



BEA WebLogic Integration™

**B2B Integration 管理
アプリケーション プ
ログラミング ガイド**

著作権

Copyright © 2002, BEA Systems, Inc. All Rights Reserved.

限定的権利条項

本ソフトウェアおよびマニュアルは、BEA Systems, Inc. 又は日本ビー・イー・エー・システムズ株式会社（以下、「BEA」といいます）の使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができ、同契約の条項通りにのみ使用またはコピーすることができます。同契約で明示的に許可されている以外の方法で同ソフトウェアをコピーすることは法律に違反します。このマニュアルの一部または全部を、BEA Systems, Inc. からの書面による事前の同意なしに、複写、複製、翻訳、あるいはいかなる電子媒体または機械可読形式への変換も行うことはできません。

米国政府による使用、複製もしくは開示は、BEA の使用許諾契約、および FAR 52.227-19 の「Commercial Computer Software-Restricted Rights」条項のサブパラグラフ (c)(1)、DFARS 252.227-7013 の「Rights in Technical Data and Computer Software」条項のサブパラグラフ (c)(1)(ii)、NASA FAR 補遺 16-52.227-86 の「Commercial Computer Software--Licensing」条項のサブパラグラフ (d)、もしくはそれらと同等の条項で定める制限の対象となります。

このマニュアルに記載されている内容は予告なく変更されることがあり、また BEA による責務を意味するものではありません。本ソフトウェアおよびマニュアルは「現状のまま」提供され、商品性や特定用途への適合性を始めとする（ただし、これらには限定されない）いかなる種類の保証も与えません。さらに、BEA は、正当性、正確さ、信頼性などについて、本ソフトウェアまたはマニュアルの使用もしくは使用結果に関していかなる確約、保証、あるいは表明も行いません。

商標または登録商標

BEA、Jolt、Tuxedo、および WebLogic は BEA Systems, Inc. の登録商標です。BEA Builder、BEA Campaign Manager for WebLogic、BEA eLink、BEA Manager、BEA WebLogic Commerce Server、BEA WebLogic Enterprise、BEA WebLogic Enterprise Platform、BEA WebLogic Express、BEA WebLogic Integration、BEA WebLogic Personalization Server、BEA WebLogic Platform、BEA WebLogic Server、BEA WebLogic Workshop および How Business Becomes E-Business は、BEA Systems, Inc の商標です。

その他の商標はすべて、関係各社が著作権を有します。

BEA Integration 管理アプリケーション プログラミング ガイド

パート番号	日付	ソフトウェアのバージョン
なし	2002年6月	7.0

目次

このマニュアルの内容

対象読者	v
e-docs Web サイト	vi
このマニュアルの印刷方法	vi
関連情報	vi
サポート情報	vii
表記規則	vii

1. はじめに

WebLogic Integration アプリケーション	1-1
管理アプリケーション	1-2

2. 管理アプリケーションの開発

管理ツールの操作	2-1
MBean と MBeanServer の操作	2-2
MBean パッケージ	2-2
MBean	2-3
MBean サーバの実装	2-4
管理アプリケーションのプログラミング	2-4
手順 1 : 必要なパッケージをインポートする	2-6
手順 2 : MBeanServer オブジェクトへの参照を取得する	2-7
手順 3 : ObjectName オブジェクトを作成する	2-8
手順 4 : MBean サーバを照会する	2-8
手順 5 : MBean の属性を読み取る	2-9
手順 6 : MBean を検索する	2-10
手順 7 : 例外を処理する	2-10

索引



このマニュアルの内容

このマニュアルでは、BEA WebLogic Integrationtm B2B Integration システムの実行時アクティビティをモニタするアプリケーションの開発方法について説明します。

このマニュアルの内容は以下のとおりです。

- 第1章「はじめに」では、WebLogic Integration 環境のためのアプリケーションの開発について概説します。
- 第2章「管理アプリケーションの開発」では、WebLogic Integration Managed Beans (MBeans) を使用して実行時の B2B Integration アクティビティをモニタするアプリケーションを作成する方法について説明します。

対象読者

このマニュアルは主に、次のユーザを対象としています。

- WebLogic Integration Studio を使用して B2B Integration 環境で使用できるワークフローを設計するビジネス プロセス設計者
- B2B Integration 環境における実行時の統計をモニタする Java アプリケーションを記述するアプリケーション開発者
- B2B Integration アプリケーションを設定および管理するシステム管理者

