WebSphere MQ internet pass-thru バージョン 1.3

SC88-9258-01
(英文原典：SC34-6100-01)

お願い

本書および本書で紹介する製品をご使用になる前に、 181 ページの『特記事項』に記載されている情報をお読みください。

本書は、WebSphere MQ internet pass-thru のバージョン 1.3 (プログラム番号 5639-L92) に適用されます。また、改訂版で特に断りのない限り、それ以降のすべてのリリースおよびモディフィケーションにも適用されます。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

http://www.ibm.com/jp/manuals/main/mail.html

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

http://www.ibm.com/jp/manuals/ の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原　典： SC34–6100–01
WebSphere MQ internet pass-thru
Version 1.3

発　行： 日本アイ・ビー・エム株式会社

担　当： ナショナル・ランゲージ・サポート

第1刷 2003.3

この文書では、平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財) 日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

目次

図

まえがき

internet pass-thru とは ?

WebSphere MQ internet pass-thru は、以前は MQSeries internet pass-thru と呼ばれていました。本書では、MQSeries を WebSphere MQ と呼ぶことにします。ただし、すべての MQSeries のマニュアルが直ちに名前を WebSphere MQ に変更するわけではなく、しばらくは、MQSeries と WebSphere MQ の両方を使用することになります。

IBM® WebSphere MQ internet pass-thru は、以下の特徴を備えています。

- WebSphere MQ 基本製品を拡張したもので、インターネットを介したリモート・サイト間でのメッセージング・ソリューションをインプリメントする場合に使用できます。
- WebSphere MQ チャネル・プロトコルを HTTP の中に組み込んだり、プロキシーとして機能させたりすることにより、このプロトコルがファイアウォールに出入りする通路をより簡単、かつより管理可能なものにします。
- WebSphere MQ メッセージ・フローの送受信が可能なスタンドアロン・サービスとして働きます。それを実行するシステムは、WebSphere MQ キュー・マネージャーにホストとしてのサービスを提供する必要がありません。
- WebSphere MQ を使用して企業間トランザクションを提供する手助けをします。
- 既存の未変更 WebSphere MQ アプリケーションをファイアウォールで使用できるようにします。
- 複数のキュー・マネージャーにアクセスする場合の単一制御点を備えています。
- すべてのデータの暗号化を可能にします。
- すべての接続試行をログに記録します。

本書では便宜上、WebSphere MQ internet pass-thru をしばしば “MQIPT” と呼んでいます。

対象読者

本書は、システム設計者、WebSphere MQ 技術管理者、ファイアウォールおよびネットワーク管理者向けに作成されています。

本書を理解する上での必要な知識

以下のことを十分に理解しておく必要があります。

- WebSphere MQ キュー・マネージャーとメッセージ・チャネルの管理は、「*WebSphere MQ システム管理ガイド*」および「*WebSphere MQ 相互通信*」に記述されています。
- ファイアウォールのインプリメント方法
- インターネット・プロトコルの経路 (ルート) 指定 / ネットワーキング
- ロード・バランシングおよび拡張可用性のための IBM Network Dispatcher

- IBM WebSphere® Application Server

前提条件

当リリースの internet pass-thru は、以下のオペレーティング・システムで稼働します。

- Windows NT® V4.0 (Service Pack 6 を適用したもの)
- Windows® 2000
- Windows XP
- Sun Solaris
- AIX® V5.1
- HP-UX 11
- Linux

J2SE V1.4.0 runtime (JRE) が MQIPT サーバーに必要となります。 SDK、V1.4.0 は、セキュリティー出口を作成する場合に必要になります。

サポートされる唯一のネットワーク・プロトコルは TCP/IP です。

Administration Client ヘルプには Netscape ブラウザーが必要です。

アクセシビリティー情報

Administration Client GUI は、アクセシビリティーを考慮に入れて作成されています。キーボード相当機能を使用すれば、マウスを使用しなくても、提供されるすべての機能を簡単に実行できます。タブやシフト・タブ、Ctrl タブ、カーソル・キーなどを標準方法で使用して、画面をナビゲートすることができます。ボタンを押す操作に代わるものとして、まずボタンを選択し、次に Enter キーを押します。

メニュー・オプションを表示するには、タブとカーソル・キーを併用するか、またはアクセラレーター・キーを使用します。アクセラレーター・キーはすべてのオプションで使用できます。たとえば、GUI をクローズする場合は、まず alt-f を選択し、次に alt-q (File->Quit) を選択します。メニュー項目を表示したならば、Enter キーを使ってそれをアクティブにすることができます。

ツリーをナビゲートする場合は、カーソル・キーを使用します。特に、右カーソル・キーと左カーソル・キーを使って MQIPT ノードをオープンしたりクローズしたりできるため、経路の表示や非表示が可能になります。

選択したチェック・ボックスの状態を変更するには、スペース・キーを使用します。編集用のフィールドを選択するには、Enter キーを使用します。

ルック・アンド・フィール

理想的には、GUI は環境のルック・アンド・フィールを持っていなければなりません。これは必ずしも常に可能ではないので、構成ファイルを提供して GUI のルック・アンド・フィールをユーザーのニーズに合わせることができます。この構成ファイルは ″custom.properties″ と呼ばれていて、bin サブディレクトリーに入れておかなければなりません。

この構成ファイルを使用して以下の構成を行います。

- 前景色 - テキストのカラー
- 背景色
- テキストのフォント
- テキストのスタイル - プレーン、太字、イタリック、または太字イタリック

″customSample.properties″ 構成ファイルが提供されており、この構成ファイルにはその変更方法示すコメントが含まれています。このファイルを `bin/custom.properties` にコピーして、必要な変更を加えることをお勧めします。

変更の要約

この項では、この版の「WebSphere MQ internet pass-thru」で加えられた変更について説明します。本書の前回の版からの変更点は、変更箇所の左側に縦線でマークが付けられています。

この版 (SC88-9258-01 (英文原典 : SC34-6100-01)) での変更点

このバージョンの WebSphere MQ internet pass-thru には、以下の機能強化が含まれています。

- クライアント接続要求を制御するためのセキュリティー出口
- CRL および ARL の LDAP サポート
- 鍵リング・パスワードの暗号化
- 鍵リングからの証明書の選択
- 新しい AES 暗号スイート
- 総称 UNIX® ディスク・イメージ
- 経路再始動アクションの制御
- AIX および HP-UX プラットフォームは、Java™ 1.4 をサポートします。

旧版 (SC88-9258-00)(第 3 版) に対する変更

このバージョンの WebSphere MQ internet pass-thru には、以下の機能強化が含まれています。

- 出力ポート・アドレス割り振りの制御
- 構成の例
- 改良された SSL トレース
- Java Security Manager
- SSL 証明書と鍵リング・ファイルを管理するための KeyMan ユーティリティー
- Linux サポート (Quality of Service for WebSphere MQ メッセージを含む)
- Windows プラットフォームで使用できる NLS インストール・イメージ
- 大文字小文字を区別しないプロパティー名
- サーブレット・バージョン
- Socks クライアントおよびサーバー・サポート
- SSL プロキシー・モード
- マルチホーム・システムのサポート
- Administration Client 用のトラフィック・ライト状況
- WebSphere MQ クラスター・サポート

第 2 版に対する変更

このバージョンの WebSphere MQ internet pass-thru には、以下の機能強化が含まれています。

- MQIPT のプラットフォームとして AIX、HP-UX、および Windows 2000 の追加
- HTTP プロキシー・サポートの追加
- Secure Socket Layer (SSL) サポートの追加
- SOCKS プロキシーを介して別の外部 MQIPT または MQSeries® サーバーと通信できる MQIPT の機能
- 1 つまたは複数の MQIPT の管理を容易にするための Administration Client GUI の使用
- IBM Network Dispatcher のサポートの追加
- トレースの小さな改善
- mqiptAdmin コマンドの小さな改善

第 1 章 WebSphere MQ internet pass-thru の紹介

WebSphere MQ internet pass-thru は、WebSphere MQ の基本製品を拡張したものです。 MQIPT は、2 つの WebSphere MQ キュー・マネージャー間、あるいは WebSphere MQ クライアントと WebSphere MQ キュー・マネージャー間で WebSphere MQ メッセージ・フローの送受信を行うことができる、スタンドアロンのサービスとして稼働します。 MQIPT は、クライアントとサーバーが同じ物理ネットワーク上にない場合でもこの接続を可能にしています。

2 つの WebSphere MQ キュー・マネージャー間、または WebSphere MQ クライアントと WebSphere MQ キュー・マネージャー間の通信パスに 1 つまたは複数の MQIPT を設定することができます。 MQIPT を使用すれば、2 つの WebSphere MQ システムは、両者間に TCP/IP 直接接続を設けなくてもメッセージ交換を行えるようになります。この方法は、ファイアウォール構成により 2 つのシステム間の TCP/IP 直接接続が禁止されている場合に有効です。

MQIPT は、1 つまたは複数の TCP/IP ポートで着信接続を listen します。そこでは、通常の WebSphere MQ メッセージや、HTTP の中に組み込まれた WebSphere MQ メッセージ、 SSL (Secure Sockets Layer) で暗号化された WebSphere MQ メッセージを送信することができます。このサービスは、複数の同時接続を処理することができます。

最初の TCP/IP 接続要求を行う WebSphere MQ チャネルは「呼び出し元」と呼ばれ、呼び出し元の接続先チャネルは「レスポンダー」、呼び出し元の最終接続先であるキュー・マネージャーは「宛先キュー・マネージャー」と呼ばれます。

MQIPT の使用法としては、次のことが考えられます。

- MQIPT をチャネル・コンセントレーターとして使用することができる。これにより、いくつかの個別のホストに接続されたチャネルが、ファイアウォールからは、それらがすべて MQIPT ホストに接続されているように見えます。このため、ファイアウォール・フィルター規則の定義と管理が容易になります。

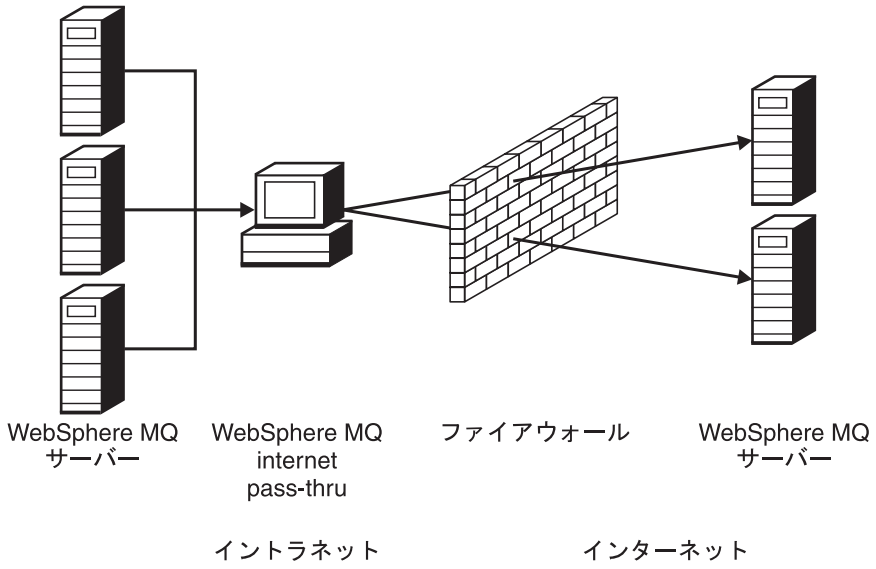

図 1. チャネル・コンセントレーターとしての MQIPT の例

- MQIPT をファイアウォールの「非武装地帯 (DMZ)」 (認識されたトラステッド・インターネット・プロトコル (IP) アドレスを持つマシン上の) に入れた場合は、MQIPT を使って、WebSphere MQ 着信チャネル接続を listen し、次にそれをトラステッド・イントラネットに転送することができます。内部ファイアウォールは、このトラステッド・マシンがインバウンド接続を行えるようにしなければなりません。この構成の場合、MQIPT は、外部からのアクセス要求からは、トラステッド・イントラネット内にある各マシンの本当の IP アドレスが見えないようにしています。このため、MQIPT は単一アクセス・ポイントを提供しています。

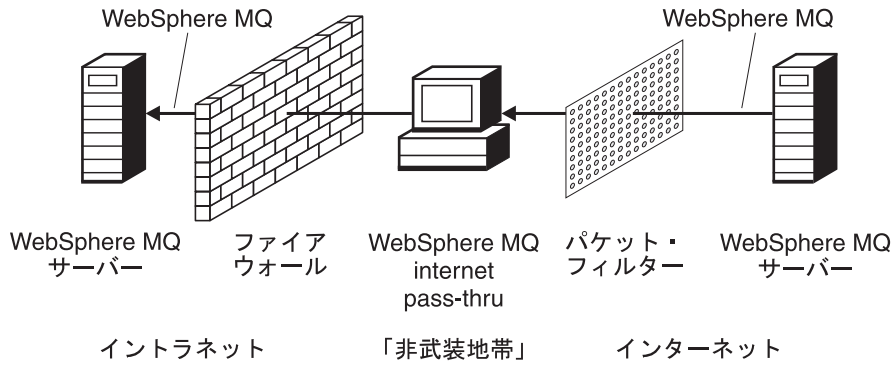

図 2. 「非武装地帯」を持つ MQIPT の例

- 2 つの MQIPT をインラインでデプロイした場合、両者は HTTP または SSL を使用して相互に通信することができます。 HTTP トンネル・フィーチャーを使用すれば、既存の HTTP プロキシーを利用することにより、要求をファイアウォール経由で送信することができます。最初の MQIPT は WebSphere MQ プロトコルを HTTP に挿入し、2 番目の MQIPT は、 WebSphere MQ プロトコルをその HTTP ラッパーから取り出して、それを宛先キュー・マネージャーに転送します。

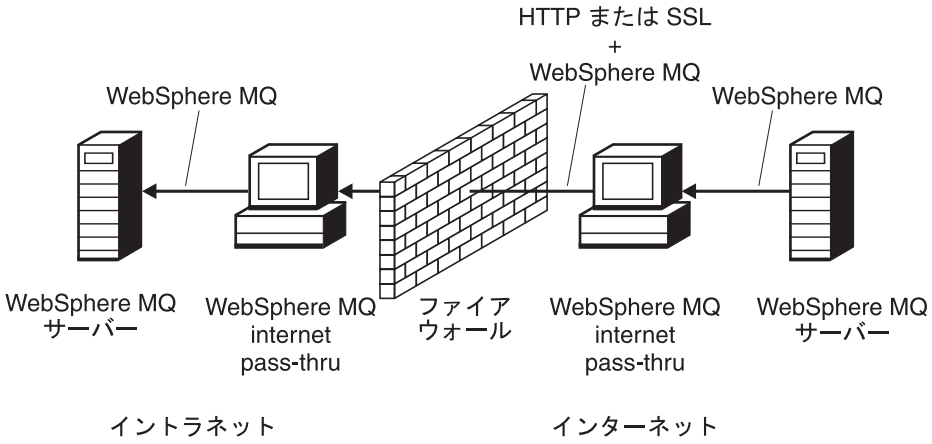

図 3. MQIPT および HTTP トンネル操作の例

- 同様に、要求は、暗号化してからファイアウォール経由で送信することができます。最初の MQIPT はデータを暗号化し、2 番目の MQIPT は、SSL を使用してそれを暗号化解除してから宛先キュー・マネージャーに送信します。

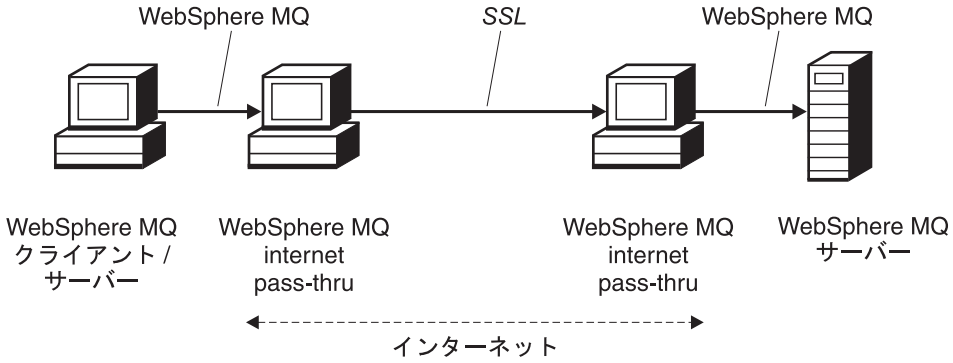

図 4. MQIPT と SSL の例

MQIPT は、データをソースから宛先へ送信する場合、それをメモリーに入れておきます。データがディスクに保管されることはありません (ただし、オペレーティング・システムによってディスクにページングされるメモリーを除きます)。 MQIPT が明示的にディスクにアクセスするのは、構成ファイルを読み取るときと、ログおよびトレース・レコードを書き込むときだけです。

全範囲の WebSphere MQ チャネル・タイプを 1 つまたは複数の MQIPT で使用することができます。通信パスに MQIPT が存在していても、接続された WebSphere MQ コンポーネントの機能特性には影響はありませんが、メッセージ転送のパフォーマンスには多少の影響がある可能性があります。

MQIPT は、WebSphere MQ Publish/Subscribe または WebSphere MQ Integrator メッセージ・ブローカーと一緒に使用できます。

5 ページの図 5 は、WebSphere MQ トポロジーの MQIPT で可能なすべての構成を示しています。この図では、「アウトバウンド接続」側のファイアウォールを超えたところにある HTTP プロキシー、SOCKS プロキシー、および MQIPT マシンがインターネット上で結合される可能性があることを示しています。たとえば、ある

MQIPT マシンは、 1 つまたは複数の SOCKS または HTTP プロキシー・マシン、さらには複数の MQIPT マシンと通信してからその宛先に到達することができます。

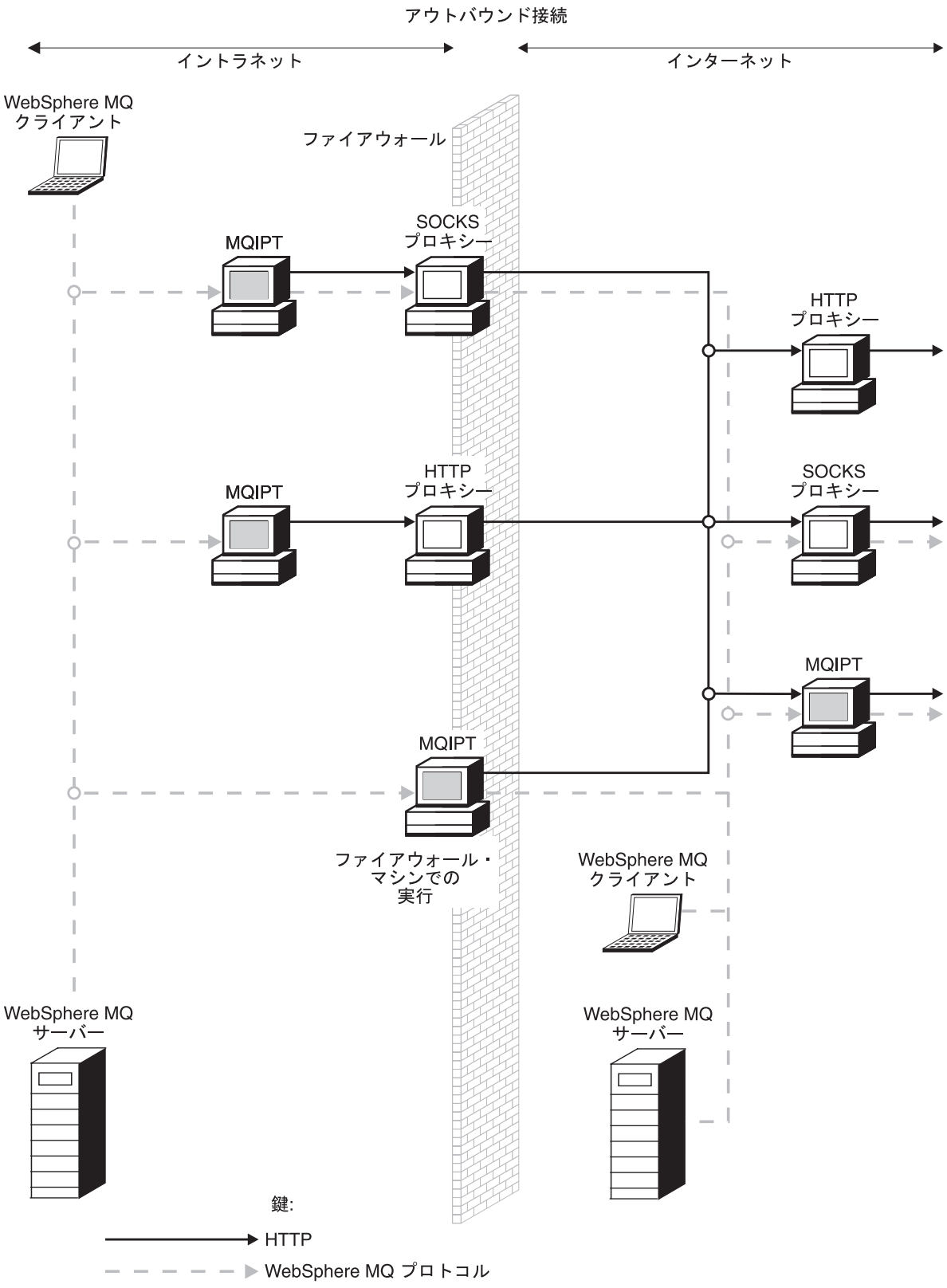

図 5. 可能な MQIPT 構成を示す WebSphere MQ トポロジー (1/2)

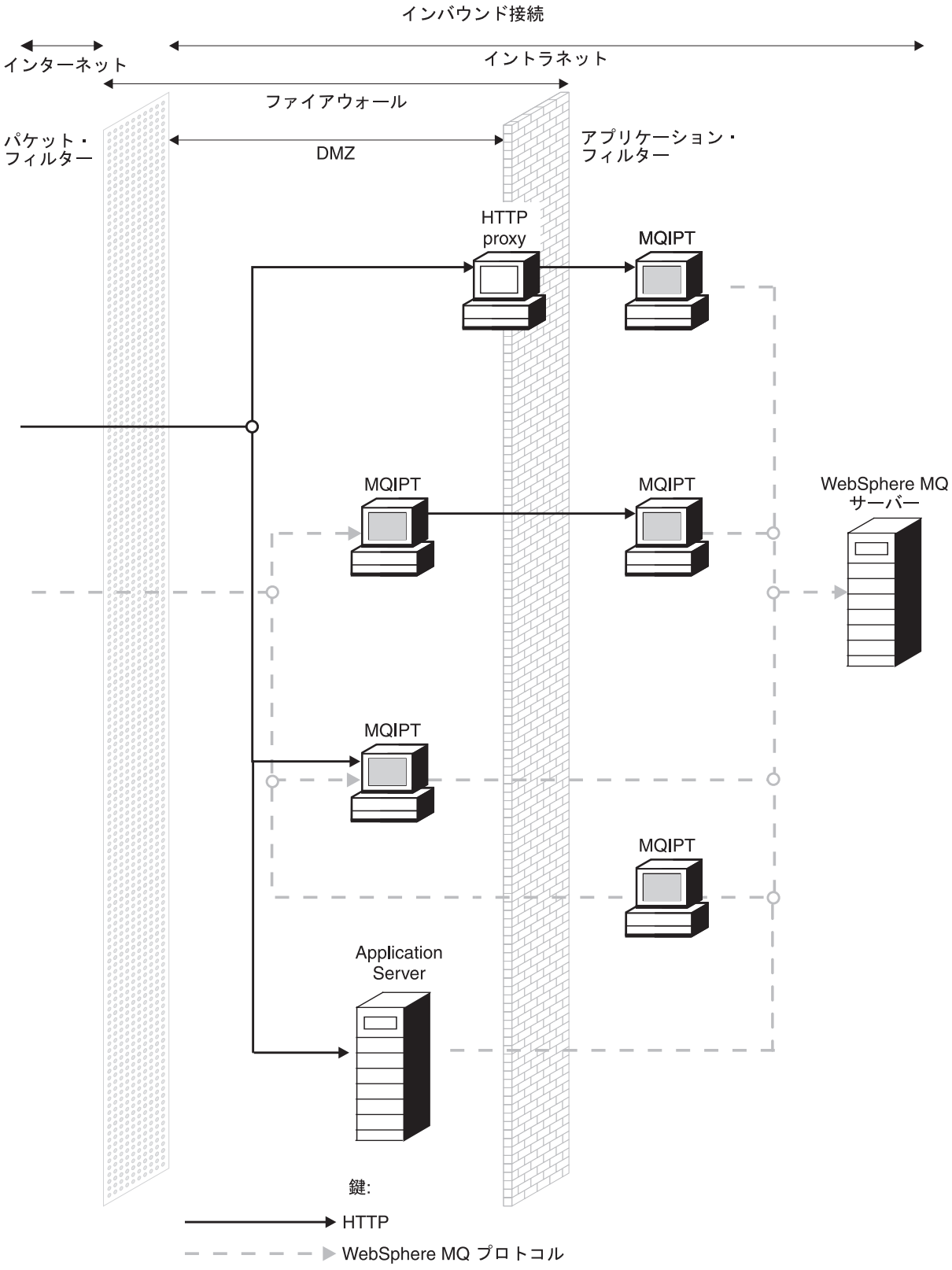

図 5. 可能な MQIPT 構成を示す WebSphere MQ トポロジー (2/2)

第 2 章 internet pass-thru の機能

この章では、internet pass-thru の機能の概要を説明します。

internet pass-thru の機能の概要

最も単純な構成の MQIPT は、WebSphere MQ プロトコル転送プログラムとして機能します。 MQIPT は、TCP/IP ポートで listen し、WebSphere MQ チャネルからの接続要求を受け入れます。正しい形式の要求を受信した場合、MQIPT は、さらに自分自身と宛先 WebSphere MQ キュー・マネージャー間の TCP/IP 接続を確立します。次に MQIPT は、着信接続から受信したすべてのプロトコル・パケットを宛先キュー・マネージャーに渡し、宛先キュー・マネージャーから受信したプロトコル・パケットを元の着信接続に戻します。

WebSphere MQ プロトコルへ (クライアント / サーバーまたはキュー・マネージャーからキュー・マネージャーへ) の変更は行われません。それは、どちらの側も中間の存在を直接認識していないからです。したがって、新しいバージョンの WebSphere MQ クライアント・コードやサーバー・コードは必要ありません。

MQIPT を使用するためには、宛先キュー・マネージャーのホスト名とポートではなく、MQIPT のホスト名とポートを使用するように呼び出し元チャネルを構成する必要があります。これは WebSphere MQ チャネルの CONNAME プロパティーで定義されます。 MQIPT は、着信データを読み取り、単にそのデータを宛先キュー・マネージャーに渡します。クライアント / サーバー・チャネルのユーザー ID やパスワードなどの他の構成フィールドも、同様に、宛先キュー・マネージャーに渡されます。

MQIPT を使用して、1 つまたは複数の宛先キュー・マネージャーへのアクセス権を許可することができます。この機能を働かせるためには、どのキュー・マネージャーに接続するかを MQIPT に指示するメカニズムが用意されていなければなりません。このため、MQIPT は、着信 TCP/IP ポート番号を使用して接続先のキュー・マネージャーを判別します。この操作については、次に説明します。

複数の宛先キュー・マネージャーにアクセスできるようにするには、複数の TCP/IP ポートで listen するように MQIPT を構成します。 listen する各ポートは、MQIPT「経路」を介して宛先キュー・マネージャーにマップされます。 MQIPT 管理者は、最高 100 経路まで定義することができます。これらの経路は、listen する TCP/IP ポートを宛先キュー・マネージャーのホスト名とポートに関連付けます。つまり、宛先キュー・マネージャーのホスト名 (IP アドレス) は発信元のチャネルには決して見えません。各経路は、自分が listen するポートと宛先の間に存在する複数の接続を処理することができます。この場合、これらの接続はそれぞれ独立して機能します。

MQIPT は mqipt.conf という名前の構成ファイルを使用します。このファイルにはすべての経路の定義とそれらに関連するプロパティーが含まれています。このファイルの詳細については、73 ページの『第 19 章 internet pass-thru の管理と構成』を参照してください。

MQIPT を起動すると、構成ファイルにある各経路を始動します。各経路の状況を示すメッセージがシステム・コンソールに書き出されます。経路についてメッセージ MQCPI078 が表示されると、その経路は接続要求を受け入れることができる状態になっています。

サポートされるチャネル構成

すべての WebSphere MQ チャネル・タイプがサポートされますが、構成は TCP/IP 接続に限定されます。 WebSphere MQ クライアントやキュー・マネージャーからは、MQIPT は宛先キュー・マネージャーのように見えます。チャネル構成に宛先ホストとポート番号が必要な場合は、MQIPT ホスト名とリスナー・ポート番号が指定されます。

クライアント / サーバー・チャネル

MQIPT は、着信クライアント接続要求を listen してから、それらを転送します (この場合、HTTP トンネル操作または SSL を使用して転送するか、標準の WebSphere MQ プロトコル・パケットとして転送するかのいずれかの方法をとります)。 MQIPT が HTTP トンネル操作か SSL を使用する場合は、2 番目の MQIPT との接続を使用して転送します。 HTTP トンネル操作を使用しない場合は、宛先キュー・マネージャーと見なすマシン (ただしこの場合、順にもう 1 つの MQIPT ということになることもある) との接続を使用して転送します。宛先キュー・マネージャーがクライアント接続を受け入れると、クライアントとサーバー間でパケットがリレーされます。

クラスター送信側 / 受信側チャネル

クラスター送信側チャネルから着信要求を受け取った場合、MQIPT は、キュー・マネージャーが SOCKS 化されていて、真の宛先アドレスは、SOCKS ハンドシェーク・プロセス時に取得されると想定します。 MQIPT は、クライアント接続チャネルの場合とまったく同じ方法で、その要求を次の MQIPT または宛先キュー・マネージャーに転送します。この操作には、自動定義されたクラスター送信側チャネルも使用されます。

送信側 / 受信側

MQIPT は、送信側チャネルから着信要求を受け取った場合、クライアント接続チャネルの場合とまったく同じ方法で、その要求を次の MQIPT または宛先キュー・マネージャーに転送します。宛先キュー・マネージャーは、その着信要求を検証し、該当する場合は、受信側チャネルを開始します。送信側チャネルと受信側チャネル間のすべての通信 (セキュリティー・フローを含む) がリレーされます。

要求発行者 / 送信側

この組み合わせは、上記のタイプと同じ方法で処理されます。接続要求の検証は、宛先キュー・マネージャーのサーバー・チャネルによって行われます。

要求発行者 / 送信側

2 つのキュー・マネージャーを相互に直接接続することは許可されていない場合で、どちらも MQIPT に接続することができ、かつそれからの接続を受け入れることができる場合は、「コールバック」構成が役に立つことがあります。

送信側 / 要求発行者および送信側 / 受信側

これらは、送信側 / 受信側構成と同じような方法で MQIPT によって処理されます。

第 3 章 HTTP サポート

転送するデータ・パケットを HTTP 要求としてエンコードするように MQIPT を構成することができます。 MQIPT は、チャンク操作を伴う HTTP トンネル操作も、チャンク操作を伴わない HTTP トンネル操作もサポートします。

今日の WebSphere MQ チャネルは HTTP 要求を受け入れないため、HTTP 要求を受信してそれを通常の WebSphere MQ プロトコル・パケットに変換するために、2 番目の MQIPT が必要になります。 2 番目の MQIPT は、HTTP ヘッダーを取り去り、着信パケットを元の標準 WebSphere MQ プロトコル・パケットに変換してから、それを宛先キュー・マネージャーに渡します。

チャンク操作を伴わない HTTP トンネル操作を使用する場合は、HTTP 応答が各 HTTP 要求ごとに最初の MQIPT に戻されます。この応答は、宛先キュー・マネージャーからの応答であったりダミーの確認通知であったりします。どちらかの WebSphere MQ システムが一連の WebSphere MQ プロトコル・パケットを送信しなければならない場合 (大きなメッセージを転送するときに発生する) は、いくつかの HTTP 要求 / 応答のペアを使用してデータを転送します。これを行うために、MQIPT は追加の要求または応答のフローを挿入します。

チャンク操作を伴う HTTP トンネル操作を使用する場合は、最初のパケットだけを HTTP ヘッダーでラップします。中間のパケットと最後のパケットにはチャンク・ヘッダーがありません。このため、2 番目の MQIPT からのダミーの確認通知を待機する必要がないため、チャンク操作を伴わない HTTP トンネル操作の場合のパフォーマンスよりも多少高いパフォーマンスが得られます。

HTTP を 2 つの MQIPT 間で使用する場合は、HTTP 要求や応答が流れる TCP/IP 接続はパーシスタントになり、メッセージ・チャネルの存続時間中、オープン状態になっています。 MQIPT は、要求 / 応答ペア間の TCP/IP 接続をクローズしないでください。

2 つの MQIPT が HTTP を介して通信している場合、HTTP 要求が長い時間、未解決のままになっていることがあります。たとえば、要求発行者 / サーバー・チャネルにおいて、サーバー・サイドが、新規のメッセージが伝送キューに到着するのを待機している場合です。 WebSphere MQ チャネル・プロトコルは「ハートビート」メカニズムを備えています。この場合は、待機している側が定期的にハートビート・メッセージを相手側に送信する必要があり (デフォルトのハートビート間隔は 5 分)、MQIPT はこのハートビートを HTTP 応答として使用します。一部のファイアウォールでタイムアウトの問題が発生するのを避けようとして、このチャネル・ハートビートを使用不可にしたり、それを過度に高い値に設定したりしないでください。

HTTP プロキシーによっては、持続接続を制御するための独自のプロパティー (たとえば、1 つの持続接続で発行可能な要求の数) を備えているものがあります。HTTP プロキシーは HTTP 1.1 プロトコルもサポートする必要があります。 IBM WebSphere Caching Proxy を使用するときは、以下のプロパティーをリセットする必要があります。

- 高い値 (たとえば、5000) に設定された MaxPersistenceRequest
- 高い値 (たとえば、12 時間) に設定された PersistentTimeout
- オンに設定された ProxyPersistence

HTTP の使用の例については、 108 ページの『HTTP プロキシー構成』を参照してください。

HTTPS

HTTP 接続での HTTPS の使用は、クライアント接続を発行する MQIPT 上で HTTPS および SSLClient の経路プロパティーを使用可能にすることによって可能になります。 MQIPT には、ターゲット HTTP プロキシー/サーバーを認証するために使用されるトラステッド CA 証明書へのアクセス権限が必要になります。 SSLClientCAKeyring プロパティーを使用して、トラステッド CA 証明書が入っている鍵リング・ファイルを定義できます。

HTTPS の共通のセットアップは、ローカル HTTP プロキシーを使用してファイアウォールを通り抜け、リモート HTTP サーバー (または別のプロキシー) に接続します。これが次にリモート MQIPT に接続します。接続のサーバー・サイドにあるこの MQIPT は、接続要求が任意の通常の HTTP 接続として扱われるため、特定の構成を必要としません。

MQIPT は、HTTPProxy プロパティーおよび HTTPServer プロパティーを使用して、ローカルおよびリモート・プロキシーを見分けます。 HTTPProxy は、ローカル HTTP プロキシーであり、HTTPServer はリモート・サーバー (またはプロキシー) と見なされます。

HTTPS 接続は、通常 HTTP プロキシー/サーバー上のリスナー・ポート・アドレス 443 に対して行われますが、HTTPProxyPort および HTTPServerPort を使用してこのデフォルトをオーバーライドできます。 HTTPS の使用の例については、 124 ページの『HTTPS 構成』を参照してください。

サーブレット

非分散アプリケーションとして Application Server にデプロイ可能な MQIPT のサーブレット・バージョン (MQIPTServlet と呼ばれる) が使用可能になりました。このバージョンは、通常の MQIPT と同じように機能しますが、1 つの経路しか持っていない場合と同様に作動します。 WebSphere MQ チャネルを開始するための着信接続要求は、MQIPTServlet のインスタンスによって処理され、それぞれのインスタンスは、宛先キュー・マネージャーとの持続接続を維持しています。後続のデータ・フローは、最初の接続要求時に作成されたセッション ID を使って、同じチャネルで維持されます。

MQIPTServlet.war という Web アプリケーション・アーカイブ・ファイルは、Web サブディレクトリーに入っています。この war は、Application Server にインポート/デプロイする必要があります。このサーブレットをインポートするときにコンテキスト名を指定する必要がある場合、デフォルトの UriName プロパティーをオーバーライドして新規のコンテキスト名を含むようにする必要があります。詳細については、 98 ページの『UriName』を参照してください。

MQIPTServlet を構成するには、web.xml ファイル (Application Server の WEB-INF サブディレクトリーに入っている) にプロパティーを設定します。 MQIPTServlet では、既存の MQIPT プロパティーのサブセットのみが適用できます。以下のプロパティーは MQIPTServlet で使用できます。

- ClientAccess
- ConnectionLog
- MaxLogFileSize
- QMgrAccess
- Trace

接続ログやトレース・ファイルは、LogDir という新規のプロパティーで定義されているディレクトリーに書き込まれます。 MQIPTServlet を始動する前にこのプロパティーを定義することをお勧めします。

MQIPTServlet で使用するリソースの量を制御するために、Application Server の一部のプロパティーを変更する必要がある場合があります。それぞれの Application Server には、構成データを管理する独自の方法がありますが、これは通常 GUI、Web インターフェースを使用するか、または構成ファイルを編集することによって行われます。変更について考慮が必要なプロパティーは、Application Server 内のアクティブ・セッションの最大数またはサーブレットのインスタンスの数です。これは、クライアント接続の数を制御しますが、MQIPT で使用される MaxConnectionThreads プロパティーと同じです。

変更が必要となる可能性のある他のプロパティーは、タイムアウト値、持続接続がサポートされているかどうか、および持続接続でどれだけ多くの要求を出すことができるかに関連します。 MQIPTServlet はターゲット・キュー・マネージャーへの持続接続に依存するため、このプロパティーを使用可能にする必要があります。その他のプロパティーは、増やす必要がある可能性がありますが、そのデフォルト値および使用される WebSphere MQ 接続のタイプに依存します。 WebSphere MQ クライアント接続は、通常は短命であるため、デフォルト値を使用することがかなり安全です。キュー・マネージャーからキュー・マネージャーへの接続は、不確定時間続きますが、その場合タイムアウト値の一部および持続接続で出すことができる要求の数を適切に増やすことをお勧めします。

また、デフォルト値が 30 分として web.xml ファイルに定義されるセッション・タイムアウト・プロパティーがあります。このプロパティーを使用してクライアントのアクティビティーの停止を制御でき、指定した時間アクティビティーが検出されなかった場合セッションをクローズします。

クライアントと MQIPTServlet との間のリンクには少なくとも 1 つの MQIPT がなければなりません。 MQIPTServlet に接続する MQIPT で ServletClient プロパティーを使用可能にする必要があり、HTTPServer プロパティーは直接 Application Server を指すか、または Application Server の入力となる HTTP サーバーを指すことができます。

MQIPTServlet が正常に始動したかどうかをテストするには、Web ブラウザーを立ち上げて、次のような URL 名を入力します。

```
http://localhost:80/MQIPTServlet
```

肯定応答がブラウザーで検出されます。

MQIPTServlet は、IBM WebSphere Application Server 5.0 (IBM HTTP Server を使用した場合と使用しない場合の両方)、Tomcat 3.3 および Tomcat 4.0 を使用してテスト済みです。 MQIPTServlet は、Java 1.4 を必要とせず、Application Server によってインプリメントされた Java のレベルを使用します。

サーブレットの使用方法の例については、121 ページの『MQIPT サーブレットの構成』を参照してください。

第 4 章 Socks サポート

Socks プロキシーは、ファイアウォールを経由する出口の制御点として使用されるネットワーク・サービスです。ファイアウォール内部で実行する、Socks を使用可能にするアプリケーションは、Socks プロキシーを使用してリモート・アプリケーションに接続できます。

MQIPT は、SocksServer プロパティーを使用可能にすることによって、Socks プロキシーとして機能でき、それによって Socks を使用可能にする WMQ アプリケーションは MQIPT を介してリモート WMQ キュー・マネージャーに接続できるようになります。このフィーチャーを使用する場合は、ターゲット宛先と宛先ポート・アドレスが SOCKS ハンドシェーク・プロセス中に取得されるため、Destination および DestinationPort 経路プロパティーはオーバーライドされます。これは WMQ クラスター化をサポートするための重要なフィーチャーです。詳細については、以下を参照してください。

MQIPT は、Socks が使用可能になっていない、ローカル WMQ アプリケーションのために Socks クライアントとしても機能できます。これは、Socks プロキシーを介してアウトバウンド接続のみを可能にするファイアウォールを使用している場合、有用です。各 MQIPT 経路は、異なる Socks プロキシーと通信できるように構成できます。

SOCKS の使用方法の例については、116 ページの『SOCKS プロキシーの構成』を参照してください。

クラスター化

WebSphere MQ クラスターを MQIPT で使用することができます。そのためには、インターネットを拡張するクラスターに各キュー・マネージャーを SOCKS 化し、MQIPT を SOCKS プロキシーとして機能させます。キュー・マネージャーをクラスターに構成するには非常に多くの方法があるため、以下の説明は、「*WebSphere* キュー・マネージャー・クラスター」(SD88-7165) に示されているタスクに基づいて行っています。次の図は、タスク『クラスターへの新規キュー・マネージャーの追加』で定義されている図を拡張したものです。 NEWYORK と CHICAGO は、HOME というクラスターに入っていて、この両者はフル・リポジトリーを保持しています。 NEWYORK、LONDON、および PARIS は、INVENTORY という別のクラスターに入っています。 CHICAGO は、MQIPT を必要としないクラスターに入っているので、SOCKS 化する必要がないことに注意してください。

INVENTORY クラスター内の各キュー・マネージャーは、MQIPT では事実上「非表示」になっています。キュー・マネージャーはすでに SOCKS 化されているため、クラスター送信側チャネルを開始すると、SOCKS プロキシーとして機能する MQIPT を使用して、要求がその宛先に送信されます。通常は、クラスター受信側チャネル上の CONNAME を使用してローカル・キュー・マネージャーを識別しますが、MQIPT と一緒に使用する場合、CONNAME は、ローカル MQIPT とその着信

リスナー・ポートを識別する必要があります。次の図では、すべての着信リスナー・ポート・アドレスが 1414 であり、発信リスナー・ポート・アドレスが 1415 です。

SOCKS 化されたキュー・マネージャーを実行するには 2 つの方法があります。 1 つの方法は、キュー・マネージャーが稼働するマシン全体の SOCKS 化です。もう 1 つの方法は、キュー・マネージャーだけの SOCKS 化です。いずれの方法の場合も、MQIPT を SOCKS プロキシーとして使用してリモート接続だけを行うように SOCKS クライアントを構成し、ユーザー認証を使用不可にする必要があります。 SOCKS サポートを可能にする多数の製品が販売されています。 SOCKS V5 プロトコルをサポートする製品を選択する必要があります。

クラスター・ネットワークの構成方法の例については、 127 ページの『MQIPT クラスター化サポートの構成』を参照してください。

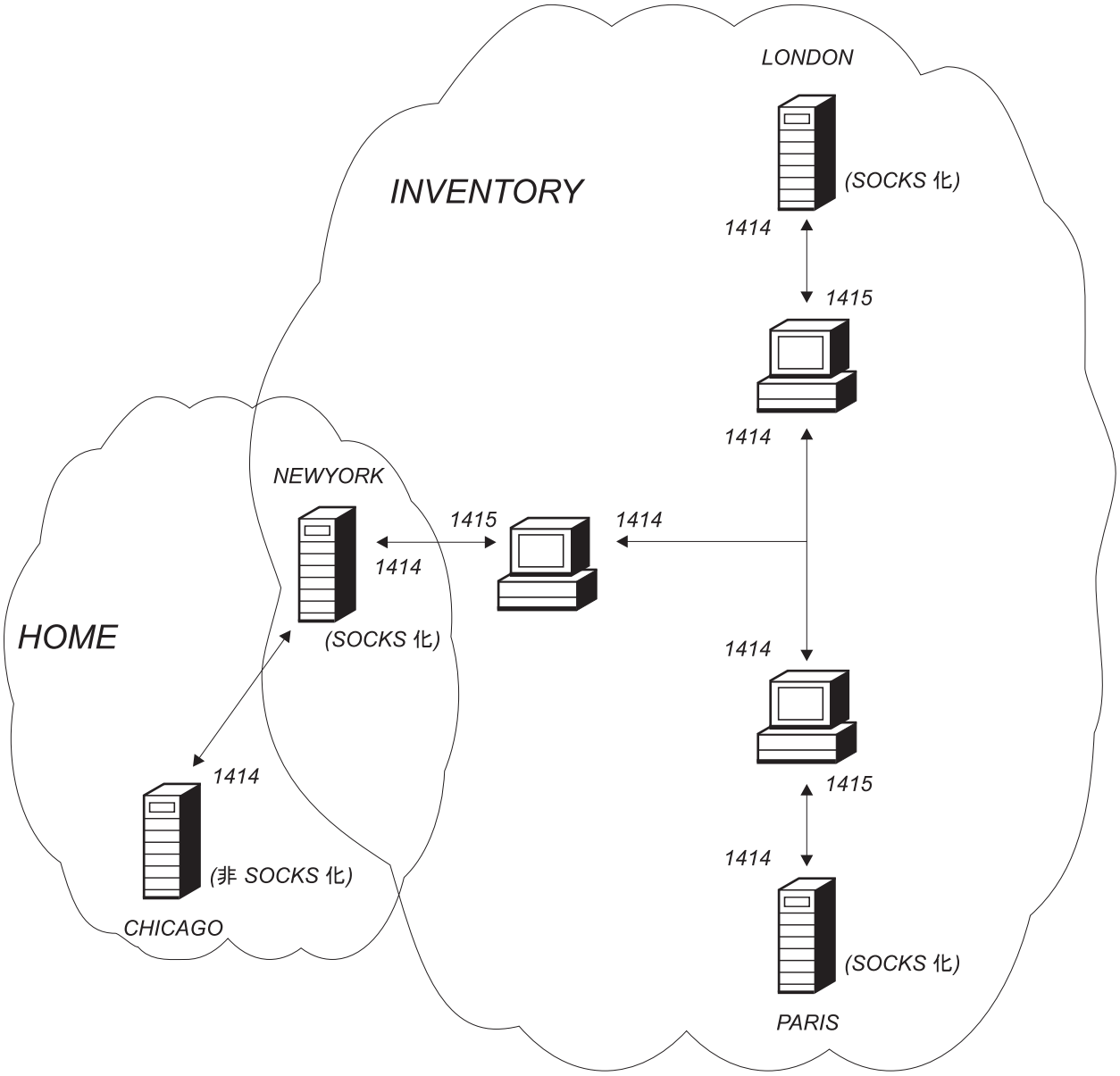

図 6. MQIPT クラスター化のサポート

第 5 章 SSL 概説およびサポート

SSL プロトコルは、不安定な通信チャネルに関する接続セキュリティーを提供し、以下の保証を行います。

通信プライバシー
クライアントとサーバー間で交換するデータを暗号化することにより、接続を専用にする (たとえば、当事者しかデータを理解できないようにする) ことができます。こうすれば、クレジット・カード番号などの専用情報を安全に転送できるようになります。

通信の保全性
接続は信頼できます。メッセージの移送には、安全なハッシュ機能に基づいたメッセージ保全性チェックが伴います。

認証 クライアントはサーバーを認証でき、認証されたサーバーはクライアントを認証することができます。つまり、情報は、意図された当事者間でのみ交換されることが保証されます。認証メカニズムは、ディジタル証明書 (X.509v3 証明書) の交換に基づいています。

SSL プロトコルは、通信者の認証にさまざまなディジタル署名アルゴリズムを使用することができます。 SSL で使用する暗号化、データ機密性のための暗号化、およびメッセージ保全性のためのセキュア・ハッシュは、クライアントとサーバー間で秘密鍵を共用することを前提にしています。 SSL は、秘密鍵の共用を可能にするさまざまな鍵交換メカニズムを備えています。 SSL は、暗号化やハッシュのための各種のアルゴリズムを使用することができます。各種の暗号アルゴリズムがサポートされており、ユーザーは、SSL 暗号スイートを使用してそれらの暗号アルゴリズムを指定します。以下の暗号スイートがサポートされています。

| SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
| SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
| SSL_DH_anon_WITH_AES_128_CBC_SHA
| SSL_DH_anon_WITH_AES_256_CBC_SHA
| SSL_DH_anon_WITH_DES_CBC_SHA
| SSL_DH_anon_WITH_RC4_40_MD5
| SSL_DH_anon_WITH_RC4_128_MD5
| SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
| SSL_DHE_DSS_WITH_3DES__EDE_CBC_SHA
| SSL_DHE_DSS_WITH_AES_128_CBC_SHA
| SSL_DHE_DSS_WITH_AES_256_CBC_SHA
| SSL_DHE_DSS_WITH_DES_CBC_SHA
| SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
| SSL_DHE_RSA_WITH_3DES__EDE_CBC_SHA
| SSL_DHE_RSA_WITH_AES_128_CBC_SHA
| SSL_DHE_RSA_WITH_AES_256_CBC_SHA
| SSL_DHE_RSA_WITH_DES_CBC_SHA
| SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
| SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5#

| SSL_RSA_EXPORT_WITH_RC4_40_MD5
| SSL_RSA_WITH_3DES_EDE_CBC_SHA
| SSL_RSA_WITH_AES_128_CBC_SHA
| SSL_RSA_WITH_AES_256_CBC_SHA
| SSL_RSA_WITH_DES_CBC_SHA
| SSL_RSA_WITH_NULL_MD5
| SSL_RSA_WITH_NULL_SHA
| SSL_RSA_WITH_RC4_128_MD5
| SSL_RSA_WITH_RC4_128_SHA

SSL ハンドシェーク

SSL ハンドシェーク・プロセスは、SSL クライアントとサーバー間で初期接続要求が出されたときに行われ、そのときには、暗号スイートの認証とネゴシエーションが実行されます。

上記の SSL 暗号スイートは、匿名暗号スイートを除き、すべてサーバー認証が必要であり、クライアント認証が可能になっています。したがって、クライアント認証を要求するようにサーバーを構成することができます。 SSL における通信ピア認証は、公開鍵暗号と X.509v3 ディジタル証明書に基づいて行われます。 SSL プロトコルでの認証を必要とするサイトには、秘密鍵とディジタル証明書 (対応する公開鍵が含まれている) が必要になるほか、そのサイトのアイデンティティー、証明書の有効期間などに関する情報も必要になります。証明書は認証局によって署名されており、このような権限の付いた証明書は署名者証明書と呼ばれます。 1 つまたは複数の署名者証明書が付随している証明書は証明書チェーンを形成します。証明書チェーンの特徴として、最初の証明書 (サイト証明書) から始めて、チェーン内の各証明書の署名を検査するために、その次の署名者証明書に含まれている公開鍵を使用できるという点が挙げられます。

