値に -1 を指定します。値を 0 にすると、SMS メッセージに変換されるメッセージコンテンツはありません。これには、SMS メッセージを生成するために電子メールメッセージのヘッダー行 (たとえば、Subject:) のみを使用した効果があります。

テキストと添付ファイルの両方を含む電子メールメッセージは、通常、2 つの部分で構成されています。プレーンテキストのメッセージ部分のみが変換されることに注意してください。その他の MIME コンテンツタイプはすべてページされます。

MAX_MESSAGE_SIZE

(整数、>= 10) このオプションを使用して、1 つの電子メールメッセージから生成された SMS メッセージに含める合計バイト数の上限を設定できます。特に、MAX_MESSAGE_SIZE バイトの最大値は、1 つ以上の生成された SMS メッセージに使用されます。それ以上のバイトはパージされます。

デフォルトでは、960 バイトが上限となります。これは MAX_MESSAGE_SIZE=960 に相当します。任意のバイト数を使用するには、値に 0 を指定します。

使用されるバイト数の計算は、電子メールメッセージを Unicode から SMSC のデフォルト文字セットまたは UCS2 に変換してから行います。つまり、UCS2 を例にすると、UCS2 の各文字は、最低でも 2 バイト長であるため、MAX_MESSAGE_SIZE が 960 バイトだと、最高でも 480 文字しか確保できません。

MAX_MESSAGE_SIZE および MAX_PAGES_PER_MESSAGE の各オプションは、どちらも結果の SMS メッセージの全体サイズを制限するという同じ目的で機能します。実際、MAX_PAGE_SIZE=960 と MAX_PAGE_SIZE=160 は、MAX_PAGES_PER_MESSAGE=6 を意味します。2 つの異なるオプションが存在する理由は、単一の SMS メッセージの最大サイズを考慮せずに全体のサイズまたはページ数を制御するのに、MAX_PAGE_SIZE が役立つからです。このことはチャネルオプションファイルでは重要ではないかもしれませんが、906 ページの「電子メールをチャネルに送信する」で説明されている MAXPAGES または MAXLEN アドレス指定属性を使用する際には重要です。

さらに、MAX_MESSAGE_SIZE と MAX_PAGE_SIZE * MAX_PAGES_PER_MESSAGE のどちらか小さい制限が使用されることに注意してください。

MAX_PAGE_SIZE

(整数、>= 10) 単一の SMS メッセージで許可される最大バイト数は、MAX_PAGE_SIZE オプションで制御します。デフォルトでは、160 バイトが値として用いられます。これは、MAX_PAGE_SIZE=160 に相当します。

MAX_PAGES_PER_MESSAGE

(整数、1 ～ 255) 1 つの電子メールメッセージに生成される最大 SMS メッセージ数は、このオプションで制御します。事実上、このオプションによって電子メールメッセージには切り捨てが実行されます。MAX_PAGES_PER_MESSAGE の SMS メッセージ数に収まる電子メールメッセージの部分のみが SMS メッセージに変換されます。詳細は、MAX_PAGE_SIZE オプションの説明を参照してください。

デフォルトでは、MAX_PAGES_PER_MESSAGE は 1、または MAX_MESSAGE_SIZE を MAX_PAGE_SIZE で割った数のうちの大きいほうに設定されています。

ROUTE_TO

(文字列、IP ホスト名、1 ～ 64 バイト) プロファイルにターゲットされたすべての SMS メッセージは、指定されている IP ホスト名に再ルートされます。このとき、次の形式の電子メールアドレスが使用されます。

```
SMS-destination-address@route-to
```

SMS-destination-address は SMS メッセージの宛先アドレスで、route-to はこのオプションで指定されている IP ホスト名です。SMS メッセージの内容全体は、結果の電子メールメッセージの内容として送信されます。PARSE_RE_* オプションは無視されます

注	PARSE_RE_* と ROUTE_TO の各オプションの使用は、互いに排他的です。これらの両方を同一のゲートウェイプロファイルで使用すると、設定エラーになります。

SMSC_DEFAULT_CHARSET

(文字列) このオプションを使用して、SMSC のデフォルトの文字セットを指定します。次のファイルに示されている文字セット名を使用してください。

installation-directory/config/charsets.txt

このオプションが指定されていない場合は、US-ASCII であると仮定されます。なお、charsets.txt で使用されるニーモニック名は、同じディレクトリの charnames.txt で定義されています。

電子メールの処理では、まずヘッダー行とテキストメッセージ部分がデコードされてから Unicode に変換されます。次に、データは SMSC のデフォルトの文字セットまたは UCS2 に変換されます。どちらに変換されるかは、USE_UCS2 オプションの値および SMS メッセージにデフォルトの文字セットにないグリフが 1 つでも含まれているかどうかによって異なります。UCS2 文字セットは、Unicode の 16 ビットのエンコード方式であり、UTF-16 と呼ばれることもあります。

USE_HEADER_FROM

(整数、0 ～ 2) このオプションは、From: アドレスを SMSC に渡すことを許可する場合に設定します。値は、From: アドレスを取り出す場所とアドレスの形式を示します。表 D-6 に、許容可能な値とその意味を示します。

表 D-6　USE_HEADER_FROM の値

値	説明
0	SMS ソースアドレスは From: アドレスから設定されません。見つかった属性と値のペアを使用してください

表 D-6　USE_HEADER_FROM の値 (続き)

値	説明
1	SMS ソースアドレスを from-local@from-domain に設定します。この場合、From: アドレスは from-route:from-local@from-domain
2	SMS ソースアドレスを from-local に設定します。この場合、From: アドレスは from-route:from-local@from-domain

USE_HEADER_PRIORITY

(0 または 1) このオプションで RFC 822 Priority: ヘッダー行の処理を制御します。デフォルトでは、Priority: ヘッダー行の情報は結果の SMS メッセージの優先順位フラグを設定するために使用され、DEFAULT_PRIORITY オプションで指定されているデフォルトの SMS 優先順位よりも優先されます。これは、USE_HEADER_PRIORITY=1 に相当します。RFC 822 Priority: ヘッダー行の使用を無効にするには、USE_HEADER_PRIORITY=0 を指定します。

SMS 優先順位フラグの処理の詳細については、DEFAULT_PRIORITY オプションの説明を参照してください。

USE_HEADER_REPLY_TO

(0 または 1) USE_HEADER_FROM =1 の場合、このオプションは Reply-to: または Resent-reply-to: ヘッダー行が SMS ソースアドレスとして使用されることを考慮するかどうかを制御します。デフォルトでは、Reply-to: および Resent-reply-to: ヘッダー行は無視されます。これはオプションの値 0 に相当します。これらのヘッダー行を考慮するようにするには、オプションの値として 1 を使います。

RFC 2822 では、Reply-to: および Resent-reply-to: ヘッダー行の使用は推奨されていないことに注意してください。

USE_HEADER_RESENT

(0 または 1) USE_HEADER_FROM =1 の場合、このオプションは Resent- ヘッダー行が SMS ソースアドレスとして使用されることを考慮するかどうかを制御します。デフォルトでは、Resent- ヘッダー行は無視されます。これはオプションの値 0 に相当します。これらのヘッダー行を考慮するようにするには、オプションの値として 1 を使います。

RFC 2822 では、Resent- ヘッダー行の使用は推奨されていないことに注意してください。

USE_HEADER_SENSITIVITY

(0 または 1) USE_HEADER_SENSITIVITY オプションは、RFC 822 Sensitivity: ヘッダー行の処理を制御します。デフォルトでは、Sensitivity: ヘッダー行の情報は結果の SMS メッセージのプライバシーフラグを設定するために使用され、DEFAULT_PRIVACY オプションで指定されているデフォルトの SMS プライバシーよりも優先されます。これがデフォルトで、USE_HEADER_SENSITIVITY=1 に相当します。RFC 822 Sensitivity: ヘッダー行の使用を無効にするには、USE_HEADER_SENSITIVITY=0 と指定します。

SMS プライバシーフラグの処理の詳細については、DEFAULT_PRIVACY オプションの説明を参照してください。

USE_UCS2

(0 または 1) 適切な場合に、チャネルは生成する SMS メッセージで UCS2 文字セットを使用します。これはデフォルトの動作であり、USE_UCS2=1 に相当します。UCS2 文字セットの使用を無効にするには、USE_UCS2=0 を指定します。文字セットの詳細については、SMSC_DEFAULT_CHARSET オプションの説明を参照してください。

表 D-7　　値 USE_UCS2 で有効な値

USE_UCS2 の値	結果
1 (デフォルト)	可能な場合は常に SMSC のデフォルトの文字セットが使用されます。元の電子メールメッセージに SMSC のデフォルトの文字セットにないグリフが含まれている場合は、UCS2 文字セットが使用されます。
0	常に SMSC のデフォルトの文字セットが使用されます。その文字セットで使用不可なグリフはニーモニックで表現されます (例 : AE の合字を「AE」で表現)。

SMS Gateway Server オプション

GATEWAY_PROFILE

SMS Gateway Server の設定ファイル sms_gateway.cnf のゲートウェイプロファイルの名前です。

SMS オプション

以下のオプションを使用して、生成された SMS メッセージの SMS フィールドに関する指定が行えます。

DEFAULT_DESTINATION_NPI

(整数、0 ～ 255) デフォルトでは、宛先アドレスには 0 の NPI (番号計画識別子) 値が割り当てられます。このオプションを使用すると、0 から 255 までの範囲の代替整数値を割り当てることができます。表 D-8 に、これらを含む一般的な NPI 値を示します。

表 D-8 番号計画識別子の値

値	説明
0	不明
1	ISDN (E.163、E.164)
3	データ (X.121)
4	テレックス (F.69)
6	地上モバイル (E.212)
8	国内
9	プライベート
10	ERMES
14	IP アドレス (インターネット)
18	WAP クライアント ID
>= 19	未定義

このオプションの値は、次の 3 つのいずれかの方法で指定します。

- 10 進数値 (例 : 10)
- 「0x」のプレフィックスが付いた 16 進値 (例 : 0x0a)
- 次に示す、大文字と小文字が区別されるテキスト文字列 (括弧内は関連付けられている 10 進値)。data (3)、default (0)、e.163 (1)、e.164 (1)、e.212 (6)、ermes (10)、f.69 (4)、Internet (14)、ip (14)、isdn (1)、land-mobile (6)、national (8)、private (9)、telex (4)、unknown (0)、wap (18)、x.121 (3)。

DEFAULT_DESTINATION_TON

(整数、0 ～ 255) デフォルトでは、宛先アドレスには 0 の TON (番号種別) 値が割り当てられています。このオプションを使用すると、0 から 255 までの範囲の代替整数値を割り当てることができます。表 D-9 に、これらを含む一般的な TON 値を示します。

表 D-9　　一般的な TON 値

値	説明
0	不明
1	国際
2	国内
3	ネットワーク固有
4	加入者番号
5	英数字
6	略記
>=7	未定義

このオプションの値は、次の 3 つのいずれかの方法で指定します。

- 10 進値 (例 : 10)
- 「0x」のプレフィックスが付いた 16 進値 (例 : 0x0a)
- 次に示す、大文字と小文字が区別されるテキスト文字列 (括弧内は関連付けられている 10 進値)。abbreviated (6)、alphanumeric (5)、default (0)、international (1)、national (2)、network-specific (3)、subscriber (4)、munknown (0)。

DEFAULT_PRIORITY

(整数、0 ～ 255) SMS メッセージには必須の優先順位フィールドがあります。表 D-10 に、SMS 優先順位値の解釈を示します。

表 D-10　　各 SMS プロファイルタイプごとに解釈される SMS 優先順位値

値	GSM	TDMA	CDMA
0	優先でない	バルク	標準
1	優先	標準	インタラクティブ
2	優先	至急	至急
3	優先	大至急	緊急

このオプションを使用すると、SMS メッセージに割り当てるデフォルトの優先度を指定できます。指定しない場合は、デフォルトの優先度 0 が PROFILE=GSM および CDMA に使用され、優先度 1 が PROFILE=TDMA に使用されます。

USE_HEADER_PRIORITY=1 であり、電子メールメッセージに RFC 822 Priority: ヘッダー行がある場合は、このヘッダー行に指定された優先順位が結果の SMS メッセージの優先順位の設定に使用されます。USE_HEADER_PRIORITY=0 の場合、SMS 優先順位フラグは常に DEFAULT_PRIORITY オプションに合わせて設定され、RFC 822 Priority: ヘッダー行は常に無視されます。USE_HEADER_PRIORITY=1 の場合、元の電子メールメッセージの RFC 822 Priority: ヘッダー行が SMS メッセージの優先順位フラグの設定に使用されます。このヘッダー行が存在しない場合、SMS 優先順位フラグは DEFAULT_PRIORITY オプションを使用して設定されます。

RFC 822 Priority: ヘッダー行の値を SMS 優先順位フラグに変換するために使用されるマッピングを次の表に示します。

表 D-11　Priority: ヘッダーから SMS 優先順位フラグに変換するためのマッピング

RFC 822	SMS 優先順位フラグ		
Priority: の値	GSM	TDMA	CDMA
Third	優先でない (0)	バルク (0)	標準 (0)
Second	優先でない (0)	バルク (0)	標準 (0)
Non-urgent	優先でない (0)	バルク (0)	標準 (0)
Normal	優先でない (0)	標準 (1)	標準 (0)
Urgent	優先 (1)	至急 (2)	至急 (2)

DEFAULT_PRIVACY

(整数、-1、0 ～ 255) DEFAULT_PRIVACY オプションと USE_HEADER_SENSITIVITY オプションでは、SMS メッセージにプライバシーフラグを設定するかどうか、またどの値を使用するかを制御します。デフォルトでは、値 -1 は DEFAULT_PRIVACY に使用されます。表 D-12 に、DEFAULT_PRIVACY および USE_HEADER_SENSITIVITY の各オプションにさまざまな値を設定した結果を示します。

表 D-12　DEFAULT_PRIVACY と USE_HEADER_SENSITIVITY の値の結果

DEFAULT_PRIVACY	USE_HEADER_SENSITIVITY	結果
-1	0	SMS プライバシーフラグは SMS メッセージに設定されません。
n >= 0	0	SMS プライバシーフラグは常に値 n に設定されます。RFC 822 Sensitivity: ヘッダー行は常に無視されます。