B2B Integration のアーキテクチャの概要については、『*B2B Integration 入門*』の「概要」を参照してください。

e-docs Web サイト

BEA の製品マニュアルは、以下の場所から入手できます。

<http://edocs.beasys.co.jp/e-docs/index.html>

このマニュアルの印刷方法

このマニュアルの PDF 版は、Web サイトで入手できます。PDF を Adobe Acrobat Reader で開くと、マニュアルの全体または一部分を印刷できます。Adobe Acrobat Reader がない場合は、次の URL にある Adobe の Web サイトから無料で入手できます。

<http://www.adobe.co.jp/>

または、Web ブラウザの [ファイル | 印刷] オプションを使用すると、Web ブラウザからこのマニュアルの HTML ファイルを一度に 1 ファイルずつ印刷できます。

関連情報

Java 2 Enterprise Edition (J2EE)、eXtensible Markup Language (XML)、および Java プログラミングについての詳細は、次の URL にある Javasoft Web サイトを参照してください。

<http://java.sun.com>

サポート情報

WebLogic Integration のドキュメントに関するユーザからのフィードバックは弊社にとって非常に重要です。質問や意見などがあれば、電子メールで docsupport-jp@bea.com までお送りください。寄せられた意見については、WebLogic Integration のドキュメントを作成および改訂する BEA の専門の担当者が直に目を通します。

電子メールのメッセージには、BEA WebLogic Integration リリース 7.0 のドキュメントをご使用の旨をお書き添えください。

本リリースの WebLogic Integration について不明な点がある場合、または WebLogic Integration のインストールおよび動作に問題がある場合は、次の URL にある BEA WebSUPPORT を通じて BEA カスタマサポートまでお問い合わせください。

<http://websupport.bea.com/custsupp>

カスタマサポートでは以下の情報をお尋ねしますので、お問い合わせの際はあらかじめご用意ください。

- お名前、電子メール アドレス、電話番号、ファクス番号
- 会社の名前と住所
- お使いの機種とコード番号
- 製品の名前とバージョン
- 問題の状況と表示されるエラー メッセージの内容

表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
太字	用語集で定義されている用語を示す。

表記法	適用
[Ctrl] + [Tab]	複数のキーを同時に押すことを示す。
<i>斜体</i>	強調または書籍のタイトルを示す。
等幅テキスト	コード サンプル、コマンドとそのオプション、データ構造体とそのメンバー、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。 <i>例</i> #include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float
太字の等幅テキスト	コード内の重要な箇所を示す。 <i>例</i> void commit ()
<i>斜体の等幅テキスト</i>	コード内の変数を示す。 <i>例</i> String expr
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 <i>例</i> LPT1 SIGNON OR
{ }	構文の中で複数の選択肢を示す。実際には、この括弧は入力しない。
[]	構文の中で任意指定の項目を示す。実際には、この括弧は入力しない。 <i>例</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
	構文の中で相互に排他的な選択肢を区切る。実際には、この記号は入力しない。

表記法	適用
...	<p>コマンドラインで以下のいずれかを示す。</p> <ul style="list-style-type: none"> ■ 引数を複数回繰り返すことができる ■ 任意指定の引数が省略されている ■ パラメータや値などの情報を追加入力できる <p>実際には、この省略記号は入力しない。</p> <p><i>例</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
.	<p>コード サンプルまたは構文で項目が省略されていることを示す。実際には、この省略記号は入力しない。</p>



1 はじめに

以下の節では、**B2B Integration** 管理アプリケーションの概要を説明します。

- **WebLogic Integration** アプリケーション
- 管理アプリケーション

WebLogic Integration アプリケーション

ここでは、**WebLogic Integration** アプリケーションに関する以下のトピックを取り上げます。

- **MBean** を使用した **B2B Integration** のアクティビティのモニタ
- 管理アプリケーション

MBean とは、**WebLogic Integration** で使用できる 3 種類のコンポーネント アプリケーションうちの 1 つです。**WebLogic Integration** では、**MBean** に加え、次の 2 つのアプリケーションが使用できます。

- **ロジック プラグイン**。カスタマイズされたルーティング、フィルタ処理、情報処理に使用します。詳細については、『***B2B Integration** ロジック プラグイン プログラミング ガイド*』を参照してください。
- **メッセージング アプリケーション**。詳細については、『***B2B Integration** メッセージング アプリケーション プログラミング ガイド*』を参照してください。