サーバー認証が必要な安全な接続を確立する場合、サーバーは、自分のアイデンティティーを証明するための証明書チェーンをクライアントに送ります。 SSL クライアントは、サーバーを認証できる (たとえば、サーバーのサイト証明書の署名が正しいことを証明できる) 場合にのみ、サーバーへの接続の確立を続行します。その署名が正しいことを証明するためには、SSL クライアントが、サーバー・サイトそのもの、またはサーバーから提供された証明書チェーンの中の少なくとも 1 人の署名者を信用する必要があります。信用できるサイトや署名者の証明書は、この検証を行うためにクライアント・サイドで保持しておく必要があります。

SSL クライアントは、サイト証明書から始めて、サーバーの証明書チェーンを調べます。サイト証明書が信用できるサイトまたは署名者証明書のリポジトリーに入っている場合、または証明書チェーン内の署名者証明書を信用できる署名者証明書のリポジトリーに基づいて検証できる場合は、サイト証明書の署名が有効であると見なします。後者の場合、SSL クライアントは、信用できる署名者証明書から始めてサーバーのサイト証明書に至るまで、証明書チェーンが本当に正しく署名されているかどうか調べます。このプロセスに係る含まれ証明書は、すべて形式の正確さと日付の妥当性の点からも調べられます。これらのどの 1 つの検査にでもパスしないと、サーバーへの接続は拒否されます。サーバー証明書の検査が済めば、クライア

ントは、その証明書に組み込まれた公開鍵を次の SSL プロトコル・ステップで使用します。 SSL 接続を確立できるのは、サーバーが実際に、対応する秘密鍵を所有している場合のみです。

クライアント認証もこれと同じ手順を踏みます。つまり、SSL サーバーがクライアント認証を必要とする場合、クライアントは、証明書チェーンのアイデンティティーを証明するためにそれをサーバーに送り、サーバーは、信用できるサイトと署名者証明書のリポジトリーに基づいてそのチェーンを検証します。クライアント証明書の検証が済めば、サーバーは、その証明書に組み込まれた公開鍵を次の SSL プロトコル・ステップで使用します。 SSL 接続を確立できるのは、クライアントが実際に、対応する秘密鍵を所有している場合のみです。

SSL プロトコルそれ自体は、極めて高度な通信セキュリティーを備えています。しかし、このプロトコルは、アプリケーションから提供された情報に基づいて作動します。その情報ベースも安全に維持されている場合にのみ、安全な通信という全体的なゴールを達成できます。たとえば、信用できるサイトおよび署名者証明書のリポジトリーが危険にさらされることになれば、非常に安全性の低い通信相手との確実な接続を確立することになる可能性があります。

WebSphere MQ internet pass-thru および SSL

SSL V3.0 は、Public Key Cryptography Standards (PKCS) #12 トークンを使用してインプリメントされています。このトークンは、 X509.V3 証明書が含まれた鍵リング・ファイル (.p12 または .pfx のファイル・タイプを持つ) に保管されています。鍵リング・ファイルには、証明書取り消しリスト (CRL) と権限取り消しリスト (ARL) も収納できます。WebSphere MQ internet pass-thru は、IBM Secure Socket Lite (SSLite) パッケージを使用します。

WebSphere MQ internet pass-thru は、接続がどちらの側から始まるかによって、SSL クライアントまたは SSL サーバーとして機能します。クライアントは接続を開始し、サーバーは接続要求を受け入れます。 1 つの WebSphere MQ internet pass-thru 経路がクライアントとサーバーの両方として機能することは可能ですが、その場合は、パフォーマンス上の理由から、SSL Proxy Mode フィーチャーを使用することをお勧めします。各 WebSphere MQ internet pass-thru 経路は、それぞれ独自の SSL プロパティー・セットを使用して独立して構成することができます。詳細については、 84 ページの『経路セクション参照情報』を参照してください。

トラストの設定

鍵リング・ファイルには、署名者証明書または署名者証明書のチェーンが組み込まれている個人用証明書が入っています。接続を行なうときに認証を使用可能にするには、証明書にトラストの設定が必要です。トラストのレベルには、次の 2 つがあります。

ピアとしてのトラスト
この証明書だけが信用でき、この証明書によって署名された証明書はどれも信用できないことを意味します。

認証局 (CA) としてのトラスト
この証明書によって署名された証明書はすべて信用できることを意味します。

SSLServerKeyRing プロパティーによって識別された、SSL サーバー・サイドの鍵リング・ファイルには、その個人用証明書が入っていなければなりません。

SSLClientCAKeyRing プロパティーにより識別される SSL クライアント・サイドの鍵リング・ファイルには、トラステッド CA 証明書のリストを含める必要があります。このリストは、サーバーから送信された証明書を認証するために使用されます。

クライアント認証も必要な場合には、SSLServerAskClientAuth プロパティーをサーバー・サイドで使用可能にし、SSLClientKeyRing プロパティーによって識別されるクライアント・サイドの鍵リング・ファイルにはその個人用証明書が入っていなければなりません。SSLServerCAKeyRing プロパティーによって識別されるサーバー・サイドの鍵リング・ファイルには、トラステッド CA 証明書のリストを含める必要があります。このリストは、クライアントを認証するために使用されます。

トラステッド CA によって署名された証明書を使用する代わりに、自己署名証明書を使用できます。これらの例は、ssl サブディレクトリーにある、MQIPT に付属して提供されているサンプル鍵リング・ファイル (sslSample.pfx および sslCAdefault.pfx) にあります。

これらの鍵リング・ファイルに保管されている PKCS#12 トークンのどちらかをオープンするには、パスワード `mqiptV1.3` を使用する必要があります。

SSL 証明書と鍵リング・ファイルの管理に使用される、KeyMan という名前のユーティリティーは、ssl サブディレクトリーにあります。インストールの指示と詳細については、25 ページの『KeyMan』を参照してください。

すべての鍵リング・ファイルとパスワード・ファイルを保護するために、オペレーティング・システムのセキュリティー・フィーチャーを使用してそれらへの無許可アクセスを防止する必要があります。

SSL のテスト

99 ページの『第 20 章 internet pass-thru の使用開始』では、SSL 接続のテストに使用できるタスクについて説明しています。

証明書や証明書管理テクノロジーが、以下に示すような多くのベンダーから提供されています。

- RSA Security (www.rsasecurity.com)
- Entrust Technologies (www.entrust.com)
- Verisign (www.verisign.com)

SSL エラー・メッセージ

いずれかの SSL メソッド呼び出しで無効なパラメーター値が使用された場合や、間違ったデータが SSL プロトコルに提供された場合は、以下のようなエラー・コードが SSLRuntimeException に表示されることがあります。

表 1. SSLRuntimeException エラー・メッセージ

ID	説明

表 1. SSLRuntimeException エラー・メッセージ (続き)

1	メソッドの使用法が間違っている、あるいは 1 つまたは複数の入力パラメーターが範囲外である
2	提供されたデータを処理できない
3	提供されたデータの署名を検証できない
10	署名者証明書のサブジェクト名が、その証明書の発行者名と一致しない
11	証明書のタイプがサポートされていない
12	有効期間前の証明書が使用されている
13	証明書の有効期限が切れている
14	証明書の署名を検証できない
15	証明書を使用できない
20	クライアントによって提示されたすべての暗号スイートがサーバーでサポートされていない
21	クライアントによって提示された圧縮方法がサーバーでサポートされていない
22	証明書が使用可能でない
23	アルゴリズムまたは形式のタイプがサポートされていない
24	古くなった情報が拒否された
25	証明書が失効した
26	CRL のセットが不完全である (一部のデルタ CRL が欠落している)
27	証明する名前がすでに存在している
28	証明される公開鍵がすでに存在している
29	一部のシリアル番号または鍵 (証明書、CRL) が間違っている
30	許可が失敗した

SSL ハンドシェーク・プロトコルの実行が終了すると、SSLException が throw されます。

表 2. SSLException エラー・メッセージ

ID	説明
3	SSLContext に定義された接続タイムアウトが時間切れになったが、ピアからの応答がない
4	SSL ハンドシェーク時に接続がピアによって打ち切られたが、エラー表示が出ない
10	予期しないメッセージを受け取った
20	無効なレコード MAC を含むメッセージを受け取った
30	解凍に失敗した
40	ハンドシェークに失敗した
41	ピアから証明書が送信されない
42	無効な証明書を受け取った
43	サポートされていない証明書を受け取った
44	失効した証明書を受け取った
45	有効期限が切れた証明書を受け取った
46	不明な証明書を受け取った
47	正しくないパラメーターを検出した

LDAP および CRL

WebSphere internet pass-thru は、ディジタル証明書で証明書取り消しリスト (CRL) 認証を実行するために Lightweight Directory Access Protocol (LDAP) サーバーの使用をサポートします。LDAP サポートは、同じ LDAP サーバーが WebSphere MQ と MQIPT の両方に使用される可能性があるため、ベースの WebSphere MQ の場合と同様にインプリメントされています。 WebSphere MQ と一緒に LDAP サーバーを使用する場合の詳細は、マニュアル「WebSphere MQ セキュリティー バージョン 5.3」SC88-9231-00、第 15 章に記載されています。参照のためそのマニュアルからの抜粋を以下に記載します。

SSL ハンドシェーク時に、通信のパートナーはディジタル証明書と相互に認証します。認証には、受け取った証明書が引き続き信頼され得ることのチェックを含めることができます。認証局 (CA) は、以下を含む、さまざまな理由から証明書を取り消します。

- 所有者が異なる組織に移動した
- 秘密鍵が秘密でなくなっている

CA は、取り消した個人用証明書を証明書の失効取り消しリスト (CRL) に公開します。取り消された CA 証明書は、権限取り消しリスト (ARL) に公開します。本章でこれ以降の CRL の参照は、ARL にも適用されます。

市販の LDAP ディレクトリー・サーバーがいくつかあります。WebSphere internet pass-thru は、IBM Directory Server を使用してテストされています。`http://www.ibm.com/software/network/directory/server` を参照してください。LDAP サーバーのインストールおよび管理の指示は、インストール済み製品に付属の資料に記載されています。

CRL と ARL の管理の詳細は、「WebSphere MQ セキュリティー バージョン 5.3」SC88-9231-00 に記載されています。

MQIPT は、各経路で最高 2 つの LDAP サーバーをサポートします。最初の LDAP サーバーはメイン・サーバーとして扱われ、2 番目の LDAP サーバーはバックアップ・サーバーと見なされ、メイン・サーバーが通信できない場合のみ使用されます。バックアップ・サーバーは、メイン・サーバーのミラー・イメージにする必要があります。

LDAP サーバーに保管されている情報へのアクセスは、ユーザー ID とパスワードによって保護することができます。この場合、LDAP*Userid と LDAP*Password のプロパティーを使用できます。

MQIPT が鍵リング・ファイルから PKCS#12 トークンをロードするとき、CA 証明書の CRL 妥当性がチェックされます。CA 証明書に CRL が付加されている場合、その有効期限が切れているかどうかがチェックされ、有効期限が切れている場合には新しい CRL が LDAP サーバーから取り出されます。取り出された CRL があれば、それは現行のトークンにロードされ、その CA 証明書に付加されます。更新済みトークンは、鍵リング・ファイルに保管できます (84 ページの『経路セクション参照情報』の LDAPSaveCRL プロパティーを参照)。

照会がメイン LDAP サーバーに送信されたとき、所定の CA に一致するエントリーがない場合には、その CA の CRL がないものと想定します。バックアップ・サーバーは使用されません。しかし、メイン LDAP サーバーと通信できないか、または所定の時間フレーム内に戻されない場合には、バックアップ・サーバーが使用されます。バックアップ・サーバーでエラーが発生した場合、クライアント接続は強制終了されます。このアクションは、プロパティー LDAPIgnoreErrors を true に設定することによってオーバーライドできます。

重要

LDAPIgnoreErrors プロパティーを使用可能にした場合、取り消された証明書を使用して SSL 接続を行うこともできます。

LDAP クライアント・モデルは、「com.sun.jndi.ldap.LdapCtxFactory」インプリメンテーションに基づいています。MQIPT によって取り出された CRL があれば、それはキャッシュに保管され、その経路上のすべての接続によって共用されます。

キャッシュ内の CRL の有効期限が切れた場合、その CRL はキャッシュから除去されて、新しい CRL が LDAP サーバーから取り出されます。新しい CRL が使用可能でない場合、接続は引き続き拒否されます。

LDAP サーバーから取り出された CRL の有効期限もチェックされ、警告システム・コンソール・メッセージが表示されます (MQCPW001)。有効期限が切れた CRL は引き続きシステムにロードされますが、その CRL を参照する接続要求はすべて拒否されます。LDAP サーバーにある有効期限が切れた CRL は現行のものと置き換える必要があります。

LDAPCacheTimeout プロパティーを使用して CRL キャッシュをクリアする頻度を制御できます。デフォルト値は 1 日です。この値を 0 に設定すると、経路を再始動するまでキャッシュ・エントリーはクリアされないことを意味します。

有効期限が切れた CRL は、鍵リング・ファイルまたは LDAP サーバーに保管できます。新しい CRL が発行されていない場合、それ以降の接続要求は拒否されます。IgnoreExpiredCRLs プロパティーを使用可能にすることによって、有効期限が切れた CRL を無視できます。

重要

IgnoreExpiredCRLs プロパティーを使用可能にした場合、取り消された証明書を使用して SSL 接続を行うこともできます。

拡張暗号化標準

拡張暗号化標準 (AES) は、注意が必要な (機密扱いでない) 情報を保護するために、米国政府機関によって使用される暗号アルゴリズムを指定する、新しい連邦情報処理標準 (FIPS) になります。米国連邦情報・技術局 (NIST) も AES は、米国政府以外の組織、研究機関、および個人 (および場合によっては、米国以外での) によって自発的に広く使用されると予想しています。

鍵リング・ファイルからの証明書の選択

複数の個人用証明書が同じ鍵リング・ファイルに保管される可能性があります。SSLClientSite* プロパティーをクライアント・サイドで使用することにより、認証のためにサーバーに送信される証明書を選択できます。また SSLServerSite* プロパティーをサーバー・サイドで使用することにより、認証のためにクライアントに送信される証明書を選択できます。

これらのプロパティーを使用すれば、Distinguish Name (公開鍵持ち主情報 (DN)) に基づいて、証明書を選択できます。または、代わりに、SSLServerSiteLabel プロパティーと SSLClientSiteLabel プロパティーを使用して、証明書ラベルを使用して証明書を選択することができます。

鍵リング・ファイルの暗号化

鍵リング・ファイルをオープンするために使用されるパスワードは、mqiptPW スクリプトを使用して暗号化できます。暗号化されたパスワードは、ファイルに保管され、以下のプロパティーのいずれからでも使用できます。
SSLClientKeyRingPW、SSLClientCAKeyRingPW、SSLServerKeyRingPW および SSLServerCAKeyRingPW

コマンド形式:

```
mqiptPW <password> <file name> <-replace>
```

ここで、

password
所定の鍵リング・ファイルをオープンするために必要な平文のパスワード

file name
作成されるパスワード・ファイルの名前

replace
`<file name>` がある場合に、それをオーバーライトするのに必要なオプション

パスワードには、スペース文字 (″ ″) を含めることができますが、これが受け入れられるためにはパスワード・ストリング全体を引用符で囲まなければなりません。パスワードの長さまたは形式には制限がありません。

注: WebSphere Internet pass-thru の前のレベルからマイグレーションをしたユーザーは、平文のパスワードが入っている現在のパスワード・ファイルを暗号化されたパスワード・ファイルのコピーで置き換える必要があります。

キー管理ユーティリティー (たとえば、KeyMan) を使用してサンプル鍵リング・ファイルのどちらかをオープンするためには、パスワード mqiptV1.3 を使用する必要があります。

KeyMan

KeyMan というスタンドアロン・ユーティリティーが WebSphere Internet pass-thru と同梱で出荷されるため、SSL 証明書と鍵リング・ファイルの管理が可能になりました。KeyMan の ZIP は ssl サブディレクトリーに入っています。 KeyMan をインストールするには、このファイルを一時ディレクトリーに UNZIP して、README.txt ファイルに入っている指示を実行します。 KeyMan には多くのフィーチャーが含まれていますが、このセクションでは、テスト証明書の作成と、PKCS#12 トークンが入っている鍵リング・ファイルの管理についてだけ説明します。

KeyMan は、公開鍵インフラストラクチャー (Public Key Infrastructure - PKI) のクライアント・サイドの管理ツールです。 KeyMan は、鍵、証明書、証明書取り消しリスト (CRL)、およびこれらの各アイテムの保管や検索を行うためのそれぞれのリポジトリーを管理します。証明書の全ライフ・サイクルおよびがユーザー証明書のプロセスがサポートされます。

KeyMan は、鍵、証明書、および取り消しリストの集合が入ったリポジトリーを管理します。リポジトリーはトークンと呼ばれます。トークンは、特定のアプリケーション (たとえば、WebSphere Internet pass-thru) のトラスト設定から構成されています。通常、トークンには、ユーザーを他のサイトに認証するための秘密鍵とそれに関連する証明書チェーンが含まれています。さらに、トークンは、信用できる通信相手と認証局 (CA) の証明書も保持しています。

サポートされるトークンのタイプ

KeyMan は、異なるタイプのいくつかのトークンをサポートします。トークンとは、鍵、証明書、CRL、およびトラスト設定を保持するリポジトリーを指します。トークンによっては、これらのアイテム・タイプのサブセットしか保管しないものもあります。

PKCS#7 トークン

証明書のセット、および関連する CRL (オプショナル) が入っています。鍵をこのタイプのリポジトリーに保管することはできません。このリポジトリーは認証を必要としません。証明書と CRL は署名によって保護されます。ただし、相手側は特定の PKCS#7 トークンに保管されたアイテムのセットを変更することができます。このタイプのトークンは、予定されたアイテム・セットを何らかのコンテキストで定義するときに使用します。

PKCS#12 トークン

秘密鍵、証明書、および関連する CRL が入っています。これらの内容はユーザー・パスワードによって保護されます。一般公開アイテム (証明書、CRL) と専用アイテム (鍵) は、それぞれ異なる強度のアルゴリズムで保護されています。

PKCS#11 (CryptoKi) リポジトリー

PKCS#11 は、暗号トークンとのインターフェースを定義します。これらの

トークンは、鍵と証明書を保管することができます。 CRL の保管はサポートされません。トークンへのアクセスは、個人識別番号 (PIN) によって保護されています。ユーザーは、KeyMan がトークンにアクセスするために使用するトークン特有の PKCS#11 DLL を指定する必要があります。

KeyMan は、PKCS#11 番号 2.01 および 2.10 DLL をサポートします。

PKCS#7 と PKCS#12 はソフト・トークンであり、異なるメディア (たとえば、ファイル、URI、クリップボードなど) から検索することができます。

KeyMan は、不明な形式を持つデータから PKCS#7 トークンを構成できる特殊な機能を備えています。 KeyMan は、このデータをスキャンして X.509 証明書と CRL を見つけ出し、検出された証明書と CRL から PKCS#7 トークンを構成します。証明書や CRL が入った E メールを受け取った場合は、KeyMan の E メール・フォルダーを開くことができるので、KeyMan は X.509 アイテムの抜き出しを試みます。もちろん、これらのデータを元の形式で保管し直すことはできません。抜き出したデータは、PKCS#7 形式を使用してファイルに保管することができます。

サポートされている標準データ形式

KeyMan は、いくつかの標準データ形式をサポートします。それらの意味と使用法について説明します。

PKCS#7

このデータ形式は、証明書と CRL の集合です。 PKCS#7 に記述されている証明書と CRL のセットは保護されません。ただし、個々の証明書と CRL は署名によって保護されます。 PKCS#7 は、予定された証明書と CRL のセットを何らかのコンテキストで定義するたびに使用されます。 Windows システムでは、PKCS#7 ファイルの標準のファイル・サフィックスは .p7r および .p7b です。

PKCS#10

PKCS#10 は認証要求メッセージを定義します。 PKCS#10 には、公開鍵と要求発行者の X.500 名に関する情報が含まれています。このメッセージは、対応する秘密鍵で署名されています。 PKCS#10 メッセージは、2 進数形式と ASCII 対応形式で生成できます。このメッセージは、認証局 (CA) へ送信する必要があります。

PKCS#12

PKCS#12 は、秘密鍵や関連する証明書のインポートとエクスポートを行うために、ブラウザーや Web サーバーによって使用されます。 KeyMan は、これらの PKCS#12 ファイルの読み取り / 書き込みを行うことができます。これらのプログラムは、PKCS#12 の非常に限られたプロファイルしか理解しませんが、KeyMan はもっと一般的な PKCS#12 ファイルを生成することができます。 KeyMan は、秘密鍵、証明書、CRL、および対応するトラスト設定を PKCS#12 ファイルに保管することができます。 PKCS#12 ファイルはパスフレーズ (パスワード) によって保護されます。通常、PKCS#12 トークンには、特定のアプリケーションのためのトラスト・ポリシーが入っています。 IBM BlueZ SSLite の場合、鍵と関連証明書チェーンはクライアント / サーバー認証に使用されます。他の証明書は、該当するトラスト設定に応じて、信用できる CA または信用できるサーバーの役目

を果たします。 Windows システムでは、PKCS#12 ファイルの標準のファイル・サフィックスは .p12 および .pfx です。

SPKAC

SignedPublicKeyAndChallenge (SPKAC) は、CA から証明書を要求するためのデータ形式です。この特定の形式は、HTML タグ <keygen> を使用するたびに Netscape によって生成されます。この形式には、署名された公開鍵と質問が入っています。 KeyMan は、このデータ形式を 2 進数形式と Base64 形式で生成することができます。

X.509 V3 証明書

KeyMan は、X.509 V3 証明書を 2 進数形式で読み取ったり、ASCII 防御形式でラップしたりできます。これらのファイルはオープンでき、また KeyMan にインポートすることもできます。また、トークンの個々の証明書をこれらの 2 つの形式で書き込むこともできます (**「Certificate Details (証明書の詳細情報)」-> 「Save Icon (保管アイコン)」**)。 Windows システムでは、X.509 証明書ファイルの標準のファイル・サフィックスは .crt、.cer、および .der です。

X.509 V2 証明書取り消しリスト (CRL)

KeyMan は、X.509 V2 CRL を 2 進数形式で読み取るか、または、ASCII 防御形式でラップすることができます。単一の CRL をオープンすることはできません。 KeyMan は、前から関連する CA 証明書が入っているトークンにだけ CRL をインポートすることができます。単一の CRL を 2 進数形式または ASCII 防御形式で書き込むことができます (**「Certificate details (証明書の詳細情報)」-> 「CRLs details (CRL の詳細情報)」 -> 「Save Icon (保管アイコン)」**)。 Windows システムでは、X.509 CRL ファイルの標準のファイル・サフィックスは .crl です。

KeyMan FAQ

暗号や関連用語に関する一般的な質問については、RSA Laboratories およびその ″Frequently Asked Questions (FAQ) About Today's Cryptography″ を参照してください。以下の FAQ では、KeyMan に関連する質問について説明します。

Netscape や Internet Explorer で生成した PKCS#12 ファイルを KeyMan で読み取ることができますか ?

Netscape ブラウザーや Internet Explorer で生成した PKCS#12 ファイルの内容を保護するパスワードを知っていれば、これらのファイルを KeyMan で読み取ることができます。

Netscape や Internet Explorer で読み取ることができる PKCS#12 ファイルを KeyMan で作成できますか ?

PKCS#12 標準では、自由にアルゴリズムを選択したり、コンテンツを調整したりできます。これらのブラウザーは、可能なすべてのオプションのうち、非常に限られたプロファイルしか受け入れません。 KeyMan は、Netscape や Internet Explorer が読み取ることができる PKCS#12 ファイルを生成することができます。 KeyMan を使用すれば PKCS#12 についてより多くのことが行えるので、これらのブラウザーが理解できないようなファイルを作成することも可能です。各ブラウザーに共通なプロファイルの場合、公開 / 専用暗号化 (**「Menu Options (メニュー・オプション)」 ->「PKCS#12 Settings (PKCS#12 の設定)」**を参照) は、それぞれ ″RC2

(40 ビット)″/″DES (168 ビット)″ になっていなければなりません。ちょうど 1 つの専用証明書が PKCS#12 トークンに入っていなければなりません。

専用証明書とはどんなものですか ?

KeyMan は、一致した鍵と証明書を検出すると、この 2 つのアイテムを専用証明書に組み入れます。つまり、どの専用証明書についても、それに対応する秘密鍵も所有することになります。証明書をトークンにインポートすると、KeyMan は、一致する秘密鍵がないか調べ、自動的にその鍵とインポートした証明書を専用証明書に組み入れます。この場合は、KeyMan からダイアログで通知されます。

CA とは何ですか、またピア証明書とは ?

トークンに入っている証明書はトラストを確立します。これらの証明書は、ユーザーが誰を信用しているかを定義しています。トラストの意味や、証明書の正確な評価は、そのトークンを使用するアプリケーションによって異なります。 KeyMan の場合は、証明書について 2 つのタイプのトラスト、すなわち CA とピアをセットアップすることができます。証明書を CA として信用する場合は、この CA によって直接または間接的に署名されたすべての証明書を暗黙に信用することになります。トラスト・レベルを「ピア」に設定すると、この証明書しか信用しないことになります。トラストは、「ピア」証明書によって署名された証明書までは拡張されません。

専用証明書でもなく、CA でもなく、ピア証明書でもないこれらの証明書は、何ですか ?

KeyMan は、各専用証明書ごとに、ルート証明書に至るまでの全チェーンを保管しようとします。これらの証明書はトラストを必要としないため、CA またはピアの証明書の中には出てきません。鍵リング「All Certificate Items (すべての証明書アイテム)」を選択した場合は、これらの証明書を見つけることができます。信用できない証明書にはアイコンがありません。

トークンとは ?

トークンとは、鍵、証明書、および CRL の集まりです。トークンは、何らかのメディア (たとえば、ファイル、URL、ハードウェアの一部など) に保管されます。トークンには、いろいろなタイプといろいろな機能があります。たとえば、ソフトウェア・トークン、ハードウェア・トークン、無保護トークン、パスワードや PIN によって保護されているトークンなど。

鍵リングとは ?

トークンは鍵リングのセットからなっています。特定の鍵リングは、特定のアイテム・セット (たとえば、同一トラスト・レベルの証明書、ユーザーが所有する秘密鍵の証明書、一致する証明書のない鍵など) を識別します。

第 6 章 Quality of Service

Quality of Service (QoS)

IBM WebSphere Edge Server は、Linux プラットフォームで Transactional Quality of Service プラグインを介して帯域幅管理ソリューションを提供します。Transactional Quality of Service (TQoS) とは、ネットワーク・ユーザーに提供される、スループットや遅延などのエレメントで表される全体的なサービスを指します。属性を設定することにより、接続を介して送信されるすべての発信データに QoS を関連付けることができます。これによりポリシー管理者は、特定のサーバーに関連するトラフィックを識別する規則や、このトラフィックに対する固有な DiffServ 制御機能を持つポリシー・アクションを定義することができます。たとえば、インストール先では、クライアント・ブラウズをサポートするサーバー・トラフィックに関連した発信トラフィックではなく、特定量の商品の販売をサポートするサーバー・トラフィックに関連した発信トラフィックを優遇するように指定するポリシーを定義することができます。さらに、管理者は、TQoS を使うことにより、対象とするサービス・レベル目標値 (接続スループット、遅延、損失率などの重要な測定値) をポリシーが提供するかどうかをモニターするために、対応するポリシーのパフォーマンス・データを収集できます。MQIPT では、Policy Agent (pagent) をインストールし、それを実行して、Quality of Service (QoS) をインプリメントするだけで済みます。

TQoS ポリシーは、ポリシー構成ファイル (`pagent.conf`) に定義されるか、または LDAP サーバーを使用して定義されます。 TQoS pagent は、ポリシー構成ファイルにアクセスするか、LDAP サーバーを使用するか、あるいはその両方を行って、TQoS ポリシー・エントリーを検索することができます。「*IBM Edge Server Administration Guide*」では、pagent について詳しく説明しています。この資料は、次の URL にあります。

`http://www.ibm.com/software/webservers/edgeserver/library.html`

このサイトから HTML をオンラインで表示することも、PDF バージョンをダウンロードすることもできます。この場合、どちらの形式を使っても、TQoS の検索を行えます。

TQoS コードは、インストールおよび管理の指示と一緒に、MQIPT と同じロケーションからダウンロードできます。`http://www.ibm.com/webspheremq/supportpacs` にある WebSphere MQ family SupportPacs サイトを参照して、「Category 3 - Product Extensions」をクリックしてください。

MQIPT には、`libmqiptqos.so` というダミー・ライブラリー (MQIPT lib サブディレクトリーに入っている) が付属して出荷されます。これを使用すると、TQoS pagent のインストールを必要とせずに、MQIPT は Linux プラットフォームで実行できます。TQoS をインストールした後で、このダミー・ライブラリーを TQoS で使用したライブラリーで置き換えることが必要となる場合があります。この作業に役立てるため、`mqiptQoS` というスクリプトが MQIPT bin サブディレクトリーにあり

ます。次のコマンドを使用してダミー・ライブラリーの名前を変更して、ソフト・リンクを実際の TQoS ランタイム・ライブラリーに定義します。

```
mqiptQoS -install
```

`mqiptQoS -remove` を使用すると、上記のアクションは逆になります。

MQIPT では、pagent をインストールし、それを実行して、Quality of Service (QoS) をインプリメントするだけで済みます。 MQIPT を使用すれば、それぞれの方向に流れるデータ用の経路にアプリケーション優先順位を設定できるため、その経路を使用するすべてのチャネルがこの影響を受けることになります。この優先順位は、MQIPT プロパティー `QosToCaller` および `QosToDest` を使用して定義され (詳細については、 84 ページの『経路セクション参照情報』を参照)、ここで使用する値は `pagent.conf` 制御ファイルの ApplicationPriority ポリシー定義と一致しなければなりません。一致するポリシーを pagent が見つけられない場合は、このデータには優先順位が割り当てられません。ポリシーに対する変更は、pagent が再始動されるまで MQIPT に反映されません。ポリシー定義の詳細については、 113 ページの『Quality of Service (QoS) の構成』を参照してください。

第 7 章 Network Dispatcher

Network Dispatcher サポート

MQIPT を IBM Network Dispatcher と一緒に使用すれば、カスタム・アドバイザーを使用できるので、多くのサーバーでの可用性とロード・バランシングを拡張することができます。このセクションでは、読者が Network Dispatcher とカスタム・アドバイザーについて詳しい知識を持っていることを前提に説明しています。

MQIPT では、2 つのカスタム・アドバイザーが提供されます。これらのカスタム・アドバイザーは、lib サブディレクトリーに入っています。カスタム・アドバイザーをインストールする場合は、「*Network Dispatcher 管理ガイド*」(GD88-7807) に示されている指示を実行してください。 図 7 は、Network Dispatcher を使用して、MQIPT のポート・アドレス 1414 をモニターする場合の例を示しています。各 MQIPT が同じ構成ファイルを持っていなければならない点に注意してください。

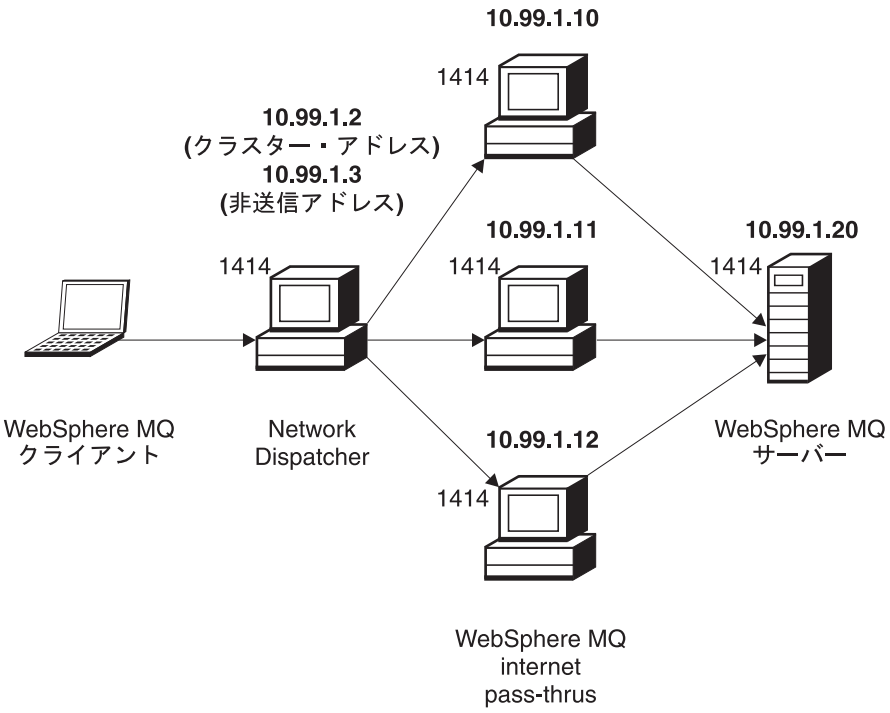

図 7. MQIPT での Network Dispatcher の使用

ポート 1414 およびロード・バランシングされたサーバー・マシンを定義するためのディスパッチャー・コンポーネントの構成方法については、「*Network Dispatcher 管理ガイド*」の第 5 章の指示を実行してください。 Administration Client のメニュー・オプション、または “ndcontrol” ライン・モード・コマンドのいずれかを使用できます。たとえば、以下のとおりです。

```
ndcontrol port add 10.99.1.2 : 1414
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.10
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.11
ndcontrol server add 10.99.1.2 : 1414 : 10.99.1.12
```

MQIPT 構成ファイルの経路定義は、以下のようになります。

```
[route]
ListenerPort=1414
Destination=10.99.1.20
DestinationPort=1414
NDAdvisor=true
```

カスタム・アドバイザーの開始 (および停止) は、コマンド行からしか行えません。たとえば、以下のとおりです。

```
ndcontrol advisor start mqipt_normal 1414
```

このコマンドは、MQIPT アドバイザーを「通常」モードで開始します。このモードの場合、ベース・アドバイザーは独自のタイミングを実行して、各 MQIPT の重み付け係数を計算します。 MQIPT アドバイザーを「置換」モードで使用するには、次の行を MQIPT 経路定義に追加します。

```
NDAdvisorReplaceMode=true
```

また、mqipt_normal カスタム・アドバイザーではなく、mqipt_replace カスタム・アドバイザーも開始する必要があります。たとえば、以下のとおりです。

```
ndcontrol advisor start mqipt_replace 1414
```

アドバイザーを使用して SSL リスナー・ポートをモニターする場合 (つまり、mqipt.conf 構成ファイルにSSLServer=true が指定されている場合) は、「トリガー」ファイルを Network Dispatcher の作業ディレクトリーに入れる必要があります。この「トリガー」ファイルには、モニターする経路に関連する特定の名前が付いています。たとえば、経路 1414 が SSLServer=true となっている場合は、mqipt1414.ssl というファイルを c:¥winnt¥system32 ディレクトリー (Windows NT 上) に入れなければなりません。詳細については、mqipt1414Sample.ssl ファイルを参照してください。

第 8 章 Java Security Manager およびセキュリティー出口

Java Security Manager

Java Security Manager のサポートは、当初、ソケット接続制御を管理するために、SSL プロキシー・モード・フィーチャーと共に使用することを目的にインプリメントされましたが、このサポートを他の任意の MQIPT フィーチャーと一緒に使用してより高いレベルのセキュリティーを提供することもできます。

MQIPT は、java.lang.SecurityManager クラスに定義されたデフォルトの Java Security Manager を使用します。 MQIPT の Java Security Manager フィーチャーは、SecurityManager グローバル・プロパティーを使用して使用可能にしたり、使用不可にしたりできます。詳細については、83 ページの『グローバル・セクション参照情報』を参照してください。

Java Security Manager は 2 つのデフォルト・ポリシー・ファイルを使用します。グローバル・システム $JREHOME/lib/security/java.policy (ここで、$JREHOME は、ユーザーの Java ランタイム環境が入っているディレクトリー) は、ホスト上の仮想計算機のすべてのインスタンスによって使用されます。 .java.policy という 2 番目のユーザー固有なポリシー・ファイルは、ユーザーのホーム・ディレクトリーに存在することができます。もう 1 つの MQIPT ポリシー・ファイルも使用できます。詳細については、83 ページの『グローバル・セクション参照情報』を参照してください。もう 1 つのポリシー・ファイルを使用する場合は、policy.allowSystemProperty プロパティーがグローバル・システム・ポリシー・ファイル (java.security) で true に設定されていることを確認します。

ポリシー・ファイルの構文は非常に複雑です。それをテキスト・エディターで変更することは可能ですが、Java から提供される policytool ユーティリティーを使用して変更することをお勧めします。 policytool ユーティリティーは、$JREHOME/bin ディレクトリーに入っていて、Java 資料に詳しく説明されています。

MQIPT ではサンプル・ポリシー・ファイル (mqiptSample.policy) が提供されていて、MQIPT を実行するにはどの許可を設定する必要があるかが分かるようになっています。誰が MQIPT に接続できるか、MQIPT は誰に接続できるかを制御するためのユーザー独自の特定要件を満たすために、追加 / 変更 / 削除が必要になるのは、java.net.SocketPermission エントリーのみです。このサンプル・ファイルでは、MQIPT がデフォルトのホーム・ディレクトリー、たとえば、c:¥Program Files¥IBM¥Websphere MQ internet pass-thru¥ にインストール済みであることを前提にしています。 MQIPT が別のロケーションにインストールされている場合は、そのことを codeBase および java.io.FilePermission 定義に反映する必要があります。

許可は通常、3 つの属性を使用して定義され、ソケット接続を制御するための値は、以下のとおりです。

クラス許可

```
java.net.SocketPermission
```

制御対象の名前

これは hostname:port 形式で構成されています。この場合、この名前の各コンポーネントは、ワイルドカードで指定することができます。ホスト名は、ドメイン・ネームであっても、IP アドレスであっても構いません。ホスト名の左端位置には、アスタリスクを指定することができます。たとえば、harry.company1.com は、以下のいずれかと一致します。

- harry
- harry.company1.com
- *.company1.com
- *
- 123.456.789 (harry.company1.com の IP アドレスであることが前提)

名前のポート・コンポーネントは、単一のポート・アドレスまたはポート・アドレスの範囲で指定できます。たとえば、以下のとおりです。

1414 ポート 1414 のみ

1414- 1414 以上のすべてのポート・アドレス

-1414 1414 以下のすべてのポート・アドレス

1-1414
1 ～ 1414 のすべてのポート・アドレス

許可されたアクション

java.net.SocketPermission で使用されるアクションは、以下のとおりです。

- 受け入れ。指定された宛先からの接続を受け入れられるようにする
- 接続。指定された宛先に接続できるようにする
- listen。指定されたポート (複数の場合もある) で接続要求を listen できるようにする
- 解決。DNS ネーム・サービスを使用してドメイン・ネームを IP アドレスに解決できるようにする

Java Security Manager の制御は、java.security.manager および java.security.policy Java システム・プロパティーを使用して行うことも可能ですが、MQIPT の制御には、SecurityManager および SecurityManagerPolicy プロパティーを使用することをお勧めします。

セキュリティー出口

重要

MQIPT は単一の JVM で実行されるため、ユーザー定義のセキュリティー出口が、以下によって MQIPT の通常の操作に悪影響を与える可能性があります。

- システム・リソースに影響を与える
- ボトルネックを生成する
- パフォーマンスを低下させる

セキュリティー出口を実稼働環境でインプリメントする前に、その影響を注意深くテストする必要があります。

セキュリティー出口の目的は、Destination 経路プロパティーによって定義されているとおり、ターゲット宛先へのアクセスを制御することです。セキュリティー出口は、クライアントから接続要求を受け取った時点で、MQIPT がターゲット宛先に接続する前に呼び出されます。初期接続プロパティーに基づいて、セキュリティー出口は接続が完了してよいかどうかを決定できます。

経路を開始すると、セキュリティー出口が呼び出されて接続要求を初期化し、処理できる状態にします。ユーザー・データがあればロードし、このデータが迅速にしかも容易にアクセスできるように準備することによって接続要求の処理に要する時間を最短にするために、初期化プロセスを使う必要があります。

それぞれの経路は独自のセキュリティー出口を持つことができます。SecurityExit プロパティーを使用してユーザー定義のセキュリティー出口を使用可能/使用不可にします。SecurityExitName プロパティーを使用してユーザー定義のセキュリティー出口のクラス名を定義します。SecurityExitPath プロパティーを使用して、クラス・ファイルが入っているディレクトリー名を定義します。このプロパティーを設定しない場合には、クラス・ファイルは出口のサブディレクトリーにあるものと想定します。また、SecurityExitPath はユーザー定義のセキュリティー出口が入っている jar ファイルの名前を定義できます。最後に、MQIPT が SecurityExitTimeout プロパティーを使用して、接続要求の確認時にセキュリティー出口からの応答をどれだけ長く待機するかを決定します。

MQIPT がユーザー定義のセキュリティー出口を呼び出すことができるようにするため、SecurityExit と呼ばれる新しいクラスが作成されました。この新しいクラスはユーザー定義のセキュリティー出口により拡張される必要があり、必要とされる機能を提供できるようにするためにそのメソッドの大部分をオーバーライドする必要があります。SecurityExitResponse オブジェクトはデータを MQIPT に戻すために使用され、このデータは MQIPT が接続要求を受け入れるか拒否するかを決めるために使用されます。SecurityExitResponse にも、経路定義プロパティーをオーバーライドするために使用される、新しい宛先と宛先ポート・アドレスを入れることができます。

セキュリティー出口のインプリメント方法を示すために、3 つのサンプル・セキュリティー出口が提供されています。SampleSecurityExit という名前の最初のサンプルは、WMQ チャネルの名前に基づいて、WebSphere MQ Queue Manager へのアクセスの制御方法を示します。これは、ストリング「MQIPT.」で始まるチャネル名のみを接続できるようにします。詳細については、145 ページの『セキュリティー出口』を参照してください。

SampleRoutingExit という名前の 2 番目のサンプルは、定義済みの WebSphere MQ サーバー (各サーバーは同じ名前および同じ属性の QM にホストとしてのサービスを提供する) のプールにクライアント接続要求を動的ルーティングできるようにします。サンプルには、サーバー名が入っている構成ファイルが含まれています。詳細については、147 ページの『セキュリティー出口のルーティング』を参照してください。

SampleOneRouteExit という名前の 3 番目のサンプルは、接続要求に使用される WMQ チャネル名から得られた WMQ QM に動的ルーティングできるようにします。サンプルには、サーバー名への QM 名のマップが入っている構成ファイルが含まれています。詳細については、150 ページの『動的 1 経路出口』を参照してください。

com.ibm.mq.ipt.SecurityExit クラス

このクラスとその共通メソッドは、一部の共通データへのアクセス権限を取得し、一部の MQIPT 初期化を行うことができるようにするために、ユーザー定義のセキュリティー出口によって拡張される必要があります。MQIPT がそれぞれのメソッドを呼びだす前に、一部のプロパティーをメソッドで使用できるようになります。このクラスに定義された該当の Get メソッドを使用してそれらの値を取り出すことができます。サポートされているメソッドの全リストについては、以下を参照してください。

メソッド

init

```
public void init () throws IPTException
```

以下のプロパティーが使用可能です。

- リスナー・ポート
- 宛先
- 宛先ポート
- バージョン

経路を開始すると、init メソッドが MQIPT によって呼び出されます。このメソッドから戻った時点で、セキュリティー出口は接続要求を確認できる状態になっている必要があります。このメソッドで例外が throw されると、経路は開始できなくなります。

refresh

```
public void refresh () throws IPTException
```

以下のプロパティーが使用可能です。

- リスナー・ポート
- 宛先
- 宛先ポート

MQIPT が MQIPT Administration クライアントによって自己のリフレッシュを要求されると、refresh メソッドが MQIPT によって呼び出されます。構成ファイルでプロパティーが変更されると、このアクションが通常呼び出されます。MQIPT はすべてのプロパティーを構成ファイルからロードし、どれが変更されたか、経路を即時に再始動する必要があるかどうか、または MQIPT が次回再始動されるまで待機できるかどうかを判別します。

このメソッドは、使用する外部データ (つまり、init メソッドのときにロードされたデータ) の再ロードを実行する必要があります。このメソッドで例外が throw されると、経路は使用不可になります。

close

```
public void close ()
```

以下のプロパティーが使用可能です。

- リスナー・ポート
- 宛先
- 宛先ポート

MQIPT が MQIPT Administration クライアントによって停止するように要求されると、close() メソッドが MQIPT によって呼び出されます。その操作中に取得したシステム・リソースがあればそのシステム・リソースを解放する必要があります。MQIPT は、シャットダウンする前にこのメソッドが完了するのを待機します。

このメソッドは、セキュリティー出口が使用可能にされたが、構成ファイル内で使用不可にされた場合にも呼び出されます。

validate

```
public SecurityExitResponse validate ()
```

以下のプロパティーが使用可能です。

- リスナー・ポート
- 宛先
- 宛先ポート
- タイムアウト
- クライアント IP アドレス
- クライアント・ポート・アドレス
- チャネル名
- キュー・マネージャー名

妥当性検査をする接続要求を MQIPT が受け取ると、validate メソッドが MQIPT によって呼び出されます。SSLProxyMode プロパティーが使用可能にされている場合、チャネル名とキュー・マネージャー名は、使用可能にな

りません。このフィーチャーは SSL データを通過させるためだけに使われるので、通常は初期データ・フローから取得されるデータに到達できなくなります。ターゲット・キュー・マネージャーへの接続が確立された後までこの情報が使用可能にならないため、キュー・マネージャー名が WMQ クライアント接続で使用可能になりません。

セキュリティー出口は、以下の情報が入っている SecurityExitResponse オブジェクトを戻す必要があります。

- 理由コード (設定が必須)
- 新規宛先アドレス (任意)
- 新規宛先リスナー・ポート・アドレス (任意)
- メッセージ (任意)

理由コードは、接続が MQIPT によって受け入れられるかまたは拒否されるかを決定します。newDestination と newDestinationPort の各フィールドは、任意に設定することができ、新しいターゲット (QM) を定義します。これらのプロパティーを設定しない場合、構成ファイルに定義された、経路の Destination と DestinationPort の各プロパティーが使用されます。メッセージがあれば、そのメッセージは接続ログ・ファイル・エントリーに付加されます。

プロパティーを取得する場合にサポートされるメソッドは、以下のとおりです。

public int getListenerPort()
経路のリスナー・ポートを検索する - ListenerPort プロパティーによって定義されている

public String getDestination()
宛先アドレスを検索する - Destination プロパティーによって定義されている

public int getDestinationPort()
宛先のリスナー・ポート・アドレスを検索する - DestinationPort プロパティーによって定義されている

public String getClientIPAddress()
接続要求を行うクライアントの IP アドレスを検索する

public int getClientPortAddress()
接続要求を行うクライアントが使用するポート・アドレスを検索する

public int getTimeout()
タイムアウト値を検索する。MQIPT はセキュリティー出口が要求を妥当性検査するのを待機する - SecurityExitTimeout プロパティーによって定義される

public int getConnThreadID()
接続要求を処理する接続スレッド ID (デバッグ目的には有用である) を検索する

public String getChannelName()
接続要求で使用された WMQ チャネル名を検索する

public String getQMName()
接続要求で使用された WMQ Queue Manager 名を検索する

public boolean getTimedout()
タイムアウトが期限切れになったかどうかを判別するためにセキュリティー出口が使用できる

com.ibm.mq.ipt.SecurityExitResponse クラス

このクラスは、ユーザー定義のセキュリティー出口から応答を MQIPT に戻すために使用され、接続要求を受け入れるか拒否するかを決めるのに使用されます。このタイプのオブジェクトは、validate メソッド (上記参照) でのみ作成されます。これらのオブジェクトを作成するための便利なコンストラクターがあり、各プロパティー用の set メソッドがあります。詳細については、サンプルのセキュリティー出口を参照してください。

デフォルトの SecurityExitResponse オブジェクトを作成すると、接続要求は拒否されます。

サポートされるコンストラクターは、以下のとおりです。

public SecurityExitResponse (String dest, int destPort, int rc, String msg) throws IPTException

ここで、

- `dest` は、新規ターゲット宛先です
- `destPort` は、新規宛先ポート・アドレスです
- `rc` は、理由コードです
- `msg` は、接続ログ・エントリーに追加されるメッセージです

public SecurityExitResponse (String dest, int destPort, int rc) throws IPTException

public SecurityExitResponse (int rc, String msg) throws IPTException

public SecurityExitResponse (int rc) throws IPTException

プロパティー値を設定する場合にサポートされるメソッドは、以下のとおりです。

public void setDestination(String dest)
接続要求の新しい宛先アドレスを設定します

public void setDestinationPort(int port) throws IPTException
接続要求の新しい宛先リスナー・ポート・アドレスを設定します - 無効のポート・アドレスの IPTException を throw する

public void setMessage(String msg)
メッセージを接続ログ・レコードに追加します

public void setReasonCode(int rc) throws IPTException
接続要求の理由コードを設定します - 不明値の IPTException を throw する

有効な理由コードは、以下のとおりです。

- `SecurityExitResponse.OK = 0`
- `SecurityExitResponse.NOT_AUTHORIZED = 1`
- `SecurityExitResponse.NOT_READY = 2`

トレース

ユーザー定義のセキュリティー出口における問題の診断に役立てるため、MQIPT が使用する機能と類似のトレース機能を使用可能にすることができます。経路の Trace プロパティーの値を 1-5 に設定すると、errors サブディレクトリーにトレース・ファイルが作成されます。トレース・ファイルの名前は、セキュリティー出口の名前と同じです。

同時に実行されているセキュリティー出口のインスタンスが複数ある可能性があるため、トレース・ファイルの個々のエントリーは、スレッド ID を使用することによって識別できます。

トレース機能の初期化は、セキュリティー出口を開始したときに MQIPT によって実行されます。その場合、行う必要があることは、トレースしたい情報の選択だけです。サンプルのユーザー出口には多くのトレースの例があります。

トレースでは `entry` 呼び出し、`exit` 呼び出し、およびトレースしたいデータを少なくとも指定する必要があります。たとえば、以下のとおりです。

```
<a_method>
{
  SecurityExit.rastlRoute.entry(RASITraceEvent.TYPE_ENTRY_EXIT,
                                this,
                                "method_name");

  .
  .
  .
  <code>

  .
  .
  .
  SecurityExit.rastlRoute.trace(RASITraceEvent.TYPE_MISC_DATA,
                                this,
                                "data");

  .
  .
  .
  <code>

  .
  .
  .
  SecurityExit.rastlRoute.exit(RASITraceEvent.TYPE_ENTRY_EXIT,
                               this,
                               "method_name");
}
```