表 D-12 DEFAULT_PRIVACY と USE_HEADER_SENSITIVITY の値の結果 (続き)

DEFAULT_PRIVACY	USE_HEADER_SENSITIVITY	結果
-1 (デフォルト)	1 (デフォルト)	SMS メッセージのプライバシーフラグは、元の電子メールメッセージに RFC 822 Sensitivity: ヘッダー行があるときのみ設定されます。その場合、SMS プライバシーフラグは Sensitivity: ヘッダー行の値と対応するように設定されます。これがデフォルトです。
n >= 0	1	SMS メッセージのプライバシーフラグは、元の電子メールメッセージの RFC 822 Sensitivity: ヘッダー行と対応するように設定されます。電子メールメッセージに Sensitivity: ヘッダー行がない場合は、SMS プライバシーフラグの値は n に設定されます。

表 D-13 に、SMS プライバシー値の解釈を示します。

表 D-13 SMS プライバシー値の解釈

値	説明
0	制限なし
1	制限あり
2	親展
3	秘密
>= 4	未定義

表 D-14 に、RFC 822 Sensitivity: ヘッダー行の値を SMS プライバシー値に変換するために使用されるマッピングを示します。

表 D-14 Sensitivity: ヘッダーから SMS プライバシー値へのマッピング変換

RFC 822 Sensitivity: の値	SMS プライバシー値
Personal	1 (制限あり)
Private	2 (親展)
Company confidential	3 (秘密)

DEFAULT_SERVICE_TYPE

(文字列、0 ～ 5 バイト) チャネルによって生成された SMS メッセージに関連付けるサービスタイプ。デフォルトでは、指定されているサービスタイプはありません (つまり、長さ 0 の文字列)。一般的なサービスタイプには次のものがあります。CMT (携帯電話メッセージング)、CPT (携帯電話ページング)、VMN (ボイスメール通知)、VMA (ボイスメール呼び出し)、WAP (無線アプリケーションプロトコル)、および USSD (非構造化補足データサービス)。

DEFAULT_SOURCE_ADDRESS

(文字列、0 ～ 20 バイト) 電子メールメッセージから生成された SMS メッセージに使用されるソースアドレス。USE_HEADER_FROM=1 の場合、このオプションで指定した値よりも電子メールメッセージの差出人のアドレスが優先されることに注意してください。デフォルトでは、値は無効になっています。つまり、値として 0 が設定されています。

DEFAULT_SOURCE_NPI

(整数、0 ～ 255) デフォルトでは、ソースアドレスには 0 の NPI 値が割り当てられています。このオプションを使用すると、0 から 255 までの範囲の代替整数値を割り当てることができます。一般的な NPI 値の表にある DEFAULT_DESTINATION_NPI オプションの説明を参照してください。

DEFAULT_SOURCE_TON

(整数、0 ～ 255) デフォルトでは、ソースアドレスには 0 の TON 指定子値が割り当てられています。このオプションを使用すると、0 から 255 までの範囲の代替整数値を割り当てることができます。一般的な TON 値の表にある DEFAULT_DESTINATION_TON オプションの説明を参照してください。

DEFAULT_VALIDITY_PERIOD

(文字列、0 ～ 252 バイト) デフォルトでは、SMS メッセージには相対有効期間は指定されていません。代わりに、SMSC のデフォルト値が使用されます。このオプションは別の相対有効期間を指定するために使用します。値は、秒、分、時、または日の各単位で指定できます。表 D-15 に、このオプションに使用するさまざまな値の形式と説明を示します。

表 D-15 DEFAULT_VALIDITY_PERIOD の形式と値

形式	説明
nnn	黙示的な秒単位。例 : 604800
*nnn*s	秒単位。例 : 604800s
*nnn*m	分単位。例 : 10080m
*nnn*h	時単位。例 : 168h

表 D-15　DEFAULT_VALIDITY_PERIOD の形式と値 (続き)

形式	説明
*nnn*d	日単位。例 : 7d

0、0s、0m、0h、または 0d という指定が SMSC のデフォルトの有効期間を選択するために使用されることがあります。0、0s、0m、0h、または 0d という指定が使用された場合は、生成された SMS メッセージの有効期間に空の文字列が指定されます。

このオプションには UTC 形式の値を使用できないので注意してください。

DESTINATION_ADDRESS_NUMERIC

(0 または 1) 電子メールエンベロープ To: アドレスから抽出された SMS 宛先アドレスからすべての非数値文字を削除するには、このオプションを使用します。たとえば、次のエンベロープ To: アドレスがあるとします。

```
"(800) 555-1212"@sms.siroe.com
```

このアドレスは次のように短くなります。

```
8005551212@sms.siroe.com
```

このような削除を有効にするには、このオプションの値に 1 を指定します。デフォルトでは、この削除処理は無効になっています。これはオプションの値が 0 である場合に相当します。有効になっている場合は、DESTINATION_ADDRESS_PREFIX オプションによって宛先アドレスのプレフィックスが追加される前に削除が実行されることに注意してください。

DESTINATION_ADDRESS_PREFIX

(文字列) すべての SMS 宛先アドレスの先頭に固定テキスト文字列 (たとえば「+」) が必ず付いていることが必要な場合があります。このオプションはこのようなプレフィックスを指定するために使用します。プレフィックスは、指定したプレフィックスが付いていないすべての SMS 宛先アドレスに追加されます。
DESTINATION_ADDRESS_NUMERIC オプションによって削除されないようにするため、このオプションは DESTINATION_ADDRESS_NUMERIC オプションのあとに適用されます。

PROFILE

(文字列) SMSC で使用される SMS プロファイルを指定します。指定できる値は、GSM、TDMA、および CDMA です。指定されていない場合は、GSM と仮定されます。このオプションは、DEFAULT_PRIORITY や DEFAULT_PRIVACY などのほかのチャネルオプション用のデフォルトを選択するためにのみ使用されます。

USE_SAR

(0 または 1) サイズの大きい電子メールメッセージは、複数の SMS メッセージに分割される場合があります。このとき、SMS sar_ フィールドを使用すると、個別の SMS メッセージに順番情報を付加することができます。これにより、「セグメント化された」SMS メッセージが生成され、受信端末で 1 つの SMS メッセージに再構成できます。USE_SAR=1 を指定すると、可能な場合にこの順番情報が付加されます。デフォルトでは順番情報を付加しません。これは USE_SAR=0 に相当します。

USE_SAR=1 が設定されている場合、REVERSE_ORDER オプションは無視されます。

SMPP オプション

以下のオプションを使用して、SMPP プロトコルパラメータに関する指定が行えます。文字列 ESME_ で始まる名前のオプションは、MTA が External Short Message Entity (ESME) として動作するときに MTA を特定するために使用します。つまりそれは、SMS メッセージをサーバーに関連付けされた SMSC に送信するために、MTA が SMPP サーバーにバインドされているときです。

ESME_ADDRESS_NPI

(整数、0 ～ 255) デフォルトでは、不明な NPI を示す 0 の ESME NPI 値がバインド動作によって指定されます。このオプションを使用すると、0 から 255 までの範囲の代替整数値を割り当てることができます。一般的な NPI 値の表にある DEFAULT_DESTINATION_NPI オプションの説明を参照してください。

ESME_ADDRESS_TON

(整数、0 ～ 255) デフォルトでは、バインド動作によって 0 の ESME TON 値が指定されます。このオプションを使用すると、0 ～ 255 の範囲で別の整数値を割り当てることができます。一般的な TON 値の表にある DEFAULT_DESTINATION_TON オプションの説明を参照してください。

ESME_IP_ADDRESS

(文字列、0 ～ 15 バイト) SMPP サーバーにバインドするとき、BIND PDU は、クライアントの (つまり、ESME の) アドレス範囲が IP アドレスであることを示します。これは、TON に 0x00、NPI に 0x0d を指定して行います。アドレス範囲フィールドの値は、SMS チャネルを実行するホストの IP アドレスに設定されます。IP アドレスは「127.0.0.1」のように、ドット付きの 10 進形式で指定します。

ESME_PASSWORD

(文字列、0 ～ 8 バイト) SMPP サーバーにバインドするとき、パスワードを要求される場合があります。その場合は、このオプションでそのパスワードを指定します。デフォルトでは、長さ 0 のパスワード文字列が指定されています。

ESME_SYSTEM_ID

(文字列、0 ～ 15 バイト) SMPP サーバーにバインドするとき、MTA のシステム ID を提示する場合があります。デフォルトでは、システム ID は指定されていません。つまり、長さゼロの文字列が使用されています。システム ID を指定するには、このオプションを使用します。

ESME_SYSTEM_TYPE

(文字列、0 ～ 12 バイト) SMPP サーバーにバインドするとき、MTA のシステムタイプを提示する場合があります。デフォルトでは、指定されているシステムタイプはありません。つまり、長さ 0 の文字列が使用されます。

MAX_PAGES_PER_BIND

(整数 >= 0) SMPP サーバーのなかには、1 回のバインドセッション中に送信される最大 SMS メッセージ数を制限するものもあります。このため、このオプションではシングルセッション中に送信される SMS メッセージの最大数について、規定することができます。この制限に達すると、チャネルはアンバインドして TCP/IP 接続を終了してから、再接続し、再バインドします。

MAX_PAGES_PER_BIND のデフォルトは 1024 です。チャネルでは、ESME_RTHROTTLED エラーも検出され、1 回のチャネルの実行中に必要に応じて MAX_PAGES_PER_BIND が調整されます。

REVERSE_ORDER

(0 または 1) 電子メールメッセージから複数の SMS メッセージが生成された場合、それらの SMS メッセージは順次 (REVERSE_ORDER=0) または逆順 (REVERSE_ORDER=1) で SMSC に送信できます。受信端末が最後に受信したメッセージから先に表示するような場合は、逆順が便利です。そのような場合、最後に受信したメッセージは、電子メールの末尾部分ではなく、先頭部分になります。デフォルトでは REVERSE_ORDER=1 が使用されます。

このオプションは、USE_SAR=1 が指定されている場合は無視されることに注意してください。

SMPP_MAX_CONNECTIONS

(整数、1 ～ 50) このオプションは、処理ごとの最大同時 SMPP 接続数を制御します。それぞれの接続には関連付けされたスレッドがあるが、このオプションでは、プロセスごとの「ワーカー」スレッドの最大数も制限します。デフォルトは SMPP_MAX_CONNECTIONS=20 です。

SMPP_PORT

(整数、1 ～ 65535) SMPP サーバーが待機する TCP ポートは、このオプションまたは port チャネルキーワードのどちらかで指定します。このポート番号は、それら 2 つのメカニズムのどちらかで指定する必要があります。両方のメカニズムで指定した場合は、SMPP_PORT オプションによる設定が優先されます。このオプションのデフォルト値はありません。

双方向 SMS の場合は、このポートが SMPP リレーの LISTEN_PORT と同じポートであることを確認してください。

SMPP_SERVER

(文字列、1 ～ 252 バイト) 片方向 SMS の場合、デフォルトの接続先 SMPP サーバーの IP ホスト名は、チャネルに関連付けられた正式なホスト名 (MTA 設定のチャネル定義の 2 行目に示されているホスト名) です。このオプションは、別のホスト名または IP アドレスを指定するために使用します。このオプションでの指定はチャネル定義での指定より優先されます。IP アドレスを指定する際は、ドット付きの 10 進表記で指定します (例 : 127.0.0.1)。

双方向 SMS の場合は、SMS Gateway Server のホスト名または IP アドレスをポイントするように設定します。SMPP リレーの LISTEN_INTERFACE_ADDRESS オプションを使用している場合は、指定したネットワークインタフェースアドレスに関連付けられているホスト名または IP アドレスを必ず使用してください。

TIMEOUT

(整数、>= 2) デフォルトでは、SMPP サーバーへのデータ書き込みが完了するまで、またはデータが SMPP サーバーから受信されるまでに 30 秒のタイムアウトが使用されます。別のタイムアウト値 (秒) を指定するには、TIMEOUT オプションを使用します。指定する値は 1 秒以上にしてください。

ローカライズオプション

SMS チャネルには、SMS メッセージの作成時に SMS メッセージに付加するいくつかの固定テキスト文字列があります。これらの文字列は、たとえば電子メールの `From:` アドレスや `Subject:` ヘッダー行に使用されます。この節で説明されているチャネルオプションを使用して、さまざまな言語用にこれらの文字列のバージョンを指定し、その後チャネルのデフォルト言語を指定できます。コード例 D-2 に、オプションファイルの言語部分を示します。

コード例 D-2 チャネルオプションファイルの言語指定部分

```
LANGUAGE=default-language

[language=i-default]
FROM_PREFIX=From:
SUBJECT_PREFIX=Subj:
CONTENT_PREFIX=Msg:
LINE_STOP=
NO_MESSAGE=[no message]
REPLY_PREFIX=Re:

[language=en]
FROM_PREFIX=From:
SUBJECT_PREFIX=Subj:
CONTENT_PREFIX=Msg:
LINE_STOP=
NO_MESSAGE=[no message]
REPLY_PREFIX=Re:
 ...
```

それぞれの `[language=x]` ブロックでは、その言語に関するローカライズオプションを指定します。ブロック内の特定のオプションが指定されていない場合は、そのオプションのグローバル値が使用されます。`[language=x]` ブロックの外で指定されたローカライズオプションが、そのオプションのグローバル値になります。

これから示すオプションでは、文字列値を US-ASCII または UTF-8 文字セットで指定する必要があります。US-ASCII 文字セットは、UTF-8 文字セットの特殊な場合です。

CONTENT_PREFIX

(文字列、0 ～ 252 バイト) SMS メッセージで電子メールメッセージの内容自体の前に置くテキスト文字列。デフォルトのグローバル値は US-ASCII 文字列「`Msg:`」です。

