

**IBM WebSphere Business Integration
Adapters**



**WebSphere Message Brokers 使用
アダプター・インプリメンテーション・ガイド**

**IBM WebSphere Business Integration
Adapters**



**WebSphere Message Brokers 使用
アダプター・インプリメンテーション・ガイド**

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、211 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、IBM WebSphere Business Integration Adapter Framework バージョン 2.4、および新しい版で明記されていない限り、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： IBM WebSphere Business Integration Adapters
Implementing Adapters with WebSphere Message Brokers

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2刷 2004.3

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

© Copyright International Business Machines Corporation 2002, 2004. All rights reserved.

© Copyright IBM Japan 2004

目次

本書について	vii
対象読者	vii
関連資料	vii
WebSphere Business Integration Adapters の資料	viii
WebSphere Message Broker の資料	viii
System Manager の資料	viii
表記上の規則	ix

改訂の要約	xi
WebSphere Business Integration Adapter Framework バージョン 2.4.0 の新機能	xi
2004 年 2 月	xi
2003 年 12 月	xi
WebSphere Business Integration Adapters v 2.3.1 での変更点	xi
WebSphere Business Integration Adapters v 2.2.0 での変更点	xii
WebSphere Business Integration Adapters v 2.1.0 での変更点	xiii
WebSphere Business Integration Adapters v 2.0.1 での変更点	xiii

第 1 部 概要と概念 1

第 1 章 WebSphere Business Integration Adapters の概要 3

必要な資料についての注意	4
WebSphere Business Integration システムとは?	4
WebSphere Message Broker とは?	5
WebSphere Business Integration Adapters とは?	5
WebSphere Business Integration システムの機能	6
ビジネス・インテグレーション・システムのデータ・フロー	8
ビジネス・インテグレーション処理の要約	12

第 2 章 ビジネス・オブジェクト 15

ビジネス・オブジェクトの役割	15
イベント	15
要求	16
応答	16
ビジネス・オブジェクトの構造	16
ビジネス・オブジェクト・タイプ	16
ビジネス・オブジェクト動詞	17
ビジネス・オブジェクト属性値	17
ビジネス・オブジェクトのタイプ	18
ビジネス・オブジェクト定義とビジネス・オブジェクト	19
ビジネス・オブジェクト定義のコンポーネント	20
動詞	22
ビジネス・オブジェクトの詳細	22

属性の編成	22
アプリケーション固有情報	23
ビジネス・オブジェクト定義を作成または変更する方法	25
ビジネス・オブジェクト定義の作成	25
ビジネス・オブジェクト定義の変更	26

第 3 章 コネクタ 27

コネクタの開始	27
イベント通知	28
アプリケーションのイベント通知メカニズムのセットアップ	28
イベントの検出	32
イベントの処理	33
保証付きイベント・デリバリー	34
要求処理	35
動詞ベースの処理	36
ビジネス・オブジェクトの構成とデコンストラクション	37
動詞に関するアプリケーション固有情報	39
コネクタ構成	40
コネクタの開発	41

第 4 章 データ移送および統合ブローカー 43

統合ブローカーの役割	43
非同期データ移送	44
同期データ移送	44
メッセージ交換のインターフェース	44
メッセージ・フォーマット	45
メッセージ・キュー	46
詳細	47

第 2 部 展開と管理 49

第 5 章 インプリメンテーションの計画 51

ビジネス・プロセス・インターフェースの開発	51
インプリメンテーションの段階	52
ビジネス・ゴールの発見および評価	52
既存のコンポーネントの評価と新規コンポーネントの設計	54
ビジネス・インテグレーション・システムの開発および構成	54
ビジネス・インテグレーション・システムの検証	55
ビジネス・インテグレーション・システムの展開	56
開発ツール	56

第 6 章 WebSphere Business Integration Adapters のインストール . . 59

Windows システムへのインストール	59
ソフトウェア要件	59

JDK のインストール	60
WebSphere MQ のインストール	60
WebSphere Business Integration Adapters のインストール	61
UNIX システムへのインストール	61
ソフトウェア要件	61
JDK のインストール	62
WebSphere MQ のインストール	64
WebSphere Business Integration Adapters のインストール	64

第 7 章 ビジネス・インテグレーション・システムの管理 67

コネクタの始動	67
Windows の場合	67
UNIX の場合	68
コネクタの停止	69
コネクタ・システムから	69
ブローカー・システムから	69
複数のコネクタ・インスタンスの作成	70
新規ディレクトリーの作成	71
Adapter Monitor および障害キュー・マネージャーの使用	72
Adapter Monitor パースペクティブ	72
Adapter Monitor の設定の変更	73
アダプターのロード	73
Adapter Monitor の表示	74
アダプターの状態の変更	75
障害キュー・マネージャー・ビューの使用	77
失敗イベントの処理	77
WebSphere MQ キューからのメッセージのクリア	78
ログ・ファイルおよびトレース・ファイルの管理	78
ログ・ファイルおよびトレース・ファイルのアーカイブ・ロギング	79
その他のファイルの管理	80
Log Viewer を使用してコネクタ・メッセージを表示する方法	81
Log Viewer の設定の指定	82
メッセージの表示法の変更	85
Log Viewer 表示出力の制御	87
メッセージのフィルター操作	87

第 8 章 WebSphere Business Integration システムの構成 89

構成タスクの概要	89
コネクタと連動するようにメッセージ・ブローカーを構成する方法	89
WebSphere MQ キューの構成	90
キュー構成の定義	92
ビジネス・オブジェクト定義の作成	93
メッセージ・ブローカー・プロジェクトの作成	94
インポーター・パスおよびワークスペース・パスの指定	95
ユーザー・プロジェクトの新規作成	97
メッセージ・ブローカー・ワークスペースへの配置	99

インテグレーター・ブローカーへの配置	102
XML ネーム・スペースの長さの選択	105
アプリケーションでコネクタを使用可能にする	105
コネクタの構成	106
コネクタのローカル・リポジトリのロケーションの指定	107
コネクタが使用するキューの指定	107
キュー・マネージャーによる接続モードの設定	107
同期実行のための構成プロパティの設定	108
ロギングおよびトレース・オプションの構成	108
コネクタの始動ファイル、ショートカット、および環境変数の構成	111
メッセージ・フローの定義	112
トランザクション管理	113
Visual Test Connector を使用したインターフェースの検証	113

付録 A. WebSphere MQ のメッセージ・フォーマット 115

メッセージ・ディスクリプター	115
メッセージ・ヘッダー	115
メッセージ本文	115

付録 B. 管理メッセージの WebSphere MQ メッセージ本文形式 123

コネクタ・フレームワークから WebSphere Message Broker へのメッセージ	123
WebSphere Message Broker からコネクタ・フレームワークへのメッセージ	123

付録 C. コネクタの標準構成プロパティ 125

新規プロパティと削除されたプロパティ	125
標準コネクタ・プロパティの構成	125
Connector Configurator の使用	126
プロパティ値の設定と更新	126
標準プロパティの要約	127
標準構成プロパティ	130
AdminInQueue	130
AdminOutQueue	130
AgentConnections	131
AgentTraceLevel	131
ApplicationName	131
BrokerType	131
CharacterEncoding	131
ConcurrentEventTriggeredFlows	131
ContainerManagedEvents	132
ControllerStoreAndForwardMode	133
ControllerTraceLevel	133
DeliveryQueue	133
DeliveryTransport	133
DuplicateEventElimination	135
FaultQueue	135
JvmMaxHeapSize	135
JvmMaxNativeStackSize	135

JvmMinHeapSize	136
jms.FactoryClassName	136
jms.MessageBrokerName	136
jms.NumConcurrentRequests	136
jms.Password	136
jms.UserName	136
ListenerConcurrency	136
Locale	137
LogAtInterchangeEnd	137
MaxEventCapacity	137
MessageFileName	138
MonitorQueue	138
OADAutoRestartAgent	138
OADMaxNumRetry	138
OADRetryTimeInterval	138
PollEndTime	139
PollFrequency	139
PollQuantity	139
PollStartTime	139
RequestQueue	139
RepositoryDirectory	140
ResponseQueue	140
RestartRetryCount	140
RestartRetryInterval	140
RHF2MessageDomain	140
SourceQueue	141
SynchronousRequestQueue	141
SynchronousResponseQueue	141
SynchronousRequestTimeout	141
WireFormat	142
WsifSynchronousRequest Timeout	142
XMLNamespaceFormat	142

付録 D. コネクター始動オプション . . . 143

Windows	143
UNIX	144

付録 E. System Manager と Eclipse

Workbench 147

System Manager とは	147
Eclipse プラットフォームとは	147
WSWB および WSADIE とは	149
System Manager とは	150
System Manager の使用	150
System Manager の始動	150
System Manager インターフェース	150
統合コンポーネント・ライブラリーでの作業	154
統合コンポーネント・ライブラリーの作成	154
ユーザー・プロジェクトでの作業	156
ユーザー・プロジェクト用に統合ブローカー設定を構成する	156
ユーザー・プロジェクトの作成	156
ユーザー・プロジェクトへのショートカットの追加	158
統合コンポーネント・ライブラリー内のコンポーネントの処理	161

Designer ツールの起動	161
新規コンポーネントの作成	162
既存のコンポーネントの変更	162
パッケージからライブラリーへのコンポーネントのインポート	163
ソリューションの処理	165
ソリューションのエクスポート	165
ソリューションのインポート	167
System Manager を使用したパッケージへのコンポーネントのエクスポート	168
依存関係および参照	170
依存関係および参照の表示	171
複数のワークベンチ・リソースで使用可能な標準の操作	171
ソース・コード制御からワークスペースへのプロジェクトの追加	171
リソースの切り取り、コピー、および貼り付け	172
リソースの最新表示	172
リソースの削除	172
Eclipse ベースのワークベンチの使用	173
パースペクティブのオープンおよびクローズ	173
ビューの表示およびクローズ	175
パースペクティブのカスタマイズ	176
パースペクティブの保管	176
デフォルトでのパースペクティブの設定	177
System Manager の設定の構成	177
System Manager で統合ブローカーに接続する際の問題のトラブルシューティング	179

付録 F. コネクター・スクリプト生成ツールの使用 181

付録 G. Visual Test Connector の使用 183

推奨されるテスト手順	183
Test Connector の始動	185
Test Connector のシャットダウン	186
コネクター・プロファイルの作成と編集	186
ファイルへのコネクター定義の保管	186
新規プロファイルの作成	186
プロファイルの編集	188
プロファイルの削除	188
コネクターのエミュレート	188
ビジネス・オブジェクトの処理	189
要求ビジネス・オブジェクトの処理	189
ビジネス・オブジェクト属性の値の設定	193
ビジネス・オブジェクトの保管	194
ビジネス・オブジェクトのロード	195
ビジネス・オブジェクトの削除	195
要求ビジネス・オブジェクトの受け入れ	195
応答ビジネス・オブジェクトの処理	196
オブジェクト・オブジェクト・インスタンスの比較	197

付録 H. WebSphere Business Integration Adapters のアップグレード 199

前提事項	199
WebSphere Business Integration Adapters のインストール	199
WebSphere Message Broker のアップグレード	200
WebSphere MQ Integrator Broker	200
WebSphere MQ Integrator	200
WebSphere Business Integration Message Broker	200
WebSphere MQ 用 Service Pack のインストール	200
応答キューの作成	200
ローカル・キュー・マネージャーのコード化文字セット ID 値の更新	201
既存のアダプターのアップグレード	201
ビジネス・オブジェクトのアップグレード	201

コネクター構成ファイルの更新	202
新しいコネクター始動スクリプトの構成	204
新規アダプターの構成	204
新規 WebSphere MQ バッチ・ファイルのカスタマイズ	204

用語集 207

索引 209

特記事項. 211

プログラミング・インターフェース情報	212
商標	213

本書について

IBM(R) WebSphere (R) Business Integration Adapter ポートフォリオは、主要な e-business テクノロジー、エンタープライズ・アプリケーション、およびレガシー・システムとメインフレーム・システムに、統合接続性を提供します。製品セットには、ビジネス・プロセスの統合に向けてコンポーネントをカスタマイズ、作成、および管理するためのツールとテンプレートが含まれています。

本書では、WebSphere Business Integration Adapters を、WebSphere Business Integration のいずれかのサポートされるメッセージ・ブローカーと共にインストール、構成、展開、および管理する方法について説明します。サポートされるメッセージ・ブローカーは、WebSphere MQ Integrator 2.1、WebSphere MQ Integrator Broker 2.1、および WebSphere Business Integration Message Broker 5.0 です。

注意:

1. 本書において「アプリケーション」とは、IBM のパートナーが開発して販売するか、または IBM のお客様が開発して使用するエンタープライズ・ソフトウェア・プロダクトを指します。アプリケーションは、IBM WebSphere Integration Adapters ソリューションの中で機能しますが、IBM WebSphere Integration Adapters 製品に属するものではありません。
2. 本書で使用する図は、構造と概念を表すことを目的とした例にすぎず、特定のビジネス・インテグレーションのシナリオを記述したものではありません。

対象読者

本書は、WebSphere Message Broker をアダプターの統合ブローカーとして使用して、WebSphere Business Integration Adapters をインプリメントまたは管理しようとしているカスタマーおよびコンサルタントを対象としています。また、読者が、使用するメッセージ・ブローカーの構成方法と管理方法に関する知識をすでに持っており、WebSphere MQ のメッセージングとメッセージ・フローを十分に理解していることを前提としています。

別の統合ブローカーを使用する WebSphere Business Integration Adapters にすでに使い慣れている場合でも、その統合ブローカー環境と WebSphere Message Broker 環境の機能性および動作には相当な違いがあるため、本書を注意して読まれることをお勧めします。

関連資料

WebSphere Message Broker を統合ブローカーとして使用する WebSphere Business Integration Adapters を展開し管理するときには、以下に挙げる複数の異なる製品のライブラリーの資料が必要になることがあります。

- 個々の WebSphere Business Integration Adapter
- WebSphere MQ Integrator Broker、WebSphere MQ Integrator、または WebSphere Business Integration Message Broker のいずれか

- WebSphere MQ

関連資料に関する情報と入手方法を以下に示します。

WebSphere Business Integration Adapters の資料

この製品に付属する資料のとして使用可能な一式の資料セットの中で、WebSphere アダプターにインストールされるあらゆる共通する機能とコンポーネントが説明されています。また、個々のコンポーネントに関する参照情報も含まれています。

次の Web サイト

<http://www.ibm.com/websphere/integration/wbiadapters/infocenter> では、この資料をインストールすることも、オンライン上でそのまま閲覧することもできます。

この資料セットは主に PDF ファイルで構成されていますが、HTML 形式のファイルも含まれています。資料を読むためには、Netscape Navigator または Internet Explorer などの HTML ブラウザーと、Adobe Acrobat Reader 4.0.5 以降が必要で、使用しているプラットフォームに対応する Adobe Acrobat Reader の最新バージョンを入手するには、Adobe Web サイト (<http://www.adobe.com>) にアクセスしてください。

WebSphere Message Broker の資料

このインプリメンテーション・ガイドは、WebSphere Business Integration Adapters と既存の WebSphere Message Broker (WebSphere MQ Integrator 2.1、WebSphere MQ Integrator Broker 2.1、または WebSphere Business Integration Message Broker 5.0 のいずれか) で構成するビジネス・インテグレーション・システムの展開に必要な情報を提供します。

WebSphere Message Broker のインストール、構成、および管理の詳細や、メッセージ・フローの作成と管理の詳細については、以下の IBM Web サイトを参照してください。

- **WebSphere MQ ファミリーの製品に関する一般情報**

<http://www.ibm.com/software/integration/mqfamily>

- **WebSphere MQ Integrator 2.1 の詳細**

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/manuals/wsmqsiv21.html>

- **WebSphere Business Integration Message Broker の詳細**

<http://www.ibm.com/software/integration/wbimessagebroker/library>

System Manager の資料

WebSphere Business Integration Adapters の構成および管理アクティビティーの多くは、System Manager というグラフィカル・ユーザー・インターフェースを使用して実行できます。このマニュアルで取り扱う範囲を超えた System Manager の詳細については、以下の 2 つのガイドを参照してください。

- 「*IBM WebSphere InterChange Server* システム管理ガイド」

- 「IBM WebSphere InterChange Server インプリメンテーション・ガイド」
これらは以下の Web サイトから入手できます。

<http://www.ibm.com/websphere/integration/wicserver/infocenter>

表記上の規則

本書では、以下の規則を使用します。

Courier フォント	コマンド名、ファイル名、入力情報、またはシステムが画面に表示する情報など、リテラル値を表します。
太字	初出語を示します。
イタリック、イタリック	変数名または相互参照を表します。
青字	オンラインでマニュアルを表示した場合にのみ見られる青の部分は、相互参照のハイパーリンクを示します。青字をクリックすると、参照の対象にジャンプします。
{ }	構文行では、中括弧によって囲まれた複数のオプションから、1つのオプションだけを選択する必要があります。
[]	構文行では、大括弧によってオプション・パラメーターが囲まれます。
...	構文行では、省略符号 (...) は直前のパラメーターの繰り返しを示します。例えば、option[,...] は、複数のオプションをコンマで区切って入力できることを示します。
< >	命名規則では、不等号括弧によって名前の個々の要素が囲まれ、各要素を区別します (例： <server_name><connector_name>tmp.log)。
/, ¥	本書では、ディレクトリー・パスに円記号 (¥) を使用します。UNIX システムの場合には、円記号 (¥) はスラッシュ (/) に置き換えてください。製品のパス名はすべて、ご使用のシステムで製品がインストールされたディレクトリーからの相対パス名です。
<i>ProductDir</i> %text% および \$text	製品のインストール先ディレクトリーを表します。 % 記号で囲まれたテキストは、Windows の text システム変数またはユーザー変数の値を示します。UNIX 環境での同等の表記は \$text です。これは、UNIX 環境変数 text の値を示します。
Windows:	先頭にこれらの記述のいずれかがあるパラグラフは、オペレーティング・システムの違いをリストする注記です。
UNIX:	
AIX:	
Solaris:	
HP-UX:	

改訂の要約

本章では、現行リリースの「*WebSphere Message Broker 使用アダプター・インプリメンテーション・ガイド*」に対する変更点について説明します。

WebSphere Business Integration Adapter Framework バージョン 2.4.0 の新機能

2004 年 2 月

本書では以下に挙げる内容が更新されました。

- System Manager および Eclipse Workbench について説明する 147 ページの『付録 E. System Manager と Eclipse Workbench』が追加されました。
- 67 ページの『第 7 章 ビジネス・インテグレーション・システムの管理』および 143 ページの『付録 D. コネクタ始動オプション』では、コネクタの開始および停止についての情報が更新されました。

2003 年 12 月

本書で更新された主要な変更内容は、以下の通りです。

- 本書の英語版の表題が変更になりました。本書の英語版は、以前は「*Implementing Adapters with WebSphere MQ Integrator Broker*」という表題でした。
- 本書の範囲は、各種メッセージ・ブローカー (WebSphere MQ Integrator 2.1、WebSphere MQ Integrator Broker 2.1、および WebSphere Business Integration Message Broker 5.0) が、アダプターの統合ブローカーとして使用できるようになったことが反映されるように変更されました。以前は WebSphere MQ Integrator Broker のみがサポートされていました。
- System Manager は、アダプター・プロジェクトを統合ブローカーに配置するために使用されるようになりました。
- 構成中に、プロジェクトが、直列化されたビジネス・オブジェクトで「long (長形式)」と「short (短形式)」のどちらの XML ネーム・スペースを使用するかを選択できるようになりました。
- 本書では、アダプターの始動、停止、および休止における System Manager の使用について説明しています。
- アダプターの新規インストールおよびパッケージ化に対応した変更

WebSphere Business Integration Adapters v 2.3.1 での変更点

本書は、下記にリストした変更を組み込むように更新されました。

- 本書の英語版の表題が変更になりました。本書の英語版は、以前は「*Implementation Guide for WebSphere MQ Integrator Broker*」という表題でした。

- 新たに追加された標準構成プロパティにより、RFH2 メッセージ・ヘッダーのメッセージ・ドメインを XML に設定できるようになりました。これについては、115 ページの『付録 A. WebSphere MQ のメッセージ・フォーマット』で説明しています。
- 199 ページの『付録 H. WebSphere Business Integration Adapters のアップグレード』がアップデートされ、リリース 2.3.1 へのアップデートに関する説明が追加されました。
- 以前に存在した『WebSphere Business Integration Adapters for Windows のインストール』および『WebSphere Business Integration Adapters for UNIX のインストール』という 2 つの章は、1 つにまとめられて『WebSphere Business Integration Adapters のインストール』という章になりました。WebSphere Integration Adapter のインストールの詳細については、「WebSphere Business Integration Adapters インストール・ガイド」という別の資料で説明しています。

WebSphere Business Integration Adapters v 2.2.0 での変更点

本書は、下記にリストした変更を組み込むように更新されました。

- 199 ページの『付録 H. WebSphere Business Integration Adapters のアップグレード』では、WebSphere Business Integration Adapters のバージョン 2.2.0 へのアップグレードについて説明します。
- 34 ページの『保証付きイベント・デリバリー』では、各イベントが 2 回以上処理されないようにする、保証付きイベント・デリバリー機能について説明します。
- 67 ページの『WebSphere Business Integration Adapters のサイレント・インストールまたはアンインストールの実行』では、人が介在することなく WBIA をインストールまたはアンインストールする方法について説明します。
- 第 7 章の 76 ページの『WebSphere Business Integration Adapters のサイレント・インストールまたはアンインストールの実行』では、人が介在することなく WBIA をインストールまたはアンインストールする方法について説明します。
- 81 ページの『WBIA バッチ・ファイルを使用して WebSphere MQ キューを構成する方法』では、ビジネス・インテグレーション・システムで WebSphere MQ キューを構成するときに使用可能な新規のバッチ・ファイルについて説明します。
- 78 ページの『WebSphere MQ キューからのメッセージのクリア』では、WebSphere MQ キューからメッセージをクリアするために新規バッチ・ファイルを使用する方法について説明します。
- 次の名前が変更されました。
 - WebSphere MQ Integrator は、WebSphere MQ Integrator Broker になりました。
 - MQSeries は、WebSphere MQ になりました。
- Borland VisiBroker は、WebSphere MQ Integrator Broker を使用する WebSphere Business Integration Adapters では不要となったため、この製品に対する参照は除去されました。

WebSphere Business Integration Adapters v 2.1.0 での変更点

WebSphere Business Integration Adapters 2.1 で加えられた変更による、本書の内容への影響はありません。

WebSphere Business Integration Adapters v 2.0.1 での変更点

本書では下記の変更が行われています。

- 199 ページの『付録 H. WebSphere Business Integration Adapters のアップグレード』では、WBIA V 2.0 から V 2.0.1 へのアップグレード方法について説明します。
- 66 ページの『WebSphere Business Integration Adapters のアンインストール』では、Windows 環境における WebSphere Business Integration システムまたは選択したアダプターのアンインストール操作について説明します。
- 75 ページの『WebSphere Business Integration Adapters のアンインストール』では、UNIX 環境における WebSphere Business Integration システムまたは選択したアダプターのアンインストール操作について説明します。
- 125 ページの『付録 C. コネクターの標準構成プロパティ』には、統合ブローカーに WebSphere MQ Integrator を使用するコネクターの、標準構成プロパティをリストします。
- 183 ページの『付録 G. Visual Test Connector の使用』では、Visual Test Connector ツールを使用して、定義および構成したビジネス・インテグレーション・コンポーネントが正常に機能しているかどうかを検証する方法について説明します。
- 94 ページの『メッセージ・ブローカー・プロジェクトの作成』は改訂され、拡張されました。
- 国際化対応をサポートするための変更に関する情報が追加されました。

第 1 部 概要と概念

第 1 章 WebSphere Business Integration Adapters の概要

接続構成および接続形態はますます複雑化し、大規模化しているため、データの形式変換およびルーティング・ルールを使用して、アプリケーション間で情報を移動する機能がこれまで以上に重要になっています。WebSphere Message Broker を WebSphere Business Integration Adapters の統合ブローカーとして使用すれば、ビジネス・ゴールを達成するために必要な、アプリケーション間での情報の接続を実現することができます。

WebSphere Business Integration Adapters では、モジュラー・コンポーネントと、アプリケーションに依存しないビジネス・ロジックを使用します。このモジュラー方式によって、より速やかな展開、アプリケーションのアップグレードが容易になる、ビジネス・プロセス・フローおよびアプリケーション・アクセス・コードの再利用性などといった、従来のカスタム統合作業にまさる多くの利点をもたらされます。このアダプターがサポートする **ビジネス・インテグレーション・システム**は、分散型で柔軟性に優れ、サイト固有のニーズとアプリケーション固有のニーズを満たすことができるカスタマイズ機能を備えています。

このインプリメンテーション・ガイドでは、アダプターの統合ブローカーの役割を果たす、サポート対象の WebSphere Message Broker と共に WebSphere Business Integration Adapters を備えた WebSphere Business Integration システムを、展開し管理する方法について説明します。統合ブローカーとしてサポートされるのは、以下の WebSphere Message Broker です。

- WebSphere MQ Integrator 2.1.0
- WebSphere MQ Integrator Broker 2.1.0
- WebSphere Business Integration Message Broker 5.0

注: 本書では、「メッセージ・ブローカー」という用語は上記のソフトウェア・プロダクトのみを指し、他のプロダクトを意味しません。「メッセージ・ブローカー」という用語が使用されている場合、その箇所での機能の説明は、上記のすべてのプロダクトに当てはまります。

このガイドは、次の 2 つのパートに分かれています。

- 第 1 部では、WebSphere Business Integration Adapters の概念的概要と、このアダプターを構成する主要コンポーネントを示し、ビジネス・インテグレーション・システムを成り立たせるためにアダプターと統合ブローカーがどのようなプロセスで対話するかについて説明します。
- 第 2 部では、WebSphere Message Broker とともに WebSphere Business Integration Adapters をインストール、構成、および管理する際に役に立つ、作業本位の情報を提供します。

本章では、ビジネス・インテグレーション・システムのアーキテクチャーを説明し、アダプターのコンポーネントについて紹介します。本章は、次の各セクションから構成されています。

- 4 ページの『WebSphere Business Integration システムとは?』

- 5 ページの『WebSphere Message Broker とは?』
- 5 ページの『WebSphere Business Integration Adapters とは?』
- 6 ページの『WebSphere Business Integration システムの機能』

必要な資料についての注意

ビジネス・インテグレーション・アダプターの展開に必要なすべてのタスクを実行するには、本書と合わせて、WebSphere Business Integration Adapters ライブラリーにあるほかの資料を使用する必要があります。特に、以下の資料が必要となります。

- ビジネス・オブジェクト開発ガイド
- 配置するアダプター用のアダプター・ユーザーズ・ガイド

カスタム・アダプターを作成する場合には、次の資料も使用する必要があります。

- コネクタ開発ガイド (Java 用)
- コネクタ開発ガイド (C++ 用)

ライブラリーにある資料の完全リストについては、viii ページの『WebSphere Business Integration Adapters の資料』を参照してください。

WebSphere Business Integration システムとは?

e-business を効果的に実行するには、企業は次の 2 つの課題を解決しなければなりません。

- 商取引を行うために業務情報を各種ソース間で移動する必要がある。
- 企業環境での多種多様なアプリケーション間で業務情報を処理し、発送する必要がある。

ビジネス・インテグレーション・システムは、これらの 2 つのニーズのいずれにも柔軟性と拡張性を示しながら対応します。

大まかに説明すると、**WebSphere Business Integration システム**は 1 つの統合ブローカー (サポートされる WebSphere Message Broker、WebSphere InterChange Server、または WebSphere Application Server) と 1 組のアダプターで構成されており、この構成では、秩序立てて行われる情報転送を介して、異種の業務アプリケーションの間でデータをビジネス・オブジェクトという形で交換することができます。

5 ページの図 1 に、WebSphere Business Integration システムの略図を示します。

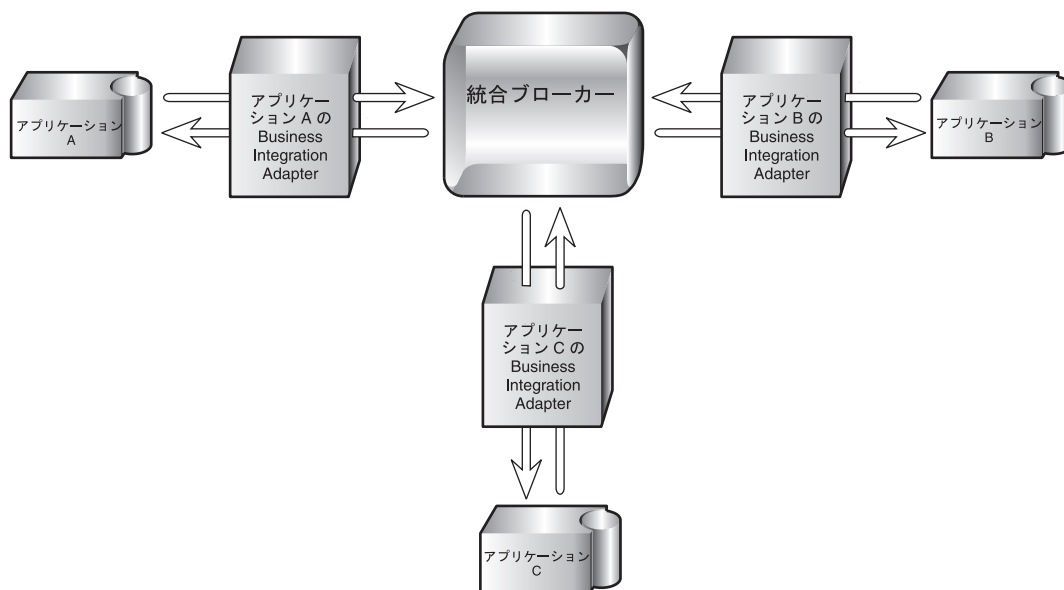


図 1. WebSphere Business Integration システムの上位図

WebSphere Message Broker とは?

WebSphere Message Broker は、WebSphere Business Integration システムで使用できる統合ブローカーの一部です。メッセージ・ブローカーは、WebSphere MQ メッセージングの基本的な接続機能とトランスポート機能を拡張します。アダプターの統合ブローカーとしてサポートされるのは、以下のメッセージ・ブローカーです。

- WebSphere MQ Integrator 2.1.0
- WebSphere MQ Integrator Broker 2.1.0
- WebSphere Business Integration Message Broker 5.0

WebSphere Message Broker を使用すると、情報がユーザー定義のルールに従って正しい形式で適切な場所に到達するために必要な処理が実行されるので、異なった形式の情報を異種のアプリケーションの間で交換できるようになります。データ交換は、ブローカーによって実行され、アプリケーション側は、データを受け取るアプリケーションのデータ規則や要件についての知識を必要としません。

WebSphere Business Integration Adapters とは?

WebSphere Business Integration Adapters ポートフォリオは、一連のソフトウェア・プログラム、アプリケーション・プログラミング・インターフェース (API)、およびツールで構成されており、アプリケーション間における WebSphere Message Broker 経由のビジネス・データ交換を可能にするために使用できます。各ビジネス・アプリケーションをビジネス・インテグレーション・システムに組み込むには、アプリケーション固有のアダプターが必要です。

各アダプターには、次のものが組み込まれています。

- アプリケーションを統合ブローカーにリンクするコネクター
- コネクターを構成する際や、アプリケーションに必要なビジネス・オブジェクト定義を作成する際に役に立つグラフィカル・ユーザー・インターフェース付きのツール
- Object Discovery Agent (ODA)。アプリケーションのデータ・ストアに対して実行し、ビジネス・オブジェクト定義を作成します。ユーザーは、この後でこれを詳細化することができます。一部のアプリケーション用の WebSphere Business Integration Adapters には、ODA が組み込まれていませんので、ご注意ください。
- Object Discovery Agent Development Kit (ODK)。ODA を作成するために使用できる一組の API で構成されます。

特定のレガシー・アプリケーションまたは特化されたアプリケーション用に事前作成されたコネクターが IBM から提供されていない場合のために、カスタム・アダプターを開発するためのフレームワークを備えた、別売の Adapter Development Kit (ADK) が用意されています。

WebSphere Business Integration システムの機能

WebSphere Business Integration システムでは、アプリケーションと統合ブローカーの間でデータを移動するためのコネクティビティーは、WebSphere MQ キューを介した Java Message Service (JMS) を使用するコネクターによって提供されます。コネクターは、どのマシンにも常駐することができ、そこから必要なキューへのアクセスやアプリケーションとの通信を行います。

各コネクターは、コネクター・フレームワークとアプリケーション固有のコンポーネントの 2 つのパーツで構成されます。

- コネクター・フレームワークは、WebSphere MQ キューを使って統合ブローカーと対話します。
- アプリケーション固有のコンポーネントは、アプリケーションと直接対話します。

コネクターのサブコンポーネントを 7 ページの図 2 に示します。



図2. コネクタのサブコンポーネント

データは、**アプリケーション固有のビジネス・オブジェクト**を使ってアプリケーション間で交換されます。これらのビジネス・オブジェクトは、コネクタ・フレームワークと統合ブローカーの間で **WebSphere MQ メッセージ** (ビジネス・オブジェクト・メッセージとも呼ばれる) として移送されます。

ビジネス・オブジェクトは、ビジネス・データをいくつかの目的でカプセル化および伝送します。ビジネス・オブジェクトは、次のものを搬送します。

- 新規データまたは変更データをソース・アプリケーションから宛先アプリケーションに搬送する。
- ソース・アプリケーションで作成したデータに対する要求を宛先アプリケーションに搬送する。
- データに対する要求に応答してアプリケーションによって戻されたデータを搬送する。

メタデータとしてエンコードされた、各データに関連付けられている命令は、検索、作成、または更新の対象となるデータの、アプリケーション・データベース内におけるロケーションを指定します。ビジネス・オブジェクトの新規インスタンスは、ビジネス・オブジェクトの属性、値、およびメタデータの構造および編成を指定した、**テンプレート (ビジネス・オブジェクト定義と呼ばれる)** に基づいてアプリケーション固有のコンポーネントによって作成されます。

ビジネス・オブジェクト定義内のアプリケーション固有の情報や他のメタデータは、アプリケーション固有のコンポーネントのアクションをガイドします。そのた

め、アプリケーション固有のコンポーネントの振る舞いは**メタデータ主導型**であると言えます。アプリケーション固有のコンポーネントには、それがサポートするビジネス・オブジェクトのタイプごとに、ハードコーディングされた命令はありません。したがって、メタデータ主導型であるアプリケーション固有のコンポーネントは柔軟性があります。対応するアプリケーション・データがコネクターのメタデータ構文によって正確に記述することができるかぎり、アプリケーション固有のコンポーネントは、再コード化または再コンパイルすることなく、新規のビジネス・オブジェクト定義を自動的にサポートします。

統合ブローカー内では、**WebSphere MQ メッセージ・フロー**が、統合ブローカーによるビジネス・オブジェクト・メッセージ処理のステップを定義します。MQ メッセージ・フローは、統合ブローカーによるメッセージの受け取りから宛先アプリケーションへのメッセージの引き渡しまでの間に実行されるアクション・セットまたはルールを指定します。

ビジネス・インテグレーション・システムのデータ・フロー

ビジネス・インテグレーション・システムでは、データ・フロー（アプリケーション間またはエンティティ間で送信されるデータの移動および処理）は、ローカル・ネットワーク上のアプリケーション間の非同期または同期のデータ交換として発生します。

アプリケーションは、そのデータ・ストア内の変更を通知するため、またはデータを入手するために、別のアプリケーションとデータを交換することが必要となる場合があります。

ビジネス・インテグレーション・システム内のデータ交換は、以下のステップで構成されます。

1. イベント通知
2. 統合ブローカー処理
3. 要求処理

上記の各ステップを次に詳細に説明します。

イベント通知

変更されたアプリケーション・データを統合ブローカーに送るプロセスは**イベント通知**と呼ばれます。ビジネス・インテグレーション・システムに加入するアプリケーションのほとんどは、構成処理中に変更されて、**イベント・ストア**（アプリケーションのデータ変更やデータ要求をロギングするテーブルなど）が組み込まれます。アプリケーションに新たに変更された共有データがあることやアプリケーションが別のアプリケーションの情報を必要としていることを検出するために、コネクタ・フレームワークは定期的間隔で**ポーリング呼び出し**を開始します。ポーリング呼び出しは、アプリケーションのイベント・ストアに変更があるかどうかをチェックするようアプリケーション固有のコンポーネントに依頼します。

最後のポーリング呼び出し以降に変更があった場合には、アプリケーション固有のコンポーネントは、変更データまたはデータ要求を表すビジネス・オブジェクト定義が存在するかどうかを判別します。コネクターのローカル・リポジトリに適切なビジネス・オブジェクト定義が存在することは、この特定の変更または要求を別

のアプリケーションに伝達する必要があることを示します。アプリケーション固有のコンポーネントは、アプリケーション・データをビジネス・オブジェクト形態でコネクタ・フレームワークに送信します。これは、**イベント・デリバリー**と呼ばれます。アプリケーション・データの変更やデータの要求はイベントと見なされるためです。

図3に、コネクタおよびそのサポート・インフラストラクチャーを示します。ここで、アプリケーションのデータ・ストアに対する変更を検出し、アプリケーション固有のビジネス・オブジェクトを組み立てて、変更データを統合ブローカーに搬送します。

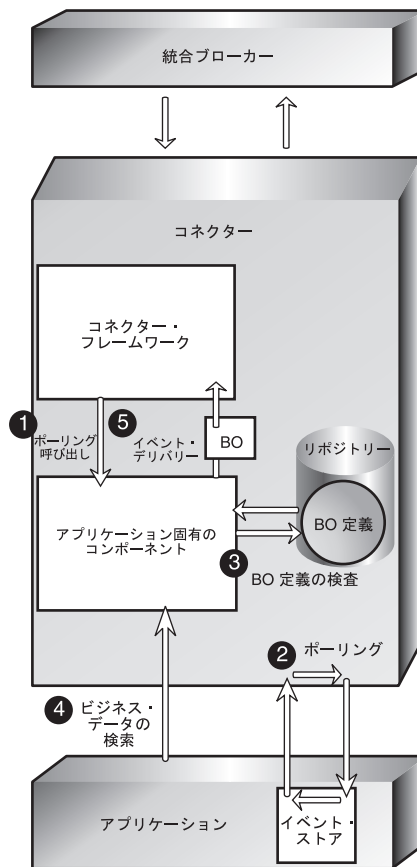


図3. イベントを検出して引き渡すコネクタ

図中の数値は、ステップの順序を示します。

1. コネクタ・フレームワークは、アプリケーション固有のコンポーネントを開始して、アプリケーションのイベント・ストアに変更があるかどうかをチェックさせます。
2. アプリケーション固有のコンポーネントは、アプリケーションのイベント・ストアに変更があるかどうか調べるためにポーリングします。
3. アプリケーション固有のコンポーネントは、サポートされているビジネス・オブジェクト定義に変更データがマップされているかどうかを判定します。
4. アプリケーション固有のコンポーネントは、ビジネス・オブジェクトをインスタンス化し、それを使用して変更データを検索します。

5. アプリケーション固有のコンポーネントは、イベントの引き渡しを開始し、ビジネス・オブジェクトをコネクタ・フレームワークに転送します。

ソース・アプリケーションのコネクタ・フレームワークは、アプリケーション固有のビジネス・オブジェクトを受け取ると、WebSphere MQ キューに置くことが可能な WebSphere MQ メッセージに、そのビジネス・オブジェクトを変換して、統合ブローカーが受け取ることができるようにします。データ・ハンドラーは、ビジネス・オブジェクトを宛先 WebSphere MQ キューに応じた適切な XML ベースのワイヤー形式のメッセージに変換するために、コネクタ・フレームワークが使用します。図 4 に、上記のプロセスを示します。

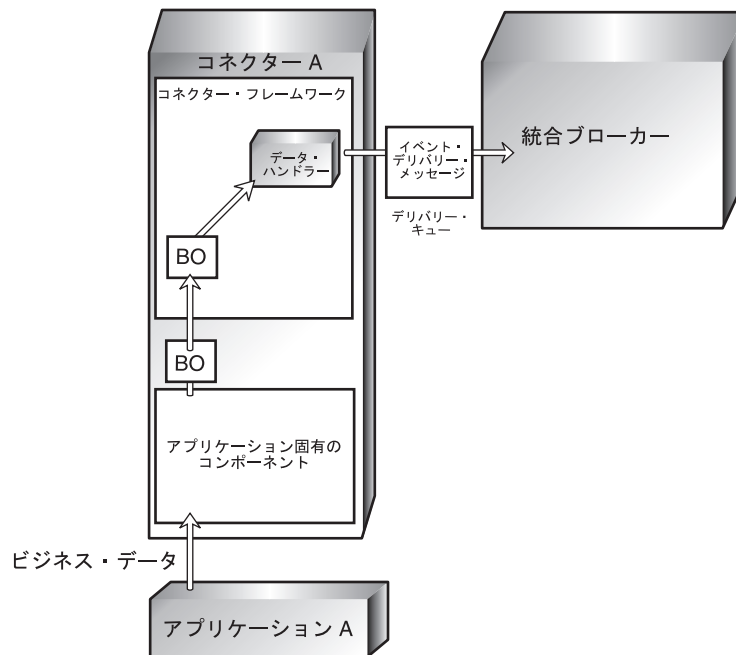


図 4. ビジネス・オブジェクトを MQ メッセージに変換するコネクタ・フレームワーク

統合ブローカー処理

メッセージが統合ブローカーの WebSphere MQ イベント・デリバリー・キューに置かれると、統合ブローカーはそのメッセージをキューから取り出して、当該キューのメッセージ・フローの中を通過させます。処理には以下の操作が含まれます。

- 値を計算してメッセージを変換する
- メッセージ内のデータからフィールドを抽出する
- メッセージを 1 つ以上の宛先にルーティングする
- メッセージをメッセージ・ウェアハウスにアーカイブする
- メッセージの内容に基づいてデータベース情報を更新する
- 宛先アプリケーションがメッセージを処理できるように、メッセージの内容または構造を変換する

結果のメッセージ（ここでは、**要求**と呼びます）は、WebSphere MQ 要求キューに置かれ、宛先アプリケーションのコネクター・フレームワークに転送されます。図 5 に、統合ブローカーが行う処理を図示します。

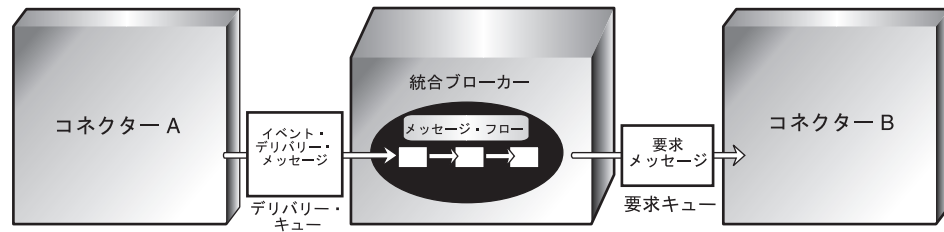


図 5. 統合ブローカー処理

要求処理

要求が宛先コネクターのキューに置かれると、**listen** メカニズムは WebSphere MQ メッセージがその要求キューに到着したこと、および処理が必要であることを、コネクター・フレームワークに通知します。コネクター・フレームワークはデータ・ハンドラーを起動して、宛先アプリケーションによる処理が可能なビジネス・オブジェクトに WebSphere MQ メッセージを変換します。図 6 を参照してください。