第 9 章 ポート・アドレスの制御

ポート・アドレスの制御

MQIPT を使用しているとき、経路に OutgoingPort プロパティーを設定することによって発信接続を行うときに使用するローカルのポート・アドレスの範囲を制限することができます。ローカルのポート・アドレスの範囲は、MaxConnectionThreads 値を使用して計算されます。たとえば、OutgoingPort を 1600 に設定し、MaxConnectionThreads を 20 に設定した場合には、その経路のローカルのポート・アドレスの範囲は、1600-1619 になります。経路にまたがってポート・アドレスの競合がないことを確認するのは、MQIPT 管理者が行います。OutgoingPort を定義していない場合、デフォルト値の 0 は、システムが割り振ったポート・アドレスが各接続に使用されることを意味します。

詳細については、サンプルの 133 ページの『ポート・アドレスの割り振り』を参照してください。

マルチホーム・システム

マルチホーム・システムを使用しているとき、LocalAddress プロパティーを使用することによって、発信接続がバインドする IP アドレスを指定できます。ホスト名は、このプロパティーではサポートされていません。

第 10 章 その他のセキュリティー上の考慮事項

その他のセキュリティー上の考慮事項

SSL を使用しないことに決定した場合は、MQIPT からチャネル・セキュリティー・フローが提供されるので、WebSphere MQ チャネル出口ルーチンを使用して、チャネル全体にわたり端から端までセキュリティーを提供することができます。

MQIPT は、このほかにも、設計者が安全なソリューションを作成する際に役立ついくつかの機能を提供します。それは、以下のとおりです。

- 内部ネットワーク内の多くのクライアントが発信接続を試行している場合は、これらのクライアントはすべて、ファイアウォールの内部にある MQIPT を通過することができます。このため、ファイアウォール管理者は、MQIPT マシンだけへの外部アクセス権を付与する必要があります。
- MQIPT は、自分が SOCKS プロキシーとして機能していないか、またはセキュリティー出口を使用していない限り、構成ファイルに明示的に構成されているキュー・マネージャーにのみ接続することができます。
- MQIPT は、自分が送受信するメッセージが有効であり、WebSphere MQ プロトコルに準拠しているか調べます。こうすることによって、MQIPT が、WebSphere MQ プロトコルの外側のセキュリティー・アタックに使用されるのを防止することができます。 MQIPT が SSL プロキシーとして機能している場合に、すべての WebSphere MQ データとプロトコルが暗号化されていれば、MQIPT は初期 SSL ハンドシェークしか保証できません。このような場合は、Java Security Manager を使用することをお勧めします。 33 ページの『Java Security Manager』を参照してください。
- これによって、チャネル出口ルーチンで、独自のエンドツーエンド・セキュリティー・プロトコルを実行することができます。
- MQIPT を使用すれば、`MaxConnectionThreads` プロパティーを設定して、着信要求の総数を制限することができます。こうすれば、攻撃を受けやすい内部キュー・マネージャーをサービス妨害攻撃から保護するのに役立ちます。

MQIPT の `mqipt.conf` 構成ファイルは内部ホストへのアクセスを制御するので、このファイルを保護する必要があります。また、コマンド・ポート (それが使用可能になっている場合) への無許可アクセスを防止する必要があります。そのようなアクセスにより、外部から MQIPT をシャットダウンすることができるからです。

第 11 章 各種フィーチャー

正常終了と失敗条件

WebSphere MQ チャネルのクローズ (正常または異常) を検出すると、MQIPT は、そのチャネル・クローズを伝搬します。管理者が MQIPT への経路をクローズすると、その経路を通るすべてのチャネルがクローズされます。

MQIPT は、オプションのアイドル・タイムアウト機能を備えています。チャネルがタイムアウトを超過して一定の時間アイドル状態になっていることを検出すると、MQIPT は、この 2 つの接続の即時シャットダウンを行います。

チャネルの両端にある 2 つの WebSphere MQ システムは、これらの異常終了状態をネットワーク障害、または相手側によるチャネルの終了のいずれかと見なします。次に、これらのチャネルは、MQIPT を使用していない場合とまったく同じように、再始動してリカバリーすることができます (障害がプロトコル未確定期間中に発生した場合)。

メッセージの安全性

高速の非持続 WebSphere MQ メッセージを使用するときに、MQIPT 経路が失敗するか、または WebSphere MQ メッセージの転送中に MQIPT を再始動すると、メッセージが消失することがあります。この経路を再始動する前に、MQIPT を使用しているすべての WebSphere MQ チャネルが非アクティブ状態になっていることを確認してください。

WebSphere MQ のメッセージとチャネルについては、「*MQSeries 相互通信*」SC88-7775 を参照してください。

接続ログ

MQIPT は、すべての成功および失敗接続試行のリストを収めた接続ログ機能を提供します。この機能の制御は、 `ConnectionLog` および `MaxLogFileSize` プロパティーを使用して行います。詳細については、 83 ページの『グローバル・セクション参照情報』を参照してください。

MQIPT を開始するたびに、新規の接続ログが作成されます。識別のため、次の例のように、ファイル名には現在のタイム・スタンプが含まれます。

```
mqiptYYYYMMDDHHmmSS.log
```

ここで、

- `YYYY` は年
- `MM` は月
- `DD` は日
- `HH` は時間
- `mm` は分

- SS は秒

監査の目的で、これらのログ・ファイルは消去されません。これらのファイルの管理や、これらが不要になったときの削除は、MQIPT 管理者が行います。

第 12 章 先行バージョンからのアップグレード

MQIPT をバージョン 1.2 からバージョン 1.3 へアップグレードするには、以下のステップを実行します。

1. mqipt.conf と client.conf の構成ファイルのコピーをとります。mqipt.conf は MQIPT ホーム・ディレクトリーにあり、client.conf は bin サブディレクトリーにあります。
2. 次のコマンドを実行して、MQIPT を停止します。

   ```
   mqiptAdmin -stop
   ```

3. MQIPT がサービスとしてインストールされている場合は、それを除去してから MQIPT をアンインストールしなければなりません。

   ```
   mqiptService -remove
   ```

4. MQIPT のアンインストール・プログラムを実行します。
5. MQIPT V1.3 をインストールした後で、保管した構成ファイルを元のロケーションにコピーします。
6. MQIPT Administration GUI を使用して、MQIPT に対する変更を管理します。V1.2 の構成ファイルはこの GUI と互換性があります。

一部のインプリメンテーションでは、ユーザー自身の組織の制御下にあるローカル MQIPT サービスと、ユーザーのクライアントの組織の制御下に置くことのできるリモート MQIPT サービスとを必要とします。この場合、両方の MQIPT サービスを同時にマイグレーションすることは非常に困難ですが、MQIPT の場合、これは問題ではありません。特に断りがない限り、MQIPT の旧バージョンは最新バージョンと互換性があります。これによって、MQIPT のマイグレーション・プロセスは一層容易になります。

また、まず MQIPT をアンインストールせずに、MQIPT のコアをアップグレードすることもできます。MQIPT を実行するために必要なすべてのクラスは、MQipt.jar ファイルに保管されています。別のマシンに MQIPT の最新バージョンをインストールして、そのインストールからご使用の実働システムに MQipt.jar ファイルをコピーできます。同じことが、Administration GUI の実行に必要なクラスについて当てはまります。これらは、guiadmin.jar ファイルに含まれています。

新規構成オプション

以下のプロパティーは、バージョン 1.3 で初めて取り入れられたものです。

- IgnoreExpiredCRLs
- LDAP
- LDAPCacheTimeout
- LDAPIgnoreErrors
- LDAPSaveCRL
- LDAPServer1
- LDAPServer1Password

- LDAPServer1Port
- LDAPServer1Timeout
- LDAPServer1Userid
- LDAPServer2
- LDAPServer2Password
- LDAPServer2Port
- LDAPServer2Timeout
- LDAPServer2Userid
- RouteRestart
- SecurityExit
- SecurityExitName
- SecurityExitPath
- SecurityExitTimeout
- SSLClientSiteDN_C
- SSLClientSiteDN_CN
- SSLClientSiteDN_L
- SSLClientSiteDN_O
- SSLClientSiteDN_OU
- SSLClientSiteDN_ST
- SSLClientSiteLabel
- SSLServerSiteDN_C
- SSLServerSiteDN_CN
- SSLServerSiteDN_L
- SSLServerSiteDN_O
- SSLServerSiteDN_OU
- SSLServerSiteDN_ST
- SSLServerSiteLabel

これらのすべてのプロパティーに関する参照情報については、 79 ページの『構成参照情報』を参照してください。

第 13 章 Windows での internet pass-thru のインストール

この章では、Windows NT、Windows 2000、または Windows XP システムで MQIPT をインストールする方法について説明します。

ファイルのダウンロードとインストール

MQIPT (MS81、カテゴリー 3 SupportPac™) は、次の WebSphere MQ SupportPac Web ページからダウンロードできます。

```
http://www.ibm.com/webspheremq/supportpacs
```

ダウンロードの指示を実行してください。

コマンド・プロンプトをオープンし、`ms81_nt.zip` を一時ディレクトリーに解凍します。 `setup.exe` を実行し、オンライン指示に従います。

MQIPT は、管理者権限を持つユーザーがインストールしなければなりません。

MQIPT には、以下の表に示されているファイルと、その次の表に示されている Administration Client GUI 用のファイル (別個にインストール可能なフィーチャーとして出荷される) が含まれています。

ファイル	目的
Readme.txt	資料に記載されていない最新情報
mqiptSample.conf	サンプル構成ファイル
ssl¥sslSample.pfx	テスト鍵リング・ファイル
ssl¥sslSample.pwd	テスト鍵リング・ファイル用のパスワード・ファイル
ssl¥sslCAdefault.pfx	サンプル認証局 (CA) 鍵リング・ファイル
ssl¥sslCAdefault.pwd	サンプル認証局 (CA) 鍵リング・ファイル用のパスワード・ファイル
ssl¥KeyMan.zip	KeyMan ユーティリティー
exits¥ SampleOneRouteExit.java	
exits¥ SampleOneRouteExit.conf	SampleOneRouteExit の構成ファイル
exits¥SampleRoutingExit.java	

ファイル	目的
exits¥SampleRoutingExit.conf	SampleRoutingExit の構成ファイル
exits¥SampleSecurityExit.java	
lib¥MQipt.jar	ランタイム、クラス、およびプロパティー・ファイルが入っている
lib¥ADV_mqipt_normal. class	「通常」モード用 Network Dispatcher アドバイザー
lib¥ADV_mqipt_replace. class	「置換」モード用 Network Dispatcher アドバイザー
lib¥mqipt1414Sample.ssl	Network Dispatcher アドバイザー用のサンプル・トリガー・ファイル
bin¥mqipt.bat	コマンド行から MQIPT を実行するためのショートカット
bin¥mqiptAdmin.bat	MQIPT を停止し、ファイル情報をリフレッシュするためのショートカット
bin¥mqiptPW.bat	鍵リング・ファイルをオープンするために使用されるパスワードを暗号化する
bin¥mqiptservice.exe	MQIPT を Windows Service Control Manager に追加したり、除去したりするためのもの
bin¥mqiptVersion.bat	MQIPT のバージョン番号の表示
web¥MQIPTServlet.war	サーブレット・バージョン用の Web アーカイブ・ファイル
doc¥<lang>¥html¥<filename>.zip	HTML 形式の「*internet pass-thru*」マニュアルのマスター・ファイル。ソフトコピー文書の詳細については、183 ページの『参照文献』を参照。

Administration Client GUI フィーチャーに関連するファイルは、以下のとおりです。

ファイル	目的
lib¥guiadmin.jar	ランタイム、クラス、およびプロパティー・ファイルが入っている
bin¥mqiptGui.bat	コマンド行から Administration Client を実行するためのショートカット
bin¥customSample. properties	Administration Client の外観およびアクセシビリティーをカスタマイズするためのサンプル・ファイル

インストーラーは、`MQipt.jar` および `guiadmin.jar` ファイルのロケーションでシステム CLASSPATH 環境変数を更新します。

internet pass-thru のセットアップ

MQIPT を初めて開始する場合は、その前に、`mqiptSample.conf` サンプル構成ファイルを `mqipt.conf` にコピーしてください。詳細については、73 ページの『第 19 章 internet pass-thru の管理と構成』を参照してください。

コマンド行からの internet pass-thru の開始

コマンド・プロンプトをオープンし、ディレクトリーを `bin` ディレクトリーに変えて、`mqipt` を実行します。たとえば、以下のとおりです。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

Windows の「Start (スタート)」->「Programs (プログラム)」メニューからも MQIPT を開始できます。

オプションを指定しないで mqipt スクリプトを実行すると、 “.” のデフォルト・ロケーションが構成ファイルに使用されます (mqipt.conf)。異なるロケーションを指定するには、次のようにします。

```
mqipt <directory name>
```

MQIPT の状況を示すメッセージがコンソールに表示されます。エラーが起こった場合は、 155 ページの『問題判別』を参照してください。以下のメッセージは、MQIPT が正常に開始された場合の例です。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from c:¥mqipt¥mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path c:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1418 ready for connection requests
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file c:¥mqipt¥KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

MQIPT を初めて呼び出すときは、以下の mqipt ホーム・ディレクトリーのサブディレクトリーが自動的に作成されます。

- 接続ログが保管されている ″logs″ ディレクトリー
- 任意の First Failure Support Technology™ (FFST™) とトレース・レコードが書き込まれる ″errors″ ディレクトリー

コマンド行からの Administration Client の開始

コマンド・プロンプトをオープンし、ディレクトリーを bin ディレクトリーに変えて、mqiptGui を実行します。たとえば、以下のとおりです。

```
c:
cd ¥mqipt¥bin
mqiptGui
```

SOCKS プロキシーを使用して、Administration Client がファイアウォール経由で外部の MQIPT に接続できるようにするには、以下のように、ホスト名またはアドレス、およびポート番号を指定します。

```
mqiptGui <socksHostName <socksPort>>
```

デフォルトの socksPort は 1080 です。

Administration Client の状況が、Administration Client のメイン・ウィンドウにメッセージで示されます。

Windows サービス制御プログラムの使用

別個のサービス制御プログラム (mqiptservice.exe) が提供されるので、MQIPT を Windows サービスとして管理したり開始したりできるようになります。

mqiptservice.exe は、以下のようなコマンド行引き数をとります。

mqiptservice -install *path*
サービスを Windows サービス・パネル上に手動サービスとして表示するように、そのサービスをインストールして登録します。サービス・パネルへ進み、設定を “automatic” に変更して、システム開始時に MQIPT が自動的に開始されるようにします。このサービスをインストールした後で、Windows をリブートする必要があります。パス・パラメーター (指定が必要) は、mqipt.conf 構成ファイルが入っているディレクトリーへの完全修飾パスです。このパス名にブランクが含まれている場合は、その名前を引用符で囲んでください。

mqiptservice -remove
サービスを除去して、サービス・パネルに表示されないようにします。

mqiptservice ?
有効な引き数をリストする米国英語のヘルプ・メッセージを表示します。

同一コマンドに「インストール」と「除去」の両方を指定すると、エラーになります。

Windows は、引き数のない mqiptservice プログラムを内部で呼び出します。ユーザーが引き数のないコマンド行からそれを呼び出すと、プログラムがタイムアウトになり、エラーが戻されます。

MQIPT サービスを開始すると、アクティブなすべての MQIPT 経路が始動します。それを停止すると、すべての経路が即時シャットダウンされます。

注: システムの PATH 環境変数には、JNI ランタイム・ライブラリーのロケーションが入っていなければなりません。 jvm.dll ファイルは、JDK の client サブディレクトリーに入っています。

Windows サービスとしての internet pass-thru のアンインストール

サービスとしての MQIPT をアンインストールするには、まず、Windows サービス・パネルからそれを停止します。次に、コマンド・プロンプトをオープンして、MQIPT の bin サブディレクトリーへ進み、以下のように入力します。

```
mqiptservice -remove
```

internet pass-thru のアンインストール

システムから MQIPT をアンインストールする前に、上記のようにして、Windows サービスとしてのそれを除去します。次に、Windows の「Start (スタート)」メニューからアンインストール・プロセスを実行します。

第 14 章 Sun Solaris での internet pass-thru のインストール

この章では、Sun Solaris システムで MQIPT をインストールする方法について説明します。

ファイルのダウンロードとインストール

MQIPT は、次の WebSphere MQ SupportPac Web ページからダウンロードできます。

```
http://www.ibm.com/webspheremq/supportpacs
```

ダウンロードの指示を実行してください。

ルートとしてログインし、ms81_sol.tar.Z を解凍して一時ディレクトリーに入れます。次の例のように、pkgadd コマンドを実行します。

```
login root
cd /tmp
uncompress -fv ms81_sol.tar.Z
tar xvf ms81_sol.tar
pkgadd -d . mqipt
```

この例では、ms81_sol.tar.Z が /tmp ディレクトリーに入っていることを前提にしています。

MQIPT には、以下の表に示されているファイル (Administration Client GUI のファイルを含む) が入っています。

ファイル	目的
Readme.txt	資料に記載されていない最新情報
mqiptSample.conf	サンプル構成ファイル
ssl/sslSample.pfx	テスト鍵リング・ファイル
ssl/sslSample.pwd	テスト鍵リング・ファイル用のパスワード・ファイル
ssl/sslCAdefault.pfx	サンプル認証局 (CA) 鍵リング・ファイル
ssl/sslCAdefault.pwd	サンプル認証局 (CA) 鍵リング・ファイル用のパスワード・ファイル
ssl/KeyMan.zip	KeyMan ユーティリティー
exits/ SampleOneRouteExit.java	サンプル・セキュリティー出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit の構成ファイル
exits/SampleRoutingExit.java	サンプル・セキュリティー出口

ファイル	目的
exits/SampleRoutingExit.conf	SampleRoutingExit の構成ファイル
exits/SampleSecurityExit.java	サンプル・セキュリティー出口
lib/MQipt.jar	ランタイム、クラス、およびプロパティー・ファイルが入っている
lib/ADV_mqipt_normal. class	「通常」モード用 Network Dispatcher アドバイザー
lib/ADV_mqipt_replace. class	「置換」モード用 Network Dispatcher アドバイザー
lib/mqipt1414Sample.ssl	Network Dispatcher アドバイザー用のサンプル・トリガー・ファイル
bin/mqipt	コマンド行から MQIPT を実行するためのショートカット
bin/mqiptAdmin	MQIPT を停止し、ファイル情報をリフレッシュするためのショートカット
bin/mqiptPW	鍵リング・ファイルをオープンするために使用されるパスワードを暗号化する
bin/mqiptVersion	MQIPT のバージョン番号の表示
bin/mqiptService	システム始動時に MQIPT が自動的に開始されるようにするための MQIPT のインストール
bin/mqiptEnv	mqipt.jar ファイルのロケーションを定義し、他のスクリプトでのみ使用する。
web/MQIPTServlet.war	サーブレット・バージョン用の Web アーカイブ・ファイル
doc/<lang>/html/ <filename>.zip	HTML 形式の「*internet pass-thru*」マニュアルのマスター・ファイル。ソフトコピー文書の詳細については、 183 ページの『参照文献』を参照。
lib/guiadmin.jar	Administration Client GUI 用のランタイム、クラス、およびプロパティー・ファイルが入っている
bin/mqiptGui	コマンド行から Administration Client GUI を実行するためのショートカット
bin/customSample. properties	Administration Client の外観およびアクセシビリティーをカスタマイズするためのサンプル・ファイル

internet pass-thru のセットアップ

MQIPT を初めて開始する場合は、その前に、mqiptSample.conf サンプル構成ファイルを mqipt.conf にコピーしてください。詳細については、 73 ページの『第 19 章 internet pass-thru の管理と構成』を参照してください。

コマンド行からの internet pass-thru の開始

ルートとしてログインし、ディレクトリーを bin ディレクトリーに変えます。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqipt ..
```

オプションを指定しないで mqipt スクリプトを実行すると、“.” のデフォルト・ロケーションが構成ファイル (mqipt.conf) に使用されます。異なるロケーションを指定するには、次のようにします。

```
mqipt <directory name>
```

MQIPT の状況を示すメッセージがコンソールに表示されます。エラーが起こった場合は、155 ページの『問題判別』を参照してください。以下のメッセージは、MQIPT が正常に開始された場合の例です。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from /opt/mqipt/mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path /opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1418 ready for connection requests
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file /opt/mqipt/KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

MQIPT を初めて呼び出すときは、mqipt ホーム・ディレクトリーの以下のサブディレクトリーが自動的に作成されます。

- 接続ログが保管されている ″logs″ ディレクトリー
- 任意の First Failure Support Technology (FFST) とトレース・レコードが書き込まれる ″errors″ ディレクトリー

internet pass-thru の自動開始

システム開始時に MQIPT が自動的に開始されるようにするには、mqiptService スクリプトを実行します。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqiptService -install
```

MQIPT が自動的に開始されないようにするには、次のようにします。

```
cd /opt/mqipt/bin
mqiptService -remove
```

コマンド行からの Administration Client の開始

コマンド・プロンプトをオープンし、ディレクトリーを bin ディレクトリーに変えて、mqiptGui を実行します。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqiptGui
```

Administration Client がファイアウォール経由で外部の MQIPT に接続できるようにするには、以下のように、ホスト名またはアドレス、およびポート番号を指定します。

```
mqiptGui <socksHostName <socksPort>>
```

デフォルトの socksPort は 1080 です。

Administration Client の状況が、Administration Client のメイン・ウィンドウにメッセージで示されます。

internet pass-thru のアンインストール

55 ページの『internet pass-thru の自動開始』に説明されているように、MQIPT をシステムからアンインストールする前に、それが自動的に開始されないようにしてください。 ルートとしてログインし、次のように pkgrm コマンドを実行します。

```
pkgrm mqipt
```

第 15 章 AIX での internet pass-thru のインストール

この章では、AIX システムで MQIPT をインストールする方法について説明します。

ファイルのダウンロードとインストール

MQIPT は、次の WebSphere MQ SupportPac Web ページからダウンロードできます。

```
http://www.ibm.com/webspheremq/supportpacs
```

ダウンロードの指示を実行してください。

ルートとしてログインし、ms81_aix.tar.Z を解凍して一時ディレクトリーに入れます。次の例のように、installp コマンドを実行します。

```
cd /tmp
uncompress -fv ms81_aix.tar.Z
tar xvf ms81_aix.tar
installp -d . -a mqipt-RT
```

この例では、ms81_aix.tar.Z が /tmp ディレクトリーに入っていることを前提にしています。

MQIPT には、以下の表に示されているファイル (Administration Client GUI のファイルを含む) が入っています。

ファイル	目的
Readme.txt	資料に記載されていない最新情報
mqiptSample.conf	サンプル構成ファイル
ssl/sslSample.pfx	テスト鍵リング・ファイル
ssl/sslSample.pwd	テスト鍵リング・ファイル用のパスワード・ファイル
ssl/sslCAdefault.pfx	サンプル認証局 (CA) 鍵リング・ファイル
ssl/sslCAdefault.pwd	サンプル認証局 (CA) 鍵リング・ファイル用のパスワード・ファイル
ssl/KeyMan.zip	KeyMan ユーティリティー
exits/ SampleOneRouteExit.java	サンプル・セキュリティー出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit の構成ファイル
exits/SampleRoutingExit.java	サンプル・セキュリティー出口

ファイル	目的
exits/SampleRoutingExit.conf	SampleRoutingExit の構成ファイル
exits/SampleSecurityExit.java	サンプル・セキュリティー出口
lib/MQipt.jar	ランタイム、クラス、およびプロパティー・ファイルが入っている
lib/ADV_mqipt_normal. class	「通常」モード用 Network Dispatcher アドバイザー
lib/ADV_mqipt_replace. class	「置換」モード用 Network Dispatcher アドバイザー
lib/mqipt1414Sample.ssl	Network Dispatcher アドバイザー用のサンプル・トリガー・ファイル
bin/mqipt	コマンド行から MQIPT を実行するためのショートカット
bin/mqiptAdmin	MQIPT を停止し、ファイル情報をリフレッシュするためのショートカット
bin/mqiptPW	鍵リング・ファイルをオープンするために使用されるパスワードを暗号化する
bin/mqiptVersion	MQIPT のバージョン番号の表示
bin/mqiptService	システム始動時に MQIPT が自動的に開始されるようにするための MQIPT のインストール
bin/mqiptEnv	mqipt.jar ファイルのロケーションを定義し、他のスクリプトでのみ使用する。
web/MQIPTServlet.war	サーブレット・バージョン用の Web アーカイブ・ファイル
doc/<lang>/html/ <filename>.zip	HTML 形式の「*internet pass-thru*」マニュアルのマスター・ファイル。ソフトコピー文書の詳細については、 183 ページの『参照文献』を参照。
lib/guiadmin.jar	Administration Client GUI 用のランタイム、クラス、およびプロパティー・ファイルが入っている
bin/mqiptGui	コマンド行から Administration Client を実行するためのショートカット
bin/customSample. properties	Administration Client の外観およびアクセシビリティーをカスタマイズするためのサンプル・ファイル

internet pass-thru のセットアップ

MQIPT を初めて開始する場合は、その前に、mqiptSample.conf サンプル構成ファイルを mqipt.conf にコピーしてください。詳細については、 73 ページの『第 19 章 internet pass-thru の管理と構成』を参照してください。

コマンド行からの internet pass-thru の開始

ルートとしてログインし、ディレクトリーを bin ディレクトリーに変えます。たとえば、以下のとおりです。

```
cd /usr/opt/mqipt/bin
mqipt ..
```

オプションを指定しないで mqipt スクリプトを実行すると、“.” のデフォルト・ロケーションが構成ファイル (mqipt.conf) に使用されます。異なるロケーションを指定するには、次のようにします。

```
mqipt <directory name>
```

MQIPT の状況を示すメッセージがコンソールに表示されます。エラーが起こった場合は、 155 ページの『問題判別』を参照してください。以下のメッセージは、MQIPT が正常に開始された場合の例です。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from /usr/opt/mqipt/mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path /usr/opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1418 ready for connection requests
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file /usr/opt/mqipt/KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

MQIPT を初めて呼び出すときは、mqipt ホーム・ディレクトリーの以下のサブディレクトリーが自動的に作成されます。

- 接続ログが保管されている ″logs″ ディレクトリー
- 任意の First Failure Support Technology (FFST) とトレース・レコードが書き込まれる ″errors″ ディレクトリー

internet pass-thru の自動開始

システム開始時に MQIPT が自動的に開始されるようにするには、mqiptService スクリプトを実行してエントリーを inittab に追加します。たとえば、以下のとおりです。

```
cd /usr/opt/mqipt/bin
../mqiptService -install
```

MQIPT が自動的に開始されないようにして、そのエントリーを inittab から除去するには、次のようにします。

```
cd /usr/opt/mqipt/bin
../mqiptService -remove
```

コマンド行からの Administration Client の開始

コマンド・プロンプトをオープンし、ディレクトリーを bin ディレクトリーに変えて、mqiptGui を実行します。たとえば、以下のとおりです。

```
cd /usr/opt/mqipt/bin
../mqiptGui
```

Administration Client がファイアウォール経由で外部の MQIPT に接続できるようにするには、以下のように、ホスト名またはアドレス、およびポート番号を指定します。

```
mqiptGui <socksHostName <socksPort>>
```

デフォルトの socksPort は 1080 です。

Administration Client の状況が、Administration Client のメイン・ウィンドウにメッセージで示されます。

internet pass-thru のアンインストール

59 ページの『internet pass-thru の自動開始』に説明されているように、MQIPT をシステムからアンインストールする前に、それが自動的に開始されないようにしてください。 ルートとしてログインし、次のように installp コマンドを実行します。

```
installp -u mqipt-RT
```

第 16 章 HP-UX での internet pass-thru のインストール

この章では、HP-UX システムで MQIPT をインストールする方法について説明します。

ファイルのダウンロードとインストール

MQIPT は、次の WebSphere MQ SupportPac Web ページからダウンロードできます。

```
http://www.ibm.com/webspheremq/supportpacs
```

ダウンロードの指示を実行してください。

ルートとしてログインし、ms81_hp11.tar.Z を解凍して一時ディレクトリーに入れます。次の例のように、swinstall コマンドを実行します。

```
login root
cd /tmp
uncompress -fv ms81_hp11.tar.Z
tar xvf ms81_hp11.tar
swinstall -s /tmp MQIPT.MQIPT-RT
```

この例では、ms81_hp11.tar.Z が /tmp ディレクトリーに入っていることを前提にしています。

MQIPT には、以下の表に示されているファイル (Administration Client GUI のファイルを含む) が入っています。

ファイル	目的
Readme.txt	資料に記載されていない最新情報
mqiptSample.conf	サンプル構成ファイル
ssl/sslSample.pfx	テスト鍵リング・ファイル
ssl/sslSample.pwd	テスト鍵リング・ファイル用のパスワード・ファイル
ssl/sslCAdefault.pfx	サンプル認証局 (CA) 鍵リング・ファイル
ssl/sslCAdefault.pwd	サンプル認証局 (CA) 鍵リング・ファイル用のパスワード・ファイル
ssl/KeyMan.zip	KeyMan ユーティリティー
exits/ SampleOneRouteExit.java	サンプル・セキュリティー出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit の構成ファイル
exits/SampleRoutingExit.java	サンプル・セキュリティー出口

ファイル	目的
exits/SampleRoutingExit.conf	SampleRoutingExit の構成ファイル
exits/SampleSecurityExit.java	サンプル・セキュリティー出口
lib/MQipt.jar	ランタイム、クラス、およびプロパティー・ファイルが入っている
lib/ADV_mqipt_normal. class	「通常」モード用 Network Dispatcher アドバイザー
lib/ADV_mqipt_replace. class	「置換」モード用 Network Dispatcher アドバイザー
lib/mqipt1414Sample.ssl	Network Dispatcher アドバイザー用のサンプル・トリガー・ファイル
bin/mqipt	コマンド行から MQIPT を実行するためのショートカット
bin/mqiptAdmin	MQIPT を停止し、ファイル情報をリフレッシュするためのショートカット
bin/mqiptPW	鍵リング・ファイルをオープンするために使用されるパスワードを暗号化する
bin/mqiptVersion	MQIPT のバージョン番号の表示
bin/mqiptService	システム始動時に MQIPT が自動的に開始されるようにするための MQIPT のインストール
bin/mqiptEnv	`mqipt.jar` ファイルのロケーションを定義し、他のスクリプトでのみ使用する。
bin/mqiptFork	システム始動時に MQIPT の立ち上げに使用
web/MQIPTServlet.war	サーブレット・バージョン用の Web アーカイブ・ファイル
doc/<lang>/html/ <filename>.zip	HTML 形式の「*internet pass-thru*」マニュアルのマスター・ファイル。ソフトコピー文書の詳細については、 183 ページの『参照文献』を参照。
lib/guiadmin.jar	Administration Client GUI 用のランタイム、クラス、およびプロパティー・ファイルが入っている
bin/mqiptGui	コマンド行から Administration Client GUI を実行するためのショートカット
bin/customSample. properties	Administration Client の外観およびアクセシビリティーをカスタマイズするためのサンプル・ファイル

internet pass-thru のセットアップ

MQIPT を初めて開始する場合は、その前に、`mqiptSample.conf` サンプル構成ファイルを `mqipt.conf` にコピーしてください。詳細については、 73 ページの『第 19 章 internet pass-thru の管理と構成』を参照してください。

コマンド行からの internet pass-thru の開始

ルートとしてログインし、ディレクトリーを `bin` ディレクトリーに変えます。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqipt ..
```

オプションを指定しないで mqipt スクリプトを実行すると、“.” のデフォルト・ロケーションが構成ファイルに使用されます (mqipt.conf)。異なるロケーションを指定するには、次のようにします。

```
mqipt <directory name>
```

MQIPT の状況を示すメッセージがコンソールに表示されます。エラーが起こった場合は、155 ページの『問題判別』を参照してください。以下のメッセージは、MQIPT が正常に開始された場合の例です。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from /opt/mqipt/mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path /opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1418 ready for connection requests
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file /opt/mqipt/KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

MQIPT を初めて呼び出すときは、mqipt ホーム・ディレクトリーの以下のサブディレクトリーが自動的に作成されます。

- 接続ログが保管されている ″logs″ ディレクトリー
- 任意の First Failure Support Technology (FFST) とトレース・レコードが書き込まれる ″errors″ ディレクトリー

internet pass-thru の自動開始

システム開始時に MQIPT が自動的に開始されるようにするには、mqiptService スクリプトを実行します。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqiptService -install
```

ここでは、JDK 1.4 がすでに /opt/java1.4 というディレクトリーにインストール済みであることを前提にしています。インストール済みでない場合は、mqipt.ske ファイルを編集して、JDK のロケーションを指すように PATH 変数を変更してください。 mqiptService -install コマンドを実行する前に、この変更を適用する必要があります。

MQIPT をサービスとして開始すると、console.log ログ・ファイルが logs サブディレクトリーに書き込まれます。このサブディレクトリーは、MQIPT を初めて実行するときに作成されるため、MQIPT をサービスとして実行する前に、少なくとも 1 回はそれを実行しておく必要があります。

MQIPT が自動的に開始されないようにするには、次のようにします。

```
cd /opt/mqipt/bin
mqiptService -remove
```

コマンド行からの Administration Client の開始

コマンド・プロンプトをオープンし、ディレクトリーを bin ディレクトリーに変えて、mqiptGui を実行します。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqiptGui
```

Administration Client がファイアウォール経由で外部の MQIPT に接続できるようにするには、以下のように、ホスト名またはアドレス、およびポート番号を指定します。

```
mqiptGui <socksHostName <socksPort>>
```

デフォルトの socksPort は 1080 です。

Administration Client の状況が、Administration Client のメイン・ウィンドウにメッセージで示されます。

internet pass-thru のアンインストール

63 ページの『internet pass-thru の自動開始』に説明されているように、MQIPT をシステムからアンインストールする前に、それが自動的に開始されないようにしてください。 ルートとしてログインし、次のように swremove コマンドを実行します。

```
swremove MQIPT
```

第 17 章 Linux での internet pass-thru のインストール

この章では、Linux システムで MQIPT をインストールする方法について説明します。

ファイルのダウンロードとインストール

MQIPT は、次の WebSphere MQ SupportPac Web ページからダウンロードできます。

```
http://www.ibm.com/webspheremq/supportpacs
```

ダウンロードの指示を実行してください。

ルートとしてログインし、ms81_linux.tar.z を解凍して一時ディレクトリーに入れます。次の例のように、rpm コマンドを実行します。

```
login root
cd /tmp
uncompress -fv ms81_linux.tar.z
tar xvf ms81_linux.tar
cd i386
rpm -i WebSphereMQ-IPT-1.3.0-0.i386.rpm
```

この例では、ms81_linux.tar.z が /tmp ディレクトリーに入っていることを前提にしています。

MQIPT には、以下の表に示されているファイル (Administration Client GUI のファイルを含む) が入っています。

ファイル	目的
Readme.txt	資料に記載されていない最新情報
mqiptSample.conf	サンプル構成ファイル
ssl/sslSample.pfx	テスト鍵リング・ファイル
ssl/sslSample.pwd	テスト鍵リング・ファイル用のパスワード・ファイル
ssl/sslCAdefault.pfx	サンプル認証局 (CA) 鍵リング・ファイル
ssl/sslCAdefault.pwd	サンプル認証局 (CA) 鍵リング・ファイル用のパスワード・ファイル
ssl/KeyMan.zip	KeyMan ユーティリティー
exits/ SampleOneRouteExit.java	サンプル・セキュリティー出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit の構成ファイル

ファイル	目的
exits/SampleRoutingExit.java	サンプル・セキュリティー出口
exits/SampleRoutingExit.conf	SampleRoutingExit の構成ファイル
exits/SampleSecurityExit.java	サンプル・セキュリティー出口
lib/libmqiptqos.so	TQoS のダミー・ライブラリー
bin/mqiptQoS	実際の TQoS ライブラリーを使用する場合
lib/MQipt.jar	ランタイム、クラス、およびプロパティー・ファイルが入っている
lib/ADV_mqipt_normal. class	「通常」モード用 Network Dispatcher アドバイザー
lib/ADV_mqipt_replace. class	「置換」モード用 Network Dispatcher アドバイザー
lib/mqipt1414Sample.ssl	Network Dispatcher アドバイザー用のサンプル・トリガー・ファイル
lib/libiptqos.so	Quality of Service サポート用のランタイム・ライブラリー
bin/mqipt	コマンド行から MQIPT を実行するためのショートカット
bin/mqiptAdmin	MQIPT を停止し、ファイル情報をリフレッシュするためのショートカット
bin/mqiptPW	鍵リング・ファイルをオープンするために使用されるパスワードを暗号化する
bin/mqiptVersion	MQIPT のバージョン番号の表示
bin/mqiptService	システム始動時に MQIPT が自動的に開始されるようにするための MQIPT のインストール
bin/mqiptEnv	`mqipt.jar` ファイルのロケーションを定義し、他のスクリプトでのみ使用する。
web/MQIPTServlet.war	サーブレット・バージョン用の Web アーカイブ・ファイル
doc/<lang>/html/ <filename>.zip	HTML 形式の「*internet pass-thru*」マニュアルのマスター・ファイル。ソフトコピー文書の詳細については、 183 ページの『参照文献』を参照。
lib/guiadmin.jar	Administration Client GUI 用のランタイム、クラス、およびプロパティー・ファイルが入っている
bin/mqiptGui	コマンド行から Administration Client GUI を実行するためのショートカット
bin/customSample. properties	Administration Client の外観およびアクセシビリティーをカスタマイズするためのサンプル・ファイル

internet pass-thru のセットアップ

MQIPT を初めて開始する場合は、その前に、`mqiptSample.conf` サンプル構成ファイルを `mqipt.conf` にコピーしてください。詳細については、 73 ページの『第 19 章 internet pass-thru の管理と構成』を参照してください。

コマンド行からの internet pass-thru の開始

ルートとしてログインし、ディレクトリーを `bin` ディレクトリーに変えます。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqipt ..
```

オプションを指定しないで mqipt スクリプトを実行すると、“.” のデフォルト・ロケーションが構成ファイル (mqipt.conf) に使用されます。異なるロケーションを指定するには、次のようにします。

```
mqipt <directory name>
```

MQIPT の状況を示すメッセージがコンソールに表示されます。エラーが起こった場合は、155 ページの『問題判別』を参照してください。以下のメッセージは、MQIPT が正常に開始された場合の例です。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from /opt/mqipt/mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path /opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1418 ready for connection requests
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file /opt/mqipt/KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

MQIPT を初めて呼び出すときは、mqipt ホーム・ディレクトリーの以下のサブディレクトリーが自動的に作成されます。

- 接続ログが保管されている ″logs″ ディレクトリー
- 任意の First Failure Support Technology (FFST) とトレース・レコードが書き込まれる ″errors″ ディレクトリー

internet pass-thru の自動開始

システム開始時に MQIPT が自動的に開始されるようにするには、mqiptService スクリプトを実行します。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqiptService -install
```

MQIPT をサービスとして開始すると、console.log ファイルが logs サブディレクトリーに書き込まれます。このサブディレクトリーは、MQIPT を初めて実行するときに作成されるので、MQIPT をサービスとして実行する前に、少なくとも 1 回はそれを実行しておく必要があります。

MQIPT が自動的に開始されないようにするには、次のようにします。

```
cd /opt/mqipt/bin
mqiptService -remove
```

コマンド行からの Administration Client の開始

コマンド・プロンプトをオープンし、ディレクトリーを bin ディレクトリーに変えて、mqiptGui を実行します。たとえば、以下のとおりです。

```
cd /opt/mqipt/bin
mqiptGui
```

Administration Client がファイアウォール経由で外部の MQIPT に接続できるようにするには、以下のように、ホスト名またはアドレス、およびポート番号を指定します。

```
mqiptGui <socksHostName <socksPort>>
```

デフォルトの socksPort は 1080 です。

Administration Client の状況が、Administration Client のメイン・ウィンドウにメッセージで示されます。

internet pass-thru のアンインストール

67 ページの『internet pass-thru の自動開始』に説明されているように、MQIPT をシステムからアンインストールする前に、それが自動的に開始されないようにしてください。 ルートとしてログインし、次のように swremove コマンドを実行します。

```
rpm -e WebSphereMQ-IPT-1.3.0-0
```

第 18 章 一般的な UNIX のインストール

すべての共通 MQIPT ファイルのディスク・イメージは、tar ファイルで提供され、一般的に使用できます。このファイルの目的は、独自のインストール・イメージを提供するという形で MQIPT がサポートしない UNIX プラットフォームに MQIPT をインストールできるようにするものです。その意図は、tar ファイルを指定したロケーションにアンパックして多少の変更を加えることにより、Java 1.4 をサポートする任意のプラットフォームで MQIPT をインプリメントできるようにすることです。bin サブディレクトリーにある、mqiptEnv スクリプトは、インストール済みファイルのロケーションを反映するために変更する必要がある場合があります。

ファイルのダウンロードとインストール

MQIPT は、次の WebSphere MQ SupportPac Web ページからダウンロードできます。

```
http://www.ibm.com/webspheremq/supportpacs
```

ダウンロードの指示を実行してください。

ルートとしてログインして、次の例のように、ターゲット・ディレクトリーに ms81.tar をアンパックします。

```
login root
cd /
mkdir mqipt
cd mqipt
cp /tmp/ms81.tar /mqipt/.
tar xvf ms81.tar
```

この例では、ms81.tar が /tmp ディレクトリーにダウンロードされていることを前提にしています。

MQIPT には、Administration Client GUI 用のファイルを含む、次の表に示されているファイルが入っています。

ファイル	目的
Readme.txt	資料に記載されていない最新情報
mqiptSample.conf	サンプル構成ファイル
ssl/sslSample.pfx	テスト鍵リング・ファイル
ssl/sslSample.pwd	テスト鍵リング・ファイル用のパスワード・ファイル

ファイル	目的
ssl/sslCAdefault.pfx	サンプル認証局 (CA) 鍵リング・ファイル
ssl/sslCAdefault.pwd	サンプル認証局 (CA) 鍵リング・ファイル用のパスワード・ファイル
ssl/KeyMan.zip	KeyMan ユーティリティー
exits/ SampleOneRouteExit.java	サンプル・セキュリティー出口
exits/ SampleOneRouteExit.conf	SampleOneRouteExit の構成ファイル
exits/SampleRoutingExit.java	サンプル・セキュリティー出口
exits/SampleRoutingExit.conf	SampleRoutingExit の構成ファイル
exits/SampleSecurityExit.java	サンプル・セキュリティー出口
lib/MQipt.jar	ランタイム、クラス、およびプロパティー・ファイルが入っている
lib/ADV_mqipt_normal. class	「通常」モード用 Network Dispatcher アドバイザー
lib/ADV_mqipt_replace. class	「置換」モード用 Network Dispatcher アドバイザー
lib/mqipt1414Sample.ssl	Network Dispatcher アドバイザー用のサンプル・トリガー・ファイル
bin/mqipt	コマンド行から MQIPT を実行するためのショートカット
bin/mqiptAdmin	MQIPT を停止し、ファイル情報をリフレッシュするためのショートカット
bin/mqiptPW	鍵リング・ファイルをオープンするために使用されるパスワードを暗号化する
bin/mqiptVersion	MQIPT のバージョン番号の表示
bin/mqiptService	システム始動時に MQIPT が自動的に開始されるようにするための MQIPT のインストール
bin/mqiptEnv	`mqipt.jar` ファイルのロケーションを定義し、他のスクリプトでのみ使用する。
web/MQIPTServlet.war	サーブレット・バージョン用の Web アーカイブ・ファイル
doc/<lang>/html/ <filename>.zip	HTML 形式の「*internet pass-thru*」マニュアルのマスター・ファイル。ソフトコピー文書の詳細については、 183 ページの『参照文献』を参照。
lib/guiadmin.jar	Administration Client GUI 用のランタイム、クラス、およびプロパティー・ファイルが入っている
bin/mqiptGui	コマンド行から Administration Client を実行するためのショートカット
bin/customSample. プロパティー	Administration Client の外観およびアクセシビリティーをカスタマイズするためのサンプル・ファイル

internet pass-thru のセットアップ

初めて MQIPT を開始する前に、サンプル・構成ファイル `mqiptSample.conf` を `mqipt.conf` にコピーします。詳細については、 73 ページの『第 19 章 internet pass-thru の管理と構成』を参照してください。

この例では、MQIPT が mqipt という名前のディレクトリーにアンパックされることを前提にしています。mqiptEnv スクリプトは、ランタイム・ライブラリーの新しいロケーションを使用して更新する必要があります。MQIPT_CP 変数のデフォルト値は、次のとおりです。

```
MQIPT_CP=/opt/mqipt/lib/MQipt.jar:/opt/mqipt/lib/guiadmin.jar
```

この例では、これは次の値に変更する必要があります。

```
MQIPT_CP=/mqipt/opt/mqipt/lib/MQipt.jar:/mqipt/opt/mqipt/lib/guiadmin.jar
```

また、ランタイム・スクリプトがあれば、それを使用する前に更新して、mqiptEnv スクリプトのロケーションの完全修飾パス名を変更する必要があります。たとえば、mqipt スクリプトを使用する前に、編集し、コメント Get classpath の後のステートメントを

```
/opt/mqipt/bin/mqiptEnv
```

から、次のように変更します。

```
/mqipt/opt/mqipt/bin/mqiptEnv
```

コマンド行からの internet pass-thru の開始

ルートとしてログインし、ディレクトリーを bin ディレクトリーに変更します。たとえば、以下のとおりです。

```
cd /mqipt/opt/mqipt/bin
mqipt ..
```

オプションを指定しないで mqipt スクリプトを実行すると、 “.” のデフォルト・ロケーションが構成ファイルに使用されます (mqipt.conf)。異なるロケーションを指定するには、次のようにします。

```
mqipt <directory name>
```

MQIPT の状況を示すメッセージがコンソールに表示されます。エラーが起こった場合は、 155 ページの『問題判別』を参照してください。以下のメッセージは、MQIPT が正常に開始された場合の例です。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from /mqipt/opt/mqipt/mqipt.conf
MQCPI008 Listening for control commands on port 1881
MQCPI011 The path /mqipt/opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1418 has started and will forward messages to :
MQCPI034 ....mqserver.company4.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1418 ready for connection requests
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....mqipt.company2.com(1415)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file /mqipt/opt/mqipt/KeyMan.pfx
MQCPI038 ......distinguished name(s) CN=*Doe O=IBM OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

MQIPT を初めて呼び出すと、以下の mqipt ホーム・ディレクトリーのサブディレクトリーが自動的に作成されます。

- 接続ログが保管されている ″logs″ ディレクトリー
- 任意の First Failure Support Technology (FFST) とトレース・レコードが書き込まれる ″errors″ ディレクトリー

internet pass-thru の自動開始

サービスを自動的に開始することは、プラットフォーム固有です。mqiptService スクリプトは、Sun Solaris システムで行われる例としてのみ提供されています。システム要件によっては、システム・サービスとして MQIPT をインストールするためにプラットフォーム固有のユーティリティーを使用する方が簡単な場合があります。

コマンド行からの Administration Client の開始

コマンド・プロンプトをオープンし、ディレクトリーを `bin` ディレクトリーに変更して、`mqiptGui` を実行します。たとえば、以下のとおりです。

```
cd /mqipt/opt/mqipt/bin
../mqiptGui
```

Administration Client がファイアウォール経由で外部の MQIPT に接続できるようにするには、以下のように、ホスト名またはアドレス、およびポート番号を指定します。

```
mqiptGui <socksHostName <socksPort>>
```

デフォルトの `socksPort` は 1080 です。

Administration Client の状況が、Administration Client のメイン・ウィンドウにメッセージで示されます。

internet pass-thru のアンインストール

MQIPT はシステム・インストール可能イメージを使用してインストールされていないため、インストールされたディレクトリー構造を削除することによってアンインストールできます。

MQIPT がシステム・サービスとして実行するように構成されている場合には、コードをアンインストールする前にそのサービスを除去してください。

第 19 章 internet pass-thru の管理と構成

MQIPT の構成を行うには、mqipt.conf 構成ファイルに変更を加えます。この変更を行うには、Administration Client を使用する (この方法を推奨) か、または選択したエディターを使用します。この章では、関連する参照情報を使用して、これらの 2 つの手法について説明します。

- 『internet pass-thru Administration Client の使用』
- 78 ページの『internet pass-thru 行モード・コマンド』
- 79 ページの『構成参照情報』

internet pass-thru Administration Client の使用

Administration Client を使用して、1 つまたは複数の MQIPT を構成したり更新したりできます。 Application Client は、MQIPT のグローバル・プロパティーと経路固有のプロパティーを表示します。

Administration Client は、Java 1.4 を前提条件としないことに注意してください。

Administration Client のローカル側に保管される唯一のデータは MQIPT のリストであり、このリストは client.conf というファイルに入っています。グローバル・プロパティーと経路プロパティーは、常に、MQIPT から取り出されてから、Administration Client に表示されます。

Administration Client の開始

Administration Client を開始する場合は、MQIPT の bin サブディレクトリーに入っている mqiptGui スクリプトを使用します。 Administration Client の開始に関する説明については、各プラットフォームのインストールの章を参照してください。

Administration Client の初回の開始時には、ダイアログ・ボックスが表示されて、ユーザーは MQIPT との接続情報の入力を求められます。必要な情報は、以下のとおりです。

「MQIPT Name (MQIPT の名前)」
この MQIPT の説明に使用する名前。この情報は必須ではありませんが、入力をお勧めします。

「Network Address (ネットワーク・アドレス)」
MQIPT が常駐するシステムのアドレス。ネーム・サーバーによって認識された名前、小数点付き 10 進数アドレス、またはローカル・ホスト (MQIPT がクライアントと同じマシンにある場合) のいずれか。

「Command Port (コマンド・ポート)」
MQIPT がコマンドを listen するポートの番号。

「Timeout (タイムアウト)」
Administration Client が MQIPT との接続を待機する時間 (秒数)。できるだけこの値を小さくして、ウィンドウの最新表示時間を減らします。

「Access Password (アクセス・パスワード)」
MQIPT と通信するとき使用するパスワード。このフィールドは、パスワード検査が有効になっている場合にのみ入力します。 (AccessPW が MQIPT 構成ファイルに提供されていて、かつヌル・ストリング以外の値である場合に、パスワード検査が有効です。)

「Save Password (パスワードの保管)」
このチェック・ボックスがブランクのままであれば、パスワードは、このセッションの期間中、または MQIPT を除去するまで記憶されています。このチェック・ボックスを選択すると、パスワードは、将来のセッションのために保管されます。

