

MQSeries® Adapter Kernel for Multiplatforms



問題判別ガイド

バージョン 1 リリース 1

MQSeries® Adapter Kernel for Multiplatforms



問題判別ガイド

バージョン 1 リリース 1

ご注意

本書の情報およびそれによってサポートされる製品を使用する前に、33ページの『特記事項』に記載する一般的情報をお読みください。

本書は、MQSeries Adapter Kernel for Multiplatforms (製品番号 5648-D75)、バージョン 1、リリース 1、モディフィケーション・レベル 1、および新版において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)

原 典： GC34-5897-01
MQSeries® Adapter Kernel for Multiplatforms
Problem Determination Guide
Version 1 Release 1

発 行： 日本アイ・ピー・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 2001.5

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

© Copyright International Business Machines Corporation 2000, 2001. All rights reserved.

Translation: © Copyright IBM Japan 2001

目次

図	v
表	vii
本書について	ix
本書の対象読者	ix
本書の構成	ix
関連情報	ix
本書の規則	x
変更の要約	x
第1章 トレース	1
トレースの概要	1
トレースの宛先	2
構成	2
トレース・レベル	3
トレースの構成	4
トレース・クライアントの定義	5
トレース・サーバーの定義	12
トレース・メッセージの読み取り	20
第2章 構成に関する一般的な問題	23
カーネル構成に関する一般問題	23
ファイル aqmsetup が見つからない	23
ファイル aqmconfig.xml が見つからない	23
メッセージに対する無応答	23
ターゲット・アダプターが見つからない	23
本体データの構文解析エラー	24
ネイティブ・アダプターからの例外	24
アダプター・デーモンのシャットダウン	24
受信したメッセージがアプリケーションに送信されない	24
着信キューからメッセージが除去されない	25
ソケット・ハンドラーに関するトレース・エラー	25
トレース・クライアントのインスタンス化の失敗	25
トレース・クライアントまたはトレース・サーバーのループ	25
MQSeries に関する問題	25
キューが見つからない	26
チャンネルのダウン	26
キュー項目数の超過	26
ログが使用可能なスペースを超過した	26
MQSeries JAR ファイルが見つからない	26
メッセージ・カタログをロードできない	26
キュー・マネージャーが使用できない	27

受信例外	27
ResourceException クラスが見つからない	27
MQSeries クラスが見つからない	28
MQSeries Integrator に関する問題	28
メッセージが理解されない	28
MQSI のメッセージ受信	28
JMS に関する問題	28
JNDI の初期設定が失敗した	28
JMS 通信モードが失敗した	29
QueueConnectionFactory または Queue オブジェクトの検出エラー	29
HP-UX 上のバインディングの問題	29
HP-UX 上のコード・ページ変換問題	29
JMS と LDAP を使用して AdapterException が発生	30
その他の問題	30
UNIX 上での表示エラー	30
Solaris 上のメモリー・ダンプ	30
AIX 上でのセグメント化違反	30
Java メモリーの問題: OutOfMemoryError	31
XML パーサーの問題	31
WebSphere の使用	31
特記事項	33
商標	35
索引	37



1.	アプリケーションのトレース構成値の設定	5
2.	デフォルトの TraceClient 定義の構造	6
3.	デフォルトの TraceClient 定義: メッセージ・カタログ	6
4.	デフォルトの TraceClient 定義: メッセージ・バッファー	7
5.	デフォルトの TraceClient 定義: ハンドラーの登録	8
6.	TraceClient 定義: 複数のハンドラーの登録	8
7.	トレース・クライアントのコンソール・ハンドラーの構成	9
8.	トレース・クライアントのファイル・ハンドラーの構成	10
9.	トレース・クライアントのソケット・ハンドラーと ENA ハンドラーの構成	11
10.	完全なデフォルト TraceClient 定義	12
11.	デフォルトの TraceServer 定義の構造	13
12.	デフォルトの TraceServer 定義: メッセージ・カタログ	14
13.	デフォルトの TraceServer 定義: メッセージ・バッファー	14
14.	デフォルトの TraceServer 定義: ハンドラーの登録	15
15.	トレース・サーバー中のマルチファイル・ハンドラーの構成	17
16.	トレース・サーバーのコンソール・ハンドラーの構成	17
17.	トレース・ワーカーのソケット・ハンドラー用の構成	19
18.	トレース・ワーカーの ENA ハンドラー用の構成	19
19.	完全なデフォルト TraceServer 定義	20
20.	2 つのトレース・メッセージ	21

一 表

1. 本書の規則	x
--------------------	---

本書について

本書では、MQSeries® Adapter Kernel に関する特定の問題を解決するためのツールについて説明します。本書では、問題の一般的な症状および考えられる解決方法を一組として記載し、また一般的でない問題を識別するためのトレースの使用法について説明しています。

本書の対象読者

本書は、MQSeries Adapter Kernel の開発者、管理者、およびユーザーを対象にしています。本書の読者は、MQSeries Adapter Kernel と MQSeries Adapter Builder ツールに精通していることが前提になっています。

本書の構成

本書は以下のような構成になっています。

- 1ページの『第1章 トレース』では、MQSeries Adapter Kernel のトレースの構成方法と使用法を説明します。
- 23ページの『第2章 構成に関する一般的な問題』では、構成に関する一般的な問題の一部を取り上げ、考えられる解決方法を提案します。

関連情報

本書で説明されているトピックに関する詳細は、以下の資料を参照してください。

- スタートアップ・ガイド

本書の規則

MQSeries Adapter Kernel の資料では、以下のような書体の規則およびキーの規則を使用しています。

表 1. 本書の規則

規則	意味
太字	コマンド名を示します。グラフィカル・ユーザー・インターフェース (GUI) に関しては、太字は、メニュー、メニュー項目、ラベル、およびボタンを示します。
モノスペース	コマンド・プロンプトで入力する必要のあるテキスト、およびファイル名、パス、ならびに関数、クラス、メソッドといったプログラム言語エレメントなどの、示されているとおりに使用しなければならない値を示します。モノスペースは画面テキストおよびコード例を示すためにも使用されます。
イタリック	指定する必要のある可変値を示します (例: <i>fileName</i> にファイルの名前を指定します)。イタリックは、強調表示および書名の表示にも使用されます。
%	root 権限を必要としないコマンドの UNIX コマンド・シェル・プロンプトを示します。
#	root 権限が必要なコマンドの UNIX コマンド・シェル・プロンプトを示します。
C:¥>	Windows [®] システムのコマンド・プロンプトを示します。
>	メニューの記述に使用している場合は、一連のメニューの選択を示します。たとえば、「 File (ファイル) 」->「 New (新規) 」をクリックします」とある場合は、「 File (ファイル) 」メニューから「 New (新規) 」コマンドをクリックする」という意味になります。
コマンド入力	コマンドを「入力」する、または「送出」すると指示された場合は、コマンドを入力してから Enter を押します。たとえば、「 ls コマンドを入力してください」という指示は、コマンド・プロンプトに ls と入力してから Enter を押すという意味になります。
[]	構文記述内のオプション項目を囲みます。
{ }	構文記述内の、項目を選択する必要があるリストを囲みます。
	構文記述の中で中括弧 ({ }) で囲まれた選択項目リストにある項目を区切るために使用されます。
...	構文記述内の省略記号は、前の項目を 1 回以上繰り返すことができることを示します。例にある省略記号は、簡潔にするために例から情報が省略されていることを示します。

変更の要約

第 2 版 (現行版) には、第 1 版からの以下の変更が加えられています。

- 23ページの『第2章 構成に関する一般的な問題』に MQSeries に関する問題およびソリューションがいくつか追加されています。25ページの『MQSeries に関する問題』を参照してください。

第1章 トレース

トレースは、特定のプロセスの実行状況に関する詳細情報を収集するための一般的なプロセスです。トレース情報を使用して、アプリケーションをデバッグしたり、製品に関する問題を診断したりできます。MQSeries Adapter Kernel の場合は、トレースを使用して、アプリケーション固有のアダプターやカーネル自体に関する情報を収集できます。トレースは、アプリケーション開発者とシステム管理者の両者にとって役立ち、製品サポートに相談する際にも不可欠です。

MQSeries Adapter Offering などの最新のソフトウェア製品には、普通はトレース情報を収集する機能が標準装備されています。トレースを使用可能化して収集するには、実行時構成パラメーターを設定するだけで済みます。この種のパラメーターはトレース情報を取り込んで特定の場所に書き込むよう製品に指示するので、その情報を収集して分析できます。

トレース情報は、トレース・メッセージの集合の形式で収集されます。トレース・メッセージの数はトレース・レベルによって決まります。トレース・レベルは、取り込む情報の種類を指定します。収集されたメッセージは、1 つまたは複数の宛先に書き込まれます。宛先の例としては、コンソール・ウィンドウや名前指定されたファイルなどがあります。