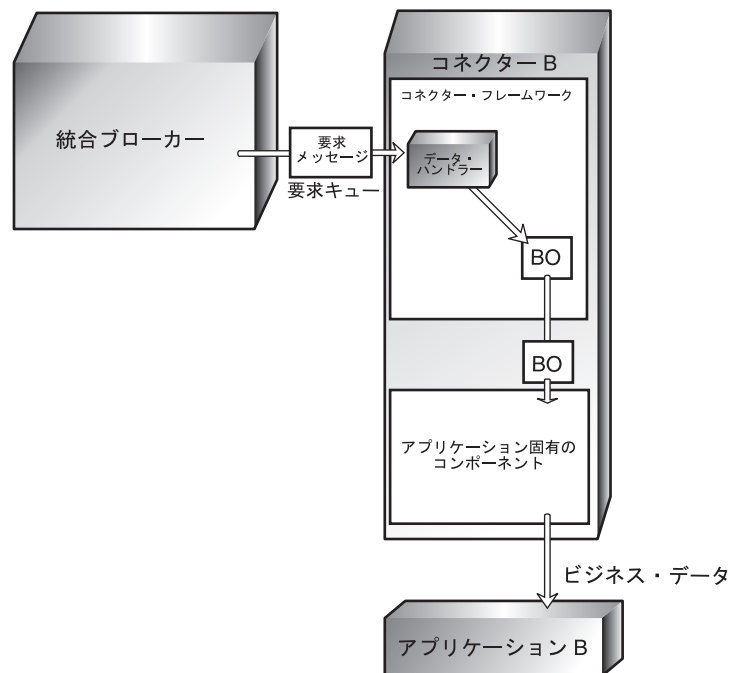


図 6. コネクターによる要求処理

場合によっては、要求が宛先アプリケーションからの**応答**を必要とすることがあります。一般に、応答は次のことを行うために使用されます。

- ソース・アプリケーションが宛先アプリケーションに要求したデータを戻す

- ソース・アプリケーションが宛先アプリケーションに作成を依頼した新しいビジネス・エンティティ（カスタマーや注文など）についての情報をソース・アプリケーションに戻す

応答が必要な場合には、アプリケーション固有のコンポーネントは、情報を搬送するために要求ビジネス・オブジェクトを変更して、ビジネス・オブジェクトをコネクタ・フレームワークに送り返します。コネクタ・フレームワークは、データ・ハンドラーを呼び出してビジネス・オブジェクトを WebSphere MQ メッセージに変換し、発信元の要求メッセージで指定された応答先キューにメッセージを置きます。応答メッセージ内の相関 ID は、応答対象メッセージを表します。図 7 に応答処理が実行される仕組みを図示します。

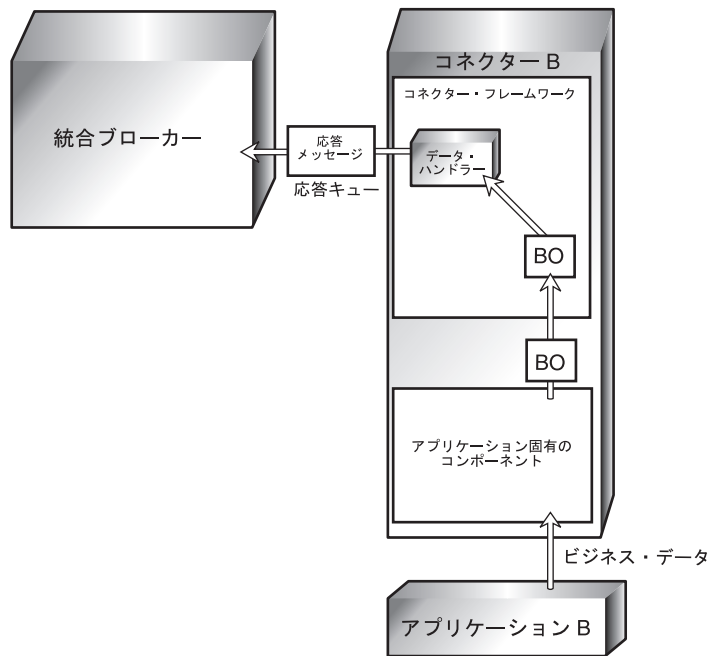


図 7. コネクタによる応答処理

ビジネス・インテグレーション処理の要約

これで、ビジネス・インテグレーション処理の各ステップが理解できたので、次にシステムを全体として復習します。13 ページの図 8 に、ビジネス・インテグレーション・システムの要約図を示します。以下に 2 つのシナリオを用意し、システムが動作する仕組みを示します。

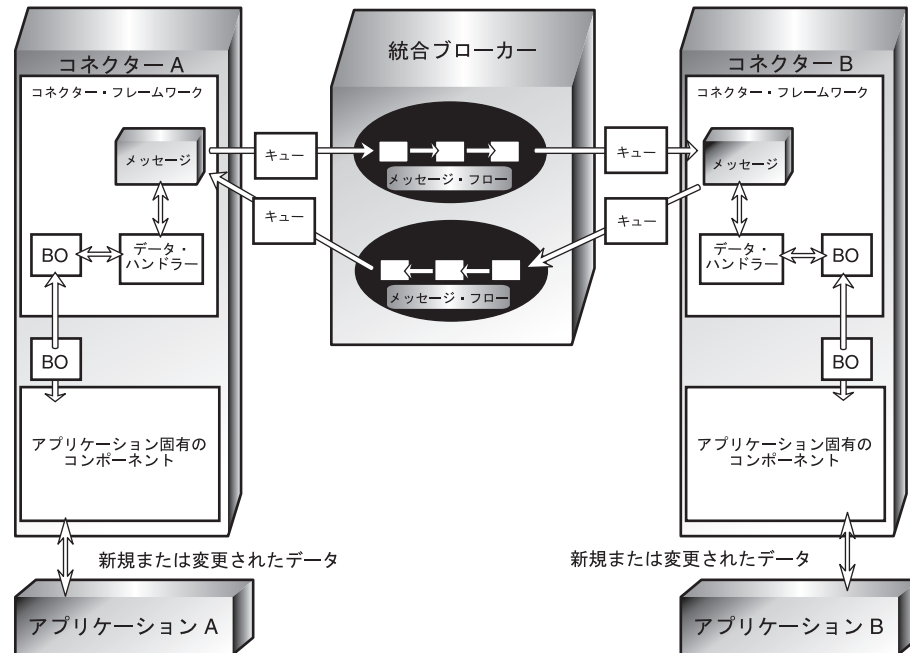


図 8. WebSphere Business Integration システムでのデータ・フロー

変更データを別のアプリケーションに送信する例

ビジネス・インテグレーション・システムが実行するステップの一例を示します。このステップによってアプリケーション A は、アプリケーション B に同期をとるための変更データを送信することができます。

1. コネクター A のアプリケーション固有のコンポーネントはアプリケーション A にデータの変更があることを検出します。このコンポーネントは、この変更を伝達するためのビジネス・オブジェクト定義が存在すると判断し、そのビジネス・オブジェクト定義を使用して、変更情報を搬送するためのビジネス・オブジェクトを構成します。
2. アプリケーション固有のコンポーネントは、ビジネス・オブジェクトをコネクター・フレームワークに渡します。
3. コネクター・フレームワークはデータ・ハンドラーを起動し、ビジネス・オブジェクトを正しい XML ベースのワイヤー形式をもつ WebSphere MQ メッセージに変換して、そのメッセージを統合ブローカーの WebSphere MQ キューに置きます。
4. 統合ブローカーはメッセージを受け取り、それをメッセージ・フローに渡します。
5. メッセージがメッセージ・フローによって処理されると、統合ブローカーは結果のメッセージをアプリケーション B のコネクターの WebSphere MQ キューに置きます。
6. コネクター B のコネクター・フレームワークはメッセージをキューから取り出し、データ・ハンドラーを起動して、アプリケーション固有のコンポーネントによる処理が可能なビジネス・オブジェクトにメッセージを変換します。

7. アプリケーション B は、自分のカスタマー情報を更新してアドレスの変更を反映させます。

アプリケーション A がアプリケーション B にデータ変更を通知するのではなく、データを要求している場合には、アプリケーション B は、アプリケーション A に応答を返信する必要があります。次の例で、このシナリオを説明します。

別のアプリケーションからデータを入手する例

次に示すビジネス・インテグレーション・システムのステップでは、アプリケーション A は、あるカスタマーの最新の購入情報をアプリケーション B から検索します。

1. コネクター A のアプリケーション固有のコンポーネントは、アプリケーション A がアプリケーション B にデータを要求したことを検出します。このコンポーネントは、この要求を伝えるためのビジネス・オブジェクト定義が存在すると判断し、そのビジネス・オブジェクト定義を使用して、要求された情報を搬送するためのビジネス・オブジェクトを構成します。
2. アプリケーション固有のコンポーネントは、ビジネス・オブジェクトをコネクター・フレームワークに渡します。
3. コネクター・フレームワークはデータ・ハンドラーを起動し、ビジネス・オブジェクトを正しいワイヤー形式の WebSphere MQ メッセージに変換して、そのメッセージを統合ブローカーの WebSphere MQ キューに置きます。
4. 統合ブローカーはメッセージを受け取り、それをメッセージ・フローに渡します。
5. メッセージがメッセージ・フローによって処理されると、統合ブローカーは結果のメッセージをアプリケーション B のコネクターの WebSphere MQ キューに置きます。
6. コネクター B のコネクター・フレームワークはメッセージをキューから取り出し、データ・ハンドラーを起動して、アプリケーション固有のコンポーネントによる処理が可能なビジネス・オブジェクトにメッセージを変換します。
7. コネクター B のアプリケーション固有のコンポーネントは、要求で指定された情報を検索し、それをビジネス・オブジェクトとしてコネクター・フレームワークに渡します。
8. コネクター B のコネクター・フレームワークは、ビジネス・オブジェクトを応答メッセージに変換するためにデータ・ハンドラーを呼び出し、発信元の要求で指定された応答先キューにメッセージを置きます。

以降の章では、本章で紹介したビジネス・インテグレーション・コンポーネントについて記述するとともに、これらのコンポーネントの働きによってアプリケーションがデータを共用できるようになる過程についても詳細に説明します。

第 2 章 ビジネス・オブジェクト

ビジネス・オブジェクトとは、データ・エンティティ (1 つのまとまりとして扱われるデータの集合) を表します。データ・エンティティの例としては、従業員に関する基本情報すべて (氏名、住所、電話番号、従業員番号、職位コード、月収など) を含む従業員レコードがあげられます。

ビジネス・インテグレーション・システムでは、エンティティに含まれる情報を反映したビジネス・オブジェクトを作成します。本書では、データ・エンティティは、それに含まれるビジネス情報の種類 (例えば、従業員エンティティ やカスタマー・エンティティ など) を指す意味でしばしば使われます。

ビジネス・オブジェクト定義とは、アプリケーション固有のコンポーネントがビジネス・オブジェクトの特定のインスタンスを作成するときのベースとなるテンプレートのことです。

本章では、ビジネス・オブジェクトについて詳細に紹介するとともに、ビジネス・インテグレーション・システムがビジネス・オブジェクトを使用して、アプリケーション間でデータを搬送する仕組みを説明します。本章は、次の各セクションから構成されています。

- 『ビジネス・オブジェクトの役割』
- 18 ページの『ビジネス・オブジェクトのタイプ』
- 19 ページの『ビジネス・オブジェクト定義とビジネス・オブジェクト』
- 22 ページの『ビジネス・オブジェクトの詳細』
- 25 ページの『ビジネス・オブジェクト定義を作成または変更する方法』

ビジネス・オブジェクトの役割

ビジネス・オブジェクトは、イベント、要求、または応答として機能できます。

イベント

ビジネス・オブジェクトには、**アプリケーション・イベント**が出現したことを報告する機能があります。これはアプリケーション内のデータ・エンティティに影響を与えた操作が出現したことを意味します。アプリケーション・イベントとしては、そのデータ集合の値の作成、削除、または変更があげられます。

コネクターがアプリケーション・イベントを検出して、ビジネス・オブジェクトを統合ブローカーに送信する場合のビジネス・オブジェクトの役割は、イベントを表すことです。そのため、ビジネス・オブジェクトはビジネス・インテグレーション・システムで**イベント**と呼ばれます。

例えば、コネクターは、統合ブローカーに代わって、新しい従業員エンティティがあるかどうかをチェックするためにアプリケーションをポーリングすることがあります。アプリケーションが新しい従業員エンティティを作成した場合には、コネクターは統合ブローカーにイベント・ビジネス・オブジェクトを送信します。

要求

要求は一般に次のように生成されます。統合ブローカーは、コネクター・フレームワークにビジネス・オブジェクト・メッセージを**要求**として送信し、アプリケーション固有のコンポーネントを使用して、あるアプリケーション内のデータを挿入、変更、削除、または取得するように指示します。

応答

コネクターは、要求処理を完了すると、通常、統合ブローカーに**応答**を戻します。例えば、コネクターは、宛先アプリケーション内で従業員レコードを作成する要求を受け取ると、作成した従業員データと作成が正常に行われたことを示す状況表示とを添付してビジネス・オブジェクトを送信します。

ビジネス・オブジェクトの構造

ビジネス・オブジェクトは、タイプ (名前)、処理命令 (動詞)、およびデータ (属性値) が含まれている自己記述ユニットです。

図9 は、単純なビジネス・オブジェクトの一例で、タイプ、動詞、および属性値が示されています。

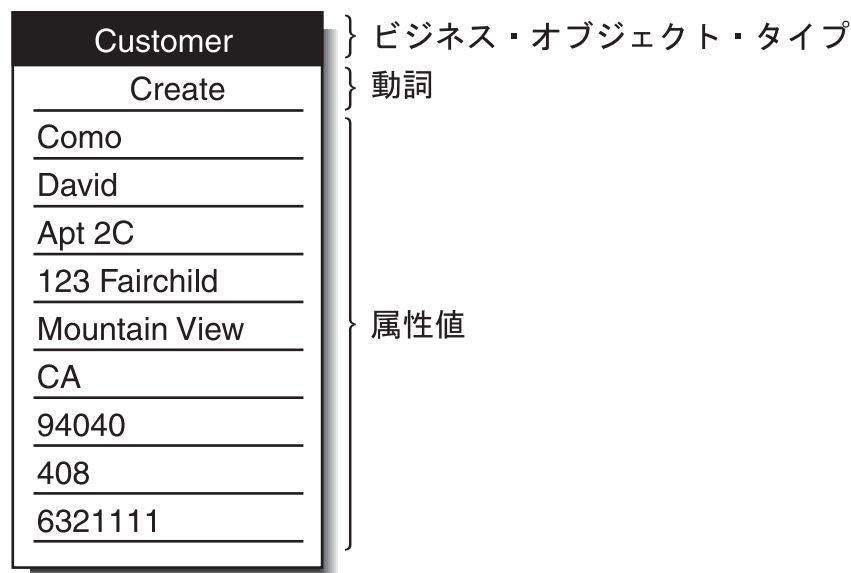


図9. ビジネス・オブジェクトのコンポーネント

次のセクションでは、これらのコンポーネントについて説明します。

ビジネス・オブジェクト・タイプ

各ビジネス・オブジェクトには、ビジネス・インテグレーション・システムでそれぞれを識別するタイプ名が付いています。例えば、タイプには、Customer、Employee、Item、Contract などがあります。

ビジネス・オブジェクト動詞

ビジネス・オブジェクト動詞は、属性値に対応するアクションを指定します。動詞は、ビジネス・オブジェクトの役割によって、さまざまなタイプのアクションを指示することができます。表 1 では、3 つのビジネス・オブジェクトの役割をリストし、それぞれの役割を持つビジネス・オブジェクトでの動詞の意味について説明します。

表 1. ビジネス・オブジェクトの動詞の意味

ビジネス・オブジェクト	
の役割	動詞の意味
イベント	アプリケーションで起こったことを記述します。例えば、イベントで、動詞 Create は、ソース・アプリケーションが新しいデータ・エンティティを作成したことを示します。
要求	ビジネス・オブジェクトを処理するためにアプリケーションと対話する方法をコネクターに指示します。例えば、動詞 Update は、コネクターに対するデータ・エンティティの更新要求です。
応答	関連した要求で指定されている動詞をリストします。例えば、応答で動詞 Retrieve は、コネクターがアプリケーションから属性値を取得したことを示します。

注: IBM の表記規則では、*business-object-type.verb* の形式を使用して、特定の動詞が指定された特定タイプのビジネス・オブジェクトを指示します。例えば、**Customer.Create** は、動詞 **Create** が指定されたビジネス・オブジェクト **Customer** です。

ビジネス・オブジェクト属性値

ビジネス・オブジェクトには、Last Name (ラストネーム)、First Name (ファーストネーム)、Employee ID (従業員 ID)、Invoice Status (送り状状況) など、データ・エンティティに関連したデータ・フィールドを表す属性値が含まれます。

一部の属性には、データが含まれる代わりに、子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列が含まれます。18 ページの図 10 に、ビジネス・オブジェクト **Contract** (契約) の構造を示します。契約の Line Item (明細品目) 情報は、子ビジネス・オブジェクトの配列内にあります。

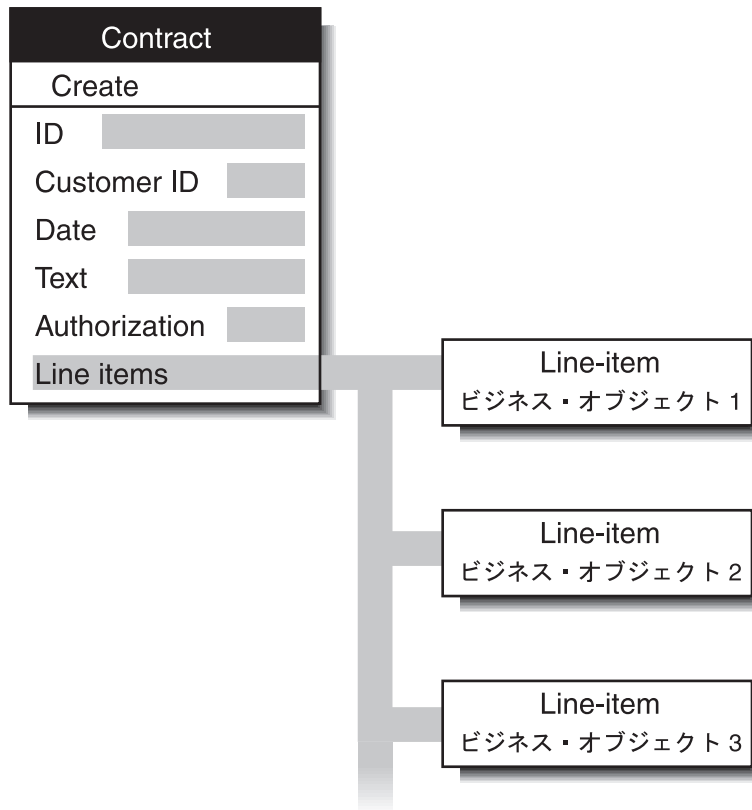


図 10. 子ビジネス・オブジェクトがあるビジネス・オブジェクト

子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列を含んだビジネス・オブジェクトは、**階層ビジネス・オブジェクト**です。データのみを含んだ属性をもつビジネス・オブジェクトは、**フラット・ビジネス・オブジェクト**です。

ビジネス・オブジェクトのタイプ

統合ブローカーとして WebSphere InterChange Server (ICS) を使用する完全ビジネス・インテグレーション・システムには、アプリケーション固有のビジネス・オブジェクトと汎用ビジネス・オブジェクトという 2 種類のビジネス・オブジェクトが含まれます。ただし、統合ブローカーとして、メッセージ・ブローカー上で動作する WebSphere Business Integration Adapter は、汎用ビジネス・オブジェクトではなく、アプリケーション固有のビジネス・オブジェクトのみを使用します。したがって、本書でビジネス・オブジェクトというときはすべて、アプリケーション固有のビジネス・オブジェクトを意味しています。IBM WebSphere Business Integration (WBI) Server の資料セットに含まれる多くの資料では、これら両方の環境が扱われるため、両タイプのビジネス・オブジェクトに言及しています。

- **アプリケーション固有のビジネス・オブジェクト**は、特定のアプリケーションまたは他のプログラマチック・エンティティのデータ・エンティティ属性およびデータ・モデルを表します。
- **汎用ビジネス・オブジェクト**には、さまざまなアプリケーションに共通した、すなわち、特定のアプリケーションのデータ・モデルに限定されない、ビジネス関連属性のセットが含まれます。汎用ビジネス・オブジェクトは、WebSphere Business Integration Adapters では使用されませんが、WebSphere Business

Integration Adapters ライブラリーと WebSphere InterChange Server ライブラリーの両方に含まれている資料では、解説されています。

アプリケーション固有のコンポーネントは、更新などのアプリケーション・イベントを検出すると、該当するデータ・エンティティをアプリケーションから取得し、それをビジネス・オブジェクトに変換します。

注: ドキュメンテーションでは、Clarify_Contact や Oracle_Customer などの、アプリケーション名が名前に含まれるビジネス・オブジェクトに言及するときは、アプリケーション固有のビジネス・オブジェクトを指します。例えば、ビジネス・オブジェクト Clarify_Contact には、Clarify アプリケーションが契約に関して保管する一連の情報が含まれます。別のアプリケーションでは、契約エンティティに保管される情報の内容、順序、形式、または名前が多少異なる場合があります。

アプリケーション固有のコンポーネントは、ビジネス・オブジェクトを作成すると、そのビジネス・オブジェクトをコネクタ・フレームワークに送信します。コネクタ・フレームワークは、データ・ハンドラーを呼び出して、そのビジネス・オブジェクトを WebSphere MQ メッセージに変換して、統合ブローカーにディスパッチできるようにします。

ビジネス・オブジェクト定義とビジネス・オブジェクト

3 ページの『第 1 章 WebSphere Business Integration Adapters の概要』では、ビジネス・オブジェクトについて紹介しましたが、ビジネス・オブジェクト定義とビジネス・オブジェクトのインスタンスの区別については簡単に触れた程度でした。ここでは、その区別についてより詳細に説明します。

- **ビジネス・オブジェクト定義**では、WebSphere Business Integration Adapters for WebSphere MQ Integrator Broker が扱う各エンティティにある情報のタイプと順序、およびサポートされる動詞を指定します。コネクタのローカル・リポジトリには、ビジネス・オブジェクト定義が保管されます。
- **ビジネス・オブジェクト**は、実データを含む、定義のインスタンスです。ビジネス・オブジェクトは、ランタイムに作成され、リポジトリに保管されません。

20 ページの図 11 に、ビジネス・オブジェクト定義とビジネス・オブジェクトの関係を示します。

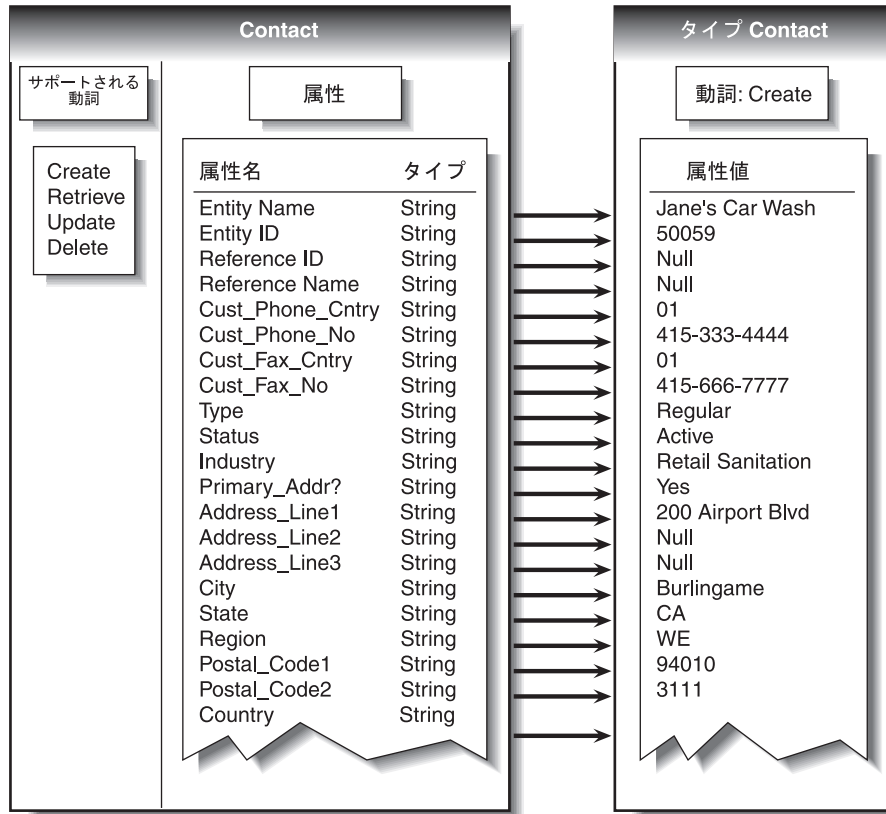


図 11. ビジネス・オブジェクト定義とビジネス・オブジェクト

ビジネス・オブジェクト定義のコンポーネント

一言で言うと、ビジネス・オブジェクトは、そのタイプ、その属性値、およびその動詞で表現されます。

全体として、ビジネス・オブジェクト定義はその名前で識別されます。名前は、Customer、VantiveCase、Invoice などの、ビジネス・オブジェクト定義タイプを示します。また、ビジネス・オブジェクトには、アプリケーション固有のコンポーネントがそれを処理するときに役に立つ、アプリケーション固有情報 (メタデータ) が含まれることもあります。また、すべてのビジネス・オブジェクトには、次の各セクションで説明する、属性および動詞も含まれます。

属性

ビジネス・オブジェクト定義にある属性には、Last Name (ラストネーム)、Employee ID (従業員 ID)、Case Number (事例番号)、Amount (量)、Date Initiated (開始日) などの、エンティティーと結びついた値が記述されます。実行時は、属性に実データが入力されます。

例えば、ビジネス・オブジェクト定義 Employee には従業員の氏名、住所、従業員 ID、および他の関連情報の属性が含まれています。ビジネス・オブジェクトの属性は、データベース表内のフォームまたは列のフィールドに類似しています。

また、属性は、契約の明細品目の配列、送り状のパーツ参照などの、子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列を指すこともあります。

ObjectEventId 属性: ObjectEventId 属性は必須の属性で、いずれのビジネス・オブジェクト中でも最後におかれます。

コネクタは、イベントをパブリッシュすると、ビジネス・オブジェクト定義の ObjectEventId 属性を使用して、作成中のビジネス・オブジェクト・インスタンスを識別する固有値を保管します。

ObjectEventId 属性の値は、ビジネス・インテグレーション・システムによって生成および処理されます。ビジネス・インテグレーション・システムは、この属性を使用して、特定のイベントのシステム内の流れを識別および追跡します。

基本属性タイプと複合属性タイプ: 属性のタイプが String、Boolean、Double、Float、Integer などの基本データ・タイプである場合、属性値は、データベース内のフィールドの値などの離散データです。例として、LastName、CustomerID、PartNumber、AssignedTo、Price などがあります。

属性のタイプが別のビジネス・オブジェクト定義の名前 (複合タイプ) である場合、属性値は子ビジネス・オブジェクトまたは子ビジネス・オブジェクトの配列です。例として、Customer、Contract、Oracle_Contact などがあります。

属性のプロパティ: プロパティの中には、属性が表す値を定義するものがたくさんあります。すべてではありませんが、図 12 に、ビジネス・オブジェクト定義での属性のプロパティの位置を示します。

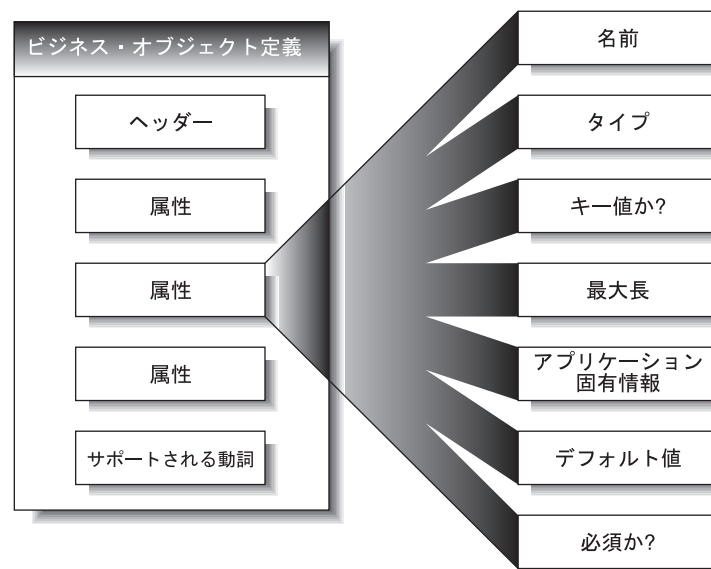


図 12. 属性のプロパティ

特定の属性のプロパティのセットは、その属性タイプが基本タイプであるか、複合タイプであるかによって内容が異なります。すなわち、属性のプロパティは、その属性が単一のデータ・ユニットを指すのか、子ビジネス・オブジェクトを指すのかによって異なります。

動詞

動詞は、ビジネス・オブジェクトのデータに対するアクションを示します。ビジネス・オブジェクト定義には、一連の動詞が含まれます。ビジネス・オブジェクトには 1 つの動詞のみが含まれます。

ビジネス・オブジェクト定義に関連した、最も一般的な動詞は Create、Retrieve、Update、および Delete です。

動詞の意味は、ビジネス・オブジェクトの役割によって異なります。動詞は、アプリケーション・イベントの記述、呼び出し、要求、または、前の要求の結果の確認を行うことができます。

注: 一部のアプリケーションでは、ハード削除の要求はサポートされません。そのようなアプリケーションの場合、ビジネス・インテグレーション・システムは、同等の論理削除 (通常は、非アクティブ状態への更新) を行います。さらに、アプリケーションでハード削除がサポートされる場合であっても、そのアプリケーションに要求を送信する際に Delete 動詞を Update 動詞に変換するように、ビジネス・インテグレーション・システムを構成することができます。

ビジネス・オブジェクトの詳細

ビジネス・オブジェクトには、アプリケーション固有のコンポーネントが特定のアプリケーションに移動したり特定のアプリケーションから移動したりするデータが含まれます。したがって、各ビジネス・オブジェクト定義には、アプリケーションのデータ・モデルやアプリケーション固有のコンポーネントのアクセス・メソッドが反映されます。

2 つのアプリケーション固有のビジネス・オブジェクトが類似したアプリケーション・エンティティを参照する場合でも、属性の編成手法や、それらのビジネス・オブジェクトのアプリケーション固有情報に違いが出ます。

属性の編成

アプリケーションが同じ情報を異なる手法で編成する場合があります。例えば、アプリケーション A は連絡先の電話番号と FAX 番号を 4 つのフィールドに保管し、アプリケーション B は同じ番号を 2 つのフィールドに保管します。

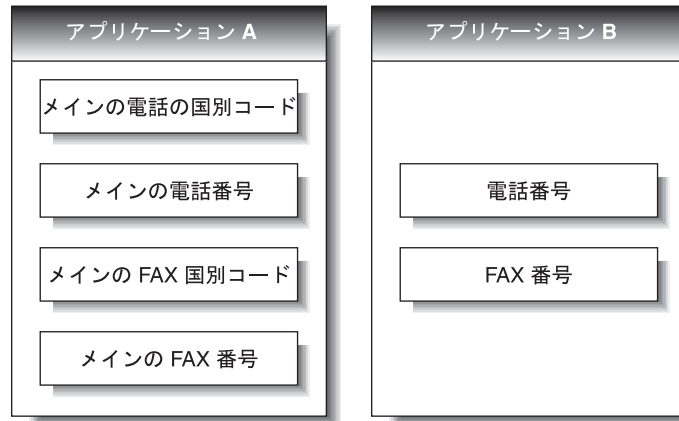


図 13. 2 つのアプリケーションでの電話データ

アプリケーション A およびアプリケーション B のビジネス・オブジェクト定義には、この違いを反映させるために異なる属性があります。

アプリケーション固有情報

また、ビジネス・オブジェクトが異なる理由として、それぞれのビジネス・オブジェクトには、オプションとして、そのアプリケーション固有のコンポーネントに対する組み込み処理命令を含めることができる点があげられます。これは、アプリケーション固有情報 (またはメタデータ) として参照されるもので、ビジネス・オブジェクトを処理するためにアプリケーション固有のコンポーネントが必要とする情報であればどんなものでも含めることができます。

ビジネス・オブジェクト定義には、ビジネス・オブジェクト全体、各属性、および各動詞それぞれに適用するアプリケーション固有情報を含めることができます。ビジネス・オブジェクト定義内でアプリケーション固有情報のある箇所それぞれに、アプリケーションとの対話でコネクタが使用する情報が示されます。

ビジネス・オブジェクトに関するアプリケーション固有情報

ビジネス・オブジェクトに関するアプリケーション固有情報は、ビジネス・オブジェクトを全体として処理する際にアプリケーション固有のコンポーネントが使用する情報を示します。

属性に関するアプリケーション固有情報

属性に適用するアプリケーション固有情報はしばしば、アプリケーション内の属性値ロケーションを識別します。アプリケーション固有のコンポーネントは、アプリケーションに対する API 呼び出しを作成する際にこの ID を使用して、属性値の取得または入力を行います。

アプリケーション固有情報は、アプリケーションごとに異なる形式をとります。アプリケーション固有のコンポーネントが、アプリケーションのフォーム名とフィールド名によって属性のロケーションを参照できる場合もあります。また、参照がより複雑な場合もあります。

24 ページの表 2 に、属性のアプリケーション固有情報に組み込むパラメーターの例を示します。これらのパラメーターは、データベース表内のデータを表すビジネス

ス・オブジェクトにのみ関するものです。

表 2. 属性アプリケーション固有情報の名前と値のペアのパラメーター例

パラメーター	説明
TN= <i>TableName</i>	データベース表の名前です。
CN= <i>col_name</i>	この属性のデータベース列の名前です。
FK=[<i>..</i>]fk_attributeName]	外部キー・プロパティの値により親子関係が定義されます。
UID=AUTO	このパラメーターは、ビジネス・オブジェクト用の固有 ID を生成するように、またこの属性に値をロードするようにコネクターに通知します。
CA= <i>set_attr_name</i>	Copy Attribute (属性のコピー) プロパティは、ある属性の値を別の属性にコピーするようコネクターに指示します。 <i>set_attr_name</i> が現在の個々のビジネス・オブジェクト内の別の属性の名前に設定されている場合には、コネクターは、Create 操作でビジネス・オブジェクトをデータベースに追加する前に、指定された属性の値を使用して、この属性の値を設定します。
OB=[ASC DESC]	Order By パラメーターの値が指定され、子ビジネス・オブジェクトに属性がある場合には、コネクターは検索クエリーの ORDER BY 文節内の属性の値を使用して、子ビジネス・オブジェクトを昇順に取得するか、降順に取得するかを判断します。
UNVL= <i>value</i>	コネクターが、ヌル値属性を持つビジネス・オブジェクトを取得するときにヌルを表すために使用する値を指定します。

1 つの属性アプリケーション固有情報が、上記にリスト例示したパラメーターをいくつか結合したものであることもあります。次の例では、パラメーターを区切るためにセミコロン (;) 区切り文字を使用しています。

```
TN=LineItems;CN=POid;FK=..PO_ID
```

上記の例では、アプリケーション固有情報が、表の名前、列の名前を指定しており、また現在の属性が子ビジネス・オブジェクトをその親にリンクする外部キーであることも指定しています。

例外的に、属性に関するアプリケーション固有情報が不要なケースもあります。例えば、アプリケーションによっては、データ・ユニットに関して非常に直接的で使用しやすい指定を行う場合があります。あるアプリケーションが、表 3 に示すようなサンプル・フィールドを識別する場合を想定してください。

表 3. サンプル・アプリケーション ID

属性	値を持つフィールドのアプリケーション ID
CustomerID (カスタマー ID)	XCustomerID
CustomerName (カスタマー名)	XCustomerName
Status (状況)	XStatus
Industry (産業)	XIndustry

24 ページの表 3 に示されている例では、アプリケーション固有のコンポーネントは、容易に属性とアプリケーション内の ID を関連付けることができます。変換規則が、X の付加、または X の除去というように非常に規則的なためです。したがって、このアプリケーションに対するビジネス・オブジェクトの属性には、アプリケーション固有情報は不要であると思われる。

動詞に関するアプリケーション固有情報

ビジネス・オブジェクト定義には、サポートする各動詞に関するアプリケーション固有情報を含めることができます。アプリケーション固有情報は、その動詞がアクティブであるときにビジネス・オブジェクトの処理方法をアプリケーション固有のコンポーネントに指示します。

ビジネス・オブジェクト定義を作成または変更する方法

各コネクターには、他のアプリケーションに伝達するデータを定義する、ビジネス・オブジェクト定義のセットが必要です。アプリケーション固有のコンポーネントは、統合ブローカーにデータを送信する必要があるとき、それがサポートするビジネス・オブジェクト定義の 1 つから新しいビジネス・オブジェクトをインスタンス化します。コネクター構成プロセスのステップの 1 つに、サポートされるビジネス・オブジェクト定義を選択するステップがあります。ただし、まず最初に、アプリケーションに関するビジネス・オブジェクト定義を作成するか、または、別の方法で生成する必要があります。

ビジネス・オブジェクト定義の作成

アプリケーションに関するビジネス・オブジェクト定義を作成または取得する方法は、いくつかあります。

- アプリケーションに対応する Object Discovery Agent (ODA) が存在する場合には、それを使用してビジネス・オブジェクト定義を作成することができます。ODA は、アプリケーションの保管データの構造および編成を検証して、その検出結果に基づいてビジネス・オブジェクト定義を構成します。アプリケーションに対応する ODA が存在しない場合には、Object Discovery Agent Development Kit (ODK) を使用して ODA を作成します。
- Business Object Designer ツールを使用すると、ODA で生成されたビジネス・オブジェクト定義を変更するか、またはビジネス・オブジェクト定義を最初から組み立てて、ビジネス・オブジェクト定義を作成することができます。

その他の参考資料

「ビジネス・オブジェクト開発ガイド」には、ビジネス・オブジェクト定義の作成に関する詳細情報が記載されています。

さらに、多くのアダプターには、サンプル・ビジネス・オブジェクトが組み込まれています。組み込まれている場合には、サンプルは次のプロダクト・ディレクトリにあります。

Windows:

¥connectors¥ConnName¥Samples

UNIX:

ビジネス・オブジェクト定義の変更

追加のアプリケーション・データの取り込み、不要となったデータの収集停止、別のアプリケーションの変更への対応などの理由で、ビジネス・オブジェクト定義の変更が必要になる場合があります。「ビジネス・オブジェクト開発ガイド」で説明されている Business Object Designer ツールを使用すると、上記の変更を最も便利に行えます。

第 3 章 コネクタ

コネクタは、ローカル・ネットワーク上でアプリケーションと統合ブローカーの間を仲介します。コネクタは、SAP R/3 バージョン 4 などの、アプリケーションに固有のものにすることも、XML や WebSphere MQ などの、データ・フォーマットやプロトコルに固有のものにすることもできます。コネクタは、アプリケーション固有のコンポーネントとコネクタ・フレームワークで構成されます。

共通の振る舞いについては、すべてのコネクタで共有されており、アプリケーションやビジネス・オブジェクトと対話する方法のみが異なります。本章では、コネクタの共通の振る舞いについて紹介するとともに、コネクタの異なる領域についても紹介します。本章は、次の各セクションから構成されています。

- 『コネクタの開始』
- 28 ページの『イベント通知』
- 35 ページの『要求処理』
- 34 ページの『保証付きイベント・デリバリー』
- 40 ページの『コネクタ構成』
- 41 ページの『コネクタの開発』

一部の環境では、コネクタは「ブラック・ボックス」です。すなわち、その内部構成をあまり考慮せずに、コネクタを単にインストール、構成、管理、および使用することができます。ただし、カスタム・コネクタを作成する必要がある場合には、コネクタの振る舞いに関する詳細な知識が必要です。コネクタの作成・変更情報については、「コネクタ開発ガイド (Java 用)」または「コネクタ開発ガイド (C++ 用)」を参照してください。

コネクタの開始

コネクタは、開始スクリプトを使用して明示的に開始する必要があります。開始スクリプトは、コマンド行から、または Windows ショートカットから起動できます。

各コネクタには、コネクタの構成ファイルおよび各ビジネス・オブジェクト定義に関する個別の XML スキーマ文書を保持するローカル・リポジトリがあります。このリポジトリは、ローカル・ファイル・システム内の、アプリケーション固有のコンポーネントがインストールされるディレクトリです。

開始時に、コネクタは次の作業を行います。

1. ローカル・リポジトリからサポート・オブジェクト定義および構成プロパティをロードする。
2. アプリケーションに接続する。

注: コネクタの開始時に統合ブローカーを実行している必要はありません。ただし、統合ブローカーがアクティブになるまで、データは転送されません。

イベント通知

トリガー・イベントをアプリケーションから受け取るコネクタは、それらのイベントに関する知識を取得し、統合ブローカーに関連データを送信する必要があります。図 14 に、イベント通知に関するコネクタの対話を示します。

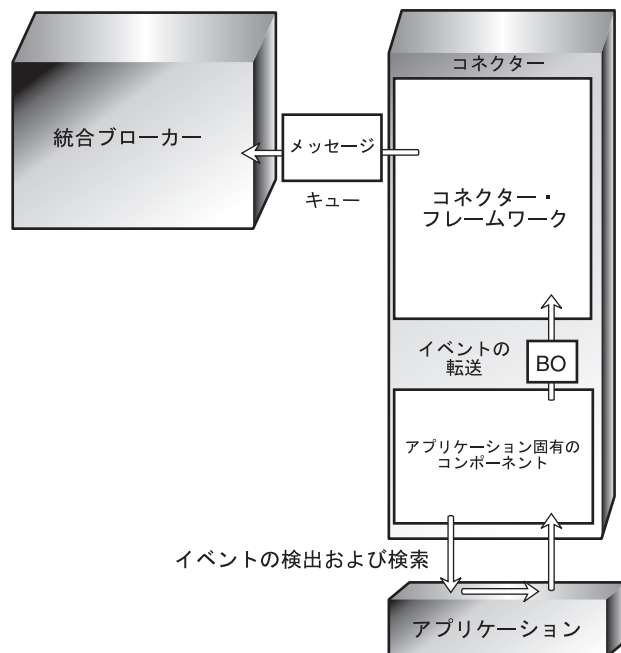


図 14. コネクタ内のイベント通知

アプリケーション固有のコンポーネントがイベントを検出および取得する手法は、コネクタによって異なります。ただし、アプリケーション固有のコンポーネントがコネクタ・フレームワークにイベントを送信する手法、およびコネクタ・フレームワークが統合ブローカーにそれらのイベントを引き渡す手法は、すべてのコネクタで標準です。

次のサブセクションでは、次の項目を中心に、大半のコネクタの操作に関する一般的概念について説明します。

- コネクタが、アプリケーション・イベント通知メカニズムを使用する仕組み
- コネクタが、イベントを検出および処理する仕組み

ここでの説明は、特定のコネクタのインプリメンテーションを記述するものではありません。

アプリケーションのイベント通知メカニズムのセットアップ

コネクタにとって、アプリケーション・イベントとは、WebSphere Business Integration Adapters のビジネス・オブジェクト定義に関連付けられたアプリケーション・エンティティのデータに影響を及ぼす何らかの操作のことです。アプリケーションには別のタイプのイベントがあります。例えば、マウス・クリックはアプリケーションのウィンドウ・システムまたはフォーム・インターフェースに対するイベントです。ただし、コネクタが関与するのは、事前定義されたデータ・レベ

ル・イベントのサブセットであって、アプリケーションのデータ・ストアの内容を作成、更新、削除、その他の方法で影響するイベントのみです。

アプリケーションによっては、イベントを明示的にトラップして報告し、使いやすいイベント管理および構成可能なイベント・テキストを提供しているものがあります。個別的な報告可能イベントの概念がなく、なんらかの発生によりデータベースを更新しても、なにも通知しないことのあるアプリケーションもあります。

WebSphere Business Integration Adapters は、両方のタイプのアプリケーションに対応したコネクタを備えています。

ほとんどのコネクタでは、コネクタの使用に関して**イベント通知メカニズム**をセットアップするために、アプリケーション構成が必要です。イベント通知メカニズムでは、アプリケーション内で発生する操作を順番に並べたリストを保持しています。このリストは、物理的には、アプリケーション・イベント・キュー、Eメール受信箱、またはデータベース表の形式をとります。