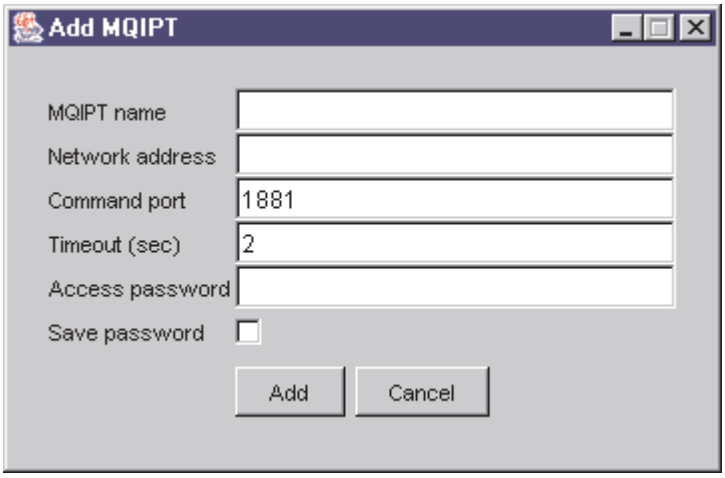

図 8. MQIPT への初回アクセス時のウィンドウ

MQIPT の管理

MQIPT の更新は一度に 1 つしか行えないため、リストから別の MQIPT を選択した場合は、未処理の変更をすべて適用してから次の作業へ進まなければなりません。いずれかのプロパティーに変更を加えても、「Apply (適用)」メニュー・オプションを使用するまでは MQIPT は影響を受けません。

リストから MQIPT を選択すると、グローバル・プロパティーと経路プロパティーが MQIPT から取り出されます。 MQIPT が稼働していない場合や、誤りの CommandPort が指定された場合は、エラー・メッセージが出ます。ホスト名や CommandPort の変更は、「Connection (接続)」メニュー・オプションから行えます。

リストの MQIPT をダブルクリックすると、経路のリストが表示されます。経路を選択すると、そのプロパティーが表示されます。プロパティーは、ユーザーの要件に合わせて調整できます。

変更を適用すると、構成ファイルは、タイム・スタンプを記録されて MQIPT へ戻され、変更内容が即時に有効になります。既存のコメント行はすべて消失します。

経路を追加するには、「Add Route (経路の追加)」メニュー・オプションを使用します。この新規経路では、グローバル・プロパティーによって定義されたデフォルトのプロパティー・セットが表示されます。

プロパティーの継承

Administration Client で MQIPT や経路のプロパティーを設定する方法には、以下のような階層があります。

1. どのプロパティーにもデフォルト値があり、プロパティーが構成ファイルに記述されていない場合や、Administration Client のユーザー処置によって明確に設定されていない場合は、このデフォルト値が使用されます。
2. MQIPT 全体に対して設定されたグローバル・プロパティーは、その適用を禁止する特定の経路情報がない限り、各 MQIPT のすべての経路で使用されます。つまり、構成ファイルの場合、追加のプロパティーが経路スタンザに設定されない限り、グローバル・スタンザに設定されたプロパティーがすべての経路に伝搬されます。 Administration Client ユーザーによって MQIPT に設定されたプロパティーは、経路に対して別途プロパティーが設定されない限り、すべての経路に伝搬されます。
3. ある経路に対して設定されたすべての値は、デフォルト値やグローバル設定値とは関係なく、その経路用として維持されます。

ファイル・メニュー・オプション

「File (ファイル)」メニューを選択すると、ツリー管理に関連するオプションのほとんどが表示されます。

「Add MQIPT (MQIPT の追加)」
クライアントを初めて使用するときに表示されるダイアログと同じダイアログが表示されます (73 ページの『Administration Client の開始』を参照)。

「Remove MQIPT (MQIPT の除去)」
現在強調表示されている MQIPT を Administration Client のツリーだけから除去します。この除去によって MQIPT の実行が影響を受けることはありません。

「Save Configuration (構成の保管)」
ツリーの MQIPT ノードを Administration Client の構成ファイルに保管して、それを次回に開始するときにこれらのノードを読み取れるようにします。MQIPT ノードのみが保管されます。グローバル・プロパティーと経路プロパティーは、常に、MQIPT から取り出されます。

「Quit (終了)」
Administration Client の実行を停止します。ただし、Administration Client は、まず、ツリーまたは現行 MQIPT が変更されたかどうかを調べます。このうちのいずれか、または両方が変更された場合は、1 つまたは複数のダイアログが表示され、クライアントの保管、または MQIPT への変更の適用、あるいはその両方を行いたいかどうかを尋ねられます。

MQIPT メニュー・オプション

「Connection (接続)」
MQIPT のアクセス・パラメーターを変更します。変更結果はツリー・ビューに

示されます。ツリー・ビューでは、 73 ページの『Administration Client の開始』に示されているようなウィンドウが表示されます。

「Password (パスワード)」
リモート MQIPT のパスワード・プロパティーを変更します。このアクションによりパスワード・ダイアログが表示され、ユーザーは、以下の入力を行うよう求められます。

- 「**Current Password (現行パスワード)**」: 不正使用のチェックのために、現行パスワードを示す必要があります。それを示さないとその変更を行えません。現在有効なパスワードがない場合は、このフィールドをブランクにされます。
- 「**New Password (新規パスワード)**」: 新規パスワードを入力します。この MQIPT でパスワードの使用を止めたい場合は、ブランクにしておきます。
- 「**New Password Again (再度新規パスワード)**」: 「New Password (新規パスワード)」フィールドへの入力ミスを防ぐために、同じ情報を再度入力するよう要求されます。
- 「**Save Password (パスワードの保管)**」: この MQIPT の他のアクセス・プロパティーと一緒に、新規パスワードをローカル側に保管するかどうかを決定するために使用されます。

「Add Route (経路の追加)」
選択した MQIPT に経路を追加します。詳細については、 77 ページの図 9 を参照してください。各経路は、MQIPT 用の固有な ListenerPort を持っていなければなりません。

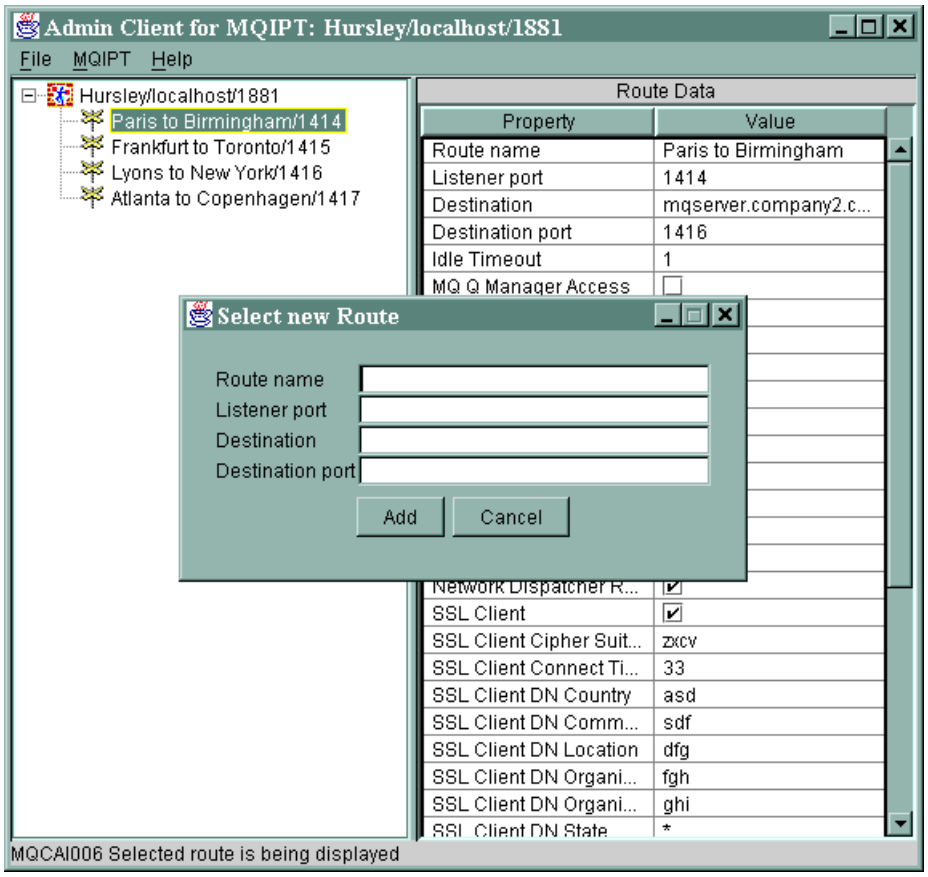

図 9. 経路の追加

「Delete Route (経路の削除)」
選択された経路を MQIPT から削除します。この削除は、「Apply (適用)」メニュー・オプションが使用されるまで、MQIPT に影響を与えません。

「Apply (適用)」
MQIPT の構成に対して行った変更に間違いがなければ、このオプションで新規の構成ファイルが MQIPT に送られ、MQIPT はそれを保管します。新規設定は直ちに有効になります。

「Refresh (リフレッシュ)」
選択された MQIPT から構成ファイルを読み取り、リフレッシュを行います。

「Stop (停止)」
STOP コマンドを MQIPT に送って、実行を停止するよう指示します。このコマンドが実行されたならば、MQIPT との連絡が切断されます。グローバル・プロパティーである RemoteShutdown がオンにされない限り、このコマンドは無視されます。

経路情報は、MQIPT グローバル情報と同じ方法で更新できます。経路のプロパティーを変更した場合は、その変更が有効になる前にそれを適用する必要があります。これを行うには、「MQIPT/Apply (MQIPT/ 適用)」メニュー・オプションを選択するか、または構成の保管を要求されたときに「はい」と答えます。

ヘルプ・メニュー・オプション

ヘルプ

Netscape を使用して、Administration Client の使用方法に関する情報を表示し、左方のペインから、「Administering and configuring internet pass-thru (internet pass-thru の管理と保管)」を選択します。 Administration Client を使用する前に、<lang>/html サブディレクトリーに入っているファイルを圧縮解除する必要があります。

製品情報

Administration Client のバージョンに関する情報が入っているウィンドウを表示します。

internet pass-thru 行モード・コマンド

Administration Client を使用しない場合は、行モード・コマンドを使用して internet pass-thru の管理と構成を行うことができます。

行モード・コマンドによる internet pass-thru の管理

選択したエディターを使用して、自分の要件を満たすように mqipt.conf 構成ファイルを変更します。変更できるプロパティーのリストについては、79 ページの『構成参照情報』を参照してください。

mqipt.conf のグローバル・セクションに CommandPort の値が指定されている場合は、MQIPT はこのポートで以下の ASCII 管理コマンドを listen します。

```
mqiptAdmin -refresh {hostname {port} }      refresh コマンドを送信する
mqiptAdmin -stop    {hostname {port} }      stop コマンドを送信する
```

mqiptAdmin スクリプトは bin サブディレクトリーに入っています。

入っていなければ、デフォルトで、ホスト名が localhost になり、ポートが 1881 になります。

STOP

MQIPT は、すべての接続をクローズし、着信接続の listen を停止してから終了します。 Administration Client の「MQIPT/Stop (MQIPT/停止)」メニュー・オプションを使用した場合も同じ結果が得られます。 mqipt.conf ファイルに RemoteShutDown=true を指定していない限り、このコマンドは無視されます。

REFRESH

MQIPT は mqipt.conf を再読み取りします。 MQIPT は以下の処理を行います。

- 現在アクティブなすべての経路に非アクティブのマークが付いている (または、それらの経路がすべて欠落している) のを検出した場合、MQIPT は、それらの経路をクローズし、これらの経路での着信接続の listen を停止します。
- 現在実行していない、構成ファイルにアクティブのマークが付いている経路を検出した場合、MQIPT はそれらの経路を始動します。
- 現在実行中の経路の構成パラメーターが変更されているのを検出した場合、MQIPT は、変更値をそれらの経路に適用します。可能であれば (たとえば、

トレースの設定値が変更された場合)、MQIPT は、稼働している接続を中断することなくこの操作を実行します。パラメーターの変更 (たとえば、宛先の変更) によっては、MQIPT がすべての接続をクローズしないと、変更内容を有効にしたり、経路を再始動したりできないものがあります。

Administration Client が MQIPT の設定値を一切変更していなければ、Administration Client の「MQIPT/Apply (MQIPT/ 適用)」メニュー・オプションを使用した場合も同じ結果が得られます。

Windows では、これらの管理機能は「Start (スタート)」->「Programs (プログラム)」メニューからも使用できます。

構成参照情報

MQIPT は、mqipt.conf と呼ばれる構成ファイルを使用して、経路を定義して MQIPT サーバーのアクションを制御します。このファイルは、セクションのセットから構成されています。 1 つのグローバル・セクションが設けられているほか、MQIPT を介して定義されている各経路ごとに 1 つずつセクションがあります。

それぞれのセクションには、名前 / 値のプロパティー・ペアが含まれています。プロパティーには、グローバル・セクションにしか現れないもの、経路セクションにしか現れないもの、また、経路セクションとグローバル・セクションの両方に現れるものがあります。あるプロパティーが経路セクションとグローバル・セクションの両方に現れる場合は、経路セクションのプロパティー値がグローバル・セクションの値をオーバーライドしますが、そのオーバーライドは当該経路についてだけ行われます。このようにして、グローバル・セクションを使用してデフォルト値を設定することにより、それらのデフォルト値を、個々の経路セクションで設定されていないプロパティーに使用することができます。

グローバル・セクションは、`[global]` の文字が入っている行で始まり、最初の経路セクションが始まるところで終了します。グローバル・セクションは、ファイル内のすべての経路セクションの先頭になければなりません。各経路セクションは、`[route]` の文字が入っている行で始まり、次の経路セクションが始まるところ、または構成ファイルの末尾に達したところで終了します。

認識されないすべてのキーワード名 (つまり、本書で定義された名前に含まれていない名前のすべての名前 / 値のペア) は無視されます。経路セクションに現れる名前 / 値のペアが認識済みの名前を持っているが、無効な値を持っている場合 (たとえば、`MinConnectionThreads=x` または `HTTP=unsure`)、その経路は使用不可になります (つまり、着信接続を一切 listen しません)。グローバル・セクションに現れる名前 / 値のペアが認識済みの名前を持っているが、無効な値を持っている場合は、すべての経路が使用不可になり、MQIPT は開始されません。プロパティーが `true` と `false` の値をとるものとしてリストされている場合は、大文字と小文字が混在する任意の文字を使用できます。

プロパティーへの変更は、mqipt.conf ファイルを編集することによってまたは Administration Client GUI を使用することによって行うことができます。変更を適用するために、アドミニストレーターは Administration Client GUI からまたは mqiptAdmin スクリプトを使用することによって、リフレッシュ・コマンドを出すことができます。

特定のプロパティーを変更しても、他のプロパティーがすでに使用可能になっている場合には、1 つの経路を再始動することになるだけです。たとえば、HTTP プロパティーを変更した場合、その HTTP プロパティーが使用可能にされている場合にのみ、変更が有効になります。

経路を再始動すると、既存の接続は終了します。この振る舞いをオーバーライドするには、RouteRestart プロパティーを false に設定します。これによって、経路が再始動できなくなり、RouteRestart プロパティーが再度使用可能になるまで既存の接続はアクティブのままになります。

いくつかの簡単な構成をセットアップする方法については、99 ページの『第 20 章 internet pass-thru の使用開始』を参照してください。サンプル構成については、MQIPT のホーム・ディレクトリーに入っている `mqiptSample.conf` ファイルを参照してください。

プロパティーの要約

表 3 は、以下のものを示しています。

- すべてのプロパティー
- そのプロパティーがグローバル・セクション、経路セクション、あるいはその両方のいずれに適用されるか
- あるプロパティーがグローバル・セクションにも経路セクションにも含まれていない場合は、デフォルト値が使用されます。

表 3. 構成プロパティーの要約

プロパティーの名前	グローバル	経路	デフォルト
AccessPW	はい	いいえ	<null>
Active	はい	はい	true
ClientAccess	はい	はい	false
CommandPort	はい	いいえ	<null>
ConnectionLog	はい	いいえ	true
Destination	いいえ	はい	<null>
DestinationPort	いいえ	はい	1414
HTTP[6,7]	はい	はい	false
HTTPChunking[1]	はい	はい	false
HTTPProxy[1]	はい	はい	<null>
HTTPProxyPort[1]	はい	はい	8080
HTTPS[1]	はい	はい	false
HTTPServer[1]	はい	はい	<null>
HTTPServerPort[1]	はい	はい	<null>
IdleTimeout	はい	はい	0
IgnoreExpiredCRLs	はい	はい	false
LDAP	はい	はい	false
LDAPIgnoreErrors[10]	はい	はい	false
LDAPCacheTimeout[10]	はい	はい	24
LDAPSaveCRL[10]	はい	はい	false

表 3. 構成プロパティーの要約 (続き)

プロパティーの名前	グローバル	経路	デフォルト
LDAPServer1[10]	はい	はい	<null>
LDAPServer1Port[10]	はい	はい	389
LDAPServer1Userid[10]	はい	はい	<null>
LDAPServer1Password[10]	はい	はい	<null>
LDAPServer1Timeout[10]	はい	はい	0
LDAPServer2[10]	はい	はい	<null>
LDAPServer2Port[10]	はい	はい	389
LDAPServer2Userid[10]	はい	はい	<null>
LDAPServer2Password[10]	はい	はい	<null>
LDAPServer2Timeout[10]	はい	はい	0
ListenerPort	いいえ	はい	<null>
LocalAddress	はい	はい	<null>
LogDir (MQIPTServlet の場合にのみ有効)	いいえ	いいえ	<null>
MaxConnectionThreads	はい	はい	100
MaxLogFileSize	はい	いいえ	50
MinConnectionThreads	はい	はい	5
Name	いいえ	はい	<null>
NDAdvisor	はい	はい	false
NDAdvisorReplaceMode[4]	はい	はい	false
OutgoingPort	いいえ	はい	0
QMgrAccess	はい	はい	true
QoS (Linux でのみ使用可)	はい	はい	false
QosToCaller[9]	はい	はい	1
QosToDest[9]	はい	はい	1
RemoteShutdown	はい	いいえ	false
RouteRestart	はい	はい	true
SecurityExit	はい	はい	false
SecurityExitName[11]	はい	はい	<null>
SecurityExitPath[11]	はい	はい	<ipthome> ¥exits
SecurityExitTimeout[11]	はい	はい	5
SecurityManager	はい	いいえ	false
SecurityManagerPolicy	はい	いいえ	<null>
ServletClient[1]	はい	はい	false
SocksClient	はい	はい	false
SocksProxyHost[8]	はい	はい	<null>
SocksProxyPort[8]	はい	はい	1080
SocksServer[7]	はい	はい	false
SSLClient	はい	はい	false

表 3. 構成プロパティーの要約 (続き)

プロパティーの名前	グローバル	経路	デフォルト
SSLClientCAKeyRing[2]	はい	はい	<null>
SSLClientCAKeyRingPW[2]	はい	はい	<null>
SSLClientCipherSuites[2]	はい	はい	<null>
SSLClientConnectTimeout[2]	はい	はい	30
SSLClientDN_C[2]	はい	はい	"*"[5]
SSLClientDN_CN[2]	はい	はい	"*"[5]
SSLClientDN_L[2]	はい	はい	"*"[5]
SSLClientDN_O[2]	はい	はい	"*"[5]
SSLClientDN_OU[2]	はい	はい	"*"[5]
SSLClientDN_ST[2]	はい	はい	"*"[5]
SSLClientKeyRing[2]	はい	はい	<null>
SSLClientKeyRingPW[2]	はい	はい	<null>
SSLClientSiteDN_C[2]	はい	はい	"*"[5]
SSLClientSiteDN_CN[2]	はい	はい	"*"[5]
SSLClientSiteDN_L[2]	はい	はい	"*"[5]
SSLClientSiteDN_O[2]	はい	はい	"*"[5]
SSLClientSiteDN_OU[2]	はい	はい	"*"[5]
SSLClientSiteDN_ST[2]	はい	はい	"*"[5]
SSLClientSiteLabel[2]	はい	はい	<null>
SSLProxyMode	はい	はい	false
SSLServer[6]	はい	はい	false
SSLServerAskClientAuth[3]	はい	はい	false
SSLServerCAKeyRing[3]	はい	はい	<null>
SSLServerCAKeyRingPW[3]	はい	はい	<null>
SSLServerCipherSuites[3]	はい	はい	<null>
SSLServerDN_C[3]	はい	はい	"*"[5]
SSLServerDN_CN[3]	はい	はい	"*"[5]
SSLServerDN_L[3]	はい	はい	"*"[5]
SSLServerDN_O[3]	はい	はい	"*"[5]
SSLServerDN_OU[3]	はい	はい	"*"[5]
SSLServerDN_ST[3]	はい	はい	"*"[5]
SSLServerKeyRing[3]	はい	はい	<null>
SSLServerKeyRingPW[3]	はい	はい	<null>
SSLServerSiteDN_C[3]	はい	はい	"*"[5]
SSLServerSiteDN_CN[3]	はい	はい	"*"[5]
SSLServerSiteDN_L[3]	はい	はい	"*"[5]
SSLServerSiteDN_O[3]	はい	はい	"*"[5]
SSLServerSiteDN_OU[3]	はい	はい	"*"[5]
SSLServerSiteDN_ST[3]	はい	はい	"*"[5]
SSLServerSiteLabel[3]	はい	はい	<null>

表 3. 構成プロパティーの要約 (続き)

プロパティーの名前	グローバル	経路	デフォルト
Trace	はい	はい	0
UriName (デフォルト設定の詳細については、 98 ページの『UriName』を参照。)[1]	はい	はい	

注:

1. これらのプロパティーを有効にするには、HTTP を true に設定します。
2. これらのプロパティーを有効にするには、SSLClient を true に設定します。
3. これらのプロパティーを有効にするには、SSLServer を true に設定します。
4. これらのプロパティーを有効にするには、NDAdvisor を true に設定します。
5. ″*″ 記号はワイルドカードを表します。
6. HTTP と SSLServer を一緒に使用することはできません。 HTTP プロパティーは、正方向接続の定義にのみ使用されます。 ListenerPort への着信データは自動的に検出されるため、SSLServer を設定するとランタイム例外が発生します。
7. HTTP と SocksServer を一緒に使用することはできません。 HTTP プロパティーは、正方向接続の定義にのみ使用されます。 ListenerPort への着信データは自動的に検出されるため、SocksServer を設定するとランタイム例外が発生します。
8. これらのプロパティーを有効にするには、SocksClient を true に設定します。
9. これらのプロパティーを有効にするには、QoS を true に設定します。
10. これらのプロパティーを有効にするには、LDAP を true に設定します。
11. これらのプロパティーを有効にするには、SecurityExit を true に設定します。

グローバル・セクション参照情報

グローバル・セクションには、ListenerPort、Destination、DestinationPort、Name および OutgoingPort のほかに、以下のプロパティーと、 84 ページの『経路セクション参照情報』に示されているすべてのプロパティーを含めることができます。

AccessPW

Administration Controller がコマンドを MQIPT に送信するときに使用するパスワード。このプロパティーがない場合やブランクに設定されている場合は、検査は行われません。

CommandPort

MQIPT が `mqiptAdmin` ユーティリティーまたは Administration Client からの構成コマンドを listen する TCP/IP ポート。 Administration Client からのコマンド・ポートは、他のすべてのプロパティーと同じ方法で変更できます。ただし、接続プロパティーは変更しないでください。新規のセットアップを MQIPT に適用すると、Administration Client が自動的に接続プロパティーを変更します。

`CommandPort` プロパティーがない場合は、MQIPT は構成コマンドを listen しません。コマンド・ポートで listen したい場合は、1881 を使用することをお勧め

します。 Administration Client は CommandPort に対するデフォルト値を持っていませんが、行モード・コマンドを使用する場合、1881 がデフォルト値になります。

ConnectionLog
true または false のいずれか。 true であれば、MQIPT はすべての接続試行 (成功またはそれ以外) を logs サブディレクトリーにログ記録し、切断イベントを mqiptYYYYMMDDHHmmSS.log ファイルにログ記録します。デフォルト値は true です。このプロパティーが true から false に変更されると、MQIPT は既存の接続ログをクローズして新規の接続ログを作成します。プロパティーが true にリセットされたとき、新規の接続ログが使用されます。

MaxLogFileSize
接続ログ・ファイルの最大サイズ (KB で指定)。ファイル・サイズがこの最大値を超えると、バックアップ・コピー (mqipt.back) が作成され、新規ファイルが開始されます。保管できるバックアップ・ファイルは 1 つだけです。したがって、メイン・ログ・ファイルがいっぱいになると、それ以前のバックアップはすべて消去されます。デフォルト値は 50 で、最小許可値は 5 です。

RemoteShutDown
true または false のいずれか。 true の場合 (およびコマンド・ポートがある場合) は、コマンド・ポートで STOP コマンドを受け取るたびに MQIPT がシャットダウンします。デフォルト値は false です。

SecurityManager
MQIPT のこのインスタンスに対して Java Security Manager を使用可能にするには、このプロパティーを true に設定します。このプロパティーは、正しい許可が付与されることを前提にしています。詳細については、 33 ページの『Java Security Manager』を参照してください。このプロパティーのデフォルト値は false です。

SecurityManagerPolicy
ポリシー・ファイルの完全修飾名。このプロパティーが設定されていないと、デフォルトのシステムとユーザー・ポリシー・ファイルが使用されます。 Java Security Manager がすでに使用可能になっている場合には、このプロパティーを変更しても Java Security Manager が使用不可にされ、再度使用可能にされるまでは、有効になりません。

経路セクション参照情報

経路セクションには、以下のプロパティーが含まれている場合があります。

Active
この経路は、Active の値が true に設定されている場合にのみ着信接続を受け入れます。つまり、Active=false と設定すれば、経路セクションを構成ファイルから削除しなくても、宛先へのアクセスを一時的にシャットオフすることができます。このプロパティーを false に変更すると、経路は、REFRESH コマンドを出したときに停止します。この経路へのすべての接続は終了します。

ClientAccess
この経路は、ClientAccess の値が true に設定されている場合にのみ着信クライアント・チャネル接続を可能にします。クライアント要求のみ、キュー・マネージャー要求のみ、または両方のタイプの要求を受け入れるように、MQIPT を構

成することができる点に注意してください。このプロパティーは、QMgrAccess プロパティーと一緒に使用してください。このプロパティーを false に変更すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

Destination

この経路の接続先キュー・マネージャー (または後続の MQIPT) のホスト名 (またはドット 10 進 IP アドレス)。各経路セクションには、明示的な Destination 値が含まれて**いなければなりません**。同一 Destination を指す複数の経路セクションを持つことができます。ある経路がこのプロパティーの変更によって影響を受けた場合は、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

DestinationPort

この経路の接続先 Destination ホスト上のポート。複数の経路が Destination と DestinationPort の同一の組み合わせを指すことは有効です。各経路セクションには、明示的な DestinationPort 値が含まれて**いなければなりません**。ある経路がこのプロパティーの変更によって影響を受けた場合は、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

HTTP

アウトバウンド HTTP トンネル操作要求を行う (つまり、HTTP を介して別の MQIPT と通信する) 経路の場合に、これを true に設定します。 WebSphere MQ キュー・マネージャーへ向かう経路の場合は false に設定します。このプロパティーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。 HTTP チャンク操作を使用するには、このプロパティーを true に設定します。このプロパティーは、以下のプロパティーと一緒に使用できます。

- QoS
- SocksClient
- SSLClient
- SSLProxyMode

HTTPChunking

チャンク操作と一緒に HTTP トンネル操作を使用してアウトバウンド要求を行う経路について、これを true に設定します。 HTTP プロパティーも true に設定する必要があります。 HTTP チャンク操作を使用しないときは false に設定します。このプロパティーを変更 (および HTTP を true に設定) すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

HTTPProxy

この経路のすべての接続が使用する HTTP プロキシーのホスト名 (またはドット 10 進 IP アドレス)。 HTTPServer も定義した場合には、通常の POST の代わりに CONNECT 要求が HTTPProxy に対して出されます。このプロパティーを変更 (および HTTP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

HTTPProxyPort

HTTP プロキシーで使用するポート・アドレス。 HTTPS が true に設定されていて、HTTPServer がなく、デフォルトが 443 である場合を除いて、デフォルト値は 8080 になります。このプロパティーを変更 (および HTTP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

HTTPServer

この経路のすべての接続が使用する HTTP サーバーのホスト名 (またはドット 10 進 IP アドレス)。このプロパティーを変更 (および HTTP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

HTTPS

HTTPS 要求を行うためにこのプロパティーを使用可能にします。HTTP プロパティーも使用可能にする必要があります。このプロパティーを変更 (および HTTP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

HTTPServerPort

HTTP サーバーで使用するポート・アドレス。 HTTPS が true に設定されていて、デフォルトが 443 である場合を除いて、デフォルト値は 8080 になります。このプロパティーを変更 (および HTTP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

IdleTimeout

アイドル接続をクローズする時刻 (分)。キュー・マネージャー間チャネルは DISCINT プロパティーも持つことに注意してください。 IdleTimeout パラメーターを設定した場合は、DISCINT をメモにとっておきます。 0 の値はアイドル・タイムアウトがないことを示します。このプロパティーの変更結果が有効になるのは、経路を再始動したときだけです。

IgnoreExpiredCRLs

期限切れの CRL を無視するには、このプロパティーを true に設定します。デフォルト値は false です。

> **重要**
>
> このプロパティーを使用可能にした場合、取り消された証明書を使用して SSL 接続を行うこともできます。

LDAP

SSL 接続の使用時に LDAP サーバーの使用を使用可能にするためにこのプロパティーを true に設定します。 MQIPT は LDAP サーバーを使用して CRL と ARL を取り出します。このプロパティーが有効になるには、SSLClient または SSLServer プロパティーも使用可能にする必要があります。

LDAPIgnoreErrors

LDAP 検索の実行時に接続またはタイムアウト・エラーを無視するには、このプロパティーを true に設定します。MQIPT が正常に検索を実行できない場

合、このプロパティーを使用可能にしていない限り、クライアント接続は完了できません。正常な検索は、CRL が取り出されたかまたは指定した CA で使用可能な CRL がないことを意味します。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

重要

このプロパティーを使用可能にした場合、取り消された証明書を使用して SSL 接続を行うこともできます。

LDAPCacheTimeout

LDAP サーバーから CRL が取り出されると、一時キャッシュにある MQIPT に内部的に保管されます。このキャッシュのエントリーは、このプロパティーが定する、特定のタイムアウトの後で有効期限が切れます。デフォルト値は 24 時間です。タイムアウト値を 0 に指定すると、経路を再始動するまではキャッシュのエントリーの有効期限が切れません。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPSaveCRL

LDAP サーバーから取り出された CRL があればそれで所定の鍵リング・ファイルを更新するために、このプロパティーを true に設定します。鍵リング・ファイルは、SSLClientKeyRing、SSLClientCAKeyRing、SSLServerKeyRing および SSLServerCAKeyRing のプロパティーを使用して指定されます。これは、MQIPT には鍵リング・ファイルへの書き込みアクセス権限が必要であることを意味します。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer1

メイン LDAP サーバーのホスト名または IP アドレスにこのプロパティーを設定します。LDAP を使用可能にした場合、このプロパティーを設定する必要があります。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer1Port

メイン LDAP サーバーの listen ポート・アドレスにこのプロパティーを設定します。そのデフォルト値は 389 です。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer1Userid

メイン LDAP サーバーのアクセスに必要なユーザー ID にこのプロパティーを設定します。メイン LDAP サーバーをアクセスするための許可が必要である場合、このプロパティーを設定する必要があります。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer1Password
メイン LDAP サーバーのアクセスに必要なパスワードにこのプロパティーを設定します。LDAPServer1Userid を使用可能にした場合、このプロパティーを設定する必要があります。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer1Timeout
メイン LDAP サーバーからの応答を MQIPT が待機する秒数にこのプロパティーを設定します。そのデフォルト値は 0 で、接続がタイムアウトにならないことを意味します。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer2
バックアップ LDAP サーバーのホスト名または IP アドレスにこのプロパティーを設定します。このプロパティーはオプショナルです。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer2Port
バックアップ LDAP サーバーの listen ポート・アドレスにこのプロパティーを設定します。そのデフォルト値は 389 です。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer2Userid
バックアップ LDAP サーバーのアクセスに必要なユーザー ID にこのプロパティーを設定します。バックアップ LDAP サーバーをアクセスするための許可が必要である場合、このプロパティーを設定する必要があります。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer2Password
バックアップ LDAP サーバーのアクセスに必要なパスワードにこのプロパティーを設定します。LDAPServer2 を使用可能にした場合、このプロパティーを設定する必要があります。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LDAPServer2Timeout
バックアップ LDAP サーバーからの応答を MQIPT が待機する秒数にこのプロパティーを設定します。そのデフォルト値は 0 で、接続がタイムアウトにならないことを意味します。このプロパティーを変更 (および LDAP を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

ListenerPort
この経路が着信要求を listen するポート番号。各経路セクションには、明示的な ListenerPort 値が含まれて**いなければなりません**。さらに、各セクションに設定された ListenerPort 値は異なっていなければなりません。選択されたポート

が、同一ホストで稼働している他の任意の TCP/IP リスナーによってすでに使用されていれば、有効な任意のポート番号を使用することができます (ポート 80 および 443 を含む)。

LocalAddress
すべての接続をバインドするためのローカル IP アドレス。このプロパティーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

LogDir
このプロパティーを使用して、ログおよびトレース・ファイルのディレクトリー名を定義します。このプロパティーに対する変更は、MQIPTServlet が停止されて再始動されるまで、有効になりません。デフォルト値は <null> です。このプロパティーは MQIPTServlet に対してのみ有効です

MaxConnectionThreads
この経路が処理できる接続スレッドの最大数、つまり、同時接続の最大数。この限度に達すると、MaxConnectionThreads 値は、すべてのスレッドが使用中になっている場合にキューに入れられる接続の数を示します。その数を超えると、後続の接続要求は拒否されます。最小許可値は、1 または MinConnectionThreads の値のいずれか大きいほうです。このプロパティーを変更したために経路が影響を受ける場合は、REFRESH コマンドを出すときにこの新規値が使用されます。すべての接続がこの新規値を即時に使用します。経路は終了します。

MinConnectionThreads
接続スレッド (この経路の着信接続を処理するスレッド) の最小数。この数は、経路を開始するときに割り振られるスレッドの数であり、割り振られたスレッドの総数は、経路がアクティブになっている間、この値より小さくなることはありません。最小許可値は 0 であり、この値は MaxConnectionThreads に対して指定した値よりも小さくなければなりません。このプロパティーの変更結果が有効になるのは、経路を再始動したときだけです。

Name
経路を識別するためのオプショナル名。この名前は、コンソール・メッセージとトレース情報に現れます。このプロパティーの変更結果が有効になるのは、経路を再始動したときだけです。

NDAdvisor
経路がカスタム・アドバイザーからの要求に応答できるようにするには、Network Dispatcher によって管理される経路についてこのプロパティーを true に設定します。このプロパティーを false に変更すると、経路は、REFRESH コマンドを出したときに停止します。この経路へのすべての接続は終了します。NDAdvisorReplaceMode プロパティーを使用するには、このプロパティーを true に設定します。

NDAdvisorReplaceMode
Network Dispatcher カスタム・アドバイザーの「置換」モードを使用するには、このプロパティーを true に設定します。この経路の ListenerPort アドレスに対して mqipt_replace カスタム・アドバイザーを開始しておかなければなりません。「通常」モードを使用するには、このプロパティーを false に設定します。このプロパティーを使用するには、NDAdvisor プロパティーを true に設定します。

OutgoingPort
これは、出力接続が使用する開始のポート・アドレスです。ポート・アドレスの範囲は、この経路の MaxConnectionThread 値に一致します。デフォルト値を 0 にすると、システム定義のポート・アドレスを使用します。このプロパティーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

QMgrAccess
この経路で着信キュー・マネージャーのチャネル接続 (たとえば、送信側チャネル) を使用できるのは、QMgrAccess の値が true 値に設定されている場合だけです。このプロパティーを false に変更すると、経路は、REFRESH コマンドを出したときに停止します。この経路へのすべての接続は終了します。

QoS
この経路上のすべての接続に対して Quality of Service を使用できるようにするには、このプロパティーを true に設定します。このプロパティーは Linux でのみ使用可能にできます。このプロパティーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。このプロパティーは、以下のプロパティーと一緒に使用できます。

- HTTP
- SSLClient
- SSLProxyMode
- SSLServer

QosToCaller
このプロパティーは、MQIPT マシンから接続イニシエーターへのすべてのトラフィックについて優先順位を設定します。たとえば、このプロパティーの設定値 1 は低優先順位、2 は中間優先順位、3 は高優先順位を表します (デフォルトは 1 です)。このプロパティーを変更 (および QoS を true に設定) すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

QosToDest
このプロパティーは、MQIPT マシンから接続宛先 (Destination プロパティーで定義) へのすべてのトラフィックについて優先順位を設定します。たとえば、このプロパティーの設定値 1 は低優先順位、2 は中間優先順位、3 は高優先順位を表します (デフォルトは 1 です)。このプロパティーを変更 (および QoS を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

RouteRestart
他の経路のプロパティーが変更されて、REFRESH コマンドが出されたときに、経路が再始動されないようにするには、このプロパティーを false に設定します。このプロパティーのデフォルト値は true です。

SecurityExit
ユーザー定義のセキュリティー出口を使用可能にするには、このプロパティーを true に設定します。 このプロパティーのデフォルト値は false です。

SecurityExitName
ユーザー定義のセキュリティー出口のクラス名。SecurityExit を true に設定した場合、このプロパティーを設定する必要があります。このプロパティーを変更

(および SecurityExit を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SecurityExitPath
ユーザー定義のセキュリティー出口が入っている完全修飾パス名。このプロパティーを設定していない場合、このデフォルトはその出口のサブディレクトリーになります。このプロパティーは、ユーザー定義のセキュリティー出口が入っている jar ファイルの名前を定義することもできます。このプロパティーを変更 (および SecurityExit を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SecurityExitTimeout
MQIPT はこのタイムアウト値を使用して、接続要求の確認時に応答を待機する時間 (秒単位で) を判別します。デフォルト値は 5 秒です。このプロパティーを変更 (および SecurityExit を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

ServletClient
MQIPT サーブレットに接続するときにこのプロパティーを `true` に設定します。 HTTP プロパティーも `true` に設定する必要があります。このプロパティーを変更 (および HTTP を `true` に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。

SocksClient
経路を Socks クライアントとして機能させ、Socks プロキシーを介するすべての接続を SocksProxyHost および SocksProxyPort プロパティーを使用して定義させるようにするには、このプロパティーを `true` に設定します。このプロパティーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。このプロパティーは、以下のプロパティーと一緒に使用できます。

- HTTP
- SocksServer
- SSLClient
- SSLProxyMode

SocksProxyHost
この経路のすべての接続が使用する Socks プロキシーのホスト名 (またはドット 10 進 IP アドレス)。このプロパティーを変更 (および SocksClient を `true` に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SocksProxyPort
Socks プロキシーで使用するポート・アドレス。デフォルト値は 1080 です。このプロパティーを変更 (および SocksClient を `true` に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SocksServer
経路を Socks クライアントとして機能させ、Socks クライアント接続を受け入れるようにするには、このプロパティーを `true` に設定します。このプロパティ

ーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。このプロパティーは、以下のプロパティーと一緒に使用できます。

- SocksClient
- SSLProxyMode
- SSLServer

SSLClient

経路を SSL クライアントとして機能させ、発信 SSL 接続を行わせるようにするには、このプロパティーを true に設定します。 true に設定することは、宛先が、SSL サーバーまたは HTTP プロキシー/サーバーとして機能する別の MQIPT であることを意味します。 SSLClientKeyRing または SSLClientCAKeyRing プロパティーのどちらかを使用して鍵リング・ファイルの名前を指定する必要があります。このプロパティーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。このプロパティーは、以下のプロパティーと一緒に使用できます。

- HTTP
- QoS
- SSLProxyMode

SSLClientCAKeyRing

SSL サーバーから証明書を認証するために使用される、CA 証明書が入っている鍵リング・ファイルの完全修飾名。Windows プラットフォームでは、ファイル分離文字として二重円記号 (¥¥) を使用する必要があります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientCAKeyRingPW

クライアント CA 鍵リングをオープンするためのパスワードが入っている完全修飾ファイル名。 Windows プラットフォームでは、ファイル分離文字として二重円記号 (¥¥) を使用する必要があります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientCipherSuites

SSL クライアント・サイドで使用する SSL 暗号スイートの名前。この名前として可能なのは、サポートされている 1 つまたは複数の暗号スイートです。この名前をブランクにしておくと、SSL クライアントは SSLClientKeyRing からとった、サポートされている暗号スイートを使用します。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientConnectTimeout

このプロパティーを、SSL クライアントが SSL 接続の受け入れを待機する秒数に設定します。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientDN_C

この国名の SSL サーバーから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての国名」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientDN_CN

この共有名の SSL サーバーから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての国名」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientDN_L

このロケーションの SSL サーバーから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべてのロケーション」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientDN_O

この組織の SSL サーバーから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての組織」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientDN_OU

この部門の SSL サーバーから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての部門」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientDN_ST

この都道府県の SSL サーバーから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての都道府県」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientKeyRing

クライアント証明書が入っている鍵リング・ファイルの完全修飾名。 **Windows**

プラットフォームでは、ファイル分離文字として二重円記号 (¥¥) を使用する必要があります。 SSLClient を true に設定する場合は、SSLClientKeyRing を指定する必要があります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientKeyRingPW
クライアント鍵リングをオープンするためのパスワードが入っている完全修飾ファイル名。 **Windows プラットフォーム**では、ファイル分離文字として二重円記号 (¥¥) を使用する必要があります。 SSLClient を true に設定する場合は、SSLClientKeyRingPW を指定する必要があります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientSiteDN_C
このプロパティーを使用して、SSL サーバーに送信する証明書を選択するための国名を指定します。このプロパティーを指定しないと、「任意の国名」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientSiteDN_CN
このプロパティーを使用して、SSL サーバーに送信する証明書を選択するための共通名を指定します。このプロパティーを正しく指定しないと、「任意の共通名」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientSiteDN_L
このプロパティーを使用して、SSL サーバーに送信する証明書を選択するためのロケーションを指定します。このプロパティーを正しく指定しないと、「任意のロケーション」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientSiteDN_O
このプロパティーを使用して、SSL サーバーに送信する証明書を選択するための組織名を指定します。このプロパティーを正しく指定しないと、「任意の組織名」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientSiteDN_OU
このプロパティーを使用して、SSL サーバーに送信する証明書を選択するための部門名を指定します。このプロパティーを正しく指定しないと、「任意の部門名」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientSiteDN_ST
このプロパティーを使用して、SSL サーバーに送信する証明書を選択するための都道府県名を指定します。このプロパティーを正しく指定しないと、「任意の都道府県名」を暗黙指定したことになります。このプロパティーを変更 (および

SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLClientSiteLabel
このプロパティーを使用して、SSL サーバーに送信する証明書を選択するためのラベル名を指定します。このプロパティーを正しく指定しないと、「任意のラベル名」を暗黙指定したことになります。このプロパティーを変更 (および SSLClient を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLProxyMode
経路が SSL クライアント接続要求のみを受け入れ、要求を直接宛先へトンネル化できるようにするには、このプロパティーを true に設定します。このプロパティーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。このプロパティーは、以下のプロパティーと一緒に使用できます。

- HTTP
- QoS
- SocksClient
- SSLClient
- SSLServer

SSLServer
経路を SSL サーバーとして機能させ、着信 SSL 接続を受け入れるようにするには、このプロパティーを true に設定します。 true に設定することは、呼び出し側が、SSL クライアントとして機能する別の MQIPT であることを意味します。このプロパティーを変更すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。このプロパティーは、以下のプロパティーと一緒に使用できます。

- QoS
- SocksServer
- SSLProxyMode

SSLServerCAKeyRing
SSL サーバーから証明書を認証するために使用される、CA 証明書が入っている鍵リング・ファイルの完全修飾ファイル名。Windows プラットフォームでは、ファイル分離文字として二重円記号 (¥¥) を使用する必要があります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerCAKeyRingPW
サーバー CA 鍵リングをオープンするためのパスワードが入っている完全修飾ファイル名。 Windows プラットフォームでは、ファイル分離文字として二重円記号 (¥¥) を使用する必要があります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerAskClientAuth
SSL サーバーによる SSL クライアント認証を要求するには、このプロパティー

を使用します。 SSL クライアントは、SSL サーバーに送信する独自の証明書を持っていなければなりません。この証明書は鍵リング・ファイルから取り出されます。このプロパティーを変更 (および SSLServer を true に設定) すると、REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerCipherSuites
SSL サーバー・サイドで使用する SSL 暗号スイートの名前。この名前として可能なのは、サポートされている 1 つまたは複数の暗号スイートです。この名前をブランクにしておくと、SSL サーバーは SSLServerKeyRing からとった、サポートされている暗号スイートを使用します。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerDN_C
この国名の SSL クライアントから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての会社名」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerDN_CN
この共有名の SSL クライアントから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての共有名」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerDN_L
このロケーションの SSL クライアントから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべてのロケーション」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerDN_O
この組織の SSL クライアントから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての組織」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerDN_OU
この部門の SSL クライアントから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付

けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての部門」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerDN_ST
この都道府県の SSL クライアントから受け取った証明書を受け入れるには、このプロパティーを使用します。この名前の先頭または末尾にアスタリスク (*) を付けて、その有効範囲を拡張することができます。このプロパティーを指定しないと、「すべての都道府県」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerKeyRing
サーバー証明書が入っている鍵リング・ファイルの完全修飾名。 **Windows プラットフォーム**では、ファイル分離文字として二重円記号 (¥¥) を使用する必要があります。 SSLServer を true にする場合は、SSLServerKeyRing を指定する必要があります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerKeyRingPW
サーバー鍵リングをオープンするためのパスワードが入っている完全修飾ファイル名。 **Windows プラットフォーム**では、ファイル分離文字として二重円記号 (¥¥) を使用する必要があります。 SSLServer を true にする場合は、SSLServerKeyRingPW を指定する必要があります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerSiteDN_C
このプロパティーを使用して、SSL クライアントに送信する証明書を選択するための国名を指定します。このプロパティーを指定しないと、「任意の国名」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerSiteDN_CN
このプロパティーを使用して、SSL クライアントに送信する証明書を選択するための共通名を指定します。このプロパティーを正しく指定しないと、「任意の共通名」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerSiteDN_L
このプロパティーを使用して、SSL クライアントに送信する証明書を選択するためのロケーションを指定します。このプロパティーを正しく指定しないと、「任意のロケーション」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerSiteDN_O
このプロパティーを使用して、SSL クライアントに送信する証明書を選択するための組織名を指定します。このプロパティーを正しく指定しないと、「任意の組織名」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerSiteDN_OU
このプロパティーを使用して、SSL クライアントに送信する証明書を選択するための部門名を指定します。このプロパティーを正しく指定しないと、「任意の部門名」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerSiteDN_ST
このプロパティーを使用して、SSL クライアントに送信する証明書を選択するための都道府県名を指定します。このプロパティーを正しく指定しないと、「任意の都道府県名」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

SSLServerSiteLabel
このプロパティーを使用して、SSL クライアントに送信する証明書を選択するためのラベル名を指定します。このプロパティーを正しく指定しないと、「任意のラベル名」を暗黙指定したことになります。このプロパティーを変更 (および SSLServer を true に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。この経路へのすべての接続は終了します。

Trace
必要なトレースのレベルは、0 ～ 5 の範囲の整数で指定できます。 0 の値はトレースを行わないことを意味し、 5 は全トレースを要求します。

このプロパティーを変更したために経路が影響を受ける場合は、REFRESH コマンドを出すときにこの新規値が使用されます。すべての接続がこの新規値を即時に使用します。経路は終了します。

UriName
このプロパティーを使用すれば、HTTP プロキシーや MQIPT サーブレットを使用するときに、リソースの Uniform Resource Identifier の名前を変更することができます。ただし、ほとんどの構成の場合、デフォルト値が使用されます。HTTP プロキシーの場合のデフォルトは、次のようになっています。

```
HTTP://<destination>:<destination_port>/mqipt
```

MQIPT サーブレットの場合のデフォルトは、次のようになっています。

```
HTTP://<destination>:<destination_port>/MQIPTServlet
```

このプロパティーを変更 (および、HTTP または ServletClient を True に設定) すると、 REFRESH コマンドを出したときに経路は停止され、再始動されます。

第 20 章 internet pass-thru の使用開始

この章では、MQIPT の使用を開始するのに役立つ情報を提供します。ここでは、本製品を正しくインストールするためのいくつかの簡単な構成をセットアップします。

この章には、以下のようなセクションが設けられています。

前提事項

各例について、以下のような前提事項を想定しています。

- Windows NT を使用する (ただし、各例は、サポートされている任意のプラットフォームで稼働する)
- ユーザーは、キュー・マネージャー、キュー、および WebSphere MQ 上のチャネルについて詳しい知識を持っている
- WebSphere MQ クライアントおよびサーバーがインストール済みである
- MQIPT が `C:¥mqipt` (Windows の場合) と呼ばれるディレクトリーにインストールされる

- クライアント、サーバー、および各 MQIPT が別々のマシンにインストール済みである
- ユーザーが、amqsputc コマンドを使用してメッセージをキューに入れる操作に慣れている
- ユーザーが、amqsgetc コマンドを使用してメッセージをキューから取り出す操作に慣れている

WebSphere MQ サーバーでは、以下の作業が完了しています。

- MQIPT.QM1 というキュー・マネージャーの定義
- MQIPT.CONN.CHANNEL というサーバー接続チャネルの定義
- MQIPT.LOCAL.QUEUE というローカル・キューの定義
- ポート 1414 での MQIPT.QM1 に対する TCP/IP の開始

同一マシン上の 1 つのポート・アドレスでは、1 つのアプリケーションしか listen できません。ポート 1414 が使用中であれば、空きポート・アドレスを選択し、以下の例の中の 1414 と置き換えます。

これを済ませておけば、amqsputc コマンドを使用してメッセージをキュー・マネージャーのローカル・キューに入れ、 amqsgetc コマンドを使用してそれを取り出すことにより、WebSphere MQ クライアントからキュー・マネージャーへの経路をテストすることができます。

構成の例

以下の例は、ダイアグラムとステップバイステップの指示で表されています。各ダイアグラムの右側にあるチェック・ボックスを使用して、例の進行状況を追跡することができます。一部の例では、mqipt.conf ファイルの編集が必要になります。このファイルは、MQIPT ホーム・ディレクトリーに収められています。