DSN_DELAYED_FORMAT

(文字列、0 ～ 256 文字) 配信遅延通知用の書式設定文字列。デフォルトでは、このオプションには空の文字列が使用されています。この場合、遅延通知の SMS への変換は行われません。このオプションを有効にするには、GATEWAY_NOTIFICATIONS を 1 に設定する必要があります。GATEWAY_NOTIFICATIONS=0 の場合、このオプションは無視されます。

DSN_FAILED_FORMAT

(文字列、0 ～ 256 文字) 永久的な配信失敗通知用の書式設定文字列。このオプションのデフォルト値は次の文字列です。

```
Unable to deliver your message to $a; no further delivery attempts
will be made.
```

失敗通知の変換が行われないようにするには、このオプションに空の文字列を指定します。このオプションを有効にするには、GATEWAY_NOTIFICATIONS を 1 に設定する必要があります。GATEWAY_NOTIFICATIONS=0 の場合、このオプションは無視されます。

DSN_RELAYED_FORMAT

(文字列、0 ～ 256 文字) リレー通知用の書式設定文字列。デフォルト値は次の文字列です。

```
Your message to $a has been relayed to a messaging system which may
not provide a final delivery confirmation
```

リレー通知の変換が行われないようにするには、このオプションに空の文字列を指定します。このオプションを有効にするには、GATEWAY_NOTIFICATIONS を 1 に設定する必要があります。GATEWAY_NOTIFICATIONS=0 の場合、このオプションは無視されます。

DSN_SUCCESS_FORMAT

(文字列、0 ～ 256 文字) 配信成功通知用の書式設定文字列。デフォルト値は次の文字列です。

```
Your message to $a has been delivered
```

配信成功通知の変換が行われないようにするには、このオプションに空の文字列を指定します。このオプションを有効にするには、GATEWAY_NOTIFICATIONS を 1 に設定する必要があります。GATEWAY_NOTIFICATIONS=0 の場合、このオプションは無視されます。

FROM_FORMAT

(文字列、0 ～ 252 バイト) SMS メッセージに挿入する差出人情報を書式設定するテンプレート。デフォルトのグローバル値は US-ASCII 文字列「$a」で、差出人の電子メールアドレスに置換されます。詳細は、948 ページの「書式設定テンプレート」を参照してください。

FROM_NONE

(文字列、0 ～ 252 バイト) 表示する差出人のアドレスがない場合に SMS メッセージに置くテキスト文字列。デフォルトのグローバル値は空の文字列です。

通常、このオプションは使用されません。一般的に、差出人のアドレスがない電子メールメッセージはサイトで拒否されるからです。

LANGUAGE

(文字列、0 ～ 40 バイト) テキスト文字列の選択源になるデフォルトの言語グループ。指定されていない場合、言語はホストのデフォルトのロケール指定から生成されます。ホストのロケール設定が利用できない場合や、「C」に対応する場合は、i-default が使用されます。i-default は、「国際的な対象者を意図した英文テキスト」に相当します。

LINE_STOP

(文字列、0 ～ 252 バイト) SMS メッセージで電子メールメッセージから抽出された行間に置くテキスト文字列。デフォルトのグローバル値は、US-ASCII スペース文字の「 」です。

NO_MESSAGE

(文字列、0 ～ 252 バイト) SMS メッセージに置く、電子メールメッセージに内容がないことを示すテキスト文字列。デフォルトのグローバル値は US-ASCII 文字列「`[no message]`」

SUBJECT_FORMAT

(文字列、0 ～ 252 バイト) SMS メッセージでの表示用に、`Subject:` ヘッダー行の内容を書式設定するテンプレート。このオプションのデフォルトのグローバル値は、US-ASCII 文字列の「`($s)`」です。詳細は、948 ページの「書式設定テンプレート」を参照してください。

`Subject:` ヘッダー行がない場合や、空の文字列である場合の処理については、`SUBJECT_NONE` オプションを参照してください。

SUBJECT_NONE

(文字列、0 ～ 252 バイト) 元の電子メールメッセージに `Subject:` ヘッダー行がないか、`Subject:` ヘッダー行の値が空の文字列の場合に表示するテキスト文字列。このオプションのデフォルトのグローバル値は空の文字列です。

DEBUG

(整数、ビットマスク) デバッグ出力を有効にします。デフォルト値は 6 であり、警告およびエラーメッセージが選択されます。ゼロ以外の値を指定すると、チャネル定義で master_debug を指定した場合と同じく、チャネル自体のデバッグ出力が有効になります。表 D-16 に、DEBUG ビットマスクのビット値を示します。

表 D-16　DEBUG ビットマスク

ビット	値	説明
0-31	-1	きわめて詳細な出力
0	1	情報メッセージ
1	2	警告メッセージ
3	4	エラーメッセージ
3	8	サブルーチン呼び出しのトレース
4	16	ハッシュテーブル診断
5	32	I/O 診断、受信
6	64	I/O 診断、送信
7	128	SMS から電子メールへの変換診断 (モバイルからの発信および SMS 通知)
8	256	PDU 診断、ヘッダーデータ
9	512	PDU 診断、本文データ
10	1024	PDU 診断、タイプの長さ値のデータ
11	2048	オプション処理。すべてのオプション設定をログファイルに送ります。

書式設定テンプレート

FROM_FORMAT、SUBJECT_FORMAT、およびすべての DSN_* チャネルオプションで指定される書式設定テンプレートは、UTF-8 文字列です。これには、リテラルテキストと置換シーケンスの組み合わせが含まれている場合があります。例として次の電子メールアドレスを使用します。

```
Jane Doe <user@siroe>
```

表 D-17 に、認識される置換シーケンスを示します。

表 D-17　置換シーケンス

シーケンス	説明
$a	差出人の電子メールアドレスのローカル部分とドメイン部分で置き換えます (例 :「user@siroe」)
$d	差出人の電子メールアドレスのドメイン部分で置き換えます (例 :「domain」)
$p	差出人の電子メールアドレスのフレーズ部分 (ある場合) で置き換えます (例 :「Jane Doe」)
$s	Subject: ヘッダー行の内容で置き換えます
$u	差出人の電子メールアドレスローカル部分で置き換えます (例 :「user」)
¥x	リテラル文字「x」で置き換えます

たとえば、次の書式設定テンプレートがあるとします。

```
From: $a
```

このテンプレートは、次のテキスト文字列を生成します。

```
From: user@siroe
```

コンストラクタ

```
${xy:alternate text}
```

がシーケンス x に関連付けられたテキストで置き換えるために使用される場合があります。このテキストが空の文字列である場合は、シーケンス y に関連付けられたテキストが代わりに使用されます。さらにこのテキストが空の文字列である場合は、代替テキストで置き換えられます。たとえば、次の書式設定テンプレートがあるとします。

```
From: ${pa:unknown sender}
```

これを次の差出人の電子メールアドレスに適用します。

```
John Doe <jdoe@siroe.com>
```

このアドレスにはフレーズ部分があるので、テンプレートによって次の結果が生成されます。

```
From: John Doe
```

今度は、次のアドレスに適用します。

```
jdoe@siroe.com
```

このアドレスにはフレーズがないので、次の結果になります。

```
From: jdoe@siroe.com
```

さらに、空の差出人アドレスに適用すると、次の結果になります。

```
From: unknown sender
```

SMS チャネルをさらに追加する

MTA が複数の SMS チャネルを持つように設定することができます。一般的に、これを行う理由は 2 つあります。

1. 異なる SMPP サーバーと通信するため

   この理由はきわめてわかりやすいものです。設定に SMS チャネルを増やすだけで、(a) 別のチャネル名を付与できる (b) 別のホスト名を関連付けできるからです。たとえば、次のように指定します。

   ```
   sms_mway port 55555 threaddepth 20
   smpp.siroe.com
   ```

   ```
   sms_ace port 777 threaddepth 20
   sms.ace.net
   ```

   新しい書き換えルールは不要です。直接一致する書き換えルールがない場合は、Messaging Sever が関連ホスト名を使用してチャネルを検索します。たとえば、`user@host.domain` とともにサーバーが提示されている場合、「host.domain」という名前のチャネルを検索します。このチャネルを見つけた場合は、そのチャネルにメッセージをルーティングします。これ以外の場合は、「.domain」の書き換えルールを検索し、該当するものがない場合は、ドット (「.」) ルールを検索します。書き換えルールの詳細については、291 ページの第 11 章「書き換えルールの設定」を参照してください。

2. 別のチャネルオプションを使用して同一の SMPP サーバーと通信するため

   別のチャネルオプションを使用して同一の SMPP サーバーと通信するには、各チャネル定義の SMPP_SERVER チャネルオプションでその SMPP サーバーを指定します。

   2 つの異なるチャネルは同一の正式ホスト名称 (チャネル定義の 2 行目に示されるホスト名) を持つことができないため、上記のメカニズムを使用する必要があります。異なるチャネルで同一の SMPP サーバーと通信できるようにするには、それぞれのチャネルオプションファイルの SMPP_SERVER でその SMPP サーバーを指定して、2 つの別個のチャネルを定義します。

   たとえば、次のようなチャネル定義をすることができます。

   ```
   sms_mway_1 port 55555 threaddepth 20
   SMS-DAEMON-1

   sms_mway_2 port 55555 threaddepth 20
   SMS-DAEMON-2
   ```

   書き換えルールは次のようになります。

   ```
   sms-1.siroe.com $u%sms-1.siroe.com@SMS-DAEMON-1
   sms-2.siroe.com $U%sms-2.siroe.com@SMS-DAEMON-2
   ```

   その後、両方で同一の SMPP サーバーを使用できるようにするために、これら 2 つのチャネルそれぞれのオプションファイルで SMPP_SERVER=smpp.siroe.com と指定します。

配信再試行の間隔を調整する

一時的なエラーが原因で SMS メッセージが配信されない場合 (たとえば、SMPP サーバーがアクセス不能な場合)、電子メールメッセージは配信キューに残され、後で再試行が行われます。別の設定が行われていないかぎり、ジョブコントローラは 1 時間後まで配信を再試行しません。SMS メッセージの場合、これはあまりにも長い待機時間です。したがって、SMS チャネルに backoff チャネルキーワードを使用して、短い間隔で配信試行を指定することをお勧めします。たとえば、次のように指定します。

```
sms_mway port 55555 threaddepth 20 ¥
  backoff pt2m pt5m pt10m pt30m notices 1
smpp.siroe.com
```

上記の設定では、再配信試行は最初の試行の 2 分後に実行されます。これが失敗した場合、2 回目の試行の 5 分後に実行されます。その次は 10 分後に実行され、最終的には 30 分ごとに実行されます。notices 1 チャネルキーワードを使用すると、1 日経ってもメッセージが配信されない場合、そのメッセージは配信不能として戻されます。

片方向設定の例 (MobileWay)

MTA SMS チャネルは SMPP V3.4 と互換性のある SMPP サーバーで使用できます。この節では、設定例をわかりやすく示すために、MobileWay SMPP サーバー で使用する場合の SMS チャネルの設定方法を説明します。MobileWay (http://www.mobileway.com/) は、グローバルデータおよび SMS 接続の大手供給元です。MobileWay を介して SMS トラフィックをルーティングすることによって、世界中の主要な SMS ネットワーク 上の SMS 加入者にアクセスできます。

MobileWay で SMPP アカウントを取得する際、次の質問に答えるように求められます。

- SMPP クライアントの IP アドレス : インターネット上のほかのドメインから見えるとおりに Messaging Server システムの IP アドレスを入力します。
- デフォルトの有効期間 : これは、送信した SMS メッセージに有効期間が指定されていない場合に MobileWay で使用される SMS 有効期間です。この有効期間内に配信できない SMS メッセージは破棄されます。妥当な値を指定してください (2 日間、7 日間など)。
- ウィンドウサイズ : これは、追加の SMS メッセージを送信する前に、SMPP クライアントが停止して SMPP サーバーからの応答を待つまでに送信する最大 SMS メッセージ数です。値として 1 メッセージを指定する必要があります。
- タイムゾーン : Messaging Server システムが動作するタイムゾーンを指定します。タイムゾーンは、GMT からのオフセットで指定してください。
- タイムアウト : 片方向 SMS メッセージングの場合は無関係です。
- 外部バインド要求用の IP アドレスおよび TCP ポート : 片方向 SMS メッセージングの場合は無関係です。

MobileWay に上記の質問に対する答えを指定すると、SMPP アカウントと SMPP サーバーとの通信に必要な情報が提供されます。次の情報が含まれます。

```
Account Address: a.b.c.d:p
Account Login: system-id
Account Passwd: secret
```

Account Address フィールドは、IP アドレス `a.b.c.d` および接続先の MobileWay SMPP サーバーの TCP ポート番号 p です。`SMPP_SERVER` および `SMPP_PORT` の各チャネルオプションにこれらの値を使用します。Account Login および Account Passwd は、それぞれ、`ESME_SYSTEM_ID` および `ESME_PASSWORD` の各チャネルオプションに使用される値です。この情報を使用して、チャネルオプションファイルには次の内容を含めます。

```
SMPP_SERVER=a.b.c.d
SMPP_PORT=p
ESME_SYSTEM_ID=system-id
ESME_PASSWORD=secret
```

MobileWay と相互運用するには、さらに 2 つのオプションを設定する必要があります。

```
ESME_ADDRESS_TON=0x01
DEFAULT_DESTINATION_TON=0x01
```

imta.cnf ファイルで書き換えルールは次のように示されます。

```
sms.your-domain $u@sms.your-domain
```

imta.cnf ファイルでチャネル定義は次のように示されます。

```
sms_mobileway
sms.your-domain
```

チャネルオプションファイル、書き換えルール、およびチャネル定義の設定が完了すると、テストメッセージを送信できます。MobileWay では、次の形式の国際的なアドレス指定が必要です。

```
+<country-code><subscriber-number>
```

たとえば、テストメッセージを北アメリカの加入者 (加入者番号 (800) 555-1212) に送信するには、電子メールメッセージの宛先を次のように指定します。

```
+18005551212@sms.your-domain
```