コネクタが使用するイベント通知メカニズムには、どのようなタイプがあるのでしょうか。以降のセクションでは、一般的な方法をいくつか紹介します。

アプリケーションにイベント・サポートがある場合

アプリケーションがイベント・ベースの場合には、コネクタなど、クライアント・アプリケーションが使用するためのイベント通知インターフェースがおそらく用意されているはずです。また、アプリケーションでは、イベント・レポートのテキストを構成できる場合もあります。そのようなアプリケーションの場合、コネクタのイベント通知メカニズムのセットアップは、通常のアプリケーション・セットアップ・タスクです。

例えば、あるアプリケーションで、特定のタイプのイベントの発生時に実行するスクリプトをインストールでき、そのスクリプトがイベント受信箱に通知を入れることができる場合を想定します。このアプリケーションに対応したコネクタをインストールするには、そのコネクタのユーザー・アカウントを作成し、追跡するイベントを処理するスクリプトを作成または取得し、スクリプトをインストールし、各スクリプトを起動するイベントのタイプを指定し、受信箱を作成します。以上の作業を完了すると、アプリケーション固有のコンポーネントは、新しいイベントの発生をチェックするために受信箱の内容を定期的に検索します。

30 ページの図 15 に、イベント受信箱が組み込まれたアプリケーション構成を示します。

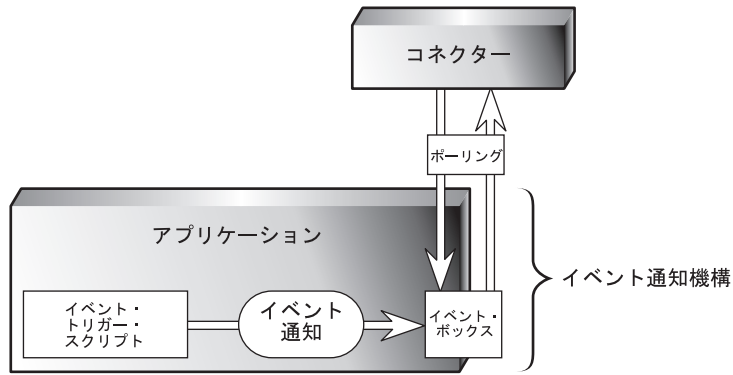


図 15. 例: イベント通知でのイベント受信箱の使用

特定の操作が実行されたときに E メール・メッセージの生成やイベント・キューへの書き込みができる内部ワークフロー・システムが用意されているアプリケーションもあります。31 ページの図 16 に、ビジネス・オブジェクトおよびイベントが定義されている、独自のビジネス・オブジェクト・リポジトリを持つアプリケーションを図示します。この図で、Customer はビジネス・オブジェクトで、Create、Delete、および Update は、このビジネス・オブジェクトに関連付けられているイベントのタイプです。

Customer.Update などのビジネス・オブジェクト・イベントが発生すると、イベントはワークフロー・システムに送信されます。続いてワークフロー・システムは、アプリケーション・データベース内のイベント表にエントリーを作成します。

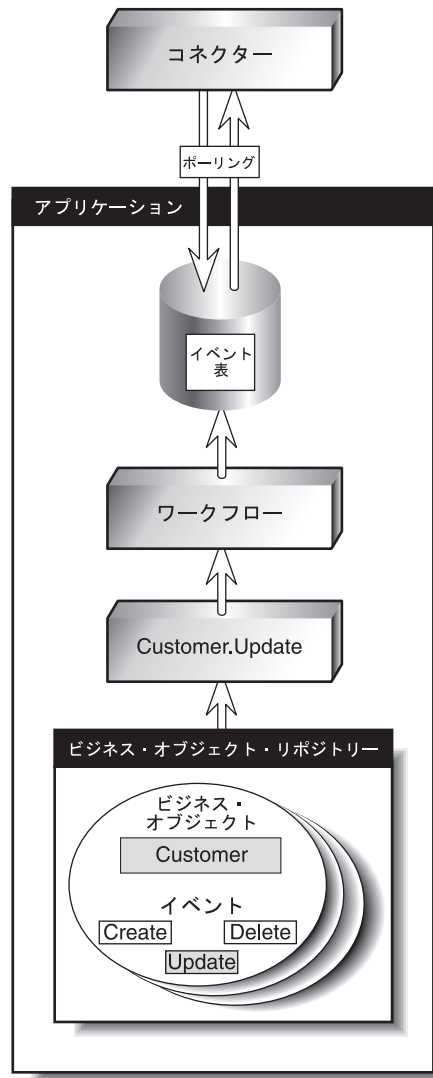


図 16. 例: イベント通知でのアプリケーション・ワークフローの使用

アプリケーションにイベント・サポートがない場合

コネクターがアプリケーション・イベントと対話するときの推奨メソッドは、アプリケーションの API を介する方法です。API には、アプリケーションのデータ・モデルとロジックを強制するフレームワークが用意されているからです。ただし、アプリケーション API によっては、イベント通知に対する固有のサポートを用意していないものもあります。

コネクターがそのようなアプリケーションからイベント通知を受け取ることができる方法の 1 つは、アプリケーション・データベースと対話することです。例えば、行に対する更新を検出するトリガーを、従業員表にセットアップすることができます。更新が行われると、トリガーは更新に関する情報を、コネクターの展開時に作成した表に挿入します。イベント表に新たに表示される各行は、イベント通知を意味します。コネクターは SQL クエリーを使用して、表から新規のイベントを検索することができます。

図 17 に、上記の方法を図示します。

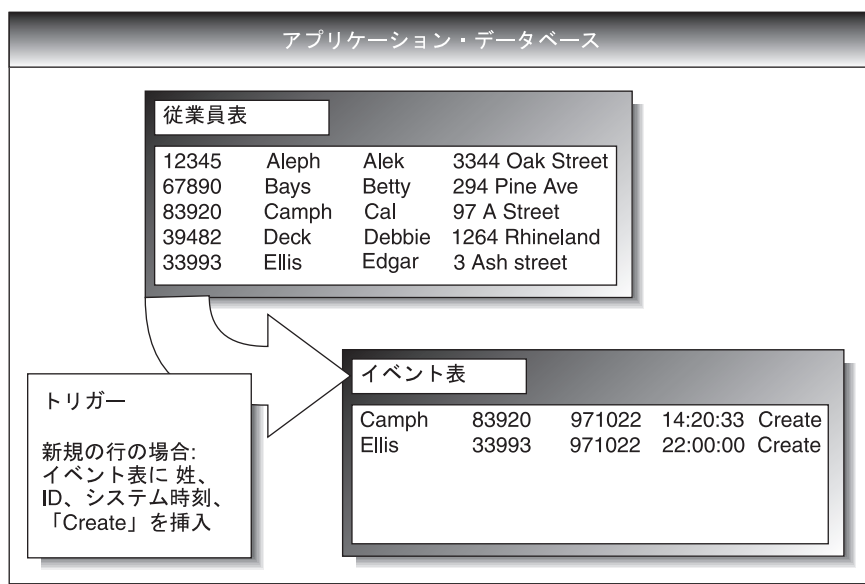


図 17. 例: イベント通知でのデータベースの使用

図 17 では、アプリケーション・データベースに、従業員表 (Employee Table) にレコードが作成される際のトリガーが用意されています。トリガーは、アプリケーションが新規のレコードを挿入するたびに、イベント表に 1 行を作成します。行には、新規従業員レコードのキー値 (ラストネームと従業員 ID)、システム時刻、およびイベント・タイプ Create が含まれます。

イベントの検出

コネクタのアプリケーション固有のコンポーネントは、そのイベント通知メカニズムを介してアプリケーション・イベントの情報を得ます。このメカニズムの最も一般的なものは、イベント・ストア内の新規イベントに対するポーリングです。ポーリング・メソッドは、コネクタが使用するイベント通知メカニズムに基づき、そのアプリケーションに固有のものです。

ポーリングは構成可能です。Connector Configurator ツールを使用してコネクタを構成する際に、次のことが行えます。

- アプリケーション固有のコンポーネントがアプリケーションをポーリングする頻度の調整
- アプリケーション固有のコンポーネントがアプリケーションをポーリングする時間帯の指定

大半のコネクタで、ポーリング呼び出しごとに処理するイベント数を指定することもできます。

アプリケーション固有のコンポーネントが、アプリケーションをポーリングする必要があるのは、別のアプリケーションがそのアプリケーションのイベントの情報を必要としている場合のみです。あるアプリケーションがイベントのソースではない場合は、Connector Configurator を使用して、ポーリング頻度を「no」に設定して、アプリケーション固有のコンポーネントのポーリングを停止することができます。

Connector Configurator の詳細については、「コネクタ開発ガイド」、または配置するアダプター用のアダプター・ユーザーズ・ガイドを参照してください。

イベントの処理

イベントを検出すると、コネクターのアプリケーション固有のコンポーネントは次のことを行います。

- アプリケーション・イベントをビジネス・オブジェクト定義に関連付け、そのビジネス・オブジェクトのインスタンスを作成する
- ビジネス・オブジェクトに動詞およびキー値の属性を設定する
- アプリケーション・データを検索し、ビジネス・オブジェクトの属性を取り込む
- ビジネス・オブジェクトをコネクター・フレームワークに送信する
- イベントをアーカイブする (オプション)

アプリケーション・イベントをビジネス・オブジェクト定義と関連付ける方法

アプリケーション固有のコンポーネントは、イベントを検索するときに、そのイベントを表すビジネス・オブジェクト定義と動詞を判別する必要があります。

アプリケーション固有のコンポーネントは、イベント・テキストを使用して、ビジネス・オブジェクト定義と動詞にイベントに関連付けます。表 4 を参照してください。

表 4. イベント・テキストおよびビジネス・オブジェクトの形成

アプリケーション・イベント 内のデータ型	例	用途
アプリケーション・エンティティ・タイプ 実行された操作	Customer, Part, Item Create, Update, Delete	関連付けられたビジネス・オブジェクト定義を判別する ビジネス・オブジェクトのアクティブな動詞を判別する

例えば、コネクターは次のイベント・テキストを Employee.Create ビジネス・オブジェクトに関連付けます。

```
1997.10.19.12:50.22 employee created lname="como" id="101961"
```

イベント・テキストは、次のもので構成されます (左から右へ)。

- イベントを一意的に識別するのに役に立つタイム・スタンプ
- アプリケーション・エンティティ「employee」
- 操作「created」
- 従業員のラストネームと従業員 ID、つまりアプリケーション固有のコンポーネントが残りの従業員情報を検索するのに使用する固有の ID (キー値)。

上記は簡単な例です。他のタイプのイベント・テキストでは、アプリケーション固有のコンポーネントによる処理をさらに必要とする場合があります。

アプリケーション固有のビジネス・オブジェクトの作成

イベントのビジネス・オブジェクト定義をサポートするようにコネクターが構成されている場合には、アプリケーション固有のコンポーネントはビジネス・オブジェクトを作成し、キー値を使用してアプリケーション・データを検索し、ビジネス・オブジェクトにアプリケーション・データを入力します。ビジネス・オブジェクトを作成するプロセスについては、37 ページの『ビジネス・オブジェクトの構成とデストラクション』を参照してください。

アプリケーション固有のビジネス・オブジェクトのコネクター・フレームワークへの送信

アプリケーション固有のコンポーネントは、ビジネス・オブジェクトをコネクター・フレームワークに送信します。この場合、ビジネス・オブジェクトに含めて搬送される情報の受信側アプリケーションの ID を知っている必要はありません。

イベントのアーカイブ

アプリケーション・イベント・アーカイブは、トラブルシューティングおよびレコードの保持に役立ちます。イベント・アーカイブには、各イベントに関する、次に示すような状況情報が含まれます。

- 統合ブローカーに正常に送信されたこと
- 処理が失敗したこと

アプリケーションにイベント・アーカイブ・フィーチャーが用意されている場合には、コネクターは一般にそれを使用します。イベントのアーカイブをサポートしないアプリケーションのコネクターには、独自のイベント・アーカイブが備わっている場合があります。例えば、コネクターのイベント通知メカニズムが 32 ページの図 17 で図示するようなデータベース・メカニズムである場合には、データベース・トリガーは、コネクターの展開時に作成するアーカイブ表に、削除されたイベントをコピーすることができます。

保証付きイベント・デリバリー

保証付きイベント・デリバリーは、発生する可能性のあるいかなるサービス中断が発生した場合でも、金融取引などの重要なイベントが正しく処理されるようにします。この機能により、コネクター・フレームワークは、各イベントの検出と、ソース・コネクターのイベント・ストアから宛先コネクターの要求キューへの送信が、確実に 1 度のみ行われるようにすることができます。

保証付きイベント・デリバリーが実行されない場合には、コネクターがイベントを検出した時点から、必要なすべての処理が完了する時点までの間、障害が発生する可能性のある短い期間が生じます。この期間内に障害が発生すると、イベントは送信されますが、イベント・レコードはイベント・ストアに残ったままになります。コネクターが再始動すると、コネクターはイベント・ストアでそのイベント・レコードを検出し、これを新しいイベントとして扱うため、このイベントは 2 回送信されてしまいます。

保証付きイベント・デリバリー、および配置するアダプターでこの機能を使用可能にする方法についての詳細は、アダプターのユーザーズ・ガイドを参照してください。

要求処理

また、統合ブローカーは、WebSphere MQ メッセージの形式で、コネクタ・フレームワークに要求を送信します。要求では、宛先アプリケーションに対して次のいずれかを行うことを依頼できます。

- ビジネス・データを検索し、それを統合ブローカーに戻すこと。
- アプリケーションのデータ・ストアを更新すること。

例えば、統合ブローカーは、契約を削除する、パーツを更新する、顧客を作成するなどの要求メッセージをコネクタに送信することがあります。

コネクタ・フレームワークは、統合ブローカーの要求を受け取ると、メッセージを適切なビジネス・オブジェクトに変換し、それをアプリケーション固有のコンポーネントに転送します。例えば、統合ブローカーが契約を削除する要求を送信する場合、アプリケーション固有のコンポーネントはその要求を `Customer.Delete` ビジネス・オブジェクトの形式で受け取ります。アプリケーション固有のコンポーネントは、そのビジネス・オブジェクトをアプリケーション要求 (通常、一組の API 呼び出し) に変換し、続いて、必要に応じて結果を戻します。

図 18 に、統合ブローカーからのメッセージの処理に関するコネクタの対話を図示します。

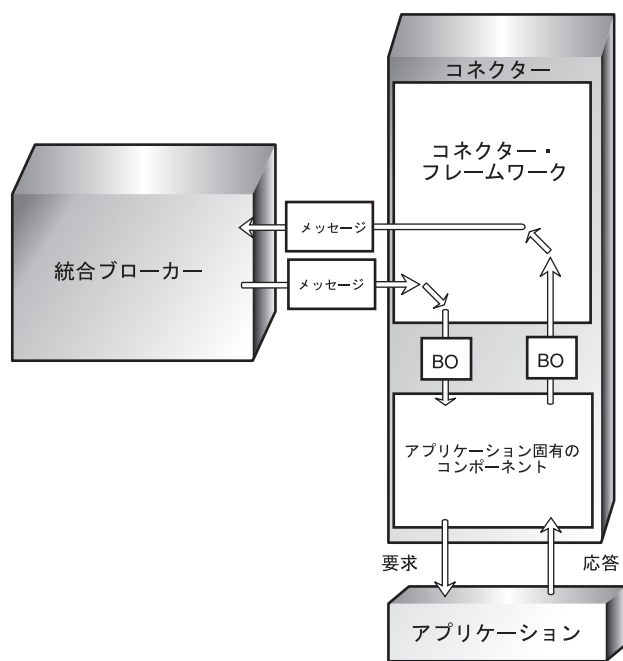


図 18. 要求の処理のためのコネクタの対話

アプリケーション固有のコンポーネントは、要求を受け取ると、次の 3 つのタイプの情報に基づいて要求の処理方法を決定します。

- ビジネス・オブジェクトの動詞
- ビジネス・オブジェクト定義自体に含まれ、ビジネス・オブジェクトの構成またはデコンストラクションで使用されるメタデータ
- 動詞に関するアプリケーション固有情報

以降の各トピックで、上記の要因について説明します。

動詞ベースの処理

コネクターのアプリケーション固有のコンポーネントは、そのアプリケーションのロジックおよび API に従って、要求に含まれている Create、Retrieve、Update、または Delete 動詞に対して応答します。さまざまなコネクターのアプリケーション固有のコンポーネントは、同じタイプの要求をそれぞれに異なる方法で処理しますが、結果は論理的に同じです。

一部のコネクターでは、要求に含まれる動詞のタイプとは無関係に、ビジネス・オブジェクトでの操作の実行に必要なメソッドは 1 つのみです。ただし、多くのコネクターでは、それぞれの動詞ごとに、異なるメソッドが必要です。

アプリケーション固有のコンポーネントは、要求を受け取ると、そのアプリケーションのメソッドでビジネス・オブジェクトのアクティブな動詞に合致するものを起動します。例えば、アプリケーション固有のコンポーネントは、AppAEmployee.Update ビジネス・オブジェクトを受け取ると、AppAEmployee オブジェクト上で Update メソッドを起動します。Update メソッドは、更新を行うためにアプリケーションと対話します。

図 19 に、動詞の処理メソッドをいくつか示します。

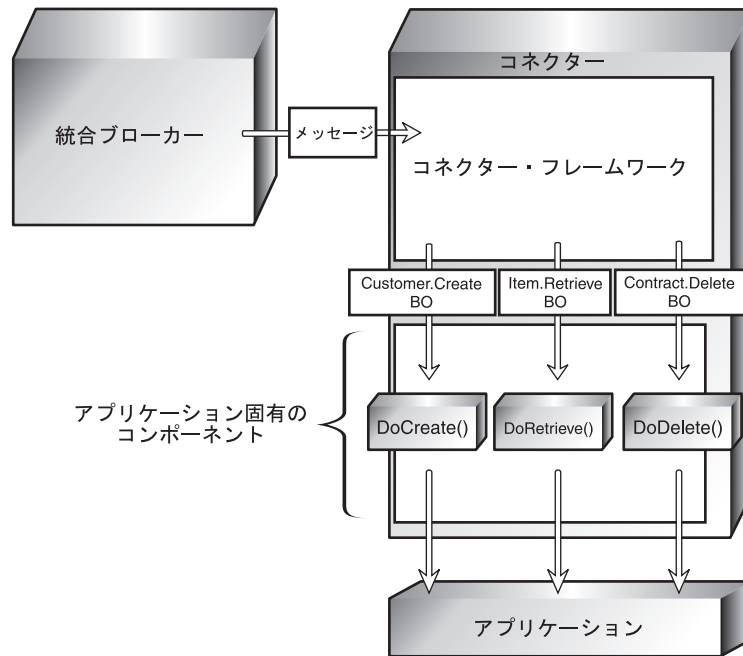


図 19. 要求の処理

図 19 で、コネクターは、Customer.Create、Item.Retrieve、または Contract.Delete 要求を受け取ると、それぞれに応じた DoCreate()、DoRetrieve()、または DoDelete() メソッドを起動します。各メソッドは、指定の操作を実行するために、そのアプリケーションに対応した適切なコマンドを作成します。

ビジネス・オブジェクトの構成とデコンストラクション

コネクタのアプリケーション固有のコンポーネントは、次に示すように、ビジネス・オブジェクトを構成およびデコンストラクションすることにより、そのイベント通知タスクおよび要求処理タスクを実行します。

- アプリケーション固有のコンポーネントは、それが統合ブローカーに送信しなければならないイベントを検出すると、そのイベントを表すビジネス・オブジェクトを構成します。
- アプリケーション固有のコンポーネントは、要求を表すビジネス・オブジェクトを統合ブローカーから受け取ると、そのビジネス・オブジェクトをデコンストラクションして、アプリケーション要求を作成します。

ビジネス・オブジェクト・メタデータおよびコネクタの動作

コネクタによる、アプリケーション・イベントからビジネス・オブジェクトへの変換およびビジネス・オブジェクトからアプリケーション要求への変換は、ビジネス・オブジェクトの設計時に定義されるデータ定義 (メタデータ) によって駆動されます。

アプリケーション固有のコンポーネントとビジネス・オブジェクト・メタデータは、組み合わせて使用するよう設計されます。アプリケーション固有のコンポーネントとそのビジネス・オブジェクトの設計は、特定の機能がソフトウェアまたはハードウェアのいずれかによってインプリメントできる、コンピューター装置の設計に類似しています。開発者は、パフォーマンス、拡張性、および他の問題を考慮して、重要な機能の実装場所を決定します。

ビジネス・オブジェクト定義には、属性のタイプ、サイズ、およびデフォルト値を指定するプロパティが組み込まれます。また、ビジネス・オブジェクト定義には、ビジネス・オブジェクトの処理方法に関しての、アプリケーション固有のコンポーネントに対する指示が含まれるアプリケーション固有のプロパティも組み込まれます。

具体例については、183 ページの『付録 G. Visual Test Connector の使用』を参照してください。ここでは、顧客を表すビジネス・オブジェクトの属性に関するアプリケーション固有情報が例示されています。表 5 に、上記の例を一部示します。

表 5. 属性に関連したアプリケーション固有情報の名前と値のペアのパラメーター例

パラメーター	説明
TN=TableName	データベース表の名前です。
CN=col_name	この属性のデータベース列の名前です。
FK=[..]fk_attributeName]	外部キー・プロパティの値により親子関係が定義されます。
UID=AUTO	このパラメーターは、ビジネス・オブジェクト用の固有 ID を生成するよう、またこの属性に値をロードするようにコネクタに通知します。

表 5. 属性に関連したアプリケーション固有情報の名前と値のペアのパラメーター例 (続き)

パラメーター	説明
CA=set_attr_name	Copy Attribute (属性のコピー) プロパティは、ある属性の値を別の属性にコピーするようコネクタに指示します。set_attr_name が現在の個々のビジネス・オブジェクト内の別の属性の名前に設定されている場合には、コネクタは、Create 操作でビジネス・オブジェクトをデータベースに追加する前に、指定された属性の値を使用して、この属性の値を設定します。

アプリケーション固有のコンポーネントは、ビジネス・オブジェクトを処理するとき、定義を読み取り、アプリケーション固有情報を使用してアプリケーション要求を作成します。ビジネス・オブジェクトの詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

アプリケーション固有のコンポーネントはメタデータ主導型です。したがって、アプリケーション固有のコンポーネントのアクションは、ビジネス・オブジェクト定義にあるアプリケーション固有情報および他のメタデータによって制御されます。アプリケーション固有のコンポーネントには、サポートするビジネス・オブジェクトのタイプごとのハードコーディング命令はありません。アプリケーション固有のコンポーネントはメタデータ主導型であるため柔軟で、対応するアプリケーション・データがコネクタのメタデータ構文によって正確に記述できるかぎり、すべての新しいビジネス・オブジェクト定義、または変更したビジネス・オブジェクト定義すべてを自動的にサポートします。

ビジネス・オブジェクト作成の例

アプリケーション固有のコンポーネントがビジネス・オブジェクトをその定義から作成する手順を次に説明します。

1. ビジネス・オブジェクト定義をそのローカル・リポジトリから取得し、それを使用してビジネス・オブジェクト・インスタンスを作成します。
2. アプリケーション固有のコンポーネントは、属性ごとにビジネス・オブジェクト・インスタンスをループし、アプリケーション固有情報を使用して、アプリケーション・エンティティを取得するための API 呼び出しの作成、または照会のビルドを行います。
3. アプリケーション固有のコンポーネントは、要求をアプリケーションに送信し、結果を検索します。
4. アプリケーション固有のコンポーネントは、結果をループし、AppSpecificInfo の値を使用して、それぞれのビジネス・オブジェクト属性を表す検索値を判別します。

39 ページの図 20 は、定義からビジネス・オブジェクトを作成するアプリケーション固有のコンポーネントの一例です。アプリケーション固有のコンポーネントは、キー値 (項目番号) が 123 である項目を含むアプリケーション・イベントを検索しています。アプリケーション固有のコンポーネントは、ビジネス・オブジェクト定

義から Item ビジネス・オブジェクトを作成する必要があります。この定義には、Group、Description、Price、および ItemNum という 4 つの属性が含まれます。

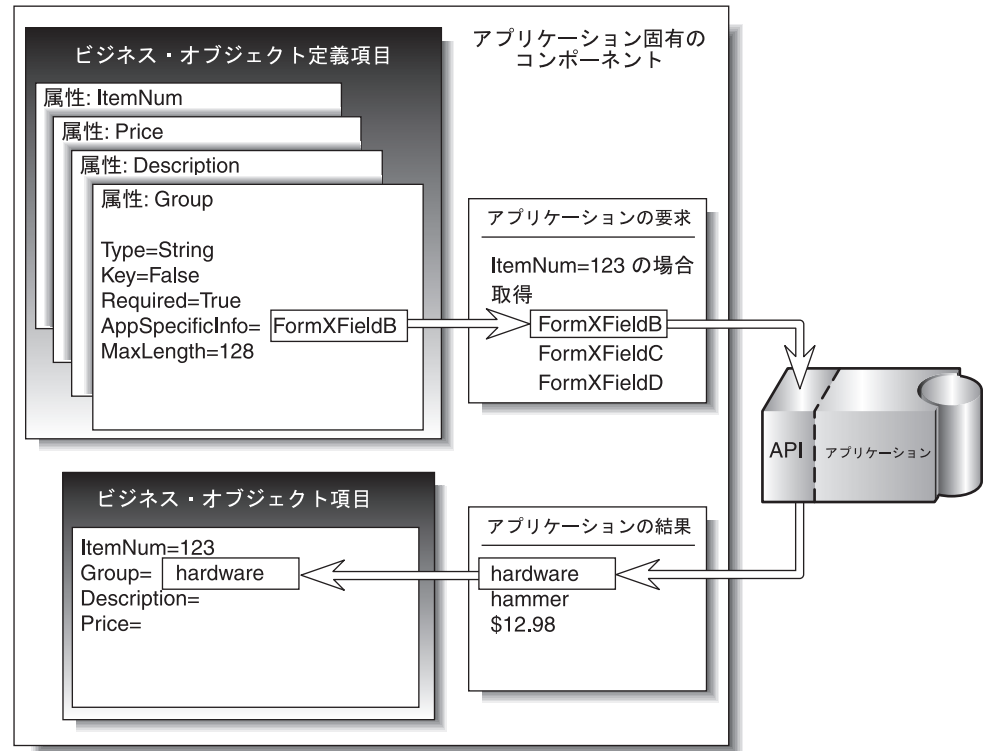


図 20. コネクター内でのビジネス・オブジェクトの作成

項目を識別するための項目番号 123 を使用して、アプリケーション固有のコンポーネントは残りの属性の値を検索します。アプリケーション固有情報は、必須データの用紙とフィールド ID を識別します。

例えば、FormXFieldB は Group データを識別します。アプリケーション固有のコンポーネントは、項目 ID 123 に関して用紙 X のフィールド B の値を要求します。続いて、アプリケーション固有のコンポーネントは戻り値「hardware」を使用して、ビジネス・オブジェクトの Group 属性の値を入力します。

デコンストラクションのプロセスは、上記と正反対です。アプリケーション固有のコンポーネントは、ビジネス・オブジェクト定義を使用して、コンポーネントが受け取ったビジネス・オブジェクトに含まれるデータに基づいてアプリケーション要求を作成する方法を決定します。

動詞に関するアプリケーション固有情報

ビジネス・オブジェクト定義内のそれぞれの動詞には、それに関連付けられた、アプリケーション固有情報を持たせることができます。動詞の持つアプリケーション固有情報の内容は、その特定のコンネクターに固有です。動詞に関するアプリケーション固有情報は、コンネクターのアプリケーション固有のコンポーネントに、ビジネス・オブジェクトを処理するための追加の指示を与えます。

例えば、ビジネス・オブジェクト定義内の Retrieve 動詞に関するアプリケーション固有情報では、そのアプリケーション固有のコンポーネント内の Retrieve メソッドに特別な入力引き数が与えられる場合があります。

一例として、MyApp アプリケーションで InventoryItem に関する情報が次の 3 つの用紙で表示される場合を想定します。

- InventoryItem-New
- InventoryItem-Change
- InventoryItem-Remove

MyApp 用のアプリケーション固有のコンポーネントは、インベントリー項目で操作を行うとき、その操作に対応する正しいフォームを参照する必要があります。

InventoryItem ビジネス・オブジェクト定義では、動詞に関するアプリケーション固有情報を使用してフォーム名を格納することができます。

動詞固有のメソッドと、それらメソッドへのアプリケーション固有の入力とを組み合わせると、アプリケーション固有のコンポーネントに固有の処理の指示を与えることができます。

コネクタ構成

コネクタを使用するにはその前に、Connector Configurator を使用して、次のものを含む構成ファイルを定義する必要があります。

- コネクタがサポートするビジネス・オブジェクト
- コネクタの構成プロパティ

コネクタ構成プロパティには、**標準プロパティ**と**アプリケーション固有のプロパティ**という、2 つのタイプがあります。

標準プロパティは、すべてのコネクタに適用され、次に示すような情報を指定します。

- コネクタで使用する統合ブローカー
- コネクタのローカル・リポジトリのロケーション
- コネクタによって使用されるキューを管理するキュー・マネージャーの名前

125 ページの『付録 C. コネクタの標準構成プロパティ』に、WebSphere Message Broker とともに機能するコネクタの標準の構成プロパティを示します。

アプリケーション固有のプロパティでは、特定のアプリケーション固有のコンポーネントがアプリケーションとのセッションを確立するために必要とする値が指定されます。また、アプリケーション固有のコンポーネントの処理の振る舞いの一部について、指図します。次に、さまざまなコネクタでのアプリケーション固有のプロパティの例を示します。

- アプリケーションを実行するマシンの名前または IP アドレス
- アプリケーション・データベースの名前
- アプリケーションへのアクセスでコネクタが使用する必要がある、ログイン・ユーザー ID とパスワード

- イベント受信箱の名前
- ポーリング・イベントごとの、検索するイベントの数

一部のコネクタ構成プロパティは、コネクタの開始時にコマンド行で設定することもできます。コマンド行で設定するプロパティは、コネクタの構成ファイルで設定された値をオーバーライドします。

コネクタの構成の詳細については、106 ページの『コネクタの構成』を参照してください。コネクタの開始の詳細については、67 ページの『コネクタの始動』を参照してください。

コネクタの開発

コネクタを変更または作成するには、クラス・ライブラリー、ヘッダー・ファイル、およびサンプルを使用して、アプリケーション固有のコンポーネントを作成します。続いて、Connector Configurator を使用して、ビジネス・オブジェクト定義を作成および変更し、コネクタ・リポジトリ定義を作成します。

コネクタ開発には、アプリケーション固有のコンポーネントと特定のアプリケーションとの関係の定義が含まれます。アプリケーション固有のコンポーネントの機械語コーディングは、通常、比較的簡単な処理です。最も難しいのは次のタスクです。

- アプリケーションのイベント通知メソッドの設計
- ビジネス・オブジェクト定義の定義
- ビジネス・オブジェクトとアプリケーション・オブジェクトとの関係の定義

コネクタのアーキテクチャー、変更、および開発の詳細については、「コネクタ開発ガイド (Java 用)」または「コネクタ開発ガイド (C++ 用)」を参照してください。

次の章、43 ページの『第 4 章 データ移送および統合ブローカー』では、ビジネス・インテグレーション・システムのコンポーネント間での情報の伝達に使用する、メカニズムとプロトコルについて詳細に説明します。

第 4 章 データ移送および統合ブローカー

WebSphere Business Integration Adapter が統合ブローカーとしてメッセージ・ブローカーを使用する場合は、コネクタ・フレームワークと統合ブローカー間の通信移送機構として、WebSphere MQ のキューおよび標準の Java Messaging Service (JMS) ソフトウェアが使用されます。WebSphere MQ Queue マネージャーが管理する定義済みキューに対して、コネクタがメッセージの読み書きを行います。統合ブローカーは、同じメソッドを用いてコネクタのコネクタ・フレームワークとの通信を行います。

本章では、WebSphere Business Integration システムのアプリケーション間データ交換に用いられるメッセージング・インターフェースおよびプロトコルについて詳細に説明します。本章は、次の各セクションから構成されています。

- 『統合ブローカーの役割』
- 44 ページの『非同期データ移送』
- 44 ページの『同期データ移送』
- 44 ページの『メッセージ交換のインターフェース』

統合ブローカーの役割

WebSphere Message Broker を使用するビジネス・インテグレーション・システムにおいて、統合ブローカーが行う作業には、アプリケーション間のメッセージのルーティング、およびメッセージ・フローを用いたメッセージ処理の 2 つがあります。具体的には、統合ブローカーは以下の処理を実行します。

- 送信アプリケーションのコネクタのアウトバウンド・キューからメッセージを受信する。
- メッセージを処理するため、それをメッセージ・フローに渡す。各キューごとに定義された 1 つのメッセージ・フローが、そのキューにあるすべてのメッセージを処理します。ただし、このメッセージ・フローには、処理対象となるメッセージのタイプごとに、異なる処理命令が混在していても支障はありません。各メッセージには、そのデータの種類に関する情報が入っているヘッダーが付加されています。メッセージ・フローは、ヘッダー情報を使用して、そのメッセージに対してどの処理命令のグループを適用すべきかを判別します。
- 宛先アプリケーションのコネクタのインバウンド・キューへメッセージを転送する。

統合ブローカーとコネクタ間の通信は、**非同期**または**同期**の、いずれかになります。

非同期データ移送

非同期メッセージング移送を使用するプログラムでは、接続の確立やメッセージ(個別単位ごとのデータ) 待ちを必要としません。各プログラムは、メッセージング・サービスと非同期で対話することによってメッセージの送受信を行います。メッセージング・サービスは、宛先プログラムが使用不可の場合にメッセージを保管したり、そのプログラムが使用可能になるまで再試行したりすることによって、メッセージの送達が確実にできるようにします。 応答の即時性よりも確実な送達の方を重視する場合には、この非同期メッセージングという通信プロトコルが有効です。

同期データ移送

現在では、製品に含まれるアダプターで同期データ移送を使用しているものはありませんが、カスタマイズされたコネクタであればこの機能が使用できます。

同期メッセージング移送を使用するプログラムでは、統合ブローカーへの要求メッセージを同期要求キューに送り、統合ブローカーからの応答メッセージを同期応答キューで受け取ります。応答メッセージの相関 ID によって、どの要求メッセージに回答しているかが識別されます。一般に回答メッセージは、ビジネス・オブジェクト・メッセージと、要求メッセージが正常に処理されたかどうかを示す状況表示で構成されています。

メッセージ交換のインターフェース

このセクションでは、メッセージ自体と、メッセージの処理に必要な情報とを送信するためにコネクタ・フレームワークおよび統合ブローカーで使用されるメッセージング・インターフェースについて説明します。

ビジネス・インテグレーション・システムにおいては、何種類ものメッセージが交換されています。メッセージの異なるタイプを識別するため、またメッセージを正しく処理して送信するために、必要不可欠な情報が、各メッセージのメッセージ・ヘッダーおよびメッセージ・ディスクリプターに格納されています。ビジネス・インテグレーション・システム向けに作成したメッセージ・フローは、処理要求のあったメッセージを認識し、正しく管理するために、下記の情報を使用します。

以下のタイプのメッセージが渡されます。

- イベント・デリバリー・メッセージは、コネクタ・フレームワークによって WebSphere Message Broker に送信され、送信元アプリケーションで発生したイベントを通知します。
- 要求メッセージは、データ要求を伝達するためにコネクタ・フレームワークと WebSphere Message Broker の間で交換されます。
- 応答メッセージは、データ要求に回答するためにコネクタ・フレームワークと WebSphere Message Broker の間で交換されます。
- 管理メッセージは、管理コマンドを伝達するためにコネクタ・フレームワークと WebSphere Message Broker の間で交換されます。

メッセージ・フォーマット

コネクタ・フレームワークと統合ブローカーの間で交換されるメッセージは、データ・ハンドラーが以下の項目に基づいてフォーマット設定します。

- コネクタの構成ファイル内の `WireFormat` 標準プロパティ
- メッセージ本文のフォーマットを詳述する XML スキーマ
- メッセージの内容: ビジネス・オブジェクトまたは管理メッセージ
- メッセージの発信元および宛先

各メッセージには、メッセージ・ディスクリプター (MQMD)、メッセージ・ヘッダー (MQRFH2)、およびメッセージ本文という 3 つのコンポーネントが含まれています。

メッセージ・ディスクリプター

WebSphere MQ メッセージ・ディスクリプター (MQMD) には、メッセージ ID およびメッセージの処理に必要な情報が入っています。

メッセージ・ヘッダー

MQRFH2 メッセージ・ヘッダーは、メッセージ内容に関連した JMS 固有のデータを搬送します。また、直接 JMS と関連付けのない追加情報も搬送できます。メッセージ・ヘッダーには次のフォルダーが含まれています。

- `<mcd>` フォルダには、メッセージの「形状」または「フォーマット」を記述するプロパティが含まれています。例えば `Msd` プロパティには、フォーマットが「テキスト」、「バイト」、「ストリーム」、「マップ」、「オブジェクト」、または「NULL」であることが記述されています。
- `<jms>` フォルダは、JMS ヘッダー・フィールドと、MQMD では完全に表現できない JMS プロパティを移送するために使用します。JMS を使ってインプリメントされ、コネクタ・フレームワークによって送信されるメッセージでは、常にこのフォルダが関係しています。ただし、ビジネス・インテグレーション・システムでは、このフォルダは無関係なため、メッセージをコネクタ・フレームワークに送信する際に統合ブローカーによって除外されます。
- `<usr>` フォルダは、メッセージに関連付けられたアプリケーションで定義済みのプロパティを移送するために使用します。アプリケーションが何らかのアプリケーションで定義済みのプロパティを設定した場合にのみ、このフォルダが存在します。ビジネス・インテグレーション・システムでは、このフォルダは、応答メッセージのリターン状況情報を送信するために使用します。この後の各表では、このフォルダが必須となるメッセージのタイプについて紹介しています。

メッセージ本文

メッセージ本文は、そのメッセージ用に指定された XML スキーマの内容に従ってフォーマット設定されます。メッセージをフォーマット設定するために正しい XML スキーマをデータ・ハンドラーが検出および使用するには、次の 3 つの名前が一致しなければなりません。

- コネクタのリポジトリに格納されている XML スキーマの名前。
- WebSphere Message Broker のメッセージ・リポジトリ内にインポートされ、メッセージ・セット定義として保管されている XML スキーマの名前。

- メッセージの MQRFH2 メッセージ・ヘッダーにある `messagetype` の内容。

コネクター・フレームワークと WebSphere Message Broker 間で交換される各種タイプのメッセージに使用するメッセージ・フォーマットおよび個々のプロパティの設定値は、115 ページの『付録 A. WebSphere MQ のメッセージ・フォーマット』にリストしています。

メッセージ・キュー

ここでは、コネクターと共に使用するための定義および構成が必要な WebSphere MQ キューについて説明します。

必要なキューのタイプ

コネクター・フレームワークと WebSphere Message Broker 間でのビジネス・オブジェクト・メッセージおよび管理メッセージの移送には、各 WebSphere MQ メッセージ・キューのセットを個別に使用します。以下のプロパティを用いてキューを定義する必要があります。

- **DeliveryQueue:** コネクター・フレームワークから WebSphere Message Broker にイベント・デリバリー・メッセージを配信します。
- **RequestQueue:** WebSphere Message Broker からコネクター・フレームワークに要求メッセージを配信します。
- **FaultQueue:** コネクター・フレームワークから WebSphere Message Broker に障害メッセージを配信します。コネクター・フレームワークは、返信先キューにメッセージを配置することができないとき、このキューにメッセージを配置します。
- **SynchronousRequestQueue:** 同期応答を要求する要求メッセージを、コネクター・フレームワークから WebSphere Message Broker へ配信します。このキューは、コネクターが同期実行方式を使用する場合にのみ必要です。
- **SynchronousResponseQueue:** 同期要求に応答して送信された応答メッセージを、WebSphere Message Broker からコネクター・フレームワークへ配信します。このキューは、コネクターが同期実行方式を使用する場合にのみ必要です。
- **AdminInQueue:** WebSphere Message Broker からコネクター・フレームワークに管理メッセージを配信します。
- **AdminOutQueue:** コネクター・フレームワークから WebSphere Message Broker に管理メッセージを配信します。

コネクター構成時に、コネクターの構成ファイル内で各キューの名前を標準プロパティとして指定します。

キュー・マネージャー

コネクターは、1 つのキュー・マネージャーを使用して、キューとのすべての対話を管理しています。コネクターの構成ファイル内の標準プロパティには、始動時にコネクターが必要とするキュー・マネージャー情報が入っています。コネクターはこの情報を使用して、WebSphere Message Broker との通信に用いるキュー・マネージャーとの接続を確立します。

WebSphere Business Integration システムでは、いくつかのキュー・マネージャーとキュー構成をサポートしています。コネクタは、以下のどのモードによっても、キュー・マネージャーと通信することができます。

- **バインディング・モード:** 統合ブローカーおよびコネクタは、TCP/IP 接続を使用せずに、直接キュー・マネージャーとの通信を行います。キュー・マネージャーおよびコネクタは、同一のマシン上に存在し、同一のキュー・マネージャーを共有する必要があります。デフォルトでは、このバインディング・モードになります。
- **リモート・キュー定義付きバインディング・モード:** 統合ブローカーとコネクタが異なるマシンにインストールされ、各マシンで専用のキュー・マネージャーが稼働している場合でも、コネクタおよび統合ブローカーはバインディング・モードでそれぞれのキュー・マネージャーと通信できますが、それにはリモート・キュー定義が必要です。
- **クライアント・モード:** TCP/IP を基本トランスポートとして使用するクライアント接続を介して通信が行われます。キュー・マネージャーとコネクタが異なるマシン上にある場合、コネクタが使用できるのはクライアント・モードのみです。

詳細

WebSphere MQ メッセージの詳細については、「*WebSphere MQ: Java の使用*」を参照してください。WebSphere MQ キューの詳細については、「*WebSphere MQ: 相互通信*」および「*WebSphere MQ: MQSC コマンド・リファレンス*」を参照してください。

第 2 部 展開と管理

第 5 章 インプリメンテーションの計画

本章では、WebSphere Business Integration システムをインプリメントする際に必要な計画の概要について説明します。本章は、次の各セクションから構成されています。

- 『ビジネス・プロセス・インターフェースの開発』
- 52 ページの『インプリメンテーションの段階』
- 56 ページの『開発ツール』

ビジネス・プロセス・インターフェースの開発

WebSphere Business Integration Adapters では、エンタープライズ・アプリケーション間の接続を可能にする、構成可能なモジュラー・エレメントが提供されています。これらのモジュラー・エレメント (コネクターやビジネス・オブジェクトを含む) は、1 つのアプリケーションから別のアプリケーションへのデータ送信に用いられるビジネス・プロセス・インターフェースを共同で形成します。

ビジネス・インテグレーション・システムのインプリメンテーションにあたっては、必要となるビジネス・プロセス・インターフェースを確認および開発する作業が非常に重要です。標準的なインプリメンテーションでは、複数のビジネス・プロセス・インターフェースが使用されます。

各インターフェースは、統合する必要のある特定のビジネス・タスクに対処します。簡単な例として、ある企業でレコード・システムとしてアプリケーション A を使用し、請求書発行にアプリケーション B を使用しているとします。両アプリケーション間で使用されているいくつかのタイプのビジネス情報に関するデータ交換を統合することがネックとなります。ビジネス情報のうちで極めて重要なものが顧客データです。顧客の氏名、住所、およびその他の詳細データは、片方でデータの内容が変更された場合にもう一方にもそれが反映されるように、双方のアプリケーション間で同期化をはかる必要があります。

そのためには、次に紹介する 2 つのインターフェースが必要です。

- アプリケーション B に対するアプリケーション A の customersync
- アプリケーション A に対するアプリケーション B の customersync

企業では、顧客に対する請求の対象品目についてフォローし、記録するということが必要です。このためには、さらに次の各インターフェースも用意しておく必要があります。

- アプリケーション B に対するアプリケーション A の itemsync
- アプリケーション A に対するアプリケーション B の itemsync

各インターフェースは、送信元アプリケーションと、交換されるビジネス・データのタイプとの組み合わせによって、他のインターフェースとは区別されます。また

各インターフェースには、独自のイベント通知メカニズム、トリガー、およびビジネス・オブジェクトが備わっており、それらによって送信元アプリケーションからのフローが開始されます。