トレースの構成に関する情報は、4ページの『トレースの構成』を参照してください。

トレース・メッセージの解釈に関する情報は、20ページの『トレース・メッセージの読み取り』を参照してください。

トレースの概要

トレース・メッセージには、カーネル内の特定のポイントにおけるメッセージ処理の状態が含まれます。トレース・メッセージを使用して、カーネルやご使用のアダプターに関する問題の診断に役立てることができます。

MQSeries Adapter Offering では、以下のコンポーネントを使用してトレースが行われます。

- トレース・クライアント
- トレース・サーバー。これは、ソケットかネイティブ・アダプターを使用してトレース・メッセージを検索し、そのトレース・メッセージをトレース・ワーカーに渡すデーモンです。
- トレース・ワーカー。これは特別なトレース・クライアントです。トレース・ワーカーには、トレース・サーバーで使用される追加のメソッドが備えられており、標準的なトレース・クライアントとは違うデフォルトの下で操作が行われます。

- 出力ハンドラー。これはトレース・メッセージを書き込む方法と場所を決めます。個々のトレース・クライアントにハンドラーの集合が割り当てられます。

トレース・クライアントのインスタンスは処理スレッド当たり複数あってもかまいません。しかしカーネル中では、トレース・クライアントのインスタンスは、ソースまたは宛先の論理 ID とスレッドの組み合わせ当たり 1 つだけ使用されます。スレッドを使用するカーネル・プロセスはすべて、そのスレッドに関連するトレース・クライアントに書き込みます。個々のトレース・メッセージにはスレッドの ID が含まれているので、特定のスレッドからすべてのトレース・メッセージを追跡できます。終端間でメッセージ・フローをトレースするには、カーネルの両方の終端からのトレース・メッセージを手操作で調べて関連させなければなりません。

トレースの宛先

個々のトレース・クライアントのトレース・メッセージを、以下のうち 1 つまたは複数に書き込むことができます。

- コンソール (デフォルト)
- 1 つまたは複数のファイル
- トレース・サーバー (ソケット接続を使用)
- トレース・サーバー (メッセージをキューに書き込む)

上記の宛先はいずれも、トレース・クライアントと関連付けられたハンドラーによって管理されます。トレース・クライアントでは、ウィンドウに書き込む場合はコンソールが、ファイルに書き込む場合はファイルが、ソケットを介してリモート・サーバーに書き込む場合はソケット・ハンドラーが使用されます。リモート・サーバーへの送達のために、メッセージをキューに書き込むトレース・クライアントは、Adapter Kernel が提供する固有のアダプター用にハンドラーを使用します。このアダプターを、ネイティブ・アダプター (ENA) ハンドラーといいます。ソケットや ENA ハンドラーを使用する場合にも、トレース・サーバーを使用する必要があります。

構成

トレースを含む、MQSeries Adapter Kernel アプリケーションのすべての局面に関する構成は、XML 構成ファイルに含まれています。カーネルのインストール時には、samples というディレクトリが設けられます。この中には、デフォルト構成ファイル aqmconfig.xml と、構成スクリプト aqmsetup が入れられます。新しいアプリケーションを使用して、これらのファイルをコピーして必要に合うよう変更を加えたり、新しいファイルを作成したりすることができます。Adapter Kernel では、環境変数 AQMSETUPFILE を検索して、使用するセットアップ・スクリプトが判断されます。このセットアップ・スクリプトには、該当する構成ファイルを含むディレクトリが指示されています。

このファイルには、個々のソースの論理 ID、宛先の論理 ID、およびアダプター・デーモンに関する構成情報が含まれています。それぞれのアプリケーションが、始動する際にアプリケーション ID を受け取ります。このアプリケーション ID は、構成ファイルの中で項目のラベルとしての役目を果たします。個々のアプリケーションでは、ファイル中の自分のアプリケーション ID が検索されて、トレースを含む自分の構成が調べられます。

個々のアプリケーションには、以下の 3 つのトレース関連の設定があります。

- トレースが使用可能か使用不可のどちらか
- トレース・レベル (『トレース・レベル』を参照)
- トレース・クライアントのアプリケーション ID

これらの値は、XML タグの対の間に格納されています。(アプリケーションの構成についての一般的な情報は、*スタートアップ・ガイド* を参照してください。)

アダプターだけでなく、トレース・クライアントとトレース・サーバーのアプリケーション ID も構成ファイル中にあります。たとえば、トレース・クライアントのアプリケーション ID の下の構成には、使用するハンドラーが指定されます。トレース・クライアントとトレース・サーバーに関するこの情報が項目として要約されているので、多数のソース論理 ID、宛先論理 ID、およびアダプター・デーモンで同じトレース構成を利用できる一方で、個々のトレースをオン / オフに切り換えられる柔軟性も維持できます。トレース・サーバーは、ソケット・ハンドラーや固有アダプター・ハンドラーで使用されますが、同様にアプリケーション ID の下で構成できます。

トレース・レベル

MQSeries Adapter Offering では、以下のトレース・レベルが定義されています。

0	トレース・メッセージなし
1	通知のみ
2	警告のみ
3	通知と警告
4	エラーのみ
7	通知、警告、およびエラー
128	メソッド入り口
256	メソッド出口
384	メソッドの入り口と出口
512	例外
903	通知、警告、エラー、例外、入り口、および出口
-1	書き込めるメッセージすべて

定義されているレベルの中には、他のレベルを組み合わせたものがあります。たとえば、メソッド入り口と出口のイベントを両方とも収集するトレース・レベル (384) は、メソッド入り口 (128) とメソッド出口 (256) のトレース・レベルを合わせたものです。組み合わせを設定するには、対応するトレース・レベルを合計します。たとえば、警告 (2) とエラー (4) をトレースするには、トレース・レベル 6 を指定します。

トレースの構成

アプリケーションのトレースを構成するには、XML 構成ファイル中のそのアプリケーションの項目にトレース情報を追加します。トレース関連の情報には、以下のものが含まれます。

- トレースが使用可能になっているかどうか
- トレースが使用可能な場合に使用するトレース・レベル
- オプションで、トレース・クライアントのアプリケーション ID。この ID を指定しないと、デフォルトのトレース・クライアント ID の `TraceClient` が使用されます。

注: XML は大文字小文字の区別があるので、タグ中で XML のエレメントや属性の名前を使用する際には、大文字の箇所も含めて正しい文字を使用しなければなりません。エレメントや属性の名前によっては、大文字を使用しているかどうかの違いしかないものもあります。

アプリケーションのトレース値を設定するには、アプリケーションの定義エレメント (`ePICAApplication` エレメント) 中で以下の XML エレメントを使用してください。

- `epictrace`: トレースを使用可能にしたり使用不可にしたりする。 `true` または `false` に設定してください。
- `epictracelevel`: トレース・レベルを設定する。事前定義済みのトレース・レベルについては、3 ページの『トレース・レベル』を参照してください。
- `epictraceclientid`: 使用するトレース・クライアント構成を識別する。デフォルト値は `TraceClient` です。

以下の例が示しているのは、アプリケーション ID (`epicappid`) `TEST1` を持つアプリケーション用のトレース値の設定の仕方です。トレースは使用可能になっており、トレース・レベルは 6 で、トレース・クライアント ID はデフォルトの `TraceClient` です。


```
<Epic o="ePIC">
  <ePICApplications o="ePICApplications">
    <ePICApplication epicappid="TEST1">
      <epictrace>true</epictrace>
      <epictracelevel>6</epictracelevel>
      <epictraceclientid>TraceClient</epictraceclientid>
      ...
    </ePICApplication>
  </ePICApplications>
</Epic>
```

図 1. アプリケーションのトレース構成値の設定

独自のトレース・クライアントを定義してから、epictraceclientid エlementによって参照されるようにすることもできます。

トレース・クライアントの定義

デフォルトの aqmconfig.xml ファイルには、デフォルトのトレース・クライアント ID である TraceClient の項目が含まれています。この項目には、TraceClient のデフォルト値が反映されています。この項目をテンプレートとして使用して新しいトレース・クライアントを構成したり、変更を加えて TraceClient の能力を高めたりできます。

TraceClient の定義に変更を加えると、トレース・クライアント ID が指定されていないアプリケーションで、TraceClient の新しい値が選出されます。

TraceClient のデフォルト値はアダプター・カーネルに組み込まれています。構成ファイル中の定義を使用してデフォルトの動作をオーバーライドすることはできますが、構成ファイルから TraceClient 定義を削除してデフォルトを除去することはできません。

TraceClient 定義を削除すると、トレース・クライアント ID が指定されていないアプリケーションでは、組み込みの値が使用されます。

追加のトレース・クライアントを定義することができますが、その際に TraceClient 定義をモデルとして利用できます。6ページの図2は、トレース・クライアントの構成項目の構造を示しています。構成ファイル中では、ePICApplication Elementを使用して、トレース・クライアントが追加アプリケーションとして定義されています。トレース・クライアントは、名前によって区別され、epicappid 属性中に指定されています。アプリケーションの ePICTraceExtensions Elementの中で構成の詳細が指定されています。このElementの1つしかない属性には固定値 cn="epicappextensions" があります。

```

<ePICAApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    ...
  </ePICTraceExtensions>
</ePICAApplication>