デバッグ

チャネルをデバッグするには、チャネル定義で master_debug チャネルキーワードを指定します。たとえば、次のように指定します。

```
sms_mway port 55555 threaddepth 20 ¥
  backoff pt2m pt5m pt10m pt30m notices 1 master_debug
```

master_debug チャネルキーワードを指定すると、チャネルの動作についての基本的な診断情報がチャネルのログファイルに出力されます。チャネルによって実行された SMPP トランザクションの詳細な診断情報が必要な場合は、さらに次のことを指定します。

```
DEBUG=-1
```

この指定はチャネルのオプションファイルに行います。

双方向 SMS 用に SMS チャネルを設定する

SMS チャネルの設定についての一般的な説明は、前出の 923 ページの「SMS チャネルの設定」以降の項を参照してください。表 D-18 に示されている例外を除いて、リモート SMSC と直接通信している場合と同じように SMS チャネルを設定します。

表 D-18　双方向設定での例外

例外	説明
master チャネルキーワード	master チャネルキーワードが指定されている場合は、削除します。 このチャネルキーワードは SMS チャネル設定には不要です。
SMPP_SERVER	SMS Gateway Server のホスト名または IP アドレスをポイントするように設定します。SMPP リレーの LISTEN_INTERFACE_ADDRESS オプション (967 ページの「設定オプション」を参照) を使用している場合は、指定されているネットワークインタフェースアドレスに関連付けられているホスト名または IP アドレスを必ず使用します。
SMPP_PORT	SMPP リレーのインスタンス化に使用される LISTEN_PORT の設定で使用されているものと同じ TCP ポート (964 ページの「SMPP リレー」を参照)。
DEFAULT_SOURCE_ADDRESS	値を選んでから、リモート SMSC がこのアドレスを Gateway SMPP サーバーに戻すように設定します。SMS チャネルのオプションファイルで、選択した値をこのオプションに指定します。
GATEWAY_PROFILE	ゲートウェイプロファイル名と一致するように設定します。963 ページの「ゲートウェイプロファイル」を参照してください。
USE_HEADER_FROM	0 に設定します。

上記以外のすべてのチャネル設定は、SMS チャネルマニュアルで説明されているように設定する必要があります。

960 ページの「双方向 SMS ルーティングを設定する」で説明されているように、リモート SMSC は、LISTEN_PORT オプションで指定されている TCP ポート番号を使用して、DEFAULT_SOURCE_ADDRESS チャネルオプションで定義されている SMS アドレスを Gateway の SMPP サーバーにルーティングするように設定されている必要があります (LISTEN_PORT の設定方法については、964 ページの「SMPP サーバー」を参照)。

複数の SMS チャネルが同一の SMPP リレーを使用することもできます。同様に、複数の SMS チャネルに対する SMS 返信および通知を処理するには、SMPP サーバーまたはゲートウェイプロファイルが 1 つだけ必要です。複数のリレー、サーバー、およびゲートウェイプロファイルが設定可能であることには、設定オプションを介してさまざまな使用上の特徴を有効にすることができるという意義があります。

SMS Gateway Server の動作方式

SMS Gateway Server は、モバイルで作成された SMS メッセージを正しい電子メールアドレスに一致させるメカニズムを提供することで、双方向 SMS をサポートします。この節には、SMS Gateway Server に関する以下の項目があります。

- 955 ページの「SMS Gateway Server の機能」
- 956 ページの「SMPP リレーおよびサーバーの動作」
- 958 ページの「SMS の返信および通知の処理」

SMS Gateway Server の機能

SMS Gateway Server は、同時に SMPP リレーとサーバーの両方として機能します。SMS Gateway Server は、各機能の複数の「インスタンス」を持つように設定できます。たとえば、3 つの SMPP リレーを持ち、それぞれが異なる TCP ポートまたはネットワークインタフェースを待機し、異なる SMPP サーバーにリレーを行うように設定できます。同様に、4 つの SMPP サーバーを持ち、それぞれが異なる TCP ポートとネットワークインタフェースの組み合わせを待機するように設定できます。

SMS Gateway Server は、SMS メッセージを電子メールに送信するためのゲートウェイプロファイルで設定します (ゲートウェイプロファイルはない場合もある)。各ゲートウェイプロファイルには、プロファイルと一致する宛先 SMS アドレス、SMS メッセージから宛先電子メールアドレスを抽出する方法、および SMS から電子メールへの変換プロセスのさまざまな特徴を記述します。SMPP リレーまたはサーバーを介して SMS Gateway Server に提示された各 SMS メッセージは、各プロファイルと照合されます。一致するものが見つかると、メッセージは電子メールにルーティングされます。

ゲートウェイプロファイルには、電子メールからモバイルに送信されたメッセージに応答してリモート SMSC が返した通知メッセージの処理方法も定義されています。

SMPP リレーおよびサーバーの動作

SMS Gateway Server は SMPP リレーとして機能しているとき、可能なかぎり透過的に動作します。ローカル SMPP クライアントからのすべての要求をリモート SMPP サーバーにリレーし、リモートサーバーの応答をリレーして返します。次の 2 つの例外があります。

- ローカル SMPP クライアントから、設定済みのゲートウェイプロファイルと一致する SMS 宛先アドレスを持つメッセージが送信された場合、その SMS メッセージは電子メールに直接送り返されます。つまり、SMS メッセージはリモート SMPP サーバーにリレーされません。
- ローカルまたはリモート SMPP クライアントから、過去の SMPP リレーで生成された一意の SMS ソースアドレスと一致する SMS 宛先アドレスを持つメッセージが送信された場合、SMS メッセージは過去にリレーされたメッセージへの返信となります。この返信は、元のメッセージの差出人に送信されます。

一般的に、SMS Gateway Server は、生成した一意の SMS ソースアドレスがゲートウェイプロファイルのどれかに一致するように設定されることに注意してください。

注	SMS Gateway Server の SMPP リレーは、正規の Sun Java System SMPP クライアント、つまり Sun Java System Messaging Server の SMS チャネルとともに使用する場合のみを対象にしています。これ以外の SMPP クライアントとともに使用する場合は対象にしていません。

SMS Gateway Server が SMPP サーバーとして機能する場合、以下の 3 つの状況で SMS メッセージは電子メールに送信されます。

- SMS メッセージはモバイルで作成されたものであり、ゲートウェイプロファイルと一致する。
- SMS メッセージはモバイルで作成されたものであり、SMS 宛先アドレスが過去に生成された一意の SMS ソースアドレスと一致する。
- SMS メッセージは、過去に SMS Gateway Server の SMPP リレーによってリレーされた電子メールからモバイルへのメッセージに対応する SMS 通知である。

上記以外のすべての SMS メッセージは SMPP サーバーによって拒否されます。

リモート SMPP から ゲートウェイ SMPP への通信

リモート SMPP クライアントは、プロトコルデータユニット (PDU) を使用してゲートウェイ SMPP サーバーに通信します。リモート SMPP クライアントは、要求 PDU を出します。ゲートウェイ SMPP サーバーはこの PDU に対して応答します。ゲートウェイ SMPP サーバーは同期的に動作します。ゲートウェイ SMPP サーバーは、要求 PDU への応答を完了してから、接続している SMPP クライアントからの次の要求 PDU を処理します。

表 D-19 に、ゲートウェイ SMPP サーバーが処理する要求 PDU およびゲートウェイ SMPP サーバーの応答を示します。

表 D-19　SMPP サーバーのプロトコルデータユニット

要求 PDU	SMPP サーバーの応答
BIND_TRANSMITTER BIND_TRANSCEIVER UNBIND	適切な応答 PDU とともに応答します。認証資格は無視されます。
OUTBIND	ゲートウェイ SMPP サーバーは BIND_RECEIVER PDU を返します。提示された認証資格は無視されます。
SUBMIT_SM DATA_SM	宛先 SMS アドレスと一意の SMS ソースアドレスまたはゲートウェイプロファイルの SELECT_RE 設定の照合を試行します。どれも一致しない場合は、PDU は拒否され、ESME_RINVDSTADR エラーが発生します。
DELIVER_SM	宛先 SMS アドレスまたは履歴レコードにある受信確認済みメッセージ ID のどちらかの検出を試行します。どちらも一致しない場合は、ESME_RINVMSGID エラーを返します。
BIND_RECEIVER	サポートされていません。ESME_RINVCMDID エラーとともに GENERIC_NAK PDU を返します。
SUBMIT_MULTI	サポートされていません。ESME_RINVCMDID エラーとともに GENERIC_NAK PDU を返します。
REPLACE_SM	サポートされていません。ESME_RINVCMDID エラーとともに GENERIC_NAK PDU を返します。
CANCEL_SM	サポートされていません。ESME_RINVCMDID エラーとともに GENERIC_NAK PDU を返します。
QUERY_SM	サポートされていません。ESME_RINVCMDID エラーとともに GENERIC_NAK PDU を返します。
QUERY_LAST_MSGS	サポートされていません。ESME_RINVCMDID エラーとともに GENERIC_NAK PDU を返します。

表 D-19　SMPP サーバーのプロトコルデータユニット (続き)

要求 PDU	SMPP サーバーの応答
QUERY_MSG_DETAILS	サポートされていません。ESME_RINVCMDID エラーとともに GENERIC_NAK PDU を返します。
ENQUIRE_LINK	ENQUIRE_LINK_RESP PDU を返します。
ALERT_NOTIFICATION	受け入れられますが、無視されます。

SMS の返信および通知の処理

SMS Gateway Server は、SMPP リレーで中継した各 SMS メッセージの履歴レコードを管理しています。履歴データを使用する必要性は、電子メールメッセージを SMS に送信する場合、メッセージの差出人の電子メールアドレスを SMS ソースアドレスに変換することは一般的に不可能であることから生じています。SMS の返信および通知はすべて SMS ソースアドレスに送信されるため、問題が発生します。この問題は、自動的に生成された一意の SMS ソースアドレスをリレーされるメッセージで使用することで解消されます。それによって、リモート SMSC は SMS ソースアドレスをゲートウェイ SMPP サーバーに返すように設定されます。

履歴データは、メッセージ ID および生成された一意の SMS ソースアドレスのメモリ内のハッシュテーブルとして表されます。このデータは、関連する電子メールの発信データとともにディスクに保存されます。ディスクベースのストレージは一連のファイルです。各ファイルはトランザクションの HASH_FILE_ROLLOVER_PERIOD 秒 (デフォルトは 30 分) に相当します。各ファイルは、RECORD_LIFETIME 秒間 (デフォルトは 3 日間) 保持されます。履歴データのメモリ内とディスク上のリソース要件については、『Sun Java System Communications Services 配備計画ガイド』(http://docs.sun.com/doc/819-1069?l=ja) を参照してください。

各レコードは、次の 3 つのコンポーネントから成ります。

- 電子メールの発信データ (エンベロープ From: および To: アドレスなど)。このデータは、MTA SMS チャネルがメッセージを送信する際に提供されます。
- SMPP リレーによって生成され、リレー対象の SMS メッセージに挿入された一意の SMS ソースアドレス。
- リモート SMSC の SMPP サーバーが送信を受け入れるときに返す受信確認済みメッセージ ID。

SMS 返信のルーティングプロセス

Gateway SMPP リレーおよびサーバーは、履歴レコードを使用して SMS 返信、SMS 通知、およびモバイルで作成されたメッセージを処理します。SMS メッセージが SMPP リレーまたはサーバーに提示されると、以下のルーティングプロセスが実行されます。

1. SMS 宛先アドレスが履歴レコードに照合され、過去に SMPP リレーが生成した一意の SMS ソースアドレスに一致するものがあるかどうか調べられます。一致するものが見つかった場合については、手順 6 に進みます。
2. 一致するものはないが、メッセージが SMS 通知 (SMPP `DELIVER_SM` PDU) である場合、受信確認済みメッセージ ID (存在する場合) が履歴レコードと照合されます。一致するものが見つかった場合は、手順 8 へ進みます。[SMS Gateway Server では、実際にメッセージを SMPP リレーまたは SMPP サーバーのどちらかに提示することができます。]
3. 一致するものがない場合、宛先 SMS アドレスは各設定済みゲートウェイプロファイルの `SELECT_RE` オプション表現と照合されます。一致するものが見つかった場合は、手順 9 へ進みます。
4. 一致するものがなく、SMS メッセージがゲートウェイ SMPP リレーに提示された場合、メッセージはリモート SMPP サーバーにリレーされます。
5. 一致するものがなく、SMS メッセージがゲートウェイ SMPP サーバーに提示された場合、メッセージは無効なメッセージであると判断され、SMPP 応答 PDU 内にエラー応答が返されます。電子メールから SMS の場合、最終的に配信不能通知 (NDN) が生成されます。
6. 一致する SMS ソースアドレスが見つかった場合は、SMS メッセージは返信であるか通知メッセージであるかどうかについてさらに調べられます。通知メッセージであるためには、受信確認済みメッセージ ID を持つ `SUBMIT_SM` PDU である必要があります。それ以外の場合は、返信であると見なされます。
7. 返信である場合は、履歴レコードにある元の電子メール情報を使用して、SMS メッセージは電子メールメッセージに変換されます。
8. 通知である場合は、RFC 1892 ～ 1894 に従って、SMS メッセージは電子メール配信ステータス通知 (DSN) に変換されます。元の電子メールメッセージの ESMTP `NOTIFY` フラグ (RFC 1891) が使用されることに注意してください (たとえば、SMS メッセージは「成功」DSN であるが、元の電子メールメッセージは「失敗」通知のみを要求していた場合、この SMS 通知は破棄される)。
9. 宛先 SMS アドレスが設定済みゲートウェイプロファイルの `SELECT_RE` オプションに一致した場合、SMS メッセージはモバイルで作成されたメッセージとして扱われ、そのゲートウェイプロファイルの `PARSE_RE_n` ルールに従って電子メールメッセージに変換し直されます。変換に失敗した場合、SMS メッセージは無効になり、エラー応答が返されます。

SMS Gateway Server の設定

この節では、電子メールからモバイルおよびモバイルから電子メールの両方の機能を使用する場合の SMS Gateway Server の設定方法について説明します。この節には、以下の項目があります。