インプリメンテーションの段階

ビジネス・インテグレーション・システムのインプリメンテーションは、いくつかの段階に分かれて実行されます。細かい部分や、各段階で生じる成果物の性質およびタイミングについては、インプリメンテーションを実行している組織によって異なります。しかし上位レベルから見れば、大まかに分けられたいくつかの段階は、WebSphere Business Integration システムのどのようなインプリメンテーションにも共通するものです。それは、以下にあげる諸段階です。

- 要件のディスカバリーおよび評価
- 使用可能なコンポーネントの評価
- 新規 (ユーザー設定) コンポーネントまたは拡張コンポーネントの設計
- 開発および構成
- 検証
- 展開

ビジネス・ゴールの発見および評価

この段階では、プロジェクトのビジネス・ゴール、システム要件、および開発作業の全体像を確認し、インプリメンテーション・プロセスに着手します。

上位レベルからディスカバリーを開始し、下位レベルの詳細へと移っていきます。まず、次のような上位レベルでの懸案事項から入ります。

- 企業にとって解決すべきビジネス上の問題とは、具体的には何を指すのか。
- ビジネス上の問題を解決するために統合すべき企業レベルのビジネス・プロセスはどれか。

下記の問題点、および企業に適用される他の問題点を提起します。

- インターフェースで使われるアプリケーションおよびそのバージョンについて把握できているか。
- 送信元アプリケーションはどれか。
- 宛先アプリケーションはどれか。
- データを記録するシステムのアプリケーションはどれか。
- アプリケーションはすでに実稼働しているか、それともまだ開発中か。
- アプリケーションは自社製の製品か、それとも購入した製品か。
- ビジネス・プロセスの統合を必要とするテクノロジー環境はどうなっているか (アプリケーション、データベース、および API を含む)。

テクノロジー環境の特性を定義する。次の各項目について調査を行う。

- データベース
- プラットフォームおよびオペレーティング・システム
- アプリケーション向けの API

- すべてのアプリケーション・クライアントおよびサーバーのプラットフォームのロケーション
- ネットワーク環境
- アプリケーションのバージョン
- 予測されるトランザクション・ボリューム

インプリメンテーションと使用コンポーネントに必要なインターフェースを特定するためには、下位レベルの詳細情報を調査します。その際には、インプリメントする具体的なビジネス・プロセス、必要となるビジネス・ロジックおよびデータ形式変更、さらに相互作用するアプリケーションおよびデータベースの詳細を特定し、記述します。この調査には、下記の情報収集作業も含まれます。

- ビジネス上の問題を解決するために必要なビジネス・プロセスを特定し、記述する。以下の点について問題提起を行う。
 - ビジネス・プロセスにおける標準のプロセス・フローはどれか。
 - トランザクションまたはデータ・フローを開始するイベントはどれか。
 - ビジネス・プロセスで使用するアプリケーションはどれか。
 - ビジネス・プロセス、アプリケーション、データを所有しているのはどの組織か。
 - 入力および出力はどうなっているのか。
 - データに前提条件または依存関係があるか。
 - フィルター操作の要件があるか。
 - 複数の宛先ポイントがある場合、データの送信先はどのように決定されるか。
 - インターフェースは双方向か。
 - トランザクション処理の頻度はどのくらいか。
 - その枠内でトランザクション処理が完了しなければならないという時間フレームが設定されているか。また、他のプロセスがそれに依存しているか。
 - データのボリュームはどのくらいか。
 - インターフェースはリアルタイムか、またはバッチか。
 - インターフェースは同期（応答が必要）か、または非同期（発信後は関知せず）か。
 - エラー処理手続きはどのようなものか。
- ビジネス・プロセス内の各アプリケーション間で交換されるデータ・エンティティの構造について記述する。
- 送信元および宛先アプリケーション間で必要となるデータ形式変更を特定する。
- ビジネス・プロセスで発生する、または発生すべきプログラマチック・イベント（ルーティング・ロジックおよびフィルター・ロジックのように、プロセスで発生するもの）を特定する。
- ビジネス・プロセス・フローをフロー・チャートで図示する。このフロー・チャートで、実行されるべき機能について分析することができます。さらに、フロー・チャートを利用して、アプリケーション間で交換されるデータのルーティングおよび変換のためのメッセージ・フローを作成することができます。

既存のコンポーネントの評価と新規コンポーネントの設計

評価および設計の内容は、ディスカバリー作業で収集した詳細情報によって決まります。

インターフェース、およびインターフェースに含まれる統合コンポーネントの詳細要件が決定したら、今度は既存のコンポーネントについて、ニーズに合うかどうかを判断するための評価を行います。要件によっては、必要なコンポーネントがすでに存在してそのまま使用できる場合もありますし、既存のコンポーネントに拡張(ニーズに合わせて修正)が必要な場合もあります。また、新規(ユーザー設定)統合コンポーネントの作成が必要な場合もあります。

各コンポーネントを個別に評価するのみならず、インターフェース全体で他のコンポーネントとどのように関係しているかという観点からも評価します。コンポーネントを設計する際には、インターフェース内でそのコンポーネントと対話する他のコンポーネントの設計作業も開始しておく必要があります。

コンポーネント設計の詳細については、下記の資料を参照してください。

- [ビジネス・オブジェクト開発ガイド](#)
- [コネクタ開発ガイド \(Java 用\)](#)
- [コネクタ開発ガイド \(C++ 用\)](#)

ビジネス・インテグレーション・システムの開発および構成

この段階では、必要な統合コンポーネントを新規に作成し、インプリメントの対象となる各ビジネス・プロセス・インターフェースごとにコンポーネントを構成します。

これは反復プロセスですが、コンポーネントの再構成または設計変更が必要になることもあります。インターフェースのコンポーネントを作成および構成しながら、単体テストを行って各コンポーネントが正しく動作することを確認します。インターフェース全体の構成が完了したら、ストリング・テストを行って、各インターフェースのコンポーネントすべてが同時に正しく動作することを確認します。

開発の全体的な流れ

上位レベルでは、次のようにオーソドックスな手順で、統合コンポーネントを開発することをお勧めします。

- コネクタ

コネクタおよびコネクタが使用するビジネス・オブジェクトは、アプリケーション本体と直接対話します。このような事情に加えて、新規コネクタの作成には他のコンポーネントを作成するよりも時間がかかるのが一般的なため、まずはコネクタについて確認(既存のものがある場合)しておくか作成しておく必要があります。ユーザー設定コネクタを作成する必要がなければ、アプリケーション固有のビジネス・オブジェクトから開発プロセスを開始します。開発または拡張すべきコネクタを決定する前に、IBM からすでに提供されているコネクタについて調査し、サイトにおける関連ライセンス規約を理解しておいてください。

- ビジネス・オブジェクト

他のコンポーネントはビジネス・オブジェクトに依存するため、まずビジネス・オブジェクトを開発します。

ユーザー設定によるアプリケーション固有のビジネス・オブジェクトは、コネクタでのテストを行い、機能を追加し、再テストするという、一連の作業の繰り返しによって作成します。コネクタは、最初の単体テストの際に構成します。そして開発と単体テストを繰り返すことで、ビジネス・オブジェクトおよび構成済みコネクタのインプリメンテーションが完成することになります。

作業の順序

開発および構成の作業は、通常各ステップを所定の順序で実行します。

1. ユーザー設定コネクタを開発する (必要な場合)。
2. アプリケーション固有のビジネス・オブジェクトを開発または機能拡張する。
3. コネクタと連動するように WebSphere MQ Integrator Broker を構成する。
4. コネクタを構成する。
5. メッセージ・フローを開発する。
6. アプリケーション固有のビジネス・オブジェクトに対する単体テストを実行する。
7. ビジネス・オブジェクトを利用してコネクタを構成する。
8. インターフェースのストリング・テストを行う。

ビジネス・インテグレーション・システムの検証

検証段階では、制御されたテスト環境でシステム・テストを実行し、システム設計がディスカバリー・フェーズで特定された要件をすべて満足していることを確認します。検証フェーズのシステム・テストでは、機能テスト、パフォーマンス・テスト、およびレグレッション・テスト (必要に応じて) を行います。

- 機能テスト - システムの機能要件すべてが満たされていることを確認します。要件マトリックスを使って、要件がテスト済みで許容範囲にあることを確認する必要があります。
- パフォーマンス・テスト - 設計または最適化によって、スループット、応答時間、待ち時間などの、タイミング要件が満たされたことを確認します。
- レグレッション・テスト - 検証フェーズでシステム設計パラメーターまたはシステム構成パラメーターに修正が加えられた場合、レグレッション・テストを実行して、他のシステム機能やスループット能力に悪影響が及んでいないことを確認する必要があります。

検証段階では、以下の作業を行います。

- 機能要件およびパフォーマンス要件をすべて含んだ要件マトリックスを作成する。
- システム・テスト環境の確定、指定、および構成を行う。
- システム・テストに使用するテスト・データ・セットを確定し、作成する。
- システム・インプリメンテーションを、開発環境からテスト環境へマイグレーションする。

- システム・テストの計画および手続きを作成し、クライアント側の同意を得るための調整を行い、同意を得る。各テスト・マトリックス項目に対して、具体的にテストを割り当てる。
- システムの機能テストおよびパフォーマンス・テストを実行する。
- システム・テストで検出された問題を特定、制御、解決する。
- レグレッション・テストを実行する (必要に応じて)。
- システム・テスト報告書を作成し、公表する。
- 開発したシステムに対する、クライアント側の受諾を得る。

ビジネス・インテグレーション・システムの展開

展開段階の目的は、開発およびテストが完了したビジネス・インテグレーション・システムをクライアントの実稼働環境へ確実にインプリメントすることです。必要に応じて最適化を行い、クライアントのサイトで直ちに実稼働できるようにします。

展開段階では、以下の作業を行います。

- 全体的なカットオーバーおよび展開計画を作成し、クライアント側の承認を取得して、既存のシステムやプロセスとの運用を中断せずに継続できることを確認する。
- 開発およびテストが完了したシステムを、テスト環境から実稼働環境へマイグレーションする。
- システム統合に関するクライアント固有のビジネス・プロセス要件およびデータ変換要件を収集し、それを文書化する。
- 固有の実稼働環境の要件をサポートするために必要な修正 (例えばアドレス、パスワードなど) を作成、インプリメント、およびテストする。
- クライアントの実動システム上に、ビジネス・インテグレーション・システム・ソフトウェアをインストールし、そのテストを実行する。
- ビジネス・インテグレーション・システムを実稼働環境用に最適化する。
- 展開したシステムに対するクライアント側の承認を取得する。

開発ツール

WebSphere Business Integration Adapters には、コネクタおよびビジネス・オブジェクトの作成と修正を支援するツールがいくつか組み込まれています。これらのツールの一覧を表 6 に示し、内容を説明します。

表 6. WebSphere Business Adapters 開発ツール

ツール	説明	詳細
Adapter Development Kit	コネクタ開発に使用可能なクラス・ライブラリーのセットおよび各ユーティリティーを提供するキット。単独で入手できます。	「コネクタ開発ガイド (Java 用)」および「コネクタ開発ガイド (C++ 用)」を参照してください。

表 6. WebSphere Business Adapters 開発ツール (続き)

ツール	説明	詳細
Connector Configurator	コネクタの構成に使用するグラフィカル・ユーザー・インターフェース (GUI)。	「コネクタ開発ガイド (Java 用)」および「コネクタ開発ガイド (C++ 用)」、または開発するコネクタ用のアダプター・ユーザーズ・ガイドを参照してください。
Business Object Designer	ビジネス・オブジェクト定義の作成に使用するグラフィカル・ユーザー・インターフェース (GUI)。手動で作成する場合にも、Object Discovery Agent (ODA) から作成する場合にも用いられます。	「ビジネス・オブジェクト開発ガイド」を参照してください。
Object Discovery Agent Development Kit (ODK)	Object Discovery Agent (ODA) を作成するための API のセット。ODA では、あるデータ・ソースに固有なビジネス・オブジェクト要件を特定し、それらの要件から定義を生成します。	「ビジネス・オブジェクト開発ガイド」を参照してください。
Visual Test Connector	ビジネス・インテグレーション・インターフェースを正常に展開したかどうかをテストできるように、コネクタのアクティビティをシミュレートするツール。	183 ページの『付録 G. Visual Test Connector の使用』を参照してください。

注: デザイナー・ツールのどれかを起動しようとしたときに「クラスが見つかりません」という内容のエラーが発生した場合は、System Manager を起動してから、デザイナー・ツールの起動を再試行する必要があります。ただし、いったんツールが起動してしまえば、System Manager を稼働状態のままにしておく必要はありません。

第 6 章 WebSphere Business Integration Adapters のインストール

本章では、WebSphere Business Integration Adapters およびサポート・ソフトウェアを WebSphere Message Broker にインストールする方法について説明します。

以前のバージョンから WebSphere Business Integration Adapters をアップグレードする場合の手順については、199 ページの『付録 H. WebSphere Business Integration Adapters のアップグレード』を参照してください。

本章では、Windows オペレーティング・システムおよび UNIX オペレーティング・システムでのインストール方法を説明します。以下のうち、ご使用のオペレーティング・システムに該当するセクションを参照してください。

- 『Windows システムへのインストール』
- 61 ページの『UNIX システムへのインストール』

本章には、サポート・ソフトウェア (JDK および WebSphere MQ) の WebSphere Message Broker へのインストールに関する特定の情報が含まれています。別の資料「*WebSphere Business Integration Adapters インストール・ガイド*」には、WebSphere Business Integration Adapters のインストールに必要なすべての詳細情報が記載されています。本章の、Windows システムまたは UNIX システムへのインストールについて説明しているセクションでは、「*WebSphere Business Integration Adapters インストール・ガイド*」を参照するよう、インストール・プロセスの中で随時指示します。

Windows システムへのインストール

このセクションのトピックは以下のとおりです。

- 『ソフトウェア要件』
- 60 ページの『JDK のインストール』
- 60 ページの『WebSphere MQ のインストール』
- 61 ページの『WebSphere Business Integration Adapters のインストール』

ソフトウェア要件

WebSphere Business Integration システムは、WebSphere Business Integration Adapters およびその他のコンポーネントから構成されます。WebSphere Business Integration Adapters および関連コンポーネントは、CD または ESD (IBM Electronic Software Delivery) で配布されます。WebSphere Business Integration システムで使用するその他のコンポーネントは、個別に入手し、インストールする必要があります。

60 ページの表 7 は、WebSphere Business Integration システムのソフトウェア要件を一覧表示したものです。

表 7. *WebSphere Business Integration* システム・ソフトウェア要件 (*Windows* システムの場合)

ソフトウェア	製品に同梱
Service Pack 4 を適用済みの <i>Windows 2000</i> (Professional、Server、または Advanced Server)。 注: <i>WebSphere Business Integration Adapter Framework 2.4.0</i> のアダプターは <i>Windows NT</i> ではサポートされません。	なし
以下のいずれかのメッセージ・ブローカー:	
<p><i>IBM WebSphere MQ Integrator Broker v 2.1.0</i> (以下が適用されているもの):</p> <ul style="list-style-type: none"> • CSD05 Service Pack。 http://www.ibm.com/software/integration/mqfamily/support/summary/mqsib.html からダウンロードしてください。 <p><i>IBM WebSphere MQ Integrator v. 2.1.0</i>. サービス・パックが必要かどうかを判断するには、お客様の <i>IBM</i> 担当員にご確認ください。</p> <p><i>IBM WebSphere Business Integration Message Broker v. 5.0</i> (以下が適用されているもの):</p> <ul style="list-style-type: none"> • CSD02 Service Pack。 http://www.ibm.com/software/integration/mqfamily/support/summary/wbib.html からダウンロードしてください。 	なし
<i>Java Runtime Environment (JRE): IBM</i> バージョン 1.3.1 SR5。	あり
<i>Java Development Kit - カスタム Java コネクターを開発する場合にのみ必要。IBM JDK</i> バージョン 1.3.1 SR5。	あり
CSD05 を適用済みの <i>IBM WebSphere MQ</i> バージョン 5.3.0.2。	なし
ブラウザ: <i>HTML</i> 文書を参照するために、 <i>Microsoft Internet Explorer</i> や <i>Netscape Navigator</i> などの <i>HTML</i> ブラウザーが必要です。サポートされている正確なバージョンについては、 http://www.ibm.com/integration/wbiadapters/library/infocenter からダウンロード可能な説明を参照してください。	なし

JDK のインストール

JDK (*Java Development Kit*) は、カスタム *Java* コネクターを開発する予定がある場合にのみ必要です (*Java* コンパイラーが必要となるためです)。 *Windows* システムで開発を行う場合、*JDK* は *WebSphere Business Integration Adapter FrameWork* に組み込まれています。 *Adapter Framework* のインストールの詳細については、「*WebSphere Business Integration Adapters* インストール・ガイド」を参照してください。

WebSphere MQ のインストール

IBM WebSphere MQ は、*WebSphere Message Broker* とアダプター間の通信を可能にするメッセージング・ソフトウェアです。

インストールおよび構成情報については、次の *WebSphere MQ* 各資料を参照してください。

- *WebSphere MQ: スタートアップ・ガイド*
- *WebSphere MQ: システム管理*
- *WebSphere MQ: 相互通信*

注: これらの資料は次の *IBM Web* サイトよりブラウズまたはダウンロードできます。www.ibm.com/software/mqseries

関連する WebSphere MQ の資料については、viii ページの『WebSphere Message Broker の資料』を参照してください。

WebSphere MQ を、WebSphere Message Broker と連動するように構成するには、89 ページの『コネクタと連動するようにメッセージ・ブローカーを構成する方法』を参照してください。

WebSphere Business Integration Adapters のインストール

WebSphere Business Integration Adapter 製品のインストールについては、「*WebSphere Business Integration Adapters* インストール・ガイド」を参照してください。この資料は、次の Web サイトの WebSphere Business Integration Adapters Infocenter にあります。

<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

UNIX システムへのインストール

このセクションのトピックは以下のとおりです。

- 『ソフトウェア要件』
- 62 ページの『JDK のインストール』
- 64 ページの『WebSphere MQ のインストール』
- 64 ページの『WebSphere Business Integration Adapters のインストール』

ソフトウェア要件

WebSphere Business Integration システムは、WebSphere Business Integration Adapters およびその他のコンポーネントから構成されます。WebSphere Business Integration Adapters および関連コンポーネントは、CD または ESD (IBM Electronic Software Delivery) で配布されます。WebSphere Business Integration システムで使用されるその他のコンポーネントは、個別に入手し、インストールする必要があります。

62 ページの表 8 は、WebSphere Business Integration システムのソフトウェア要件を一覧表示したものです。

表 8. WebSphere Business Integration システム・ソフトウェア要件 (UNIX システムの場合)

ソフトウェア	製品に同梱
<p>オペレーティング・システム:</p> <ul style="list-style-type: none"> • AIX - AIX 5L バージョン 5.1 (保守レベル 1) または 5.2 (保守レベル 4)。 • Solaris - Solaris 7.0 または 8.0。2003 年 7 月 23 日リリースの Patch Cluster が適用されているもの。 • HP-UX - HP-UX 11.11 (11i, r=B.11.110306.4)。2003 年 6 月リリースの GOLDBASE 11i, r=B.11.110306.4 バンドルおよび 2003 年 6 月リリースの GOLDAPPS11i 11i, r=B.11.110306.4 バンドルを適用済みのもの。 	いいえ
<p>以下のいずれかのメッセージ・ブローカー:</p> <p>IBM WebSphere MQ Integrator Broker v 2.1.0 (以下が適用されているもの):</p> <ul style="list-style-type: none"> • CSD05 Service Pack。 http://www.ibm.com/software/integration/mqfamily/support/summary/mqsib.html からダウンロードしてください。 <p>IBM WebSphere MQ Integrator v. 2.1.0.サービス・パックが必要かどうかを判断するには、お客様の IBM 担当員にご確認ください。</p> <p>IBM WebSphere Business Integration Message Broker v. 5.0 (以下が適用されているもの):</p> <ul style="list-style-type: none"> • CSD02 Service Pack。 http://www.ibm.com/software/integration/mqfamily/support/summary/wbib.html からダウンロードしてください。 	いいえ
<p>Java Runtime Environment:</p> <ul style="list-style-type: none"> • AIX - IBM JRE バージョン 1.3.1 SR5。 • Solaris - Sun JRE バージョン 1.3.1.7。 • HP-UX - Sun JRE バージョン 1.3.1.6。 	はい
<p>Java Development Kit - カスタム Java コネクタを開発する場合にのみ必要。</p> <ul style="list-style-type: none"> • AIX: IBM JDK バージョン 1.3.1 SR5。 • Solaris: Sun JDK バージョン 1.3.1.7。 • HP-UX: Sun JDK バージョン 1.3.1.6。 	いいえ
<p>CSD05 を適用済みの WebSphere MQ バージョン 5.3.0.2。</p>	いいえ
<p>ブラウザ: HTML 文書を参照するために、Microsoft Internet Explorer や Netscape Navigator などの HTML ブラウザーが必要です。サポートされている正確なバージョンについては、http://www.ibm.com/integration/wbiadapters/library/infocenter からダウンロード可能な説明を参照してください。</p>	いいえ

JDK のインストール

JRE には、WebSphere Business Integration Adapters を実行するために必要な Java 仮想マシン (JVM) が組み込まれています。ただし、JavaC (Java コンパイラ) などの開発ツールは組み込まれていません。カスタム・コネクタを作成しない場合は、JRE を使用する。

JRE には、WebSphere Business Integration システムの実行に必要な Java ソフトウェアのランタイム・コンポーネントが含まれています。JRE は、WebSphere Business Integration Adapter Framework に組み込まれています。したがって、JRE を個別にインストールする必要はありません。

カスタム・コネクターを作成する場合は、完全な JDK をインストールしてください。これは、Sun または IBM の Web サイトからダウンロードできます。Solaris および HP-UX の場合、JDK は <http://java.sun.com/products/jdk/1.3> からダウンロードできます。AIX の場合、JDK は <http://www.ibm.com/developerworks/java/jdk> からダウンロードできます。

JDK を、現在のシステムにインストールするには、次の手順に従います。

1. JDK のインストール先となるディレクトリーへ移動する。

```
cd /install_dir
```

ここで、*install_dir* は JDK ソフトウェアがインストールされるディレクトリーのパスです。このソフトウェアは任意の場所にインストールできます。

- JDK は、通常 /usr ディレクトリーのサブディレクトリーにインストールされます。/usr ファイル・システムに十分なスペースがある場合は、ここに JDK をインストールします (例えば、/usr/jdk1.3)。
 - JDK の抽出ディレクトリー (例えば、/home_dir/jdk1.3) を作成して、このディレクトリーに移動することもできます。/usr/jdk1.3 ディレクトリーからの抽出ディレクトリーへのシンボリック・リンクを作成する必要があります。
2. Sun または IBM の Web サイトで、JDK をダウンロードする Web ページを検索する。

JDK は通常、Java 2 SDK Standard Edition for UNIX の下にあります。

3. 付属の README ファイルをダウンロードして、JDK のダウンロード方法を確認する。
4. JDK を、インストール用ディレクトリーにダウンロードする。

IBM および Sun では、自己解凍機能のある tar ファイルを用意しており、その内容を任意のディレクトリーへ格納することができます。このファイルは、通常は *JDK_version.bin* のフォーマットになっています (ここで、*JDK_version* はダウンロードする JDK のバージョンです)。このファイルから JDK ソフトウェアを取り出す方法については、README ファイルを参照してください。

5. JDK についてリストされたオペレーティング・システム用パッチの中から、ご使用の UNIX オペレーティング・システムのバージョンに適したものをダウンロードする。

これらのパッチをインストールする前に、まだパッチがインストールされていないことを確認してください。

6. ダウンロード・ディレクトリー (例えば /usr/jdk1.3 ディレクトリー) から、Java ディレクトリーへのリンクを作成する。

```
a. rm /usr/java
```

```
b. ln -s /usr/jdk1.3 /usr/java
```

7. インストール後、JDK の bin ディレクトリーが PATH 環境変数に組み込まれていることを確認する。

これは、次のいずれかの方法で行います。

- ユーザーの始動時に読み込まれる、`/etc/profile` を編集する。 `ksh`、`bash`、および `sh` などのシェルは、`/etc/profile` ファイル内の設定値を使用します。

例えば、JDK を `/usr/jdk1.3` ディレクトリーにインストールする場合は、`/etc/profile` の `PATH` エントリーを次のように編集します。

```
PATH=/usr/jdk1.3/bin:$PATH
export PATH
```

- WebSphere Business Integration 管理者のホーム・ディレクトリーにある、個人用プロファイル・ファイルを編集する。

この個人用プロファイル・ファイルの名前は、WebSphere Business Integration 管理者のアカウントで使用されている特定のシェルによって異なります。この個人用プロファイルに対する変更は、WebSphere Business Integration 管理者としてログインしたユーザーのみ に影響します。

特定のシェルに最適な構文を使用して、`PATH` 環境変数に `JDK bin` ディレクトリーを組み込みます。例えば、WebSphere Business Integration 管理者のアカウントが `sh` シェルを使用している場合、`JRE` パス (`/usr/java/bin`) を追加する行を次のように編集して、`JDK` パスを組み込みます。

```
PATH=/usr/jdk1.3/bin:/usr/java/bin:$PATH
export PATH
```

これらの行を、WebSphere Business Integration 管理者のアカウントの個人用プロファイル・ファイルで `CWSharedEnv.sh` ファイルを指定している行の後に追加します。

WebSphere MQ のインストール

IBM WebSphere MQ は、WebSphere Message Broker とアダプター間の通信を可能にするメッセージング・ソフトウェアです。

インストールおよび構成情報については、次の WebSphere MQ 各資料を参照してください。

- *WebSphere MQ: スタートアップ・ガイド*
- *WebSphere MQ: システム管理*
- *WebSphere MQ: 相互通信*

注: これらの資料は次の IBM Web サイトよりブラウズまたはダウンロードできません。 www.ibm.com/software/mqseries

WebSphere MQ を、WebSphere Message Broker と連動するように構成するには、89 ページの『コネクタと連動するようにメッセージ・ブローカーを構成する方法』を参照してください。

WebSphere Business Integration Adapters のインストール

WebSphere Business Integration Adapters をインストールするには、まずサポートされているバージョンの製品を入手します。次に、Web サイト <http://www.ibm.com/software/websphere/wbiadapters/infocenter> にある WebSphere Business Integration Adapters InfoCenter から、そのリリースの資料をダウンロード

します。「*WebSphere Business Integration Adapters* インストール・ガイド」の手順に従って、ご使用のビジネス・インテグレーション・システムにインストールしてください。

第 7 章 ビジネス・インテグレーション・システムの管理

本章では、WebSphere Business Integration Adapters のために実行する必要がある、管理用タスクについて説明します。使用している WebSphere Message Broker のみに関連する管理用タスクは、各メッセージ・ブローカーの管理資料で説明されています (詳細については、vii ページの『関連資料』を参照してください)。

本章は、次の各セクションから構成されます。

- 『コネクターの始動』
- 69 ページの『コネクターの停止』
- 70 ページの『複数のコネクター・インスタンスの作成』
- 72 ページの『Adapter Monitor および障害キュー・マネージャーの使用』
- 78 ページの『ログ・ファイルおよびトレース・ファイルの管理』
- 81 ページの『Log Viewer を使用してコネクター・メッセージを表示する方法』

コネクターの始動

コネクターの始動方法は、動作環境が UNIX と Windows のいずれのシステムであるかによって異なります。デフォルトでは、コネクター構成ファイル内の設定値がコネクター・プロパティの値となります。ただし、コネクターの構成ファイルで指定された一部のコネクター・プロパティは、実行時にオーバーライドすることができます。143 ページの『付録 D. コネクター始動オプション』では、コネクターの構成ファイル内の設定済みプロパティをオーバーライドするために、使用するオプションについて紹介しています。

注: 必ず、111 ページの『コネクターの始動ファイル、ショートカット、および環境変数の構成』での指示に従ってから、コネクターを始動してください。

Windows の場合

以下の方法により、コネクターを始動します。

- ショートカットを作成済みの場合は、デスクトップ上でそのショートカットをダブルクリックする。
- 「プログラム」>「IBM WebSphere Business Integration Adapters」>「アダプター」>「コネクター」の順に選択し、そこで対象のコネクターを選択する。プログラム名は、デフォルトでは「IBM WebSphere Business Integration Adapters」となっていますが、それをカスタマイズすることもできます。プログラム名は、デフォルトでは「IBM WebSphere Business Integration Adapters」となっていますが、それをカスタマイズすることもできます。
- System Manager の Adapter Monitor パースペクティブからコネクターを始動できます。詳しくは、75 ページの『アダプターの状態の変更』を参照してください。
- DOS ウィンドウで次の処理を実行します。
 - 大半のコネクターでは、以下のコマンドを入力します。

```
start_connName connectorName brokerName -cConfigFile
```

ここで、

connName および *connectorName* はコネクター名、*brokerName* は WebSphere Message Broker の名前、*configFile* はコネクターの構成ファイルの絶対パス名を指定しています。

コネクターの始動オプションは、コネクターのショートカット・プロパティを編集するか、またはコネクターの *start_connName.bat* ファイルを直接編集すれば、その内容を変更することができます。

注: 大半のコネクター、例えば Java ベースのすべてのコネクターや IBM が提供するほぼすべてのコネクターでは、上記の構文を使用します。ご使用のコネクターがこの構文を使用するかどうかは、ディレクトリー *ProductDir/connectors* 内を調べると確認できます。*start_connName* という名前のスクリプトが存在する場合は、上記の構文を使用します。C++ 言語で記述された一部のコネクターでは、代わりに後のセクションで説明する構文を使用する必要があります。このディレクトリー内にコネクターに関連付けられた *start_connName* スクリプトが存在しない場合は、後述する構文を使用します。

- 上記の *start_connName* 構文を使用できない C++ コネクター (一部の IBM コネクターおよびカスタム・コネクター) の場合、コネクターがインストールされているディレクトリーにナビゲートします。デフォルトでは、このディレクトリーは *ProductDir* です。下記のコマンドを入力します。

```
start_connector connectorName brokerName -cConfigFile
```

ここで、

connectorName はコネクター名、*brokerName* は WebSphere Message Broker の名前、*configFile* はコネクターの構成ファイルの絶対パス名を指定しています。

Java コネクターの始動オプションは、コネクターのショートカット・プロパティを編集するか、またはコネクターの *start_connector.bat* ファイルを直接編集すれば、その内容を変更することができます。

UNIX の場合

以下の方法により、コネクターを始動します。

1. *ProductDir/bin* ディレクトリーにナビゲートする。
2. 下記のコマンドを発行してコネクター・マネージャー・スクリプトを実行する。

```
connector_manager_connName -start
```

ここで、

connName はコネクター名です。

このコネクター名は、大文字や小文字およびスペルでも、*ProductDir/bin/connectors* の配下にあるコネクターのサブディレクトリー名と一致しなければなりません。

コネクターの停止

コネクターは、そのコネクターがインストールされているシステム、または統合ブローカーがインストールされているシステムのいずれかから停止することができます。コネクターを停止するには、以下のようないくつかのメソッドを使用できます。

コネクター・システムから

Windows から

- Windows では、System Manager の Adapter Monitor パースペクティブからコネクターを停止できます。詳しくは、75 ページの『アダプターの状態の変更』を参照してください。
- Adapter Monitor からコネクターを停止できない場合、コネクター・タイプに応じたコンソール・ウィンドウで「Q」と入力し、Enter キーを押します。これによって、コネクターのプロセスが強制終了されます。

UNIX の場合

UNIX システムの場合は、以下のコマンドを使用します。

1. `ProductDir/bin` ディレクトリーにナビゲートします。
2. 以下のように入力します。 `connector_manager_connName - option`

ここで、`option` は以下のオプションのいずれかです。

表9. `connector_manager` コマンド停止オプション

オプション	説明
- stopgraceful	コネクター・エージェント側から Java ユーティリティを呼び出して、コネクター・エージェントを停止します。ただし、自動再始動機能が有効になっている場合、コネクター・エージェントは自動的に再始動します。
-stop	コネクター・エージェントを停止する Java ユーティリティを呼び出します。自動再始動機能が有効な場合でも、コネクター・エージェントは強制的にシャットダウンされます。
-kill	コネクター・エージェント用のオペレーティング・システム・プロセスを強制終了します。他の方法でコネクターを停止できなかった場合に使用します。

ブローカー・システムから

Windows または UNIX のいずれのシステムでも、WebSphere Business Integration Adapters Utility がインストールされていれば、統合ブローカーがインストールされているシステムから、統合ブローカーのコマンド行で、`mqsi remotestoptadapter` コマンドを発行できます。

注: このコマンドを使用するには、WebSphere MQ Integrator Broker SupportPac IO02: WebSphere Business Integration Adapters Utility がインストールされている必要があります。

このコマンドの構文は、下記のとおりです。

```
mq siremotestopadapter brokerName -c adapterName -n adapterQueue  
[-h hostName] [-p port] [-q adapterQueueManager]
```

詳細については、以下のセクション『mq siremotestopadapter コマンド行パラメーター』を参照してください。

mq siremotestopadapter コマンド行パラメーター

必須パラメーター

brokerName

アダプターが通信する WebSphere Message Broker の名前。

-c *adapterName*

シャットダウンするアダプターの名前を識別します。

-n *adapterQueue*

アダプターがシャットダウン・メッセージをモニターする WebSphere MQ キューの名前。

オプション・パラメーター

-h *hostName*

-q パラメーターで指定された WebSphere MQ キュー・マネージャーが動作しているマシンの名前または IP アドレス (デフォルトではローカル・マシンに設定されています)。

-p *port*

-q パラメーターで指定された WebSphere MQ キュー・マネージャーが受信する要求を待機するポート。デフォルト値は 1414 です。

-q *adapterQueueManager*

アダプターが使用する WebSphere MQ キュー・マネージャーを指定します。**-n** パラメーターで指定されたキューは、このキュー・マネージャーで管理する必要があります。このキュー・マネージャーは、アダプターと同じマシンに存在します。

複数のコネクター・インスタンスの作成

コネクターの複数のインスタンスを作成する作業は、いろいろな意味で、カスタム・コネクターの作成と同じです。以下に示すステップを実行することによって、コネクターの複数のインスタンスを作成して実行するように、ご使用のシステムを設定することができます。次のようにする必要があります。

- コネクター・インスタンス用に新規ディレクトリーを作成します。
- 必要なビジネス・オブジェクト定義が設定されていることを確認します。
- 新規コネクター定義ファイルを作成します。
- 新規始動スクリプトを作成します。

新規ディレクトリーの作成

それぞれのコネクタ・インスタンスごとにコネクタ・ディレクトリーを作成する必要があります。このコネクタ・ディレクトリーには、次の名前を付けなければなりません。

```
ProductDir¥connectors¥connectorInstance
```

ここで connectorInstance は、コネクタ・インスタンスを一意的に示します。

コネクタに、コネクタ固有のメタオブジェクトがある場合、コネクタ・インスタンス用のメタオブジェクトを作成する必要があります。メタオブジェクトをファイルとして保管する場合は、次のディレクトリーを作成して、ファイルをそこに格納します。

```
ProductDir¥repository¥connectorInstance
```

ビジネス・オブジェクト定義の作成

各コネクタ・インスタンスのビジネス・オブジェクト定義がプロジェクト内にまだ存在しない場合は、それらを作成する必要があります。

1. 初期コネクタに関連付けられているビジネス・オブジェクト定義を変更する必要がある場合は、適切なファイルをコピーし、Business Object Designer を使用してそれらのファイルをインポートします。初期コネクタの任意のファイルをコピーできます。変更を加えた場合は、名前を変更してください。
2. 初期コネクタのファイルは、次のディレクトリーに入っていないければなりません。

```
ProductDir¥repository¥initialConnectorInstance
```

作成した追加ファイルは、ProductDir¥repository の適切な connectorInstance サブディレクトリー内に存在している必要があります。

コネクタ定義の作成

Connector Configurator 内で、コネクタ・インスタンスの構成ファイル (コネクタ定義) を作成します。これを行うには、以下のステップを実行します。

1. 初期コネクタの構成ファイル (コネクタ定義) をコピーし、名前変更します。
2. 各コネクタ・インスタンスが、サポートされるビジネス・オブジェクト (および関連メタオブジェクト) を正しくリストしていることを確認します。
3. 必要に応じて、コネクタ・プロパティをカスタマイズします。

始動スクリプトの作成

始動スクリプトは以下のように作成します。

1. 初期コネクタの始動スクリプトをコピーし、コネクタ・ディレクトリーの名前を含む名前を付けます。

```
dirname
```

2. この始動スクリプトを、『新規ディレクトリーの作成』で作成したコネクタ・ディレクトリーに格納します。
3. 始動スクリプトのショートカットを作成します (Windows のみ)。

4. 初期コネクターのショートカット・テキストをコピーし、新規コネクター・インスタンスの名前に一致するように (コマンド行で) 初期コネクターの名前を変更します。

これで、ご使用の統合サーバー上でコネクターの両方のインスタンスを同時に実行することができます。

カスタム・コネクター作成の詳細については、「コネクター開発ガイド (C++ 用)」または「コネクター開発ガイド (Java 用)」を参照してください。

Adapter Monitor および障害キュー・マネージャーの使用

Adapter Monitor は、アダプターの状態の変更に使用できる System Manager のパースペクティブであり、障害キューで受信した失敗イベントを、障害キュー・マネージャーを通じて処理することを可能にします。System Manager の使用法の詳細については、147 ページの『付録 E. System Manager と Eclipse Workbench』を参照してください。

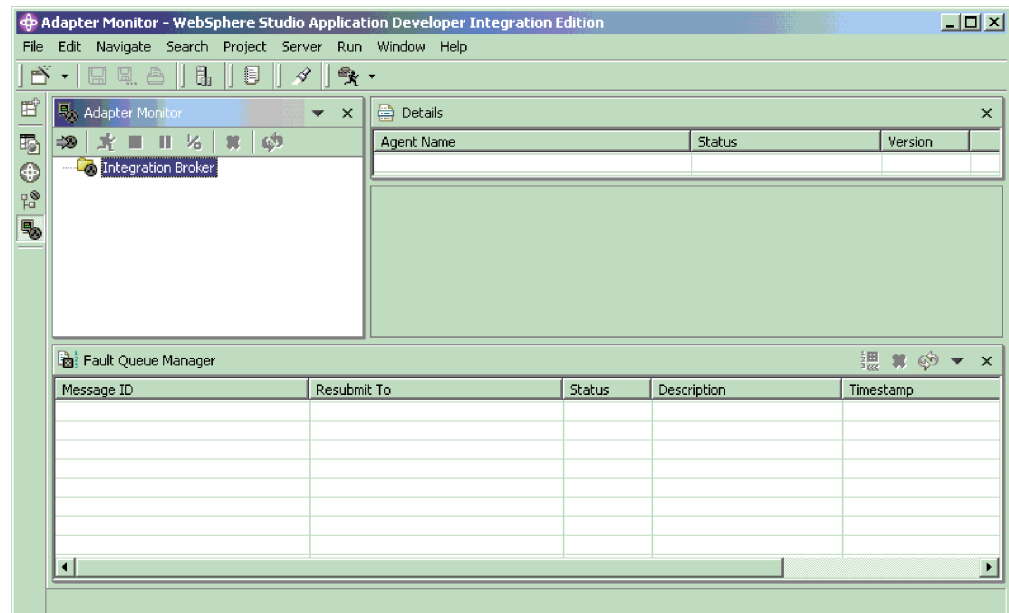
Adapter Monitor パースペクティブ

Adapter Monitor パースペクティブを使用すると、アダプターを管理できます。また、サブミットされたイベントの処理時にエラーが発生した場合には、障害キュー・マネージャー・パネルからメッセージを再サブミットすることもできます。

Adapter Monitor は、JMS と共に使用するよう構成されているアダプターに対してのみ使用できます。

Adapter Monitor の開始

Adapter Monitor の開始方法: System Manager パースペクティブでプロジェクト・フォルダーを展開し、コネクター定義アイコンを右マウス・ボタンでクリックし、ポップアップ・ダイアログから「Adapter Monitor」を選択します。「Adapter Monitor」ウィンドウが表示されます。



Adapter Monitor の設定の変更

Adapter Monitor がアダプターの状態をポーリングする間隔を決定する設定や、障害キュー・マネージャーに表示されるメッセージの数を決定する設定は、変更することができます。設定を変更するには、WSADIE メニュー・バーから「ウィンドウ」>「設定」>「Adapter Monitor の設定」を選択し、以下に対する値を設定します。

- Adapter Monitor ビュー

アダプター状況のポーリングが行われてからその次に行われるまでの間の経過秒数を表す数値を入力します。

- 障害キュー・マネージャー

障害キュー・マネージャーに表示するメッセージの数の最大値を表す数値を入力します。

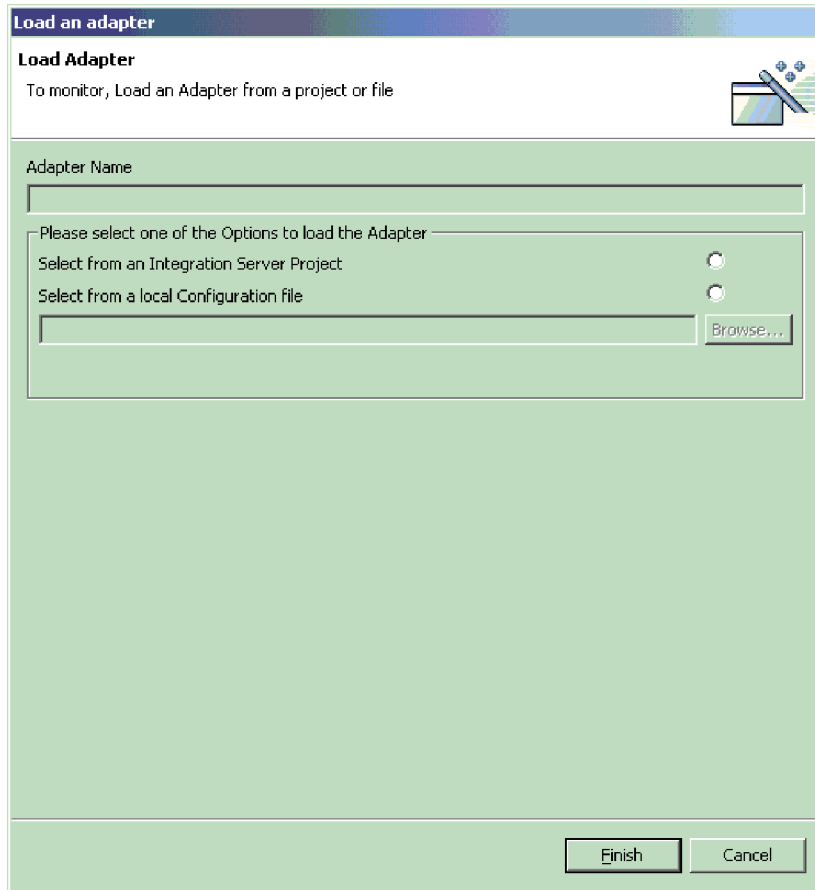
「適用」または「OK」を選択します。

アダプターのロード

Adapter Monitor を使用するには、まずキュー・マネージャーを始動します。次に、アダプターを始動し、Adapter Monitor ビューにそのアダプターをロードします (Adapter Monitor ビューにアダプターをロードしてから、アダプターを始動してもかまいません)。

アダプターを始動するには、コネクタ始動スクリプト、ショートカット、Visual Test Connector などの、そのアダプターを始動するためにセットアップした手段を使用します。

Adapter Monitor にアダプターをロードするには、「統合ブローカー」を右マウス・ボタンでクリックし、「アダプターのロード」を選択します。次の図に示すダイアログが表示されます。



アダプターの構成ファイルと統合プロジェクトのどちらからアダプターをロードするかを選択することができます。

- プロジェクトをロード元として選択した場合、ダイアログには、System Manager で現在有効なユーザー・プロジェクトが表示されます。プロジェクト内のコネクターを選択し、「完了」を選択します。(有効なユーザー・プロジェクトがない場合には、何も表示されません。この場合、新しいユーザー・プロジェクトを作成するか、構成ファイルをロード元にします。)
- ファイルをロード元として選択した場合、参照ボックスが開きます。アダプターの構成ファイル (通常、構成ファイルの拡張子は *.cfg ですが、別の拡張子になっている場合もあります) の格納先に移動してこのファイルを選択し、「保管」を選択します。選択した構成ファイルの名前が、ボックスに表示されます。「完了」を選択します。Adapter Monitor にアダプターがロードされ、そのアダプターの現在のデータが表示されます。