開始する前に、以下の作業を完了していることを確認してください。

- mqiptSample.conf を mqipt.conf にコピーする
- mqipt.conf を編集し、すべての経路を削除する
- ClientAccess のエントリーを True に変更する
- Destination を mqserver.company2.com からキュー・マネージャーの宛先へ変更する
- DestinationPort アドレスをキュー・マネージャーで使用するアドレスへ変更する
- 99 ページの『前提事項』を読む

インストール検証テスト

これは、MQIPT が正しくインストールされたことを確認するための簡単な構成です。

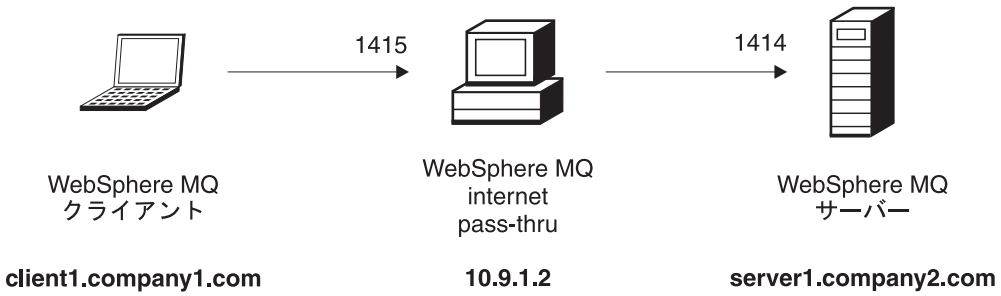

図 10. IVT ネットワーク・ダイアグラム

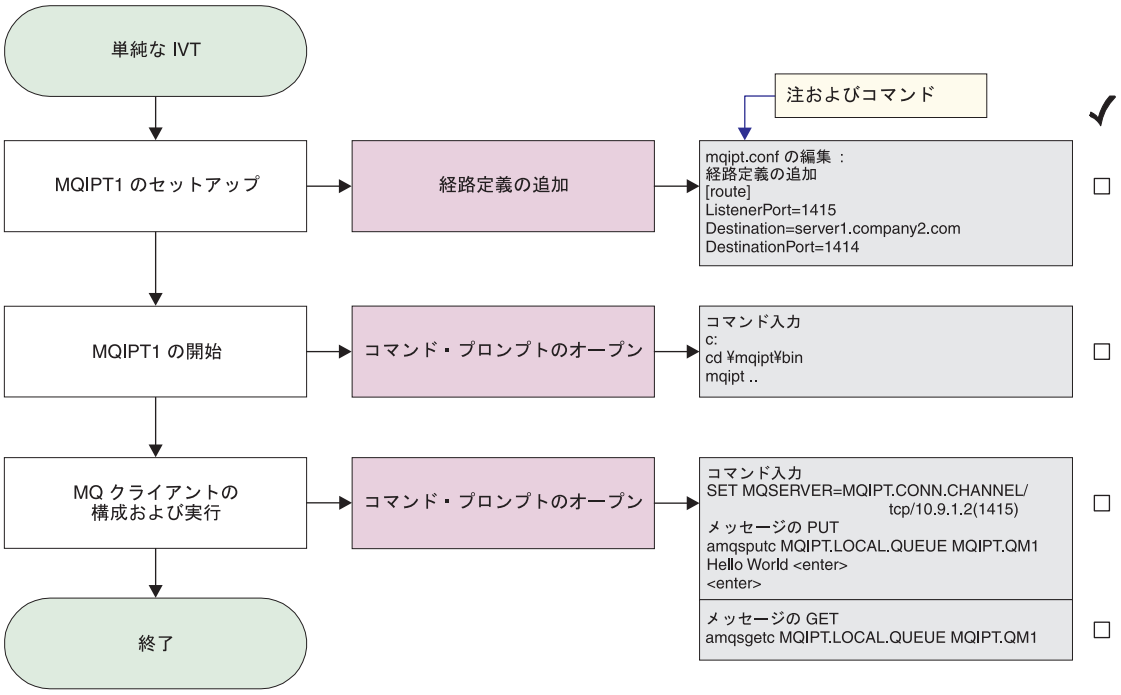

図 11. IVT 構成

1. MQIPT1 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   ```

2. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
   MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
   ```

```
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1415 ready for connection requests
```

3. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

4. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

5. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

「Hello world (ようこそ)」が表示されます。

SSL サーバー認証

この例では、2 つの MQIPT を介して WebSphere MQ クライアントと WebSphere MQ サーバーを接続することにより、サンプル・テスト証明書 (sslsample.pfx 鍵リング・ファイル) を使用して SSL 接続をテストします。 SSL ハンドシェーク時に、サーバーはそのテスト証明書をクライアントに送信します。クライアントは、その証明書 (trust-as-peer フラグが付いている) のコピーを使用してサーバーを認証します。デフォルトの暗号スイート SSL_RSA_WITH_RC4_128_MD5 が使用されます (100 ページの『インストール検証テスト』から作成された mqipt.conf に基づく)。この例で使用するためのテスト証明書を作成する方法の詳細については、120 ページの『SSL テスト証明書の作成』を参照してください。

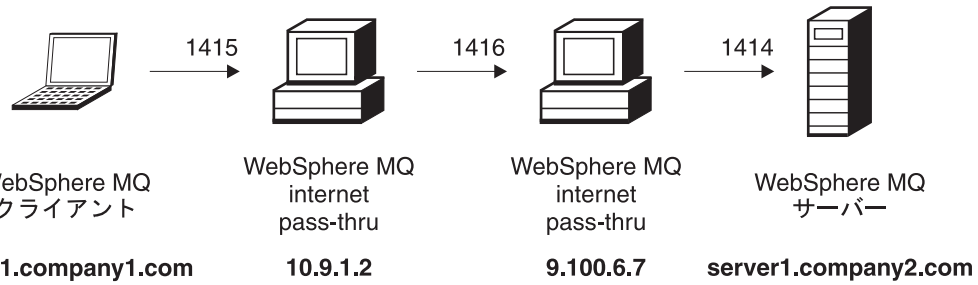

図 12. SSL サーバー・ネットワーク・ダイアグラム

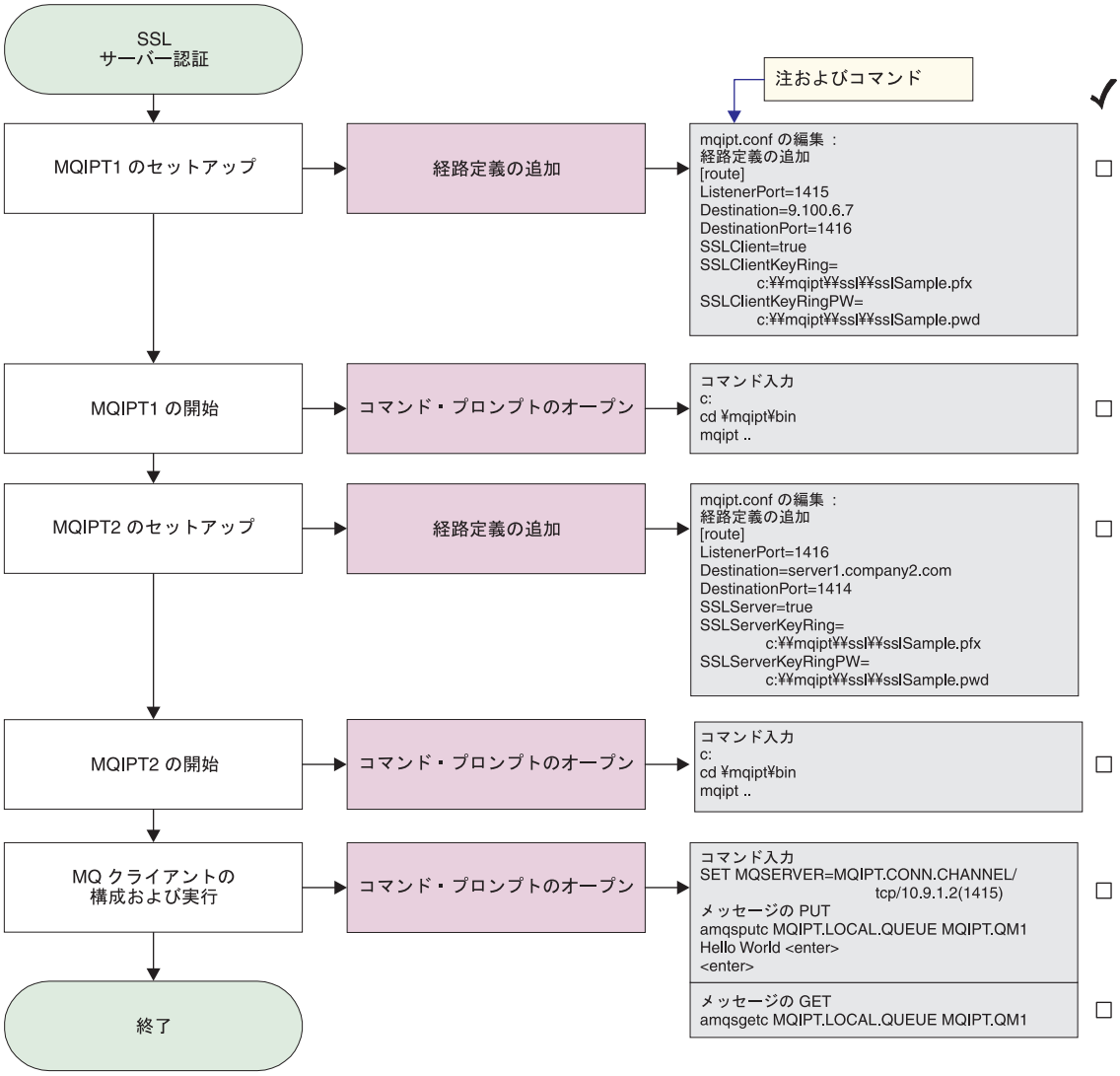

図 13. SSL サーバー認証

1. MQIPT1 をセットアップします

   `mqipt.conf` を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=9.100.6.7
   DestinationPort=1416
   SSLClient=true
   SSLClientKeyRing=C:¥¥mqipt¥¥sslSample.pfx
   SSLClientKeyRingPW=C:¥¥mqipt¥¥sslSample.pwd
   ```

2. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI011 The path c:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....9.100.6.7(1416)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file c:¥mqipt¥sslSample.pfx
MQCPI047 ......CA keyring file <null>
MQCPI038 ......distinguished name(s) CN=* O=* OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

3. MQIPT2 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1416
   Destination=Server1.company2.com
   DestinationPort=1414
   SSLClient=true
   SSLServerKeyRing=C:¥¥mqipt¥¥sslSample.pfx
   SSLServerKeyRingPW=C:¥¥mqipt¥¥sslSample.pwd
   ```

4. MQIPT2 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI011 The path c:¥mqipt¥logs will be used to store the log files
   MQCPI006 Route 1416 has started and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocols
   MQCPI037 ....SSL Server side enabled with properties :
   MQCPI031 ......cipher suites <null>
   MQCPI032 ......keyring file c:¥mqipt¥sslSample.pfx
   MQCPI047 ......CA keyring file <null>
   MQCPI038 ......distinguished name(s) CN=* O=* OU=* L=* ST=* C=*
   MQCPI033 ......client authentication set to false
   MQCPI078 Route 1416 ready for connection requests
   ```

5. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

6. 以下のコマンドを使用してメッセージを入力します。

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

7. 以下のコマンドを使用してメッセージを入手します。

   ```
   amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   ```

「Hello world (ようこそ)」が表示されます。

SSL クライアント認証

この例では、サンプル・テスト証明書を使用して SSL 接続をテストします。このテストではサーバーとクライアントの認証を行います。 SSL ハンドシェーク時に、サーバーはそのテスト証明書をクライアントに送信します。クライアントは、trust-as-peer フラグが付いている、その証明書のコピーを使用してサーバーを認証します。次に、クライアントはそのテスト証明書をサーバーに送信します。サーバーは、trust-as-peer フラグが付いている、その証明書のコピーを使用してクライアントを認証します。デフォルトの暗号スイート SSL_RSA_WITH_RC4_128_MD5 が使用されます (100 ページの『インストール検証テスト』から作成された `mqipt.conf` に基づく)。

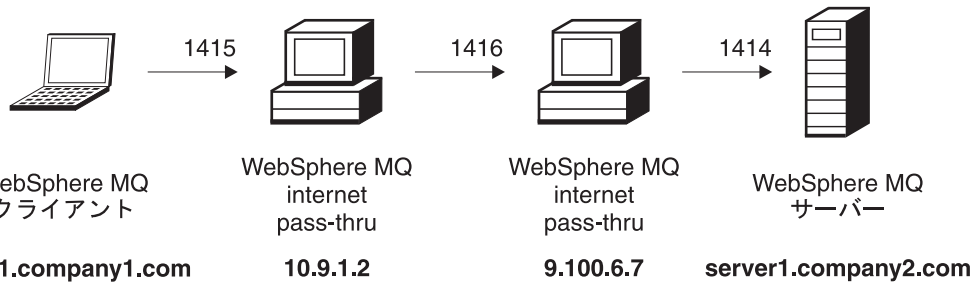

図 14. SSL クライアント・ネットワーク・ダイアグラム

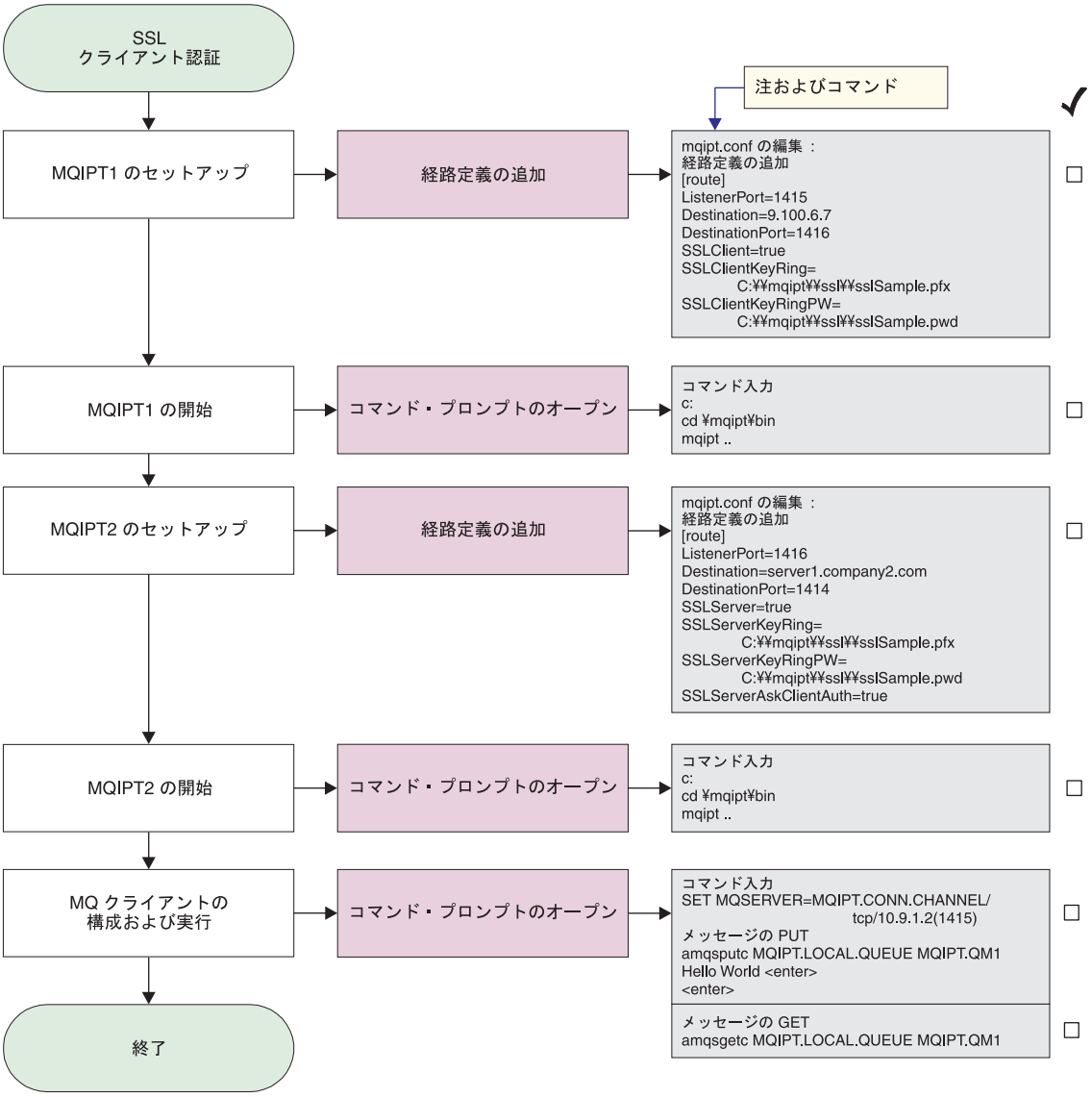

図 15. SSL クライアント認証

1. MQIPT1 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=9.100.6.7
   DestinationPort=1416
   SSLClient=true
   SSLClientKeyRing=C:¥¥mqipt¥¥sslSample.pfx
   SSLClientKeyRingPW=C:¥¥mqipt¥¥sslSample.pwd
   ```

2. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI011 The path c:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....9.100.6.7(1416)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file c:¥mqipt¥sslSample.pfx
MQCPI047 ......CA keyring file <null>
MQCPI038 ......distinguished name(s) CN=* O=* OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

3. MQIPT2 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1416
   Destination=Server1.company2.com
   DestinationPort=1414
   SSLClient=true
   SSLServerKeyRing=C:¥¥mqipt¥¥sslSample.pfx
   SSLServerKeyRingPW=C:¥¥mqipt¥¥sslSample.pwd
   ```

4. MQIPT2 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI011 The path c:¥mqipt¥logs will be used to store the log files
   MQCPI006 Route 1416 has started and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocols
   MQCPI037 ....SSL Server side enabled with properties :
   MQCPI031 ......cipher suites <null>
   MQCPI032 ......keyring file c:¥mqipt¥sslSample.pfx
   MQCPI047 ......CA keyring file <null>
   MQCPI038 ......distinguished name(s) CN=* O=* OU=* L=* ST=* C=*
   MQCPI033 ......client authentication set to true
   MQCPI078 Route 1416 ready for connection requests
   ```

5. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

6. 以下のコマンドを使用してメッセージを入力します。

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

7. 以下のコマンドを使用してメッセージを入手します。

   ```
   amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   ```

「Hello world (ようこそ)」が表示されます。

HTTP プロキシー構成

この例では、HTTP プロキシー (IBM Caching Proxy) を使用して接続をテストします。 CP はレベル 3.6 以上でなければなりません。また、以下の点についてもチェックが必要です。

- `ProxyPersistence` はオンでなければなりません。これによってパーシスタント接続が可能になります。
- `MaxPersistRequest` は 5000 でなければなりません。この数値は、接続を切断する前に単一接続で行える要求の数です。
- `PersistTimeout` は 12hrs でなければなりません。これは接続が存在できる時間です。

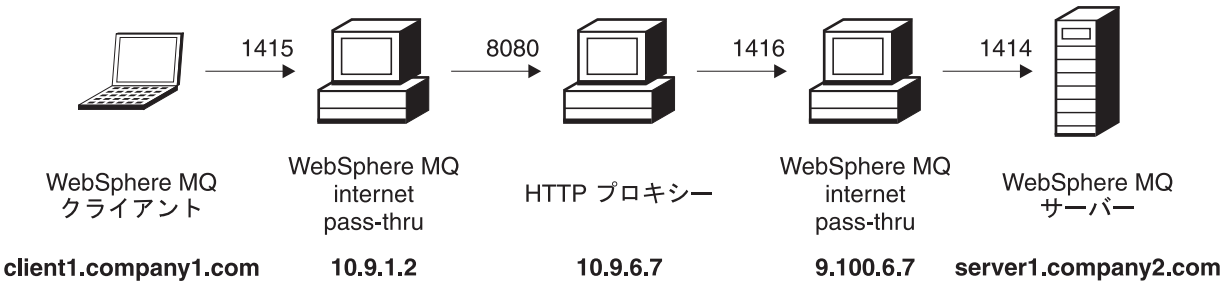

図 16. HTTP プロキシー・ネットワーク・ダイアグラム

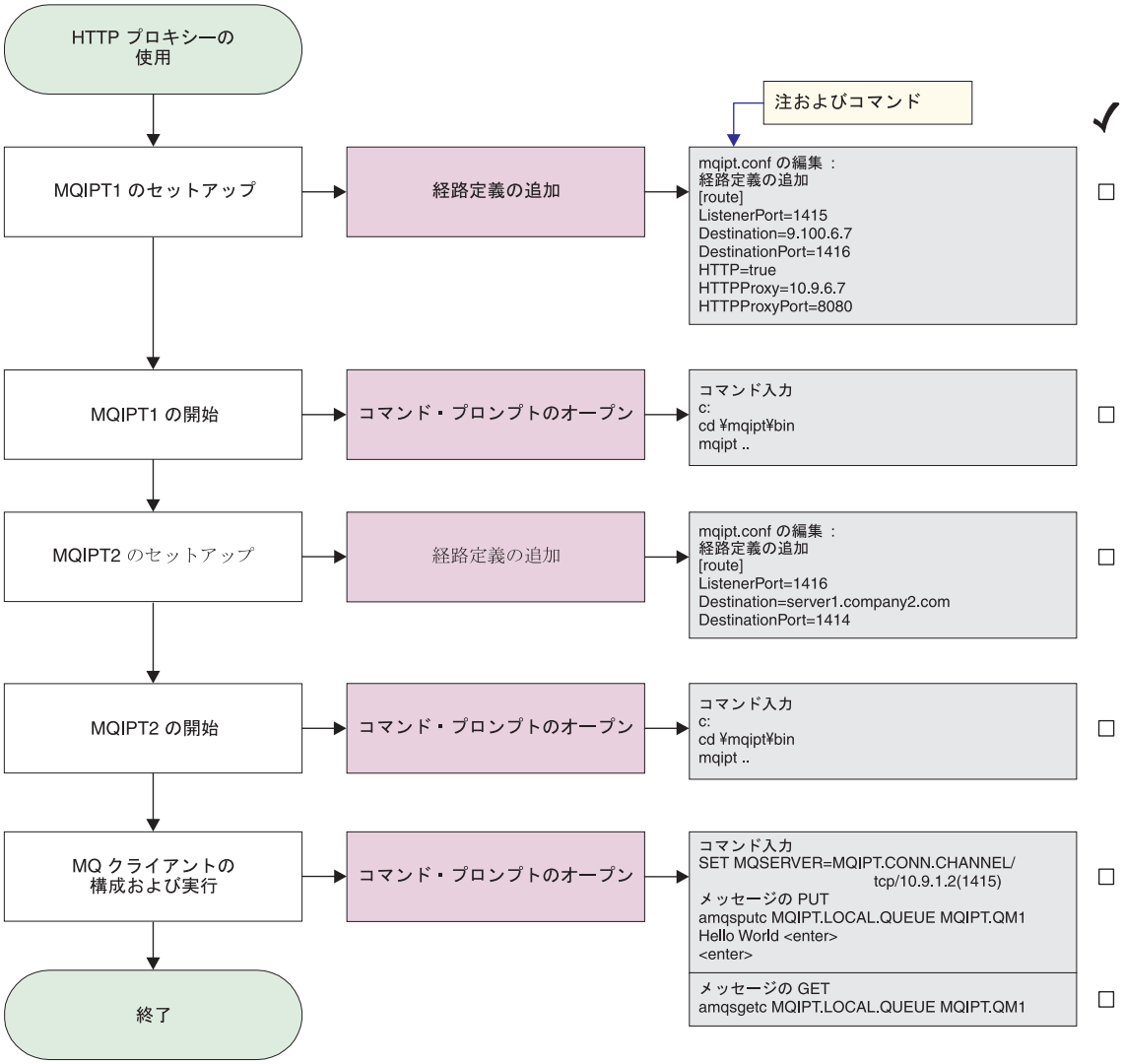

図 17. HTTP プロキシー構成

1. MQIPT1 をセットアップします

   `mqipt.conf` を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=9.100.6.7
   DestinationPort=1416
   HTTP=true
   HTTPProxy=true
   HTTPProxyPort=8080
   ```

2. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
   ```

```
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....9.100.6.7(1416)
MQCPI035 ....using HTTP
MQCPI024 ....and HTTP proxy at 10.9.6.7(1080)
MQCPI078 Route 1415 ready for connection requests
```

3. MQIPT2 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1416
   Destination=Server1.company2.com
   DestinationPort=1414
   ```

4. MQIPT2 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
   MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
   MQCPI006 Route 1416 has started and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocols
   MQCPI078 Route 1416 ready for connection requests
   ```

5. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

6. 以下のコマンドを使用してメッセージを入力します。

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

7. 以下のコマンドを使用してメッセージを入手します。

   ```
   amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   ```

「Hello world (ようこそ)」が表示されます。

構成アクセス制御

この例では、セキュリティー検査を MQIPT リスナー・ポートに追加することによって、特定のクライアントからの接続だけを受け入れるように MQIPT をセットアップします。ここでは、Java Security Manager を使用します。

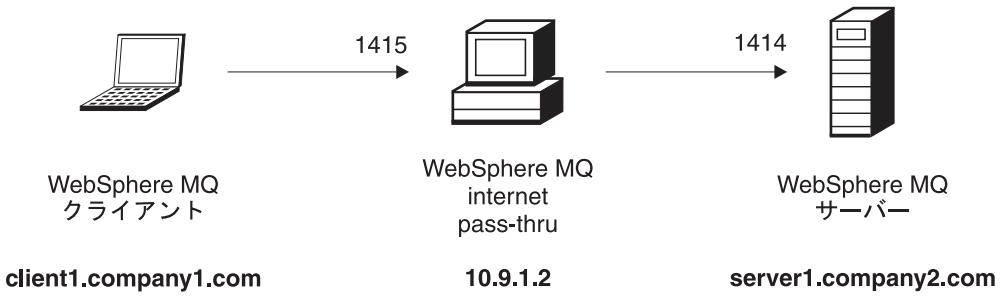

図 18. アクセス制御ネットワーク・ダイアグラム

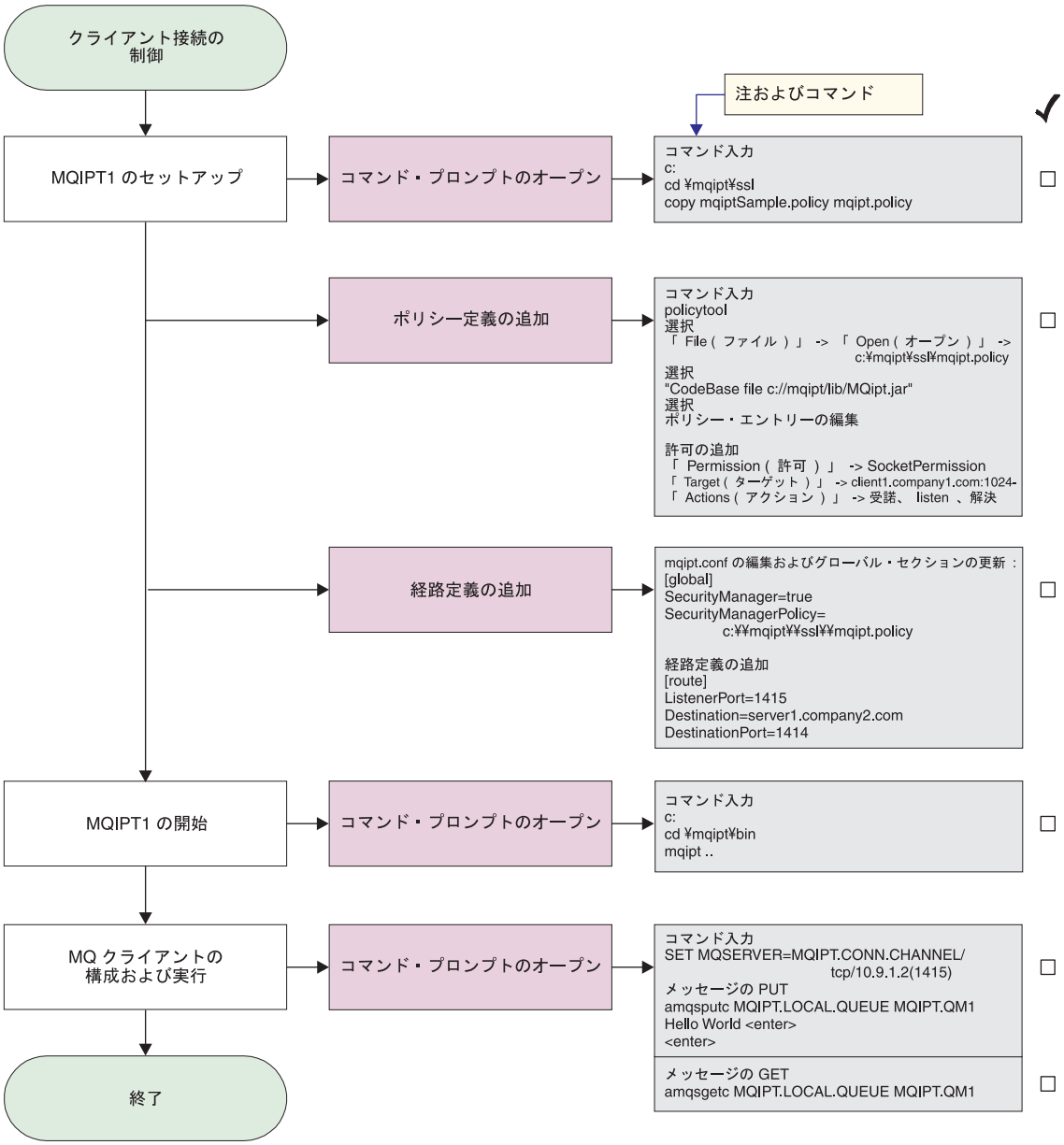

図 19. アクセス制御構成

1. MQIPT1 をセットアップします

a. コマンド・プロンプトをオープンし、次のように入力します。

```
c:
cd ¥mqipt¥ssl
copy c:¥mqipt¥ssl¥mqiptSample.policy to mqipt.policy
```

b. 以下のコマンドを使用してポリシー定義を追加します。

```
policytool
```

1) 「File (ファイル)」->「Open (オープン)」->
「c:¥mqipt¥ssl¥mqipt.policy」と選択します。

2) 以下のコマンドを選択します。

```
file://C:/Program Files/IBM/WebSphere MQ internet pass-thru/lib/MQipt.jar
```

3) CodeBase を、

```
file://C:/Program Files/IBM/WebSphere MQ internet pass-thru/lib/MQipt.jar
```

から、次のように変更します。

```
file://C:/mqipt/lib/MQipt.jar
```

4) すべての許可を、

```
C:¥¥Program Files¥¥IBM¥¥WebSphere MQ internet pass-thru
```

から、次のように変更します。

```
C:¥¥mqipt
```

5) SocketPermission を追加します。

```
Permission=SocketPermission
Target=client1.company1.com:1024-
Actions=accept, listen, resolve
```

c. mqipt.conf を編集し、

1) 2 つのプロパティーを次のグローバル・セクションに追加します。

```
[global]
SecurityManager=true
SecurityManagerPolicy=c:¥¥mqipt¥¥ssl¥¥mqipt.policy
```

2) 経路定義は、以下のとおりです。

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
```

2. MQIPT1 を開始します

コマンド・プロンプトをオープンし、次のように入力します。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI055 Setting the java.security.policy to c:¥mqipt¥mqipt.policy
MQCPI053 Starting the Java Security Manager
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1415 ready for connection requests
```

3. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

4. 以下のコマンドを使用してメッセージを入力します。

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

5. 以下のコマンドを使用してメッセージを入手します。

   ```
   amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   ```

「Hello world (ようこそ)」が表示されます。

Quality of Service (QoS) の構成

この例では、TQoS が MQIPT と同じマシンにインストールされていることを前提にしています。

この例では、Quality of Service (QoS) を MQIPT 経路上のすべてのチャネルに適用します。これは、MQIPT を Linux プラットフォームで実行するときにのみインプリメントできます。このサンプルは、MQIPT から WebSphere MQ クライアントに送信されたすべてのデータに関して「平均」の優先順位を設定し、WebSphere MQ サーバーに送信されたすべてのデーに関して「良好」の優先順位を設定します。下記のサンプル pagent ポリシーを使用すれば、以下の優先順位を `QosToCaller` と `QosToDest` に適用することができます。

- 1 - 平均
- 2 - 良好
- 3 - 非常に良好

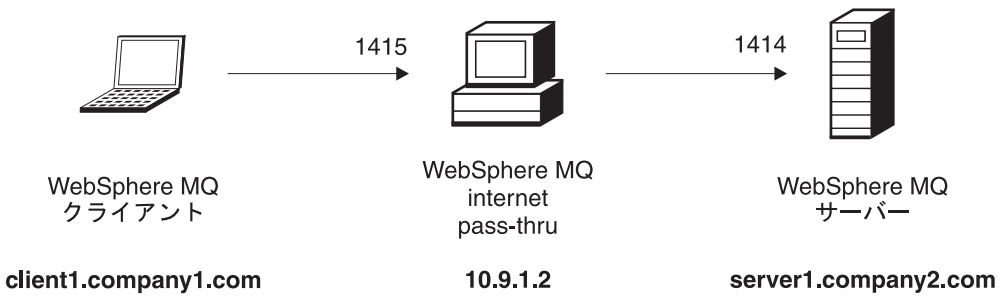

図 20. QoS ネットワーク・ダイアグラム

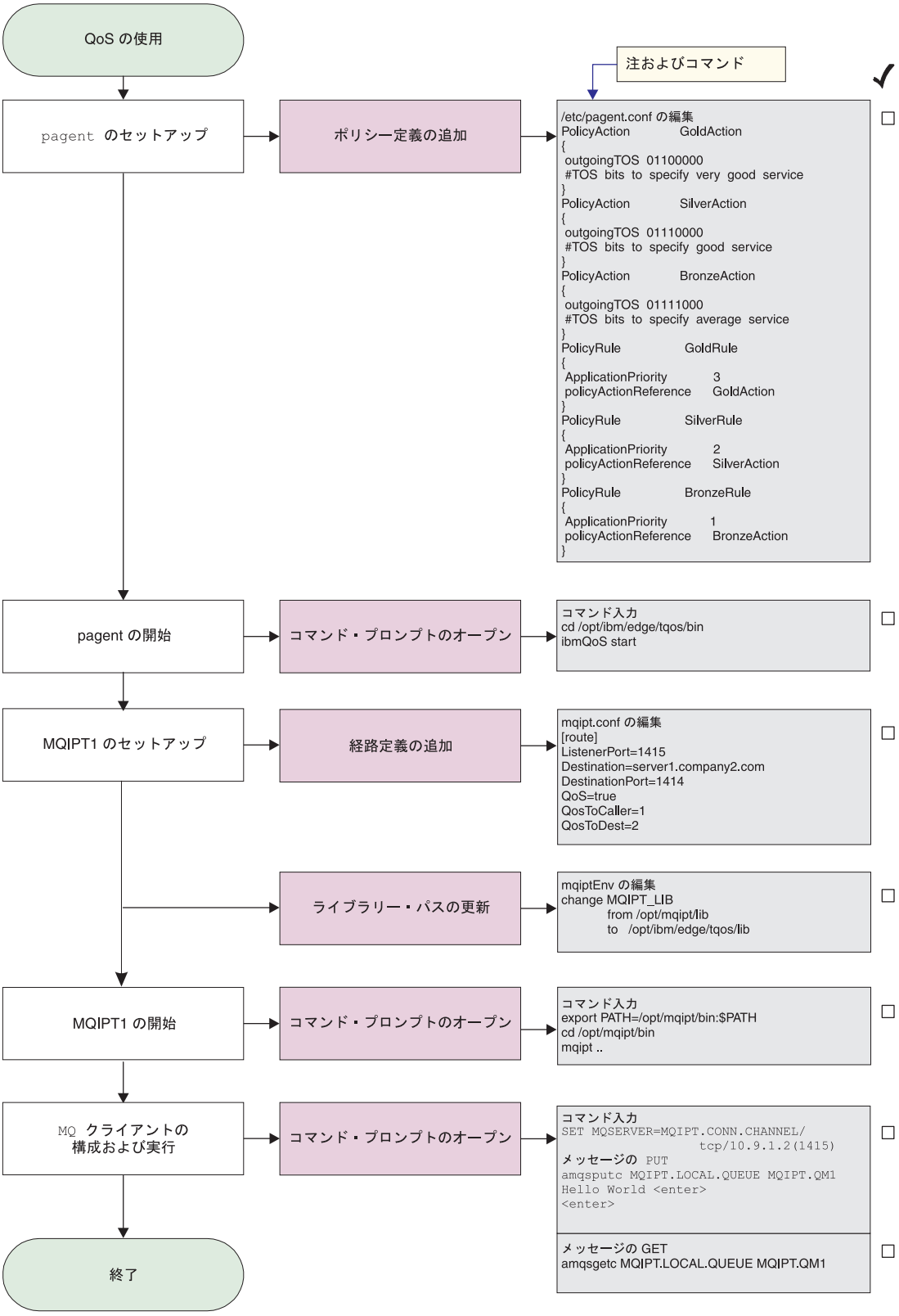

図 21. QoS 構成

1. pagent をセットアップする
   `/etc/pagent.conf` を編集し、以下のコマンドを追加します。

```
PolicyAction      GoldAction
{
 outgoingTOS 01100000
 #TOS bits to specify very good service
}
PolicyAction      SilverAction
{
 outgoingTOS 01110000
 #TOS bits to specify good service
}
PolicyAction      BronzeAction
{
 outgoingTOS 01111000
 #TOS bits to specify average service
}
PolicyRule          GoldRule
{
 ApplicationPriority    3
 policyActionReference  GoldAction
}
PolicyRule          SilverRule
{
 ApplicationPriority    2
 policyActionReference  SilverAction
}
PolicyRule          BronzeRule
{
 ApplicationPriority    1
 policyActionReference  BronzeAction
}
```

上で定義した規則についてパフォーマンス・データ収集をオンにするには、ステートメント PolicyPerformanceCollection を使用して、使用可能にします。このステートメントの説明とフォーマットについては、Pagent.conf を参照してください。

2. pagent を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   cd /opt/ibm/edge/tqos/bin
   ibmQoS start
   ```

3. MQIPT1 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   QoS=true
   QosToCaller=1
   QosToDest=2
   ```

4. ライブラリー・パスを更新します。

   mqiptEnv (/opt/mqipt/bin に入っている) を編集し、MQIPT_LIB を、

   ```
   /opt/mqipt/lib
   ```

   から、次のように変更します。

   ```
   /opt/ibm/edge/tqos/lib
   ```

5. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

```
export PATH=/opt/mqipt/bin:$PATH
cd /opt/mqipt/bin
mqipt ..
```

以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from /opt/mqipt/mqipt.conf
MQCPI011 The path /opt/mqipt/logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI049 ....QoS priority to dest = 2, to caller = 1
MQCPI078 Route 1415 ready for connection requests
```

6. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

7. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

8. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

「Hello world (ようこそ)」が表示されます。

SOCKS プロキシーの構成

この例では、MQIPT を SOCKS プロキシーとして機能させることができます。このサンプルを実行する前に、WebSphere MQ クライアントを SOCKS 化しておかなければなりません。また SOCKS 構成が SOCKS プロキシーとしての MQIPT を指していなければなりません。 MQIPT Destination および DestinationPort プロパティーの定義は何でも構いません。それは、SOCKS ハンドシェーク・プロセス時に真の宛先を WebSphere MQ クライアントから入手するからです。

開始する前に、マシン全体または WebSphere MQ クライアント・アプリケーション (amqsputc/amqsgetc) のいずれかを SOCKS 化する必要があります。また、SOCKS クライアントも以下のように構成する必要があります。

- SOCKS プロキシーとしての MQIPT を指す
- SOCKS V5 サポートを使用可能にする
- ユーザー認証を使用不可にする
- MQIPT ネットワーク・アドレスだけと接続する

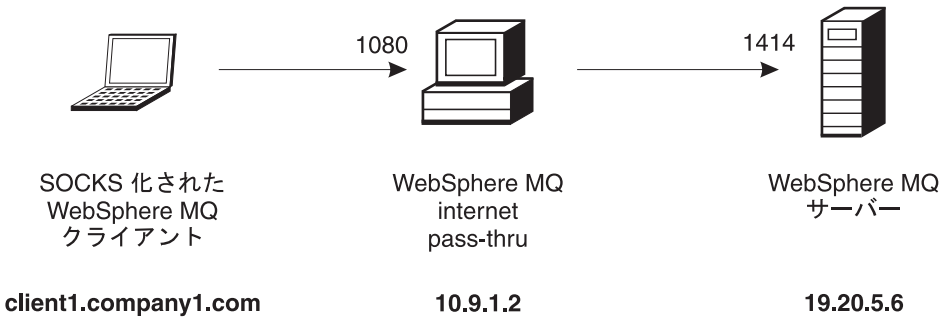

図 22. SOCKS プロキシー・ネットワーク・ダイアグラム

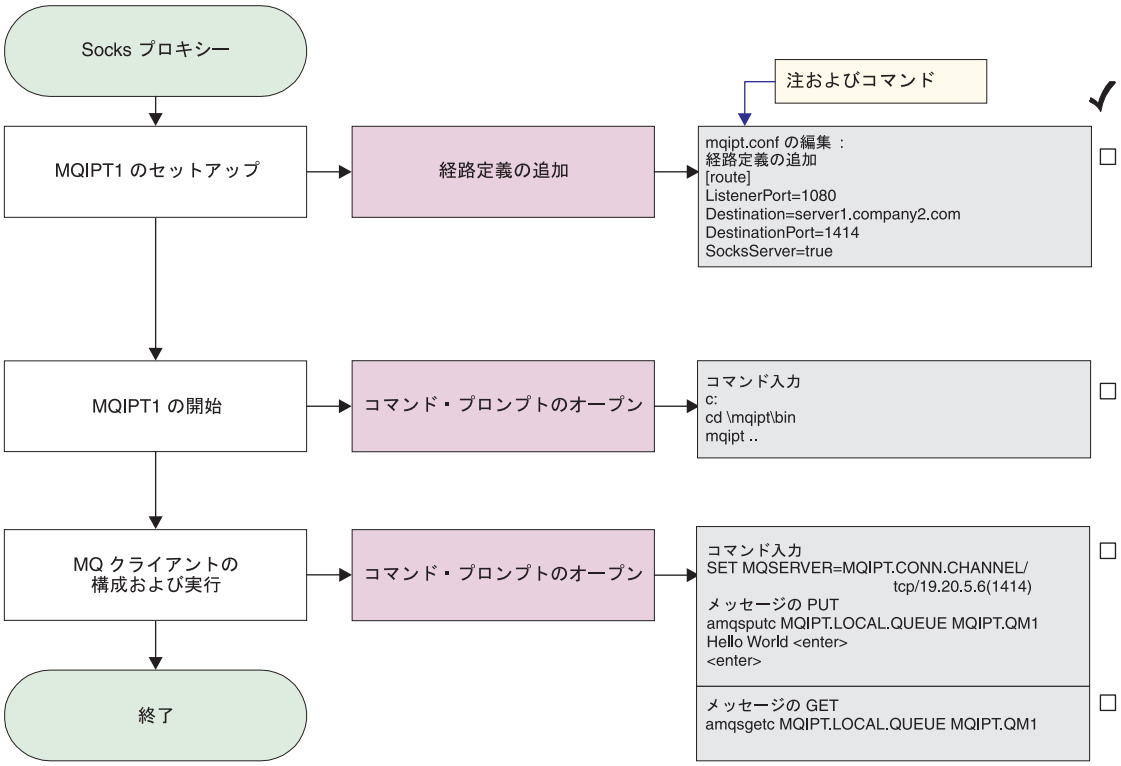

図 23. SOCKS プロキシー構成

1. MQIPT1 をセットアップします

   `mqipt.conf` を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1080
   Destination=server1.company2.com
   DestinationPort=1414
   SocksServer=true
   ```

2. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1080 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI052 ....Socks server side enabled
MQCPI078 Route 1080 ready for connection requests
```

3. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/19.20.5.6(1414)
```

4. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

5. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

「Hello world (ようこそ)」が表示されます。

SOCKS クライアントの構成

この例では、既存の SOCKS プロキシーを使用して、MQIPT をあたかも SOCKS 化されているかのように機能させます。この方法は 116 ページの『SOCKS プロキシーの構成』の場合と似ていますが、MQIPT が、WebSphere MQ クライアントではなく、SOCKS 化された接続を行う点が異なります。

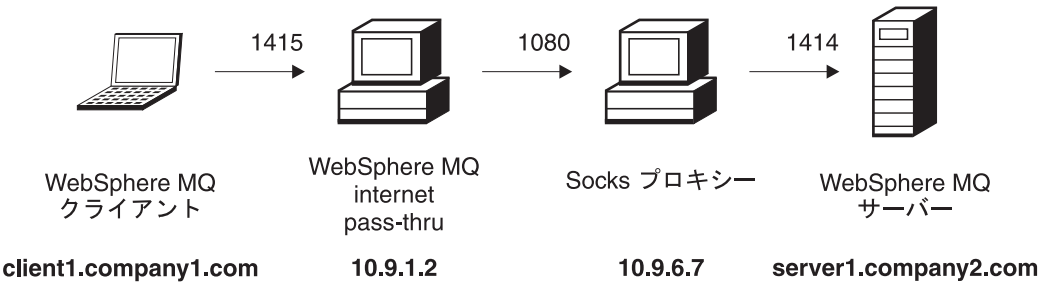

図 24. SOCKS クライアント・ネットワーク・ダイアグラム

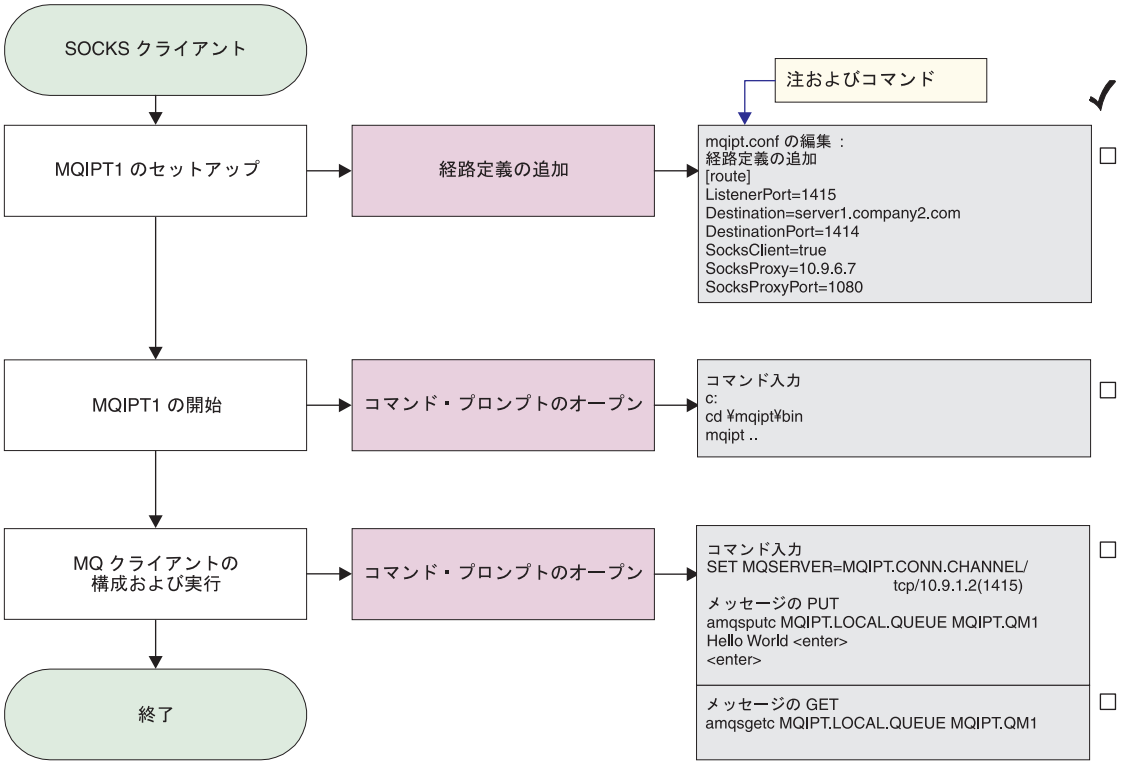

図 25. SOCKS クライアント構成

1. MQIPT1 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   SocksClient=true
   SocksProxy=10.9.6.7
   SocksProxyPort=1080
   ```

2. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
   MQCPI022 Password checking has been disabled on the command port
   MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
   MQCPI006 Route 1415 has started and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocols
   MQCPI039 ....and Socks proxy at 10.9.6.7(1080)
   MQCPI078 Route 1415 ready for connection requests
   ```

3. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

4. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

5. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

「Hello world (ようこそ)」が表示されます。

SSL テスト証明書の作成

この例では、MQIPT 経路のテストに使用できる自己署名証明書を作成する方法を示します。この証明書では、trust-as-peer フラグをオンにします。

1. KeyMan を開始します
2. 「Create new... (新規作成...)」を選択します
3. 「PKCS#12 Token (PKCS#12 トークン)」を選択します
4. 「Action (アクション)」->「Generate Key (鍵を生成)」と選択します

   新規の鍵ペアが ″RSA / 1024-bit″ リストに表示されます
5. 新規の鍵ペアを選択します
6. 「Action (アクション)」->「Create Certificate (証明書を作成)」と選択します
7. 「Self-signed Certificate (自己署名証明書)」を選択します
8. 証明書の詳細情報を入力します

   ダイアログが表示され、「専用証明書が鍵と結合され、ラベルの入力はオプショナルである」ことが示されます。
9. 新規の証明書を選択します
10. 証明書の詳細情報を表示します
11. 証明書プロパティーを変更します
12. trust-as-peer フラグをオンにします
13. ダイアログをクローズし、「File (ファイル)」->「Save (保管)」と選択します
14. パスワードを入力します (たとえば、myPassWord)
15. 新規の鍵リング・ファイルのファイル名を入力します (たとえば、c:¥mqipt¥ssl¥testRoute1414.pfx)

    「File format as PKCS#12 / PFX (PKCS#12 / PFX としてのファイル形式)」を保持し、「Wrap key ring into a Java class (鍵リングを Java クラスにラップする)」に**チェックマークを付けないでください**
16. 上記操作で使用したパスワード (myPassWord) が入っているテキスト・ファイルを作成します。

    たとえば、c:¥mqipt¥ssl¥testRoute1414.pwd

これで、この鍵リング・ファイルを 102 ページの『SSL サーバー認証』例で使用できるようになりました。

MQIPT サーブレットの構成

99 ページの『前提事項』に加えて、この例では以下を前提事項とします。

- Tomcat Application Server が次のディレクトリーにインストールされています。