双方向 SMS ルーティングを設定する

MTA と SMSC 間の双方向の電子メールおよび SMS ルーティングを設定する場合に推奨される方法は、次の 3 手順のプロセスです。

- SMS アドレスプレフィックスを設定する - SMS アドレスプレフィックスを選択します。10 文字以内の任意のプレフィックスが使用できます。
- ゲートウェイプロファイルを設定する - SMS Gateway Server に使用するためにそのプレフィックスを確保します (ゲートウェイプロファイルを設定する)。
- SMSC を設定する - そのプレフィックスで始まる SMS ゲートウェイ SMPP サーバーに SMS 宛先アドレスをルーティングするように、SMSC を設定します。モバイルで作成された電子メールには、プレフィックスのみがあります。返信および通知には、プレフィックスに続いて 10 桁の 10 進法の数字があります。

SMS アドレスプレフィックスを設定する

MTA SMS チャネルによって生成されたソース SMS アドレスは、選択した SMS アドレスプレフィックスに一致するように設定する必要があります。以下の設定を行います。

- MTA SMS チャネルオプションを次のように設定します。

  `USE_HEADER_FROM=0`

  `DEFAULT_SOURCE_ADDRESS=`*`prefix`*

最初の設定によって、チャネルは、電子メールメッセージにある情報から SMS ソースアドレスを設定しなくなります。2 番目の設定によって、ほかのソースから設定されていない場合、SMS ソースアドレスは選択したプレフィックスに設定されます。

- 受け入れて電子メールにルーティングする SMS 宛先アドレスとして、プレフィックスを認識します。これを行うには、SELECT_RE ゲートウェイプロファイルオプションを次のように指定します。

  SELECT_RE=*prefix*

ゲートウェイプロファイルを設定する

次に、SMS Gateway Server のゲートウェイプロファイルを設定して、リレー対象のすべての SMS ソースアドレスを一意にする必要があります。これはデフォルト設定ですが、ゲートウェイプロファイルオプション MAKE_SOURCE_ADDRESSES_UNIQUE=1 を指定することによって明示的に設定することもできます。この設定によって、リレー対象の SMS ソースアドレスは次の形式になります。

prefixnnnnnnnnnn

nnnnnnnnnn は、一意の 10 桁の 10 進数です。

SMSC を設定する

最後に、SMSC を設定して、プレフィックス (プレフィックスのみ、またはプレフィックスと 10 桁の 10 進数のどちらか) と一致するすべての SMS 宛先アドレスを SMS Gateway Server の SMPP サーバーにルーティングする必要があります。このようなルーティングの正規表現は、次のようになります。

prefix([0-9]{10,10}){0,1}

prefix は DEFAULT_SOURCE_ADDRESS の値です。[0-9] は 10 桁の 10 進数として許容される値を示し、{10, 10} は最小値が 10 桁あり、最大値が 10 桁あるということを示します。{0, 1} は、ゼロまたは 10 桁の数字のどれかが可能であることを示します。

SMS Gateway Server の有効化と無効化

- SMS Gateway Server を有効にするには、configutil パラメータ local.smsgateway.enable の値を 1 に設定する必要があります。これを設定するには、次の設定ユーティリティコマンドを使用します。

  ```
  # configutil -o local.smsgateway.enable -v 1
  ```

- SMS Gatewey Server を無効にするには、local.smsgateway.enable の値を 0 に設定します。これには次のコマンドを使用します。

  ```
  # configutil -o local.smsgateway.enable -v 0
  ```

SMS Gateway Server の起動と停止

SMS Gateway Server が有効になった後は、次のコマンドを使用して起動および停止することができます。

```
# start-msg sms
```

および

```
# stop-msg sms
```

SMS Gateway Server の設定ファイル

SMS Gateway Server が機能するためには、設定ファイルが必要です。設定ファイルは、UTF-8 を使用してエンコードされた Unicode テキストファイルです。設定ファイルは、ASCII テキストファイルの場合もあります。ファイル名は次のようにする必要があります。

installation-directory/config/sms_gateway.cnf

ファイル内の各オプション設定は、次の形式です。

```
option-name=option-value
```

オプショングループに属しているオプションは、次の形式で表示されます。

```
[group-type=group-name]
option-name-1=option-value-1
option-name-2=option-value-2
...
option-name-n=option-value-n
```

Gateway Server 上に電子メールからモバイルの処理を設定する

双方向 SMS の電子メールからモバイルの部分を実装するには、次の設定を行う必要があります。

- 963 ページの「ゲートウェイプロファイル」
- 964 ページの「SMPP リレー」
- 964 ページの「SMPP サーバー」

ゲートウェイプロファイル

電子メールからモバイルへのゲートウェイプロファイルを設定するには、次の手順に従います。

1. SMS Gateway Server 設定ファイルにゲートウェイプロファイルを追加します。

   オプショングループを追加するには、次の形式を使用します。

   ```
   [GATEWAY_PROFILE=profile_name]
   option-name-1=option-value-1
   option-name-2=option-value-2a
   ...
   option-name-n=option-value-n
   ```

   上記の形式のゲートウェイプロファイル名 `profile_name` の長さは、11 バイトを超えないようにしてください。この名前は、SMS チャネルオプションファイルの `GATEWAY_PROFILE` チャネルオプションの名前と同じである必要があります。名前は、大文字と小文字が区別されます。有効なチャネルオプションの一覧は、927 ページの「使用可能なオプション」を参照してください。

2. ゲートウェイプロファイルオプション (例 : `SMSC_DEFAULT_CHARSET`) を、リモート SMSC の特徴と一致するように設定します。

3. SMS チャネルの電子メールの特徴と一致するように、ほかのゲートウェイプロファイルオプションを設定します。

   ゲートウェイプロファイルオプションの詳細については、976 ページの「ゲートウェイプロファイルのオプション」を参照してください。

4. `CHANNEL` オプションを設定します。

   値を MTA SMS チャネルの名前に設定します。

   ゲートウェイを介して通知が電子メールに送信される場合、結果の電子メールメッセージはこのチャネル名を使用して MTA のキューに入れられます。

SMPP リレー

SMPP リレーを設定するには、次の手順を実行します。

1. SMPP リレーインスタンス (オプショングループ) を SMS Gateway Server の設定ファイルに追加します。

   オプショングループを追加するには、次の形式を使用します。

   ```
   [SMPP_RELAY=relay_name]
   option-name-1=option-value-1
   option-name-2=option-value-2
   ...
   option-name-n=option-value-n
   ```

   リレー名には任意の名前を使用できます。その名前が同一の設定ファイル内のほかの SMPP リレーインスタンスに使用されていないことにのみ注意してください。

2. `LISTEN_PORT` オプションを設定します。

   SMS チャネルの `SMPP_PORT` オプションに使用される値は、リレーの `LISTEN_PORT` オプションで使用される値と一致する必要があります。`LISTEN_PORT` の TCP ポート番号を選択します。この TCP ポート番号には、ほかの SMPP リレーまたはサーバーインスタンスで使用されていないもの、同一のコンピュータ上で実行されているほかのサーバーで使用されていないものを選択します。

3. `SERVER_HOST` オプションを設定します。

   リレーの `SERVER_HOST` オプションは、リモート SMSC の SMPP サーバーのホスト名を示す必要があります。ホスト名の代わりに IP アドレスを使用することもできます。

4. `SERVER_PORT` オプションを設定します。

   リレーの `SERVER_PORT` オプションは、リモート SMSC の SMPP サーバーの TCP ポートを示す必要があります。

SMPP リレーオプションの詳細については、971 ページの「SMPP リレーオプション」を参照してください。

SMPP サーバー

SMPP サーバーを設定するには、次の手順を実行します。

1. SMPP サーバーインスタンス (オプショングループ) を SMS Gateway Server の設定ファイルに追加します。

   オプショングループを追加するには、次の形式を使用します。

   ```
   [SMPP_SERVER=server_name]
   option-name-1=option-value-1
   option-name-2=option-value-2
   ...
   option-name-n=option-value-n
   ```

   サーバー名には任意の名前を使用できます。その名前が同一の設定ファイル内のほかの SMPP サーバーインスタンスに使用されていないことにのみ注意してください。

2. `LISTEN_PORT` オプションを設定します。

   ほかのサーバーまたはリレーインスタンスに使用されていない TCP ポート番号を選択します。また、ポート番号は、同一コンピュータ上のほかのサーバーで使用されていないものにしてください。

   この TCP ポートを使用して SMPP を介して通知を SMS Gateway Server システムにルーティングするように、リモート SMSC を設定する必要があります。

SMPP サーバーオプションの詳細については、974 ページの「SMPP サーバーオプション」を参照してください。

モバイルから電子メールの処理を設定する

モバイルから電子メールの処理を設定するには、2 つの設定手順を実行する必要があります。

- 965 ページの「モバイルから電子メールへのゲートウェイプロファイルを設定する」
- 966 ページの「モバイルから電子メールの SMPP サーバーを設定する」

複数のゲートウェイプロファイルは同一の SMPP サーバーインスタンスを使用することもできます。実際、SMPP サーバーインスタンスは、電子メールからモバイル、モバイルから電子メールの両方の用途に使用される場合があります。

モバイルから電子メールへのゲートウェイプロファイルを設定する

モバイルが起点である場合、ゲートウェイプロファイルは 2 つの重要な情報を提供します。そのゲートウェイプロファイルを使用する SMS メッセージを特定する方法とその SMS メッセージを電子メールメッセージに変換する方法です。このプロファイルは、電子メールからモバイルの場合と同じものが使用できます。ただし、`SELECT_RE` オプションを追加する必要があります。

ゲートウェイプロファイルを設定するには、次の手順に従います。

1. SMS Gateway Server の設定ファイルにゲートウェイプロファイル (オプショングループ) を追加します。

   オプショングループを追加するには、次の形式を使用します。

   ```
   [GATEWAY_PROFILE=profile_name]
   option-name-1=option-value-1
   option-name-2=option-value-2
   ...
   option-name-n=option-value-n
   ```

   プロファイル名には、11 文字以内の任意の名前を使用できます。同一の設定ファイル内のほかのゲートウェイプロファイルで使用されていない名前にしてください。

2. 各ゲートウェイプロファイルに、SELECT_RE オプションを指定する必要があります。

   このオプションの値には、SMS 宛先アドレスと照合できるように ASCII 正規表現を使用します。SMS 宛先アドレスが正規表現と一致した場合は、SMS メッセージはゲートウェイを介して電子メールに送信されます。このとき、一致したプロファイルで示されている特徴が使用されます。

   重複する一連の SMS アドレスを持つ複数のゲートウェイプロファイルを設定することは可能です (たとえば、アドレス 000 と一致するプロファイルとほかの任意の 3 桁のアドレスと一致する別のプロファイル)。ただし、これは避けてください。SMS メッセージが 1 つのゲートウェイプロファイル (一致する最初のプロファイル) のみに渡されることになるからです。また、SMS アドレスが照合される順序が不定になるからです。

3. CHANNEL オプションを設定します。

   この値は MTA SMS チャネルの名前にする必要があります。

モバイルを起点とする場合のオプションの詳細については、976 ページの「ゲートウェイプロファイルのオプション」を参照してください。

モバイルから電子メールの SMPP サーバーを設定する

SMPP サーバーの追加方法は、電子メールからモバイルの SMPP サーバーの場合と同じです (964 ページの「SMPP サーバー」を参照)。

SMS 通信をゲートウェイ SMPP サーバーにルーティングするように、リモート SMSC を設定する必要があります。そのためには、SMSC がモバイルから電子メールへの通信をルーティングするために使用する SMS 宛先アドレスが、ゲートウェイプロファイルオプション SELECT_RE に設定された値である必要があります。

たとえば、モバイルから電子メールの通信に SMS アドレス 000 を使用する場合、SMS 宛先アドレス 000 の通信をゲートウェイ SMPP サーバーにルーティングするように SMSC を設定する必要があります。ゲートウェイプロファイルは SELECT_RE=000 のオプション設定を使用する必要があります。

設定オプション

この節では、SMS Gateway Server 設定ファイルのオプションについて詳しく説明します。次の表に、すべての使用可能な設定オプションの一覧を簡単な説明とともに示します。グローバルオプション、SMPP リレーオプション、SMPP サーバーオプション、および SMS Gateway Server プロファイルオプションそれぞれについての表があります。

各表に続く項目では、すべての使用可能な設定オプションについて詳しく説明します。以下の項目があります。

- 967 ページの「グローバルオプション」

  グローバルオプションは、設定ファイルの最上部 (どのオプショングループよりも前) に配置する必要があります。これ以外のオプションは、オプショングループ内に配置してください。
- 971 ページの「SMPP リレーオプション」
- 974 ページの「SMPP サーバーオプション」
- 976 ページの「ゲートウェイプロファイルのオプション」

グローバルオプション

現在のところ、SMS Gateway Server には、次の 3 つのカテゴリのグローバルオプションがあります。

- スレッドチューニングオプション
- 履歴データの調整
- その他

グローバルオプションは、設定ファイルの最上部、オプショングループよりも先に指定する必要があります。表 D-20 に、グローバル設定オプションをすべて示します。

表 D-20　グローバルオプション

オプション	デフォルト	説明
DEBUG	6	生成される診断出力のタイプを選択します

表 D-20　グローバルオプション (続き)

オプション	デフォルト	説明
HISTORY_FILE_DIRECTORY		履歴データのファイルの絶対ディレクトリパス
HISTORY_FILE_MODE	0770	履歴データのファイルへの許可
HISTORY_FILE_ROLLOVER_PERIOD	30 分	1 つの履歴データのファイルに書き込むための最大時間
LISTEN_CONNECTION_MAX		すべての SMPP リレーおよびサーバーインスタンスでの最大同時受信接続数
RECORD_LIFETIME	3 日	履歴データアーカイブのレコードの存続期間
THREAD_COUNT_INITIAL	10 スレッド	最初のワーカースレッド数
THREAD_COUNT_MAXIMUM	50 スレッド	最大ワーカースレッド数
THREAD_STACK_SIZE	64K バイト	各ワーカースレッドのスタックサイズ