B2B Integration システムの概要については、『***B2B Integration** 入門*』を参照してください。

管理アプリケーション

WebLogic Integration 管理アプリケーションを使用すると、B2B Integration をサポートする実行時アクティビティ（メッセージトラフィック、会話統計など）をモニタできます。WebLogic Integration では、管理ツールが WebLogic Integration B2B Console として統合されており、実行時に B2B エンジンモニタできます。BEA 提供のシステム ツールに加え、同等のモニタ機能をもつカスタム管理アプリケーションを開発することもできます。

たとえば、以下のような用途の管理アプリケーションを開発できます。

- WebLogic Server インスタンスのアクティビティのモニタ
- サーバ インスタンス、配信チャネル、ビジネストランザクション定義、トレーディング パートナ、ビジネス メッセージなどの実行時統計の取得

WebLogic Integration では、すべての管理アプリケーションに対して Managed Bean (*MBeans*) が提供されます。MBean は、管理操作のための属性およびメソッドを備えた特別な JavaBean で、Java Management Extensions (JMX) Managed Beans API の BEA の実装です。JMX Managed Beans API は、Sun Microsystems, Inc. 発行の Java Management Extensions 仕様に定義されています。

2 管理アプリケーションの開発

以下の節では、**WebLogic Integration B2B** エンジンの実行時アクティビティのモニタに使用する管理アプリケーションの作成方法について説明します。

- 管理ツールの操作
- MBean と MBeanServer の操作
- 管理アプリケーションのプログラミング

管理ツールの操作

WebLogic Integration B2B Console を使用すると、**B2B Integration** のアクティビティを実行時にモニタできます。また、必要に応じてカスタム管理アプリケーションを作成し、**B2B Console** と同等の情報をモニタすることもできます。

カスタム管理アプリケーションは、以下のような用途に使用できます。

- 特定の会話内で交換されたメッセージの数、**B2B Integration** で受信したメッセージの数などのリアルタイムの統計への読み取り専用アクセス
- 特定の配信チャネルや **WebLogic Server** のシャットダウン、特定の会話からの退出、会話の終了などの管理タスク

注意： **WebLogic Integration** リポジトリは、カスタム管理アプリケーションでは更新できません。リポジトリは、次のいずれかのツールを使用して更新します。

- **WebLogic Integration B2B Console**。詳細については、『**B2B Integration Administration Console** オンラインヘルプ』を参照してください。
- **Bulk Loader** については、『**B2B Integration 管理ガイド**』の「**Bulk Loader** の操作」を参照してください。

MBean と MBeanServer の操作

WebLogic Integration では、B2B エンジンの実行時アクティビティをモニタするカスタム管理アプリケーションの作成に必要なアプリケーション プログラミング インタフェース (API) が用意されています。B2B Console でもこれらの API を使用してリアルタイムのモニタ情報を表示しています。

これらの API は、管理操作のための属性およびメソッドを備えた特別な JavaBean である Java Management Extensions (JMX) Managed Bean (MBean) のセットで構成されています。JMX、特に JMX API (MBeanServer および MBean を含む) の詳細については、Sun Microsystems, Inc. 発行の Java Management Extensions 仕様 (下記 URL) を参照してください。

<http://www.java.sun.com/products/JavaManagement/index.html>

MBean パッケージ

WebLogic Integration では、カスタム管理アプリケーションの作成に使用する `com.bea.b2b.management` パッケージが提供されています。このパッケージでは以下が提供されます。

- B2B エンジンの実行時アクティビティのモニタを可能にする MBean
- 実行時管理アプリケーションで発生するエラーを処理するための `ManagementException` クラス

このパッケージの詳細については、『*BEA WebLogic Integration Javadoc*』を参照してください。

このリリースでは、すべての MBean が標準 MBean として実装されます。これらの標準 MBean によって、独自の MBean インタフェースを実装するクラスが構成されます。WebLogic Integration MBean は、リモート MBean としては実装されません。したがって、すべての管理アプリケーションを、モニタする B2B エンジンに常駐させる必要があります。

MBean

次の表では、各 WebLogic Integration MBean について説明します。

表 2-1 WebLogic Integration MBean

ラベル	説明
WLCMBean	B2B Integration のインスタンスを表す。実行時のインスタンスのモニタに使用する。
DeliveryChannelMBean	配信チャネルを表す。実行時の B2B Integration システム上の配信チャネルのモニタに使用する。
ConversationMBean	B2B Integration インスタンス上のトランザクション マネージャによって管理されるビジネス会話を表す。配信チャネル内のアクティブなトランザクションのモニタに使用する。
TradingPartnerSessionMBean	トレーディング パートナのセッションを表す。トレーディング パートナのモニタに使用する。
MessageMBean	会話内のメッセージを表す。メッセージのモニタに使用する。
CollaborationAgreementMBean	コラボレーション アグリーメントを表す。コラボレーション アグリーメントは、どのプロトコルを使用してどのように通信するかについての 2 パーティ間の技術的なアグリーメントを表す。