  ```
  c:¥jakarta-tomcat-4.0.1
  ```

  次のアドレスから Tomcat をダウンロードできます。

  ```
  http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.3/
  ```

- IBM Web Traffic Express が次のアドレスにインストールされています。

  ```
  c:¥wte
  ```

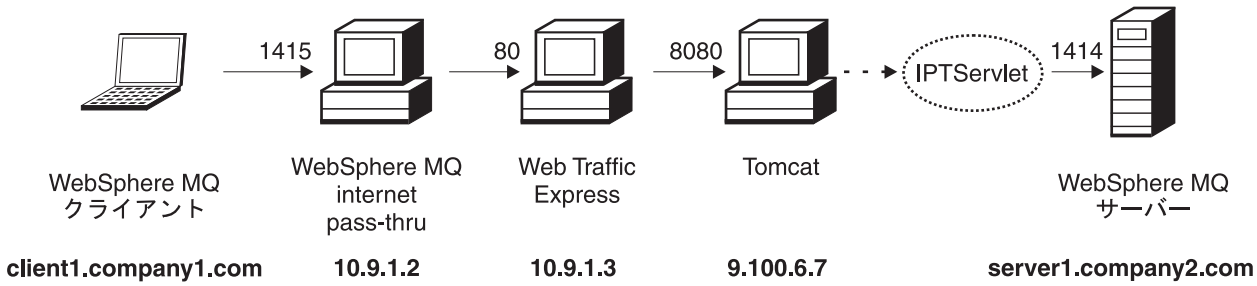

図 26. サーブレット・ネットワーク・ダイアグラム

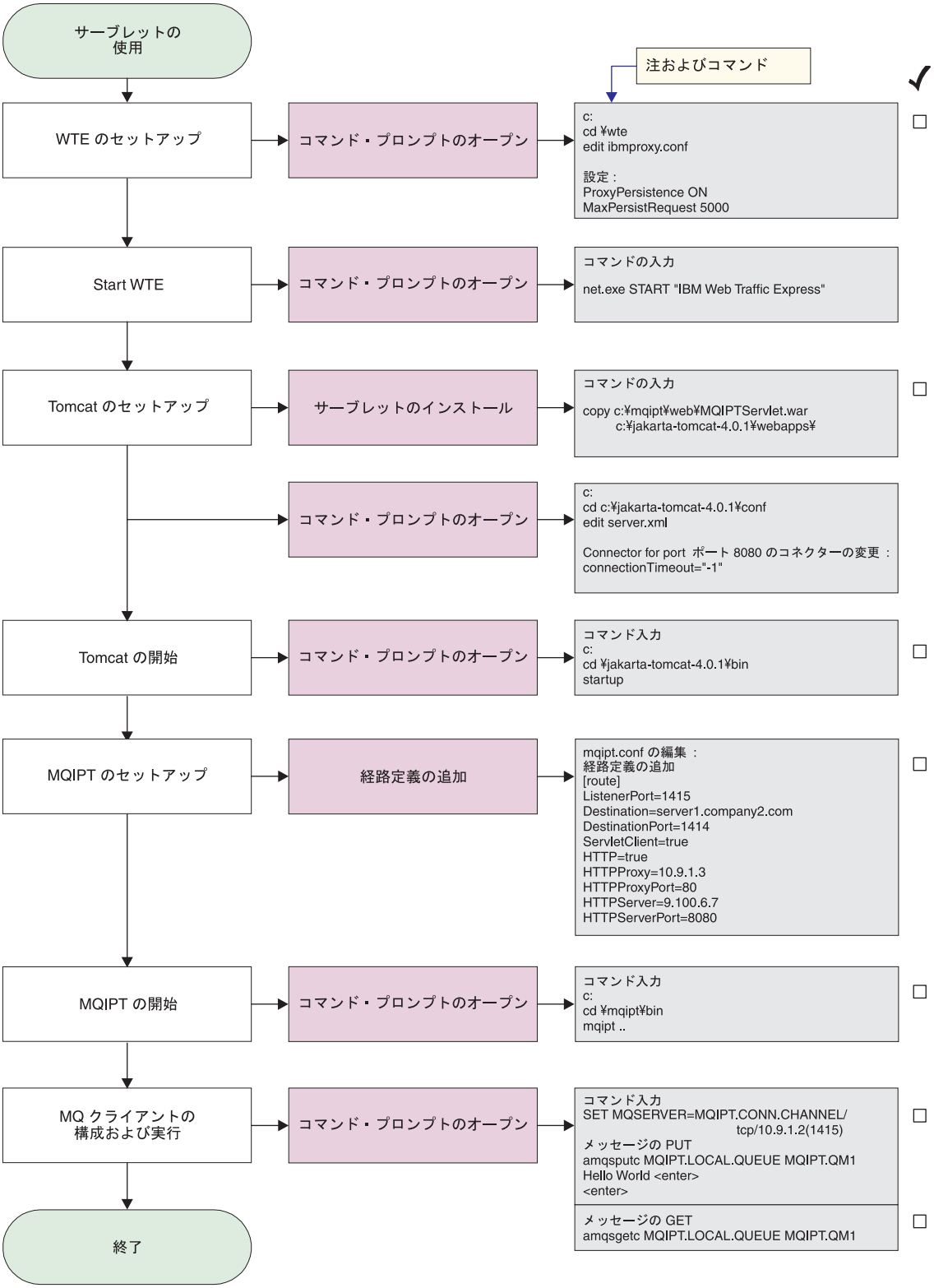

図 27. サーブレット構成

1. Web Traffic Express をセットアップします

   `c:¥wte¥ibmroxy.conf` を編集して、以下のプロパティーを設定します。

```
ProxyPersistence ON
MaxPersistRequest  5000
```

2. Web Traffic Express を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   net.exe Start "IBM Web Traffic Express"
   ```

3. Tomcat をセットアップします

   Servlet をインストールするには、次を

   ```
   c:¥mqipt¥web¥MQIPTServlet.war
   ```

   次のコマンドへコピーします。

   ```
   c:¥jakarta-tomcat-4.0.1¥webapps
   ```

   c:¥jakarta-tomcat-4.0.1¥conf¥server.xml を編集して、ポート 8443 のコネクターを使用可能にし、ConnectionTimeout プロパティーを -1 に設定します。

4. Tomcat を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥jakarta-tomcat-4.0.1¥bin
   startup
   ```

5. MQIPT1 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   ServletClient=true
   HTTP=true
   HTTPProxy=10.9.1.3
   HTTPProxyPort=80
   HTTPServer=9.100.6.7
   HTTPServerPort=8080
   ```

6. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
   MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
   MQCPI006 Route 1415 has started and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using HTTP
   MQCPI024 ....and HTTP proxy at 10.9.1.3(80)
   MQCPI066 ....and HTTP server at 9.100.6.7(8080)
   MQCPI059 ....servlet client enabled
   MQCPI078 Route 1415 ready for connection requests
   ```

7. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

8. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

9. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

「Hello world (ようこそ)」が表示されます。

HTTPS 構成

99 ページの『前提事項』 に加えて、この例では以下を前提事項とします。

- Tomcat Application Server が次のディレクトリーにインストールされています。

  ```
  c:¥jakarta-tomcat-4.0.1
  ```

  次のアドレスから Tomcat をダウンロードできます。

  ```
  http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.3/
  ```

- IBM Web Traffic Express が次のアドレスにインストールされています。

  ```
  c:¥wte
  ```

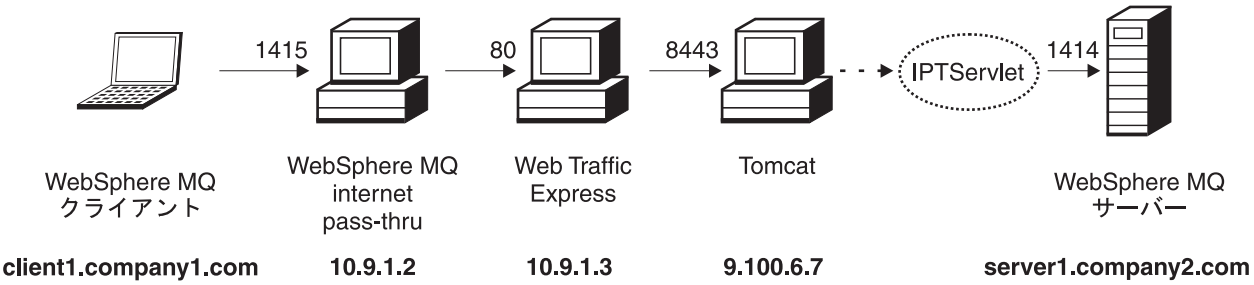

図 28. HTTPS ネットワーク・ダイアグラム

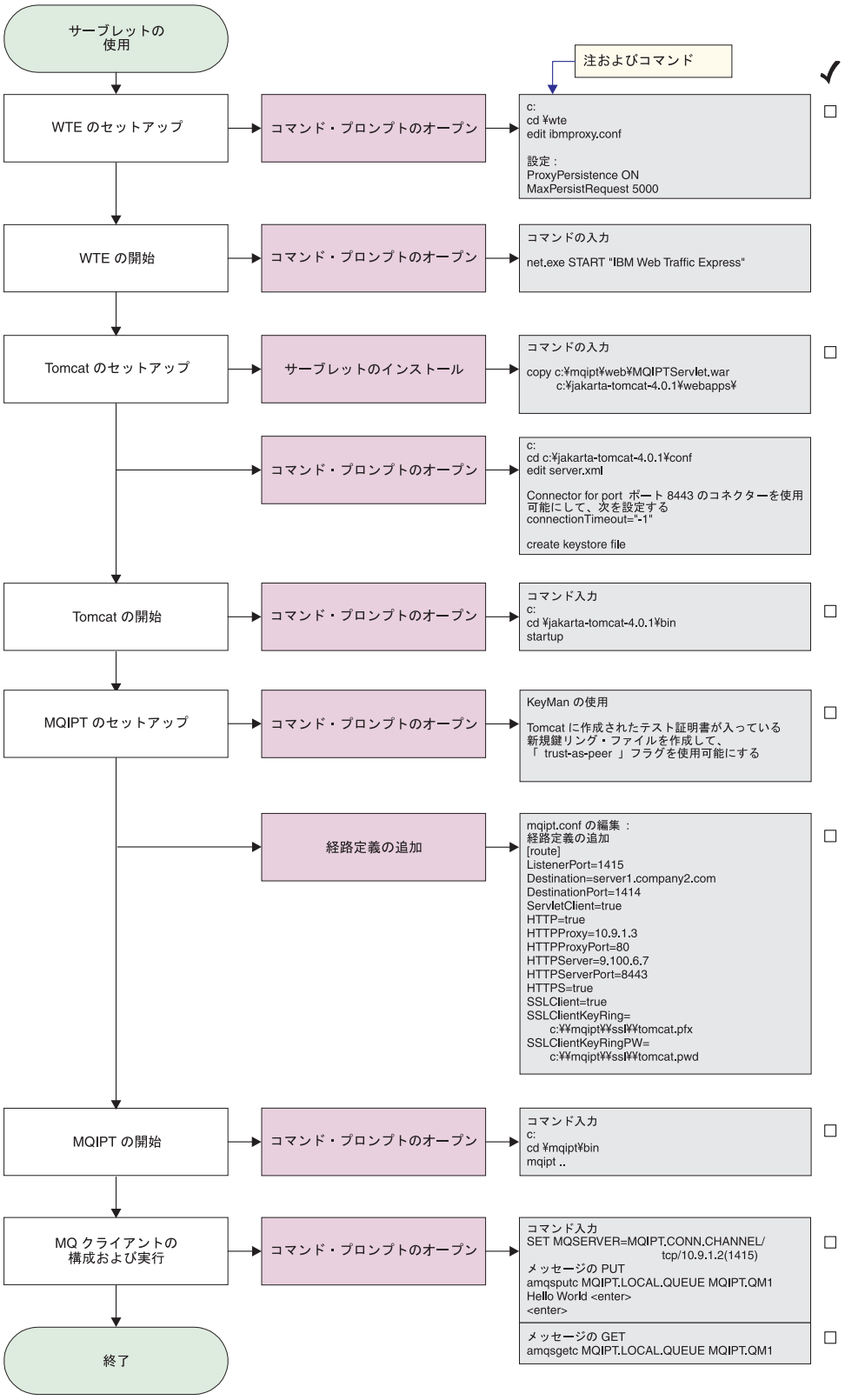

図 29. HTTPS 構成

1. Web Traffic Express をセットアップします
   `c:¥wte¥ibmroxy.conf` を編集して、以下のプロパティーを設定します。

```
ProxyPersistence ON
MaxPersistRequest  5000
```

2. Web Traffic Express を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   net.exe Start "IBM Web Traffic Express"
   ```

3. Tomcat をセットアップします

   Servlet をインストールするには、次を

   ```
   c:¥mqipt¥web¥MQIPTServlet.war
   ```

   次のコマンドへコピーします。

   ```
   c:¥jakarta-tomcat-4.0.1¥webapps
   ```

   c:¥jakarta-tomcat-4.0.1¥conf¥server.xml を編集して、ポート 8443 のコネクターを使用可能にし、ConnectionTimeout プロパティーを -1 に設定します。

   Tomcat の資料を使用してください。この資料は次の Web サイトから入手できます。

   ```
   http://jakarta.apache.org/tomcat/tomcat-4.0-doc/index.html
   ```

   「SSL Configuration HOW-TO」の指示に従ってポート 8443 で SSL 接続を使用可能にしてください。C:¥winnt¥profiles¥<userid>¥.keystore と呼ばれるファイルを作成する、テストの自己署名証明書が入っている鍵リング・ファイルを作成します。

4. Tomcat を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥jakarta-tomcat-4.0.1¥bin
   startup
   ```

5. 新規の鍵ストア・ファイルを Tomcat マシンから MQIPT マシンにコピーします。 KeyMan を使用し、新規の鍵ストア・ファイル (デフォルトのパスワードは changeit) をオープンして、″trust-as-peer″ フラグ (詳細は、 120 ページの『SSL テスト証明書の作成』を参照) をオンにします。
   c:¥mqipt¥ssl¥tomcat.pfx としてこのファイルを保管して、パスワード changeit が入っている c:¥mqipt¥ssl¥tomcat.pwd と呼ばれるテキスト・ファイルを作成します。

6. MQIPT1 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   ServletClient=true
   HTTP=true
   HTTPProxy=10.9.1.3
   HTTPProxyPort=80
   HTTPServer=9.100.6.7
   HTTPServerPort=8443
   HTTPS=true
   ```

```
SSLClient=true
SSLClientKeyRing=c:¥¥mqipt¥¥ssl¥¥tomcat.pfx
SSLClientKeyRingPW=c:¥¥mqipt¥¥ssl¥¥tomcat.pwd
```

7. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using HTTP
MQCPI024 ....and HTTP proxy at 10.9.1.3(80)
MQCPI066 ....and HTTP server at 9.100.6.7(8080)
MQCPI059 ....servlet client enabled
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <null>
MQCPI032 ......keyring file c:¥mqipt¥ssl¥tomcat.pfx
MQCPI047 ......CA keyring file <null>
MQCPI038 ......distinguished name(s) CN=* O=* OU=* L=* ST=* C=*
MQCPI078 Route 1415 ready for connection requests
```

8. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
   ```

9. 以下のコマンドを使用してメッセージを入力します。

   ```
   amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
   Hello world <enter>
    <enter>
   ```

10. 以下のコマンドを使用してメッセージを入手します。

    ```
    amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
    ```

「Hello world (ようこそ)」が表示されます。

MQIPT クラスター化サポートの構成

この例では、 99 ページの『前提事項』に加え、以下の作業も完了している必要があります。

WebSphere MQ サーバー LONDON については、

- LONDON というキュー・マネージャーの定義
- MQIPT.CONN.CHANNEL というサーバー接続チャネルの定義
- ポート 1414 での LONDON に対する TCP/IP の開始
- キュー・マネージャーの SOCKS 化

WebSphere MQ サーバー NEWYORK については、

- NEWYORK というキュー・マネージャーの定義
- MQIPT.CONN.CHANNEL というサーバー接続チャネルの定義
- ポート 1414 での NEWYORK に対する TCP/IP リスナーの開始

- キュー・マネージャーの SOCKS 化

キュー・マネージャーを SOCKS 化するには、マシン全体を SOCKS 化するか、または WebSphere MQ サーバー・アプリケーションだけを SOCKS 化します。以下の操作を行うように、SOCKS クライアントを構成します。

- SOCKS プロキシーとしての MQIPT を指す
- SOCKS V5 サポートを使用可能にする
- ユーザー認証を使用不可にする
- MQIPT だけとのリモート接続を行う

同一マシン上の 1 つのポート・アドレスでは、1 つのアプリケーションしか listen できません。ポート 1414 が使用中であれば、空きポート・アドレスを選択し、例の中の 1414 と置き換えます。これを済ませておけば、メッセージを LONDON のローカル・キューに入れ、それを NEWYORK から取り出すことで、キュー・マネージャー間の経路をテストすることができます。

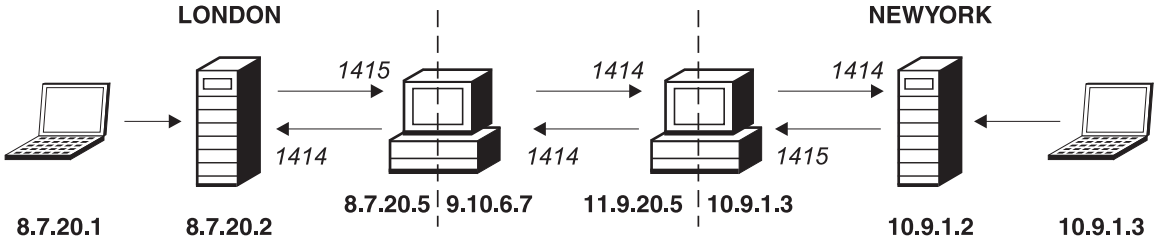

図 30. クラスター化ネットワーク・ダイアグラム

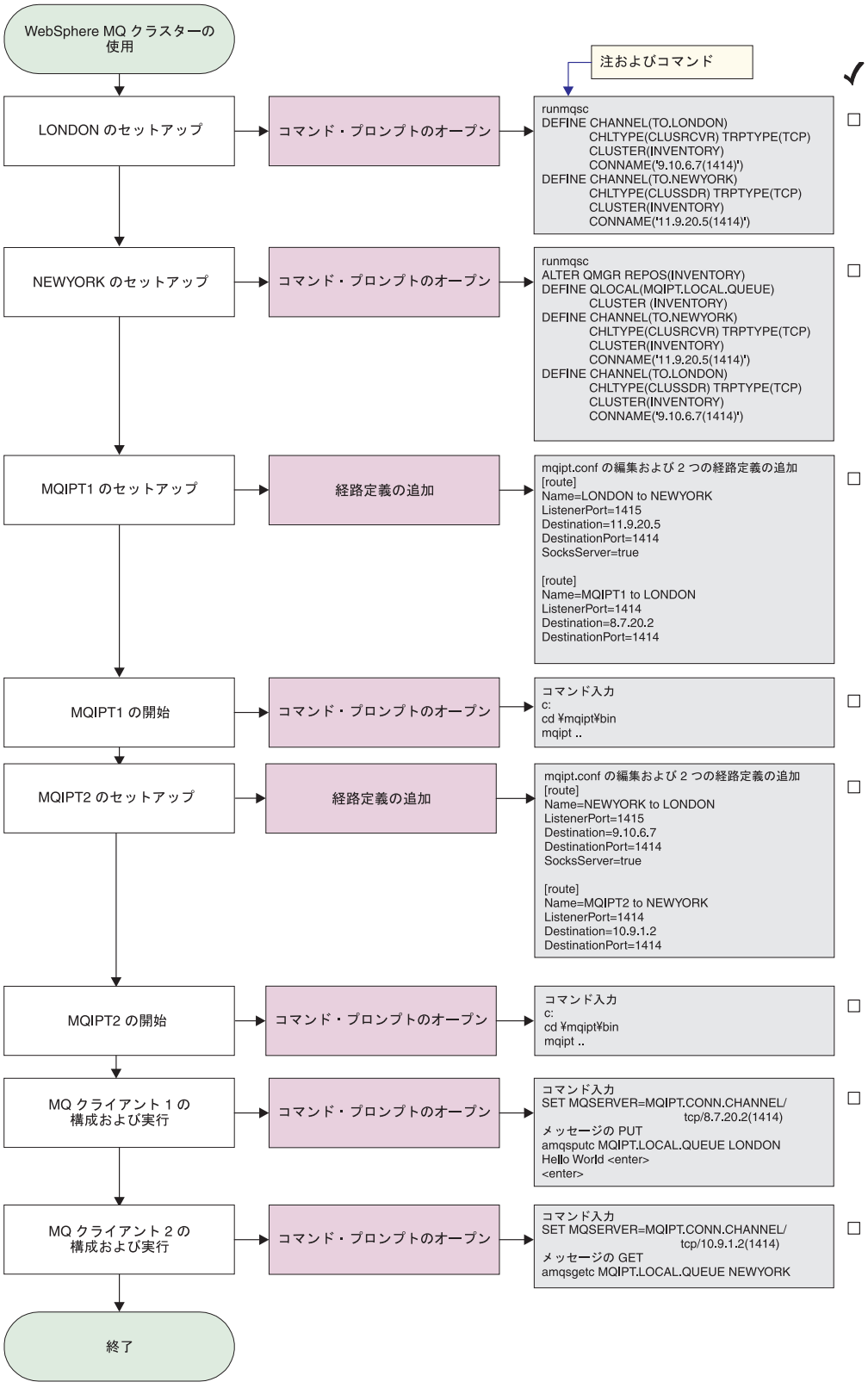

図 31. クラスター化構成

1. LONDON をセットアップします
   コマンド・プロンプトをオープンし、次のように入力します。

```
runmqsc
DEFINE CHANNEL(TO.LONDON) +
       CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('9.10.6.7(1414)')
DEFINE CHANNEL(TO.NEWYORK) +
       CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('11.9.20.5(1414)')
```

2. NEWYORK をセットアップします

   コマンド・プロンプトをオープンし、次のように入力します。

```
runmqsc
ALTER QMGR REPOS(INVENTORY)
DEFINE QLOCAL(MQIPT.LOCAL.QUEUE) +
       CLUSTER(INVENTORY)
DEFINE CHANNEL(TO.NEWYORK) +
       CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('11.9.20.5(1414)')
DEFINE CHANNEL(TO.LONDON) +
       CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
       CLUSTER(INVENTORY) +
       CONNAME('9.10.6.7(1414)')
```

3. MQIPT1 をセットアップします

   mqipt.conf を編集し、2 つの経路定義を追加します。

```
[route]
Name=LONDON to NEWYORK
ListenerPort=1415
Destination=11.9.20.5
DestinationPort=1414
SocksServer=true
```

```
[route]
Name=MQIPT1 to LONDON
ListenerPort=1414
Destination=8.7.20.2
DestinationPort=1414
```

4. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

   以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....11.9.20.5(1414)
MQCPI035 ....using MQ protocols
MQCPI052 ....Socks server side enabled
MQCPI078 Route 1415 ready for connection requests
MQCPI006 Route 1414 has started and will forward messages to :
MQCPI034 ....8.7.20.2(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1414 ready for connection requests
```

5. MQIPT2 をセットアップします

   mqipt.conf を編集し、2 つの経路定義を追加します。

```
[route]
Name=NEWYORK to LONDON
ListenerPort=1415
Destination=9.10.6.7
DestinationPort=1414
SocksServer=true
[route]
Name=MQIPT2 to NEWYORK
ListenerPort=1414
Destination=10.9.1.2
DestinationPort=1414
```

6. MQIPT2 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
   MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
   MQCPI006 Route 1415 has started and will forward messages to :
   MQCPI034 ....9.10.6.7(1414)
   MQCPI035 ....using MQ protocols
   MQCPI052 ....Socks server side enabled
   MQCPI078 Route 1415 ready for connection requests
   MQCPI006 Route 1414 has started and will forward messages to :
   MQCPI034 ....10.9.1.2(1414)
   MQCPI035 ....using MQ protocols
   MQCPI078 Route 1414 ready for connection requests
   ```

7. 最初の WebSphere MQ クライアント・マシン (8. 7. 20. 1) のコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/8.7.20.2(1414)
   ```

8. 以下のコマンドを使用してメッセージを入力します。

   ```
   amqsputc MQIPT.LOCAL.QUEUE LONDON
   Hello world <enter>
    <enter>
   ```

9. 2 番目の WebSphere MQ クライアント・マシン (10. 9. 1. 3) のコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1414)
   ```

10. 2 番目の WebSphere MQ クライアント・マシンで、次のコマンドを使用してこのメッセージを入手します。

    ```
    amqsgetc MQIPT.LOCAL.QUEUE NEWYORK
    ```

「Hello world (ようこそ)」が表示されます。

鍵リング・ファイルの作成

このサンプルでは、ユーザーが Keyman を使用してトラステッド CA から新規の証明書を要求し、ユーザーの個人用証明書がファイル (たとえば、server.cer) でユーザーに戻されたことを前提にしています。サーバー認証を行うにはこれで十分です。クライアント認証が必要な場合は、2 番目の証明書 (たとえば、client.cer) を要求し、以下のステップを 2 回実行して 2 つの鍵リング・ファイルを作成する必要があります。

1. KeyMan を開始します
2. 「Create new... (新規作成...)」を選択します
3. 「PKCS#12 Token (PKCS#12 トークン)」を選択します
4. 「Action (アクション)」->「Generate Key (鍵を生成)」と選択します
   新規の鍵ペアが ″RSA / 1024-bit″ リストに表示されます
5. 新規の鍵ペアを選択します
6. 「Action (アクション)」->「Request Certificate (証明書を要求)」と選択します
   オンライン指示に従います
7. 「File (ファイル)」->「Save (保管)」と選択します
8. パスワードを入力します
9. 新規の鍵リング・ファイルのファイル名を入力します
   たとえば、`c:¥mqipt¥ssl¥myServer.pfx`
10. 「File format as PKCS#12 / PFX (PKCS#12 / PFX としてのファイル形式)」を保持し、「`Wrap key ring into a Java class` (鍵リングを Java クラスにラップする)」に**チェックマークを付けないでください**
11. 「File (ファイル)」->「Exit (終了)」と選択します
12. 上記操作で使用したパスワード (myPassWord) が入っているテキスト・ファイルを作成します。
    たとえば、`c:¥mqipt¥ssl¥myServer.pwd`

証明書を戻してもらう場合は、元の鍵リング・ファイル (`myServer.pfx`) をオープンします。次に、以下の操作を行います。

1. KeyMan を開始します
2. 「Open existing... (既存のファイルのオープン...)」を選択します
3. 「Local resource (ローカル・リソース)」を選択します
4. 「Open a file... (ファイルのオープン...)」を選択します
5. 個人用証明書ファイルの名前を入力します
   たとえば、`c:¥mqipt¥ssl¥myServer.pfx`
6. パスワードを入力します
7. 「File (ファイル)」->「Import (インポート)」と選択します
8. 「Local resource (ローカル・リソース)」を選択します
9. 「Open a file... (ファイルのオープン...)」を選択します
10. server.cer を入力します
    ダイアログが表示され、「専用証明書が秘密鍵と結合される」ことが示されます。
11. 「File (ファイル)」->「Save (保管)」と選択します
12. 「File (ファイル)」->「Exit (終了)」と選択します

上記ステップを繰り返し、`client.cer` ファイルから `myClient.pfx` を作成します。KeyMan を使用してサンプル CA 鍵リング・ファイル `sslCAdefault.pfx` の内容を調べ、自分の個人用証明書がリスト内のいずれかの CA によって署名されているかどうか確認します。署名されていれば、そのサンプル CA 鍵リング・ファイルを使用できます。そうでなければ、自分の個人用証明書を署名した CA 証明書が含まれ

ている鍵リング・ファイルを作成する必要があります。このファイルは、個人用証明書と一緒に戻されていることがあります。戻されていなければ、自分の個人用証明書を提供した CA に CA 証明書を要求し、それを sslCAdefault.pfx にインポートする必要があります。 CA 鍵リング・ファイルは、クライアント・サイドで使用することも、サーバー・サイドで使用することもできます。これらの新規鍵リング・ファイルをサーバー認証に使用する際は、 102 ページの『SSL サーバー認証』の例を参照して、以下の経路プロパティーを設定してください。

```
SSLClientCAKeyRing=c:¥¥mqipt¥¥ssl¥¥sslCAdefault.pfx
SSLClientCAKeyRingPW=c:¥¥mqipt¥¥ssl¥¥sslCAdefault.pwd
SSLServerKeyRing=c:¥¥mqipt¥¥ssl¥¥myServer.pfx
SSLServerKeyRingPW=c:¥¥mqipt¥¥ssl¥¥myServer.pwd
SSLServerCAKeyRing=c:¥¥mqipt¥¥ssl¥¥sslCAdefault.pfx
SSLServerCAKeyRingPW=c:¥¥mqipt¥¥ssl¥¥sslCAdefault.pwd
```

これらの新規鍵リング・ファイルをクライアントおよびサーバー認証に使用する際は、 105 ページの『SSL クライアント認証』の例を参照して、以下の経路プロパティーを設定してください。

```
SSLClientKeyRing=c:¥¥mqipt¥¥ssl¥¥myClient.pfx
SSLClientKeyRingPW=c:¥¥mqipt¥¥ssl¥¥myClient.pwd
SSLClientCAKeyRing=c:¥¥mqipt¥¥ssl¥¥sslCAdefault.pfx
SSLClientCAKeyRingPW=c:¥¥mqipt¥¥ssl¥¥sslCAdefault.pwd
SSLServerKeyRing=c:¥¥mqipt¥¥ssl¥¥myServer.pfx
SSLServerKeyRingPW=c:¥¥mqipt¥¥ssl¥¥myServer.pwd
SSLServerCAKeyRing=c:¥¥mqipt¥¥ssl¥¥sslCAdefault.pfx
SSLServerCAKeyRingPW=c:¥¥mqipt¥¥ssl¥¥sslCAdefault.pwd
```

ポート・アドレスの割り振り

この例は、発信接続を行うときに使用されるローカル・ポート・アドレスの制御方法を示します。この例では、マルチホーム・マシンに MQIPT がインストールされているものと想定しています。

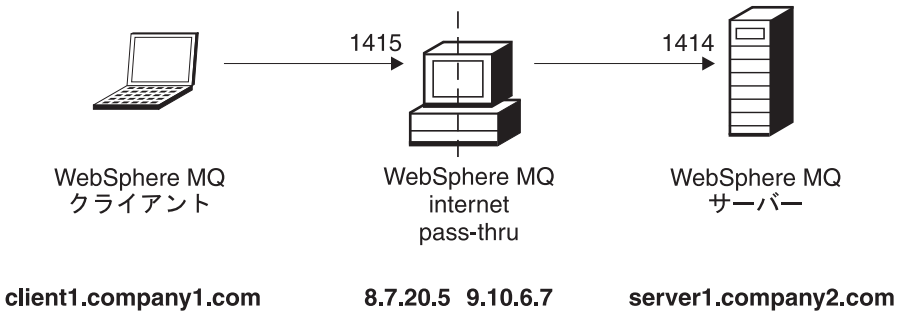

図 32. ポート割り振りネットワーク・ダイアグラム

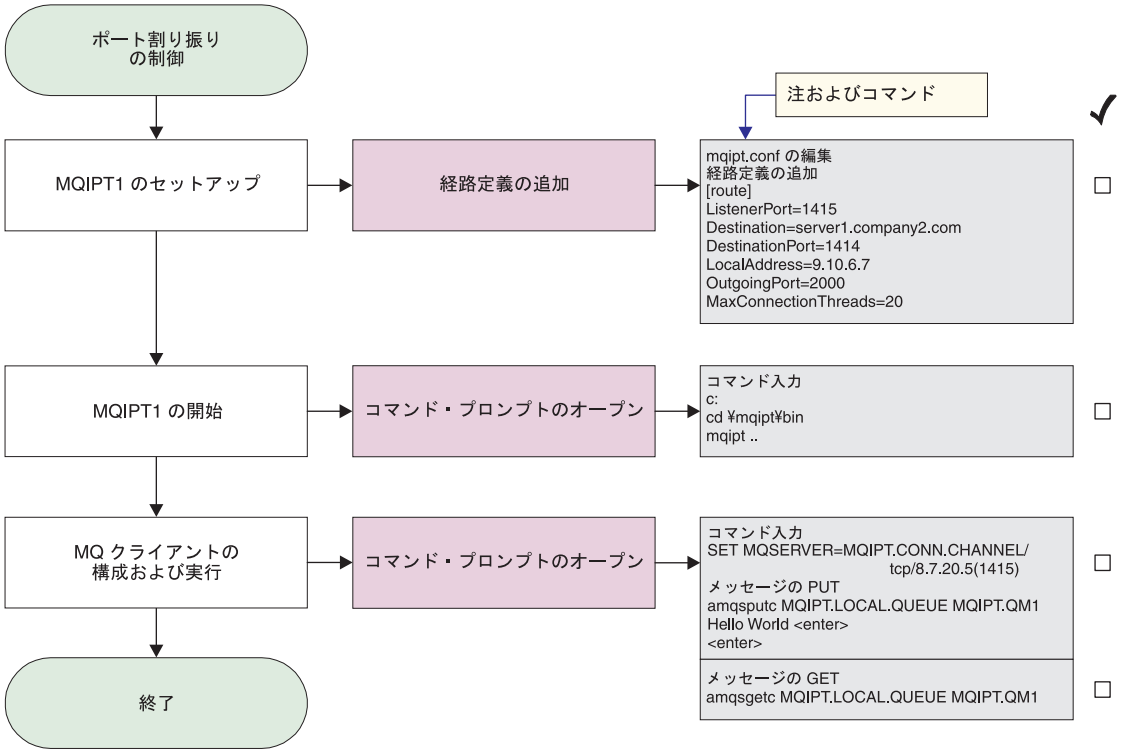

図 33. ポート割り振り構成

1. MQIPT1 をセットアップします

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   LocalAddress=9.10.6.7
   OutgoingPort=2000
   MaxConnectionThreads=20
   ```

2. MQIPT1 を開始します

   コマンド・プロンプトをオープンし、次のように入力します。

   ```
   c:
   cd ¥mqipt¥bin
   mqipt ..
   ```

   以下のメッセージが正常終了を示します。

   ```
   5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
   MQCPI001 WebSphere MQ internet pass-thru Version 1.3.0 starting
   MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
   MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
   MQCPI006 Route 1415 has started and will forward messages to :
   MQCPI034 ....server1.company2.com(1414)
   MQCPI035 ....using MQ protocols
   MQCPI069 ....binding to local address 9.10.6.7
   MQCPI070 ....using local port address range 2000-2019
   MQCPI078 Route 1415 ready for connection requests
   ```

3. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/8.7.20.5(1415)
   ```

4. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT1.QM1
Hello world <enter>
 <enter>
```

5. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT1.QM1
```

「Hello world (ようこそ)」が表示されます。

LDAP サーバーの使用

このサンプルは、LDAP サーバーを使用して CRL を取り出すための MQIPT の構成方法を示します。このサンプルは、LDAP サーバーのインストールやセットアップ方法や、個人用またはトラステッド証明書が入っている鍵リング・ファイルの作成方法を説明するためのものではありません。LDAP サーバーは既知のトラステッド認証局 (CA) から入手できるものと想定します。バックアップ LDAP サーバーは使用されていませんが、該当する Route プロパティーを追加することによって容易にインプリメントすることができます。

この例について、以下のような前提事項を想定しています。

- IPT2 には、トラステッド CA によって発行され、myCert.pfx と呼ばれる鍵リング・ファイルに保管される、個人用証明書があります。この鍵リング・ファイルをオープンするために使用される、暗号化されたパスワードはファイル myCert.pwd に保管されます。
- IPT1 には、トラステッド CA 証明書のコピーがあり、これを使用して IPT2 から送信された証明書を認証します。この証明書は、caCerts.pfx と呼ばれる鍵リング・ファイルに保管され、この鍵リング・ファイルをオープンするために使用される暗号化されたパスワードはファイル caCerts.pwd に保管されます。
- 暗号化されたパスワードのファイルは、mqiptPW スクリプトを使用して作成されています。

このサンプルを実行すると、WMQ クライアントは Queue Manager (QM) に接続して WMQ メッセージをターゲット・キューに入れることができます。IPT1 で MQIPT トレースを実行すると、使用されている LDAP サーバーが示されますが、CRL の機能を説明するには、IPT2 が使用する個人用証明書をトラステッド CA によって取り消す必要があります。次に、この場合、WMQ クライアントは、IPT1 から IPT2 への接続がリジェクトされるため、QM に接続できなくなります。

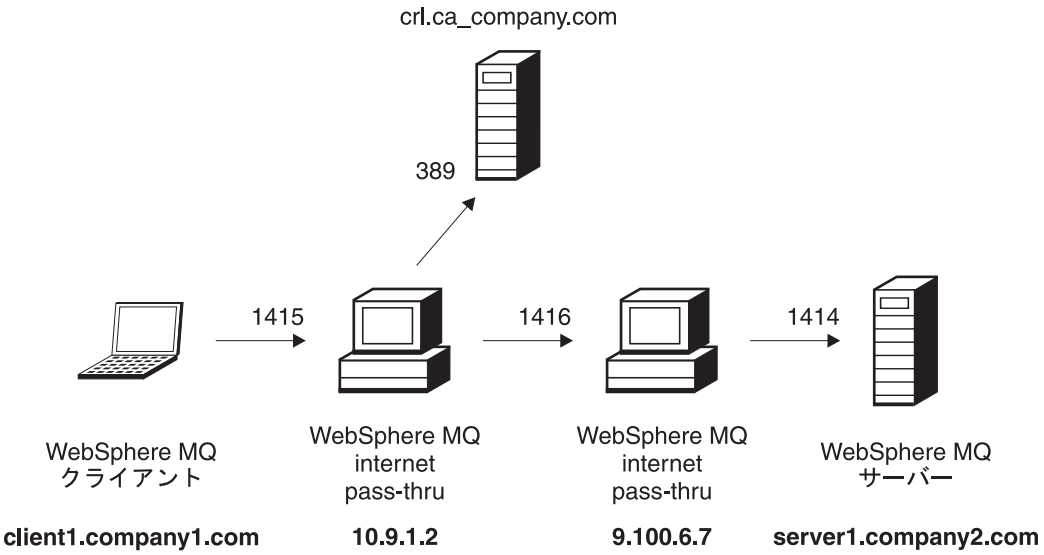

図 34. LDAP サーバー・ネットワーク・ダイアグラム

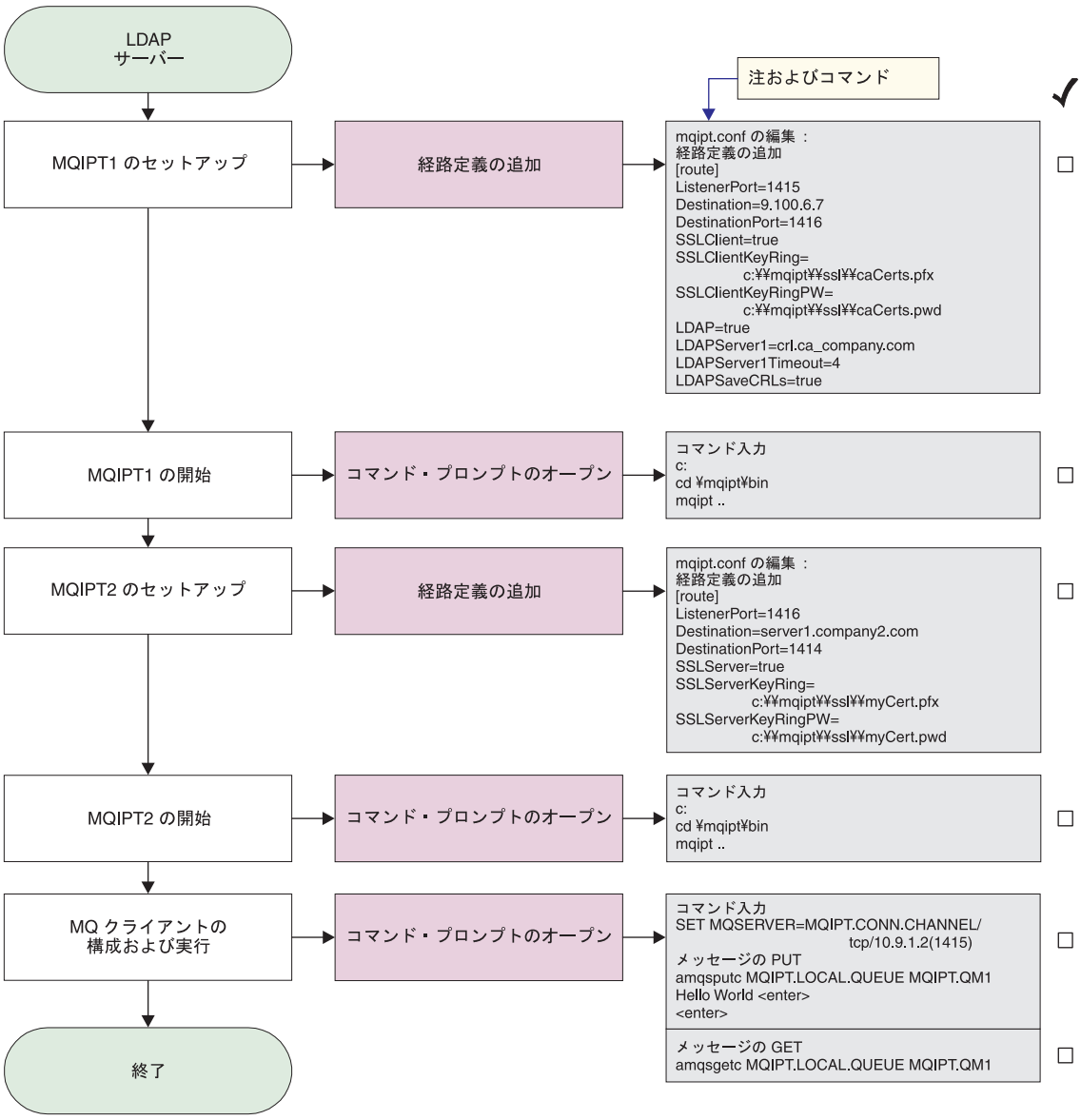

図 35. LDAP サーバー構成

1. IPT1 上で

   mqipt.conf を編集し、経路定義を追加します。

```
[route]
ListenerPort=1415
Destination=9.100.6.7
DestinationPort=1416
SSLClient=true
SSLClientKeyRing=c:¥¥mqipt¥¥ssl¥¥caCerts.pfx
SSLClientKeyRingPW=c:¥¥mqipt¥¥ssl¥¥caCerts.pwd
LDAP=true
LDAPServer1=crl.ca_company.com
LDAPServer1Timeout=4
LDAPSaveCRLs=true
```

   次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....9.100.6.7(1416)
MQCPI035 ....using MQ protocols
MQCPI036 ....SSL Client side enabled with properties :
MQCPI031 ......cipher suites <NULL>
MQCPI032 ......keyring file <NULL>
MQCPI047 ......CA keyring file c:¥mqipt¥ssl¥caCerts.pfx
MQCPI071 ......site certificate uses CN=* O=* OU=* L=* ST=* C=*
MQCPI038 ......peer certificate uses CN=* O=* OU=* L=* ST=* C=*
MQCPI075 ....LDAP main server at crl.ca_company.com(389)
MQCPI086 ......timeout of 4 second(s)
MQCPI084 ....CRL cache expiry timeout is 1 hour(s)
MQCPI085 ....CRLs will be saved in the key ring file(s)
MQCPI078 Route 1415 ready for connection requests
```

2. IPT2 上で

   mqipt.conf を編集し、経路定義を追加します。

```
[route]
ListenerPort=1416
Destination=server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerKeyRing=c:¥¥mqipt¥¥ssl¥¥myCert.pfx
SSLServerKeyRingPW=c:¥¥mqipt¥¥ssl¥¥myCert.pwd
```

   次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

   以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 IBM WebSphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1416 is starting and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI037 ....SSL Server side enabled with properties :
MQCPI031 ......cipher suites <NULL>
MQCPI032 ......keyring file c:¥mqipt¥ssl¥myCert.pfx
MQCPI047 ......CA keyring file <NULL>
MQCPI071 ......site certificate uses CN=* O=* OU=* L=* ST=* C=*
MQCPI038 ......peer certificate uses CN=* O=* OU=* L=* ST=* C=*
MQCPI033 ......client authentication set to false
MQCPI078 Route 1416 ready for connection requests
```

3. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

4. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world <enter>
<enter>
```

5. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

「Hello world (ようこそ)」が表示されます。

SSL プロキシー・モード

このサンプルでは、MQIPT を SSL プロキシー・モードで実行し、MQIPT が SSL クライアントからの SSL 接続要求を受け入れて、それを SSL サーバーへトンネル操作で送信できるようにします。 WMQ クライアントとサーバーは、両方とも V5.3 で SSL 接続を使用するように構成されているものと想定します。

WMQ 用に SSL をセットアップする方法については、「WebSphere MQ セキュリティー バージョン 5.3」、SC88-9231-00 を参照してください。

この例について、以下のような前提事項を想定しています。

- MQClient と QM は SSL チャネルを使用するようにセットアップされています。

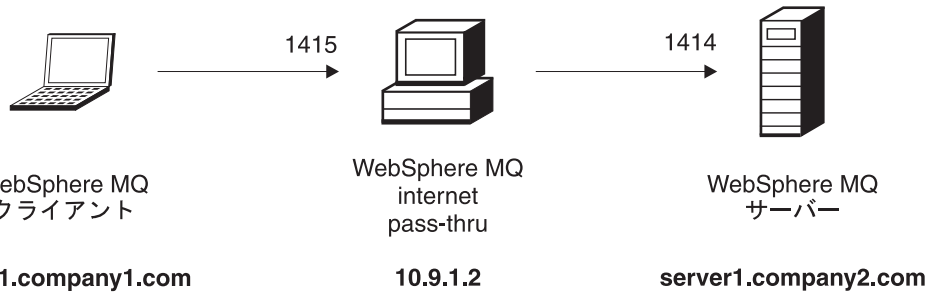

図 36. SSL プロキシー・ネットワーク・ダイアグラム

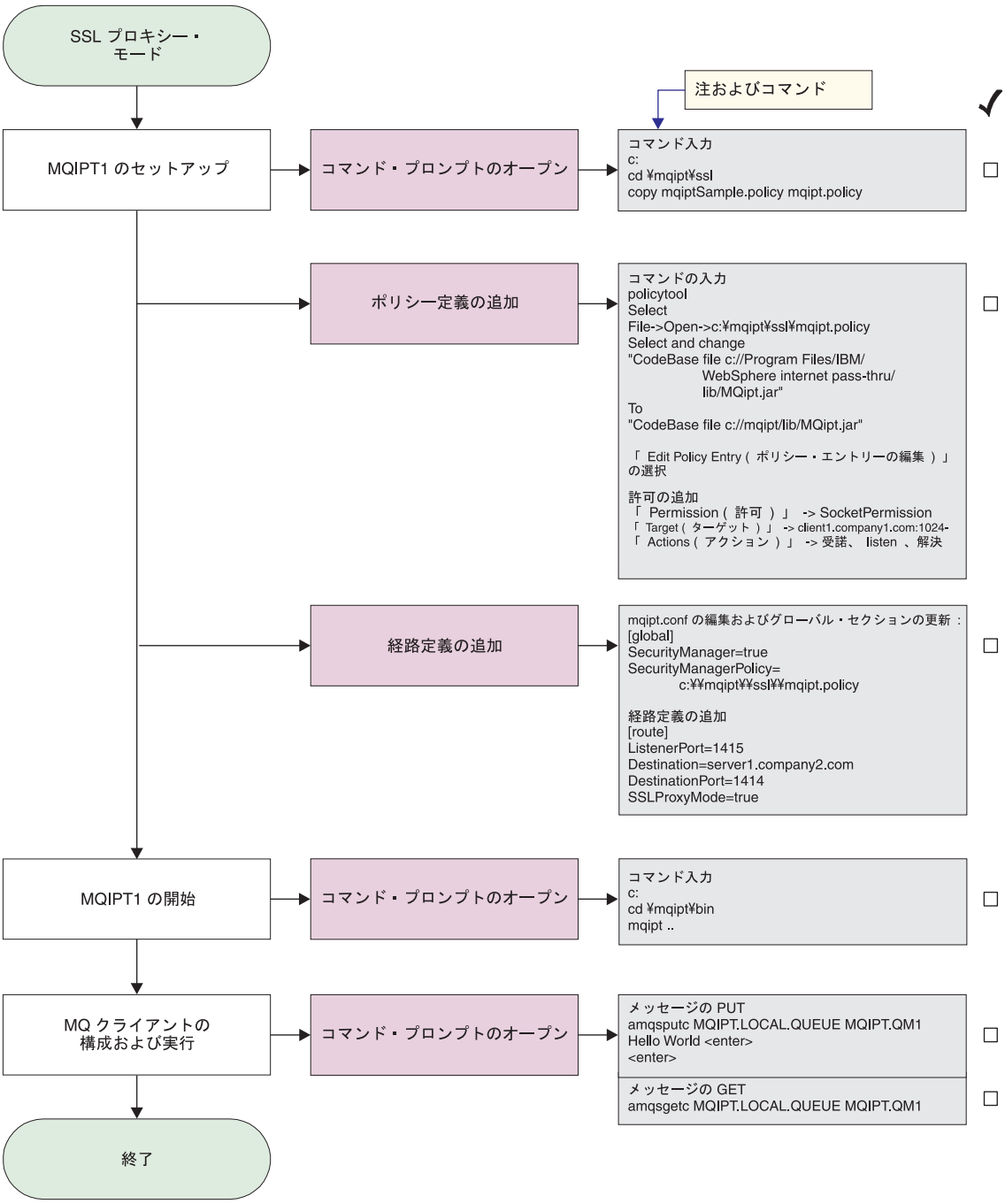

図 37. SSL プロキシー・モード構成

1. IPT1 上で
   a. コマンド・プロンプトをオープンし、次のように入力します。

      ```
      copy c:¥mqipt¥ssl¥mqiptSample.policy to mqipt.policy
      ```

   b. 以下のコマンドを使用してポリシー定義を追加します。

      ```
      policytool
      ```

      1) **「File (ファイル)」 —> 「Open (オープン)」 —> 「c:¥mqipt¥ssl¥mqipt.policy」と選択します。**
      2) 以下のコマンドを選択します。

```
"file://C:/Program Files/IBM/WebSphere MQ internet pass-thru/lib/MQipt.jar"
```

3) CodeBase を、

```
"file://C:/Program Files/IBM/WebSphere MQ internet pass-thru/lib/MQipt.jar"
```

から、次のように変更します。