スレッドチューニングオプション

各受信 TCP 接続はそれぞれが 1 つの SMPP セッションです。セッションの処理は、スレッドプールのワーカースレッドによって行われます。セッションの処理を I/O 要求が完了するまで待つ必要がある場合は、ワーカースレッドはそのセッションを保留し、ほかの処理を実行します。I/O 要求が完了すると、プール内の使用されていないワーカースレッドによってセッションが再開されます。

以下のオプションを使用して、ワーカースレッドのプールの処理を調整できます。THREAD_COUNT_INITIAL、THREAD_COUNT_MAXIMUM、THREAD_STACK_SIZE。

THREAD_COUNT_INITIAL

(整数、> 0) ワーカースレッドのプールに最初に作成するスレッド数。この数には、メモリ内の履歴データ専用に使用されるスレッド (2 スレッド) を含みません。また、着信 TCP 接続の待機専用に使用されるスレッド (SMS Gateway Server が待機する TCP ポートおよびインタフェースアドレスペアにつき 1 スレッド) も含みません。THREAD_COUNT_INITIAL のデフォルト値は 10 スレッドです。

THREAD_COUNT_MAXIMUM

(整数、>= *THREAD_COUNT_INITIAL*) ワーカースレッドのプールに許可する最大スレッド数。デフォルト値は 50 スレッドです。

THREAD_STACK_SIZE

(整数、>0) ワーカースレッドのプール内の各ワーカースレッドのスタックサイズ (単位 : バイト)。デフォルト値は 65,536 バイト (64K バイト) です。

履歴データの調整

SMS メッセージのリレー時、受信側のリモート SMPP サーバーによって生成されるメッセージ ID は、メモリ内のハッシュテーブルに保存されます。このメッセージ ID とともに、元の電子メールメッセージについての情報も保存されます。その後メッセージ ID が SMS 通知によって参照された場合、この情報が取り出されることがあります。取り出された情報は、SMS 通知を適切な電子メール受取人に送信するために使用されます。

メモリ内のハッシュテーブルは、専用のスレッドでディスクに返されます。その結果ディスクファイルは「履歴ファイル」として参照されます。履歴ファイルは、次の 2 つの目的で使用されます。SMS Gateway Server を再起動した後にメモリ内ハッシュテーブルを復元するのに必要なデータを不揮発性の形式で保存するため、また、非常に長くなる可能性のあるデータをディスクに保存することによって、仮想メモリを節約するためです。各履歴ファイルは、HASH_FILE_ROLLOVER_PERIOD 秒間のみ書き込まれます。この時間を過ぎると、ファイルは終了し、新しい履歴ファイルが作成されます。履歴ファイルの存続期間が RECORD_LIFETIME 秒を超えると、ファイルはディスクから削除されます。

履歴ファイルの調整には次のオプションが使用できます。HISTORY_FILE_DIRECTORY、HISTORY_FILE_MODE、HISTORY_FILE_ROLLOVER_PERIOD、RECORD_LIFETIME。

HISTORY_FILE_DIRECTORY

(文字列、絶対ディレクトリパス) 履歴ファイルの書き込み先のディレクトリへの絶対パス。ディレクトリパスが存在しない場合は作成されます。このオプションのデフォルト値は、次のとおりです。

msg_svr_base/data/sms_gateway_cache/

使用するディレクトリは、相応に高速なディスクシステム上に存在し、予測される保存量よりも大きい空き容量がある必要があります。ストレージ計画の情報は、983 ページの「SMS Gateway Server のストレージ要件」を参照してください。このオプションは、サイトで適切な値に変更することをお勧めします。

HISTORY_FILE_MODE

(整数、8 進値) 履歴ファイルに関連付けるファイル許可。デフォルトでは、0770 (8 進値) の値が使用されています。

HISTORY_FILE_ROLLOVER_PERIOD

(整数、秒) 現在の履歴ファイルが終了し、新しいものが HASH_FILE_ROLLOVER_PERIOD 秒ごとに作成されます。デフォルトでは、1800 秒 (30 分) の値が使用されています。

RECORD_LIFETIME

(整数、秒 > 0) 履歴レコードの存続期間 (単位 : 秒)。この存続期間を過ぎたレコードは、メモリからパージされます。この存続期間を過ぎた履歴ファイルは、ディスクから削除されます。デフォルトでは、259,200 秒 (3 日) の値が使用されています。メモリに保存されているレコードは、メモリ内データの管理専用のスレッドによって一斉にパージされます。このパージは、HASH_FILE_ROLLOVER_PERIOD 秒ごとに実行されます。ディスク上のファイルは、新しい履歴ファイルを開く必要が生じたときにパージされます。

その他

その他のオプションには、次の 2 つがあります。DEBUG および LISTEN_CONNECTION_MAX です。

DEBUG

(整数、ビットマスク) デバッグ出力を有効にします。デフォルト値は 6 であり、警告およびエラーメッセージが選択されます。

表 D-21 に、DEBUG ビットマスクのビット値を示します。

表 D-21　DEBUG ビットマスク

ビット	値	説明
0-31	-1	きわめて詳細な出力
0	1	情報メッセージ
1	2	警告メッセージ
3	4	エラーメッセージ
3	8	サブルーチン呼び出しのトレース
4	16	ハッシュテーブル診断
5	32	I/O 診断、受信
6	64	I/O 診断、送信

表 D-21　DEBUG ビットマスク (続き)

ビット	値	説明
7	128	SMS から電子メールへの変換診断 (モバイルからの発信および SMS 通知)
8	256	PDU 診断、ヘッダーデータ
9	512	PDU 診断、本文データ
10	1024	PDU 診断、タイプの長さ値のデータ
11	2048	オプション処理。すべてのオプション設定をログファイルに送ります。

LISTEN_CONNECTION_MAX

(整数、>= 0) すべての SMPP リレーおよびサーバーインスタンス全体に許可される最大同時受信 TCP 接続数。0 (ゼロ) の値は、接続数に関するグローバル制限はないことを示します。ただし、リレーまたはサーバー単位では、特定のリレーまたはサーバーインスタンスによって指定される制限がある場合があります。

SMPP リレーオプション

SMS Gateway Server には、異なる特徴を持つ複数の SMPP リレーインスタンスを設定することができます。最も重要なインスタンスは、待機対象の TCP ポートとインタフェースです。言い換えると、SMPP リレーが待機するネットワークインタフェースと TCP ポートの各ペアに、別個の特徴を設定することができます。このような特徴は、この節で説明するオプションを使用して指定します。

各インスタンスは次の形式のオプショングループ内に配置する必要があります。

```
[SMPP_RELAY=relay-name]
option-name-1=option-value-1
option-name-2=option-value-2
...
option-name-n=option-value-n
```

文字列 `relay-name` は、このインスタンスをほかのインスタンスから区別するためだけに使用されます。

表 D-22 に、SMPP リレーの設定オプションの一覧を示します。

表 D-22　SMPP リレーオプション

オプション	デフォルト	説明
LISTEN_BACKLOG	255	受信 SMPP クライアント接続の接続バックログ
LISTEN_CONNECTION_MAX		最大同時受信接続数
LISTEN_INTERFACE_ADDRESS		受信 SMPP クライアント接続のネットワークインタフェース
LISTEN_PORT		受信 SMPP クライアント接続の TCP ポート
LISTEN_RECEIVE_TIMEOUT	600 秒	SMPP クライアントからの受信接続の読み取りタイムアウト
LISTEN_TRANSMIT_TIMEOUT	120 秒	SMPP クライアントからの受信接続の書き込みタイムアウト
MAKE_SOURCE_ADDRESSES_UNIQUE	1	リレー対象の SMS ソースアドレスを一意にして、返信可能にします
SERVER_HOST		リレー先の SMPP サーバーのホスト名または IP アドレス
SERVER_PORT		リレー先の SMPP サーバーの TCP ポート
SERVER_RECEIVE_TIMEOUT	600 秒	送信 SMPP サーバー接続の読み取りタイムアウト
SERVER_TRANSMIT_TIMEOUT	120 秒	送信 SMPP サーバー接続の書き込みタイムアウト

LISTEN_BACKLOG

(整数、0 ～ 255 でその両端も含む) 受信 SMPP クライアント接続の TCP スタックによって許容される接続バックログ。デフォルトは 255 です。

LISTEN_CONNECTION_MAX

(整数、>= 0) この SMPP リレーインスタンスで許可される最大同時受信 TCP 接続数。この値は、グローバル設定の LISTEN_CONNECTION_MAX の値を超えた場合は無視されます。

LISTEN_INTERFACE_ADDRESS

(文字列、「*INADDR_ANY*」または ドット付き 10 進表記の IP アドレス) 受信 SMPP クライアント接続で待機対象のネットワークインタフェースの IP アドレス。文字列「INADDR_ANY」(すべての使用可能なインタフェース) またはドット付き 10 進表記の IP アドレスのどちらかです (例 : 193.168.100.1)。デフォルト値は「INADDR_ANY」です。この値を HA 論理 IP アドレスと対応させるには、クラスタ化された HA 設定が必要です。

LISTEN_PORT

(整数、TCP ポート番号) 受信 SMPP クライアント接続を受け入れるためのバインド先 TCP ポート。このオプションの指定は必須です。このオプションにはデフォルト値はありません。このサービスには、Internet Assigned Numbers Authority (IANA) からの割り当てはないことにも注意してください。

LISTEN_RECEIVE_TIMEOUT

(整数、秒 > 0) SMPP クライアントからデータを読み取るために待つ場合のタイムアウト。デフォルト値は 600 秒 (10 分) です。

LISTEN_TRANSMIT_TIMEOUT

(整数、秒 > 0) SMPP クライアントにデータを送信する場合のタイムアウト。デフォルト値は 120 秒 (2 分) です。

MAKE_SOURCE_ADDRESSES_UNIQUE

(0 または 1) デフォルトでは、SMPP リレーは各 SMS ソースアドレスに一意の 10 桁の文字列を付加します。結果の SMS ソースアドレスは、ほかの履歴データとともに保存されます。その結果、SMS ユーザーが返信することのできる一意の SMS アドレスになります。このアドレスが SMS 宛先アドレスとして使用されたとき、SMPP サーバーはこのアドレスを検出し、SMS メッセージを正しい電子メール差出人に送信します。

一意の SMS ソースアドレスの生成を無効にするには (片方向 SMS の場合)、このオプションの値に 0 を指定します。

SERVER_HOST

(文字列、TCP ホスト名またはドット付き 10 進表記の IP アドレス) SMPP クライアント通信のリレー先 SMPP サーバー。ホスト名または IP アドレスのどちらかを指定します。このオプションの指定は必須です。このオプションにはデフォルト値はありません。

SERVER_PORT

(整数、TCP ポート番号) リモート SMPP サーバーがリレーする TCP ポート。このオプションの指定は必須です。このオプションにはデフォルト値はありません。このサービスには IANA からの割り当てはありません。IANA からの SNPP の割り当てと混同しないでください。

SERVER_RECEIVE_TIMEOUT

(整数、秒 > 0) SMPP サーバーからデータを読み取るために待つ場合のタイムアウト。デフォルト値は 600 秒 (10 分) です。

SERVER_TRANSMIT_TIMEOUT

(整数、秒 > 0) SMPP サーバーにデータを送信する場合のタイムアウト。デフォルト値は 120 秒 (2 分) です。

SMPP サーバーオプション

SMS Gateway Server には、異なる特徴を持つ複数の SMPP サーバーインスタンスを設定することができます。最も重要なインスタンスは、待機対象の TCP ポートとインタフェースです。言い換えると、SMPP サーバーが待機するネットワークインタフェースと TCP ポートの各ペアに、別個の特徴を設定することができます。このような特徴は、この節で説明するオプションを使用して指定します。

各インスタンスは次の形式のオプショングループ内に配置する必要があります。

```
[SMPP_SERVER=server-name]
option-value-1=option-value-1
option-value-2=option-value-2
...
option-name-n=option-value-n
```

文字列 `server-name` は、このインスタンスをほかのインスタンスから区別するためだけに使用されます。

表 D-23 に、SMPP サーバーの設定オプションの一覧を示します。

表 D-23　SMPP サーバーオプション

オプション	デフォルト	説明
LISTEN_BACKLOG	255	受信 SMPP サーバー接続の接続バックログ
LISTEN_CONNECTION_MAX		最大同時受信接続数
LISTEN_INTERFACE_ADDRESS		受信 SMPP サーバー接続のネットワークインタフェース

表 D-23　SMPP サーバーオプション (続き)

オプション	デフォルト	説明
LISTEN_PORT		受信 SMPP サーバー接続の TCP ポート
LISTEN_RECEIVE_TIMEOUT	600 秒	受信 SMPP サーバー接続の読み取りタイムアウト
LISTEN_TRANSMIT_TIMEOUT	120 秒	受信 SMPP サーバー接続の書き込みタイムアウト

LISTEN_BACKLOG

(整数、0 ～ 255 でその両端も含む) 受信 SMPP クライアント接続の TCP スタックによって許容される接続バックログ。デフォルトは 255 です。

LISTEN_CONNECTION_MAX

(整数、>= 0) この SMPP サーバーインスタンスで許可される最大同時受信 TCP 接続数。この値は、グローバル設定の LISTEN_CONNECTION_MAX の値を超えた場合は無視されます。

LISTEN_INTERFACE_ADDRESS

(文字列、「*INADDR_ANY*」またはドット付き 10 進表記の IP アドレス) 受信 SMPP クライアント接続で待機対象のネットワークインタフェースの IP アドレス。文字列「INADDR_ANY」(すべての使用可能なインタフェース) またはドット付き 10 進表記の IP アドレスのどちらかです (例 : 193.168.100.1)。デフォルト値は、「INADDR_ANY」です。

LISTEN_PORT

(整数、TCP ポート番号) 受信 SMPP クライアント接続を受け入れるためのバインド先 TCP ポート。このオプションの指定は必須です。このオプションにはデフォルト値はありません。このサービスには、IANA からの割り当てはないことに注意してください。