```

図 2. デフォルトの *TraceClient* 定義の構造

トレース・クライアントのメッセージ・ファイル

内部的には、トレース・メッセージは数字コード、つまりメッセージ *ID* として扱われます。カーネルによりトレース・メッセージが書き込まれると、メッセージを生成する内部機能によりメッセージ・カタログ・ファイルが使用され、メッセージ *ID* に対応するストリングが検索されます。この作業によりメッセージ・カタログを複数の言語に翻訳できるので、アプリケーションで希望する言語を選択できます。21ページの図20は、メッセージ・コードと対応するメッセージ・ストリングを図示しています。

`epictracemessagefile` エレメントには、メッセージ・カタログがメッセージ生成機能に明示的に渡されない場合に使用するメッセージ・カタログの名前を指定します。

TraceClient の定義には、デフォルトのカタログ

`com.ibm.epic.trace.client.TraceMessage` の名前が明示的に指定されています。他のトレース・クライアントでは、デフォルトのファイルを受け入れてよい場合は `epictracemessagefile` エレメントを省略できます。図3 が示しているのは、*TraceClient* 定義のデフォルトの例です。

```

<ePICAApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    <epictracemessagefile>
      com.ibm.epic.trace.client.TraceMessage
    </epictracemessagefile>
    ...
  </ePICTraceExtensions>
</ePICAApplication>

```

図 3. デフォルトの *TraceClient* 定義: メッセージ・カタログ

トレース・クライアントのメッセージ・バッファー

メッセージを書き込む際には、同期的にも非同期的にも書き込めます。

`epictracesyncoperation` エレメントによってこの動作を制御できます。このエレメントはトレース・ハンドラーに渡されます。同期動作の場合は `true` の値を使用し、非同期動作の場合は `false` (デフォルト) を使用してください。7ページの図4 で示すように、*TraceClient* の定義では、明示的に非同期動作を要求しています。他のトレース・クライ

アントでは、非同期書き込みが許容される場合、`epictracesyncoperation` エレメントを省略できます。

```
<ePICApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracesyncoperation>false</epictracesyncoperation>
    ...
  </ePICTraceExtensions>
</ePICApplication>
```

図4. デフォルトの *TraceClient* 定義: メッセージ・バッファ

トレース・クライアントのメッセージ・ハンドラー

トレース・クライアント用に登録されたメッセージ・ハンドラーにより、収集されたトレース・メッセージの送信場所が指定されます。以下の 4 つの宛先を指定できます。

- コンソール
- ファイル
- トレース・サーバー (ソケット接続を使用)
- トレース・サーバー (固有アダプターを使用)

上記の出力宛先はそれぞれ別のハンドラーによって管理されます。トレース・クライアントの場合、以下のハンドラーが登録可能です。

- `com.ibm.logging.ConsoleHandler`
- `com.ibm.logging.FileHandler` (1 つのファイル)
- `com.ibm.logging.MultiFileHandler` (バックアップ循環ファイル)
- `com.ibm.logging.SocketHandler`
- `com.ibm.epic.trace.client.ENAHandler`

ハンドラーを登録するには、`epictracehandler` エレメントを使用します。`TraceClient` 定義のデフォルトでは、トレース・メッセージはコンソールに送信されます。8ページの図5 に示すように、他のトレース・クライアントでは、1 つまたは複数のハンドラーを指定しなければなりません。ハンドラーを登録したら、構成も行わなければなりません。詳しくは、8ページの『ハンドラーの構成』を参照してください。

```

<ePICApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracehandler>com.ibm.logging.ConsoleHandler</epictracehandler>
    ...
  </ePICTraceExtensions>
</ePICApplication>

```

図 5. デフォルトの *TraceClient* 定義: ハンドラーの登録

複数のハンドラーを指定するには、`epictracehandler` エレメント中で `Value` エレメントを使用してください。たとえば、図6 には、`TraceClient` のコンソール・ハンドラーに加えて、ファイル・ハンドラーも登録する方法を示しています。

```

<ePICApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracehandler>
      <Value>com.ibm.logging.ConsoleHandler</Value>
      <Value>com.ibm.logging.FileHandler</Value>
    </epictracehandler>
    ...
  </ePICTraceExtensions>
</ePICApplication>

```

図 6. *TraceClient* 定義: 複数のハンドラーの登録

ハンドラーの構成: 5 つのハンドラーの要件はそれぞれ違うので、ハンドラーごとに独自の構成が必要です。たとえば、ハンドラーごとに出力の形式を設定する方法を構成する必要があります。また、ファイル・ハンドラーの場合は書き込み先のファイルも構成する必要があります。`aqmconfig.xml` ファイル中の `TraceClient` 項目には、5 つすべてのタイプのハンドラーの構成が示されています。しかし、他のトレース・クライアントの場合は、構成する必要があるのは登録されているハンドラーだけです。

`epictracehandler` エレメントを使用して、トレース・クライアントによって使用されるハンドラーを登録します。ハンドラーを登録したら構成も行わなければなりません。`ePICTraceHandler` エレメントを使用して、個々のハンドラーを構成します。このエレメントで属性 `epictracehandler` を使用して、構成するハンドラーを指示します。この方法で、複数のハンドラーを構成できます。特定のハンドラーの構成値を入力しないと、そのハンドラーではデフォルト値が使用されます。現在登録されていないハンドラーも構成できます。ただし、このハンドラーは登録しない限り使用されません。

コンソール・ハンドラー: トレース・クライアントのコンソール・ハンドラーを登録したら、そのハンドラーで使用されるフォーマッターも指示できます。MQSeries Adapter Kernel には `com.ibm.epic.trace.client.EpicTraceFormatter` というクラスが備えられており、コンソール・ハンドラーやファイル・ハンドラーで使用できます。このフォーマッターはデフォルトです。コンソール・ハンドラーの場合は追加の構成は必要ありま

せん。フォーマッターを指定する場合、図7 に示すように、epictraceformatter エレメントの中で指示します。

```
<ePICAApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracehandler>com.ibm.logging.ConsoleHandler</epictracehandler>
    <ePICTraceHandler epictracehandler="com.ibm.logging.ConsoleHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.EpicTraceFormatter
      </epictraceformatter>
    </ePICTraceHandler>
    ...
  </ePICTraceExtensions>
</ePICAApplication>
```

図7. トレース・クライアントのコンソール・ハンドラーの構成

ファイル・ハンドラー: トレース・クライアントのファイル・ハンドラーを登録したら、そのハンドラーで使用されるフォーマッターと、生成されたメッセージを保持するファイルも指示できます。MQSeries Adapter Kernel には `com.ibm.epic.trace.client.EpicTraceFormatter` というクラスが備えられており、コンソール・ハンドラーやファイル・ハンドラーで使用できます。このフォーマッターはデフォルトです。ファイル名は任意です。TraceClient のデフォルトのファイル名は `trc.log` です。epictracehandler エレメントでファイル・ハンドラーが登録されていないと、TraceClient ではファイルへの書き込みが行われないので注意してください。

10ページの図8 が示すように、フォーマッターは epictraceformatter エレメントの中で明示的に指定できます。ファイルは epictracefilename エレメントの中で指定します。このハンドラーにより 1 つのファイルが生成され、このファイルのサイズは無限に大きくなることができます。

```

<ePICApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracehandler>com.ibm.logging.FileHandler</epictracehandler>
    ...
    <ePICTraceHandler epictracehandler="com.ibm.logging.FileHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.EpicTraceFormatter
      </epictraceformatter>
      <epictracefilename>trc.log</epictracefilename>
    </ePICTraceHandler>
    ...
  </ePICTraceExtensions>
</ePICApplication>

```

図 8. トレース・クライアントのファイル・ハンドラーの構成

マルチファイル・ハンドラー: マルチファイル・ハンドラーの構成について詳しくは、16ページの『マルチファイル・ハンドラー』を参照してください。

ソケット・ハンドラーと ENA ハンドラー: トレース・クライアントのソケット・ハンドラーか ENA ハンドラーを登録したら、そのハンドラーで使用されるフォーマッターと、宛先のトレース・サーバーのアプリケーション ID も指示できます。MQSeries Adapter Kernel には `com.ibm.epic.trace.client.EpicXMLFormatter` というクラスが備えられており、ソケット・ハンドラーや ENA ハンドラーで使用できます。このフォーマッターはデフォルトです。コンソール・ハンドラーやファイル・ハンドラーと同様に、`epictraceformatter` エlement中にフォーマッターを明示的に指定できます。