Adapter Monitor の表示

アダプターをロードすると、左上のパネルの「統合ブローカー」フォルダーの下に、そのアダプターを表すアイコンが表示されます。このアイコンは、アダプターの現在の状態を示しています。Adapter Monitor にアダプターをロードする操作そのものによって、アダプターの状態が変化することはありません。アダプターのロードが完了すると、アダプターの状態を変更するためのアクションを Adapter Monitor から実行することができます。

Adapter Monitor の表示は、Adapter Monitor ビューの設定で指定したポーリング時間間隔に従って、定期的に更新されます。ただし、ツールバーまたはメニュー・バーの「最新表示」ボタンを選択すれば、いつでもすぐに更新することができます。

Adapter Monitor の右上のパネルは、「詳細」パネルです。「詳細」パネルには、アダプターの名前、状況、およびバージョンが表示されます。

Adapter Monitor の下部パネルは、障害キュー・マネージャー・ビューになっています。このビューには、キュー・マネージャーの障害キューへ発信されたメッセージが表示されます。この障害キュー・マネージャー・ビューでは、イベント・フローの失敗の結果として障害キューに置かれたメッセージを、再サブミットするか、削除することができます。

アダプターの状態の変更

Adapter Monitor を使用すると、アダプターの状態をモニターして変更することができます。アダプターの状態とは、アダプターによって実行されている（または実行されていない）処理のことを指します。

注: アダプターの「状態」が存在すると、アダプターが始動している状態であることが推定できます。始動していないアダプターやシャットダウン後に再始動していないアダプターには状態が存在しないため、Adapter Monitor の動作による影響はありません。アダプターの開始方法については、67 ページの『コネクタの始動』を参照してください。

通常、アダプターは、次の 2 種類の処理を実行します。

- イベント通知のポーリング

アダプターは、そのアプリケーションのイベント・ストアを複数のイベントに対してポーリングし、これらのイベントをビジネス・オブジェクト・メッセージとして統合ブローカーに送信します。

- 要求処理

アダプターは、統合ブローカーからアプリケーションに送信された要求ビジネス・オブジェクトを受信します。

アダプターの状態	要求処理	ポーリング
アクティブ	あり	あり
一時停止	あり	なし
非アクティブ	なし	なし

アダプターの状態を変更するには、そのアダプターを表すアイコンを右マウス・ボタンでクリックし、次のいずれかを選択します。

- アクティブ化

アダプターを「一時停止」状態または「非アクティブ」状態から「アクティブ」状態へ変更します。

- 非アクティブ化

アダプターを「一時停止」状態または「アクティブ」状態から「非アクティブ」状態へ変更します。

- 一時停止

アダプターを「アクティブ」状態状態から「一時停止」状態へ変更します。

- シャットダウン

アダプターを停止します。この動作により、コネクタ始動スクリプトが終了し、アプリケーションとの接続が切断されて割り当てられているリソースがすべて解放されます。コネクタは再始動するまでシャットダウンしたままです。Adapter Monitor からコネクタを再始動できません。始動および再始動に関する説明については、『コネクタの始動』を参照してください。

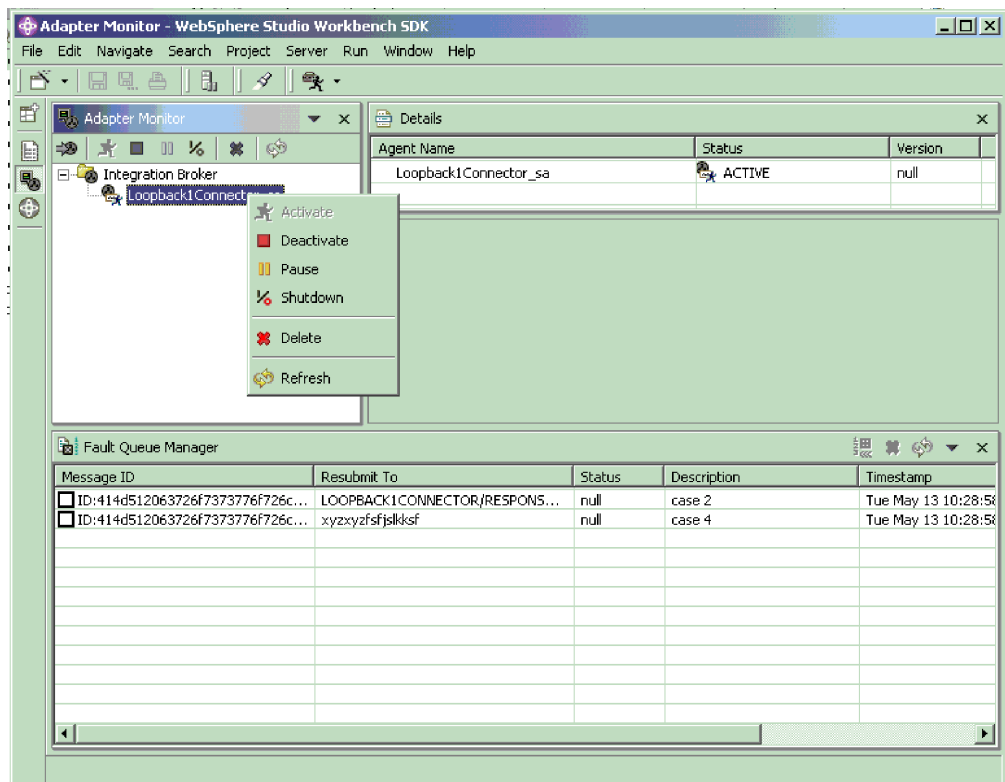
次の 2 つのコマンドは、コネクタの処理に直接の影響を与えることはありませんが、Adapter Monitor がコネクタについて表示する内容には影響があります。

- 削除

Adapter Monitor からこのアダプターの構成情報を削除します。これにより、アダプターが変更されたり、アダプターの状態が変化することはありません。単に、このアダプターを Adapter Monitor から選択できなくなるだけです。

- 最新表示

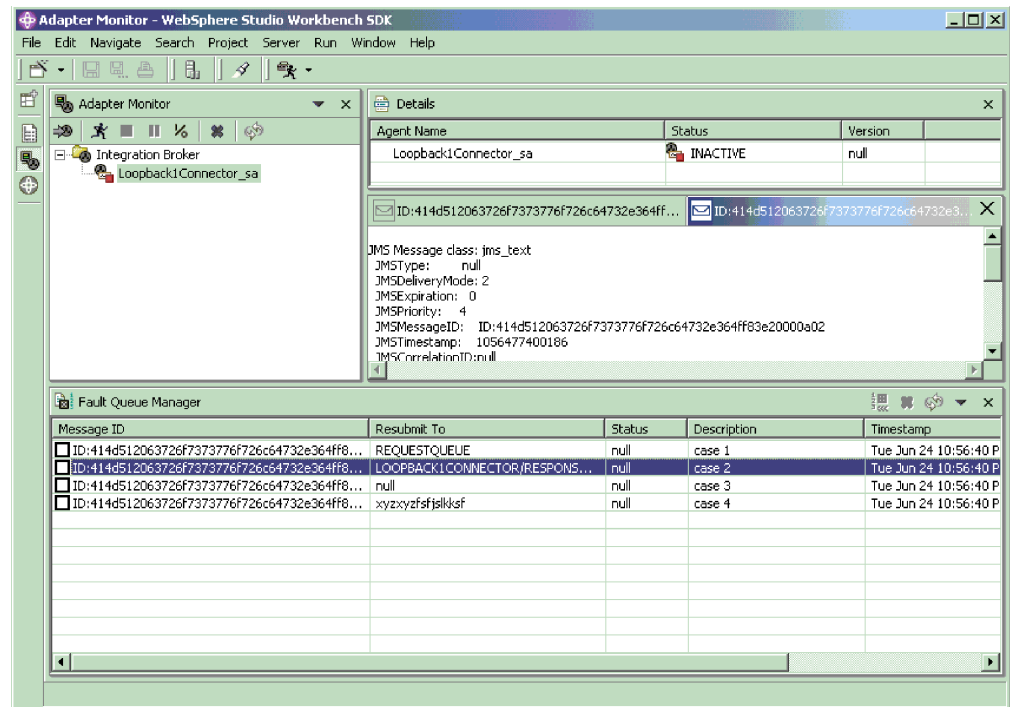
コネクタ・エージェントの現在の状態を読み取る GETSTATUS コマンドを起動します。



障害キュー・マネージャー・ビューの使用

障害キュー・マネージャー・ビューには、障害キューで受信した失敗イベントが表示されます。このビューには、設定により指定した数の失敗メッセージが、受信順にリスト表示されます。

リスト表示されている障害キュー・メッセージに関するキュー・マネージャーからの情報を表示するには、メッセージ・フィールドをダブルクリックします。次の図に示すように、そのメッセージの情報がエディター・ビュー・パネルに読み取り専用で表示されます。



失敗イベントの処理

障害キュー・マネージャーには、HubRequest (要求/応答) およびエージェント・デリバリー (コンテナ管理イベントの場合) という 2 種類の対話パターンからの失敗イベント・メッセージがリストされ、これらを処理することができます。

イベント・メッセージをキューから削除するか、イベント・メッセージの再サブミットを試みることができます。

イベントを再サブミットするには、そのイベントの「メッセージ ID」フィールドのチェック・ボックスにマークを付け、右マウス・ボタンでクリックして「再サブミット」を選択します。

Adapter Monitor が、選択されたイベントの再サブミットを試みます。再サブミットが成功すると、イベントは障害キュー・マネージャー・ビューから除去されます。

「ResubmitTo」フィールドの値がヌルのメッセージを再サブミットすることはできません。イベントを再サブミットしようとして失敗するのは、メッセージ自体が無効であるか、ResubmitTo キューが無効または使用不可であるためです。

メッセージが無効になるのは、そのメッセージが無効な JMS タイプが含まれているか、そのメッセージをビジネス・オブジェクトに変換できないためです。どちらに該当する場合でも、無効なメッセージを再サブミットしようとして失敗したときには、その無効なメッセージに関するエラーが障害キュー・マネージャーによって表示されます。「OK」を選択すると、そのメッセージは再サブミットされなくなり、障害キューのビューから除去されます。「OK」を選択しないでダイアログを閉じると、そのメッセージは障害キューのビューに残されます。

メッセージ自体は有効であるにもかかわらず、メッセージの ResubmitTo キューがヌル、無効、または使用不可であるために再サブミットに失敗した場合は、「再サブミット」ダイアログが表示され、ここでメッセージの値を確認できます。キューにメッセージを残す（「キャンセル」を選択する）か、キューからメッセージを削除する（「OK」を選択する）かを選ぶことができます。

WebSphere MQ キューからのメッセージのクリア

WebSphere Business Integration Adapters にはサンプル・バッチ・ファイルが用意されています。サンプル・バッチ・ファイルを使用して、ビジネス・インテグレーション・システムの WebSphere MQ キューからのメッセージをクリアすることができます。ビジネス・インテグレーション・システムに問題があることから、処理のためにメッセージを除去できない場合は、キューのクリアが必要になることがあります。

WebSphere MQ キューからのメッセージをクリアするには、バッチ・ファイル `clear_mq.bat` (Windows) または `clear_mq` (UNIX) を実行します。これらのバッチ・ファイルは、`ProductDir¥templates` ディレクトリーにあります。このバッチ・ファイルにより、`crossworlds_mq.tst` ファイルに指定されたキューからメッセージがクリアされます。`crossworlds_mq.tst` の編集についての詳細は、91 ページの『WebSphere Business Integration Adapters バッチ・ファイルを使用して WebSphere MQ キューを構成する方法』を参照してください。

ログ・ファイルおよびトレース・ファイルの管理

次の各ツールには、メッセージのログギングおよびトレースを構成および表示するための、グラフィカル・ユーザー・インターフェース (GUI) が提供されています。

- コネクターのログギングおよびトレースをセットアップまたは変更する Connector Configurator。
- ログ・ファイルおよびトレース・ファイルを表示する Log Viewer。

Log Viewer を使用してログを表示するほかに、テキスト・エディターでログを開くこともできます。

始動時に、コネクターにより一時的ログ・ファイルが生成されます。このファイルには、コネクター・フレームワークに渡されるコネクター・プロパティーおよびビジネス・オブジェクト定義など、始動時にログとして記録されるメッセージがすべ

て格納されます。ファイル名は `broker_name_connector_name_tmp.log` で、`ProductDir` ディレクトリーに書き込まれます。コネクタを実行すると、標準のコネクタ構成プロパティに構成されたとおりに、ロギングおよびトレースが処理されます。

コネクタの構成ファイルで指定されたロケールの UTF-8 エンコードを使用してロギング・メッセージおよびトレース・メッセージを書き込んでいる間に、ロケールに依存しない形式で時刻情報が書き込まれます。

コネクタの始動時にログ・ファイルが存在していない場合は、コネクタの構成ファイルで指定されたロケーションに、ログ・ファイルも作成されます。ログ・ファイルが存在している場合は、新規のエントリーが追加されます。コネクタのログ・ファイルにサイズの上限值が設定されていない場合、そのサイズは最近に管理されてからの経過時間およびシステムを経由したトランザクション量によって決まります。コネクタのログ・ファイルのサイズが制限なしのままそのファイルを作成すると、ファイルのボリュームが増加し続けて、ファイルが開けなくなったりディスク・スペースが不足したりするような、事態を招くこともありえます。

トレースが使用可能である場合、トレース・ファイルは、まだ存在していなければ始動時に作成されます。トレース・ファイルのサイズは、コネクタのログ・ファイルの場合と同じ方法で管理することにより、ボリュームの肥大化によるトラブルを回避する必要があります。

109 ページの表 13 では、コネクタがロギングおよびトレース情報を格納するために使用するファイルが、紹介されています。

ログ・ファイルおよびトレース・ファイルを管理するには、Connector Configurator を使用して以下の作業を実行します。

- ログ・ファイルおよびトレース・ファイルの、サイズの上限值を指定する。
- サイズの上限值に達したら、ファイルを自動的にアーカイブする。
- 保存するアーカイブ・ファイルの数を指定する。

Connector Configurator を使用した、これらのオプションの設定に関する詳細については、108 ページの『ロギングおよびトレース・オプションの構成』を参照してください。

ログ・ファイルおよびトレース・ファイルのアーカイブ・ロギング

アーカイブ・ロギングの使用が可能である場合、コネクタのログ・ファイルまたはトレース・ファイルのボリュームが最大値に達するたびに、新規のアーカイブ・ファイルに名前が変更されます。アーカイブ・ファイルの名前は、オリジナルのログ・ファイルまたはトレース・ファイル名から派生しており、その名前には `_Arc_number` が挿入されています。

例えば、5 つのアーカイブ・ファイルが使用され、ログ・ファイルの名前が `Connector.log` である場合は、次のようになります。

- 最初に作成されるアーカイブの名前は `Connector_Arc_01.log` です。
- 新規のログ・ファイルが満杯になると、`Connector_Arc_01.log` の名前が `Connector_Arc_02.log` に変わります。

- 新規のログ情報は再度 Connector_Arc_01.log に格納され、5 つのアーカイブ・ファイルになるまで、このように循環します。
- 新規のログ・ファイルを作成するときに、アーカイブ・ファイルがすでに 5 つ存在している場合は、最も古いアーカイブ・ファイル (5 番) が削除されます。次に、残りのアーカイブ・ファイルは名前が変わって番号がアップし、アーカイブ数と構成した数が合うようになっています。図 21 に、この構成を使用したファイルの循環状況を示します。

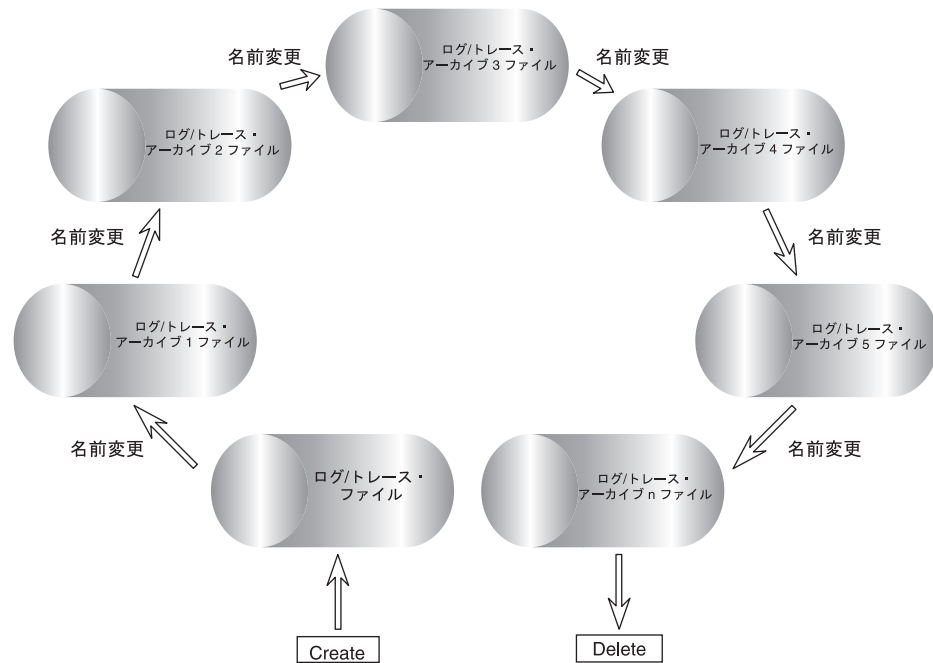


図 21. 循環アーカイブ・ロギング

詳細については、108 ページの『ロギングおよびトレース・オプションの構成』の構成作業を参照してください。

その他のファイルの管理

ログ・ファイルおよびトレース・ファイルはアーカイブ・ロギングで管理できますが、各アプリケーションに固有なその他のログ・ファイルについては、手作業で管理する必要があります。これらのファイルの多くは、存在していない場合は実行時に作成されます。既存のファイルである場合は、そこに新規の情報が追加されます。

ファイル管理手順は任意に使用することができますが、IBM では次のログ・ファイル管理を定期的に行うことをお勧めします。

- ファイル名に日付を付加して名前を変更する。
- ファイルをアーカイブ・ディレクトリーに移動する。

Log Viewer を使用してコネクタ・メッセージを表示する方法

Log Viewer を使用すると、コネクタのログ・ファイルおよびトレース・ファイルに格納されているメッセージを閲覧することができます。出力表示内容のソートおよびフィルター、さらにファイルの印刷、保存、電子メール送信を行うこともできます。ロギングおよびトレース・オプションと、生成されたファイルのロケーションは、コネクタの構成ファイル内のプロパティとして指定されています。

注: Log Viewer は、Windows 2000 マシン上でのみ動作します。Log Viewer を使用して UNIX ログ・ファイルを構成または表示するには、UNIX マシンから Windows マシンにログ・ファイルをコピーして、Windows マシンで表示します。

Log Viewer の始動は、次のいずれかの方法で行います。

- 「IBM WebSphere Business Integration Adapters Start」メニューのショートカットで、「ツール」サブメニューの「Log Viewer」オプションを選択する。「ファイル」メニューの「開く」オプションを使用して、ログ・ファイルをブラウズする。
- 「スタート」メニューの「実行」コマンドを使用して LogViewer.exe ファイルをブラウズする。「ファイル」メニューの「開く」オプションを使用して、ログ・ファイルをブラウズする。

「Log Viewer」メニュー・オプションを使用すると、以下の各タスクが実行できます。

- 82 ページの『Log Viewer の設定の指定』
- 85 ページの『メッセージの表示法の変更』
- 87 ページの『Log Viewer 表示出力の制御』

サンプル・ログ・ファイルを表示している Log Viewer を、82 ページの図 22 に示しています。

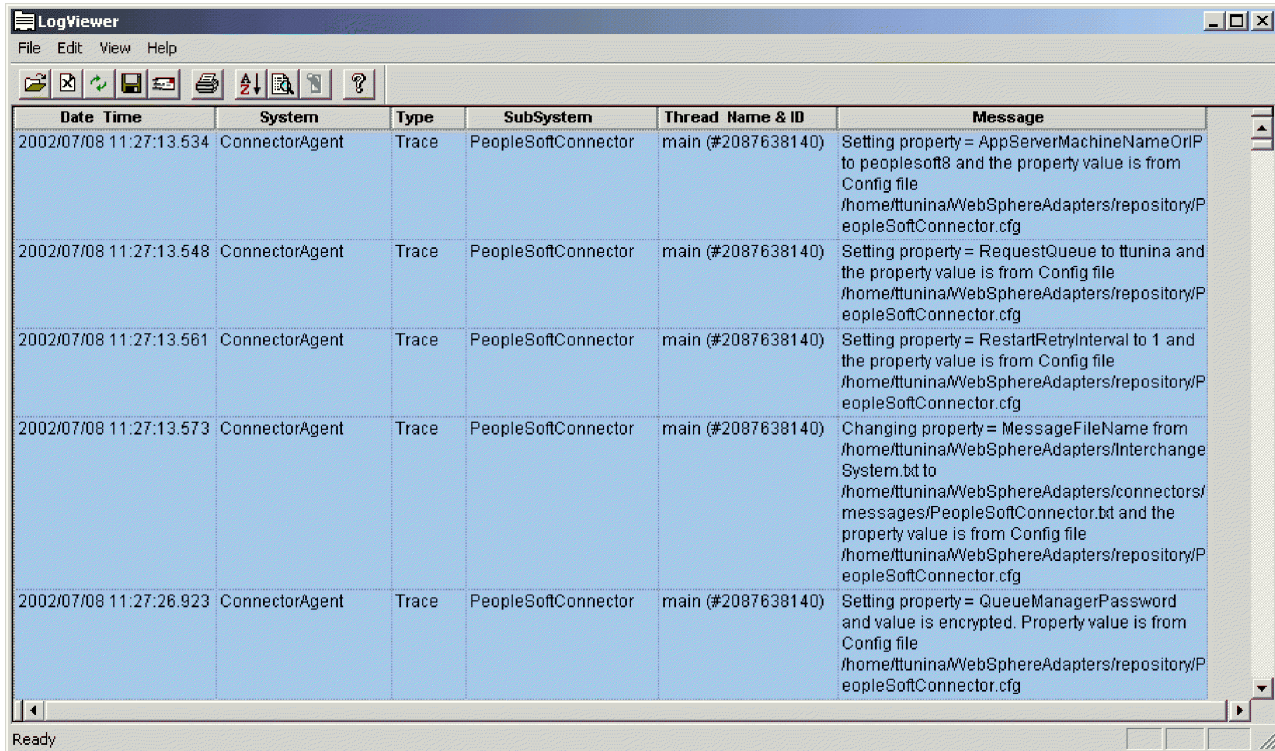


図 22. サンプル・ログを表示する Log Viewer

Log Viewer の設定の指定

1. Log Viewer の設定を指定するには、メニュー・バーで「編集」 > 「設定」と選択します。

「ユーザー構成オプション - 一般プロパティ」ダイアログ・ボックスが表示されます (83 ページの図 23 を参照)。

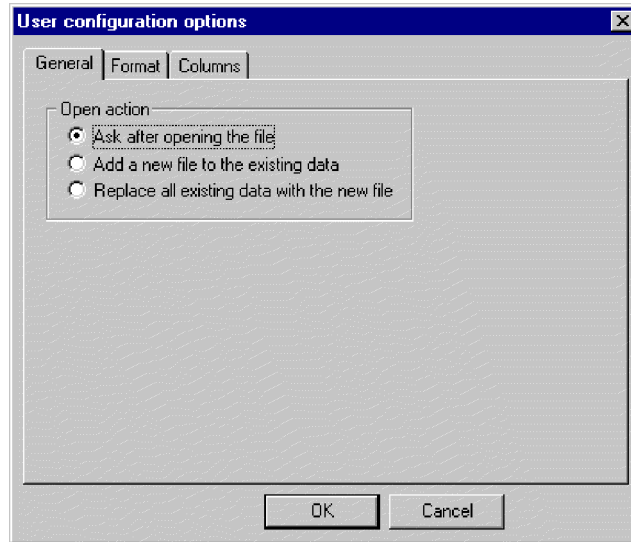


図 23. 「Log Viewer ユーザー構成オプション - 一般プロパティ」画面

このダイアログ・ボックスで、ログ・ファイルを開いたときの表示方法を指定します。使用可能な選択項目は次のとおりです。

- ログ・ファイルを開くたびに設定内容を照会する。
 - 開くログ・ファイルと表示中のログ・ファイルをマージする。
 - 表示中のログ・ファイルを、開くログ・ファイルの内容で置き換える。
2. 背景色と Log Viewer メッセージのフォントを変更するために、「フォーマット」タブをクリックする。

「ユーザー構成オプション - フォーマット・プロパティ」ダイアログ・ボックスが表示されます (図 24 を参照)。

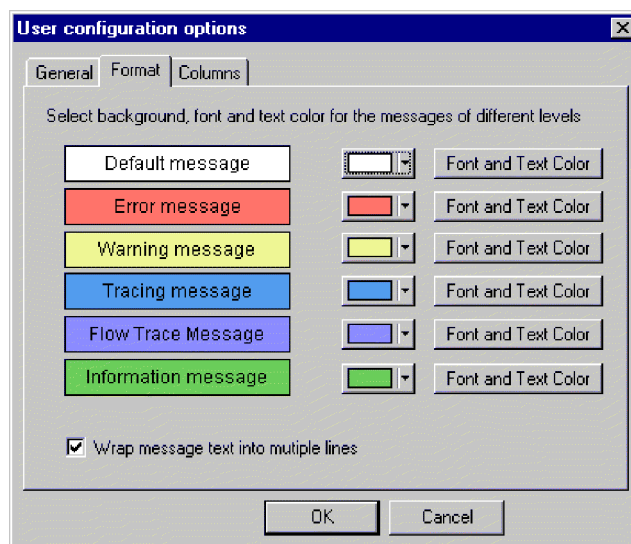


図 24. 「Log Viewer ユーザー構成オプション - フォーマット・プロパティ」画面

このダイアログ・ボックスで、ログ・メッセージの表示方法を指定します。使用可能な選択項目は次のとおりです。

- 表示するメッセージ・タイプごとに異なる背景色とフォントを割り当てて、その重要度が簡単に識別できるようにする (例えば、警告メッセージの場合は赤色の背景色と大きなフォント)。
- メッセージのテキストがカラムの幅を上回る場合は、そのテキストを折り返す。

注: フロー・トレース・メッセージは、WebSphere Message Broker を使用するコネクターでは生成されません。

3. 表示する Log Viewer 列を変更するには、「列」タブをクリックします。

「ユーザー構成オプション - 列プロパティ」ダイアログ・ボックスが表示されます (図 25 を参照)。

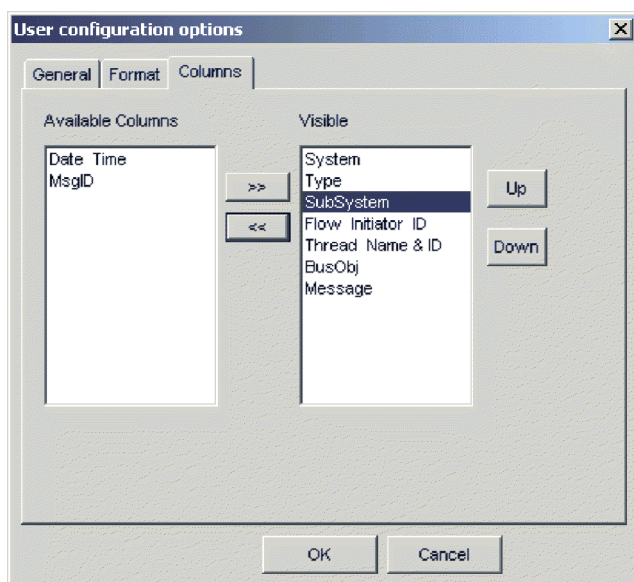


図 25. 「Log Viewer ユーザー構成オプション - 列プロパティ」画面

このダイアログ・ボックスでは、Log Viewer にどの列を表示するかを、次のように指定できます。

- 列を表示するには、「使用可能な列」ペインで列名を強調表示して「>>」ボタンをクリックし、「可視」ペインに移動する。
- 列を非表示にするには、「可視」ペインで列名を強調表示して「<<」ボタンをクリックし、「使用可能な列」ペインに移動する。
- 「使用可能な列」ペイン内の任意の列名をクリックし、「上へ」または「下へ」ボタンをクリックして、Log Viewer 表示内での左から右への表示順を変更する。「上へ」ボタンで列が左へ移動し、「下へ」ボタンで列が右へ移動します。
- 「空の列を自動的に非表示にする」の横のチェック・ボックスをクリックして、Log Viewer の表示をコンパクトにする。

注: 「フロー・イニシエーター ID」列は、WebSphere Message Broker を使用するコネクタには関係ありません。

メッセージの表示法の変更

「表示」メニューには Log Viewer の表示を変更するその他のオプションが用意されています。そのメニューで、以下の機能が使用できます。

- Log Viewer ツールバーの表示/非表示。
- Log Viewer ステータス・バーの表示/非表示。
- ウィンドウの複数のビューへの分割。
- 時刻範囲などの「フィルター」タブのフィルター・オプションをチェックする、またはメッセージ・タイプによる、メッセージすべてのフィルター操作または表示 (86 ページの図 26 および 87 ページの表 10 を参照)。フィルター・オプションの設定は、次の手順で行います。
 1. メニュー・バーから「表示」>「フィルター」>「フィルターの使用」の順に選択する。「フィルターの設定」ダイアログ・ボックスが表示されます。
 2. 「フィルターの活動化」領域で、適用するフィルター・オプションを含んだタブに関連するボックスをクリックする。
 3. 「OK」をクリックしてフィルターを使用可能にする。

フィルター出力は、ツールバーの「フィルターの切り替え」ボタンで、オンまたはオフに切り替えることができます。

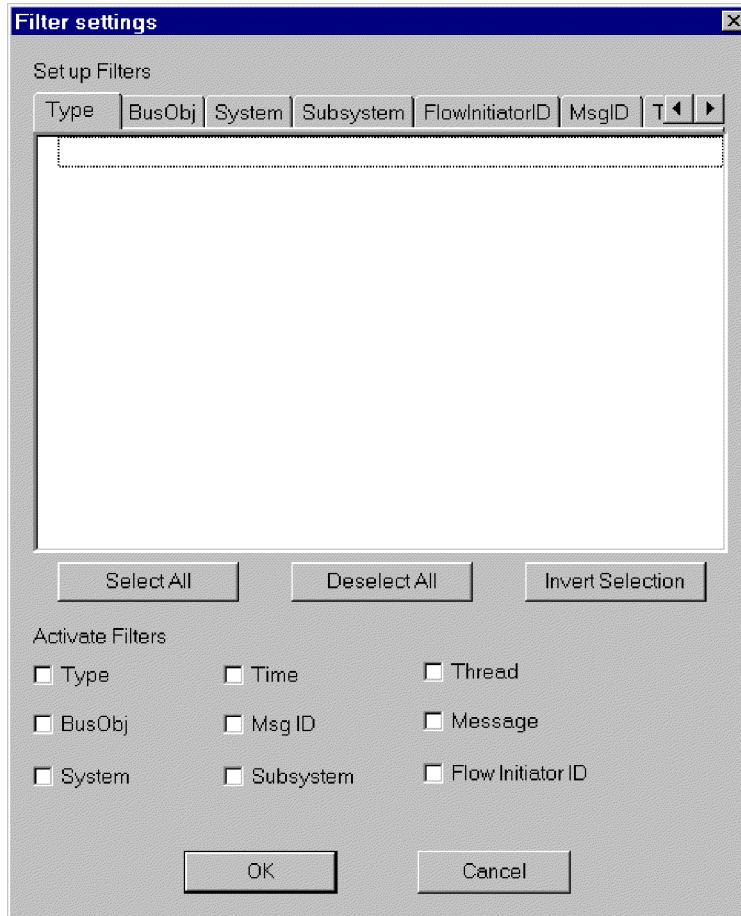


図 26. Log Viewer の「フィルター」画面

注: 「フロー・イニシエーター ID」タブは、WebSphere Message Broker を使用するコネクタには関係ありません。

- メッセージのソートについては、図 27 で「ソート」オプションを紹介します。各ソート・フィールドにある下矢印をクリックして、「日付/時刻」または「イベント ID」を選択します。昇順または降順のいずれかでソートします。

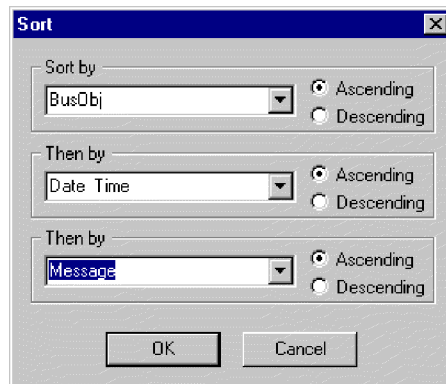


図 27. Log Viewer の「ソート・プロパティ」画面

Log Viewer 表示出力の制御

Log Viewer 出力を制御するためのオプションがいくつか用意されています。「ファイル」メニューには、印刷プレビュー、印刷、保管、最新表示、電子メール宛先への送信、さらにページ設定やヘッダーおよびフッターの、スタイルの決定に関するオプションが提供されています。ヘッダーおよびフッター用の変数は次のとおりです。

変数名	説明
\$F	ファイル名。
\$A	アプリケーション名。
\$P	ページ番号。
\$N	合計ページ数。
\$D	日付 (直後にパラメーターを追加可能 (例えば、\$D{%y:%h:%m}))。

メッセージのフィルター操作

Log Viewer に表示されるメッセージをフィルター操作するには、「Log Viewer」メニュー・バーから「表示」->「フィルター」->「フィルターの使用」の順に選択します。「フィルターの設定」ダイアログ・ボックスには、ロギング・メッセージ・フォーマットのパラメーターに対応するカテゴリーが表示されています。メッセージ・フォーマット・パラメーターを、表 10 で紹介します。

表 10. ログ・ファイルのメッセージ・フォーマット・パラメーター

変数	説明
<i>Time</i>	タイム・スタンプ: 年/月/日時 形式でのロギング日付。
<i>System</i>	WMQI 統合ブローカーを使用するコネクタの場合、 <i>system</i> はコネクタのアプリケーション固有のコンポーネント。
<i>Thread</i>	スレッド名およびスレッド ID。
<i>Name</i>	ClarifyConnector などのコンポーネント名。
<i>MsgType</i>	メッセージの重大度を示す。88 ページの表 11 を参照してください。
<i>MsgID</i>	メッセージ番号。
<i>SubSystem</i>	コネクタ名。
<i>BO</i>	ビジネス・オブジェクト名。
<i>MsgText</i>	メッセージ番号の関連テキスト。
<i>BOD</i>	ビジネス・オブジェクト・ダンプ。ビジネス・オブジェクトに含まれているデータ。

「フィルターの設定」ダイアログで、まず使用するフィルター・カテゴリーを選択し、次に各カテゴリーから表示する項目を指定し、さらに現在の Log Viewer ビューでアクティブ状態にするフィルターを選択します。

次の手順に従います。

1. 「フィルターの設定」ダイアログで、「フィルターのセットアップ」にあるタブを選択して、メッセージのフィルター操作に使用する項目を表示する。例えば、

メッセージのタイム・スタンプに従ってフィルター操作する場合は、「時刻」を選択します。複数のフィルターをセットアップして、これらを別々または一緒に使用することができます。

- 表示された項目リストから、Log Viewer でメッセージを表示する項目をそれぞれ選択する。例えば、2002 年 3 月 5 日午前 9 時から 2002 年 3 月 6 日午後 5 時までのタイム・スタンプのあるメッセージのみを表示する場合、「時刻」タブでこの時間範囲を選択します。

リスト・ボックスの下にあるボタンを使用して、表示された項目すべての選択、表示された項目すべての選択解除、または現在の選択肢の反転を行うことができます。

- 「フィルターの活動化」で、活動化する各フィルター・タイプのボックスを選択する。例えば、特定のタイム・スタンプで、特定のメッセージ ID を持つメッセージのみを表示する場合は、メッセージ ID フィルターと時刻フィルターの両方をアクティブ状態にします。
- 「OK」をクリックする。「フィルターの設定」ダイアログ・ボックスがクローズされ、Log Viewer の最新表示により、フィルターを通過したメッセージのみが表示されます。

カテゴリーに従ったフィルター操作のほかに、特定のテキスト・ストリングを含むメッセージのみを表示することもできます。このためには、「フィルターのセットアップ」で「メッセージ」を選択し、表示するメッセージの特定のテキストを入力して、「フィルターの活動化」で「メッセージ」のボックスを選択します。

メッセージ・タイプ

表 11 で、WebSphere Business Integration Adapters から発行されるメッセージのタイプについて紹介します。

表 11. メッセージ・タイプ

メッセージ・タイプ	説明
情報	通知のみ。対処は必要ありません。
警告	InterChange Server で選択されたデフォルトの条件。
エラー	調査を必要とする重大な問題。
致命的エラー	オペレーションを停止し、報告を必要とするエラー。
トレース	指定されたトレース・レベルのトレース情報。
フロー・トレース	ビジネス・オブジェクトのフロー・トレース情報。
内部エラー	調査を必要とする、重大な内部的問題。
内部致命的エラー	オペレーションを停止する内部エラー。報告の必要あり。

注: タイプが内部エラーまたは内部致命的エラーのメッセージが表示された場合は、障害発生時での環境の状態を記録して IBM テクニカル・サポートまでご連絡ください。

第 8 章 WebSphere Business Integration システムの構成

本章では、WebSphere Business Integration システムのコンポーネントである、統合ブローカー、ビジネス・オブジェクト、およびコネクターの構成方法について説明します。本章は、次の各セクションから構成されています。

- 『構成タスクの概要』
- 『コネクターと連動するようにメッセージ・ブローカーを構成する方法』
- 93 ページの『ビジネス・オブジェクト定義の作成』
- 94 ページの『メッセージ・ブローカー・プロジェクトの作成』
- 105 ページの『アプリケーションでコネクターを使用可能にする』
- 106 ページの『コネクターの構成』
- 112 ページの『メッセージ・フローの定義』
- 113 ページの『Visual Test Connector を使用したインターフェースの検証』

構成タスクの概要

ビジネス・インテグレーション・システムを構成するには、以下の作業を実行する必要があります。

1. 必要なキューを定義して、コネクターをサポートするようにメッセージ・ブローカーを構成する。
2. コネクターが使用するビジネス・オブジェクト定義を生成する。
3. メッセージ・ブローカー・プロジェクトを作成して、メッセージ・ブローカーに配置する。
4. コネクター用の標準およびアプリケーション固有の構成プロパティを構成する。
5. コネクター用のトレース、ロギング、およびメッセージングの各オプションを構成する。
6. メッセージ・フローを作成して、メッセージ・ブローカーが受信したビジネス・オブジェクト・メッセージを処理する方法を定義する。

次に、これら各タスクの詳細について説明します。

コネクターと連動するようにメッセージ・ブローカーを構成する方法

WebSphere Message Broker がコネクターと連動できるようにするには、コネクターと統合ブローカー間でメッセージを送信する WebSphere MQ キューを構成し、また最適なキュー構成を定義しておく必要があります。また、コネクターの構成ファイルには、正しく指定されたキューおよびキュー・マネージャー情報が格納されていることを確認する必要があります。

46 ページの『メッセージ・キュー』では、WebSphere Business Integration システムで WebSphere MQ キューがどのように使用されるかについて説明しています。107 ページの『キュー・マネージャーによる接続モードの設定』では、コネクターの構

成ファイルに接続モードを指定する方法について説明しています。WebSphere MQ キュー、キュー・マネージャー、およびキュー構成の詳細については、「*WebSphere MQ: 相互通信*」を参照してください。

WebSphere MQ キューの構成

ビジネス・インテグレーション・システムでは、下記のプロパティを使用してキューを構成する必要があります。

注: 107 ページの『コネクターが使用するキューの指定』でコネクターを構成する際、各キューの名前をコネクターの標準プロパティとしてコネクターの構成ファイルに指定する必要があります。

- **DeliveryQueue:** コネクター・フレームワークから WebSphere Message Broker にイベント・デリバリー・メッセージを配信します。
- **RequestQueue:** WebSphere Message Broker からコネクター・フレームワークに要求メッセージを配信します。
- **ResponseQueue:** このキューは、WebSphere Message Broker では使用されませんが、WebSphere InterChange Server との整合性を保つために定義する必要があります。
- **FaultQueue:** コネクター・フレームワークからメッセージ・ブローカーに障害メッセージを配信します。コネクター・フレームワークは、返信先キューにメッセージを配置することができないとき、このキューにメッセージを配置します。
- **SynchronousRequestQueue:** 同期応答を必要とする要求メッセージを、コネクター・フレームワークからメッセージ・ブローカーに配信します。このキューは、コネクターが同期実行方式を使用する場合にのみ必要です。同期実行の場合、コネクター・フレームワークは、`SynchronousRequestQueue` にメッセージを送信し、`SynchronousResponseQueue` でメッセージ・ブローカーから戻される応答を待機します。コネクターに送信される応答メッセージには、元のメッセージの ID を指定する相関 ID が含まれています。
- **SynchronousResponseQueue:** 同期要求に応答して送信された応答メッセージを、メッセージ・ブローカーからコネクター・フレームワークに配信します。このキューは、コネクターが同期実行方式を使用する場合にのみ必要です。
- **AdminInQueue:** メッセージ・ブローカーからコネクター・フレームワークに管理メッセージを配信します。
- **AdminOutQueue:** コネクター・フレームワークからメッセージ・ブローカーに管理メッセージを配信します。

キューを定義する方法

次のいずれかの方法を使用して、ご使用のアダプターのために必要な WebSphere MQ キューを構成することができます。

- WebSphere Business Integration Adapters に用意されているバッチ・ファイルをカスタマイズし、実行する。
- WebSphere MQ Explorer を使用する。
- WebSphere MQ コマンドを実行する。

ヒント

キューが関連付けられているコネクタを簡単に識別するには、コネクタ名をキュー名の接頭部として使用します。例えば、Clarify コネクタのイベント・デリバリー・キューの名前は次のようにします。
clarifyconnector/deliveryqueue。

WebSphere Business Integration Adapters バッチ・ファイルを使用して

WebSphere MQ キューを構成する方法: WebSphere Business Integration Adapters には、1 セットのバッチ・ファイルが用意されており、それらのバッチ・ファイルを使用して、展開するアダプターに必要な WebSphere MQ キューを構成することができます。ProductDir¥templates にあるバッチ・ファイルは、以下のファイルから構成されています。

- **configure_mq.bat** (Windows)
configure_mq (UNIX)

このバッチ・ファイルを実行して、crossworlds_mq.tst に指定されている WebSphere MQ キューを構成します。

- **crossworlds_mq.tst**

このファイルを編集して、ビジネス・インテグレーション・システムの WebSphere MQ キューを指定します。このファイルは、WebSphere MQ キューからのメッセージをクリアするために WebSphere Business Integration Adapters が用意しているバッチ・ファイル、configure_mq.bat および clear_mq.bat によって入力データとして読み取られます。

clear_mq.bat の使用についての詳細は、78 ページの『WebSphere MQ キューからのメッセージのクリア』を参照してください。

crossworlds_mq.tst ファイルの内容を以下に示します。この 1 つのファイルを使用して、構成する各アダプターに必要なキューを指定できます。このファイルは、次のように編集してください。

1. 以下のステートメントを削除します。

```
DEFINE QLOCAL(IC/SERVER_NAME/DestinationAdapter)
DEFINE QLOCAL(AP/DestinationAdapter/SERVER_NAME)
```