```
"file://C:/mqipt/lib/MQipt.jar"
```

4) すべての許可を、

```
"C:¥¥Program Files¥¥IBM¥¥WebSphere MQ internet pass-thru"
```

から、次のように変更します。

```
"C:¥¥mqipt"
```

5) SocketPermission を追加します。

```
Permission=SocketPermission
Target = "client1.company1.com:1024-"
Actions = "accept, listen, resolve"
```

2. mqipt.conf を編集して、次の 2 つのプロパティーをグローバル・セクションと経路定義に追加します。

```
[global]
SecurityManager=true
SecurityManagerPolicy=c:¥¥mqipt¥¥ssl¥¥mqipt.policy

[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SSLProxyMode=true
```

3. 次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using SSLProxyMode
MQCPI078 Route 1415 ready for connection requests
```

4. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world <enter>
<enter>
```

5. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

「Hello world (ようこそ)」が表示されます。

Apache 再書き込み

この例について、以下のような前提事項を想定しています。

- Apache HTTP が c:¥apache にインストールされている
- IBM Web Traffic Express が c:¥wte にインストールされている

このサンプルは、再書き込みディレクティブを使用して HTTP 要求を内部 Apache プロキシー・リダイレクトに変換する方法を示します。プロキシー・モジュールと再書き込みモジュールをロードする必要がありますが、Apache はプロキシー・モードで実際には機能していないため、すべてのプロキシー・ディレクティブはコメント化されたままにしてもかまいません。

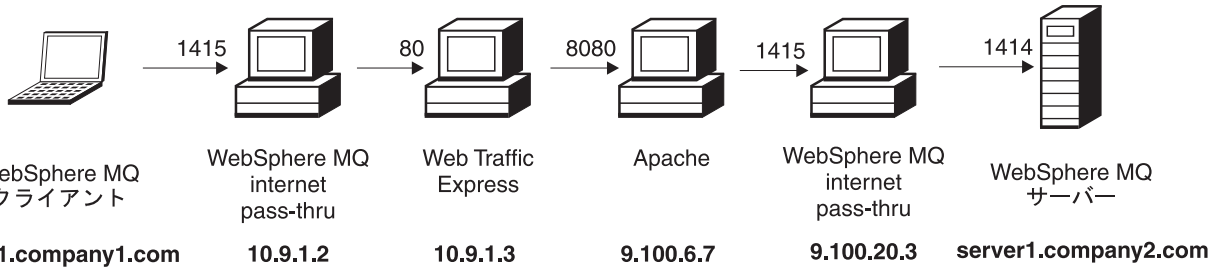

図 38. Apache 再書き込みネットワーク・ダイアグラム

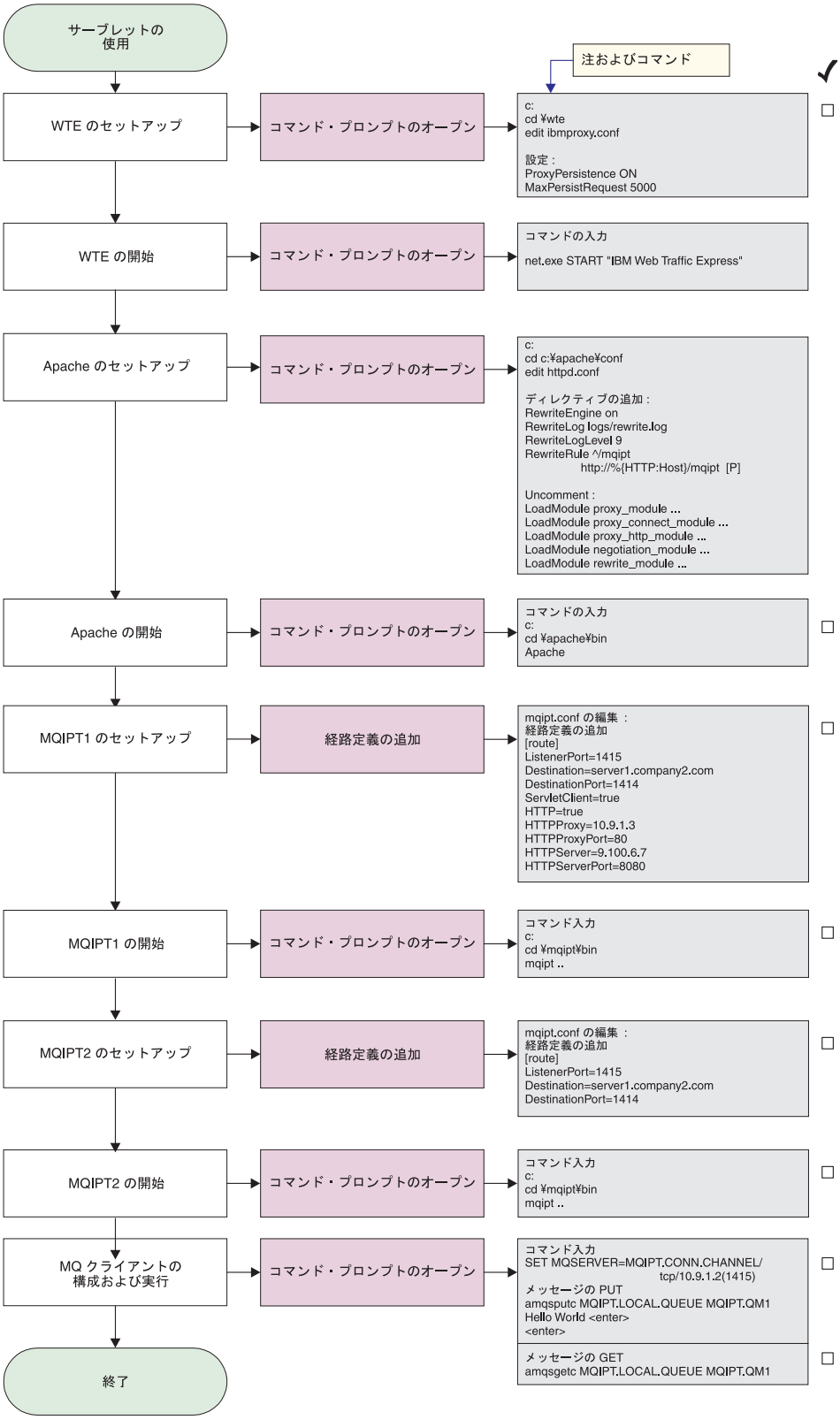

図 39. Apache 再書き込み構成

1. WTE 上で
   `c:¥wte¥ibmroxy.conf` を編集します

以下のプロパティーを変更します

```
ProxyPersistence ON
MaxPersistRequest  5000
```

2. Apache 上で

   c:¥apache¥conf¥httpd.conf を編集します

```
RewriteEngine on
RewriteLog logs/rewrite.log
RewriteLogLevel 9
RewriteRule ^/mqipt  http://%{HTTP:Host}/mqipt  [P]

LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule rewrite_module modules/mod_rewrite.so

start Apache
```

3. IPT1 上で

   mqipt.conf を編集し、経路定義を追加します。

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
HTTP=true
HTTPProxy=10.9.1.3
HTTPProxyPort=80
HTTPServer=9.100.6.7
HTTPServerPort=8080
```

   次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

   以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using HTTP
MQCPI024 ....and HTTP proxy at 10.9.1.3(80)
MQCPI066 ....and HTTP server at 9.100.6.7(8080)
MQCPI078 Route 1415 ready for connection requests
```

4. IPT2 上で

   mqipt.conf を編集し、経路定義を追加します。

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
```

   次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from C:¥mqipt¥mqipt.conf
MQCPI011 The path C:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI078 Route 1415 ready for connection requests
```

5. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

6. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world <enter>
<enter>
```

7. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

「Hello world (ようこそ)」が表示されます。

セキュリティー出口

この例について、以下のような前提事項を想定しています。

- Java 1.4 SDK がインストールされている
- Java bin サブディレクトリーが PATH 環境変数に追加されている

これは、SampleSecurityExit と呼ばれる提供されたサンプルのセキュリティー出口の使用方法を示すための簡単なテストです。このセキュリティー出口は、文字「MQIPT.」で始まるチャネル名を使用して、クライアント接続のみができるようにするように書かれています。

″MQIPT.CONN.CHANNEL″ (これらのサンプルの大部分で使用されている) という推奨 srvconn チャネル名を使用して、クライアント接続は完了でき、WMQ メッセージをキューに入れることができます。

セキュリティー出口が予期したとおり機能していることを証明するために、文字「MQIPT.」で始まらない任意の名前 (たとえば、「TEST.CONN.CHANNEL」) をもつ別の srvconn チャネルを定義して、amqsputc コマンドを再度試行します。しかし、新しいチャネル名を使用するために MQSERVER 環境変数は変更されています。今回接続は拒否され、2059 エラーが表示されます。

セキュリティー出口を使用せずに、″TEST.CONN.CHANNEL″ が機能していることを示すには、WMQ リスナー・ポート (たとえば、1414) を直接指すように MQSERVER 環境変数を設定します。したがって、MQIPT は使用されていません。今回 amqsputc コマンドは予期したとおりに機能します。

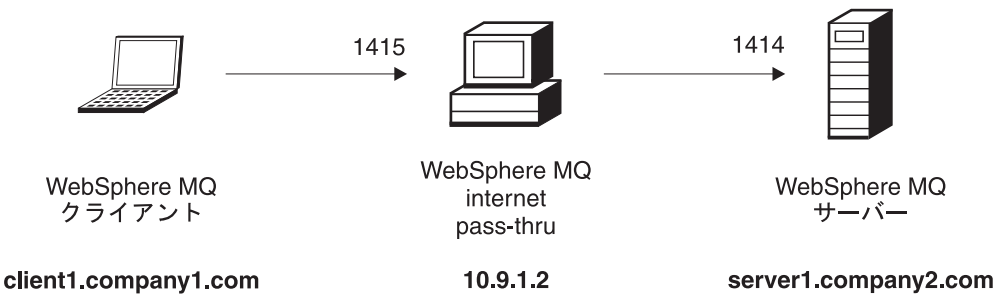

図 40. セキュリティー出口ネットワーク・ダイアグラム

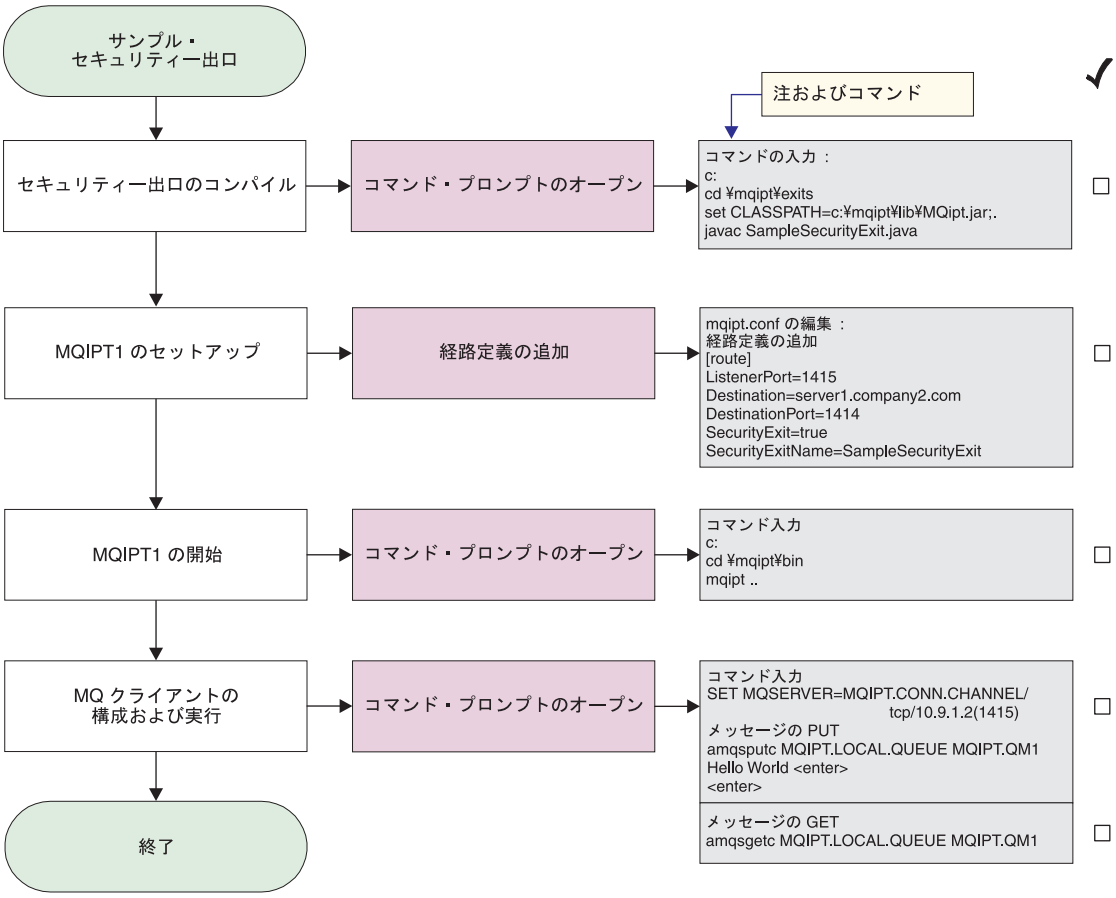

図 41. セキュリティー出口構成

1. IPT1 上で

   次のように、コマンド・プロンプトをオープンします。

   ```
   c:
   cd ¥mqipt¥exits
   set CLASSPATH=c:¥mqipt¥lib¥MQipt.jar;.
   javac SampleSecurityExit.java
   ```

   `mqipt.conf` を編集し、経路定義を追加します。

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SecurityExit=true
SecurityExitName=SampleSecurityExit
```

次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from c:¥mqipt¥mqipt.conf
MQCPI011 The path c:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI079 ....using security exit c:¥mqipt¥exits¥SampleSecurityExit
MQCPI080 ......and timeout of 5 seconds
MQCPI078 Route 1415 ready for connection requests
```

2. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

3. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world <enter>
<enter>
```

4. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

「Hello world (ようこそ)」が表示されます。

セキュリティー出口のルーティング

この例について、以下のような前提事項を想定しています。

- Java 1.4 SDK がインストールされている
- Java bin サブディレクトリーが PATH 環境変数に追加されている
- 3 つの同一のキュー・マネージャーが 3 つの別々のサーバーに作成されている

これは、ラウンドロビン方式でクライアント接続要求を WMQ Queue Manager サーバーのグループに動的にルーティングする実例です。グループ内の各サーバー上の Queue Manager は、対応するもう一方のミラー・イメージでなければなりません。

サーバー名のリストは、構成ファイルから読み取られます。構成ファイルの名前と位置は、SecurityExitName プロパティーと SecurityExitPath プロパティーで定義されます。SampleRoutingExit.conf と呼ばれる、サンプルの構成ファイルには、以下のエントリーが入っています。

```
server1.company.com:1414
server2.company.com:1415
server3.company.com:1416
```

これらのサーバー名は、ご使用の環境に合わせて変更する必要があります。

amqsputc コマンドが初めて出されると、WMQ メッセージが server1 上の QM の MQIPT.LOCAL.QUEUE に入れられます。次回このコマンドが出されると、メッセージは順次 server2 上の QM に入れられます。このセットアップを使用すると、amqsgetc コマンドが使用するクライアント接続要求がリストの次の QM に渡されるため、amqsgetc コマンドはキューに入れられたメッセージを取り出すことができません。しかし、3 つの amqsputc コマンドに、3 つの amqsgetc コマンドを続けて出すと、確実に各メッセージが同じ順序で取り出されます。もちろん、別の WMQ クライアントを使用することによって、QM (すなわち、このサンプルでは MQIPT を使用していない) に直接接続すれば、どのキュー・マネージャーからもメッセージを選択して取り出すことができます。

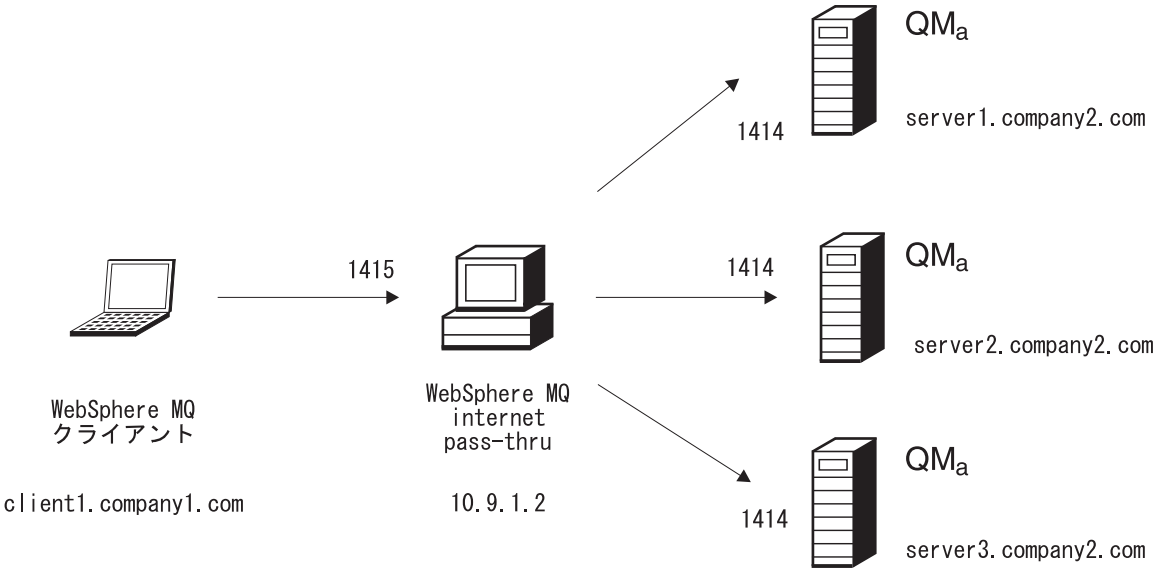

図 42. ルーティング・セキュリティー出口ネットワーク・ダイアグラム

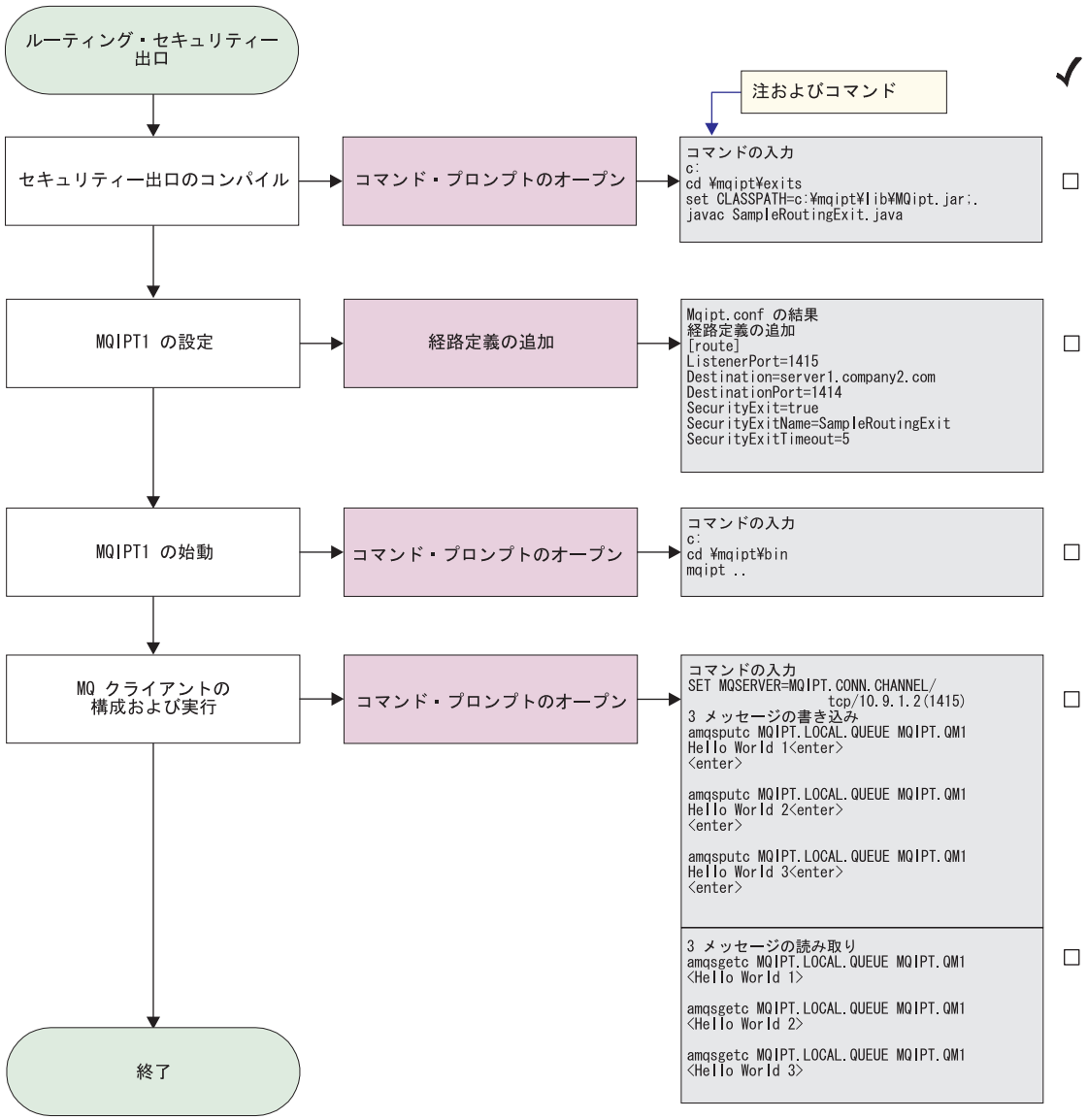

図 43. ルーティング・セキュリティー出口構成

1. IPT1 上で

   次のように、コマンド・プロンプトをオープンします。

   ```
   c:
   cd ¥mqipt¥exits
   set CLASSPATH=c:¥mqipt¥lib¥MQipt.jar;.
   javac SampleRoutingExit.java
   ```

   mqipt.conf を編集し、経路定義を追加します。

   ```
   [route]
   ListenerPort=1415
   Destination=server1.company2.com
   DestinationPort=1414
   SecurityExit=true
   SecurityExitName=SampleRoutingExit
   ```

   次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from c:¥mqipt¥mqipt.conf
MQCPI011 The path c:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI079 ....using security exit c:¥mqipt¥exits¥SampleRoutingExit
MQCPI080 ......and timeout of 5 seconds
MQCPI078 Route 1415 ready for connection requests
```

2. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

3. 以下のコマンドを使用して 3 つのメッセージを書き込みます。

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 1 <enter>
<enter>
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 2 <enter>
<enter>
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 3 <enter>
<enter>
```

4. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

「Hello world 1 (ようこそ 1)」、「Hello world 2 (ようこそ 2)」、および「Hello world 3 (ようこそ 3)」が表示されます。

動的 1 経路出口

この例について、以下のような前提事項を想定しています。

- Java 1.4 SDK がインストールされている
- Java bin サブディレクトリーが PATH 環境変数に追加されている
- 3 つの異なるキュー・マネージャーが 3 つの別々のサーバーに作成されている

これは、使用しているチャネルの名前に基づいて、ターゲット・サーバーにクライアント接続要求を動的にルーティングする方法を示す実例です。チャネル名の最初の部分は、Queue Manager の名前で、たとえば、QM1 に接続する場合、svrconn チャネルの名前は QM1.MQIPT.CONN.CHANNEL となります。このチャネルの命名規則を使用すれば、すべての接続要求をサービスするために MQIPT は 1 つの経路のみの使用を必要とします。

Queue Manager およびサーバー名のリストは、構成ファイルから読み取られます。構成ファイルの名前と位置は、SecurityExitName プロパティーと SecurityExitPath プ

ロパティーで定義されます。SampleOneRouteExit.conf と呼ばれる、サンプルの構成ファイルには、以下のエントリーが入っています。

```
QM1   server1.company.com:1414
QM2   server2.company.com:1415
QM3   server3.company.com:1416
```

これらのサーバー名は、ご使用の環境に合わせて変更する必要があります。

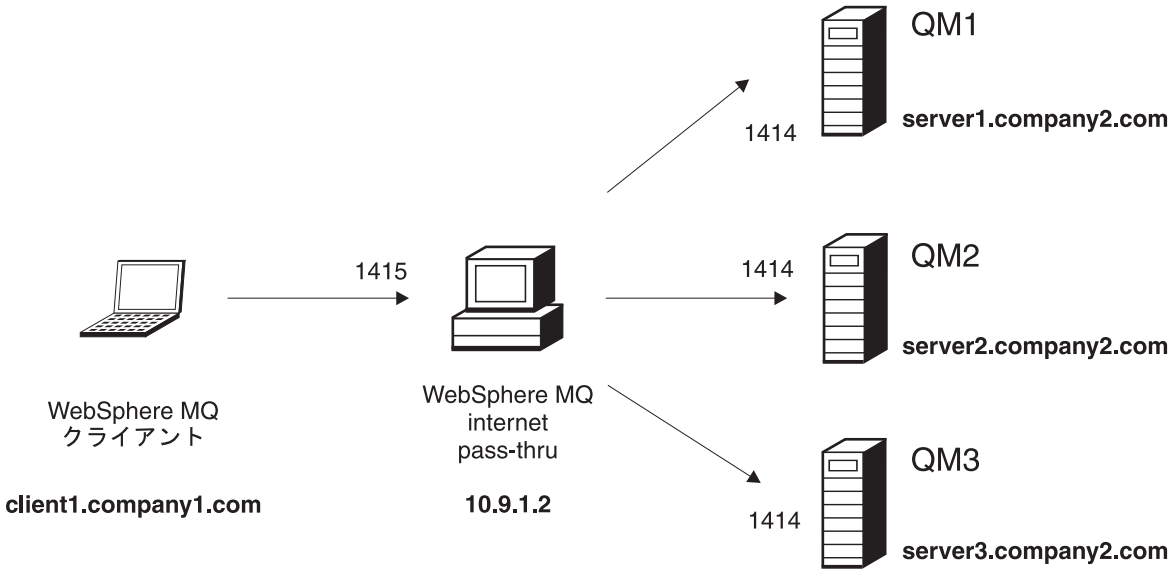

図 44. 動的 1 経路出口ネットワーク・ダイアグラム

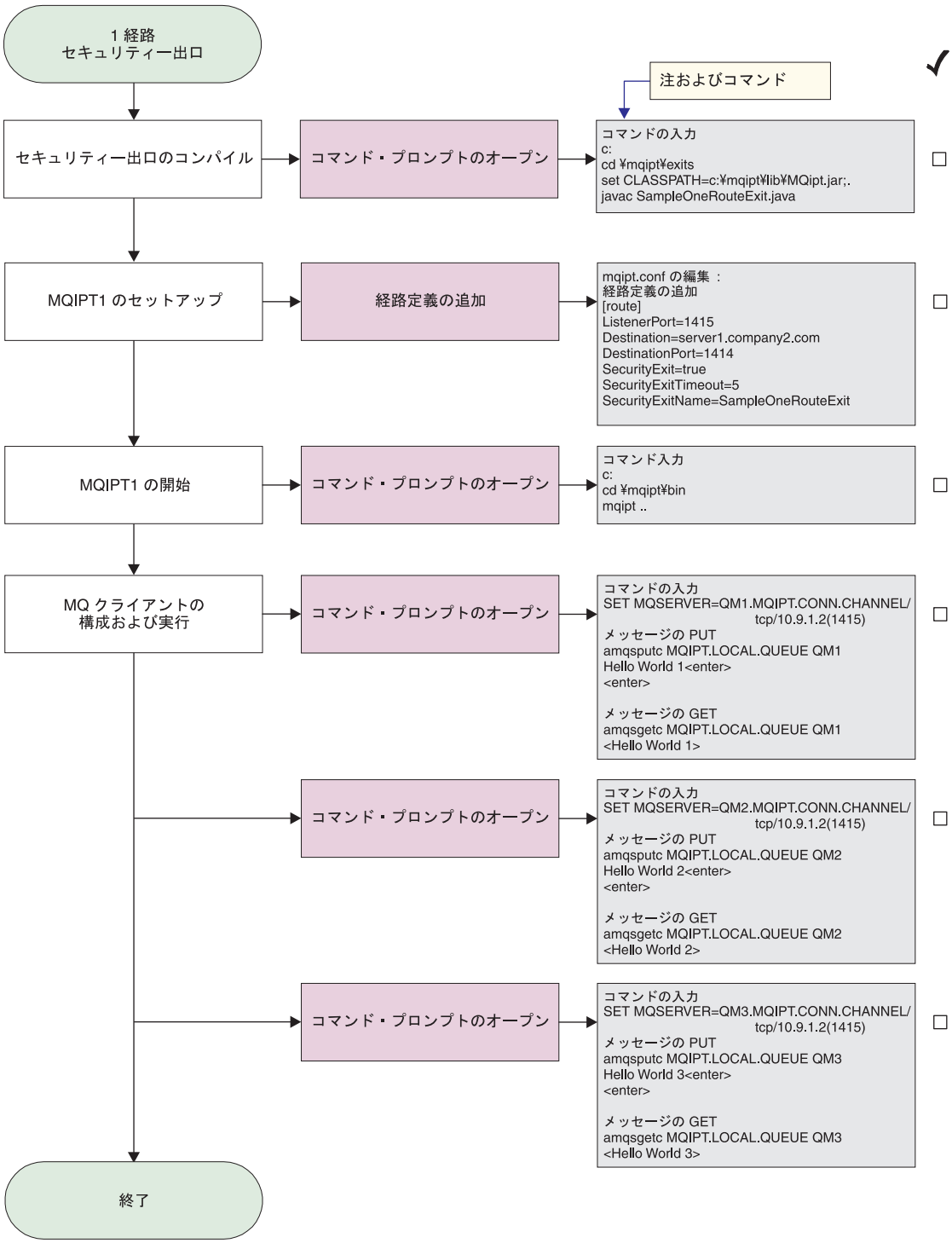

図 45. 動的 1 経路出口構成

1. IPT1 上で

   次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥exits
set CLASSPATH=c:¥mqipt¥lib¥MQipt.jar;.
javac SampleOneRouteExit.java
```

mqipt.conf を編集し、経路定義を追加します。

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SecurityExit=true
SecurityExitName=SampleOneRouteExit
```

次のように、コマンド・プロンプトをオープンします。

```
c:
cd ¥mqipt¥bin
mqipt ..
```

以下のメッセージが正常終了を示します。

```
5639-L92 (C) Copyright IBM Corp. 2000, 2003 All Rights Reserved
MQCPI001 Websphere MQ internet pass-thru Version 1.3.0 starting
MQCPI004 Reading configuration information from c:¥mqipt¥mqipt.conf
MQCPI011 The path c:¥mqipt¥logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocols
MQCPI079 ....using security exit c:¥mqipt¥exits¥SampleOneRouteExit
MQCPI080 ......and timeout of 5 seconds
MQCPI078 Route 1415 ready for connection requests
```

2. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=QM1.MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
   ```

3. 以下のコマンドを使用してメッセージを入力します。

   ```
   amqsputc MQIPT.LOCAL.QUEUE QM1
   Hello world 1 <enter>
   <enter>
   ```

4. 以下のコマンドを使用してメッセージを入手します。

   ```
   amqsgetc MQIPT.LOCAL.QUEUE QM1
   ```

   「Hello world 1 (ようこそ 1)」が表示されます。

5. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=QM2.MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
   ```

6. 以下のコマンドを使用してメッセージを入力します。

   ```
   amqsputc MQIPT.LOCAL.QUEUE QM2
   Hello world 2 <enter>
   <enter>
   ```

7. 以下のコマンドを使用してメッセージを入手します。

   ```
   amqsgetc MQIPT.LOCAL.QUEUE QM2
   ```

   「Hello world 2 (ようこそ 2)」が表示されます。

8. WebSphere MQ クライアント・マシンのコマンド・プロンプトに対して、次のように入力します。

   ```
   SET MQSERVER=QM3.MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
   ```

9. 以下のコマンドを使用してメッセージを入力します。

```
amqsputc MQIPT.LOCAL.QUEUE QM3
Hello world 3 <enter>
<enter>
```

10. 以下のコマンドを使用してメッセージを入手します。