注意： WebLogic Integration リリース 2.1、7.0 および BEA WebLogic Collaborate 2.0 では、すべての MBean がこのパッケージに集められています。WebLogic Collaborate の旧バージョンでは、MBean は c-hub パッケージと c-enabler パッケージに分けられていました。WebLogic Collaborate 1.0 または 1.0.1 からアップグレードする場合は、以前に記述した管理アプリケーションを、新しい MBean を使用するように変更する必要があります。次の表に、リリース 1.x MBean からリリース 2.x MBean への変更点をまとめます。

表 2-2 MBean ラベル

リリース 1.x のラベル	リリース 2.x のラベル
EnablerMBean	WLCMBean
HubMBean	WLCMBean
CSpaceMBean	DeliveryChannelMBean
ConversationMBean	ConversationMBean
GlobalConversationMBean	ConversationMBean
EnablerSessionMBean	TradingPartnerSessionMBean
CollaboratorMBean	TradingPartnerSessionMBean
MessageMBean	MessageMBean

MBean サーバの実装

MBean をアプリケーションで使用する場合、WebLogic Integration では、WebLogic Server インスタンスの開始時に作成された MBean Server インスタンスに MBean が登録されます。WebLogic Server MBean Server の詳細については、次の URL の『*WebLogic Server JMX Services プログラマーズガイド*』を参照してください。

<http://edocs.beasys.co.jp/e-docs/wls/docs70/jms/index.html>

管理アプリケーションのプログラミング

JMX API を使用して WebLogic Integration MBeans にアクセスするには、Java アプリケーションで以下の手順を実行する必要があります。

- 手順 1: 必要なパッケージをインポートする
- 手順 2: MBeanServer オブジェクトへの参照を取得する

- 手順 3 : ObjectName オブジェクトを作成する
- 手順 4 : MBean サーバを照会する
- 手順 5 : MBean の属性を読み取る
- 手順 6 : MBean を検索する
- 手順 7 : 例外を処理する

情報を検索する場合は、クラスタ内の全ノードを通る **app** が必要です。ノードはそれぞれ、スタンドアロン ノードとして機能しなければなりません。上記の手順を繰り返してください。

注意： 管理アプリケーションをマルチノード クラスタにデプロイする場合は、全てのノードを個別に通るロジックを構築する必要があります。管理アプリケーションでは、マシンがそれぞれスタンドアロン ノードとして機能します。

手順 1 : 必要なパッケージをインポートする

MBean を操作するには、必要なパッケージを管理アプリケーションにインポートする必要があります。最低でも、次の表に示すパッケージをインポートしなければなりません。

表 2-3 インポートが必要なパッケージ

ラベル	説明
<code>javax.management.*;</code>	Sun Microsystems, Inc. 発行の Java Management Extensions 仕様で必須であるため、JMX MBean で必要。
<code>javax.naming.*;</code>	JNDI ルックアップを使用して MBean サーバ オブジェクトを検索するために必要。以下のパッケージのみ必要。 <ul style="list-style-type: none">■ <code>javax.naming.Context</code>■ <code>javax.naming.InitialContext</code>
<code>com.bea.b2b.management.*</code>	すべての管理アプリケーションで必要。
<code>weblogic.management.*</code>	すべての管理実装で必要。MBeanServer および MBeanHome を取得する。

次のリストのコードでは、管理アプリケーションで必要なパッケージをインポートします。

コード リスト 2-1 管理アプリケーション用のパッケージのインポート

```
import javax.management.*;
import javax.naming.Context;
import javax.naming.InitialContext;
import com.bea.b2b.management.*;
import weblogic.management.*;
```

手順 2 : MBeanServer オブジェクトへの参照を取得する

WebLogic Integration では、WebLogic Server のインスタンスの起動時にインスタンス化された MBeanServer を使用します。MBeanServer への参照を取得するには、そのサーバの MBeanHome が必要です。MBeanServer の MBeanHome は、次に示すサーバの JNDI ツリーから取得できます。

```
weblogic.management.MBeanHome.JNDI_NAME.serverName
```