LISTEN_RECEIVE_TIMEOUT

(整数、秒 > 0) SMPP クライアントからデータを読み取るために待つ場合のタイムアウト。デフォルト値は 600 秒 (10 分) です。

LISTEN_TRANSMIT_TIMEOUT

(整数、秒 > 0) SMPP クライアントにデータを送信する場合のタイムアウト。デフォルト値は 120 秒 (2 分) です。

ゲートウェイプロファイルのオプション

ゲートウェイプロファイルの数はゼロ以上です。SMS Gateway Sever の設定ファイルのオプショングループ内で、各ゲートウェイプロファイルは次の形式で宣言されています。

```
[GATEWAY_PROFILE=profile-name]
option-name-1=option-value-1
option-name-2=option-value-2
...
option-name-n=option-value-n
```

文字列 profile-name は、このプロファイルをほかのオリジナルのプロファイルから区別するためだけに使用されます。

表 D-24 に、SMS Gateway Server プロファイルオプションの一覧を示します。

表 D-24　SMS Gateway Server プロファイルオプション

オプション	デフォルト	説明
CHANNEL	sms	メッセージをキューに入れるために使用されるチャネル
EMAIL_BODY_CHARSET	US-ASCII	電子メールメッセージ本文に使用される文字セット
EMAIL_HEADER_CHARSET	US-ASCII	電子メールメッセージヘッダーに使用される文字セット
FROM_DOMAIN		電子メールを SMS にルーティングし直すために使用されるドメイン名
PARSE_RE_0, PARSE_RE_1, ..., PARSE_RE_9		SMS メッセージテキストを構文解析するために使用される正規表現
PROFILE	GSM	GSM、TDMA、または CDMA の環境で機能する SMS プロファイル
SELECT_RE		プラグインの選択に使用される正規表現
SMSC_DEFAULT_CHARSET	US-ASCII	SMSC のデフォルトの文字セット
USE_SMS_PRIORITY	0	Gateway SMS の電子メールへの優先順位フラグ
USE_SMS_PRIVACY	0	Gateway SMS の電子メールへのプライバシーインジケータ

CHANNEL

(文字列、1 ～ 40 文字) 電子メールメッセージをキューに入れるために使用される MTA チャネルの名前。指定されていない場合は、「sms」と仮定されます。指定するチャネルは、MTA の設定で定義されている必要があります。

EMAIL_BODY_CHARSET

(文字列、文字セット名) 電子メールメッセージ本文への挿入前に、SMS テキストを変換するために使用する文字セット。必要に応じて、変換後のテキストは MIME でエンコードされます。デフォルト値は US-ASCII です。SMS メッセージに文字セットにないグリフが含まれている場合、そのグリフはニーモニック文字に変換されます。ニーモニック文字は、受取人にとっては意味をなさない場合があります。

MTA に認識される文字セットの一覧は、次のファイルで示されています。

installation-directory/config/charsets.txt

EMAIL_HEADER_CHARSET

(文字列、文字セット名) RFC 822 `Subject:` ヘッダー行への挿入前に、SMS テキストを変換するために使用する文字セット。必要に応じて、変換後の文字列は MIME でエンコードされます。デフォルト値は US-ASCII です。SMS メッセージに文字セットにないグリフが含まれている場合、そのグリフはニーモニック文字に変換されます。ニーモニック文字は、受取人にとっては意味をなさない場合があります。

FROM_DOMAIN

(文字列、IP ホスト名、1 ～ 64 文字) 電子メール用にエンベロープ `From:` アドレスを作成する際、SMS ソースアドレスに付加するドメイン名。指定する名前は、電子メールを SMS にルーティングし直す場合に使用される正しい名前である必要があります (たとえば、MTA SMS チャネルに関連付けられたホスト名)。指定しない場合は、`CHANNEL` オプションで指定されている正式なチャネル名が使用されます。

PARSE_RE_0, PARSE_RE_1, ..., PARSE_RE_9

(文字列、UTF-8 正規表現) モバイルを起点とする電子メールの場合、ゲートウェイプロファイルは SMS メッセージのテキストから宛先電子メールアドレスを抽出する必要があります。これは、1 つまたは複数の POSIX 準拠の正規表現 (RE) を使用することによって処理されます。SMS メッセージのテキストは、各正規表現によって、宛先電子メールアドレスを生成する一致が見つかるか、あるいは正規表現のリストが尽きるまで評価されます。

注 `PARSE_RE_*` と `ROUTE_TO` の各オプションの使用は、互いに排他的です。これらの両方を同一のゲートウェイプロファイルで使用すると、設定エラーになります。

各正規表現は、POSIX に準拠していて、UTF-8 文字セットにエンコードされている必要があります。正規表現では、宛先アドレスは文字列 0 として出力されます。状況に応じて、Subject: ヘッダー行で使用されるテキストは文字列 1 として、メッセージ本文で使用されるテキストは文字列 2 として出力されることがあります。正規表現によって「消費」されないテキストはいずれも、メッセージ本文で使用され、文字列 2 のテキスト出力に続きます。

正規表現は、PARSE_RE_0、PARSE_RE_1、...、の順序で、PARSE_RE_9 まで試されます。正規表現が指定されていない場合、次に示すデフォルトの正規表現が使用されます。

```
[ ¥t]*([^¥( ]*)[ ¥t]*(?:¥(([^¥)]*¥))?[ ¥t]*(.*)
```

このデフォルトの正規表現は、次に示す構成要素から成ります。

```
[ ¥t]*
```

先頭のホワイトスペース文字 (SPACE および TAB) を無視します。

```
([^¥( ]*)
```

宛先電子メールアドレス。これは最初に報告される文字列です。

```
[ ¥t]*
```

ホワイトスペース文字を無視します。

```
(?:¥(([^¥)]*)$1¥))?
```

括弧で囲まれたオプションの件名テキスト。これは 2 番目に報告される文字列です。先頭の ?: によって、外側の括弧は文字列を報告しなくなります。括弧は、括弧内の内容を末尾の ? の単一の RE にグループ化するためにのみ使用されます。末尾の ? によって、この RE 構成要素は、0 回または 1 回のみ照合されます。これは {0,1} の表現と同等です。

```
[ ¥t]*
```

ホワイトスペース文字を無視します。

```
(.*)
```

残りのテキストをメッセージ本文へ。これは 3 番目に報告される文字列です。

例として、上記の正規表現で次のサンプル SMS メッセージを処理する場合を示します。

```
dan@sesta.com(Testing)This is a test
```

この場合、次の電子メールメッセージが生成されます。

```
To: dan@sesta.com
Subject: Testing

This is a test
```

別の例として、次の SMS メッセージの場合を示します。

```
sue@sesta.com This is another test
```

この場合、次の電子メールメッセージが生成されます。

```
To: sue@sesta.com

This is another test
```

これらの正規表現で評価される前に、SMS メッセージは Unicode のエンコード方式である UTF-16 に変換されることに注意してください。その後、変換されたテキストは、UTF-8 から UTF-16 に変換済みの正規表現で評価されます。評価の結果は、宛先電子メールアドレスの場合は US-ASCII に変換されます。`Subject:` テキスト (ある場合) には `EMAIL_HEADER_CHARSET`、メッセージ本文 (ある場合) には `EMAIL_BODY_CHARSET` が使用されます。

PROFILE

(文字列、「GSM」、「TDMA」、または「CDMA」) 仮定される SMS プロファイル。現在のところ、この情報は SMS 優先順位フラグを RFC 822 `Priority:` ヘッダー行にマップするためにのみ使用されます。したがって、このオプションは、`USE_SMS_PRIORITY=0` (デフォルト) の場合は無効です。

SELECT_RE

(文字列、US-ASCII 正規表現) US-ASCII POSIX 準拠の正規表現。各 SMS メッセージの SMS 宛先アドレスと照合するために使用します。SMS メッセージの宛先アドレスがこの RE と一致した場合、SMS メッセージは通過するゲートウェイのゲートウェイプロファイルに合致する電子メールに送信されます。

SMS メッセージの宛先アドレスは US-ASCII 文字セットで指定されているので、この正規表現も US-ASCII で表現されている必要があることに注意してください。

SMSC_DEFAULT_CHARSET

(文字列、文字セット名) リモート SMSC で使用されるデフォルトの文字セットの名前。このオプションに選択する一般的な値は 2 つあり、それは US-ASCII と UTF-16-BE (USC2) です。指定されていない場合は、US-ASCII と仮定されます。

USE_SMS_PRIORITY

(整数、0 または 1) デフォルト (USE_SMS_PRIORITY=0) では、SMS メッセージ内の優先順位フラグは無視され、電子メールメッセージとともには送信されません。優先順位フラグを電子メールに付けて渡すには、USE_SMS_PRIORITY=1 と指定します。表 D-25 に、優先順位フラグを電子メールに付けて渡した場合の SMS から電子メールへのマッピングを示します。

表 D-25 優先順位フラグの SMS から電子メールへのマッピング

SMS プロファイル	SMS 優先度フラグ	電子メールの Priority: ヘッダー行
GSM	0 (優先でない)	ヘッダー行なし (Normal を示す)
	1、2、3 (優先)	Urgent
TDMA	0 (バルク)	Nonurgent
	1 (標準)	ヘッダー行なし (Normal を示す)
	2 (至急)	Urgent
	3 (大至急)	Urgent
CDMA	0 (標準)	ヘッダー行なし (Normal を示す)
	1 (インタラクティブ)	Urgent
	2 (至急)	Urgent
	3 (緊急)	Urgent

電子メールの Priority: ヘッダー行の値は、Nonurgent、Normal、および Urgent です。

USE_SMS_PRIVACY

(整数、0 または 1) デフォルト (USE_SMS_PRIVACY=0) では、SMS プライバシーの指示は無視され、電子メールメッセージとともには送信されません。この情報を電子メールに付けて渡すには、USE_SMS_PRIVACY=1 と指定します。表 D-26 に、プライバシー情報を電子メールに付けて渡した場合の SMS から電子メールへのマッピングを示します。

表 D-26 プライバシーフラグの SMS から電子メールへのマッピング

SMS プライバシーフラグ	電子メールの Sensitivity: ヘッダー行
0 (制約なし)	ヘッダー行なし
1 (制限あり)	Personal
2 (親展)	Private
3 (秘密)	Company-confidential

電子メールの Sensitivity: ヘッダー行の値は、Personal、Private、および Company-confidential です。

双方向 SMS の設定例

動作についての仮定

この例では、次の動作を設定するものと仮定します。

- 次のアドレスに宛てた電子メールメッセージがある

  sms-id@sms.domain.com

  この電子メールメッセージは次の SMS アドレスに送信される

  sms-id

  000*nnnnnnnnnn* の範囲の一意の SMS ソースアドレスを付与する。

- SMS アドレス 000 に宛てたモバイルの SMS メッセージは、SMS メッセージテキストの冒頭から抽出された電子メールアドレスとともに、ゲートウェイを介して電子メールに送信される。

  たとえば、次の SMS メッセージテキストの場合、

  jdoe@domain.com Interested in a movie?

  メッセージ「Interested in a movie?」は jdoe@domain.com に送信される。

- 000*nnnnnnnnnn* に送信された SMS 通知はゲートウェイを介して電子メールに送信され、メッセージの差出人に配信され、受信確認される。

この動作を実現するために、次の仮定と指示に従ってください。

追加の仮定と指示

- MTA の SMS チャネルはドメイン名 sms.domain.com を使用する。
- SMS Gateway Server はホストゲートウェイ .domain.com 上で実行され、以下のものを使用する。
  - SMPP リレー用に TCP ポート 503
  - SMPP サーバー用に TCP ポート 504
- リモート SMSC の SMPP サーバーはホスト smpp.domain.com 上で実行され、TCP ポート 377 を待機する。
- リモート SMSC のデフォルトの文字セットは、UCS2 (aka、UTF-16) である。

SMS チャネルの設定

上記の動作を有効にするには、次に示す SMS チャネルの設定を imta.cnf ファイルで使用します (以下の行をファイルの最下部に追加)。

```
（空白行）
sms
sms.domain.com
```

SMS チャネルオプションファイル

チャネルのオプションファイル sms_option には、次の設定を含めます。

```
SMPP_SERVER=gateway.domain.com
SMPP_PORT=503
USE_HEADER_FROM=0
DEFAULT_SOURCE_ADDRESS=000
GATEWAY_PROFILE=sms1
SMSC_DEFAULT_CHARSET=UCS2
```

SMS Gateway Server の設定

Gateway Server の設定ファイル sms_gateway.cnf は次のようになります。

```
HISTORY_FILE_DIRECTORY=/sms_gateway_cache/

[SMPP_RELAY=relay1]
LISTEN_PORT=503
SERVER_HOST=smpp.domain.com
SERVER_PORT=377

[SMPP_SERVER=server1]
LISTEN_PORT=504

[GATEWAY_PROFILE=sms1]
SELECT_RE=000([0-9]{10,10}){0,1}
SMSC_DEFAULT_CHARSET=UCS2
```

この設定をテストする

テストに使用する SMSC がない場合は、ループバックテストを実行する必要があります。sms_option ファイルにいくつか追加で設定すると、上記の設定の単純なループバックテストを実行できます。

sms_option ファイルへの追加設定

sms_option ファイルへの追加設定は、次のとおりです。

```
! テキストを SMS メッセージの本文に追加しないようにする設定
FROM_FORMAT=
SUBJECT_FORMAT=
CONTENT_PREFIX=
```

この設定を行わないと、次の内容の電子メールは、

```
user@domain.com (Sample subject) Sample text
```

次の SMS メッセージに変換されます。

```
From:user@domain.com Subject:Sample Subject Msg:Sample text
```

これは、モバイルから電子メールのコードで期待される形式にはなりません。期待される形式は次のとおりです。

```
user@domain.com (Sample subject) Sample text
```

したがって、ループバックテストを行う場合は、空の文字列を FROM_FORMAT、SUBJECT_FORMAT、および CONTENT_PREFIX オプションに指定する必要があります。

ループバックテストを実行する

次のようなテスト電子メールメッセージを 000@sms.domain.com 宛に送信します。

```
user@domain.com (Test message) This is a test message which
should loop back
```

その結果、この電子メールメッセージは電子メール受信者 user@domain.com にルートバックされます。このテストに使用する DNS または ホストテーブルには、sms.domain.com を必ず追加しておいてください。