```
amqsgetc MQIPT.LOCAL.QUEUE QM3
```

「Hello world 3 (ようこそ 3)」が表示されます。

第 21 章 internet pass-thru の維持

この章では、internet pass-thru の稼働を維持する方法について説明します。この章には、以下のセクションがあります。

- 『保守』
- 『問題判別』
- 158 ページの『パフォーマンス・チューニング』

保守

通常のバックアップ手順の一環として、定期的に以下のファイルのバックアップをとる必要があります。

- mqipt.conf 構成ファイル
- 以下のプロパティーで定義された mqipt.conf 内の SSL 鍵リング・ファイル。
  - SSLClientKeyRing
  - SSLClientCAKeyRing
  - SSLServerKeyRing
  - SSLServerCAKeyRing
- 以下のプロパティーで定義された mqipt.conf 内の SSL 鍵リング・パスワード・ファイル。
  - SSLClientKeyRingPW
  - SSLClientCAKeyRingPW
  - SSLServerKeyRingPW
  - SSLServerCAKeyRingPW
- Administration Client 構成ファイル (client.conf)。このファイルには、Administration Client に認識されているすべての MQIPT に関する接続情報が収められています。

問題判別

問題が発生したかどうかを最初に調べる場合、以下のようないくつかの共通な落とし穴があります。

- MQIPT システムがインストールされたばかりで、まだリブートされていない。
- キュー・マネージャーに直接接続された経路で HTTP が true に設定されている。
- キュー・マネージャーに直接接続された経路で SSLClient が true に設定されている。
- CLASSPATH が正しく設定されていない。
- PATH が正しく設定されていない。
- 鍵リング・ファイル用に保管されたパスワードに大文字小文字の区別がある。

次のステップでは、図 46に示されているフローチャートに従います。数字は、下に示されている注の番号を指しています。

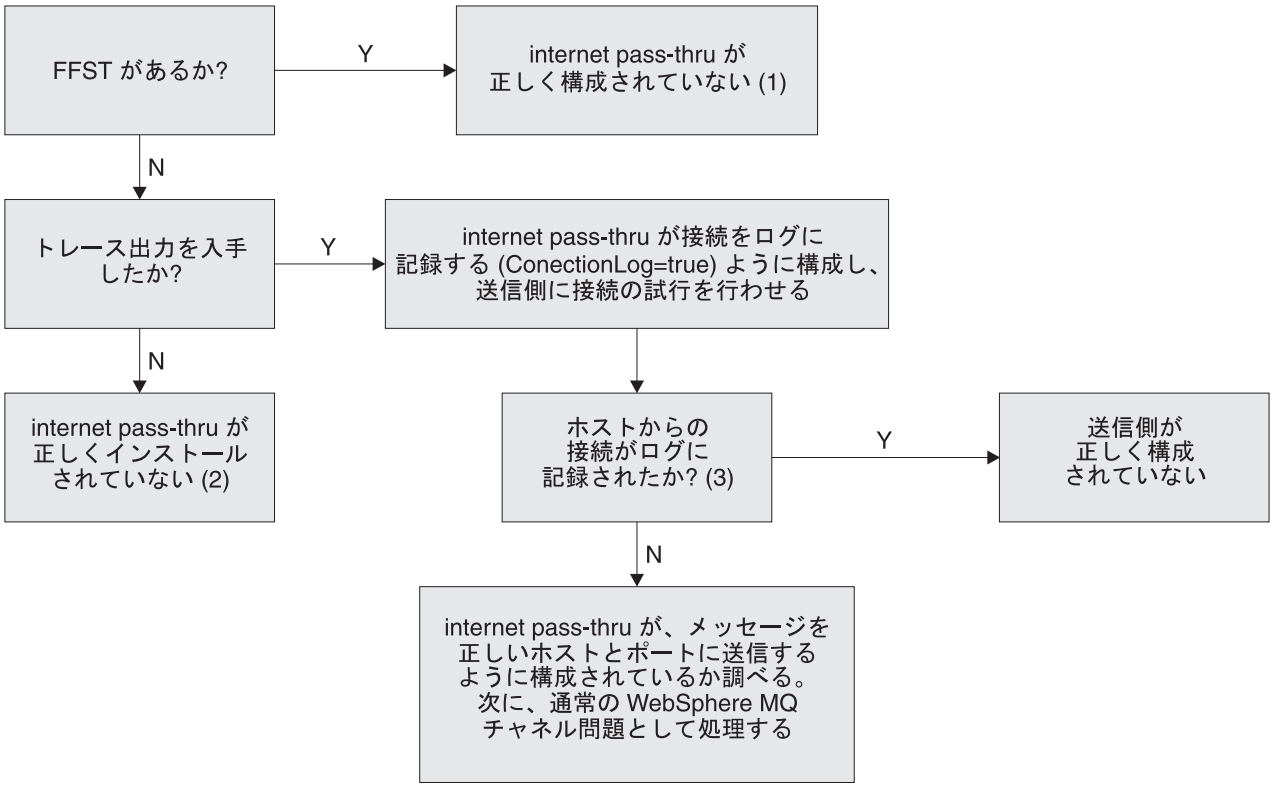

図 46. 問題判別フローチャート

注:

1. FFST レポートが (errors サブディレクトリーに) 見つかると、MQIPT が正しくインストールされていることが分かります。構成に問題があるかもしれません。

   各 FFST は、MQIPT、または経路の始動プロセスを終了させる問題を報告します。各 FFST の原因になった問題を修正してください。次に、古い FFST を削除するか、または MQIPT をリフレッシュしてください。
2. MQIPT が正しくインストールされていない場合は、すべてのファイルが正しい位置に置かれ、CLASSPATH が更新されていることを確認します。これが正しいかどうかを調べるには、MQIPT を手動で開始してみてください。
3. 手動で MQIPT を開始します。

   コマンド・プロンプトをオープンします。 `bin` サブディレクトリーへ進み、次のように入力します。

   ```
   mqipt xxx
   ```

   ここで、*xxx* は MQIPT ホーム・ディレクトリーです。この場合は、それは “`..`” です。

   これにより、MQIPT が開始し、構成を見つけるためにホーム・ディレクトリーを探します。エラー・メッセージや FFST がないか、`errors` サブディレクトリーを探します。

エラー・メッセージがないか、MQIPT のテキスト出力を調べ、エラーを訂正します。 FFST を調べ、エラーがあればそれを訂正します。構成ファイルのグローバル・セクションに問題があれば、MQIPT は開始しません。構成ファイルの経路セクションに問題があれば、経路は開始しません。

internet pass-thru の自動的開始

MQIPT を Windows NT Service としてインストールする場合、その開始が自動的に行われるように変更すると、MQIPT はシステムの始動時に自動的に開始されます。インストールが正しく行われたことを確認するには、MQIPT を Windows NT Service としてインストールする前に必ず MQIPT を 1 回手動で開始してください。詳細については、52 ページの『Windows サービス制御プログラムの使用』を参照してください。

エラー・メッセージ “Unable to locate DLL...” が出た場合は、間違った `mqiptService` プログラムを使用しているか、またはシステムの PATH 環境変数が正しく構成されていません。 PATH には、JNI ランタイム・ライブラリーのロケーションが入っていなければなりません。このファイル (`jvm.dll`) は、JDK のクライアント・サブディレクトリーに入っています。

エンドツーエンド接続の検査

MQIPT が正しくインストールされたならば、次にとるステップは、経路が正しくセットアップされているかどうかの検査です。

`mqipt.conf` 構成ファイルでは、`ConnectionLog` プロパティーを true に設定します。 MQIPT を開始またはリフレッシュして接続を試みます。ホーム・ディレクトリーの下のログ・ディレクトリーに接続ログが作成されます。接続ログが作成されない場合は、MQIPT が正しくインストールされていないことがわかります。接続の試行が記録されない場合は、送信側が正しくセットアップされていません。接続の試行が記録されていれば、MQIPT がメッセージを正しいアドレスに転送しているかどうか調べてください。

エラーのトレース

MQIPT は、詳細な実行トレース機能を提供します。この機能は、トレース属性によって制御されます。経路はそれぞれ独立してトレースできます。トレース・ファイルは *xxx*¥errors ディレクトリーに書き込まれます (ここで、*xxx* は `mqipt.conf` が入っているディレクトリーです)。作成された各トレース・ファイルには、次のような形式の名前が付けられています。

`iptroute`*nnnnn*`.trc`

ここで、*nnnnn* は、経路が listen するポートの番号です。特定の経路に直接関連していないスレッド (たとえば、コマンド入力を処理するスレッド) からのトレース出力は、`iptmain.trc` という別個のファイルに書き込まれます。

予期しない致命的エラーは、FFST レコードとしてエラー・ログ・ファイルに書き込まれ、*xxx*¥errors ディレクトリーに保持されます (ここで、*xxx* は、`mqipt.conf` が入っているディレクトリーです)。FFST ファイルの形式は次のようになっています。

`ipt`*xxx*`.FFST`

ここで、*xxx* は、FFST が生成された順序です (1 が最も古いものです)。長時間実行システムでは、システムで生成可能な最大数に達することがあります。この場合は、生成されたすべての FFST が `mqipt0.FFST` ファイルに書き込まれます。`mqipt0.FFST` ファイルが作成された場合は、都合のつき次第、MQIPT を停止して再始動し、古いファイルを削除する必要があります。

問題の報告

問題を IBM サービス・センターに報告する必要がある場合は、以下の情報を提供していただくと、問題の解決が速まることがあります。

- 使用している簡単なネットワーク・ダイアグラム (IP アドレスを含む) を提供する。
- 複数の MQIPT を使用している場合は、各 MQIPT マシンのシステム・クロックを同期化する。こうしておけば、各 MQIPT のトレース・エントリーを突き合わせる際に役立ちます。
- 古いトレース・ファイルを削除する。
- クライアントを実行して問題を作成する。こうすれば、トレース・ファイルには、問題のインスタンスが 1 つしか入りません。
- すべての MQIPT .trc および .log ファイルのコピーを送信する。

パフォーマンス・チューニング

ここでは、システムをチューニングする場合のいくつかのヒントを示します。

スレッド・プール管理

各経路の相対パフォーマンスは、スレッド・プールとアイドル・タイムアウト仕様を組み合わせて使用してチューニングすることができます。

接続スレッド

各 MQIPT 経路には、着信通信要求を処理している、並行して実行しているスレッドの作業プールが割り当てられています。初期化時に、スレッドのプールが作成され (そのサイズは、経路の `MinConnectionThreads` 属性に指定されている)、1 つのスレッドが、最初の着信要求を処理するように指名されます。この要求が到着すると、そのスレッドはこの要求の処理を即時に開始し、その次のスレッドが、次の着信要求を処理するように割り当てられます。すべてのスレッドが作業を割り当てられていると、新規のスレッドが作成されて作業プールに追加され、作業が割り当てられます。このようにして、プールは `MaxConnectionThreads` に達するまで増大します。作業スレッドの数が `MaxConnectionThreads` に達すると、次の着信要求は、1 つのスレッドが解放されて作業プールに戻されるまで待ちます。これが経路の最大作業容量であり、この限度を超えると、追加の要求は受け入れられません。会話が終了するか、または指定されたタイムアウト期間を過ぎると、スレッドはプールに戻されます。

アイドル・タイムアウト

デフォルトでは、非アクティブ状態になっていることが理由で作業スレッドが終了させられることはありません。あるスレッドをある会話に割り当てると、そのスレッドは、その会話が正常終了するか、経路が非活動になるか、または MQIPT がシ

ャットダウンするまでその会話に割り当てられたままになっています。オプションで、アイドル・タイムアウト間隔を指定できるため、指定された期間 (分単位) 非アクティブ状態になっているすべてのスレッドが終了します。モニター・スレッドは、スレッド・アイドル時間について定期的な検査を行い、しきい値を超えたスレッドを終了させます。スレッドは、作業プールに戻されてリサイクルされます。

第 22 章 メッセージ

MQIPT をコマンド行から実行すると、MQIPT は、少数の通知メッセージ、警告メッセージ、およびエラー・メッセージをコンソール上に表示します。

以下の点に注意してください。

- MQCAxxxx メッセージは Administration Client メッセージです。
- MQCPxxxx メッセージは MQIPT メッセージです。
- MQCxIxxx メッセージは通知メッセージです。
- MQCxWxxx メッセージは警告メッセージです。
- MQCxExxx メッセージはエラー・メッセージです。

MQCAE001 Unknown host: {0}

説明: MQIPT ホストが見つかりません。

ユーザーの処置: MQIPT の所在を示すホスト名が正しく指定されているか調べてください。

MQCAE002 The following error was reported by the system: {0}

説明: エラーが起こりました。システム・コマンドの実行中に、エラーが報告されました。

MQCAE005 No valid destination address has been defined

説明: 経路の追加操作で、宛先フィールドがブランクのまま残されました。

ユーザーの処置: 有効な宛先アドレスを入力してください。

MQCAE006 No valid destination port has been defined

説明: 経路の追加操作で、宛先ポート・アドレス・フィールドがブランクのまま残されました。

ユーザーの処置: 有効な宛先ポート・アドレスを入力してください。

MQCAE007 No valid listener port has been defined

説明: 経路の追加操作で、リスナー・ポート・アドレス・フィールドがブランクのまま残されました。

ユーザーの処置: 有効なリスナー・ポート・アドレス (1 ～ 65535) を入力してください。

MQCAE008 No valid network address has been defined

説明: MQIPT の追加操作で、ネットワーク・アドレス・フィールドがブランクのまま残されました。

ユーザーの処置: 有効なネットワーク・アドレスを入力してください。

MQCAE009 No valid command port has been defined

説明: MQIPT の追加操作で、無効なコマンド・ポート・アドレスが使用されました。

ユーザーの処置: 有効なコマンド・ポート・アドレス (1 ～ 65535) を入力してください。

MQCAE010 Could not show online help

説明: オンライン・ヘルプのファイルはありますが、表示できません。

ユーザーの処置: Web ブラウザーがインストールされていて、システム PATH 環境変数で使用可能であることを確認してください。

MQCAE011 Could not parse parameter

説明: 内部エラーが発生して、テーブルに入っていないパラメーターを更新しようとしました。

ユーザーの処置: この状態が継続する場合は、IBM 技術支援に連絡してください。

MQCAE012 Could not find online help file {0}

説明: ″passtfrm.htm″ ファイルが見つかりません。

ユーザーの処置: このファイルが doc 言語サブディレクトリーでアクセス可能であることを確認してください。

MQCAE013 Interrupted while trying to show online help

説明: オンライン・ヘルプを表示しているときにシステム・エラーが起こりました。

ユーザーの処置: もう一度操作を行ってください。この状態が継続する場合は、IBM 技術支援に連絡してください。

MQCAE015 The password you have just entered has not been recognized

説明: MQIPT は有効なパスワードを期待しています。最後に使用したパスワードが間違っています。パスワードは、構成ファイルに定義したものと一致していなければなりません。

ユーザーの処置: 「**MQIPT**」->「**Connection (接続)**」パネルを使用してそのパスワードを変更し、もう一度最後のコマンドを実行してみてください。

MQCAE016 Node mismatch

説明: ツリーで選択したノードとメモリー内のデータ間に内部矛盾があります。

ユーザーの処置: Administration Client をクローズして、もう一度コマンドを実行してみてください。この状態が継続する場合は、IBM 技術支援に連絡してください。

MQCAE017 Could not create NLS text for message {0}

説明: 定義されたメッセージ番号に該当する NLS テキストが見つかりません。

ユーザーの処置: ″guiadmin.properties″ ファイルが破壊されたため、指定されたメッセージ番号が見つからないのかもしれません。以下の検査を行ってください。

- 該当する新規メッセージが README ファイルに入っているか
- ″guiadmin.jar″ ファイルがシステム CLASSPATH に入っているか
- ″guiadmin.properties″ ファイルが ″guiadmin.jar″ ファイルに入っているか
- メッセージ番号が ″guiadmin.properties″ ファイルに入っているか

MQCAE018 Could not create NLS text for message MQCAE017

説明: メッセージ番号 {0} がシステム・プロパティー・リストに入っていません。

ユーザーの処置: ″guiadmin.properties″ ファイルが破壊されている可能性があります。以下の検査を行ってください。

- ″guiadmin.jar″ ファイルがシステム CLASSPATH に入っているか
- ″guiadmin.properties″ ファイルが ″guiadmin.jar″ ファイルに入っているか
- メッセージ番号が ″guiadmin.properties″ ファイルに入っているか

MQCAE019 You have failed to repeat your proposed new password

説明: パスワードの変更時に、検証用の 2 回の入力が行われませんでした。

ユーザーの処置: 新規パスワードを該当フィールドにもう一度入力してください。

MQCAE020 Failed to change MQIPT access parameters

説明: MQIPT アクセス・パラメーターの変更時に、内部エラーが検出されました。

ユーザーの処置: Administration Client をクローズして、もう一度コマンドを実行してみてください。この状態が継続する場合は、IBM 技術支援に連絡してください。

MQCAE021 Internal failure to identify MQIPT

説明: 構成ファイルを MQIPT に保管中に、内部エラーが検出されました。

ユーザーの処置: Administration Client をクローズして、もう一度コマンドを実行してみてください。この状態が継続する場合は、IBM 技術支援に連絡してください。

MQCAE022 Internal failure to save MQIPT configuration

説明: 構成ファイルを MQIPT に保管中に、内部エラーが検出されました。

ユーザーの処置: Administration Client をクローズし

て、もう一度コマンドを実行してみてください。この状態が継続する場合は、IBM 技術支援に連絡してください。

MQCAE023 MQIPT {0} did not recognize your password.

説明: MQIPT は有効なパスワードを期待しています。最後に使用したパスワードが間違っています。パスワードは、構成ファイルに定義したものと一致していなければなりません。

ユーザーの処置: 「**MQIPT**」->「**Connection (接続)**」メニューを使用してそのパスワードを変更し、もう一度コマンドを実行してみてください。

MQCAE024 MQIPT {0} has not recognized the command.

説明: Administration Client が MQIPT に送信したコマンドが認識されません。

ユーザーの処置: Administration Client が使用したコードのバージョンが MQIPT と同じであるか調べてください。

MQCAE025 MQIPT {0} has failed to send configuration file.

説明: MQIPT が構成ファイルを送信しようとして失敗しました。

ユーザーの処置: Administration Client をクローズして、もう一度コマンドを実行してみてください。それでもうまくいかない場合は、MQIPT をいったん停止した後、再始動してください。

MQCAE026 Remote shutdown is disabled on MQIPT {0}.

説明: リモート・シャットダウンが構成ファイルで使用可能になっていないため、MQIPT をリモート側でシャットダウンしようとして失敗しました。

ユーザーの処置: MQIPT のリモート・シャットダウンを使用可能にするには、構成ファイルを編集し、RemoteShutDown プロパティーを true に設定します。

MQCAE027 Look and feel {0} is not supported.

説明: 使用しているプラットフォーム用の推奨ルック・アンド・フィールは使用できません。

ユーザーの処置: システム・デフォルトのルック・アンド・フィールで処理が続行されます。

MQCAE028 Look and feel class {0} cannot be found.

説明: 使用しているプラットフォーム用の推奨ルック・アンド・フィールは使用できません。

ユーザーの処置: システム・デフォルトのルック・アンド・フィールで処理が続行されます。

MQCAE029 Minimum Connection Threads must be non-negative and no bigger than Maximum Connection Threads

説明: 最小接続スレッド値は、最大接続スレッド値以下でなければなりません。

ユーザーの処置: 値を適宜変更してください。

MQCAE030 Maximum Connection Threads must be greater than zero and at least as big as Minimum Connection Threads

説明: 最大接続スレッド値は、最小接続スレッド値より大きくなければなりません。

ユーザーの処置: 値を適宜変更してください。

MQCAE031 Port numbers must be in the range 0 to 65535

説明: 仕様に合致しない値を設定しようとしています。

ユーザーの処置: 値を適宜変更してください。

MQCAE032 Trace must be in the range 0 to 5

説明: 仕様に合致しない値を設定しようとしています。

ユーザーの処置: 値を適宜変更してください。

MQCAE033 Max Log file size must be in the range 5 to 50

説明: 仕様に合致しない値を設定しようとしています。

ユーザーの処置: 値を適宜変更してください。

MQCAE049 No route has been selected on any MQIPT

説明: 削除する経路をまず選択しないで、その経路を削除しようとしました。

ユーザーの処置: 経路を選択してから、もう一度コマンドを実行してみてください。

MQCAE050 Could not connect to MQIPT {0}

説明: Administration Client が、指定された MQIPT に接続できません。

ユーザーの処置: この原因としては、以下のことが考えられます。

- MQIPT が稼働していない。
- MQIPT が自分のコマンド・ポートで listen していない。
- 1 つの Administration Client しか MQIPT CommandPort を使用していない。
- 要求がタイムアウトになった。

MQCAE051 Could not read reply from MQIPT {0}

説明: MQIPT から応答が送られてきましたが、それが所定のプロトコルに準拠していません。

ユーザーの処置: Administration Client が使用したコードのバージョンが MQIPT と同じであるか調べてください。

MQCAE052 Configuration has not been saved

説明: MQIPT から有効な応答が送られてきましたが、後続の構成ファイルへの保管に失敗しました。

ユーザーの処置: MQIPT が構成ファイルへの書き込みアクセス権限を持っているか調べてください。

MQCAE053 MQIPT has not confirmed saving of configuration

説明: 構成ファイルが MQIPT へ送信されましたが、MQIPT はその確認に失敗しました。

ユーザーの処置: この原因としては、以下のことが考えられます。

- MQIPT が稼働していない。
- MQIPT が自分のコマンド・ポートで listen していない。
- 1 つの Administration Client しか MQIPT CommandPort を使用していない。
- 要求がタイムアウトになった。

MQCAE054 MQIPT data has not been refreshed

説明: MQIPT に連絡しましたが、Administration Client は構成ファイルを読み取ることができませんでした。

ユーザーの処置: この原因としては、以下のことが考えられます。

1. MQIPT が失敗した。
2. 要求がタイムアウトになった。

MQCAE055 No MQIPT or route on an MQIPT has been selected

説明: MQIPT または経路が選択されていないため、ユーザーが選択したメニュー・オプションを実行できません。

ユーザーの処置: 適切な MQIPT または経路を選択して、もう一度実行してみてください。

MQCAE056 Duplicate listener port has been rejected

説明: 指定されたリスナー・ポートが別の経路で使用されているため、そのポートが拒否されました。

ユーザーの処置: 別のリスナー・ポートを選択して、もう一度実行してみてください。

MQCAI002 The MQIPT has been removed from display

説明: ツリーから選択したノードの MQIPT が、クライアントのメモリーから除去されました。

MQCAI003 New route added to the display

説明: 今指定した新規経路が現行の MQIPT に追加されました。

MQCAI004 Route has been removed from the display

説明: ツリーから選択した経路がクライアントのメモリーから除去されました。

MQCAI005 Selected MQIPT is being displayed

説明: ツリーから選択した MQIPT のグローバル・パラメーターがテーブルに示されています。

MQCAI006 Selected route is being displayed

説明: ツリーから選択した経路のパラメーターがテーブルに示されています。

MQCAI007 Client configuration has been saved

説明: ツリー上のすべての MQIPT に関するアクセス・パラメーターが保管されました。

MQCAI008 Display of online help succeeded

説明: オンライン・ヘルプが要求どおりに表示されました。

MQCAI009 Table has been updated

説明: テーブルに入力された値を使用して、メモリー内のモデルが更新されました。

MQCAI010 No MQIPT or route has been selected.

説明: アクションをとるための情報が不十分なため、アクションがとられませんでした。

MQCAI011 User Action has been cancelled

説明: 開始済みのポップアップ・ウィンドウ関連のアクションがキャンセルされました。

MQCAI014 Configuration has been saved on MQIPT

説明: 現在ツリー上で選択されている MQIPT に新規の構成ファイルが保管され、それを使用して MQIPT が再始動されました。

MQCAI015 Online help has terminated

説明: オンライン・ヘルプが要求どおりに表示され、その後で終了しました。

MQCAI017 Select File/Add MQIPT to add an MQIPT to the tree

説明: このメッセージは、ツリー上に MQIPT がないときに表示され、その追加方法を知らせます。

MQCAI018 New MQIPT added to display

説明: 指示どおり、新規の MQIPT がツリーに追加されました。

MQCAI019 MQIPT access parameters have been changed

説明: 現在ツリー上で選択されている MQIPT のアクセス・パラメーターが変更されました。

MQCAI021 Select an MQIPT or route on the tree to display its contents

説明: このメッセージは、情報がテーブルに表示されていないときに表示され、その表示方法を知らせます。

MQCAI022 The command port has changed

説明: 変更を指示された MQIPT のコマンド・ポートが今変更されました。

MQCAI023 The password has changed

説明: 今後、変更された MQIPT と通信する場合は、この新規パスワードが使用されます。

MQCAI025 MQIPT {0} has been refreshed.

説明: MQIPT に関して保持されている情報が、その構成ファイルの読み取りで更新されました。

MQCAI026 MQIPT {0} has received shutdown request.

説明: MQIPT がシャットダウン要求の受信を確認して、今シャットダウンするところです。

MQCAI027 Client configuration has been refreshed

説明: Administration Client に表示されている情報が、ローカル "client.conf" ファイルからリフレッシュされました。

MQCAI028 MQIPT {0} is active

説明: MQIPT が PING 要求に正常に応答しました。

MQCAI029 MQIPT {0} is not active

説明: MQIPT が、指定時間内に PING 要求に応答しませんでした。

ユーザーの処置: この原因としては、以下のことが考えられます。

- MQIPT が稼働していない。
- MQIPT が自分のコマンド・ポートで listen していない。
- 要求がタイムアウトになった。このタイムアウト時間は、MQIPT に関する接続情報のタイムアウト・プロパティーを変更することによって増やすことができます。

MQCAI030 Route {0} is active

説明: MQIPT が PING 要求に正常に応答しました。

MQCAI031 Route {0} is not active

説明: MQIPT 経路が、指定時間内に PING 要求に応答しませんでした。

ユーザーの処置: この原因としては、以下のことが考えられます。

- MQIPT が稼働していない。
- MQIPT が自分のコマンド・ポートで listen していない。
- 要求がタイムアウトになった。このタイムアウト時間は、MQIPT に関する接続情報のタイムアウト・プロパティーを変更することによって増やすことができます。

MQCAI100 This script is used to start the Administration Client for {0}. Specifying a SOCKS proxy will allow the Administrator Client to talk to an MQIPT through a firewall.

説明: mqiptGui スクリプトに関するオンライン・ヘルプ情報。

MQCAI101 Format of command is:

説明: mqiptGui スクリプトに関するオンライン・ヘルプ情報。

MQCAI102 mqiptGui {socks_host{socks_port}}

説明: mqiptGui スクリプトに関するオンライン・ヘルプ情報。

MQCAI103 socks_host-host name of SOCKS proxy (optional)

説明: mqiptGui スクリプトに関するオンライン・ヘルプ情報。

MQCAI104 socks_port-SOCKS proxy port address (optional-default 1080)

説明: mqiptGui スクリプトに関するオンライン・ヘルプ情報。

MQCPE000 Could not locate message data when handling message {0}

説明: メッセージ番号 {0} がシステム・プロパティー・リストに入っていません。

ユーザーの処置: ″mqipt.properties″ ファイルが破壊されたため、指定されたメッセージ番号が見つかりません。以下の検査を行ってください。

- ″MQipt.jar″ ファイルがシステム CLASSPATH に入っているか
- ″mqipt.properties″ ファイルが ″MQipt.jar″ ファイルに入っているか
- メッセージ番号が ″mqipt.properties″ ファイルに入っているか

MQCPE001 Directory does not exist or is not a directory

説明: 初期化時に、必要なディレクトリーが見つかりませんでした。このメッセージは、MQIPT 構成ファイル mqipt.conf、または デフォルト・ディレクトリーの MQIPT コマンド行始動オプションのいずれかに指定されたディレクトリーを指しています。

ユーザーの処置: 正しいディレクトリーを指定して、もう一度コマンドを実行してみてください。

MQCPE004 Route startup failed on port {0}

説明: 指定された ListenerPort 番号の経路を開始できません。

ユーザーの処置: 経路始動時に入出力エラーが起こりました。この問題の詳細については、他の隣接エラー・メッセージやログ・レコードを調べてください。

MQCPE005 The configuration file {0} could not be found

説明: 指定されたディレクトリーに MQIPT 構成ファイル ″mqipt.conf″ が入っていません。

ユーザーの処置: 正しいディレクトリーを指定して、もう一度コマンドを実行してみてください。

MQCPE006 The number of routes has exceeded {0}. MQIPT will start but this configuration is unsupported.

説明: ユーザーの構成が、MQIPT の 1 つのインスタンスに関してサポートされる経路の最大数を超えました。操作は停止されませんが、結果としてシステムが不安定になったり、過負荷になったりすることがありま

す。示された経路の最大数を超える構成はサポートされません。

ユーザーの処置: インスタンス当たりの経路数が少ない MQIPT の別のインスタンスを開始することを考えてください。

MQCPE007 Route not restarted on listener port {0}

説明: REFRESH 操作で、指定された ListenerPort で作動する経路が新規構成で再始動されませんでした。

ユーザーの処置: この問題の詳細については、他の隣接エラー・メッセージを調べてください。

MQCPE008 Duplicate route defined for listener port {0}

説明: 同じ ListenerPort 値を持つ複数の経路が定義されています。

ユーザーの処置: 重複した経路を構成ファイルから除去し、もう一度コマンドを実行してみてください。

MQCPE009 Log directory {0} is not valid.

説明: テキストに示されているログ・パスが存在しないか、または所定の時点にアクセス可能でありません。

ユーザーの処置: ディレクトリーが存在し、それが MQIPT からアクセス可能であるか調べてください。

MQCPE010 Listener or command port number {0} is not valid

説明: コマンド・ポートまたはリスナー・ポート・パラメーターついて提供されたポート番号が無効です。

ユーザーの処置: 使用可能な有効なポート番号を指定してください。ネットワークでのポート番号の使用の指針については、ネットワーク管理者にお尋ねください。

MQCPE011 The trace level {0} is outside the valid range 0 - 5

説明: 指定されたトレース・オプションが要求されましたが、それが 0 ～ 5 の有効範囲に入っていません。

ユーザーの処置: 0 ～ 5 のトレース値を指定してください。

MQCPE012 The value {0} is not valid for the attribute {1}

説明: 無効なプロパティー値が指定されています。

ユーザーの処置: 各制御パラメーターに関する有効値の詳細については、本書の該当個所を参照してください。

MQCPE013 ListenerPort property was not found in route {0}

説明: MQIPT が、ListenerPort プロパティーを含んでいない経路を構成ファイルに検出しました。ListenerPort プロパティーは、各経路に関する基本的な固有 ID であるため、必須です。

ユーザーの処置: 所定の各経路に対して有効な ListenerPort を指定してください。

MQCPE014 ListenerPort property value {0} is not valid

説明: 経路の ListenerPort プロパティーに対して無効なポート・アドレスが指定されています。

ユーザーの処置: ポート・アドレスは 0 ～ 65535 の範囲でなければなりません。構成ファイル内の各 ListenerPort を調べてください。

MQCPE015 No text was found for message number {0}

説明: 内部エラーが検出されましたが、そのエラーについての説明は用意されていません。

ユーザーの処置: ″mqipt.properties″ ファイルが破壊されたため、指定されたメッセージ番号が見つかりません。以下の検査を行ってください。

- 該当する新規メッセージが README ファイルに入っているか
- ″MQipt.jar″ ファイルがシステム CLASSPATH に入っているか
- ″mqipt.properties″ ファイルが ″MQipt.jar″ ファイルに入っているか
- メッセージ番号が ″mqipt.properties″ ファイルに入っているか

MQCPE016 The maximum number of connection threads is {0} but this is less than the minimum number of connection threads, which is {1}

説明: ユーザーの構成で接続スレッドの最小数が指定されていますが、それが接続スレッドの最大数を超えた値になっています。

ユーザーの処置: 考えられる原因は、単一経路でのエラー、プロパティー・プロパティーと経路プロパティーの間の矛盾、または経路プロパティーによるシステム・デフォルト値のオーバーライドなどです。有効な値と適用

可能なデフォルト値の詳細については、本書の該当する章を参照してください。

MQCPE017 The exception {0} was thrown, causing MQIPT to shut down

説明: MQIPT が異常終了してシャットダウンしました。この状態の発生原因としては、システムの環境条件または制約 (たとえば、メモリー・オーバーフロー) が考えられます。

ユーザーの処置: この状態が継続する場合は、IBM 技術支援に連絡してください。

MQCPE018 The ListenerPort property is blank - the route will not start

説明: 経路で ListenerPort 番号が欠落しています。

ユーザーの処置: 構成ファイルを編集し、有効な ListenerPort を追加してください。

MQCPE019 The stanza {0} was not found before the following: {1}

説明: 構成ファイルでシーケンス・エラーが起こりました。

ユーザーの処置: 構成ファイルを編集し、すべての [route] エントリーが [global] エントリーの後に来ていることを確認してください。

MQCPE020 The new value for MaxConnectionThreads is {0}. This must be greater than the current value {1}

説明: 経路を開始した後は、MaxConnectionThread プロパティーを増やすことだけができます。

ユーザーの処置: 構成ファイルを編集し、MaxConnectionThread プロパティーを変更してください。

MQCPE021 The property Destination was not supplied for route {0}

説明: Destination プロパティーは経路内に必須ですが、指定された経路で欠落しています。

ユーザーの処置: 構成ファイルを編集し、所定の経路について Destination プロパティーを追加してください。

MQCPE022 The CommandPort value {0} is outside the valid range 1 - 65535.

説明: CommandPort プロパティーが 1 ～ 65535 の範囲外です。

ユーザーの処置: 構成ファイルを編集し、CommandPort プロパティーを有効なポート・アドレスに変更してください。

MQCPE023 Request for shutdown from Administration Client {0} is ignored because it is disabled.

説明: リモート・シャットダウンが構成ファイルで使用可能になっていないため、MQIPT をリモート側でシャットダウンしようとして失敗しました。

ユーザーの処置: MQIPT のリモート・シャットダウンを使用可能にするには、構成ファイルを編集し、RemoteShutDown プロパティーを true に設定します。

MQCPE024 The command received by the MQIPT controller has not been recognized.

説明: MQIPT が、認識できないコマンドをコマンド・ポートから受け取りました。

ユーザーの処置: ″mqipt.log″ ファイルにそのコマンドのアイデンティティーが含まれているか調べてください。

MQCPE025 Failed to connect to server on host {0}, port {1}.

説明: 行モード (非 GUI) Administration Client が MQIPT との通信に失敗しました。

ユーザーの処置: CommandPort プロパティーが構成ファイルに {1} として指定されており、MQIPT が {0} で稼働していることを確認してください。

MQCPE026 No reply received from server on host {0}, port {1}.

説明: 行モード (非 GUI) Administration Client が MQIPT との通信に失敗しましたが、まだ応答を受け取っていません。

ユーザーの処置: これは、要求がタイムアウトになったか、または MQIPT に問題があることを示しています。

MQCPE027 Reply from MQIPT not recognized.

説明: 行モード (非 GUI) Administration Client が、認識できない応答を MQIPT から受け取りました。

ユーザーの処置: mqiptAdmin スクリプトが MQIPT と同じバージョンの ″MQipt.jar″ ファイルを使用しているか調べてください。

MQCPE028 Invalid stanza detected: {0}

説明: 示されている未認識のスタンザが構成ファイルで見つかりました。

ユーザーの処置: 構成ファイルでは、[global] スタンザと [route] スタンザのみが有効です。

MQCPE029 Was not able to flush log output.

説明: 通信バッファーがフラッシュされたため、一部のメッセージがログに書き込まれなかった可能性があります。

ユーザーの処置: MQIPT ホーム・ディレクトリーのディスクがいっぱいになっていないか、MQIPT がまだログ・サブディレクトリーへのアクセス権を持っているか調べてください。

MQCPE030 {0} not found in CLASSPATH.

説明: 指定された JAR ファイルがシステム環境 CLASSPATH 変数に入っていません。

ユーザーの処置: 指定されたファイルをシステム CLASSPATH に追加してください。

MQCPE031 {0} class not found.

説明: このメッセージは、MQIPT のバージョン番号を表示するときに生成されます。指定されたクラスが MQIPT JAR ファイルに入っていないか、またはシステム環境 CLASSPATH 変数が破壊されています。

ユーザーの処置: 指定されたクラス・ファイルが ″MQipt.jar″ ファイルに入っているか、また ″MQipt.jar″ ファイルがシステム CLASSPATH に入っているか調べてください。

MQCPE033 Failed to send configuration file to Administration Client at {0}

説明: 構成ファイルを Administration Client に送信しているときにエラーが起こりました。

ユーザーの処置: 構成ファイルが MQIPT ホーム・ディレクトリーに入っているか、また別のプロセスで共用されていないか調べてください。

MQCPE034 Administration Client at {0} did not supply the correct password.

説明: 構成ファイルの AccessPW プロパティーが、Administration Client によって提供されたものと一致しません。

ユーザーの処置: 構成ファイルの AccessPW プロパティーを変更するか、または Administration Client に保管されているパスワードを変更してください。

MQCPE035 Failed to start command listener on port {0}

説明: 指定されたポート・アドレスのコマンド・リスナーを開始するときに入出力エラーが起こりました。

ユーザーの処置: 構成ファイルの CommandPort プロパティーで使用されたポート・アドレスを調べてください。

MQCPE038 MQIPT has not started as expected

説明: このメッセージは、MQIPT をシステム・サービスとして開始する mqipt fork プロセスによって生成されます。

ユーザーの処置: 詳細については、エラー・ログを調べてください。 MQIPT が稼働しているかどうかを調べる前に、IPTFork で使用するスリープ時間を増やしてみることができます。 mqiptFork スクリプトを編集し、IPTFork に渡したパラメーター値を大きくしてください。

MQCPE039 I/O error occurred running mqipt script

説明: fork プロセスから MQIPT を立ち上げるときにエラーが起こりました。

ユーザーの処置: システム PATH 環境変数に JDK のロケーションが入っている、また mqipt スクリプトが実行権限を持っているか調べてください。

MQCPE040 Interruption occurred running mqipt script

説明: fork プロセスから MQIPT を立ち上げた後にエラーが起こりました。

ユーザーの処置: 詳細については、エラー・ログを調べてください。この状態が継続する場合は、IBM 技術支援に連絡してください。

MQCPE041 Unsupported level of Java - {0}

説明: MQIPT が、指定されたレベルの Java を使用して開始されました。

ユーザーの処置: 詳細については、本書の該当個所に示されている前提条件を調べてください。

MQCPE042 There is a conflict with the following properties on route {0}:

説明: 一部のプロパティーは、他のプロパティーと一緒に使用できません。このメッセージは、対立するプロパティーのリストの前に表示されます。

ユーザーの処置: 以下のエラー・メッセージを調べて、適切なアクションをとってください。

MQCPE043 ....{0} and {1}

説明: これらのプロパティーは、同じ経路上で同時に設定することはできません。

ユーザーの処置: 構成ファイルを編集し、所定の経路上の指定されたプロパティーの 1 つを使用不可にしてください。

MQCPE044 {0} is only valid on the {1} operating system

説明: MQIPT の一部のフィーチャーは特定のプラットフォームでのみ有効です。

ユーザーの処置: 構成ファイルを編集し、指定されたプロパティーを使用不可にしてください。

MQCPE045 ....HTTP proxy name is missing

説明: HTTP プロパティーが true に設定されている場合は、HTTPProxy プロパティーを設定する必要があります。

ユーザーの処置: 構成ファイルを編集し、所定の経路について HTTPProxy を定義してください。

MQCPE046 {0} is not allowed as Pagent has failed to initialize

説明: Pagent は、MQIPT の Quality of Service を提供するアプリケーションです。 MQIPT が始動時にこのアプリケーションの初期化に失敗し、所定の経路について QoS プロパティーが true に設定されました。

ユーザーの処置: 構成ファイルを編集し、所定の経路について QoS を使用不可にしてください。

MQCPE047 Pagent has failed to initialize

説明: Pagent は、MQIPT の Quality of Service を提供するアプリケーションです。 MQIPT は始動時にこのアプリケーションの初期化に失敗しました。

ユーザーの処置: Pagent を使用していなければ、このエラー・メッセージを無視することができますが、QoS プロパティーを false に設定する必要があります。

MQCPE048 Route startup failed on port {0}, exception was : {1}

説明: 指定された ListenerPort 番号の経路を開始できません。

ユーザーの処置: この問題の詳細については、他の隣接エラー・メッセージやログ・レコードを調べてください。

MQCPE049 Error starting or stopping the Java Security Manager {0}

説明: Java Security Manager を開始または停止しようとしているときに、例外が throw されました。

ユーザーの処置: Java Security Manager はすでに使用可能になっていますが、ランタイム許可がまだ使用可能になっていません。 setSecurityManager の RuntimePermission をローカル・ポリシー・ファイルに追加してください。変更結果を有効にするには、MQIPT を再始動する必要があります。

MQCPE050 Security exception on port {0} from the Administration Client

説明: Administration Client からの接続を受け入れているときに、セキュリティー例外が throw されました。

ユーザーの処置: Java Security Manager はすでに使用可能になっていますが、エラー・メッセージに示されているホストについては許可が付与されていません。このホストを MQIPT に接続できるようにするには、CommandPort のポート・アドレスでの接続を受け入れ / 解決するための SocketPermission を追加してください。変更結果を有効にするには、Java Security Manager を再始動する必要があります。

MQCPE051 Security exception accepting a connection on route {0}

説明: 指定された経路への接続を受け入れているときに、セキュリティー例外が throw されました。

ユーザーの処置: Java Security Manager はすでに使用可能になっていますが、エラー・メッセージに示されて

いるホストについては許可が付与されていません。ホストをこの経路に接続できるようにするには、ListenerPort への接続を受け入れ / 解決するための SocketPermission を追加してください。変更結果を有効にするには、Java Security Manager を再始動する必要があります。

MQCPE052 Connection request on route {0} failed : {1}

説明: このメッセージは、接続要求のセキュリティー例外を記録する接続ログに出されます。

ユーザーの処置: Java Security Manager はすでに使用可能になっていますが、エラー・メッセージに示されているホストについては許可が付与されていません。ホストをこの経路に接続できるようにするには、ListenerPort への接続を受け入れ / 解決するための SocketPermission を追加してください。変更結果を有効にするには、Java Security Manager を再始動する必要があります。

MQCPE053 Security exception making a connection to {0}({1})

説明: 指定された経路への接続を作成しているときに、セキュリティー例外が throw されました。

ユーザーの処置: Java Security Manager はすでに使用可能になっていますが、エラー・メッセージに示されているホストについては許可が付与されていません。ホストをこの経路に接続できるようにするには、ListenerPort への接続を受け入れ / 解決するための SocketPermission を追加してください。変更結果を有効にするには、Java Security Manager を再始動する必要があります。

MQCPE054 Connection request to {0}({1}) failed : {2}

説明: このメッセージは、ターゲット・ホストに対する接続要求のセキュリティー例外を記録する接続ログに出されます。

ユーザーの処置: Java Security Manager はすでに使用可能になっていますが、エラー・メッセージに示されているホストについては許可が付与されていません。ホストをこの経路に接続できるようにするには、ListenerPort への接続を受け入れ / 解決するための SocketPermission を追加してください。変更結果を有効にするには、Java Security Manager を再始動する必要があります。

MQCPE055 ....Socks proxy name is missing

説明: SocksClient プロパティーが true に設定されている場合は、SocksProxy プロパティーを設定する必要があります。

ユーザーの処置: 構成ファイルを編集し、所定の経路について SocksProxy を定義してください。

MQCPE056 Conflict with route properties

説明: 一部のプロパティーは、他のプロパティーと一緒に使用できません。

ユーザーの処置: このエラーの詳細についてコンソール・メッセージを調べ、適切なアクションをとってください。

MQCPE057 SSL protocol ({0}) was not recognized

説明: この経路が SSL プロキシー・モードになり、初期データ・フローが認識されません。

ユーザーの処置: この経路に対して SSL 接続のみが行われていることを確認してください。

MQCPE058 CONNECT request to {2}({3}) through {0}({1}) failed

説明: HTTP サーバーへの SSL トンネルを作成するために、HTTP CONNECT 要求が HTTP プロキシーに送信されました。HTTP プロキシーはこの要求に対する ″200 OK″ 応答を戻しませんでした。

ユーザーの処置: これはさまざまな問題が原因となります。経路上のトレースを使用可能にして接続を再試行してください。トレース・ファイルが実際のエラーを表示します。

MQCPE059 There are no defined key ring files

説明: 1 つの鍵リング・ファイルも指定せずに SSL クライアントまたはサーバーが定義されました。

ユーザーの処置: クライアント・サイドに SSLClientKeyRing と SSLClientCAKeyRing のプロパティーを使用するかまたはサーバー・サイドに SSLServerKeyRing と SSLServerCAKeyRing を使用して鍵リング・ファイルを定義して、次にその経路を再始動してください。

MQCPE060 Runtime error setting SSL client connect timeout to {0} seconds

説明: タイムアウト値を設定しているクライアント・サイドに SSL ランタイム・エラーが発生しました。

ユーザーの処置: SSLClientConnectTimeout プロパティーに指定した値が有効であることをチェックしてください。所定の経路でトレースを実行するとエラー情報がさらに表示されます。

MQCPE061 There are no enabled cipher suites

説明: SSL クライアントまたはサーバーが始動されましたが、MQIPT が有効な暗号スイートを判別できません。

ユーザーの処置: 定義した鍵リング・ファイル (または複数の) に有効な証明書があることをチェックしてください。証明書を生成するために使用した専用鍵と公開鍵および使用した暗号化アルゴリズムは、サポートされる暗号スイートのリストに準拠する必要があります。このリストは MQIPT ブックで検出されます。

MQCPE062 Runtime error setting SSL cipher suite {0}

説明: クライアントまたはサーバー・サイドにサポートされない SSL 暗号スイートが定義されています。

ユーザーの処置: SSLClientCipherSuites または SSLServerCipherSuites に指定した値が有効であり、この接続でサポートされていることをチェックしてください。所定の経路でトレースを実行すると、使用可能にされた暗号スイートのリストが表示されます。 MQIPT ブックにはサポートされる暗号スイートのリストが収納されています。

MQCPE063 File {0} already exists - use the replace option

説明: mqiptPW スクリプト用に指定したファイル名パラメーターはすでに存在しています。

ユーザーの処置: 別のファイル名を選択するか、replace オプションを使ってください。

MQCPE064 Runtime error generating decryption keys :¥n {0}

説明: 鍵リング・ファイルをオープンするために使用されるパスワードの暗号化解除を行うための暗号鍵の生成時にエラーが発生しました。

ユーザーの処置: メッセージにリストされている実行時エラーを訂正して、コマンドを再度実行してください。

MQCPE065 LDAP server name is missing

説明: LDAP プロパティーが true に設定されている場合は、LDAPServer1 または LDAPServer2 プロパティーを設定する必要があります。

ユーザーの処置: 構成ファイルを編集し、所定の経路について LDAPServer* を定義してください。

MQCPE066 LDAP password is missing for LDAPServer{0}Password property

説明: パスワードなしで LDAP ユーザー ID が指定されました。

ユーザーの処置: 構成ファイルを編集し、所定の経路について LDAPServer*Password を定義してください。

MQCPE067 SSLClient or SSLServer missing for LDAP server

説明: LDAP プロパティーが true に設定されている場合は、SSLClient または SSLServer プロパティーを設定する必要があります。

ユーザーの処置: 構成ファイルを編集し、所定の経路について SSLClient または SSLServer を定義してください。

MQCPE068 Security exit name is missing

説明: SecurityExit プロパティーが true に設定されている場合は、SecurityExitName プロパティーを設定する必要があります。

ユーザーの処置: 構成ファイルを編集し、所定の経路について SecurityExitName を定義してください。

MQCPE069 Invalid port address {0} in security exit response

説明: SecurityExitResponse に指定したポート・アドレスは無効です。

ユーザーの処置: ポート・アドレスは 1024 ～ 65535 の範囲でなければなりません。

MQCPE070 Unknown reason code {0) in security exit response

説明: SecurityExitResponse に指定した理由コードはサポートされていません。

ユーザーの処置: サポートされている理由コードのリストについては、MQIPT ブックを参照してください。

MQCPE071 Error writing to {0}

説明: 指定したファイルの作成または更新時にエラーが発生しました。エラー・メッセージにも throw された例外が含まれています。

ユーザーの処置: 例外にリストされているエラーを訂正して、コマンドを再度実行してください。

MQCPE072 An unknown error occurred in security exit {0}

説明: 接続要求の確認時にユーザー定義のセキュリティー出口にエラーが発生しました。

ユーザーの処置: セキュリティー出口でトレースを使用可能にして、再度接続要求を試行してください。エラーはセキュリティー出口のトレース・ファイルに記録されます。

MQCPI001 {0} starting

説明: この MQIPT インスタンスが実行を開始しています。この後も、引き続き初期化メッセージが出されます。

MQCPI002 {0} shutting down

説明: MQIPT がシャットダウンしようとしています。このシャットダウンは、STOP コマンドが出された結果起こることもあれば、構成エラーのために正常な始動や REFRESH アクションが行われない場合に自動的に起こることもあります。

MQCPI003 {0} shutdown complete

説明: シャットダウン・プロセスが完了しました。これで、すべての MQIPT プロセスが終了しました。

MQCPI004 Reading configuration information from {0}

説明: MQIPT 構成ファイル mqipt.conf が、このメッセージに示されているディレクトリーから読み込まれています。

MQCPI005 Listener port specified as not active - {0} -> {1}({2})

説明: このメッセージで参照されている経路が非アクティブとしてマークされています。この経路では、どの通信要求も受け入れられません。

MQCPI006 Route {0} is starting and will forward messages to :

説明: このメッセージに示されているリスナー・ポートで経路が開始されました。このメッセージの後には、この経路に関連するすべてのプロパティーをリストした他のメッセージが続きます。経路が接続を受け入れることができる状態になると、メッセージ MQCPI078 が出されます。

MQCPI007 Route {0} has been stopped

説明: 指定された ListenerPort で作動する経路がシャットダウンしようとしています。このアクションは、通常、REFRESH コマンドを MQIPT に出し、経路構成が変更されたときにとられます。

MQCPI008 Listening for control commands on port {0}

説明: この MQIPT インスタンスは、指定されたポートで制御コマンドを listen しています。

MQCPI009 Control command received: {0}

説明: このメッセージは、制御コマンドがコマンド・ポートで受け取られたことを示しています。該当する場合は、詳細情報がこのメッセージに組み込まれます。

MQCPI010 Stopping command port on {0}

説明: REFRESH 操作の場合、コマンド・ポートは新規構成では使用されなくなりました。指定されたポートでは、コマンドは受け取られなくなりました。

MQCPI011 The path {0} will be used to store the log files

説明: 現行構成では、ロギング出力はこのメッセージに示されているロケーションに送られます。

ユーザーの処置: 構成に修正を加えた場合や REFRESH 操作を要求した場合には、これが変更されることがあります。

MQCPI012 Changing the value of MinConnectionThreads has no effect after the route is started

説明: 経路の始動時に接続スレッドの最少値が割り当てられますが、この値は MQIPT を再始動するまで変更できません。

MQCPI013 Connection from {0} to host {1} closed

説明: このメッセージは、接続アクティビティーを記録する接続ログに出されます。

MQCPI014 Eyecatcher protocol ({0}) not recognized

説明: このメッセージは、接続アクティビティーを記録する接続ログに出されます。

MQCPI015 Client access has been disabled on this route

説明: このメッセージは、接続アクティビティーを記録する接続ログに出されます。

MQCPI016 Queue Manager access has been disabled on this route

説明: このメッセージは、接続アクティビティーを記録する接続ログに出されます。

MQCPI017 A queue manager on {0} was connected to host {1}

説明: このメッセージは、接続アクティビティーを記録する接続ログに出されます。

MQCPI018 A client on {0} was connected to host {1}

説明: このメッセージは、接続アクティビティーを記録する接続ログに出されます。

MQCPI019 {0} routes have been created - this exceeds the maximum number of supported routes, which is {1}

説明: サポートされている経路の最大数を超えました。

ユーザーの処置: MQIPT は引き続き作動しますが、2 番目の MQIPT インスタンスを作成し、両者間で経路を分割することをお勧めします。

MQCPI020 The configuration file has been sent to the Administration Client.

説明: Administration Client からの要求の結果、構成ファイルが送信されました。

MQCPI021 Password checking has been enabled on the command port.

説明: このメッセージは、コマンド・ポートにアクセスするにはパスワードが必要であることを示しています。

MQCPI022 Password checking has been disabled on the command port.

説明: このメッセージは、コマンド・ポートにアクセスするのにパスワードが必要でないことを示しています。

MQCPI024 ....using HTTP proxy {0}({1})

説明: このメッセージは、この経路の発信接続がこの HTTP プロキシーを使用して行われることを示しています。

MQCPI025 The refresh requested by Administration Client {0} has finished.

説明: REFRESH コマンドを受け取った結果、MQIPT はその構成ファイルを再読み取りし、再始動しました。

MQCPI026 Administration Client {0} has requested shutdown.

説明: STOP コマンドを受け取った結果、MQIPT はシャットダウンします。

MQCPI027 {0} sent to {1} on port {2}

説明: このメッセージは、行モード (非 GUI) Administration Client から指定 MQIPT へ送信されたコマンドをシステム・コンソールに表示します。

MQCPI031 ......cipher suites {0}

説明: このメッセージは、この経路に使用されている暗号スイートをリストします。

MQCPI032 ......key ring file {0}

説明: このメッセージは、この経路の鍵リングのファイル名を示しています。

MQCPI033 ......client authentication set to {0}

説明: このメッセージは、SSL サーバーがこの経路のクライアント認証を要求しているかどうかを定義します。

MQCPI034 ....{0}({1})

説明: このメッセージは、この経路の宛先と宛先ポート・アドレスを示しています。

MQCPI035 ....using {0}

説明: このメッセージは、使用されているプロトコルを宛先に示します。それは、MQSeries プロトコル、HTTP トンネル操作、または HTTP チャンク操作のいずれかです。

MQCPI036 ....SSL Client side enabled with properties :

説明: このメッセージは、この経路が SSL を使用して宛先ホストにデータを送信することを示します。

MQCPI037 ....SSL Server side enabled with properties :

説明: このメッセージは、この経路が SSL を使用して送信元のホストからデータを受け取ることを示します。

MQCPI038 ......peer certificate uses {0}

説明: このメッセージは、対等証明書の認証を制御するために使用する Distinguish Name (公開鍵持ち主情報) をリストします。

MQCPI039 ....via Socks proxy {0}({1})

説明: このメッセージは、この経路の発信接続がこの Socks プロキシー (MQIPT をコマンド行から開始するときに定義される) を使用して行われることを示しています。

MQCPI040 Command port has been accessed by Administration Client {0}

説明: このメッセージは、システム・コンソールと MQIPT ログ・ファイル (ロギングが使用可能になっている場合) に書き込まれます。 MQIPT が Administration Client からの接続を受け取りました。

MQCPI041 ....will reply to Network Dispatcher advisor requests in {0} mode

説明: このメッセージは、経路の開始時にシステム・コンソールに書き出されます。 MQIPT が Network Dispatcher アドバイザーに応答するために使用するモードを示すために使用されます。有効なオプションは、「通常」と「置換」です。

MQCPI042 Maximum connections reached on route {0} - further requests will be blocked

説明: このメッセージは、所定の経路に関する接続の最大数に達したときにシステム・コンソールに書き出されます。それ以降の要求は、接続が解放されるか、または MaxConnectionThreads 値を増やすまでブロックされます。

MQCPI043 Connections on route {0} now unblocked

説明: このメッセージは、所定の経路が接続要求についてブロックを解かれたときにシステム・コンソールに書き出されます。

MQCPI044 MQIPT has been launched from system startup

説明: MQIPT がシステム・サービスとして開始されました。

MQCPI045 Launching MQIPT from system startup

説明: MQIPT がシステム・サービスとして開始されるところです。

MQCPI046 Sleeping for {0} seconds while MQIPT is launched from system startup

説明: fork プロセスは、MQIPT がシステム・サービスとして正常に開始されたかどうか確認する前に、この時間だけスリープします。

MQCPI047 .....CA keyring file {0}

説明: このメッセージは、この経路の CA 鍵リングのファイル名を示しています。

MQCPI048 The ping by Administration Client {0} has finished

説明: IPTController から Administration Client への応答メッセージ。

MQCPI049 ....QoS priority to dest = {0}, to caller = {1}

説明: このメッセージは、この経路における両方向のトラフィックの優先順位を示しています。

MQCPI050 Adding entry to inittab to automatically start MQIPT at system startup

説明: ユーザーが mqiptService スクリプトを実行して MQIPT をシステム・サービスとして開始しました。

MQCPI051 Removing entry from inittab that automatically starts MQIPT at system startup

説明: ユーザーが mqiptService スクリプトを実行して、システム・サービスとしての MQIPT の開始を中止しました。

MQCPI052 ....Socks server side enabled

説明: この経路は SOCKS サーバー (プロキシー) として機能し、SOCKS 化されたアプリケーションからの接続を受け入れます。

MQCPI053 Starting the Java Security Manager

説明: SecurityManager プロパティーが true に設定されているため、デフォルトの Java Security Manager が開始されます。

MQCPI054 Stopping the Java Security Manager

説明: SecurityManager プロパティーが false に設定されているため、デフォルトの Java Security Manager will be stopped as the SecurityManager が停止されます。

MQCPI055 Setting the java.security.policy to {0}

説明: デフォルトの Java Security Manager が開始されるところです。提供されたポリシー・ファイルを使用します。

MQCPI056 The Java Security Manager must be restarted to use a new policy file

説明: SecurityManagerPolicy プロパティーが変更されましたが、Java Security Manager を再始動するまで有効になりません。

ユーザーの処置: SecurityManager プロパティーを false に変更し、REFRESH コマンドを出して、Java Security Manager を停止してください。次に、SecurityManager を true に戻し、再度 REFRESH コマンドを出して、Java Security Manager を新規ポリシー・ファイルで開始してください。

MQCPI057 ....trace level {0} enabled

説明: このメッセージは、経路の開始時にシステム・コンソールに書き出されます。この経路で使用可能なトレースのレベルを表示するために使用されます。

MQCPI058 ....and a URI name of {0}

説明: このメッセージは、経路の開始時にシステム・コンソールに書き出されます。この経路の Uniform Resource Identifier 名を表示するために使用されます。

MQCPI059 ....servlet client enabled

説明: このメッセージは、経路の開始時にシステム・コンソールに書き出されます。この経路は MQIPT サーブレットに接続します。

MQCPI060 Installing files to automatically start MQIPT at system startup

説明: ユーザーが mqiptService スクリプトを実行して MQIPT をシステム・サービスとして開始しました。

MQCPI061 Removing files that automatically starts MQIPT at system startup

説明: ユーザーが mqiptService スクリプトを実行して、システム・サービスとしての MQIPT の開始を中止しました。

MQCPI064 ....no SSL authentication on this route

説明: このメッセージは、経路を開始したときにシステム・コンソールに書き出され、また、匿名暗号スイートが指定されているため、この経路に対して SSL 認証が使用されていないことを示します。

MQCPI065 ....in SSL proxy mode

説明: このメッセージは、経路を開始したときにシステム・コンソールに書き出され、また、この経路が SSL プロキシー・モードで作動していることを示します。

MQCPI066 ....and HTTP server at {0}{1}

説明: このメッセージは、この経路の発信接続がこの HTTP サーバーを使用して行われることを示しています。

MQCPI067 Setting up links to TQoS runtime libraries

説明: ユーザーは、実際の TQoS ランタイム・ライブラリーにリンクさせるため mqiptQoS スクリプトを実行しました。

MQCPI068 Removing links to TQoS runtime libraries

説明: ユーザーは、実際の TQoS ランタイム・ライブラリーへのリンクを除去するため mqiptQoS スクリプトを実行しました。

MQCPI069 ....binding to local address {0}

説明: このメッセージは、それぞれの接続がバインドされるローカル IP アドレスを示します。これはマルチホーム・システムでのみ使用する必要があります。

MQCPI070 ....using local port address range {0}-{10}

説明: このメッセージは、接続に使用されるローカル・ポート・アドレスを示します。これによって、ファイアウォール・アドミニストレーターは MQIPT からの接続を制限することができます。

MQCPI071 site certificate uses {0}

説明: このメッセージは、サイト証明書の選択を制御するために使用する Distinguish Name (公開鍵持ち主情報) をリストします。

MQCPI072 .....and certificate label {0}

説明: このメッセージは、サイト証明書の選択を制御するために使用するラベル名をリストします。

MQCPI073 Updated file {0}

説明: mqiptPW スクリプト用に指定したファイル名は更新されました。

MQCPI074 Created file {0}

説明: mqiptPW スクリプト用に指定したファイル名は作成されました。

MQCPI075 ....LDAP main server at {0}({1})

説明: このメッセージは、CRL サポートに使用したメイン LDAP サーバーの名前をリストします。

MQCPI076 ....LDAP backup server at {0}({1})

説明: このメッセージは、CRL サポートに使用したバックアップ LDAP サーバーの名前をリストします。

MQCPI077 ....LDAP errors will be ignored

説明: このメッセージは、LDAP から受信したエラーがあってもそのエラーは無視されることを意味します。

MQCPI078 Route {0} ready for connection requests

説明: 経路が接続要求を受け入れることができる状態になると、このメッセージが表示されます。

MQCPI079 ....using security exit {0}

説明: このメッセージは、経路の開始時にシステム・コンソールに書き出されます。これはセキュリティー出口の完全修飾名を示すために使用されます。

MQCPI080 .....and timeout of {0} seconds

説明: このメッセージは、経路の開始時にシステム・コンソールに書き出されます。これはセキュリティー出口のタイムアウト値を示すために使用されます。

MQCPI081 Start message for WebSphere MQ internet pass-thru

説明: サービスとしての WebSphere MQ internet pass-thru の開始メッセージ。

MQCPI082 Stop message for WebSphere MQ internet pass-thru

説明: サービスとしての WebSphere MQ internet pass-thru の停止メッセージ。

MQCPI083 ....refresh commands will not restart the route

説明: このメッセージは、リフレッシュ・コマンドが出されてもその経路は再始動されないことを示します。

MQCPI084 ....CRL cache expiry timeout is {0} hour(s)

説明: このコンソール・メッセージは、CRL (または ARL) が MQIPT キャッシュに残っている時間を表示します。

MQCPI085 ....CRLs will be saved in the key ring file(s)

説明: このコンソール・メッセージは、LDAP サーバーから取り出された CRL (または ARL) があると、それらは関連した CA 証明書に付加された鍵リング・フ

ファイルに保管されることを意味します。

MQCPI086 .....timeout of {0} second(s)

説明: このメッセージは、経路の開始時にシステム・コンソールに書き出されます。 LDAP サーバーへの接続のタイムアウト値を示すのに使用されます。

MQCPI087 .....userid is {0}

説明: このメッセージは、経路の開始時にシステム・コンソールに書き出されます。 LDAP サーバーに接続するためのユーザー ID 名を示すのに使用されます。

MQCPI100 This script is used to start {0}

説明: mqipt スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI101 Format of command is :

説明: mqipt スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI102 mqipt {dir_name}

説明: mqipt スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI103 dir_name - directory containing mqipt.conf

説明: mqipt スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI106 This script is used to display the current version number

説明: mqiptVersion スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI107 mqiptVersion {-v}

説明: mqiptVersion スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI108 where -v will also display the build timestamp

説明: mqiptVersion スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI109 This script is used to start {0}, from system startup, in another JVM and is only used in mqipt.ske. Use the mqipt script to start MQIPT from the command line.

説明: mqiptFork スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI110 This class is used to display a simple NLS message on the console

説明: IPTMessages クラスからのオンライン・ヘルプ・メッセージ。

MQCPI111 java com.ibm.mq.ipt.IPTMessages (message_id1) {message_id2} {message_id...}

説明: IPTMessages クラスからのオンライン・ヘルプ・メッセージ。

MQCPI112 where message_id matches a key in the file mqipt.properties

説明: IPTMessages クラスからのオンライン・ヘルプ・メッセージ。

MQCPI113 This script is used to manage MQIPT as a system service

説明: mqiptService スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI114 mqiptService (-install | -remove)

説明: mqiptService スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI115 -install will install files to start MQIPT automatically at system startup

説明: mqiptService スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI116 -remove will remove files that start MQIPT automatically at system startup

説明: mqiptService スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI117 This script is used to manage links to the TQoS runtime libraries

説明: mqiptService スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI118 mqiptQoS (-install | -remove)

説明: mqiptService スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI119 -install will setup links to the real TQoS runtime libraries

説明: mqiptService スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI120 -remove will remove links to the real TQoS runtime libraries

説明: mqiptService スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI121 Use this script to encrypt a password and store it in a file

説明: mqiptPW スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI122 mqiptPW password file_name { -replace }

説明: mqiptPW スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI123 password - password used to open a key ring file

説明: mqiptPW スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI124 file_name - encrypted password will be stored in this file

説明: mqiptPW スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI125 replace option must be used to update an existing file

説明: mqiptPW スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPI126 mqipt (-start | -stop)

説明: mqiptQoS スクリプトからのオンライン・ヘルプ・メッセージ。

MQCPW001 CRL expired for {0}

説明: LDAP サーバーまたは鍵リング・ファイルから CRL (または ARL) が取り出されると、このメッセージが表示されます。

ユーザーの処置: LDAP サーバーまたは鍵リング・ファイルにある指定した CRL を更新してください。

MQCPW002 Error updating key ring file {0} with CRL

説明: LDAPSaveCRLs プロパティーを使用可能にしていて、指定した鍵リング・ファイルを更新できないときに、このメッセージが表示されます。

ユーザーの処置: 指定したファイルが破壊されている可能性があります。以下の検査を行ってください。

1. 書き込みアクセス権限を MQIPT のために使用可能にする必要がある
2. このファイルが別のアプリケーションによってオープンされていない

MQCPW003 ....Expired CRLs will be ignored

説明: このコンソール・メッセージは、有効期限が切れた CRL (または ARL) があればそれが無視され、接続要求を受け入れることができることを意味します。

付録. 特記事項

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。 国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

本書に記載の製品、プログラム、またはサービスが日本においては提供されていない場合があります。日本で利用可能な製品、プログラム、またはサービスについては、日本アイ・ビー・エムの営業担当員にお尋ねください。

本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。 IBM 製品、プログラムまたはサービスに代えて、IBM の知的所有権を侵害することのない機能的に同等のプログラムまたは製品を使用することができます。ただし、IBM によって明示的に指定されたものを除き、他社の製品と組み合わせた場合の操作の評価と検証はお客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

本書に含まれる情報は、IBM の正式なテストを受けておらず、現存するままの状態で配布されます。この情報の利用またはこうした手法の導入は、お客様の責任であるとともに、これを評価しお客様の稼働環境への統合するお客様の能力に依存します。個々の項目は、特定の状況における正確性について IBM によって検討されていますが、全く同一または同様な結果が得られる保証はありません。お客様自身の環境にこれらの技法を適用しようとする場合は、お客様自身のリスクにおいて行っていただきます。

商標

以下は、IBM Corporation の商標です。

AIX	FFST	First Failure Support Technology
IBM	IBMLink	MQSeries
SupportPac	WebSphere	

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。

参照文献

本書は、インストール製品の一部として HTML で提供されます。この HTML は、doc¥<locale>¥html¥<filename>.zip ディレクトリーに自己解凍型の zip ファイルに収納されています。「Administration Client」を使用する前に、<locale>/html サブディレクトリーに入っているファイルを UNZIP する必要があります。本書は、以下の言語で作成されています。言語とその対応ファイル名については、以下の表を参照してください。

表 4. 言語とファイル名の要約

言語	ロケール	HTML ファイル名
中国語 (簡体字)	zn_CN	amqyzb01.zip
ドイツ語	de_DE	amqygb01.zip
日本語	ja_JP	amqyjb01.zip
韓国語	ko_KR	amqykb01.zip
ブラジル・ポルトガル語	pt_BR	amqybb01.zip
スペイン語	es_ES	amqysb01.zip
米国英語	en_US	amqyab01.zip

翻訳済みの PDF は、次の URL からダウンロードできます。

http://www.ibm.com/webspheremq/downloads

これは以下の言語で使用できます。

表 5. PDF 言語とファイル名

言語	ロケール	PDF ファイル名
中国語 (簡体字)	zn_CN	amqyzb01.pdf
ドイツ語	de_DE	amqygb01.pdf
日本語	ja_JP	amqyjb01.pdf
韓国語	ko_KR	amqykb01.pdf
ブラジル・ポルトガル語	pt_BR	amqybb01.pdf
スペイン語	es_ES	amqysb01.pdf
米国英語	en_US	amqyab01.pdf

以下の資料も有用です。

- *WebSphere MQ 相互通信*、SC88-9223
- *WebSphere MQ システム管理ガイド*、SC88-9239
- *WebSphere MQ クライアント*、GC88-9222
- *WebSphere MQ キュー・マネージャー・クラスター*、 SC88-9224

これらの資料は、WebSphere MQ チャネルとその属性の定義に関する情報、特に CONNAME の定義に関する情報を提供します。

WebSphere MQ 資料は、以下の URL から入手できます。

http://www.ibm.com/webspheremq/library

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

[カ行]

[サ行]

[タ行]

[ハ行]

[マ行]

［ヤ行］

A

C

D

F

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

W

X