これらのステートメントは、WebSphere InterChange Server を使用するビジネス・インテグレーション・システムにのみ適用されます。

2. DEFINE QLOCAL(AdapterName/AdminInQueue) から始まるステートメントをテンプレートとして使用して、配置するアダプターごとに個別のキュー定義ステートメント・セットを作成します。
3. リモート・キュー定義を指定しバインディング・モードを使用している場合は、構成する必要がある各キュー・マネージャーごとに、要求された情報を使用して、ステートメント DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) をカスタマイズしてください。キュー構成にクライアント・モードを使用している場合には、ステートメントは現状のままにしてください。サポートされているキュー構成の詳細については、92 ページの『キュー構成の定義』を参照してください。

```

*****/
*                                                                    */
*   Define the local queues for all Server/Adapter pairs.           */
*   For MQ queues, they must have the following definition:        */
*       Application = DEFINE QLOCAL (AP/AdapterName/ServerName)    */
*                                                                    */
*   Example:                                                         */
*   DEFINE QLOCAL(AP/ClarifyConnector/CrossWorlds)                 */
*                                                                    */
*   DEFINE QLOCAL(AP/SAPConnector/CrossWorlds)                     */
*                                                                    */
*   If your server is named something different than 'CrossWorlds' */
*   make sure to change the entries to reflect that.               */
*****/
*   DEFINE QLOCAL(IC/SERVER_NAME/DestinationAdapter)
*   DEFINE QLOCAL(AP/DestinationAdapter/SERVER_NAME)
*****/
*   For each JMS queue (delivery Transport is JMS),
*   default values follow the convention:
*       AdapterName/QueueName
*****/
*   DEFINE QLOCAL(AdapterName/AdminInQueue)
*   DEFINE QLOCAL(AdapterName/AdminOutQueue)
*   DEFINE QLOCAL(AdapterName/DeliveryQueue)
*   DEFINE QLOCAL(AdapterName/RequestQueue)
*   DEFINE QLOCAL(AdapterName/ResponseQueue)
*   DEFINE QLOCAL(AdapterName/FaultQueue)
*   DEFINE QLOCAL(AdapterName/SynchronousRequestQueue)
*   DEFINE QLOCAL(AdapterName/SynchronousResponseQueue)
*****/
*   Define the default CrossWorlds channel type                       */
*****/
*   DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
*****/
*   End of CrossWorlds MQSeries Object Definitions                   */
*****/

```

WebSphere MQ Explorer を使用して WebSphere MQ キューを構成する方法:

WebSphere MQ Explorer を使用したキューの構成については、WebSphere MQ Explorer を開いて、そのオンライン・ヘルプを参照してください。

WebSphere MQ コマンドを使用して WebSphere MQ キューを構成する方法:

WebSphere MQ コマンドを使用してキューを構成する方法の詳細は、「WebSphere MQ: システム管理ガイド」および「WebSphere MQ: MQSC コマンド・リファレンス」を参照してください。

キュー構成の定義

WebSphere Business Integration システムでは、一部のキュー・マネージャーとキュー構成をサポートしています。コネクタは、以下のどのモードにおいても、キュー・マネージャーと通信することができます。

バインディング・モード

バインディング・モードでは、WebSphere Message Broker とコネクタはキュー・マネージャーを使用して直接通信することが可能で、TCP/IP 接続は使用しません。統合ブローカーとコネクタは、同じキュー・マネージャーを共用できるように、同一のマシンにインストールする必要があります。デフォルトでは、このバインディング・モードになります。

リモート・キュー定義でのバインディング・モード

WebSphere Message Broker とコネクタが別々のマシンにインストールされ、各マシンで独自のキュー・マネージャーが動作している場合、コネクタと統合ブローカーは、バインディング・モードを使ってそれぞれのキュー・マネージャーとそのまま通信することができます。ただし、次の例で説明するとおりに、リモート・キュー定義を指定する必要があります。

例えば、`brokerQM` は統合ブローカーが使用するキュー・マネージャーで、`connQM` はコネクタが使用するキュー・マネージャーであるとしています。双方のキュー・マネージャー間で通信ができるようにするには、次のチャンネル定義をセットアップする必要があります。

- コネクタからメッセージ・ブローカーにメッセージを送信する各キューでは、`brokerQM` のローカル・キューを指すリモート・キュー定義を `connQM` に作成する必要があります。この要件は次のキューに適用されます。
 - `DeliveryQueue`
 - `SynchronousRequestQueue`
 - `FaultQueue`
 - `AdminOutQueue`
- 統合ブローカーからコネクタにメッセージを転送する各キューでは、`connQM` のローカル・キューを指すリモート・キュー定義を `brokerQM` に作成する必要があります。この要件は次のキューに適用されます。
 - `RequestQueue`
 - `AdminInQueue`
- 応答先キューの場合：統合ブローカーがコネクタ・フレームワークに要求メッセージを送信するときに、メッセージ・ヘッダーにキュー・マネージャーおよび応答の送信先となるキューを指定します。これは、コネクタ・フレームワークから統合ブローカーに送信される要求にも当てはまります。115 ページの『付録 A. WebSphere MQ のメッセージ・フォーマット』では、このヘッダー・フィールドの詳細について説明しています。特定の管理用タスクを実行して、応答メッセージが正しい応答先キューへ送信されるようにする必要があります。これらについては、「*WebSphere MQ: 相互通信*」を参照してください。

クライアント・モード

メッセージ・ブローカーとコネクタがそれぞれのキュー・マネージャーと通信するために TCP/IP を使用する必要がある場合は、クライアント・モード接続を使用しなければなりません。通信は、基本となるトランスポート層として TCP/IP を使用するクライアント接続を介して行われます。

ビジネス・オブジェクト定義の作成

コネクタで使用されるビジネス・オブジェクト定義を作成する場合のオプションは、いくつか用意されています。すべてのオプションの詳細が「*ビジネス・オブジェクト開発ガイド*」に解説されています。

注: ただし、WebSphere Business Integration Adapters ではアプリケーション固有のビジネス・オブジェクトのみが使用され、汎用ビジネス・オブジェクトは使用

されません。本書でビジネス・オブジェクトというときはすべて、アプリケーション固有のビジネス・オブジェクトを意味しています。汎用ビジネス・オブジェクトは、WebSphere InterChange Server 統合ブローカーを基にしたビジネス・インテグレーション・システムで使用されます。「ビジネス・オブジェクト開発ガイド」など、WebSphere Business Integration Adapters ライブラリーの一部の資料は、WebSphere InterChange Server ライブラリーの一部をなすものでもあるため、両タイプのビジネス・オブジェクトに言及しています。

ビジネス・オブジェクト定義を作成するためのオプションは以下のとおりです。

- ODA (Object Discovery Agent) を使用して、アプリケーション固有のビジネス・オブジェクト定義を生成する。ODA により、アプリケーション内の指定されたオブジェクトが検査され、ビジネス・オブジェクト属性に対応するこれらのオブジェクト属性の要素とその属性が「発見」され、情報を表現するビジネス・オブジェクト属性定義が生成されます。Business Object Designer には、Object Discovery Agent にアクセスしたり、そのエージェントと対話的に作業を行ったりするためのグラフィカル・インターフェースが用意されています。構成するコネクタに ODA (Object Discovery Agent) が用意されているかどうかを判別するには、当該コネクタ用のアダプター・ユーザズ・ガイドを参照してください。
- ODA がビジネス・インテグレーション・アダプターに組み込まれていない場合は、ODK (Object Discovery Agent Development Kit) を使用して ODA を開発してから、それをアプリケーションに対して実行する。
- Business Object Designer ツールを使用して、ビジネス・オブジェクト定義を手動で作成する。

さらに、アダプターの多くにはサンプルのビジネス・オブジェクトが付属しています。付属している場合、サンプルは次の製品ディレクトリー内に存在しています。

```
ProductDir¥connectors¥ConnName¥Samples
```

アプリケーション固有のビジネス・オブジェクト用のビジネス・オブジェクト定義を作成し終わったら、183 ページの『付録 G. Visual Test Connector の使用』をお読みください。これには、ビジネス・オブジェクト定義の作成後にそれらの定義をテストする方法が説明されています。

メッセージ・ブローカー・プロジェクトの作成

コネクタがサポートするビジネス・オブジェクト定義を作成したら、WebSphere Business Integration Message Broker の場合はメッセージ・ブローカー・ワークスペースに、WebSphere MQ Integrator または WebSphere MQ Integrator Broker の場合はメッセージ・ブローカーに、作成したオブジェクト定義を配置する必要があります。この作業は、System Manager を使用して行います。System Manager は Adapter Framework と共にインストールされ、各アダプターを構成し、管理するためのグラフィカル・ユーザー・インターフェースを備えています。(System Manager の詳細については、147 ページの『付録 E. System Manager と Eclipse Workbench』を参照してください。)

注: コネクタは、メッセージ・ブローカーにプロジェクトを配置する前に、インストールしておく必要があります。インストール手順については、「WebSphere Business Integration Adapters インストール・ガイド」を参照してください。

System Manager を始動するには、Windows の「スタート」メニューから「**IBM WebSphere Business Integration Adapters**」>「**ツール**」>「**System Manager**」を選択します。System Manager のパースペクティブが表示されていることを確認してください。新規プロジェクトを作成する前に、まずインポーター・パスを指定する必要があります。

インポーター・パスおよびワークスペース・パスの指定

ブローカー・インポーターおよびメッセージ・ブローカーのワークスペース・ディレクトリーが同じワークスペース内に存在しない場合は、その両方に対するパスを指定してから、メッセージ・ブローカーにプロジェクトを配置する必要があります。

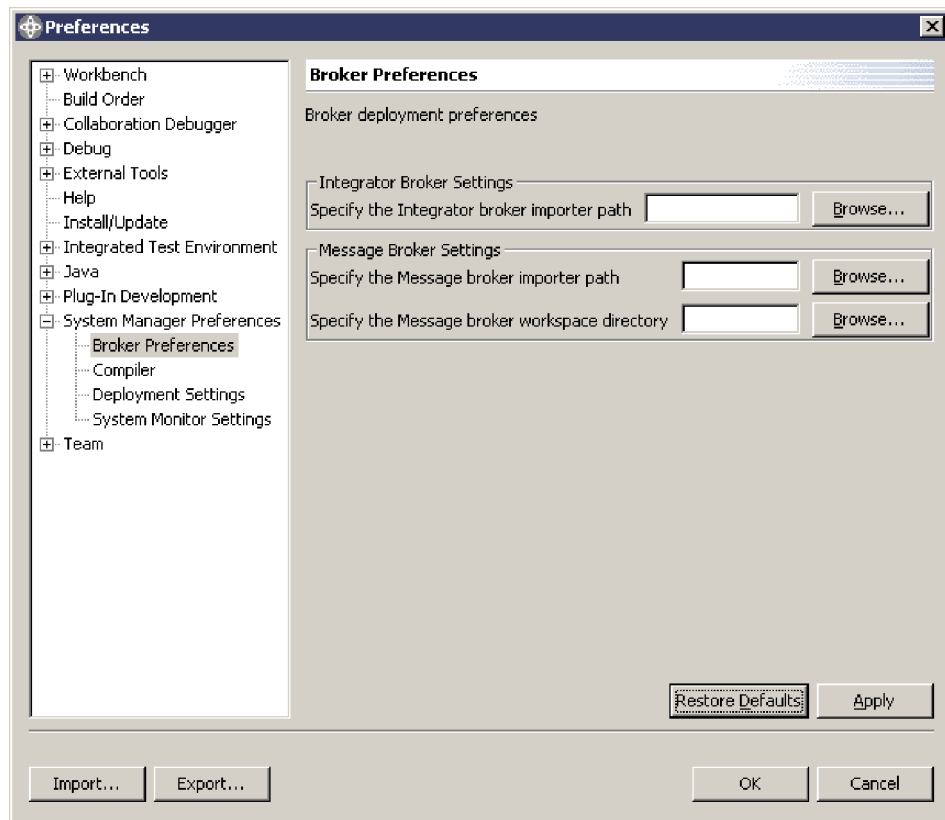
注: WebSphere Business Integration Message Broker にビジネス・オブジェクトを配置するときに、System Manager と Message Broker Toolkit が同じプラグイン・ディレクトリーに存在する場合は、このステップを実行する必要はありません。配置時に、指定されたプロジェクトが現在の System Manager ワークスペース内に作成されます。

ヒント

配置後、System Manager ワークベンチ内のブローカー・アプリケーション開発 (Broker Application Development) パースペクティブに、配置対象メッセージ・セット・プロジェクトの項目が表示されます。ワークベンチの始動時に、System Manager と Message Broker Toolkit の両方のプラグインが起動される場合、それらは同じワークスペース内に存在しています。System Manager と同じワークベンチ内にブローカー・アプリケーション開発 (Broker Application Development) パースペクティブが表示される場合は、Message Broker Toolkit プラグインも起動されています。

ブローカー・インポーターとメッセージ・ブローカーのワークスペースが、別のディレクトリーに存在する場合は、以下のようにして、それらのパスを指定します。

1. System Manager から、「**ウィンドウ**」>「**設定**」>「**System Manager の設定 (System Manager Preferences)**」>「**ブローカーの設定 (Broker Preferences)**」を選択します。次のウィンドウが表示されます。



2. パス名を次のように入力します (または「ブラウズ」をクリックしてパス名を探します)。

WebSphere MQ Integrator Broker または WebSphere MQ Integrator の場合、(「インテグレーター・ブローカー 設定 (Integrator Broker Settings)」の下に) インテグレーター・インポーターのパスを入力します。インポーターは、`mqsilmpXMLSchema.exe` と呼ばれ、ブローカーのインストール先の `bin` ディレクトリーにある必要があります。

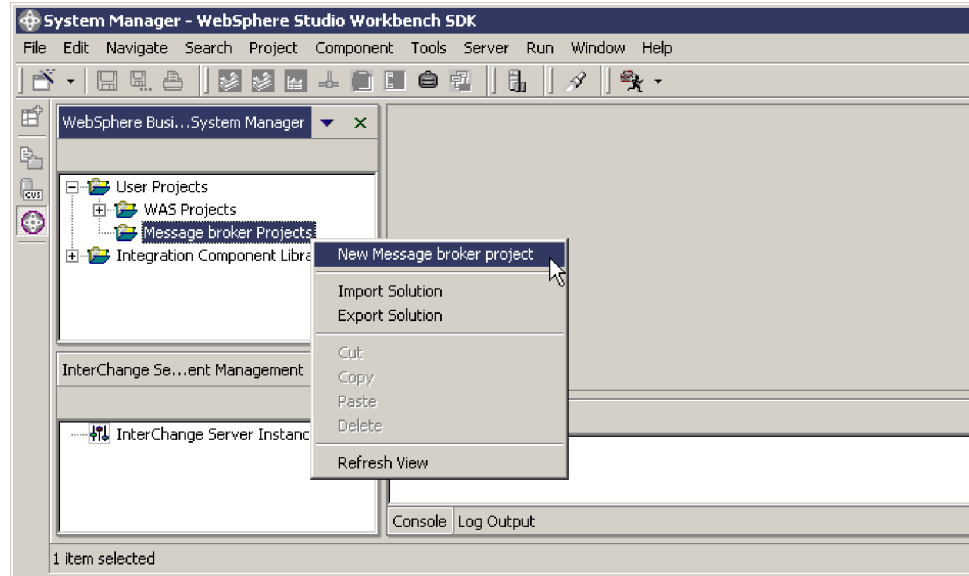
WebSphere Business Integration Message Broker の場合は、メッセージ・ブローカーのインポーター・パスおよびメッセージ・ブローカー・ワークスペース・ディレクトリーを (「メッセージ・ブローカーの設定 (Message Broker Settings)」の下に) 入力します。インポーターは、`mqsicreatemsgdefs.exe` と呼ばれ、ブローカーのインストール先の `eclipse` ディレクトリーにある必要があります。System Manager ワークスペースではなく、必ずメッセージ・ブローカー・ワークスペースを選択してください。メッセージ・ブローカー・ワークスペースのデフォルトのパスは、`eclipse\workspace` です (メッセージ・ブローカー・ワークスペースは、メッセージ・セットとメッセージ・フロー・プロジェクト、および WebSphere Business Integration Message Broker ツールのその他のすべてのプロジェクトの格納に使用されるワークスペースです)。

注: 入力するパス名は、絶対パス名である必要があります。

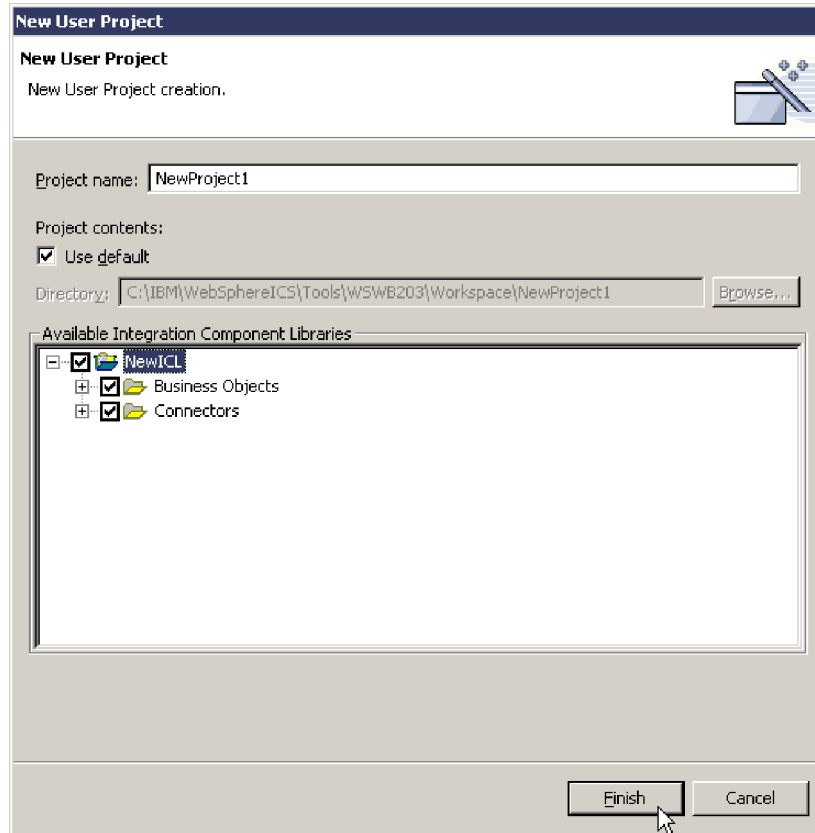
ユーザー・プロジェクトの新規作成

ユーザー・プロジェクトを新規に作成するには、次の手順で行います。

1. 以下の図に示すように、「ユーザー・プロジェクト」メニューから「メッセージ・ブローカー・プロジェクト (Message broker Projects)」を選択した後、「新規メッセージ・ブローカー・プロジェクト (New Message broker project)」を選択します。



2. 「新規ユーザー・プロジェクト」ウィンドウが表示されます。プロジェクトの名前を入力します。例えば、次のように入力します。



3. 「使用可能な統合コンポーネント・ライブラリー」ウィンドウに、検出された統合コンポーネント・ライブラリーのリストが表示されます (注: 各ライブラリーへのパスは、System Manager のインストール時に入力する必要があります)。正符号 (+) を選択して、「使用可能な統合コンポーネント・ライブラリー」ウィンドウのチェック・ボックスを展開し、このプロジェクトに組み込むビジネス・オブジェクト定義とコネクターを選択します。

注: 「ビジネス・オブジェクト」の横のチェック・ボックスを選択すると (例のように)、使用可能なすべてのビジネス・オブジェクト定義が選択されます。
「コネクター」の横のチェック・ボックスを選択すると (例のように)、使用可能なすべてのコネクターが選択されます。

4. 「完了」をクリックします。System Manager の「ブローカー・プロジェクト (Broker Projects)」リストに新規プロジェクトの名前が表示されます。
5. 「ユーザー・プロジェクト」パネルで、表示された新規プロジェクトの名前、および「ビジネス・オブジェクト」と「コネクター」の各項目を展開すると、選択したビジネス・オブジェクトとコネクターの名前が表示されます。

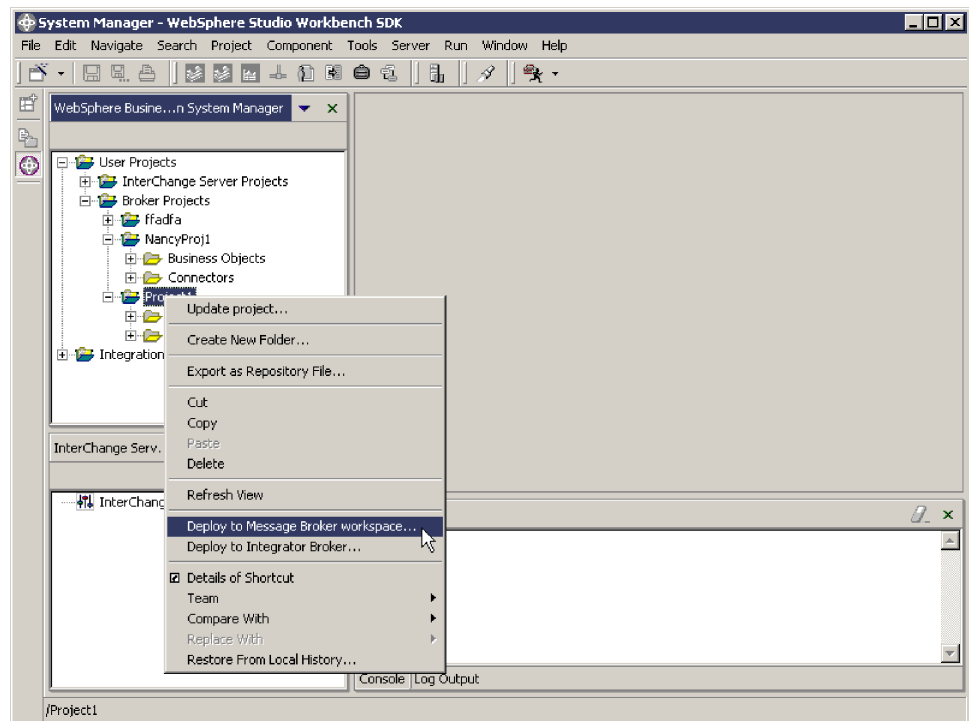
配置のプロセスのこの後の手順は、統合ブローカーとしてどのメッセージ・ブローカーを使用するかによって異なります。次のステップで、新規プロジェクトの名前を右マウス・ボタンでクリックし、次に示す 2 種類の配置のいずれかを選択します。

使用するメッセージ・ブローカー	選択する項目	参照先
WebSphere Business Integration Message Broker	「メッセージ・ブローカー・ワークスペースに配置 (Deploy to message broker workspace)」	99 ページの『メッセージ・ブローカー・ワークスペースへの配置』
WebSphere MQ Integrator	「インテグレーター・ブローカーに配置 (Deploy to integrator broker)」	102 ページの『インテグレーター・ブローカーへの配置』
WebSphere MQ Integrator Broker		

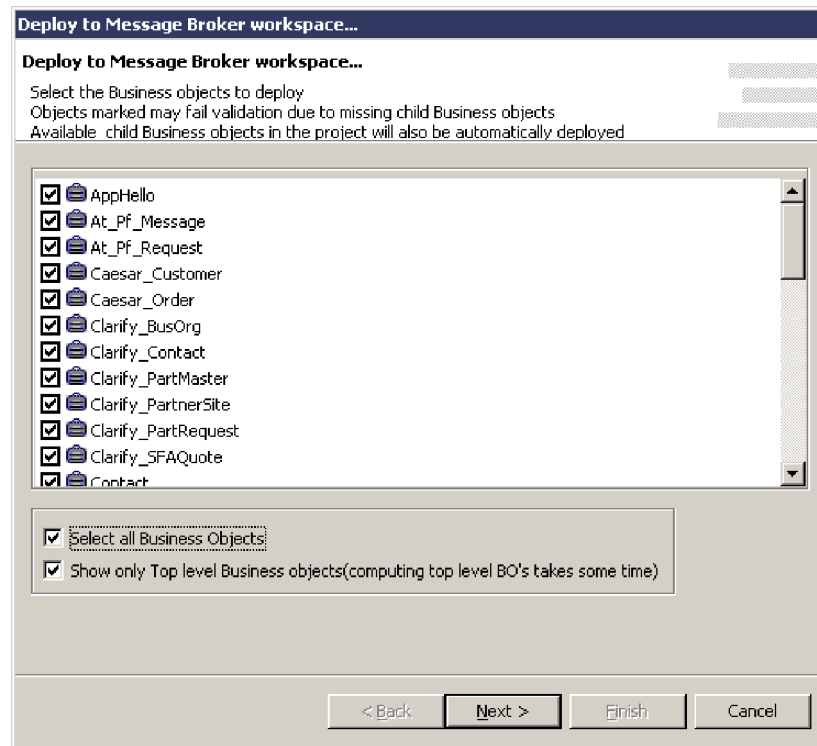
メッセージ・ブローカー・ワークスペースへの配置

メッセージ・ブローカー・ワークスペースにプロジェクトを配置するには (WebSphere Business Integration Message Broker を統合ブローカーとして使用する場合)、System Manager から次の手順で行います。

1. 「ユーザー・プロジェクト」パネルで、新規プロジェクトの名前を右マウス・ボタンでクリックしてから、以下に図示するように、「メッセージ・ブローカー・ワークスペースへの配置 (Deploy to message broker workspace)」を選択します。



以下のウィンドウが現れ、使用可能なビジネス・オブジェクトを表示します。



2. リストから必要なビジネス・オブジェクトを選択します (あるいは、「ビジネス・オブジェクトをすべて選択 (Select all business objects)」または「トップレベルのビジネス・オブジェクトのみを表示 (Show only Top level Business objects)」を選択します)。

3. 「次へ」をクリックします。以下のウィンドウが表示されます。

Select the parameters

Select the destination project name and parameters

Select the Destination Project

Enter the Message Set project name: NancyProj1

Enter a base Message set project(optional):

Enter the base Message set(optional):

Select the parameters

Replace existing project with the same name

Namespace aware(Only applicable to new projects)

Deploy in verbose mode

long Please select the xml namespace format (Connectors in the source project will be configured accordingly).

< Back Next > Finish Cancel

4. メッセージ・セット・プロジェクトの名前 (必須) およびその他のパラメーター (オプション) を入力します。以下、パラメーターについて詳細に説明します。

メッセージ・セット・プロジェクト名 (Message set project name) このプロジェクトの名前を指定する任意のテキスト・ストリングです。この名前は、配置の完了後、「System Manager」ウィンドウに表示されます。

ベース・メッセージ・セット・プロジェクト (Base message set project) とベース・メッセージ・セット (base message set) 既存のベース・メッセージ・セット・プロジェクトとベース・メッセージ・セットの両方の名前を入力した場合は、ベース・メッセージ・セットの内容が、新規の配置対象プロジェクトのベースとして使用されます。ベース・プロジェクト内のメッセージ・セット定義がすべて、新規の配置対象プロジェクトにコピーされます。さらに、配置対象プロジェクトには、今回の配置時に、前のウィンドウから選択したビジネス・オブジェクトもすべて含まれます。既存のメッセージ・セット・プロジェクトとベース・メッセージ・セットの名前は、Message Broker Tooling の「ブローカー・アプリケーション開発 (Broker Application Development)」パースペクティブから表示することができます。

注: ベース・メッセージ・セット・プロジェクトとベース・メッセージ・セットの両方を入力する必要があります。いずれか一方のパラメーターしか指定しない場合は、パラメーターは無視され、今回の配置時に選択したビジネス・オブジェクトのみが、プロジェクトに含まれることになります。

既存のプロジェクトを同名のプロジェクトで置き換える (Replace existing project with the same name) このオプションは、「メッセージ・セット・プロジェクト名の入力 (Enter message set project name)」フィールドに入力し

た名前が、既存のプロジェクトと同じである場合に適用されます。同名の場合に、このオプションが選択されていると、今回の配置時に行った選択内容によって、既存のプロジェクトの内容が置き換えられます (同一なのは名前のみになります)。このオプションにチェックマークを付けず、指定したプロジェクト名が既存のプロジェクトと一致する場合、既存のプロジェクトは、すでに含まれているビジネス・オブジェクトおよび今回の配置時に選択したビジネス・オブジェクトを保存します。同名の既存のビジネス・オブジェクトはすべて上書きされます。このオプションにチェックマークを付け、プロジェクト名が既存のプロジェクトと一致しない場合は、このオプションは無視され、新規のプロジェクトが作成されます。

ネーム・スペース認識 (Namespace aware) と xml ネーム・スペース形式 (xml namespace format) デフォルトでは、XML ネーム・スペースの形式は short (短形式) に設定されます。「ネーム・スペース認識 (Namespace aware)」と「xml ネーム・スペース形式 (xml namespace format)」の正しい組み合わせを選択することが重要であり、その選択を誤ると、配置は失敗する可能性があります。このパラメーターに long (長形式) を選択するか、short (短形式) を選択するかについては、105 ページの『XML ネーム・スペースの長さの選択』を参照してください。

冗長モードでの配置 (Deploy in verbose mode) このオプションを選択すると、配置時に配置処理の詳細が表示されるかログに記録されます。

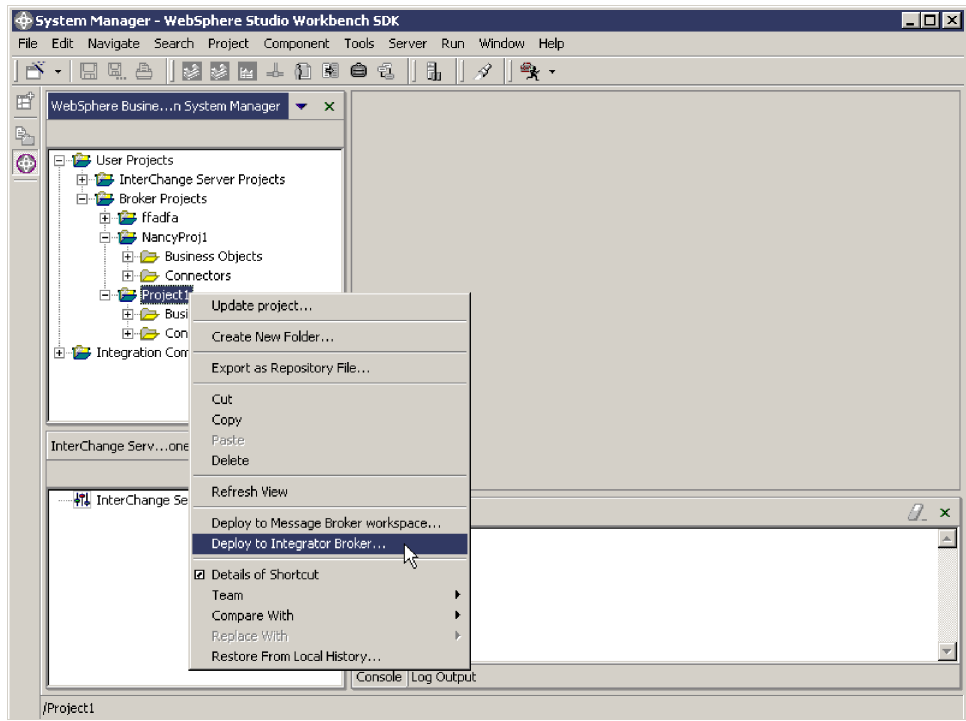
5. 「完了」をクリックします。プロジェクトの配置が正常に終了すると、配置の正常終了を示すメッセージがウィンドウに表示されます。さらに、配置の結果が、インポーター・ディレクトリー (デフォルトのディレクトリーは eclipse) に存在するデフォルトのログ・ファイル `mqsicreatemsgdefs.report.txt` に記録されます。System Manager のコンソール・パネルも参照してください。どのビジネス・オブジェクトが正常に配置されたかが表示されます。

注: 配置が正常に終了して「完了」をクリックするときには、Message Broker Toolkit はクローズしている必要があります。その時、Toolkit がオープンされていると、すべてまたは一部のメッセージ・セットとビジネス・オブジェクトの配置が失敗したことを示すメッセージが表示されます。Toolkit がクローズしていることを確認して、配置処理の最初に戻り再配置を行う必要があります。

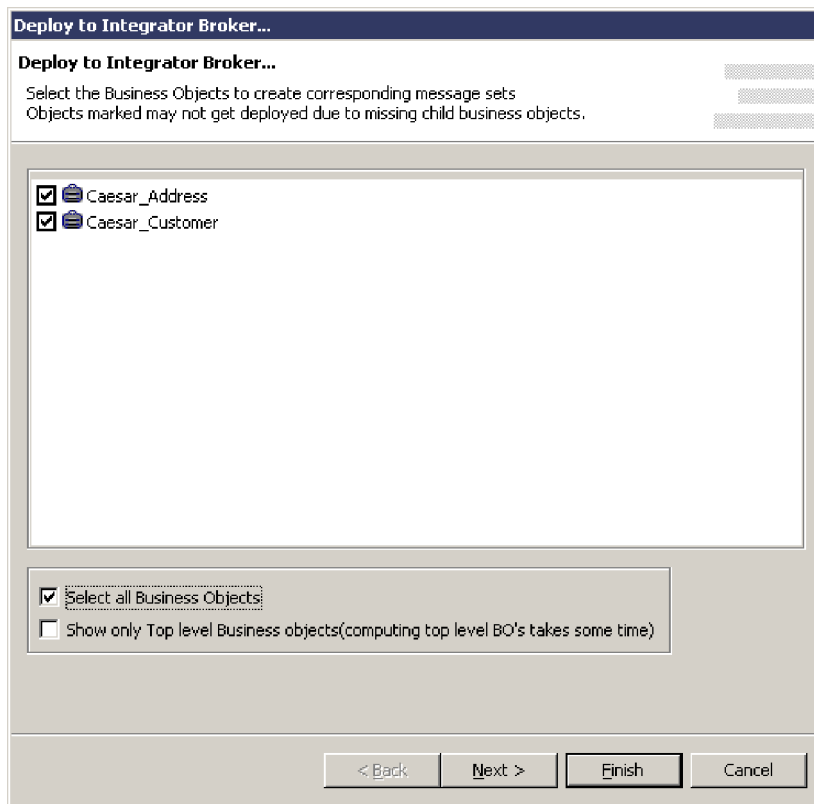
インテグレーター・ブローカーへの配置

インテグレーター・ブローカーにプロジェクトを配置するには (WebSphere MQ Integrator または WebSphere MQ Integrator Broker を統合ブローカーとして使用する場合)、System Manager から次の手順で行います。

1. 「ユーザー・プロジェクト」パネルで、新規プロジェクトの名前を右マウス・ボタンでクリックした後、以下の図に示すように「インテグレーター・ブローカーに配置 (Deploy to integrator broker)」を選択します。



以下のウィンドウが現れ、使用可能なビジネス・オブジェクトを表示します。



- リストから必要なビジネス・オブジェクトを選択します (あるいは、「ビジネス・オブジェクトをすべて選択 (Select all business objects)」または「トップレベルのビジネス・オブジェクトのみを表示 (Show only Top level Business objects)」を選択します)。
- 「次へ」をクリックします。「パラメーターの選択 (Parameter Selection)」ウィンドウが表示されます。このウィンドウでは、WebSphere MQ Integrator 値を変更することやコネクタに対する XML ネーム・スペース形式を選択することができます。例:

Property	Value	Description
MRM_DB	MRM_DB1	Message Repository Manager DB Name
DB_User	MRMDBAdmin	User Name with permissions to modify DB
DB_PW	*****	Password
xsiPrefix	xsi	Prefix used for XML Schema instance namespace
DefNameSpace		Specify a default namespace
NoNameSpaceLoc		NoNameSpaceLoc
XMLTabName	CwXML	XML Wire format
ReportOnly	N	Pseudo-import(no permanent changes to Data...
Verbose	N	Verbose(for troubleshooting)
Trace	N	Trace(for troubleshooting)
DateStringOnly	N	Model parameters using MRM STRING data type?
InclTimeFrac	N	Want fractions of a second in MRM DATETIME?
ShowSchema	N	Dump the Schema tags(troubleshooting)?

Select the connector(s) xml namespace format:

Connectors in the project are configured accordingly

< Back Next > Finish Cancel

- パスワード (DB_PW プロパティ) およびその他の値を入力し、デフォルト値からの変更を行います。(このウィンドウの前の使用時から入力されていた値が保管されており、デフォルト値として表示されます。) デフォルトでは、XML ネーム・スペースの形式は short (短形式) に設定されることに注意してください。「ネーム・スペース認識 (Namespace aware)」と「XML ネーム・スペース形式 (XML namespace format)」の正しい組み合わせを選択することが重要であり、その選択を誤ると、配置は失敗する可能性があります。このパラメーターに long (長形式) を選択するか、short (短形式) を選択するかについては、105 ページの『XML ネーム・スペースの長さの選択』を参照してください。
- 「完了」をクリックします。配置が成功した場合、ウィンドウが現れ、配置の正常終了を示すメッセージを表示します。System Manager のコンソール・パネルも参照してください。どのビジネス・オブジェクトが正常に配置されたかが表示されます。

XML ネーム・スペースの長さの選択

メッセージ・ブローカーにプロジェクトを配置する場合、long または short の XMLNamespace 形式を選択することができます。デフォルトは short です。XMLNamespaceFormat は、アダプターの標準構成接続プロパティです。System Manager 内で行った選択は、アダプターの構成ファイル内にすでに存在している可能性のある値をオーバーライドします。WebSphere Business Integration Message Broker に配置する場合は、「パラメーターを選択 (Select the parameters)」ウィンドウで、プロジェクトをネーム・スペースを認識するプロジェクトとして指定することができます。ネーム・スペース認識プロジェクトの場合、short (短形式) または long (長形式) のどちらかの形式を指定できます。ネーム・スペースを認識しないプロジェクトの場合は、long を指定する必要があります。ネーム・スペースを認識しないプロジェクトに short を選択すると、配置は失敗します。

注: IBM では、業界標準との継続的な互換性を保つため、新規プロジェクトはネーム・スペース認識モードに設定することをお勧めしています。IBM は、パフォーマンス向上のため、可能であれば XMLNamespace の形式を short に設定することをお勧めしています。

WebSphere MQ Integrator または WebSphere MQ Integrator Broker に配置する場合は、long または short を設定することができます。ただし、short を選択する場合は、コネクターの RFH2messagedomain プロパティを **xml** に設定する必要があります。RFH2messagedomain を **mrm** に設定した場合、ネーム・スペースの形式が long の XML メッセージしか処理できません。

注: WebSphere MQ Integrator と WebSphere MQ Integrator Broker は、ネーム・スペースをサポートしていません。したがって、それらのブローカーに配置されるプロジェクトに対して、ネーム・スペース認識を指定することはできません。

表 12 に、選択可能な XMLNamespace の形式を示します。

表 12. 選択可能な XMLNamespace の形式

メッセージ・ブローカー	ネーム・スペースを認識するかどうか	
	認識する	認識しない
WebSphere Business Integration Message Broker	long または short	long
WebSphere MQ Integrator または WebSphere MQ Integrator Broker	不適用	long または short (short の場合、RFH2messagedomain を xml に設定する)

アプリケーションでコネクターを使用可能にする

コネクターのアプリケーション固有のコンポーネントが、アプリケーションとの間で業務データを送受信できるようにするには、アプリケーションのコネクターに専用の、ユーザー・アカウントを設定する必要があります。コネクターの構成ファイルを作成する際には、このアカウントのユーザー ID とパスワードを指定する必要があります。

ほとんどのコネクターに関して、アプリケーションはイベント検出機構を実装するように構成されている必要があります。構成が完了したアプリケーションは、エンティティの変更を検出してイベント・レコードをイベント・ストアに書き込むことができます。その後は、コネクターによって取り出され、処理されます。トリガーの作成は、WebSphere MQ メッセージ・フローで処理される、ビジネス・オブジェクト用とオペレーション用に限定します。そうしないと、処理にともない除去の対象となることのないメッセージで、メッセージ・キューが満杯の状態になります。

アプリケーションがコネクターと連動できるようにするために必要な全作業に関する詳細については、構成するコネクターのアダプター・ユーザース・ガイドを参照してください。

コネクターの構成

Connector Configurator ツールには、コネクターを構成するためのグラフィカル・ユーザー・インターフェース (GUI) が用意されています。コネクターの構成プロパティーの値を指定する作業が完了すると、Connector Configurator によりコネクターの構成ファイルが生成され、コネクターのローカル・リポジトリに構成ファイルが配置されます。

重要: ビジネス・インテグレーション・アダプターが UNIX 上で動作する場合、Windows で Connector Configurator を使用して構成ファイルを作成し、そのファイルを自分の UNIX マシンにコピーする必要があります。構成ファイルを作成する場合、プロパティーを設定する際に UNIX パスとファイル名の命名規則を守ってください。

IBM WebSphere Business Integration Adapter のインストーラーを実行した場合、コネクター用のコネクター定義ファイルが、`ProductDir¥connectors¥repository¥ConnName` にロードされます。コネクター定義ファイルにより、一部の構成ファイル・プロパティーの初期値が設定されています。次のステップでは、Connector Configurator を使用して、コネクター用の構成ファイルを作成します。コネクター定義ファイルに構成ファイルの開始値が一部用意されていますが、構成ファイルにはコネクターの標準およびアプリケーション固有のプロパティーがすべて組み込まれ、サポートされるビジネス・オブジェクトが指定されています。これらのプロパティーの詳細は、125 ページの『付録 C. コネクターの標準構成プロパティー』に説明されています。

新規のコネクター (事前に構築された IBM のコネクターではないもの) を手動でインストールした場合は、Connector Configurator を使用して、そのコネクター用に新規の構成ファイルを作成する必要があります。

ヒント

構成するコネクターのアダプター・ユーザーズ・ガイドと、これ以降の説明を参照してください。ユーザーズ・ガイドには、次の情報が記載されています。

- Connector Configurator の詳細な使用手順。
- コネクターの標準プロパティ、アプリケーション固有のプロパティ、ロギングおよびトレース・オプションに関する、サポートされた設定についての詳細情報。

以降の各セクションでは、WebSphere Message Broker と動作するすべてのコネクターに適用される Connector Configurator の設定について説明します。

コネクターのローカル・リポジトリのロケーションの指定

標準プロパティ (RepositoryDirectory) を使用するローカル・リポジトリ用に、ロケーションを指定することができます。デフォルトの位置は *ProductDir* *¥repository* です。

コネクターが使用するキューの指定

90 ページの『WebSphere MQ キューの構成』では、メッセージ・ブローカーと通信するためにコネクターが使用するキュー・セットを定義しました。Connector Configurator で、「標準のプロパティ」タブをクリックして、これらのキューを、次の標準プロパティを使用するコネクターに割り当てます。

- DeliveryQueue
- RequestQueue
- ResponseQueue (このキューは WebSphere InterChange Server でのみ使用されるが、互換性のために定義する必要がある)
- FaultQueue
- SynchronousRequestQueue (コネクターが同期実行方式を使用する場合にのみ必要)
- SynchronousResponseQueue (コネクターが同期実行方式を使用する場合にのみ必要)
- AdminInQueue
- AdminOutQueue

キュー・マネージャーによる接続モードの設定

デフォルトの接続モードはバインディング・モードです。クライアント・モードは、次のように指定してください。

1. Connector Configurator で、「標準のプロパティ」タブをクリックします。
2. 標準プロパティ *jms.MessageBrokerName* に値 *QueueMgrName:[Channel]:[HostName]:[PortNumber]* を割り当てます。ここで、変数はそれぞれ以下のものを表します。

QueueMgrName

キュー・マネージャーの名前。

Channel

クライアントが使用するチャンネル。

HostName

キュー・マネージャーが常駐するマシンの名前。

PortNumber

キュー・マネージャーが `listen` に使用するポート番号。

これには、次のものがあります。

```
jms.MessageBrokerName = WMQIB.Queue.Manager:CHANNEL1:RemoteMachine:1456
```

同期実行のための構成プロパティの設定

コネクタが同期実行方式を使用する場合は、「標準のプロパティ」をタブをクリックして、次のプロパティを設定します。

- `SynchronousRequestQueue` = `SynchronousRequestQueue`
- `SynchronousResponseQueue` = `SynchronousResponseQueue`
- `SynchronousRequestTimeout` = 10000