ソケット・ハンドラーの場合はソケット接続を介して、ENA ハンドラーの場合は組み込みの固有アダプターを介して、トレース・メッセージがトレース・サーバーに転送されます。これらのトレース・サーバーも、アプリケーション ID の下で構成されています。ソケット・ハンドラーと ENA ハンドラーの両方とも、使用されるトレース・サーバーを指示しなければなりません。トレース・サーバーを指示するには、`epicdepappid` (dependency application - 依存関係にあるアプリケーション) Elementで指定します。このElementは `ePICTraceHandler` Element中でネストされず、`ePICTraceExtensions` Elementに直属します。MQSeries Adapter Kernel には、デフォルトのトレース・サーバーの `TraceServer` が組み込まれています。トレース・サーバーを指定しないと、デフォルトの `TraceServer` 名が使用されます。

`epictracehandler` Elementでソケット・ハンドラーか ENA ハンドラーが登録されていないと、`TraceClient` ではトレース・サーバーが使用されないの注意してください。

デフォルトの `TraceServer` 以外の場合、`epicdepappid` Elementで名前が付けられたトレース・サーバーはまた、構成ファイルの中で構成もしなければなりません。ソケット・ハンドラーでは、トレース・サーバー・アプリケーション ID (`epicdepappid` Element中で使用されている) を使用して、トレース・サーバーに接続する際のホスト名

とポート番号が判別されます。ソケット・ハンドラーにデフォルトを使用しない場合は、トレース・サーバーを定義する際にソケット・ハンドラーの構成も行う必要があります。詳しくは、12ページの『トレース・サーバーの定義』を参照してください。

ENA ハンドラーでは、ネイティブ・アダプターを使用して、トレース・メッセージがトレース・サーバーに送信されます。図9 に示すように、トレース・サーバー・アプリケーション ID (`epicdepappid` エレメント中で使用される) が、ENA ハンドラーによって宛先アプリケーション ID として使用されます。トレース・サーバーの構成項目には、ネイティブ・アダプターのための `AdapterRouting` エレメントの中で、適切なアプリケーション通信情報を組み込まなければなりません。詳しくは、12ページの『トレース・サーバーの定義』を参照してください。

```
<ePICAApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracehandler>com.ibm.logging.ConsoleHandler</epictracehandler>
    ...
    <epicdepappid>TraceServer</epicdepappid>
    <ePICTraceHandler epictracehandler="com.ibm.epic.trace.client.ENAHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.EpicXMLFormatter
      </epictraceformatter>
    </ePICTraceHandler>
    <ePICTraceHandler epictracehandler="com.ibm.logging.SocketHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.EpicXMLFormatter
      </epictraceformatter>
    </ePICTraceHandler>
  </ePICTraceExtensions>
</ePICAApplication>
```

図9. トレース・クライアントのソケット・ハンドラーと ENA ハンドラーの構成

`TraceClient` アプリケーションの完全な構成項目が、12ページの図10 に示されています。このクライアントではコンソール・ハンドラーだけが登録されていますが、構成項目にはファイル、ソケット、および ENA ハンドラーの構成も示されています。これらのハンドラーは、コンソール・ハンドラーの代わりまたは追加として登録しない限り使用されません。

```

<ePICApplication epicappid="TraceClient">
  <ePICTraceExtensions cn="epicappextensions">
    <epictracemessagefile>
      com.ibm.epic.trace.client.TraceMessage
    </epictracemessagefile>
    <epictracesyncoperation>false</epictracesyncoperation>
    <epictracehandler>com.ibm.logging.ConsoleHandler</epictracehandler>
    <ePICTraceHandler epictracehandler="com.ibm.logging.ConsoleHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.EpicTraceFormatter
      </epictraceformatter>
    </ePICTraceHandler>
    <ePICTraceHandler epictracehandler="com.ibm.logging.FileHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.EpicTraceFormatter
      </epictraceformatter>
      <epictracefilename>trc.log</epictracefilename>
    </ePICTraceHandler>
    <epicdepappid>TraceServer</epicdepappid>
    <ePICTraceHandler epictracehandler="com.ibm.epic.trace.client.ENAHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.EpicXMLFormatter
      </epictraceformatter>
    </ePICTraceHandler>
    <ePICTraceHandler epictracehandler="com.ibm.logging.SocketHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.EpicXMLFormatter
      </epictraceformatter>
    </ePICTraceHandler>
  </ePICTraceExtensions>
</ePICApplication>

```

図 10. 完全なデフォルト *TraceClient* 定義

トレース・サーバーの定義

カーネルのコンポーネントを開始すると、構成ファイルが読み取られ、実行時の設定が判断されます。トレースに関して、始動時に受け取ったアプリケーション ID を使用し、トレースを使用可能にするかどうか決定します。

トレースを使用可能にする場合は、まずトレース・クライアントがインスタンス化されます。続いてトレース・クライアントで、インスタンス化を実行したアプリケーションのアプリケーション ID に基づいて、トレースを使用可能にするかどうか、トレース・レベル、およびトレース・クライアント ID が判別されます。続いてトレース・クライアントで、トレース・クライアント ID に基づいて、使用するハンドラーが判別されず。

ソケット・ハンドラーか ENA ハンドラーを登録する場合は、トレース・クライアントの epicdepappid エレメントで使用されているアプリケーション ID を使用して、トレ

ース・サーバーも開始しなければなりません。続いてトレース・サーバーにより、トレース・ワーカーという特別なトレース・クライアントが複数インスタンス化されます。トレース・ワーカーの構成は、トレース・サーバーの構成に組み込まれます。トレース・ワーカーの構成項目は、独立したアプリケーション ID の下にはありません。

デフォルトの `aqmconfig.xml` ファイルには、デフォルトのトレース・サーバー ID である `TraceServer` の項目が含まれています。この項目には、`TraceServer` のデフォルト値が反映されています。この項目をテンプレートとして使用して新しいトレース・サーバーを構成したり、変更を加えて `TraceServer` の能力を高めたりできます。

`TraceServer` のデフォルト値はアダプター・カーネルに組み込まれています。構成ファイル中の定義を使用してデフォルトの動作をオーバーライドすることはできますが、構成ファイルから `TraceServer` 定義を削除してデフォルトを除去することはできません。`TraceServer` 定義を削除すると、トレース・サーバー ID が指定されていないアプリケーションでは、組み込みの値が使用されます。

トレース・クライアントと同様に、トレース・サーバーは `ePICApplication` エレメント中で構成され、アプリケーション ID は属性 `epicappid` により指定されます。図11に示すように、構成項目の本体は、`ePICApplication` エレメント、`AdapterRouting` エレメント、および `ePICTraceExtensions` エレメントの中でネストされる 2 つのエレメントから成ります。ENA ハンドラーを使用する場合、トレース・サーバーは `AdapterRouting` エレメントを使用します。なぜなら、ENA ハンドラーがネイティブ・アダプターを使用するからです。ネイティブ・アダプターは、`AdapterRouting` エレメントからその構成値を取得します。この説明は、`ePICTraceExtensions` エレメントに集中的に記述されます。`AdapterRouting` エレメントの作成に関する情報は、`スタートアップ・ガイド` を参照してください。

```
<ePICApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    <epicmqppqueuemgr>DEFAULT</epicmqppqueuemgr>
    <ePICBodyCategory epicbodycategory="DEFAULT">
      <ePICBodyType epicbodytype="DEFAULT">
        <epicreceivemode>MQPP</epicreceivemode>
        <epicreceivemqppqueue>TraceServerAIQ</epicreceivemqppqueue>
      </ePICBodyType>
    </ePICBodyCategory>
  </AdapterRouting>
  <ePICTraceExtensions cn="epicappextensions">
    ...
  </ePICTraceExtensions>
</ePICApplication>
```

図 11. デフォルトの `TraceServer` 定義の構造

トレース・サーバーのメッセージ・ファイル

トレース・クライアントと同様に、トレース・サーバーでも `epictracemessagefile` エレメントにより、メッセージ・ストリングを生成する内部機能に対するメッセージ・カタログが備えられます。(詳しくは、6ページの『トレース・クライアントのメッセージ・ファイル』を参照してください。) 図12 に示すように、`TraceServer` の定義には、デフォルトのカタログ `com.ibm.epic.trace.server.TraceServerMessage` の名前が明示的に指定されています。他のトレース・サーバーでは、デフォルト・ファイルを受け入れ可能な場合 `epictracemessagefile` エレメントを省略できます。

```
<ePICAApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    ...
  </AdapterRouting>
  <ePICTraceExtensions cn="epicappextensions">
    <epictracemessagefile>
      com.ibm.epic.trace.server.TraceServerMessage
    </epictracemessagefile>
    ...
  </ePICTraceExtensions>
</ePICAApplication>
```