管理サーバは、JNDI ツリー上のドメイン内にある各サーバに対して MBeanHome をパブリッシュします。管理 MBeanHome は、次に示す管理サーバの JNDI ツリーから取得できます。

```
weblogic.management.MBeanHome.ADMIN_JNDI_NAME
```

すべての MBeanHome の基盤となる MBeanServer は、その MBeanHome の `getMBeanServer()` メソッドを呼び出すことによって取得できます。

次のコードは、管理サーバ MBeanHome 用の JNDI ルックアップの例です。

コード リスト 2-2 MBeanServer オブジェクトへの参照の取得

```
import javax.naming.Context;
import javax.naming.NamingException;
import javax.naming.AuthenticationException;
import javax.naming.CommunicationException;
import weblogic.jndi.Environment;
import weblogic.management.MBeanHome;

...
MbeanHome home = null;
try {
    Environment env = new Environment();
    ctx = env.getInitialContext();
    home = (MBeanHome) ctx.lookup(MBeanHome.ADMIN_JNDI_NAME);
    RemoteMBeanServer server = home.getMBeanServer();
}
catch (AuthenticationException e) {
    ... // エラー処理
}
catch (CommunicationException e) {
    ... // エラー処理
}
catch (NamingException e) {
    ... // エラー処理
}
```

手順 3 : ObjectName オブジェクトを作成する

MBeans は、MBeanServer 内では一意のオブジェクト名で識別されます。ObjectName クラスがオブジェクト名を表します。

WebLogic Integration リリース 2.1、7.0 および WebLogic Collaborate 2.0 では、MBean Server に登録されるのは WLCMBean だけです。他の MBean はすべて、WLCMBean から検索されます。WLCMBean の属性は、名前、タイプ、およびドメインの 3 つです。これらの属性は、MBean の JMX Object Name に反映されます。Object Name は、所定の MBean に対する、すべてのドメインでユニークな識別子で、以下のような構造を持っています。

```
domain name:Name=name,Type=type[,attr=value]...
```

name の値は、指定したドメインとタイプにおいてユニークである必要があります。たとえば、

```
mydomain:Name=WLC,Type=WLC  
ObjectName objectName = new ObjectName("WLC", "WLC", "mydomain");
```

MBean の場合、オブジェクト名式が使用されているクエリ操作にオブジェクト名を使用することもできます。MBeanServer では、登録されている MBean のオブジェクト名でパターン マッチングが行われます。マッチング構文は、Sun Microsystems, Inc. 発行の Java Management Extensions 仕様で示されているファイルグロービングに準拠しています。

- アスタリスク (*) は、どの文字シーケンスにもマッチします。
- 疑問符 (?) は、1 文字にマッチします。

手順 4 : MBean サーバを照会する

オブジェクト名式を構築したら、その式に対応する ObjectName オブジェクトを渡して MBeanServer を照会します。登録された MBean のセット (オブジェクト名式を満たす名前) を検索するには、次のメソッドを使用します。

```
javax.management.MBeanServer.queryNames()
```

クエリ条件を満たすオブジェクトのセットが返されます。これらは、MBean を表す ObjectName オブジェクトであり、MBean 自体への直接の参照ではないことに注意してください。

手順 5 : MBean の属性を読み取る

他の MBean にアクセスするには、前のステップで取得した ObjectName インスタンスを使用します。ただし、ObjectName には、1 つまたは複数の MBean タイプの属性が必要です。MBean の属性を読み取るには、次のメソッドを使用します。パラメータとして ObjectName オブジェクトを渡します。

```
javax.management.MBeanServer.getAttribute()
```

最初の MBean の ObjectName を渡して getAttribute メソッドを呼び出した後は、他の MBean インスタンスへの直接の参照を取得できます。

次のコードでは、WLCMBean に関連付けられた属性のセットを検索していません。

コード リスト 2-3 会話属性の検索

```
MBeanHome home = Admin.getMBeanHome();
server = home.getMBeanServer();
ObjectName objectName = new ObjectName("WLC", "WLC", "mydomain");
beans = server.queryNames(objectName, null);
Iterator it = beans.iterator();
while (it != null && it.hasNext())
{
    // 1 つのみ
    obj = (ObjectName)it.next();
    break;
}
if (obj != null)
{
    Date startTime = (Date)server.getAttribute(obj, "ActiveSince");
    Date lastTime = (Date)server.getAttribute(obj, "LastMessageSentTime");
    ConversationMBean[ ] convs = (Conversation[ ]) server.getAttribute(obj,
        "ActiveConversations");
    if (convs != null)
    {
        for (int ii=0; ii< convs.length; ii++)
        {
            String protocol = convs[ii].getBusinessProtocolName();
        }
    }
}
...