ロギングおよびトレース・オプションの構成

ロギングは、システム・メッセージ、コンポーネント状態変更、障害、およびトレース情報の通信に使用されます。次のファイルが生成されます。

表 13. コネクタのロギングおよびトレース・ファイル

デフォルトのファイル名およびパス	説明
<p>一時的ログ・ファイル:</p> <p><i>ProductDir</i>¥<i>broker_name_connector_name_tmp.log</i>.</p>	<p>始動時に、コネクタにより一時的ログ・ファイルが生成されます。このファイルには、コネクタ・フレームワークに渡されるコネクタ・プロパティおよびビジネス・オブジェクト定義など、始動時にログとして記録されるメッセージがすべて格納されます。また、ファイルは製品がインストールされているディレクトリーに書き込まれます。</p>
<p>コネクタ・ログ・ファイル:</p> <p>UNIX: メッセージはコネクタにより、デフォルトでは <code>STDOUT</code> にロギングされ、その後 <i>ProductDir/logs/connector_manager_ConnName.log</i> に転送されます。</p> <p>WINDOWS: メッセージはコネクタによりデフォルトでは <code>STDOUT</code> にロギングされますが、<i>ProductDir</i> ディレクトリー内のローカル宛先ログ・ファイルに送信するように構成することもできます。</p>	<p>コネクタのログ・ファイルは、コネクタから発行されたメッセージを保管するために使用します。また、WebSphere MQ 通信エラーに関するメッセージも格納されます。</p>
<p>コネクタ・メッセージ・ファイル:</p> <p><i>ProductDir</i> ¥<i>connectors</i>¥<i>messages</i>¥<i>ConnName_LocaleName .txt</i></p>	<p>このファイルには、コネクタから発行される各メッセージのテキスト全文が格納されます。このファイルを使用して、ログ・ファイルに表示されるメッセージ ID の本文を検索することができます。コネクタ構成ファイルで指定したロケールがサポートされていない場合は、ファイル <i>ConnName.txt</i> が使用されます。</p>
<p>トレース・ファイル:</p> <p>UNIX と Windows の双方とも、デフォルトでは <code>STDOUT</code> に設定されています。</p>	<p>選択したトレース・レベルで指定されたトレース・メッセージが格納されます。</p>

ロギング・システムは常にアクティブで、コネクタの正確なモニター情報が出力されます。

障害発生時の対処では、トレースをオンにします。トレース・メッセージにより、ビジネス・インテグレーション・システムのコンポーネントでの動作状況をモニターすることができます。トレース・レベルにより、トレース・ファイルに書き込む詳細情報の量が定義されます。トレース・レベルが高いほど、詳細な情報が書き込まれます。トレースは次の点で、ロギングとは異なります。

- ロギングは常時実行されますが、トレースは必要に応じて、オン/オフ状態にすることができます。
- トレースでは、コンポーネントの状態とコンポーネントの動作について、ロギングよりも詳細な情報が記録されます。
- ロギングとトレースの設定値は、リブート後も有効です。

トレースは、デフォルトではオフの状態になっています。これは通常必要とされる以上に詳細なメッセージを出力するからです。

LogViewer を使用したロギングおよびトレース・メッセージの表示については、81 ページの『Log Viewer を使用してコネクタ・メッセージを表示する方法』を参照してください。

コネクタ・ロギングの構成

コネクタ・ロギング・オプションを構成するには、「トレース/ログ・ファイル」タブをクリックして、下記のとおり指定します。

1. ログ・メッセージを STDOUT に転送する場合は、「コンソールに (STDOUT)」チェック・ボックスをクリックする。
2. ログ・メッセージをファイルに転送する場合は、「ファイルに」チェック・ボックスをクリックし、使用するログ・ファイルの絶対パス名を指定する。「コンソールに」と「ファイルに」の両オプションを指定して、ログ・メッセージをコンソールおよびファイルに転送することもできます。
3. ログ・ファイルの使用を指定した場合は、次のオプションも指定する。
 - a. ログ・ファイルのサイズを制限するには、「ログ・ファイルのサイズ」に数値と測定単位を設定する。
 - b. ログ・ファイルのサイズを制限なしとするには、「無制限」チェック・ボックスをクリックする。
 - c. ログ・ファイルの最大サイズを設定し、ファイルのアーカイブを使用する場合は、「アーカイブの数」に保存するアーカイブ・ファイル数を設定する。

ログ・ファイル管理の詳細については、78 ページの『ログ・ファイルおよびトレース・ファイルの管理』を参照してください。

コネクタ・トレースの構成

コネクタのトレース・オプションを構成する作業は、次の手順で行います。

1. 「トレース/ログ・ファイル」タブをクリックする。
2. トレース・メッセージを STDOUT に転送するため、「コンソールに (STDOUT)」にチェックマークを付ける。
3. トレース・メッセージをファイルに転送するため、「ファイルに」にチェックマークを付け、使用するトレース・ファイルの絶対パス名を指定する。「コンソールに」と「ファイルに」の両オプションを指定して、トレース・メッセージをコンソールおよびファイルに転送することもできます。
4. トレース・ファイルの使用を指定した場合、次のオプションも指定する。
 - a. トレース・ファイルのサイズを制限するには、「トレース・ファイルのサイズ」に数値と測定単位を設定する。
 - b. トレース・ファイルのサイズを制限なしとするには、「無制限」にチェックマークを付ける。
 - c. トレース・ファイルの最大サイズを設定し、ファイルのアーカイブを使用する場合は、「アーカイブの数」に保存するアーカイブ・ファイル数を設定する。
5. トレース・レベルを設定するには、次の作業を実行します。
 - a. 「標準のプロパティ」タブをクリックする。
 - b. **AgentTraceLevel** プロパティに、111 ページの表 14 にリストされている値のうちの 1 つを設定する。

構成するコネクタの場合に個々のトレース・レベルで生成される情報の詳細については、そのコネクタのアダプター・ユーザズ・ガイドを参照してください。コネクタのトレースは、次のいずれかのレベルに設定します。

表 14. コネクタ・トレース・レベル

トレース・レベル	説明
1	トレースの初期化、およびビジネス・オブジェクトの送受信。
2	レベル 1 のメッセージ印刷。さらに、同じタイプのイベントについて、レベル 1 よりも詳細な情報を出力。
3	レベル 1 および 2 のメッセージ印刷。さらに、コネクタ・エージェントとメッセージング・ドライバー間におけるメッセージ交換のトレース。
4	レベル 1 から 3 のメッセージ印刷。さらに、コネクタの内部レベル間におけるビジネス・オブジェクトの引き渡し処理のトレース。
5	レベル 1 から 4 のメッセージ印刷。さらに、コネクタの内部レベル間における管理メッセージの引き渡し処理のトレース。

新規または変更済みのトレース・レベルは、コネクタを再始動すると有効になります。

トレース・ファイルをアーカイブする作業の詳細については、78 ページの『ログ・ファイルおよびトレース・ファイルの管理』を参照してください。

コネクタの始動ファイル、ショートカット、および環境変数の構成

コネクタの始動手順も、必要なセットアップ・タスクも、コネクタが動作するプラットフォームにより、以下のように異なります。

Windows の場合

Windows に WebSphere Business Integration Adapters をインストールすると、インストールされた各コネクタについて WebSphere Business Integration Adapters プログラム・メニュー（「スタート」>「プログラム」>「IBM WebSphere Business Integration Adapters」>「アダプター」>「コネクタ」）にショートカットが作成されます。下記のをカスタマイズするには、143 ページの『付録 D. コネクタ始動オプション』にリストされている始動オプションを使用する必要があります。

- コネクタのショートカット・プロパティ
- コネクタの始動ファイル `start_connName.bat` (Java で作成されたコネクタの場合)。
- コネクタの始動ファイル `start_connector.bat` (C++ で作成されたコネクタの場合)。

UNIX の場合

UNIX 環境では、`connector_manager_connName` スクリプトを実行してコネクタを始動します。このスクリプトは、汎用コネクタ・マネージャー・スクリプト (`ProductDir/bin/connector_manager`) のラッパーです。このラッパーには、以下の情報が組み込まれています。

- 始動または停止するコネクタ名。
- 汎用コネクタ・マネージャに対応するコマンド行オプション。これには、次のものがあります。
 - SAP コネクタには `-t` コマンド行オプションが必要です。したがって、その始動スクリプトには `-t` オプションが組み込まれます。
 - UNIX コネクタはすべて `-b` オプションを使用して動作します。したがって、コネクタ始動スクリプトすべてに `-b` オプションが組み込まれます。コネクタをフォアグラウンドで実行するには、汎用コネクタ・マネージャ・スクリプト (`connector_manager`) から `-b` オプションを除去します。
- 構成ファイルの名前。

カスタム・アダプターを作成した場合や、アダプターを ESD (Electronic Software Delivery) を使用してインストールした場合は、コネクタを初めて始動する際に、以下の手順を行う必要があります。

1. コネクタ・スクリプト生成ツールを実行し、`connector_manager_connName` スクリプトを、コネクタの構成ファイルの名前で更新する。このツールの実行の詳細については、181 ページの『付録 F. コネクタ・スクリプト生成ツールの使用』を参照してください。

あるいは、`ProductDir/bin` ディレクトリーにナビゲートして `connector_manager_connName` ファイルを編集し、コネクタの構成ファイル名を指定する。ファイル内で、`AGENTCONFIG_FILE` 変数を検索し、その変数に次のように構成ファイルの絶対パス名を設定する。

```
AGENTCONFIG_FILE=ConfigFile
```

2. 必要であれば、`PATH` 環境変数を更新して `ProductDir/bin` ディレクトリーを組み合わせる。
3. `CWSharedEnv.sh` ファイルは、自分のアカウントのシェル始動スクリプト (`.cshrc` など) が、ソースになっていることを確認する。

始動スクリプトのカスタマイズ: 汎用コネクタ・マネージャ・スクリプトは、特定のコネクタを管理する実スクリプトである、該当の `start_connector.sh` スクリプトを呼び出します。各 WebSphere Business Integration Adapter には、`start_connector.sh` スクリプトが含まれています。`start_connector.sh` スクリプトを変更して、143 ページの『付録 D. コネクタ始動オプション』で紹介されている各サポート始動オプションを、任意に組み合わせることができます。

注: コネクタ用の始動ファイルの作成については、「コネクタ開発ガイド (C++ 用)」または「コネクタ開発ガイド (Java 用)」を参照してください。

メッセージ・フローの定義

WebSphere Business Integration adapter は、その統合ブローカーとしてメッセージ・ブローカーを使用する場合、WebSphere MQ メッセージ・フローを使用して、ビジネス・アプリケーションが相互に送信するデータや要求を表すビジネス・オブジェクト・メッセージを処理し、一定経路で送信します。各キューごとに定義された単一のメッセージ・フローが、そのキューにあるメッセージをすべて処理しています。MQ Integrator Broker Control Center (WebSphere MQ Integrator Broker または WebSphere MQ Integrator の場合) または Message Brokers Toolkit (WebSphere

Business Integration Message Broker の場合) を使用すると、メッセージ処理プリミティブからメッセージ・フローを構築し、メッセージ・ヘッダーまたはメッセージ内容のいずれかに基づいて処理決定を行うことが可能になります。すなわち、メッセージ・フローにより、処理が予想される各メッセージ・タイプごとに、異なる処理ステップを指定することができるのです。

各ビジネス・メッセージ (コネクタ・フレームワークによる統合ブローカーへの送信、または統合ブローカーからコネクタ・フレームワークへの送信) には、メッセージが表すビジネス・オブジェクトを識別する、メッセージ・ヘッダーとメッセージ・ディスクリプター情報が組み込まれています。

ビジネス・インテグレーション・システムのメッセージ・フローを定義する前に、各キューから処理されるビジネス・オブジェクトを識別する必要があります。また、メッセージ・フローが処理するメッセージ・タイプごとの、メッセージ・ディスクリプター、メッセージ・ヘッダー、およびメッセージ・フォーマットについても、習熟しておく必要があります。詳細については、44 ページの『メッセージ交換のインターフェース』と 115 ページの『付録 A. WebSphere MQ のメッセージ・フォーマット』を参照してください。

メッセージ・フロー作成の詳細については、ご使用のメッセージ・ブローカーの資料を参照してください。

トランザクション管理

WebSphere Message Broker では、分散トランザクション・サポートを用意しています。メッセージ・フローは、全世界に連係されたトランザクション内で実行でき、WebSphere MQ キューとの間で送受信されたメッセージは、メッセージ処理中に実行されたすべてのデータベース更新と、連係を取ることができます。メッセージ・フローは WebSphere MQ 機能を使用して Transaction Manager として動作し、WebSphere MQ 作業単位内でのデータベース更新を調整することができます。ビジネス・インテグレーション・システムへのトランザクション管理の実装についての詳細は、メッセージ・ブローカーの管理に関する資料を参照してください。

Visual Test Connector を使用したインターフェースの検証

Visual Test Connector は、コネクタの活動をシミュレートすることで、実際にコネクタを実行するという手間をかけずに、統合コンポーネントをテストできます。Visual Test Connector を使用して、ソース・コネクタおよび宛先コネクタが正しく構成されているかどうか、およびサポートされているビジネス・オブジェクト定義が正しく指定されているかどうかを検証できます。Visual Test Connector の使用方法についての詳細は、183 ページの『付録 G. Visual Test Connector の使用』を参照してください。

付録 A. WebSphere MQ のメッセージ・フォーマット

各メッセージには、メッセージ・ディスクリプター (MQMD)、メッセージ・ヘッダー (MQRFH2)、およびメッセージ本文という 3 つのコンポーネントが含まれています。

メッセージ・ディスクリプター

WebSphere MQ メッセージ・ディスクリプター (MQMD) には、メッセージ ID およびメッセージの処理に必要な情報が入っています。

メッセージ・ヘッダー

MQRFH2 メッセージ・ヘッダーは、メッセージ内容に関連した JMS 固有のデータを搬送します。また、直接 JMS と関連のない追加情報も搬送できます。メッセージ・ヘッダーには次のフォルダーが含まれています。

- <mcld> フォルダーには、メッセージの「形状」または「フォーマット」を記述するプロパティーが含まれています。
- <jms> フォルダーは、JMS ヘッダー・フィールド と、MQMD では完全に表現できない JMS プロパティーを移送するために使用します。JMS を使ってインプリメントされ、コネクタ・フレームワークによって送信されるメッセージでは、常にこのフォルダーが関係しています。ただし、ビジネス・インテグレーション・システムでは、このフォルダーは関連性がないため、メッセージをコネクタ・フレームワークに送信する際に WebSphere Message Broker によって除外されます。
- <usr> フォルダーは、メッセージに関連付けられたアプリケーションで定義済みのプロパティーを移送するために使用します。アプリケーションが何らかのアプリケーションで定義済みのプロパティーを設定した場合にのみ、このフォルダーが存在します。ビジネス・インテグレーション・システムでは、このフォルダーは、応答メッセージのリターン状況情報を送信するために使用します。この後の各表では、このフォルダーを必要とするメッセージのタイプについて紹介しています。

メッセージ本文

メッセージ本文は、そのメッセージ用に指定された XML スキーマの内容に従ってフォーマット設定されます。メッセージをフォーマット設定するために正しい XML スキーマをデータ・ハンドラーが検出および使用するには、次の 3 つの名前が一致しなければなりません。

- コネクタのリポジトリに格納されている XML スキーマの名前。
- WebSphere Message Broker のメッセージ・リポジトリにインポートされ、メッセージ・セット定義として保管された XML スキーマの名前。
- メッセージの MQRFH2 メッセージ・ヘッダーにある `messagetype` の内容。

コネクタ・フレームワークおよび WebSphere MQ Integrator によって交換される各種タイプのメッセージに使用するメッセージ・フォーマットおよび特定のプロパティの設定値について、次の各一覧表で紹介します。

表 15. コネクタ・フレームワークから WebSphere Message Broker へ送信されるイベント・デリバリー・メッセージのフォーマットおよびプロパティの設定値

MQMD	関連情報はありません。
RFH2 メッセージ・ヘッダー	<p><mcid> フォルダで、以下のフィールドには、メッセージ、そのフォーマット、およびメッセージの構文解析必要条件を識別する情報が含まれます。</p> <ul style="list-style-type: none"> • メッセージ・ドメインは、デフォルトでは mrn に設定されます。この設定で、統合ブローカーは MRM 管理メッセージ (メッセージ・リポジトリで完全モデルとなったメッセージ) 専用のパーサーを使用するように指定されます。 RFH2MessageDomain 標準コネクタ構成プロパティを変更することにより、この設定値を xml に変更できます。詳細については、『付録 C. コネクタの標準構成プロパティ』を参照してください。 • メッセージ・タイプは、メッセージが表現するトップレベルのビジネス・オブジェクトの名前を識別します。 • メッセージ・セットは、そのメッセージが関連付けられているメッセージ・セットを識別します。メッセージ・セットは、各タイプのビジネス・オブジェクトごとに、個別に作成されます。 • メッセージ・フォーマットは CwXML に設定されます。
メッセージ本文	XML スキーマに準拠した XML インスタンス文書と、RFH2 ヘッダーでメッセージ・タイプにより指定されたビジネス・オブジェクト用の、インポート済みメッセージ・ブローカーのメッセージ定義が含まれています。

表 16. コネクター・フレームワークから *WebSphere Message Broker* へ送信される要求メッセージのフォーマットおよびプロパティの設定値

<p>MQMD</p>	<p>応答先情報は、ReplyToQ および ReplyToQMgr という 2 つのフィールドに設定されています。両フィールドには、統合ブローカーがメッセージ応答メッセージを送信するのに必要な、キュー名およびキュー・マネージャー名が入っています。JMS メッセージでは、両フィールドで、要求メッセージの JMSReplyTo 宛先を指定します。MessageID (JMSMessageID) には固有値が含まれていますが、この固有値は応答メッセージ上の CorrelID プロパティ・フィールドにコピーされます。</p>
<p>RFH2 メッセージ・ヘッダー</p>	<p><mcld> フォルダーで、次のフィールドには、メッセージ、そのフォーマット、およびメッセージの構文解析必要条件を識別する情報が含まれます。</p> <ul style="list-style-type: none"> • メッセージ・ドメインは、デフォルトでは mrml に設定されます。この設定値は、統合ブローカーが MRM 管理メッセージ (メッセージ・リポジトリで完全にモデル化されたメッセージ) 専用のパーサーを使用することを指定します。 RHF2MessageDomain 標準コネクター構成プロパティを変更することにより、この設定値を xml に変更できます。詳細については、『付録 C. コネクターの標準構成プロパティ』を参照してください。 • メッセージ・タイプは、メッセージが表現するトップレベルのビジネス・オブジェクトの名前を識別します。 • メッセージ・セットは、そのメッセージが関連付けられているメッセージ・セットを識別します。メッセージ・セットは、各タイプのビジネス・オブジェクトごとに、個別に作成されます。 • メッセージ・フォーマットは CwXML に設定されます。 <p><jms> フォルダーで、Rto (JMSReplyTo) フィールドには、統合ブローカーがメッセージ応答メッセージを送信するのに必要な、キュー名およびキュー・マネージャー名をエンコードする URI が入っています。この URI を指定する方法については、「<i>WebSphere MQ: Java の使用</i>」を参照してください。MQMD の Rto 内および ReplyToQ/ReplyToQMgr 内の応答先情報は同一です。</p>
<p>メッセージ本文</p>	<p>XML スキーマに準拠した XML インスタンス文書と、RFH2 ヘッダー内のメッセージ・タイプにより指定されたビジネス・オブジェクト用のインポート済みメッセージ・ブローカーのメッセージ・セット定義が含まれます。</p>

表 17. WebSphere メッセージ・ブローカーからコネクタ・フレームワークへの応答メッセージのフォーマットおよびプロパティの設定値

MQMD	<p>CorrelID プロパティ・フィールドには、統合ブローカーが応答する要求のメッセージ ID が含まれています。JMS メッセージの場合、このフィールドは JMSCorrelationID を定義するために使用します。</p>
RFH2 メッセージ・ヘッダー	<p><mcld> フォルダーで、次のフィールドには、メッセージ、そのフォーマット、およびメッセージの構文解析必要条件を識別する情報が含まれます。</p> <ul style="list-style-type: none"> • メッセージ・ドメインは、デフォルトでは mrm に設定されます。この設定で、統合ブローカーは MRM 管理メッセージ (メッセージ・リポジトリで完全モデルとなったメッセージ) 専用のパーサーを使用するように指定されます。 RHF2MessageDomain 標準コネクタ構成プロパティを変更することにより、この設定値を xml に変更できます。詳細については、『付録 C. コネクタの標準構成プロパティ』を参照してください。 • メッセージ・タイプは、メッセージが表現するトップレベルのビジネス・オブジェクトの名前を識別します。 • メッセージ・セットは、そのメッセージが関連付けられているメッセージ・セットを識別します。メッセージ・セットは、各タイプのビジネス・オブジェクトごとに、個別に作成されます。 • メッセージ・フォーマットは CwXML に設定されます。 <p><usr> フォルダーで、以下のフィールドには、リターン状況情報が含まれています。</p> <ul style="list-style-type: none"> • 状況フィールドには、リターン状況情報を含んだストリングが入っています。 <p>想定されるストリングの内容は、次のいずれかです。</p> <ul style="list-style-type: none"> -1: 要求したオペレーションは失敗しました。 0: 要求したオペレーションは正常に実行されました。 1: 要求したオペレーションは正常に実行されました。アプリケーションから変更済みビジネス・オブジェクトが戻されました。 <ul style="list-style-type: none"> • 説明フィールド - 状況が -1 に設定されている場合、このフィールドには統合ブローカーが送信したメッセージを含んだ、拡張エラー・ストリングが入っています。
メッセージ本文	<p>XML スキーマに準拠した XML インスタンス文書と、RFH2 ヘッダー内のメッセージ・タイプにより指定されたビジネス・オブジェクト用のインポート済みメッセージ・ブローカーのメッセージ・セット定義が含まれます。</p>

表 18. WebSphere Message Broker からコネクタ・フレームワークへ送信される要求メッセージのフォーマットおよびプロパティの設定値

<p>MQMD</p>	<p>応答先情報は、ReplyToQ および ReplyToQMgr という 2 つのフィールドに設定されています。両フィールドには、統合ブローカーがメッセージ応答メッセージを送信するのに必要な、キュー名およびキュー・マネージャー名が入っています。JMS メッセージでは、両フィールドで、要求メッセージの JMSReplyTo 宛先を指定します。ReplyToQ および ReplyToQMgr フィールドを空白のままにすると、コネクタ・フレームワークからの応答は返ってきません。応答が必要な場合には、メッセージでメッセージ・ヘッダーの Rto プロパティ・フィールドに応答先情報を指定することもできます。MessageID (JMSMessageID) には固有値が含まれていますが、この固有値は応答メッセージ上の CorrelID プロパティ・フィールドにコピーされます。</p>
<p>RFH2 メッセージ・ヘッダー</p>	<p><mcld> フォルダーで、次のフィールドには、メッセージ、そのフォーマット、およびメッセージの構文解析必要条件を識別する情報が含まれます。</p> <ul style="list-style-type: none"> • メッセージ・ドメインは mrm に設定されます。この設定値は、統合ブローカーが MRM 管理メッセージ (メッセージ・リポジトリで完全にモデル化されたメッセージ) 専用のパーサーを使用することを指定します。RHF2MessageDomain 標準コネクタ構成プロパティを変更することにより、この設定値を xml に変更できます。詳細については、『付録 C. コネクタの標準構成プロパティ』を参照してください。 • メッセージ・タイプは、メッセージが表現するトップレベルのビジネス・オブジェクトの名前を識別します。 • メッセージ・セットは、そのメッセージが関連付けられているメッセージ・セットを識別します。メッセージ・セットは、各タイプのビジネス・オブジェクトごとに、個別に作成されます。 • メッセージ・フォーマットは CwXML に設定されます。 <p><jms> フォルダーで、Rto (JMSReplyTo) プロパティ・フィールドには、コネクタ・フレームワークが応答メッセージを送信するのに必要なキュー名およびキュー・マネージャー名を、オプションとして含めることができます。</p>
<p>メッセージ本文</p>	<p>XML スキーマに準拠した XML インスタンス文書と、RFH2 ヘッダー内のメッセージ・タイプにより指定されたビジネス・オブジェクト用のインポート済みメッセージ・ブローカーのメッセージ・セット定義が含まれます。</p>

表 19. コネクター・フレームワークから *WebSphere Message Broker* へ送信される応答メッセージのフォーマットおよびプロパティの設定値

MQMD	CorrelID プロパティ・フィールドには、コネクター・フレームワークの応答先となる要求メッセージの ID が含まれています。
RFH2 メッセージ・ヘッダー	<p><mcid> フォルダーで、次のフィールドには、メッセージ、そのフォーマット、およびメッセージの構文解析必要条件を識別する情報が含まれます。</p> <ul style="list-style-type: none"> • メッセージ・ドメインは mrm に設定されます。この設定値は、統合ブローカーが MRM 管理メッセージ (メッセージ・リポジトリで完全にモデル化されたメッセージ) 専用のパーサーを使用することを指定します。RFH2MessageDomain 標準コネクター構成プロパティを変更することにより、この設定値を xml に変更できます。詳細については、『付録 C. コネクターの標準構成プロパティ』を参照してください。 • メッセージ・タイプは、メッセージが表現するトップレベルのビジネス・オブジェクトの名前を識別します。 • メッセージ・セットは、そのメッセージが関連付けられているメッセージ・セットを識別します。メッセージ・セットは、各タイプのビジネス・オブジェクトごとに、個別に作成されます。 • メッセージ・フォーマットは CwXML に設定されます。 <p><usr> フォルダーで、以下のフィールドには、リターン状況情報が含まれています。</p> <ul style="list-style-type: none"> • 状況フィールドには、リターン状況表示を含んだストリングが入っています。 <p>想定されるストリングの内容は、次のいずれかです。</p> <ul style="list-style-type: none"> -1: 要求したオペレーションは失敗しました。 0: 要求したオペレーションは正常に実行されました。 1: 要求したオペレーションは正常に実行されました。アプリケーションから変更済みビジネス・オブジェクトが戻されました。 <ul style="list-style-type: none"> • 説明プロパティ・フィールド - 状況が -1 に設定されている場合、本フィールドにはコネクター・フレームワークが送信したメッセージを含んだ、拡張エラー・ストリングが入っています。
メッセージ本文	XML スキーマに準拠した XML インスタンス文書と、RFH2 ヘッダー内のメッセージ・タイプにより指定されたビジネス・オブジェクト用のインポート済みメッセージ・ブローカーのメッセージ・セット定義が含まれます。

表 20. コネクター・フレームワークから *WebSphere Message Broker* へ送信される管理メッセージのフォーマットおよびプロパティの設定値

MQMD	関連情報はありません。
RFH2 メッセージ・ヘッダー	<p><mcid> フォルダーで、次のフィールドには、メッセージ、そのフォーマット、およびメッセージの構文解析必要条件を識別する情報が含まれます。</p> <ul style="list-style-type: none"> • メッセージ・ドメイン を xml に設定すると、メッセージは <i>WebSphere Message Broker</i> の汎用 XML パーサーによって構文解析されるように指定されます。
メッセージ本文	メッセージ本文の内容に関する詳細については、143 ページの『付録 D. コネクター始動オプション』を参照してください。

表 21. *WebSphere Message Broker* からコネクター・フレームワークへ送信される管理メッセージのフォーマットおよびプロパティの設定値

MQMD	管理メッセージが「コネクター停止」の場合、Format プロパティは次のとおりに設定されます。MQC.MQFMT_STRING および Expiry (JMSEExpiration) プロパティ・フィールドを 1 分とする。
メッセージ本文	メッセージ本文の内容に関する詳細については、143 ページの『付録 D. コネクター始動オプション』を参照してください。

付録 B. 管理メッセージの WebSphere MQ メッセージ本文形式

この付録では、コネクタ・フレームワークと WebSphere Message Broker の間で交換される管理メッセージの、メッセージ本文内容のサンプルを紹介します。

コネクタ・フレームワークから WebSphere Message Broker へのメッセージ

エラーが発生してコネクタをシャットダウンする必要がある場合、コネクタ・フレームワークから下記のメッセージ本文内容を含んだメッセージが統合ブローカーへ送信されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<CwConnectorCommand
  xmlns="http://www.ibm.com/websphere/crossworlds/2002/CwConnectorCommand"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/websphere/crossworlds/2002/
    CwConnectorCommand file:CwConnectorCommand.xsd">
  <Source>Connector1</Source>
  <Destination>IntegrationBroker</Destination>
  <Command>Shutdown</Command>
</CwConnectorCommand>
```

WebSphere Message Broker からコネクタ・フレームワークへのメッセージ

コネクタのシャットダウンを開始するため、統合ブローカーからは下記のメッセージ本文内容を含んだメッセージがコネクタ・フレームワークへ送信されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<CwConnectorCommand
  xmlns="http://www.ibm.com/websphere/crossworlds/2002/CwConnectorCommand"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/websphere/crossworlds/2002/
    CwConnectorCommand file:CwConnectorCommand.xsd">
  <Source>IntegrationBroker</Source>
  <Destination>Adapter</Destination>
  <Command>Shutdown</Command>
</CwConnectorCommand>
```

付録 C. コネクターの標準構成プロパティ

この付録では、WebSphere Business Integration Adapters のコネクター・コンポーネントの標準構成プロパティについて説明します。この付録の内容は、以下の統合ブローカーと連携するコネクターを対象としています。

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker (WebSphere Message Brokers (WMQI) と総称)
- WebSphere Application Server (WAS)

コネクターによっては、一部の標準プロパティが使用されないことがあります。Connector Configurator から統合ブローカーを選択すると、その統合ブローカーと連携するアダプターのために構成する必要のある標準プロパティのリストが表示されます。

コネクター固有のプロパティの詳細については、該当するアダプターのユーザーズ・ガイドを参照してください。

注: 本書では、ディレクトリー・パスに円記号 (¥) を使用します。UNIX システムを使用している場合は、円記号をスラッシュ (/) に置き換えてください。また、各オペレーティング・システムの規則に従ってください。

新規プロパティと削除されたプロパティ

以下の標準プロパティは、本リリースで追加されました。

新規プロパティ

- XMLNamespaceFormat

削除されたプロパティ

- RestartCount

標準コネクター・プロパティの構成

アダプター・コネクターには 2 つのタイプの構成プロパティがあります。

- 標準構成プロパティ
- コネクター固有の構成プロパティ

このセクションでは、標準構成プロパティについて説明します。コネクター固有の構成プロパティについては、該当するアダプターのユーザーズ・ガイドを参照してください。

Connector Configurator の使用

Connector Configurator からコネクタ・プロパティを構成します。Connector Configurator には、System Manager からアクセスします。Connector Configurator の使用法の詳細については、付録の『Connector Configurator』を参照してください。

注: Connector Configurator と System Manager は、Windows システム上でのみ動作します。コネクタを UNIX システム上で稼働している場合でも、これらのツールがインストールされた Windows マシンが必要です。UNIX 上で動作するコネクタのコネクタ・プロパティを設定する場合は、Windows マシン上で System Manager を起動し、UNIX の統合ブローカーに接続してから、コネクタ用の Connector Configurator を開く必要があります。

プロパティ値の設定と更新

プロパティ・フィールドのデフォルトの長さは 255 文字です。

コネクタは、以下の順序に従ってプロパティの値を決定します (最も番号の大きい項目が他の項目よりも優先されます)。

1. デフォルト
2. リポジトリ (WebSphere InterChange Server が統合ブローカーである場合のみ)
3. ローカル構成ファイル
4. コマンド行

コネクタは、始動時に構成値を取得します。実行時セッション中に 1 つ以上のコネクタ・プロパティの値を変更した場合、そのプロパティの**更新メソッド**によって、変更を有効にする方法が決まります。標準コネクタ・プロパティには、以下の 4 種類の更新メソッドがあります。

• 動的

変更を System Manager に保管すると、変更が即時に有効になります。コネクタが System Manager から独立してスタンドアロン・モードで稼働している場合 (例えばいずれかの WebSphere Message Broker と連携している場合) は、構成ファイルでのみプロパティを変更できます。この場合、動的更新は実行できません。

• コンポーネント再始動

System Manager でコネクタを停止して再始動しなければ、変更が有効になりません。アプリケーション固有コンポーネントまたは統合ブローカーを停止、再始動する必要はありません。

• サーバー再始動

アプリケーション固有のコンポーネントおよび統合ブローカーを停止して再始動しなければ、変更が有効になりません。

• エージェント再始動 (ICS のみ)

アプリケーション固有のコンポーネントを停止して再始動しなければ、変更が有効になりません。

特定のプロパティの更新方法を確認するには、「Connector Configurator」ウィンドウ内の「更新メソッド」列を参照するか、次に示すプロパティの要約の表の「更新メソッド」列を参照してください。

標準プロパティの要約

表 22 は、標準コネクタ構成プロパティの早見表です。標準プロパティの依存関係は RepositoryDirectory に基づいているため、コネクタによっては使用されないプロパティがあり、使用する統合ブローカーによってプロパティの設定が異なる可能性があります。

コネクタを実行する前に、これらのプロパティの一部の値を設定する必要があります。各プロパティの説明については、次のセクションを参照してください。

表 22. 標準構成プロパティの要約

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
AdminInQueue	有効な JMS キュー名	CONNECTORNAME /ADMININQUEUE	コンポーネント再始動	Delivery Transport は JMS
AdminOutQueue	有効な JMS キュー名	CONNECTORNAME/ADMINOUTQUEUE	コンポーネント再始動	Delivery Transport は JMS
AgentConnections	1 から 4	1	コンポーネント再始動	Delivery Transport は MQ および IDL: Repository Directory は <REMOTE>
AgentTraceLevel	0 から 5	0	動的	
ApplicationName	アプリケーション名	コネクタ・アプリケーション名として指定された値	コンポーネント再始動	
BrokerType	ICS、WMQI、WAS			
CharacterEncoding	ascii7、ascii8、SJIS、Cp949、GBK、Big5、Cp297、Cp273、Cp280、Cp284、Cp037、Cp437 注: これは、サポートされる値の一部です。	ascii7	コンポーネント再始動	
ConcurrentEventTriggeredFlows	1 から 32,767	1	コンポーネント再始動	Repository Directory は <REMOTE>
ContainerManagedEvents	値なし、または JMS	値なし	コンポーネント再始動	Delivery Transport は JMS
ControllerStoreAndForwardMode	True または False	True	動的	Repository Directory は <REMOTE>
ControllerTraceLevel	0 から 5	0	動的	Repository Directory は <REMOTE>
DeliveryQueue		CONNECTORNAME/DELIVERYQUEUE	コンポーネント再始動	JMS トランスポートのみ
DeliveryTransport	MQ、IDL、または JMS	JMS	コンポーネント再始動	Repository Directory がローカルの場合は、値は JMS のみ
DuplicateEventElimination	True または False	False	コンポーネント再始動	JMS トランスポートのみ、Container Managed Events は <NONE> でなければならない

表 22. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
FaultQueue		CONNECTORNAME/FAULTQUEUE	コンポーネント再始動	JMS トランスポートのみ
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory または CxCommon.Messaging.jms.SonicMQFactory または任意の Java クラス名	CxCommon.Messaging.jms.IBMMQSeriesFactory	コンポーネント再始動	JMS トランスポートのみ
jms.MessageBrokerName	FactoryClassName が IBM の場合は、crossworlds.queue.manager を使用。 FactoryClassName が Sonic の場合は、localhost:2506 を使用。	crossworlds.queue.manager	コンポーネント再始動	JMS トランスポートのみ
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	JMS トランスポートのみ
jms.Password	任意の有効なパスワード		コンポーネント再始動	JMS トランスポートのみ
jms.UserName	任意の有効な名前		コンポーネント再始動	JMS トランスポートのみ
JvmMaxHeapSize	ヒープ・サイズ (メガバイト単位)	128m	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMaxNativeStackSize	スタックのサイズ (キロバイト単位)	128k	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMinHeapSize	ヒープ・サイズ (メガバイト単位)	1m	コンポーネント再始動	Repository Directory は <REMOTE>
ListenerConcurrency	1 から 100	1	コンポーネント再始動	Delivery Transport は MQ でなければならない
Locale	en_US、ja_JP、ko_KR、zh_CN、zh_TW、fr_FR、de_DE、it_IT、es_ES、pt_BR 注: これは、サポートされるロケールの一部です。	en_US	コンポーネント再始動	
LogAtInterchangeEnd	True または False	False	コンポーネント再始動	Repository Directory は <REMOTE> でなければならない
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE> でなければならない
MessageFileName	パスまたはファイル名	InterchangeSystem.txt	コンポーネント再始動	

表 22. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
MonitorQueue	任意の有効なキュー名	CONNECTORNAME/MONITORQUEUE	コンポーネント再始動	JMS トランスポートのみ: DuplicateEvent Elimination は True でなければならぬ
OADAutoRestartAgent	True または False	False	動的	Repository Directory は <REMOTE> でなければならぬ
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE> でなければならぬ
OADRetryTimeInterval	正数 (分単位)	10	動的	Repository Directory は <REMOTE> でなければならぬ
PollEndTime	HH:MM	HH:MM	コンポーネント再始動	
PollFrequency	正整数 (単位: ミリ秒) no (ポーリングを使用不可にする) key (コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力された場合にのみポーリングする)	10000	動的	
PollQuantity	1 から 500	1	エージェント再始動	JMS トランスポートのみ: Container Managed Events を指定
PollStartTime	HH:MM (HH は 0 から 23、MM は 0 から 59)	HH:MM	コンポーネント再始動	
RepositoryDirectory	メタデータ・リポジトリの場所		エージェント再始動	ICS: <REMOTE> に設定する。 WebSphere MQ Message Brokers および WAS の場合: C:¥crossworlds¥ repository に設定する。
RequestQueue	有効な JMS キュー名	CONNECTORNAME/REQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
ResponseQueue	有効な JMS キュー名	CONNECTORNAME/RESPONSEQUEUE	コンポーネント再始動	Delivery Transport が JMS の場合: Repository Directory が <REMOTE> の場合のみ必要

表 22. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
RestartRetryCount	0 から 99	3	動的	
RestartRetryInterval	適切な正数 (単位: 分): 1 から 2147483547	1	動的	
RHF2MessageDomain	mrm、xml	mrm	コンポーネント再始動	Delivery Transport が JMS であり、かつ WireFormat が CwXML である
SourceQueue	有効な WebSphere MQ 名	CONNECTORNAME/SOURCEQUEUE	エージェント再始動	Delivery Transport が JMS であり、かつ Container Managed Events が指定されている場合のみ
SynchronousRequestQueue		CONNECTORNAME/ SYNCHRONOUSREQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
SynchronousRequestTimeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	Delivery Transport は JMS
SynchronousResponseQueue		CONNECTORNAME/ SYNCHRONOUSRESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS
WireFormat	CwXML、CwBO	CwXML	エージェント再始動	Repository Directory が <REMOTE> でない場合は CwXML。Repository Directory が <REMOTE> であれば CwBO
WsifSynchronousRequestTimeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	WAS のみ
XMLNameSpaceFormat	short、long	short	エージェント再始動	WebSphere MQ Message Brokers および WAS のみ

標準構成プロパティ

このセクションでは、各標準コネクタ構成プロパティの定義を示します。

AdminInQueue

統合ブローカーからコネクタへ管理メッセージが送信されるときに使用されるキューです。

デフォルト値は CONNECTORNAME/ADMININQUEUE です。

AdminOutQueue

コネクタから統合ブローカーへ管理メッセージが送信されるときに使用されるキューです。

デフォルト値は `CONNECTORNAME/ADMINOUTQUEUE` です。

AgentConnections

`RepositoryDirectory` が `<REMOTE>` の場合のみ適用されます。

`AgentConnections` プロパティは、`orb.init[]` により開かれる ORB 接続の数を制御します。

デフォルトでは、このプロパティの値は 1 に設定されます。このデフォルト値を変更する必要はありません。

AgentTraceLevel

アプリケーション固有のコンポーネントのトレース・メッセージのレベルです。デフォルトは、0 です。コネクタは、設定されたトレース・レベル以下の該当するトレース・メッセージをすべて搬送します。

ApplicationName

コネクタのアプリケーションを一意的に特定する名前です。この名前は、システム管理者が WebSphere Business Integration システム環境をモニターするために使用されます。コネクタを実行する前に、このプロパティに値を指定する必要があります。

BrokerType

使用する統合ブローカー・タイプを指定します。オプションは ICS、WebSphere Message Brokers (WMQI、WMQIB または WBIMB) または WAS です。

CharacterEncoding

文字 (アルファベットの文字、数値表現、句読記号など) から数値へのマッピングに使用する文字コード・セットを指定します。

注: Java ベースのコネクタでは、このプロパティは使用しません。C++ コネクタでは、現在、このプロパティに `ascii7` という値が使用されています。

デフォルトでは、ドロップ・リストには、サポートされる文字エンコード方式の一部が表示されます。サポートされるその他の値をドロップ・リストに追加するには、製品ディレクトリーにある `¥Data¥Std¥stdConnProps.xml` ファイルを手動で変更する必要があります。詳細については、Connector Configurator に関する付録を参照してください。

ConcurrentEventTriggeredFlows

`RepositoryDirectory` が `<REMOTE>` の場合のみ適用されます。

コネクタがイベント・デリバリー時に並行処理できるビジネス・オブジェクトの数を決定します。この属性の値は、並行してマップおよびデリバリーされるようにするビジネス・オブジェクトの数に設定します。例えば、このプロパティの値を 5 に設定すると、5 個のビジネス・オブジェクトが並行して処理されます。デフォルト値は 1 です。

このプロパティを 1 よりも大きい値に設定すると、ソース・アプリケーションのコネクターでは、複数のイベント・ビジネス・オブジェクトを同時にマップして、複数のコラボレーション・インスタンスにそれらのビジネス・オブジェクトを同時にデリバリーすることができます。これにより、統合ブローカーへのビジネス・オブジェクトのデリバリーにかかる時間、特にビジネス・オブジェクトが複雑なマップを使用している場合のデリバリー時間が短縮されます。ビジネス・オブジェクトがコラボレーションに到達する速度を増大させると、システム全体のパフォーマンスを向上させることができます。

ソース・アプリケーションから宛先アプリケーションまでのフロー全体に並行処理を実装するには、次のようにする必要があります。

- コラボレーションの `Maximum number of concurrent events` プロパティの値を増やして、コラボレーションが複数のスレッドを使用できるように構成します。
- 宛先アプリケーションのアプリケーション固有のコンポーネントが複数の要求を並行して実行できることを確認します。つまり、このコンポーネントが、マルチスレッド化されているか、または、コネクター・エージェント並列処理を使用でき、マルチプロセスに対応できるように構成されていなければなりません。
`Parallel Process Degree` 構成プロパティに、1 よりも大きい値を設定する必要があります。

`ConcurrentEventTriggeredFlows` プロパティは、順次実行される単一スレッド処理であるコネクターのポーリングでは無効です。

ContainerManagedEvents

このプロパティにより、JMS イベント・ストアを使用する JMS 対応コネクターが、保証付きイベント・デリバリーを提供できるようになります。保証付きイベント・デリバリーでは、イベントはソース・キューから除去され、単一 JMS トランザクションとして宛先キューに配置されます。

デフォルト値は `No value` です。

`ContainerManagedEvents` を JMS に設定した場合には、保証付きイベント・デリバリーを使用できるように次のプロパティも構成する必要があります。

- `PollQuantity` = 1 から 500
- `SourceQueue` = `CONNECTORNAME/SOURCEQUEUE`

また、`MimeType`、`DHClass`、および `DataHandlerConfigMOName` (オプション) プロパティを設定したデータ・ハンドラーも構成する必要があります。これらのプロパティの値を設定するには、`Connector Configurator` の「データ・ハンドラー」タブを使用します。「データ・ハンドラー (Data Handler)」タブの値のフィールドは、`ContainerManagedEvents` を JMS に設定した場合のみ表示されます。

注: `ContainerManagedEvents` を JMS に設定すると、コネクターはその `pollForEvents()` メソッドを呼び出さなくなるため、そのメソッドの機能が使用できなくなります。

このプロパティは、`DeliveryTransport` プロパティが値 JMS に設定されている場合にのみ表示されます。

ControllerStoreAndForwardMode

RepositoryDirectory が <REMOTE> の場合のみ適用されます。

宛先側のアプリケーション固有のコンポーネントが使用不可であることをコネクタ
ー・コントローラーが検出した場合に、コネクタ
ー・コントローラーが実行する動
作を設定します。