図 12. デフォルトの `TraceServer` 定義: メッセージ・カタログ

トレース・サーバーのメッセージ・バッファ

トレース・クライアントと同様に、トレース・サーバーは `epictracesyncoperation` エレメントを使用して、メッセージを同期的に書き込むか非同期的に書き込むかを指示します。図13 に示すように、デフォルト値の `false` は非同期動作を指定します。(詳しくは、6ページの『トレース・クライアントのメッセージ・バッファ』を参照してください。) `TraceServer` の定義では、非同期動作が明示的に要求されます。他のトレース・サーバーでは、非同期書き込みを受け入れてよい場合は `epictracesyncoperation` エレメントを省略できます。

```
<ePICAApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    ...
  </AdapterRouting>
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracesyncoperation>false</epictracesyncoperation>
    ...
  </ePICTraceExtensions>
</ePICAApplication>
```

図 13. デフォルトの `TraceServer` 定義: メッセージ・バッファ

トレース・サーバーのメッセージ・ハンドラー

トレース・サーバー用に登録されたメッセージ・ハンドラーにより、収集されたトレース・メッセージの送信場所が指定されます。以下の 3 つの宛先を指定できます。

- コンソール
- ファイル
- ファイルの集合

上記の出力宛先はそれぞれ別のハンドラーによって管理されます。トレース・サーバーの場合、以下のハンドラーを登録できます。

- `com.ibm.logging.ConsoleHandler`
- `com.ibm.logging.FileHandler` (1 つのファイル)
- `com.ibm.logging.MultiFileHandler` (バックアップ循環ファイル)

トレース・サーバーのソケット・ハンドラーや ENA ハンドラーを登録しないでください。トレース・サーバーは、ソケット・ハンドラーや ENA ハンドラーをすでに使用しているトレース・クライアントからのメッセージの経路を定めるのに使用されるからです。トレース・サーバーでもこれらのハンドラーを使用することはできません。

図14 に示すように、`epictracehandler` エレメントを使用してハンドラーを登録します。TraceServer 定義では、トレース・メッセージはマルチファイル・ハンドラーに送信されます。他のトレース・サーバーでは、1 つまたは複数のハンドラーを指定しなければなりません。複数のハンドラーを登録することに関する情報は、8ページの図6 を参照してください。ハンドラーを登録したら、構成も行わなければなりません。詳しくは、『ハンドラーの構成』を参照してください。

```
<ePICApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    ...
  </AdapterRouting>
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracehandler>com.ibm.logging.MultiFileHandler</epictracehandler>
    ...
  </ePICTraceExtensions>
</ePICApplication>
```

図 14. デフォルトの TraceServer 定義: ハンドラーの登録

ハンドラーの構成: ハンドラーの要件はそれぞれ違うので、ハンドラーごとに独自の構成が必要です。たとえば、ハンドラーごとに出力の形式を設定する方法を構成する必要があります。MQSeries Adapter Kernel には `com.ibm.epic.trace.client.ReFormatter` というクラスが備えられており、コンソール・ハンドラーやファイル・ハンドラーを指定したトレース・サーバーで使用できます。また、ファイル・ハンドラーの場合は書き

込み先のファイルも構成する必要があります。aqmconfig.xml ファイル中の TraceServer 項目には、両方のタイプのハンドラーの構成が示されています。しかし、他のトレース・サーバーの場合は、構成する必要があるのは登録されているハンドラーだけです。

epICTraceHandler エレメントを使用して、個々のハンドラーを構成します。このエレメントで属性 epictracehandler を使用して、構成するハンドラーを指示します。この方法で、複数のハンドラーを構成できます。しかし、epictracehandler エレメントで登録されたハンドラーだけが使用されます。

マルチファイル・ハンドラー: トレース・サーバーでマルチファイル・ハンドラーを登録すると、ハンドラーで使用されるフォーマッター、ファイルの基底名、ファイルの数、およびファイルのサイズを指示できます。MQSeries Adapter Kernel には com.ibm.epic.trace.client.ReFormatter というクラスが備えられており、コンソール・ハンドラーやファイル・ハンドラーを指定したトレース・サーバーで使用できます。これはデフォルトのフォーマッターです。ファイル名は任意です。TraceServer のデフォルトのファイル名は、trc.log です。17 ページの図 15 に示すように、デフォルトのファイル数は 3 つで、デフォルトのサイズは 1,000,000 バイト (1-KB のブロック 1000 個分) です。

フォーマッターは、epictraceformatter エレメントの中で明示的に指定できます。ファイルは epictracefilename エレメントの中で指定し、ファイルの数は epictracefilenumber エレメントの中で指定し、サイズ (1-KB のブロック単位) は epictracefilesize エレメントの中に指定します。

マルチファイル・ハンドラーはまず trc1.log というファイルに書き込むことから始め、このファイルが指定のサイズに達するまで書き込みを行います。指定のサイズに達したら、trc1.log を trc2.log という名前のファイルにコピーし、ファイル trc1.log を上書きし始めます。次にファイルが指定のサイズに達すると、ファイル trc2.log が trc3.log という名前のファイルにコピーされ、trc1.log が trc2.log にコピーされ、ハンドラーは trc1.log の上書きを再開します。ファイルの数が指定値に達した場合 (この例では 3、trc3.log に達した場合) は、ハンドラーは、最も古くて数値の大きいファイルを保管しなくなります。この例では、ファイル trc3.log が新しい名前のファイルにコピーされることはありません。trc2.log ファイルがコピーされるたびに置き換えられます。

```

<ePICAApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    ...
  </AdapterRouting>
  <ePICATraceExtensions cn="epicappextensions">
    ...
    <epictracetracehandler>com.ibm.logging.MultiFileHandler</epictracetracehandler>
    <ePICATraceHandler epictracetracehandler="com.ibm.logging.MultiFileHandler">
      <epictracetraceformatter>
        com.ibm.epic.trace.client.ReFormatter
      </epictracetraceformatter>
      <epictracetracefilename>trc.log</epictracetracefilename>
      <epictracetracefilenumber>3</epictracetracefilenumber>
      <epictracetracefilesize>1000</epictracetracefilesize>
    </ePICATraceHandler>
    ...
  </ePICATraceExtensions>
</ePICAApplication>

```

図 15. トレース・サーバー中のマルチファイル・ハンドラーの構成

コンソール・ハンドラー: トレース・サーバーのコンソール・ハンドラーを登録したら、そのハンドラーで使用されるフォーマッターも指示できます。MQSeries Adapter Kernel には `com.ibm.epic.trace.client.ReFormatter` というクラスが備えられており、コンソール・ハンドラーやファイル・ハンドラーを指定したトレース・サーバーで使用できます。これはデフォルトです。図16 に示すように、フォーマッターは `epictracetraceformatter` エレメントの中に明示的に指定できます。コンソール・ハンドラーの場合は追加の構成は必要ありません。コンソール・ハンドラーを `epictracetracehandler` エレメント中に登録しないと、トレース・サーバーのコンソールへの書き込みは行われないので注意してください。

```

<ePICAApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    ...
  </AdapterRouting>
  <ePICATraceExtensions cn="epicappextensions">
    ..
    <epictracetracehandler>com.ibm.logging.ConsoleHandler</epictracetracehandler>
    ...
    <ePICATraceHandler epictracetracehandler="com.ibm.logging.ConsoleHandler">
      <epictracetraceformatter>
        com.ibm.epic.trace.client.ReFormatter
      </epictracetraceformatter>
    </ePICATraceHandler>
  </ePICATraceExtensions>
</ePICAApplication>

```

図 16. トレース・サーバーのコンソール・ハンドラーの構成

トレース・ワーカーの構成

トレース・サーバーは、トレース・クライアントでソケット・ハンドラーか ENA ハンドラーが登録されている場合に限り使用されます。トレース・サーバー自体は、ワーカーという特別なトレース・クライアントを利用します。ワーカーは、正規のトレース・クライアントとは違って、独立したアプリケーション ID の下では構成されません。トレース・ワーカーの構成は、トレース・サーバーの構成中に組み込まれます。

トレース・サーバーはソケット・ハンドラーか ENA ハンドラーを使用するトレース・クライアント用に構成されるので、トレース・サーバーの構成にはソケット・ハンドラーか ENA ハンドラーの `ePICTraceHandler` エLEMENTが含まれます。しかし、トレース・サーバーでこれらのハンドラーを登録することはできません。代わりに、ソケット・ハンドラーの構成は、ソケット・ハンドラーを登録したトレース・クライアントからのメッセージを受信するトレース・サーバー中のトレース・ワーカーにより使用されます。同様に、ENA ハンドラーは、ENA ハンドラーを使用するトレース・クライアントからのメッセージがトレース・サーバーで受信された際にトレース・ワーカーで使用されます。

`aqmconfig.xml` ファイル中の `TraceServer` 項目には、両方のタイプのワーカーの構成が示されています。しかし、他のトレース・サーバーの場合は、構成する必要があるのはトレース・クライアントに必要なワーカーだけです。