```

上記のコードで示されているすべての属性は、`getAttribute` を呼び出すことによって検索できます。`WLCMBean` の `shutDown` などのメソッドを呼び出すには、`MBeanServer` を呼び出します。詳細については、次の URL の JMX 仕様を参照してください。

<http://java.sun.com>

手順 6 : MBean を検索する

論理的に関連する MBean には、相互に参照を検索するためのアクセサ メソッドが用意されています。これらのメソッドは強く型付けされており、正確な MBean タイプが返されます。たとえば、

`WLCMBean.getActiveDeliveryChannels()` メソッドでは、システム内のすべてのアクティブな配信チャネルを表す `DeliveryChannelMBean` タイプの配列が返されます。同様に、

`TradingPartnerSessionMBean.getActiveConversations()` メソッドでは、セッション内のすべてのアクティブな会話を表す `ConversationMBean` タイプの配列が返されます。

これらのメソッドの詳細については、『*BEA WebLogic Integration Javadoc*』を参照してください。

手順 7 : 例外を処理する

B2B Integration 管理アプリケーションの実行時にエラーが発生すると、`com.bea.b2b.management.ManagementException` が送出されます。この例外は、管理アプリケーションによって捕捉されて適切に処理されます。以下に例を示します。

コード リスト 2-4 管理アプリケーションによる ManagementExceptions の処理

```
catch (ManagementException me){
    String msg = "Exception in Management Application: " + me;
    debug(msg);
    throw new Exception(msg);
}
```

索引

B

B2B Integration、モニタ 2-1

C

CollaborationAgreementMBean 2-3
CollaboratorMBean 2-4
ConversationMBean 2-3, 2-4
CSPACEMBean 2-4

D

DeliveryChannelMBean 2-3, 2-4

E

EnablerMBean 2-4
EnablerSessionMBean 2-4

G

GlobalConversationMBean 2-4

H

HubMBean 2-4

J

Java Management Extensions 2-2
JMX 2-2
JNDI ルックアップ、例 2-7

M

Managed Bean 2-2
MBean

CollaborationAgreementMBean 2-3
ConversationMBean 2-3, 2-4
CSPACEMBean 2-4
DeliveryChannelMBean 2-3, 2-4
EnablerMBean 2-4
HubMBean 2-4
TradingPartnerSessionMBean 2-3
WLCMBean 2-3, 2-4
概要 2-3
定義 1-2, 2-2
パッケージ 2-2

MBeann

CollaboratorMBean 2-4
ConversationMBean 2-4
EnablerSessionMBean 2-4
GlobalConversationMBean 2-4
TradingPartnerSessionMBean 2-4
検索 2-10
属性の読み取り 2-9

MBean サーバ

概要 2-2
参照の取得 2-7
実装 2-4
照会 2-8

MBean サーバの照会 2-8

MBean サーバへの参照の取得 2-7

MBean の検索 2-10

MBean の属性の読み取り 2-9

MessageMBean 2-3, 2-4

T

TradingPartnerSessionMBean 2-3, 2-4

W

WLCMBean 2-3, 2-4

あ

アプリケーション、概要 1-1

お

オブジェクト、作成 2-8

オブジェクト名式 2-8

オブジェクト名の式 2-8

か

カスタマ vii

カスタマ サポート vii

管理アプリケーション

概要 1-2, 2-2

管理ツール 2-1

関連情報 vi

け

検索、例 2-9

こ

コンフィグレーション、リポジトリ 2-1

さ

作成、オブジェクト 2-8

サポート vii

参照

MBean サーバ 2-7

オブジェクト 2-7

し

実装、MBean サーバ 2-4

そ

属性

読み取り 2-9

例、検索 2-9

て

定義、MBean 1-2, 2-2

テクニカル vii

テクニカル サポート vii

は

パッケージ

MBean 2-2

インポート 2-6

パッケージのインポート 2-6

ひ

表記規則 vii

ふ

プログラミング ステップ 2-4

ま

マニュアル

印刷 vi

マニュアル入手先 vi

マニュアルの印刷 vi

も

モニタ、B2B Integration 2-1

り

リポジトリ、コンフィグレーション 2-1

れ

例

JNDI ルックアップ 2-7

参照の取得 2-7

属性の検索 2-9

例外、処理 2-10

例外処理 2-10

連絡先 vii