SMS Gateway Server のストレージ要件

SMS Gateway Server に必要なリソース量を判断するには、表 D-27 の要件から算出した数字とともに、1 秒間にリレーされるメッセージの期待数および RECORD_LIFETIME 設定を考慮します。

表 D-27 に、履歴レコード、SMPP リレー、および SMPP サーバーの要件を示します。

表 D-27　SMS Gateway Server のストレージ要件

コンポーネント	要件
メモリ内履歴レコード	リレーされるメッセージごとに 33+m+s バイトの仮想メモリが必要です。m はメッセージの SMS メッセージ ID の長さ (1 <= m <= 64) で、s はメッセージの SMS ソースアドレスの長さ (1 <= s<= 20) です。 MAKE_SOURCE_ADDRESS_UNIQUE=0 の場合、16+m バイトのみが使用されます。64 ビットのオペレーティングシステムの場合、49+m+s バイトの仮想メモリがレコードごとに消費されます [MAKE_SOURCE_ADDRESS_UNIQUE=0 の場合は 24+m]。 ヒープアロケータは、実際には各レコードに大きめの仮想メモリを割り当てる場合があることにも注意してください。 最大レコード数は、430 億 (2**32-1) です。1680 万レコード (2**24) 未満の場合、ハッシュテーブルは約 16M バイトを消費します。6710 万レコード (2**26) 未満の場合、ハッシュテーブルは約 64M バイトを消費します。6710 万レコードを超えると、ハッシュテーブルは約 256M バイトを消費します。 64 ビットのオペレーティングシステムの場合は、メモリ消費量は 2 倍になります。 これらの消費量は、各レコード自体が必要とするメモリ消費量に追加されます。

表 D-27　SMS Gateway Server のストレージ要件 (続き)

コンポーネント	要件
ディスク上の履歴レコード	リレーされるメッセージごとに、平均で次のバイト数が必要です。 81+m+2s+3a+S+2i ここで、 • m は SMS メッセージ ID の平均の長さであり、1<=m<=64 • s は SMS ソースアドレスの平均の長さであり、1 <= s <= 20 • a は電子メールアドレスの平均の長さであり、3<=a<=129 • S は Subject: ヘッダー行の平均の長さであり、0 <= S<= 80 • i は 電子メールメッセージのエンベロープ ID の平均の長さであり、0 <= i <= 129 レコードのサイズは、メッセージのエンベロープ From: アドレスと To: アドレス、エンベロープ ID とメッセージ ID、および Subject: ヘッダー行の長さから影響を受けます。 最大レコード長は 910 バイトです。 MAKE_SOURCE_ADDRESS_UNIQUE=0 の場合、各レコードのサイズ (単位 : バイト) は次のようになります。 78+m+3a+S+2i
SMPP リレー	リレーされる SMPP セッションごとに、2 つの TCP ソケットを消費します。1 つはローカル SMPP クライアントに使用され、もう 1 つはリモート SMPP サーバーに使用されます。32 ビットのオペレーティングシステムの場合、約 1K バイトの仮想メモリが接続ごとに消費されます。64 ビットのオペレーティングシステムの場合は、約 2K バイト消費されます。
SMPP サーバー	着信接続ごとに 1 つの TCP ソケットを消費します。32 ビットのオペレーティングシステムの場合、約 1K バイトの仮想メモリが接続ごとに消費されます。64 ビットのオペレーティングシステムの場合は、約 2K バイト消費されます。

たとえば、1 秒間に平均 50 メッセージがリレーされると予想し、SMS ソースアドレスの長さは 13 バイト、SMS メッセージ ID は一般的な長さの 12 バイト、電子メールアドレスは 24 バイト、Subject: 行は 40 バイト、電子メールメッセージと ID がそれぞれ 40 バイト、履歴データは 7 日間保持されるとする場合、結果は次のようになります。

- 保存対象の 3240 万件の履歴レコードが存在します。各レコードはメモリ内で平均 58 バイト、ディスク上で 311 バイトの長さです。
- メモリ内の履歴レコードの消費量は、約 1.70G バイト (1.63G バイト + 64M バイト) になります。
- ディスク上のストレージは、約 8.76G バイトになります。

容量が十分あるディスクを使用すれば、いずれのディスク要件にも対応できるものの、32 ビットマシンの仮想メモリ要件は約 2G バイトであり、これは厳しい制限です。仮想メモリまたはディスクストレージの所要量を削減するには、RECORD_LIFETIME オプションを使用して、レコードが保持される期間を短くしてください。

インストールワークシート

この付録では、インストールを計画するためのワークシートを示します。この付録には以下のワークシートが付いています。

- Directory Server のインストール
- 管理サーバーの初期実行時設定
- Directory Server 設定スクリプト (comm_dssetup.pl)
- Messaging Server の初期実行時設定

Directory Server のインストール

Directory Server を Java Enterprise System インストーラまたは以前のインストールによってインストールします。表 E-1 の Directory Server のインストールおよび設定パラメータに記入してください。これは『Communications Services 配備計画ガイド』に示されているワークシートのレプリカです。Administraion Server と Messaging Server をインストールして設定するときは、これらのパラメータが必要になります。

表 E-1 Directory Server のインストールパラメータ

パラメータ :	説明 :	例 :	使用する場所 :	回答 :
Directory インストールルート	サーバープログラム、設定、保守、および情報ファイルを保管するための、専用のディレクトリサーバーマシン上のディレクトリ。	`/var/mps/serverroot/`	`comm_dssetup.pl` Perl スクリプト 43 ページの「Messaging Server 設定用に Directory Server を準備するには」を参照してください。	

表 E-1　Directory Server のインストールパラメータ (続き)

パラメータ :	説明 :	例 :	使用する場所 :	回答 :
ホスト	ホスト名は、IP ホスト名。IP ホスト名としては、「短縮形」のホスト名 (fiddle など) または完全修飾ホスト名が使用されます。完全修飾ホスト名は、ホスト名とドメイン名の 2 つの部分から構成されます。	fiddle.west.sesta.com	管理サーバーの設定 43 ページの「Messaging Server 設定用に Directory Server を準備するには」を参照してください。	
LDAP ディレクトリのポート番号	LDAP ディレクトリサーバーのデフォルト値は 389 です。	389	管理サーバーの設定と Messaging Server の設定 43 ページの「Messaging Server 設定用に Directory Server を準備するには」および 53 ページの「Messaging Server の初期実行時設定を作成するには」を参照してください。	
管理者 ID およびパスワード	設定情報の管理責任がある管理者。 管理者のパスワード	Admin PaSsWoRd	管理サーバーの設定 43 ページの「Messaging Server 設定用に Directory Server を準備するには」を参照してください。	
ユーザーおよびグループツリーのサフィックス	ディレクトリツリーの最上位にある LDAP エントリの識別名。ユーザーデータおよびグループデータはその下に保管されます。	o=usergroup	comm_dssetup.pl Perl スクリプト 43 ページの「Messaging Server 設定用に Directory Server を準備するには」を参照してください。	

表 E-1　Directory Server のインストールパラメータ (続き)

パラメータ :	説明 :	例 :	使用する場所 :	回答 :

Directory Manager の DN およびパスワード	UNIX のスーパーユーザーに相当する権限を持つディレクトリ管理者。通常は、この管理者がユーザーデータおよびグループデータを管理します。
	
	Directory Manager のパスワード。
		`cn=Directory Manager`
		
		`pAsSwOrD`
			`comm_dssetup.pl` Perl スクリプトおよび Messaging Server の設定
			
			43 ページの「Messaging Server 設定用に Directory Server を準備するには」および 53 ページの「Messaging Server の初期実行時設定を作成するには」を参照してください。	

管理ドメイン	管理制御の対象範囲。	`System Lab`	管理サーバーの設定 43 ページの「Messaging Server 設定用に Directory Server を準備するには」を参照してください。	

管理サーバーの初期実行時設定

管理サーバーの初期実行時設定プログラムを Java Enterprise System インストーラから実行するときは、表 E-2 にインストールパラメータを記入してください。これは『Communications Services 配備計画ガイド』に示されているワークシートのレプリカです。Messaging Server の初期実行時設定では、これらのパラメータの一部が必要になります。いくつかの質問に対する回答には、987 ページの「Directory Server のインストール」のチェックリストも参考になります。

表 E-2　管理サーバーの初期実行時設定プログラムのパラメータ

パラメータ	説明	例	回答 :
完全指定ドメイン名	ホストマシンの完全指定ドメイン。	`fiddle.west.sesta.com`	
サーバールートの定義	サーバープログラム、設定、保守、および情報ファイルを保管するための、専用の管理サーバーのインストールルート。	`/var/mps/serverroot`	
UNIX システムユーザー	システムユーザーが、実行するプロセスに対して適切なアクセス権を持つように指定する特定の権限。	`inetuser`	
UNIX システムグループ	特定の UNIX システムユーザーが所属するグループ。	`inetgroup`	
Configuration Directory Server	987 ページの「Directory Server のインストール」中に指定したホストとポート。	ホスト `fiddle.west.sesta.com` ポート `389`	
Configuration Directory Server の管理者およびパスワード	987 ページの「Directory Server のインストール」中に指定した管理者 ID。 管理者 ID のパスワード	`Admin` PaSsWoRd	
管理ドメイン	管理制御の対象範囲。 Messaging Server と Directory Server を同じマシン上にインストールした場合は、987 ページの「Directory Server のインストール」時に同じ管理ドメインを選択する必要があります。	System Lab2	
管理サーバーのポート	管理サーバー専用の一意のポート番号。	`5555`	

Directory Server 設定スクリプト (comm_dssetup.pl)

Directory Server 設定スクリプト (comm_dssetup.pl) を実行して Messaging Server の設定用に Directory Server を準備する場合は、表 E-3 にインストールパラメータを記入してください。Messaging Server の初期実行時設定では、これらのパラメータの一部が必要になります。

表 E-3 comm_dssetup.pl スクリプトパラメータ

パラメータ	説明	例	回答 :
サーバールート	サーバープログラム、設定、保守、および情報ファイルを保管するための、専用の Directory Server のインストールルート。	/var/mps/serverroot/	
サーバーインスタンス	ほとんどの機能を管理する LDAP Directory Server デーモンまたはサービス。導入によっては、インスタンスをユーザーとグループの保守専用に取っておき、設定用に別のインスタンスを管理することがあります。	slapd-varrius	
DC ルート	二重ツリーの DIT プロビジョニングモデル (Sun LDAP Schema 1 または Sun ONE LDAP Schema.2 の互換モード) を持ちたい場合は、DC ツリーがローカル DNS 構造をミラー化し、システムが DC ツリーを、ユーザーデータおよびグループデータのエントリを含む組織ツリーへのインデックスとして使用します。	o=internet	
ユーザーおよびグループベースのサフィックス	ユーザーエントリおよびグループエントリのネームスペースを保持する、組織ツリー内の最上位エントリ。	o=usergroup	
Directory Manager の DN およびパスワード	組織ツリー内のユーザーデータおよびグループデータを管理する管理者。Sun Java Enterprise System インストーラで指定した管理者と同じ管理者にする必要があります。 Directory Manager DN のパスワード	cn=Directory Manager pAsSwOrD	

Messaging Server の初期実行時設定

Messaging Server の初期実行時設定プログラムを実行するときは、表 E-4 にインストールパラメータを記入してください。いくつかの質問に対する回答には、987 ページの「Directory Server のインストール」のチェックリストも参考になります。

表 E-4　Messaging Server の初期実行時設定プログラムのパラメータ

パラメータ	説明	例	回答：
設定およびデータディレクトリ	Messaging Server 設定ファイルのすべてを含みます *msg_svr_base*/ データディレクトリは、このディレクトリへのシンボリックリンクを作成します。	`/var/opt/SUNWmsgsr`	
UNIX システムユーザー	システムユーザーが、実行するプロセスに対して適切なアクセス権を持つように指定する特定の権限。このシステムユーザーは、管理サーバーの初期実行時設定で指定したユーザーと同じにしません。	`mailsrv`	
UNIX システムグループ	特定の UNIX システムユーザーが所属するグループ。このシステムグループは、管理サーバーの初期実行時設定で指定したグループと同じにしません。	`mail`	
Configuration Directory LDAP URL、Directory Manager、およびパスワード	Configuration Directory Server、LDAP URL、バインド DN、およびパスワード	ldap://fiddle.west.sesta.com:389 cn=Directory Manager PaSsWoRd	

User/Group Directory LDAP URL、Directory Manager、およびパスワード	User/Group Directory Server、LDAP URL、バインド DN、およびパスワード。
	
	ユーザーおよびグループディレクトリは設定ディレクトリとは別のものにすることをお勧めします。
		`ldap://fiddle.west.sesta.com:389`
		
		`cn=Directory Manager`
		
		`PaSsWoRd`	

ポストマスターの電子メールアドレス	ポストマスターのメールを監視する管理者の電子メールアドレス。アドレスはメールボックスと関連付けられているため、完全指定アドレスにする必要があります。	`pma@siroe.com`	

表 E-4　Messaging Server の初期実行時設定プログラムのパラメータ (続き)

パラメータ	説明	例	回答 :
管理者アカウントのパスワード	サービス管理者、ユーザーおよびグループの管理者、エンドユーザー管理権限、および PAB 管理者と SSL パスワードに使用するパスワード。	paSSwoRD	
デフォルトの電子メールドメイン	ドメインが指定されていない場合に使用される、電子メールのデフォルト	siroe.com	
デフォルトの電子メールドメインの組織名	組織が属していて、組織ツリーの構築に使用される組織名。	たとえば、組織名が Engineering の場合、siroe.com (デフォルトの電子メールドメイン) のすべてのユーザーは、LDAP DN o=Engineering, o=usergroup の下に置かれます ユーザーおよびグループディレクトリサフィックスは、comm_dssetup.pl で指定されています。	

用語集

このマニュアルセットで使用されている用語の完全なリストについては、『Sun Java Enterprise System 用語集』(http://docs.sun.com/doc/819-1933?l=ja) を参照してください。

索引

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

ほ

ま

み

め

も

ゆ

よ

り

る

れ

ろ

わ