ソケット・ハンドラー: ソケット・ハンドラーを使用するようにトレース・ワーカーを構成することはできません。しかし、ソケット・ハンドラーを使用してトレース・クライアントからトレース・サーバーに接続する場合に、デフォルトのサーバー・ホストとポート番号を使用しないのであれば、トレース・サーバーにソケット・ハンドラーの項目が必要です。19ページの図17 に示すように、`ePICTraceHandler` エLEMENTにこのハンドラーを指定し、このELEMENT中で接続先のホストとポートを指定してください。`epictracesocketserverhost` エLEMENTはホストを指示します。デフォルトは `"localhost"` です。`epictraceportnumber` エLEMENTはポートを指示します。デフォルトは `8181` です。

```

<ePICApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    ...
  </AdapterRouting>
  <ePICTraceExtensions cn="epicappextensions">
    ...
    <epictracehandler>com.ibm.logging.MultiFileHandler</epictracehandler>
    ...
    <ePICTraceHandler epictracehandler="com.ibm.logging.SocketHandler">
      <epictracesocketserverhost>localhost</epictracesocketserverhost>
      <epictraceportnumber>8181</epictraceportnumber>
    </ePICTraceHandler>
  </ePICTraceExtensions>
</ePICApplication>

```

図 17. トレース・ワーカーのソケット・ハンドラー用の構成

ENA ハンドラー: ENA ハンドラーを使用するようにトレース・ワーカーを構成することはできません。しかし、ENA ハンドラーを使用してトレース・クライアントからトレース・サーバーに接続する場合は、トレース・サーバーの構成に AdapterRouting エレメントを組み込まなければなりません。図18 に示すように、このエレメントは、クライアントの ENA ハンドラーとトレース・サーバー・デーモンによって使用されます。AdapterRouting エレメントの作成について詳しくは、*スタートアップ・ガイド* を参照してください。

```

<ePICApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    <epicmqppqueuemgr>DEFAULT</epicmqppqueuemgr>
    <ePICBodyCategory epicbodycategory="DEFAULT">
      <ePICBodyType epicbodytype="DEFAULT">
        <epicreceivemode>MQPP</epicreceivemode>
        <epicreceivemqppqueue>TraceServerAIQ</epicreceivemqppqueue>
      </ePICBodyType>
    </ePICBodyCategory>
  </AdapterRouting>
</ePICApplication>

```

図 18. トレース・ワーカーの ENA ハンドラー用の構成

TraceServer アプリケーションの完全な構成項目が、20ページの図19 に示されています。このサーバーにはマルチファイル・ハンドラーしか登録されていませんが、項目にはコンソール・ハンドラーの構成も示されています。このハンドラーは、マルチファイル・ハンドラーの代わりまたは追加として登録しない限り使用されません。このサーバーではソケット・ハンドラーも構成されています。トレース・サーバーでは、このハンドラーは登録できません。このハンドラーは、ソケット・ハンドラーを使用するトレース・クライアントからのメッセージを受信するトレース・ワーカーによって使用されません。

```

<ePICApplication epicappid="TraceServer">
  <AdapterRouting cn="epicadapterrouting">
    <epicmqppqueuemgr>DEFAULT</epicmqppqueuemgr>
    <ePICBodyCategory epicbodycategory="DEFAULT">
      <ePICBodyType epicbodytype="DEFAULT">
        <epicreceivemode>MQPP</epicreceivemode>
        <epicreceivemqppqueue>TraceServerAIQ</epicreceivemqppqueue>
      </ePICBodyType>
    </ePICBodyCategory>
  </AdapterRouting>
  <ePICTraceExtensions cn="epicappextensions">
    <epictracesyncoperation>false</epictracesyncoperation>
    <epictracemessagefile>
      com.ibm.epic.trace.server.TraceServerMessage
    </epictracemessagefile>
    <epictracehandler>com.ibm.logging.MultiFileHandler</epictracehandler>
    <ePICTraceHandler epictracehandler="com.ibm.logging.SocketHandler">
      <epictracesocketserverhost>localhost</epictracesocketserverhost>
      <epictraceportnumber>8181</epictraceportnumber>
    </ePICTraceHandler>
    <ePICTraceHandler epictracehandler="com.ibm.logging.ConsoleHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.ReFormatter
      </epictraceformatter>
    </ePICTraceHandler>
    <ePICTraceHandler epictracehandler="com.ibm.logging.MultiFileHandler">
      <epictraceformatter>
        com.ibm.epic.trace.client.ReFormatter
      </epictraceformatter>
      <epictracefilename>trc.log</epictracefilename>
      <epictracefilenumber>3</epictracefilenumber>
      <epictracefilesize>1000</epictracefilesize>
    </ePICTraceHandler>
  </ePICTraceExtensions>
</ePICApplication>

```

図 19. 完全なデフォルト *TraceServer* 定義

トレース・メッセージの読み取り

トレースを要求すると、カーネルでトレース・メッセージが生成されてご希望の宛先に送信されます。トレース・メッセージの内容は、アプリケーションやトレース要求のレベルに応じて変わります。しかし、以下のものは共通して組み込まれます。

- タイム・スタンプ (日時)
- トレース・メソッドを含むクラスの名前
- スレッド名 (たとえば main、Thread-1、Thread-2)
- トレース・メソッドの名前と署名

- 編成 (常に "ePIC")
- アプリケーション ID
- ソース論理 ID (使用可能な場合)
- 宛先論理 ID (使用可能な場合)
- 本体のカテゴリ (使用可能な場合)
- 本体のタイプ (使用可能な場合)
- メッセージ ID (使用可能な場合)
- トランザクション ID (使用可能な場合)
- トレース・レベル (使用可能な場合)
- イベントを記述したトレース情報。これには、メッセージ・コード (ストリング AQM の後に 4 桁の数値がある) と、対応する情報ストリングが組み込まれます。

以下の例は、2 つの代表的なトレース・メッセージを示しています。

```
2000.05.18 08:57:48.742 com.ibm.epic.adapters.eak.nativeadapter.EpicNativeAdapter
Thread Name=main getLMSInstanceForApplication(String, String, String) ePIC TEST2
TYPE_INFO AQM5003: <CommunicationMode> は、値 <MQPP> を持ちます。
2000.05.18 09:02:27.883 com.ibm.epic.adapters.eak.nativeadapter.LMSMQ
Thread Name=Thread-2 getReceiveQName(String, String, String) ePIC TEST2
TYPE_INFO AQM5010: BodyType <DEFAULT>,BodyCategory <DEFAULT>
アプリケーション ID <TEST2> 属性フィルター <epicreceivemqppqueue> 値 <TEST2AIQ>
の AdapterDirectory からの結果。
```

図 20. 2 つのトレース・メッセージ

1 つ目のメッセージは、EpicNativeAdapter クラスの getLMSInstanceForApplication メソッドに基づいています。このメッセージは、アプリケーション TEST2 のメイン・スレッドにより収集されたものです。ストリング 「<CommunicationMode> は、値 <MQPP> を持ちます。」がトレース情報です。

2 つ目のメッセージは、LMSMQ クラスの getReceiveQName メソッドに基づいています。このメッセージは、アプリケーション TEST2 のスレッド 2 により収集されたものです。先頭が 「・・・の AdapterDirectory からの結果。」のストリングがトレース情報です。

第2章 構成に関する一般的な問題

以下に、構成に関する一般的な問題の一部を取り上げ、考えられる解決方法を説明します。以下の情報の大部分は MQSeries Adapter Kernel に直接適用されますが、MQSeries に関する特定の問題 (25ページの『MQSeries に関する問題』を参照)、MQSeries Integrator に関する特定の問題 (28ページの『MQSeries Integrator に関する問題』を参照)、およびその他の特定の問題 (30ページの『その他の問題』を参照) も扱っていません。

構成上の問題の多くは、スペルの検査という 1 つのステップで解決できます。キュー名、クラス名、アプリケーション ID などのつづりを誤っていると、実際よりも重大に見えるエラーを起こすことがあります。

カーネル構成に関する一般問題

ファイル aqmsetup が見つからない

問題: aqmsetup ファイルが見つかりませんでした。

応答: AQMSETUPFILE 環境変数が、aqmsetup ファイルのあるディレクトリーに正しく設定されているか確認してください。ファイルの名前が正しいか確認してください。

ファイル aqmconfig.xml が見つからない

問題: aqmconfig.xml ファイルが見つかりませんでした。

応答: aqmconfig.xml ファイルが正しいディレクトリー中にあるか確認してください。

応答: 対応する aqmsetup ファイルを探し出して、AQMCONFIG= 項目が完全修飾パスを使用して正しいディレクトリーを指しているか確認してください。

メッセージに対する無応答

問題: ターゲット・アダプターでメッセージが正常に受信されていますが、応答が受信されていません。

応答: 元のメッセージのヘッダー中に、確認通知フラグが設定されているか確認してください。

ターゲット・アダプターが見つからない

問題: ターゲット・アダプターが見つからなかったため、ロードされませんでした。

応答: 例外メッセージで指定されているターゲット・アダプターが存在しているか確認してください。

応答: CLASSPATH 環境変数に、ターゲット・アダプターがあるディレクトリーが含まれているか確認してください。

本体データの構文解析エラー

問題: ターゲット・アダプターで、メッセージの本体データを構文解析できません。

応答: 本体データが XML で DTD を参照している場合は、原因として最も可能性が高いのは、ターゲット・アダプターで XML DTD を見つけられなかったことです。aqmsetup ファイルが XML_DTD_DIRECTORY 環境変数を定義して DTD ファイル (複数の場合もあります) を指していることを確認してください。

ネイティブ・アダプターからの例外

問題: ネイティブ・アダプター (送信または受信) が例外をスローします。

応答: 適切なキュー名が指定されていることを確認してください。ネイティブ・アダプターがトレース・クライアントに属している場合は、トレース・サーバーのキュー名が指定されているか確認してください。

応答: 有効な通信モードが指定されているか確認してください。

応答: 通信モードで、MQSeries を使用することが必要な場合、CLASSPATH 環境変数に MQSeries JAR ファイルが組み込まれていることを確認してください。

アダプター・デーモンのシャットダウン

問題: アダプター・デーモンがシャットダウンします。

応答: EpicSystemExceptionFilesequence-number.log ファイルがあるか検査してください。

応答: 使用可能なメモリーが十分あるか確認してください。Java™ OutOfMemoryError がアダプター・ワーカーのシャットダウンの原因になります。(詳しくは、31ページの『Java メモリーの問題: OutOfMemoryError』を参照してください。)

応答: メッセージがエラー・キューに転送されたため、エラー・キューがいっぱいになっているかどうかを判別してください。該当する場合は、メッセージがエラー・キューに転送される理由を判別してください。

受信したメッセージがアプリケーションに送信されない

問題: メッセージが着信キューから正常に除去されますが、受信先アプリケーションに転送されません。

応答: メッセージがエラー・キューに経路指定されていないか確認してください。この問題には、以下の症状があります。

- ターゲット・アダプター・コマンドを見つけれられない。
- ターゲット・アダプター・コマンドによってエラーが戻される。
- EpicSystemExceptionFilesequence-number.log ファイルが生成される。

着信キューからメッセージが除去されない

問題: メッセージが着信キューから正常に除去されません。

応答: アダプター・デーモンが実行されているか確認してください。(aqmstrad コマンドを使用すると開始できます。)

応答: EpicSystemExceptionFilesequence-number.log ファイルがあるか検査してください。

ソケット・ハンドラーに関するトレース・エラー

問題: ソケット・ハンドラーを使用してトレース・メッセージを収集しようとしたが、失敗しました。

応答: トレース・サーバーが実行されているか確認してください。(aqmstrtd コマンドを使用すると開始できます。)

トレース・クライアントのインスタンス化の失敗

問題: トレース・クライアントをインスタンス化しようとしたが、失敗しました。

応答: 原因として最も可能性が高いのは、アプリケーションか、またはインスタンス化を試行したトレース・クライアントの構成中のエラーです。

EpicSystemExceptionFilesequence-number.log ファイルを探してください。

トレース・クライアントまたはトレース・サーバーのループ

問題: トレース・クライアントまたはトレース・サーバーがループしています。

応答: トレース・クライアントまたはトレース・サーバー自体が、トレースがオンになるよう構成されているか確認してください。

応答: トレース・サーバーが、ソケット・ハンドラーまたは固有アダプター・ハンドラーを使用するよう構成されているか確認してください。

MQSeries に関する問題

アプリケーションでアダプターが使用される場合には、アプリケーション間でのメッセージの移動という基礎的な移送が行えなければなりません。移送に関する問題が起きると、移送に依存しているアプリケーションに影響があります。以下に、MQSeries に関連して起こる問題について説明します。

キューが見つからない

問題: 名前が指定されているキューがありません。

応答: この名前が存在していて、スペルが適切か確認してください。

応答: MQSeries ユーティリティを使用して、キューを作成してください。

チャンネルのダウン

問題: MQSeries チャンネルが利用不能です。

応答: MQSeries ユーティリティを使用して、必要なコンポーネントがすべて実行されていることを確認してください。

キュー項目数の超過

問題: MQSeries キューの項目数が限度を超えています。

応答: MQSeries 環境変数 `MAX_QUEUE_DEPTH` の値を適切に設定してください。

応答: MQSeries またはアプリケーション固有のユーティリティを使用して、キューからエレメントを除去してください。

応答: エレメントがキュー上に残っているのは、別の問題が起きたことが原因であるかどうか判別してください。

ログが使用可能なスペースを超過した

問題: MQSeries ログ・ファイルが、ファイル・システム中の使用可能なスペースの限度を超えています。

応答: ログを調べて、どのログを保管するか決めてください。必要なログを 2 次ストレージに移動して、残りのログをファイル・システムから除去してください。

MQSeries JAR ファイルが見つからない

問題: MQSeries JAR ファイルが見つかりません。

応答: `CLASSPATH` 環境変数に、MQSeries JAR ファイルを含むディレクトリーが組み込まれているか確認してください。

メッセージ・カタログをロードできない

問題: MQJI メッセージ・カタログをロードできません。

応答: `CLASSPATH` 環境変数に、`mjji*.properties` ファイルを含むディレクトリーが組み込まれているか確認してください。

キュー・マネージャーが使用できない

問題: AIX®上で、MQSeries が理由コード 2059 (QMgr 使用不能) でエラーを戻します。共用メモリー域に、not valid from shmat および XC211017, xstConnectSegmentViaFile メッセージを出す問題があります。

応答: このエラーは、AIX 上で Java Development Kit (JDK) のいくつかのバージョンでのメモリー競合が原因で起こります。詳細は、www.ibm.com/software/ts/mqseries/support/summary/jvreadme.html を参照してください。

本書は以下のような構成になっています。

- MQSeries 5.1 を使用している場合、Java Virtual Machine (JVM)を始動する前に、LDR_CNTRL 環境変数の値を 0x30000000 に設定します。
- MQSeries 5.2. を使用している場合、以下のように IPCCBaseAddress パラメーターの値を、mq5.ini ファイルの各キュー・マネージャーごとに、12 に設定します。

```
QueueManager:  
  Name=MQJavaTest  
  Prefix=/var/mqm  
  IPCCBaseAddress=12
```

値は 12 を推奨します。しかし、4、5、8、9、10、11 も許容されます。詳しくは、MQSeries 5.2 の資料を参照してください。

受信例外

問題: 受信例外が `java.lang.NoClassDefFoundError: class_name` というエラー・メッセージとともに発生します。ここで、`class_name` は `com/ibm/mq/MQException`、または別の MQSeries Java クラスです。

応答: MQSeries JAR ファイルがクラス・パスに組み込まれていることを確認してください。MQSeries 5.2 を実行している場合は、Java をサポートする SupportPac MA88 を必ずインストールしてください。

ResourceException クラスが見つからない

問題: `java.lang.NoClassDefFoundError` が、`javax/resource/ResourceException` クラス用に受信されます。

応答: Sun Microsystems Java Web サイトから、`connector.jar` という名前の Java JAR ファイルを取得してください。詳細とリンクについては、www.ibm.com/software/ts/mqseries/support/summary/jvreadme.html を参照してください。2001 年 2 月 28 日付 MQSeries SupportPac MA88 には、この JAR ファイルは組み込まれていません。しかし、この SupportPac 以降のリリースには、`connector.jar` ファイルからの必要なクラスが必ず組み込まれています。(この JAR ファイルは SupportPac に含まれてなく不要です。)

MQSeries クラスが見つからない

問題: java.lang.NoClassDefFoundError: Message information:
com/ibm/mqbind/MQSESSION に類似した例外が受信されます。

応答: インストールした MQSeries Java クラス・ファイルが古いバージョンでないことを確認してください。IBM Connectors feature of IBM VisualAge™ for Java バージョン 3.02 をインストールした場合、古いファイルがインストールされた可能性があります。古いクラス・ファイルをインストールした場合、クラス・パスから古いクラス・ファイルを除去してください。

MQSeries Integrator に関する問題

メッセージが理解されない

問題: MQSeries Integrator でメッセージが理解されません。

応答: 正しい通信モード (MQRFH2 など) が指定されているか確認してください。

MQSI のメッセージ受信

問題: MQSeries Integrator からのメッセージを受信すると、アプリケーションでエラーが取得されます。

応答: 受信側アプリケーションの通信モードが、MQSI でメッセージを送信する際の通信モードと一致しているか確認してください。

JMS に関する問題

アプリケーションでアダプターが使用される場合には、アプリケーション間でのメッセージの移動という基礎的な移送が行えなければなりません。移送に関する問題が起きると、移送に依存しているアプリケーションに影響があります。以下に、JMS を使用するアプリケーションに関する問題について説明します。

注: Java Message Service (JMS) の IBM のインプリメンテーションは、MQSeries で構築されています。このインプリメンテーションを使用する場合、MQSeries が正しく動作していることを確認してください。MQSeries に関する問題について詳しくは、25ページの『MQSeries に関する問題』を参照してください。

JNDI の初期設定が失敗した

問題: Java Naming and Directory Interface™ (JNDI) の初期化に失敗しました。

応答: FSContext を使用している場合は、CLASSPATH 環境変数に、fscontext.jar ファイルに対するパスが組み込まれているか確認してください。

応答: FSContext を使用している場合、PROVIDER_URL 環境変数が既存のディレクトリーを正しく指していることを確認してください。

JMS 通信モードが失敗した

問題: JMS 通信モードを使用しようとしたのですが、`java.lang.NoClassDefFoundError` 例外で失敗しました。

応答: JMS 基本パッケージを明示的にインストールしていない場合は、MQSeries SupportPac サイトからダウンロードしてください。MQSeries Adapter Kernel には JMS パッケージは備えられていません。

応答: IBM の MQSeries JMS インプリメンテーションを使用するには、複数の JAR ファイルを CLASSPATH 環境変数に追加する必要があります。(詳細については、製品の資料を参照してください。) 必要な JAR ファイルがすべて CLASSPATH 環境変数で指定されているか確認してください。

QueueConnectionFactory または Queue オブジェクトの検出エラー

問題: QueueConnectionFactory または Queue オブジェクトの検出に関するエラーが、アプリケーションにより報告されています。

応答: QueueConnectionFactory オブジェクトが作成され、格納されているか確認してください。

応答: FSContext を使用している場合は、`aqmconfig.xml` 構成ファイルに、ルート・ディレクトリーの `context` エlementが組み込まれているか確認してください。

HP-UX 上のバインディングの問題

問題: アプリケーションで `libmqjbnd01.so` ライブラリー・ファイルを解決できないので、バインディングが失敗します。

応答: 最新の MQSeries Adapter Offering Corrective Service Diskette (CSD) がインストールされているか確認します。JMS SupportPac もインストールする必要がある場合もあります。

HP-UX 上のコード・ページ変換問題

問題: アプリケーションで、デフォルト・コード・ページの変換に関する問題が検出されています。

応答: HP-UX デフォルトのコード・ページ `en_US.roman8 (1051)` を `en_US.iso88591 (819)` に一時的に変更して、QManagerオブジェクトを再作成してください。

JMS と LDAP を使用して AdapterException が発生

問題: JMS と LDAP を使用すると、カーネルは組み込み NullPointerException とともに AdapterException をスローします。

応答: この問題は、カーネルが LDAP から JMS QueueConnectionFactory を検索しようとする場合に発生することがあります。IBM SecureWay Directory 製品を旧型のマシンで動作している、または動作してきた場合、SecureWay Directory から提供される `ibmjndi.jar` および `jndi.jar` ファイルがクラス・パスにはないことを確認してください。これらの JAR ファイルには、LDAP からオブジェクトを検索する際に必要な機能が組み込まれていません。その代わりに、MQSeries SupportPac MA88 で提供される `ldap.jar` および `jndi.jar` ファイルを使用してください。

その他の問題

UNIX 上での表示エラー

問題: UNIX 上でインストール・プログラムを実行しようとする、表示内容をオープンできないと報告されます。

応答: DISPLAY 環境変数が適切に設定されているか確認してください。必須値は、ホスト・マシンの名前の後にストリング `:0.0` が付いたものになります。 `grizelda` という名前のマシンの場合、有効な値は `grizelda:0.0` です。

応答: + (プラス) 引き数を指定してコマンド `xhost +` を実行し、X Window セキュリティー・プログラム `xhost` を使用してコンソールのアクセス制御をオフにします。このコマンドは、コンソールから実行しなければなりません。リモート側で実行することはできません。

Solaris 上のメモリー・ダンプ

問題: JDK 1.2.2 と MQSeries 使用して メッセージを送受信すると、`*** panic: libthread loaded into green threads Abort (coredump).` が表示されます。

応答: THREADS_FLAG 環境変数の値を `native` に設定してください。たとえば、`export THREADS_FLAG=native` コマンドを入力します。

AIX 上でのセグメント化違反

問題: AIX 上で C 言語のアダプターを実行すると、セグメント化違反 (SIGSEGV) とメモリー・ダンプが起こります。

応答: AIXTHREAD_SCOPE 環境変数の値を `S` に設定してください。たとえば、`export AIXTHREAD_SCOPE=S` コマンドを入力します。

Java メモリーの問題: OutOfMemoryError

問題: アプリケーションで Java から OutOfMemoryError が生成されました。これは、アプリケーションで非常に大きなメッセージが処理されているか、またはアダプターにより大量のメモリーが使用されていることを示すシグナルである可能性があります。

応答: Java で開始されたプロセスでエラーが起きた場合は、始動時に Java のメモリー値を大きくしてください。

応答: C 言語で開始されたプロセスでエラーが起きた場合は、aqmsetup ファイル中でパラメーターを設定して、Java のメモリー値を大きくしてみてください。

XML パーサーの問題

問題: XML データの構文解析でエラーが発生します。

応答: CLASSPATH 環境変数が、正しいバージョンの XML4J パーサーである、バージョン 2_0_15 を指しているか確認してください。

WebSphere の使用

問題: WebSphere Application Server も使用する際の環境問題。

応答: CLASSPATH、PATH、および AQMSETUPFILE のような環境変数は、WebSphere の構成ツール内で設定しなければなりません。

特記事項

本書は米国 IBM 社が提供する製品およびサービスについて作成したものであり、米国以外の国においては本書で述べる製品、サービス、または機能を提供しない場合があります。日本で利用可能な製品、サービス、およびフィーチャーについては、日本 IBM 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。これらのプログラムまたは製品に代えて、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 製以外の製品と組み合わせた場合、その操作の評価と検証については、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。実施権の許諾については、下記の宛先に、書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31

AP 事業所

IBM World Trade Asia Corporation

Intellectual Property Law & Licensing

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を提供し、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。

本書は定期的に見直され、必要な変更 (たとえば、技術的に不適確な表現や誤植など) は、本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。また、IBM 以外の製品に関するパフォーマンスの正確性、互換性、またはその他の要求は確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

商標

次のものは、IBM Corporation の米国およびその他の国における商標です。

AIX	OS/400
AS/400	RISC System/6000
IBM	RS/6000
MQSeries	WebSphere

Lotus および LotusScript は、Lotus Development 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。

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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標または登録商標です。

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。

[カ行]

環境変数

AQMSETUPFILE 23, 31

CLASSPATH 23, 31

PATH 31

構成

トレース 2, 4

構成上の問題

アダプター・デーモン 24, 25

確認通知 23

ソケット・ハンドラー 25

ターゲット・アダプター 23, 24

ターゲット・アダプターの検出 23

トレース 24, 25

トレース・クライアント 24, 25

ネイティブ・アダプター 24

本体データの構文解析 24

メッセージの受信 24, 25

メッセージの送信 24

AIX 30

aqmconfig.xml ファイル 23

aqmsetup ファイル 23

Java メモリー 31

Solaris 30

UNIX の表示 30

WebSphere 31

XML パーサー 31

[タ行]

ターゲット・アダプター

構成上の問題 23, 24

トレース

アプリケーション ID 3

概要 1

構成 2, 4

コンソール・ハンドラー 2

コンポーネント 1

トレース (続き)

出力の宛先 2

ソケット・ハンドラー 2

トレース・レベル 3

ネイティブ・アダプター・ハンドラー 2

ハンドラー 2

ファイル・ハンドラー 2

メッセージの相関 2

ENA ハンドラー 2

トレース・コンポーネント

トレース・ワーカー 1

トレース・クライアント 1

トレース・サーバー 1

ハンドラー 1

A

aqmsetup ファイル

環境変数 23

aqmconfig.xml 参照 23

AQMSETUPFILE

環境変数 23, 31

J

JMS の問題

通信モードが失敗 29

HP-UX 上でのバインディング 29

HP-UX 上のコード・ページ変換 29

JMS と LDAP を使用して AdapterException 30

JNDI の初期設定に失敗 28

Queue オブジェクトの検索 29

QueueConnectionFactory オブジェクトの検索 29

M

MQSeries Integrator の問題

メッセージが理解されない 28

メッセージの受信 28

MQSeries の問題

キューが見つからない 26

キュー項目数の超過 26

キュー・マネージャーが使用できない 27

MQSeries の問題 (続き)

- 共用メモリー・エラー 27
- 受信例外 27
- チャンネルのダウン 26
- メッセージ・カタログをロードできない 26
- ログのスペース超過 26
- JAR が見つからない 26
- MQJI プロパティ 26
- MQSeries クラスが見つからない 28
- ResourceException クラス 27



Printed in Japan

GC88-8888-01



日本アイ・ビー・エム株式会社

〒106-8711 東京都港区六本木3-2-12