このプロパティを true に設定した場合、イベントが ICS に到達したときに宛先
側のアプリケーション固有のコンポーネントが使用不可であれば、コネクタ
ー・コ
ントローラーはそのアプリケーション固有のコンポーネントへの要求をブロックし
ます。該当のアプリケーション固有のコンポーネントが作動可能になると、コネク
ター・コントローラーはそのアプリケーション固有のコンポーネントに要求を転送
します。

ただし、コネクタ
ー・コントローラーが宛先側のアプリケーション固有のコンポー
ネントにサービス呼び出し要求を転送した後でこのコンポーネントが使用不可にな
った場合、コネクタ
ー・コントローラーはその要求を失敗させます。

このプロパティを false に設定した場合、コネクタ
ー・コントローラーは、宛先
側のアプリケーション固有のコンポーネントが使用不可であることを検出すると、
ただちにすべてのサービス呼び出し要求を失敗させます。

デフォルトは true です。

ControllerTraceLevel

RepositoryDirectory が <REMOTE> の場合のみ適用されます。

コネクタ
ー・コントローラーのトレース・メッセージのレベルです。デフォルト
は、0 です。

DeliveryQueue

DeliveryTransport が JMS の場合のみ適用されます。

コネクタ
ーがビジネス・オブジェクトを統合ブローカーに送信するために使用され
るキュー。

デフォルト値は CONNECTORNAME/DELIVERYQUEUE です。

DeliveryTransport

イベント・デリバリーのためのトランスポート機構を指定します。指定可能な値
は、WebSphere MQ の MQ、CORBA IIOP の IDL、Java Messaging Service の JMS
です。

- ICS がブローカー・タイプの場合は、DeliveryTransport プロパティに指定可
能な値は MQ、IDL、または JMS であり、デフォルトは IDL になります。
- RepositoryDirectory がローカル・ディレクトリーの場合は、指定可能な値は
JMS のみです。

DeliveryTransport プロパティに指定されている値が、MQ または IDL である場合、コネクタは、CORBA IIOP を使用してサービス呼び出し要求と管理メッセージを送信します。

WebSphere MQ および IDL

イベントのデリバリー・トランスポートには、IDL ではなく WebSphere MQ を使用してください (1 種類の製品だけを使用する必要がある場合を除きます)。

WebSphere MQ が IDL よりも優れている点は以下のとおりです。

- 非同期通信:
WebSphere MQ を使用すると、アプリケーション固有のコンポーネントは、サーバーが利用不能である場合でも、イベントをポーリングして永続的に格納することができます。
- サーバー・サイド・パフォーマンス:
WebSphere MQ を使用すると、サーバー・サイドのパフォーマンスが向上します。最適化モードでは、WebSphere MQ はイベントへのポインターのみをリポジトリ・データベースに格納するので、実際のイベントは WebSphere MQ キュー内に残ります。これにより、サイズが大きい可能性のあるイベントをリポジトリ・データベースに書き込む必要がありません。
- エージェント・サイド・パフォーマンス:
WebSphere MQ を使用すると、アプリケーション固有のコンポーネント側のパフォーマンスが向上します。WebSphere MQ を使用すると、コネクタのポーリング・スレッドは、イベントを選出した後、コネクタのキューにそのイベントを入れ、次のイベントを選出します。この方法は IDL よりも高速で、IDL の場合、コネクタのポーリング・スレッドは、イベントを選出した後、ネットワーク経由でサーバー・プロセスにアクセスしてそのイベントをリポジトリ・データベースに永続的に格納してから、次のイベントを選出する必要があります。

JMS

Java Messaging Service (JMS) を使用しての、コネクタとクライアント・コネクタ・フレームワークとの間の通信を可能にします。

JMS をデリバリー・トランスポートとして選択した場合、

`jms.MessageBrokerName`、`jms.FactoryClassName`、`jms.Password`、`jms.UserName` などの追加の JMS プロパティが Connector Configurator に表示されます。このうち最初の 2 つは、このトランスポートの必須プロパティです。

重要: 以下の環境では、コネクタに JMS トランスポート機構を使用すると、メモリ制限が発生することもあります。

- AIX 5.0
- WebSphere MQ 5.3.0.1
- ICS が統合ブローカーである場合

この環境では、WebSphere MQ クライアント内でメモリが使用されるため、(サーバー・サイドの) コネクタ・コントローラーと (クライアント・サイドの) コネクタの両方を始動するのは困難な場合があります。ご使用のシステムのプロセス・ヒープ・サイズが 768M 未満である場合には、次のように設定することをお勧めします。

- CWSharedEnv.sh スクリプト内で LDR_CNTRL 環境変数を設定する。

このスクリプトは、製品ディレクトリー配下の ¥bin ディレクトリーにあります。テキスト・エディターを使用して、CWSharedEnv.sh スクリプトの最初の行として次の行を追加します。

```
export LDR_CNTRL=MAXDATA=0x30000000
```

この行は、ヒープ・メモリーの使用量を最大 768 MB (3 セグメント * 256 MB) に制限します。プロセス・メモリーがこの制限値を超えると、ページ・スワッピングが発生し、システムのパフォーマンスに悪影響を与える場合があります。

- IPCCBaseAddress プロパティーの値を 11 または 12 に設定する。このプロパティーの詳細については、「システム・インストール・ガイド (UNIX 版)」を参照してください。

DuplicateEventElimination

このプロパティーを true に設定すると、JMS 対応コネクターによってデリバリー・キューに重複イベントがデリバリーされるのを防止することができます。この機能を使用するには、コネクターに対し、アプリケーション固有のコード内でビジネス・オブジェクトの **ObjectEventId** 属性として一意のイベント ID が設定されている必要があります。これはコネクター開発時に設定されます。

このプロパティーは、false に設定することもできます。

注: DuplicateEventElimination を true に設定した場合は、保証付きイベント・デリバリーを使用可能にするために、MonitorQueue プロパティーも構成する必要があります。

FaultQueue

コネクターでメッセージを処理中にエラーが発生すると、コネクターは、そのメッセージを状況表示および問題説明とともにこのプロパティーに指定されているキューに移動します。

デフォルト値は CONNECTORNAME/FAULTQUEUE です。

JvmMaxHeapSize

エージェントの最大ヒープ・サイズ (メガバイト単位)。このプロパティーは、RepositoryDirectory の値が <REMOTE> の場合のみ適用できます。

デフォルト値は 128M です。

JvmMaxNativeStackSize

エージェントの最大ネイティブ・スタック・サイズ (キロバイト単位)。このプロパティーは、RepositoryDirectory の値が <REMOTE> の場合のみ適用できます。

デフォルト値は 128K です。

JvmMinHeapSize

エージェントの最小ヒープ・サイズ (メガバイト単位)。このプロパティは、RepositoryDirectory の値が <REMOTE> の場合のみ適用できます。

デフォルト値は 1M です。

jms.FactoryClassName

JMS プロバイダー用にインスタンスを生成するためのクラス名を指定します。JMS をデリバリー・トランスポート機構 (DeliveryTransport) として選択する際は、このコネクター・プロパティを必ず設定してください。

デフォルト値は CxCommon.Messaging.jms.IBMMQSeriesFactory です。

jms.MessageBrokerName

JMS プロバイダー用に使用するブローカー名を指定します。JMS をデリバリー・トランスポート機構 (DeliveryTransport) として選択する際は、このコネクター・プロパティを必ず設定してください。

デフォルト値は crossworlds.queue.manager です。

jms.NumConcurrentRequests

1 つのコネクターに対して同時に送信可能な並行サービス呼び出し要求の最大数を指定します。この最大値に達した場合、新規のサービス呼び出し要求はブロックされ、既存のいずれかの要求が完了した後で処理されます。

デフォルト値は 10 です。

jms.Password

JMS プロバイダー用のパスワードを指定します。このプロパティの値はオプションです。

デフォルトはありません。

jms.UserName

JMS プロバイダー用のユーザー名を指定します。このプロパティの値はオプションです。

デフォルトはありません。

ListenerConcurrency

このプロパティは、ICS が統合ブローカーである場合に、MQ Listener でのマルチスレッド化をサポートします。このプロパティによって、データベースへの複数イベントの書き込みをバッチ処理可能にすることができます。これにより、システムのパフォーマンスが向上します。デフォルト値は 1 です。

このプロパティは、MQ トランスポートを使用するコネクターにのみ適用されます。DeliveryTransport プロパティは MQ に設定する必要があります。

Locale

言語コード、国または地域、および、希望する場合には、関連した文字コード・セットを指定します。このプロパティの値は、データの照合やソートの順序、日付と時刻の形式、通貨記号などの国/地域別情報を決定します。

ロケール名は、次の書式で指定します。

`ll_TT.codeset`

ここで、

<code>ll</code>	2 文字の言語コード (普通は小文字)
<code>TT</code>	2 文字の国コードまたは地域コード (普通は大文字)
<code>codeset</code>	関連文字コード・セットの名前。名前のこの部分は、通常、オプションです。

デフォルトでは、ドロップ・リストには、サポートされるロケールの一部のみが表示されます。サポートされるその他の値をドロップ・リストに追加するには、製品ディレクトリーにある `¥Data¥Std¥stdConnProps.xml` ファイルを手動で変更する必要があります。詳細については、Connector Configurator に関する付録を参照してください。

デフォルト値は `en_US` です。コネクタがグローバル化に対応していない場合、このプロパティの有効な値は `en_US` のみです。特定のコネクタがグローバル化に対応しているかどうかを判別するには、以下の Web サイトにあるコネクタのバージョン・リストを参照してください。

<http://www.ibm.com/software/websphere/wbiadapters/infocenter>、または
<http://www.ibm.com/websphere/integration/wicsserver/infocenter>

LogAtInterchangeEnd

`RepositoryDirectory` が `<REMOTE>` の場合のみ適用されます。

統合ブローカーのログ宛先にエラーを記録するかどうかを指定します。ブローカーのログ宛先にログを記録すると、電子メール通知もオンになります。これにより、エラーまたは致命的エラーが発生すると、`InterchangeSystem.cfg` ファイルに指定された `MESSAGE_RECIPIENT` に対する電子メール・メッセージが生成されます。

例えば、`LogAtInterChangeEnd` を `true` に設定した場合、コネクタからアプリケーションへの接続が失われると、指定されたメッセージ宛先に電子メール・メッセージが送信されます。デフォルト値は `false` です。

MaxEventCapacity

コントローラー・バッファ内のイベントの最大数。このプロパティはフロー制御が使用し、`RepositoryDirectory` プロパティの値が `<REMOTE>` の場合のみ適用できます。

値は 1 から 2147483647 の間の正整数です。デフォルト値は 2147483647 です。

MessageFileName

コネクター・メッセージ・ファイルの名前です。メッセージ・ファイルの標準の格納場所は `¥connectors¥messages` です。メッセージ・ファイルが標準位置に格納されていない場合は、メッセージ・ファイル名を絶対パスで指定します。

コネクター・メッセージ・ファイルが存在しない場合は、コネクターは `InterchangeSystem.txt` をメッセージ・ファイルとして使用します。このファイルは、製品ディレクトリーに格納されています。

注: 特定のコネクターについて、コネクター独自のメッセージ・ファイルがあるかどうかを判別するには、該当するアダプターのユーザーズ・ガイドを参照してください。

MonitorQueue

コネクターが重複イベントをモニターするために使用する論理キューです。このプロパティーは、`DeliveryTransport` プロパティー値が `JMS` であり、かつ `DuplicateEventElimination` が `TRUE` に設定されている場合のみ使用されます。

デフォルト値は `CONNECTORNAME/MONITORQUEUE` です。

OADAutoRestartAgent

`RepositoryDirectory` が `<REMOTE>` の場合のみ有効です。

コネクターが自動再始動機能およびリモート再始動機能を使用するかどうかを指定します。この機能では、MQ によりトリガーされる `Object Activation Daemon (OAD)` を使用して、異常シャットダウン後にコネクターを再始動したり、`System Monitor` からリモート・コネクターを始動したりします。

自動再始動機能およびリモート再始動機能を使用可能にするには、このプロパティーを `true` に設定する必要があります。MQ によりトリガーされる OAD 機能の構成方法については、「システム・インストール・ガイド (Windows 版)」または「システム・インストール・ガイド (UNIX 版)」を参照してください。

デフォルト値は `false` です。

OADMaxNumRetry

`RepositoryDirectory` が `<REMOTE>` の場合のみ有効です。

異常シャットダウンの後で MQ によりトリガーされる OAD がコネクターの再始動を自動的に試行する回数の最大数を指定します。このプロパティーを有効にするには、`OADAutoRestartAgent` プロパティーを `true` に設定する必要があります。

デフォルト値は `1000` です。

OADRetryTimeInterval

`RepositoryDirectory` が `<REMOTE>` の場合のみ有効です。

MQ によりトリガーされる OAD の再試行時間間隔の分数を指定します。コネクター・エージェントがこの再試行時間間隔内に再始動しない場合は、コネクター・コ

ントローラーはコネクタ・エージェントを再び再始動するように OAD に要求します。OAD はこの再試行プロセスを OADMaxNumRetry プロパティで指定された回数だけ繰り返します。このプロパティを有効にするには、OADAutoRestartAgent プロパティを true に設定する必要があります。

デフォルトは 10 です。

PollEndTime

イベント・キューのポーリングを停止する時刻です。形式は HH:MM です。ここで、HH は 0 から 23 時を表し、MM は 0 から 59 分を表します。

このプロパティには、有効な値をユーザーが指定する必要があります。デフォルト値は HH:MM ですが、変更する必要があります。

PollFrequency

各ポーリング・アクション間での時間。PollFrequency は以下の値のいずれかに設定します。

- 各ポーリング・アクション間でのミリ秒数。
- ワード key。コネクタは、コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力されたときにのみポーリングを実行します。入力は小文字で行います。
- ワード no。コネクタは、ポーリングを実行しません。入力は小文字で行います。

デフォルトは 10000 です。

重要: このプロパティの使用に関して制約事項のあるコネクタもあります。特定のコネクタが該当するかどうかを判別するには、それぞれのアダプター・ガイドのインストールと構成に関する章を参照してください。

PollQuantity

コネクタがアプリケーションからポーリングする項目の数を指定します。アダプターにコネクタ固有のポーリング数設定プロパティがある場合、標準プロパティの値は、このコネクタ固有のプロパティの設定値によりオーバーライドされます。

PollStartTime

イベント・キューのポーリングを開始する時刻。形式は HH:MM です。ここで、HH は 0 から 23 時を表し、MM は 0 から 59 分を表します。

このプロパティには、有効な値をユーザーが指定する必要があります。デフォルト値は HH:MM ですが、変更する必要があります。

RequestQueue

統合ブローカーが、ビジネス・オブジェクトをコネクタに送信するときに使用されるキューです。

デフォルト値は CONNECTOR/REQUESTQUEUE です。

RepositoryDirectory

コネクタが XML スキーマ文書を読み取るリポジトリの場所です。この XML スキーマ文書には、ビジネス・オブジェクト定義のメタデータが含まれています。

統合ブローカーが ICS の場合には、この値を <REMOTE> に設定する必要があります。これは、コネクタが InterChange Server のリポジトリからこの情報を取得するためです。

統合ブローカーが WebSphere Message Broker または WAS の場合には、この値を <local directory> に設定する必要があります。

ResponseQueue

DeliveryTransport が JMS の場合のみ適用され、RepositoryDirectory が <REMOTE> の場合のみ必要です。

JMS 応答キューを指定します。このキューは、応答メッセージをコネクタ・フレームワークから統合ブローカーに配信します。統合ブローカーが ICS である場合、統合ブローカーは、要求を送信した後、JMS 応答キューで応答メッセージを待機します。

RestartRetryCount

コネクタが再始動を試行する回数を指定します。このプロパティを並列コネクタに対して使用する場合、コネクタのマスター側のアプリケーション固有のコンポーネントがスレーブ側のアプリケーション固有のコンポーネントの再始動を試行する回数が指定されます。

デフォルト値は 3 です。

RestartRetryInterval

コネクタが再始動を試行する間隔 (単位: 分) を指定します。このプロパティを並列コネクタに対して使用する場合、コネクタのマスター側のアプリケーション固有のコンポーネントがスレーブ側のアプリケーション固有のコンポーネントの再始動を試行する間隔が指定されます。指定可能な値の範囲は 1 から 2147483647 です。

デフォルト値は 1 です。

RHF2MessageDomain

WebSphere Message Brokers および WAS でのみ使用されます。

このプロパティにより、JMS ヘッダーのドメイン名フィールドの値を構成できます。JMS トランスポートを介してデータを WMQI に送信するときに、アダプター・フレームワークにより JMS ヘッダー情報、ドメイン名、および固定値 mrm が書き込まれます。この構成可能なドメイン名により、ユーザーは WMQI ブローカーによるメッセージ・データの処理方法を追跡できます。

サンプル・ヘッダーを以下に示します。

```
<mcd><Msd>mrm</Msd><Set>3</Set><Type>
Retek_POPhyDesc</Type><Fmt>CwXML</Fmt></mcd>
```

デフォルト値は `mrm` ですが、このプロパティーには `xml` も設定できます。このプロパティーは、`DeliveryTransport` が `JMS` に設定されており、かつ `WireFormat` が `CwXML` に設定されている場合にだけ表示されます。

SourceQueue

`DeliveryTransport` が `JMS` で、`ContainerManagedEvents` が指定されている場合のみ適用されます。

`JMS` イベント・ストアを使用する `JMS` 対応コネクターの保証付きイベント・デリバリーをサポートする、コネクター・フレームワークの `JMS` ソース・キューを指定します。詳細については、132 ページの『`ContainerManagedEvents`』を参照してください。

デフォルト値は `CONNECTOR/SOURCEQUEUE` です。

SynchronousRequestQueue

`DeliveryTransport` が `JMS` の場合のみ適用されます。

同期応答を要求する要求メッセージを、コネクター・フレームワークからブローカーに配信します。このキューは、コネクターが同期実行方式を使用する場合にのみ必要です。同期実行の場合、コネクター・フレームワークは、`SynchronousRequestQueue` にメッセージを送信し、`SynchronousResponseQueue` でブローカーから戻される応答を待機します。コネクターに送信される応答メッセージには、元のメッセージの `ID` を指定する相関 `ID` が含まれています。

デフォルトは `CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE` です。

SynchronousResponseQueue

`DeliveryTransport` が `JMS` の場合のみ適用されます。

同期要求に対する応答として送信される応答メッセージを、ブローカーからコネクター・フレームワークに配信します。このキューは、コネクターが同期実行方式を使用する場合にのみ必要です。

デフォルトは `CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE` です。

SynchronousRequestTimeout

`DeliveryTransport` が `JMS` の場合のみ適用されます。

コネクターが同期要求に対する応答を待機する時間 (単位: 分) を指定します。指定された時間内に応答が受信されない場合、コネクターは、エラー・メッセージと共に元の同期要求メッセージを障害キューに移動します。

デフォルト値は `0` です。

WireFormat

トランスポートのメッセージ・フォーマットです。

- RepositoryDirectory がローカル・ディレクトリーの場合は、設定は CwXML になります。
- RepositoryDirectory の値が <REMOTE> である場合には、設定値は CwBO です。

WsifSynchronousRequest Timeout

WAS 統合ブローカーでのみ使用されます。

コネクタが同期要求に対する応答を待機する時間 (単位: 分) を指定します。指定された時間内に応答が受信されない場合、コネクタは、エラー・メッセージと共に元の同期要求メッセージを障害キューに移動します。

デフォルト値は 0 です。

XMLNamespaceFormat

WebSphere Message Broker と WAS 統合ブローカーでのみ使用されます。

ビジネス・オブジェクト定義の XML 形式でネーム・スペースを short と long のどちらにするかをユーザーが指定できるようにするための、強力なプロパティです。

デフォルト値は short です。

付録 D. コネクタ始動オプション

この後の各一覧表で、Windows または UNIX でコネクタを始動する際に指定できるコマンド行オプションについて紹介します。オプションによっては、コネクタの構成ファイル内で選択したプロパティ設定値をオーバーライドするものもあります。

注: 始動スクリプトの構造と作成方法の詳細は、「コネクタ開発ガイド (C++ 用)」または「コネクタ開発ガイド (Java 用)」を参照してください。

Windows

表 23 に記載されたコネクタ開始オプションを使用するには、下記の項目を編集してコネクタの開始条件を変更します。

- コネクタのショートカット・プロパティ。
- コネクタの始動ファイル。通常、`start_connName.bat`。

注: ご使用のコネクタがこの構文を使用するかどうかは、ディレクトリー `ProductDir/connectors` 内を調べると確認できます。`start_connName` という名前のスクリプト (ここで、`connName` はご使用のコネクタ名) が存在する場合、このファイルを編集します。

- コネクタの始動ファイル、`start_connector.bat` ファイル (`start_connName.bat` 始動ファイルを使用しない、C++ で記述されたコネクタの場合のみ)。

注: コネクタを始動するには、`-c` オプションが必要です。コネクタの始動にショートカットを使用する場合、ショートカット・プロパティにはターゲット・フィールドでこのオプションを組み込んでおく必要があります。同様に、`start_connector.bat` または `start_connName.bat` を使用してコネクタを始動する場合、コマンド行にこのオプションを指定する必要があります。

表 23. Windows でのコマンド行コネクタ始動オプション

オプション	説明
<code>-c configFile</code>	始動時に使用する構成ファイルの絶対パス名。当オプションは必須です。
<code>-f pollFrequency</code>	各ポーリング・アクション間での時間。 <code>pollFrequency</code> で指定される値は以下のとおりです。 <ul style="list-style-type: none">• 各ポーリング・アクション間でのミリ秒数。• key: コネクタは、コネクタの「コマンド・プロンプト」ウィンドウに英文字 <code>p</code> を入力したときのみ、ポーリングを行います。入力は小文字で行います。• no: コネクタはポーリングを行いません。入力は小文字で行います。 デフォルトは 10000 です。

表 23. Windows でのコマンド行コネクタ始動オプション (続き)

オプション	説明
-j	コネクタを Java で作成することを指定します。-l <i>className</i> を指定した場合、当オプションは任意となります。
-t	注: インストーラーでは、コネクタ側の必要に応じて、ショートカット内に -t オプションが組み込まれて (または除外されて) います。コマンド行からコネクタを始動する場合は、-t に指定する値が、ショートカットで指定した -t の値と一致していなければなりません。コネクタ・プロパティ SingleThreadAppCalls をオンの状態にします。このプロパティにより、アプリケーション固有のコネクタ・コードに対してコネクタ・フレームワークが行うすべての呼び出しに 1 つのイベント・トリガー・フローがあることが保証されます。デフォルト値は false です。このプロパティは、出荷時の値のままにしておいてください。各コネクタには、そのアーキテクチャーに応じて、当オプションに対して最適な設定値が用意されています。
-x <i>connectorProps</i>	アプリケーション固有のコネクタ・プロパティをコネクタに受け渡します。入力する値ごとに、フォーマット <i>prop_name=value</i> を使用します。

UNIX

UNIX 環境では、*connector_manager_connName* スクリプトを実行してコネクタを始動します。このスクリプトは、汎用コネクタ・マネージャー・スクリプト (*ProductDir/bin/connector_manager*) のラッパーです。汎用コネクタ・マネージャー・スクリプトは、該当する *start_connName.sh* スクリプトを呼び出します。このスクリプトは、コネクタに代わって実際のコネクタ管理を行います。

各 WebSphere Business Integration Adapter には、*start_connName.sh* スクリプトが含まれています。このスクリプトを変更して、表 24 にある、サポートされている始動オプションのいずれかを指定することもできます。

表 24. *start_connector.sh* スクリプトのオプション

オプション	説明
- <i>fpoll_freq</i>	各ポーリング・アクション間での時間。poll_freq で指定される値は以下のとおりです。 <ul style="list-style-type: none"> 各ポーリング・アクション間でのミリ秒数。 key: コネクタは、コネクタの「コマンド・プロンプト」ウィンドウに英文字 p を入力したときにのみ、ポーリングを行います。入力は小文字で行います。 no: コネクタはポーリングを行いません。入力は小文字で行います。 デフォルトは 10000 です。-f オプションは、 <i>connector_manager_connector</i> のコマンド行呼び出しで有効です。コネクタ・マネージャー・スクリプトは当オプションを関連した <i>start_connector.sh</i> スクリプトに受け渡すことができます。当オプションで、コネクタの構成ファイルで指定済みのポーリング頻度がオーバーライドされます。

表 24. `start_connector.sh` スクリプトのオプション (続き)

オプション	説明
<p><code>-tthreading_type</code></p>	<p><code>threading_type</code> オプションでは、スレッド化モデルを指定します。</p> <p>注: カスタム開発コネクタを始動する場合は、<code>-t</code> オプションのみを使用します。WebSphere Business Integration Adapters インストーラーを使用してインストールされるコネクタには <code>connector_manager_connector</code> 始動スクリプトがすでに存在します。この始動スクリプトは、アプリケーション側の必要に応じて、コネクタを開始する行で必須の <code>-t</code> オプションを指定 (または除外) します。 <code>threading_type</code> で使用される値は以下のとおりです。</p> <ul style="list-style-type: none"> • <code>SINGLE_THREADED</code>: 単一スレッドのみがアプリケーションにアクセスします。 • <code>MAIN_SINGLE_THREADED</code>: メイン・スレッドのみがアプリケーションにアクセスします。 • <code>MULTI_THREADED</code>: 複数のスレッドからアプリケーションにアクセスできます。 <p><code>-t</code> オプションは、<code>connector_manager_connName</code> のコマンド行呼び出しでは無効です。<code>start_connector.sh</code> スクリプトの起動時に汎用の <code>connector_manager</code> スクリプト内で、当オプションを指定します。</p>

付録 E. System Manager と Eclipse Workbench

この付録では、System Manager と Eclipse Workbench の機能の使用について、一般的な説明の一部を示します。

注: 一部の System Manager 機能の詳細は、本書で説明されています。Adapter Monitor および障害キュー・マネージャーの詳細については、72 ページの『Adapter Monitor および障害キュー・マネージャーの使用』を参照してください。新規プロジェクトの作成については、94 ページの『メッセージ・ブローカー・プロジェクトの作成』を参照してください。

この章には、以下のセクションが含まれています。

- 『System Manager とは』
- 150 ページの『System Manager の使用』
- 154 ページの『統合コンポーネント・ライブラリーでの作業』
- 156 ページの『ユーザー・プロジェクトでの作業』
- 161 ページの『統合コンポーネント・ライブラリー内のコンポーネントの処理』
- 165 ページの『ソリューションの処理』
- 168 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』
- 170 ページの『依存関係および参照』
- 171 ページの『複数のワークベンチ・リソースで使用可能な標準の操作』
- 173 ページの『Eclipse ベースのワークベンチの使用』
- 179 ページの『System Manager で統合ブローカーに接続する際の問題のトラブルシューティング』

System Manager とは

System Manager とは、WebSphere Studio Workbench および WebSphere Studio Application Developer Integration Edition という Eclipse ベースのツール・フレームワーク内で稼動するプラグインです。このセクションでは、Eclipse フレームワーク (WSWB と WSADIE) および IBM WebSphere アダプター・ツールのプラグインの概要について説明します。

Eclipse プラットフォームとは

Eclipse プラットフォームとは、ツール作成用のオープン・ソースの統合開発環境 (IDE) です。これにより、ツール開発者に開発キットとランタイムが提供され、開発者は、ユーザーが特定のタイプのリソースを使用して作業できるようにプラグインを作成することができます。

IBM では、WebSphere Studio WorkBench (WSWB) と WebSphere Studio Application Developer Integration Edition (WSADIE) という 2 つのバージョンの Eclipse プラットフォームを用意しています。

プラグイン

プラグインとは、ソフトウェア・ベンダーが開発して Eclipse ベースのワークベンチに機能性を追加するモジュラー拡張機能です。プラグインには、ワークベンチのユーザーが特定のタイプのリソースを使用して作業するためのパースペクティブ、エディター、およびビューがカプセル化されます。

例えば、あるプラグインはテキスト・エディター機能を提供します。別のプラグインは、HTML エディター機能を提供します。System Manager ツールのプラグインは、統合コンポーネントと連動する機能を提供します。プラグイン・モデルの利点は、ユーザーが多様なリソースを使用して作業する際に、各リソース・タイプの専用ツールではなく、単一のツールを使用できることです。

プラグインをインストールするには、プラグインを示す 1 つ以上の圧縮アーカイブを、ワークベンチの製品ディレクトリー内の `plugins` ディレクトリーに解凍します。System Manager プラグインが、インストーラーによって `plugins` ディレクトリーに解凍されます。

IBM では、ビジネス・インテグレーション・リソースを使用して作業するために、System Manager プラグインを WebSphere Business Integration Adapter フレームワークとともに提供します。これらのプラグインは、ワークベンチの `plugins` ディレクトリー内の多数の解凍ディレクトリーによって `com.ibm.btools` ネーム・スペース内に統合されます。統合コンポーネントの作成時に使用する基本インターフェースのほとんど (例えば System Manager) は、`com.ibm.btools.csm` プラグイン・ディレクトリーに含まれます。

ワークベンチ

ワークベンチとは、Eclipse ベースのツール・フレームワーク内でアクティブなパースペクティブ、エディター、およびビューの集合であり、すでにインストールされて使用可能なプラグインの集合からの影響を受けます。ワークベンチは、ユーザーが作業している Eclipse ベースのインターフェースを指す一般用語であり、そのインターフェースがユーザーの使用に応じて変更されることとは関係ありません。

ワークスペース

ワークスペースとは、プロジェクトのコンテナです。ワークスペースは、デフォルトでプロジェクトの保管先にするようプロンプトが出される、ファイル・システム内のディレクトリーです。

プロジェクト

プロジェクトとは、ユーザー定義のリソース・グループであり、最終的にはファイル・システム内のディレクトリーです。

ビジネス・プロセス・インターフェースを開発する際の最初の作業の 1 つは、統合コンポーネント・ライブラリー、つまり開発するコンポーネントを含むプロジェクトを定義することです。統合コンポーネント・ライブラリーを作成するときは、その保管先となるファイル・システム内の場所を指定します (デフォルトはワークスペース・ディレクトリー)。その場所に、統合コンポーネント・ライブラリーに指定した名前をもつフォルダーが作成されます。ライブラリー・フォルダー内には、統合コンポーネントのタイプごとに多数のフォルダーが作成されます (例えば、`BusinessObjects` や `Connectors` という名前のフォルダーが作成されます)。

また、**ユーザー・プロジェクト**という名前のプロジェクトも作成します。ユーザー・プロジェクトとは、統合コンポーネントを参照するショートカットの集合です。コンポーネントを統合ブローカーに配置するには、統合コンポーネント・ライブラリーの統合コンポーネントをユーザー・プロジェクトに追加する必要があります。ユーザー・プロジェクトは、コンポーネントをサーバーに配置するのに必要であるほか、ユーザーがコンポーネントを機能的にグループ化できるよう設計されています。統合コンポーネント・ライブラリーは、作業に必要なすべてのコンポーネントの集合ですが、ユーザー・プロジェクトは、ユーザーが作業するコンポーネントを特定のインターフェース用にグループ化できるよう設計されています。

リソース

リソースとは、ユーザーがワークベンチ内で作業するプロジェクト、ファイル、およびフォルダーです。

統合コンポーネントを作成すると、統合コンポーネント・ライブラリー・プロジェクト内の該当するフォルダーにファイルとして保管されます。統合コンポーネントは、タイプごとに異なる拡張子が付けられて保管されます (例えば、コネクターには `.con` という拡張子が付けられます) が、すべて XML 形式で保管されます。

パースペクティブ

パースペクティブとはエディターやビューをグループ化したものであり、特定ユーザーの役割に必要なものを提供します。例えば、System Manager パースペクティブは、統合コンポーネント・ライブラリーでの作業用のビューを提供します。

エディター

エディターを使用すると、ワークベンチでリソースをオープン、保管、およびクローズできます。

ビュー

ビューは、ユーザーがワークベンチで作業しているリソースについての情報を提供します。

例えば System Manager には、統合コンポーネント・ライブラリーおよびユーザー・プロジェクトに対するビューとして、WebSphere Business Integration System Manager ビューがあります。

WSWB および WSADIE とは

WebSphere Studio Workbench (WSWB) は、IBM リリースの Eclipse プラットフォームです。WSWB は統合ブローカーに組み込まれているので、コア・インフラストラクチャーとともにインストールすることができます。WSWB は、WebSphere Business Integration Adapters 統合コンポーネントの作成に必要なすべてのプラグインを実行することができます。

WebSphere Studio Application Developer Integration Edition (WSADIE) は、WSWB と同様に IBM リリースの Eclipse プラットフォームですが、新規プラグインの開発に使用することもできます。WSADIE は統合ブローカーに組み込まれていません。新規プラグインの作成機能は、統合コンポーネントの作成には必要ないためです。ただし、これをインストールすると、必要な System Manager プラグインおよび Integrated Test Environment プラグインを実行する際に使用できます。

System Manager とは

System Manager とは、ユーザーが WebSphere Business Integration Adapters ビジネス・インテグレーション・システムの統合コンポーネントおよびサーバー・インスタンスの作業を行うパースペクティブです。System Manager は、主に次の作業に使用します。

- WebSphere Business Integration Adapters ツール・セット内で他のツールを起動する
- 一部の統合コンポーネントを開発および構成する
- 統合コンポーネントをリポジトリまたはブローカーに配置する

System Manager の使用

このセクションでは、System Manager パースペクティブの始動方法と使用方法について説明します。

System Manager の始動

IBM WebSphere 統合ブローカーをインストールする場合には、ブローカーとの対話に必要なプラグインのサポートとともに WebSphere Studio Workbench をインストールするか、既存の WebSphere Studio Application Developer Integration Edition にプラグインをインストールするかを選択できます。

System Manager を始動するには、次の手順を実行します。

1. 「スタート」 > 「プログラム」 > 「**WebSphere Business Integration Adapters**」 > 「ツール」 > 「**System Manager**」を選択します。
2. メニュー・バーから、「**Windows**」 > 「**パースペクティブを開く**」 > 「**その他**」を選択します。
3. パースペクティブのリストから System Manager を選択し、「**OK**」をクリックします。

WebSphere Studio Workbench が始動し、表示されます。151 ページの図 28 に、System Manager パースペクティブを示します。『System Manager インターフェース』では、インターフェースおよびそのエレメントについて説明します。

System Manager インターフェース

System Manager パースペクティブが開くときのデフォルト構成には、幾つかのビューとエディターがあります。151 ページの図 28 に、System Manager パースペクティブを示します。

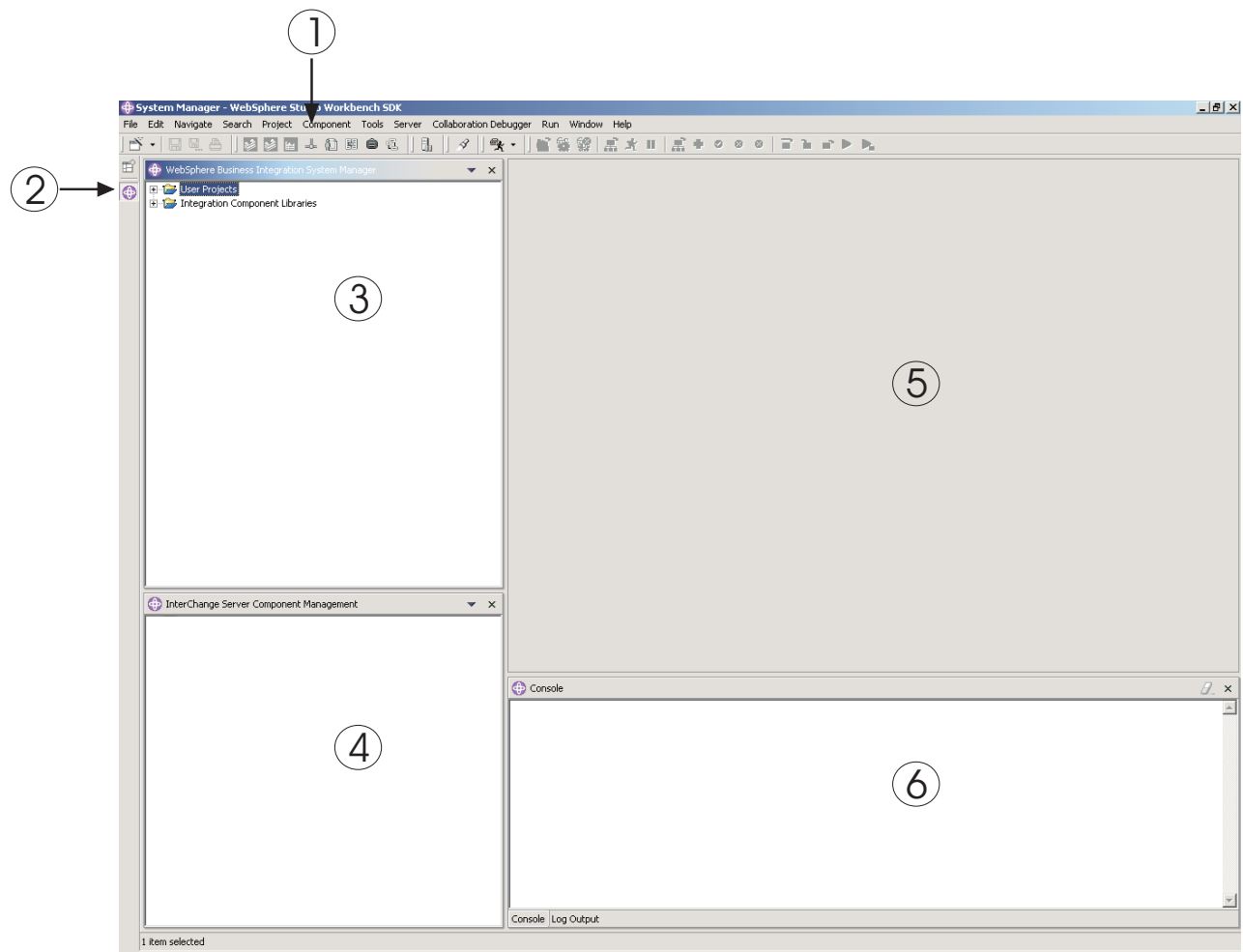


図 28. System Manager パースペクティブ

表 25 では、System Manager パースペクティブのインターフェース・エレメントについて、151 ページの図 28 の番号ごとに説明します。

表 25. System Manager パースペクティブのインターフェース・エレメント

インターフェース・エレメント番号	インターフェース・エレメント名
1	152 ページの『メニュー・バーとツールバー』
2	153 ページの『パースペクティブ・ショートカット・バー』
3	153 ページの『WebSphere Business Integration System Manager ビュー』
4	Interchange Server Component Management ビュー。このセクションが関連するのは、WebSphere Interchange Server を統合ブローカーとして実行している場合のみです。別のブローカーを使用している場合は、無視して構いません。
5	153 ページの『エディター・ビュー』
6	154 ページの『コンソール・ビュー』

以降のセクションでは、System Manager のインターフェース・エレメントについて詳しく説明します。

メニュー・バーとツールバー

メニュー・バーとツールバーは、Eclipse ベースのツール・フレームワークおよび統合ブローカー・コンポーネントでの作業を行う場合に使用します。メニュー・バー項目の多くにはツールバーと同等のものがあるので、以降のセクションではメニュー・バーとその項目のみを説明します。

「ファイル」メニュー: リソースの作業に使用する Eclipse 標準メニュー。主に、新規の統合コンポーネント・ライブラリーおよびユーザー・プロジェクトの作成に使用します。

統合コンポーネント・ライブラリーおよびユーザー・プロジェクトの作成の詳細については、154 ページの『統合コンポーネント・ライブラリーでの作業』および 156 ページの『ユーザー・プロジェクトでの作業』を参照してください。

「編集」メニュー: 「切り取り」、「コピー」、「貼り付け」など多数の標準項目がある Eclipse 標準メニュー。

作成したコンポーネントの切り取り、コピー、貼り付け、および削除の詳細については、161 ページの『統合コンポーネント・ライブラリー内のコンポーネントの処理』を参照してください。

「ナビゲート」メニュー: ワークベンチ内のリソース間をナビゲートするための Eclipse 標準メニュー。このメニューの詳細については、ワークベンチの資料を参照してください。

「検索」メニュー: リソースの検索およびリソース内の検索を可能にする Eclipse 標準メニュー。このメニューの詳細については、ワークベンチの資料を参照してください。

「プロジェクト」メニュー: プロジェクト・リソースを操作するためのメニュー項目を持つ Eclipse 標準メニュー。このメニュー項目は、System Manager パースペクティブの作業を行う際には使用されません。このメニューの詳細については、ワークベンチの資料を参照してください。

「コンポーネント」メニュー: このメニューは System Manager パースペクティブとともに提供され、ユーザーが作成した統合コンポーネントの作業を行う際に便利です。このメニューの項目は、本書内および他の資料の特定のタスクについて説明しているセクションに文書化されています。

「ツール」メニュー: このメニューは System Manager パースペクティブとともに提供され、統合コンポーネントの作成に使用するツールの起動に使用されます。

詳しくは、161 ページの『統合コンポーネント・ライブラリー内のコンポーネントの処理』を参照してください。

「サーバー」メニュー: このメニューの「サーバーの登録 (Register Servers)」項目を使用すると、InterChange Server インスタンスを登録できます。統合ブローカーが InterChange Server 以外の場合は、この項目は関係ありません。

「実行」メニュー: このメニューには、外部プログラムやバッチ・ファイルを実行するための外部ツールを構成したり、スクリプトを作成するための項目があります。詳しくは、ワークベンチの資料を参照してください。

「ウィンドウ」メニュー: このメニューには、パースペクティブ、ビュー、エディター、および設定での作業を行うための項目があります。

これらの項目については、ワークベンチの資料および本書のさまざまなセクションで説明しています。

「ヘルプ」メニュー: このメニューには、ワークベンチの資料を起動したり、ワークベンチおよびパースペクティブのバージョン情報を示す項目があります。

パースペクティブ・ショートカット・バー

パースペクティブ・ショートカット・バーを使用すると、異なるパースペクティブ間を簡単にナビゲートできます。例えば、System Manager パースペクティブと Java パースペクティブを開いている場合、パースペクティブ・ショートカット・ツールバーのワークスペース・アイコンをクリックすると、パースペクティブを切り替えることができます。

また、次のように「ウィンドウ」メニューを使用して、他のパースペクティブにナビゲートすることもできます。

- パースペクティブ・ショートカット・バーにおいて、現在表示しているパースペクティブのアイコンよりも上にあるアイコンのパースペクティブにナビゲートするには、キーボード・ショートカット **Alt + 上矢印** を使用します。
- パースペクティブ・ショートカット・バーにおいて、現在表示しているパースペクティブのアイコンよりも下にあるアイコンのパースペクティブにナビゲートするには、メニュー・バーから「**パースペクティブ (Perspective)**」 > 「**次へ**」を選択するか、キーボード・ショートカット **Alt + 下矢印** を使用します。

WebSphere Business Integration System Manager ビュー

このビューには、統合ブローカー・プロジェクトのタイプである**ユーザー・プロジェクト・ノード**と**統合コンポーネント・ライブラリー・ノード**があります。

これらのプロジェクト・タイプの作業の詳細については、154 ページの『**統合コンポーネント・ライブラリーでの作業**』および 156 ページの『**ユーザー・プロジェクトでの作業**』を参照してください。

InterChange Server Component Management ビュー

このビューは、統合ブローカーとして WebSphere InterChange Server を使用している場合に使用できます。その他の統合ブローカーの場合は関係ありません。

エディター・ビュー

このビューは、フレームワーク内でさまざまなリソース (ファイルや統合コンポーネント定義など) の作業を行う場合に使用します。異なるタイプのリソースの作業を行うには、それぞれ別のエディターを開きます。例えば、テキスト・ファイルはテキスト・エディター内で開きます。

コンソール・ビュー

このビューには、「コンソール」と「ログ出力」という 2 つのタブがあります。System Manager でマップまたはコラボレーション・テンプレートをコンパイルする場合、「コンソール」タブには、各コンポーネントのコンパイルが正常に完了したかどうかを示すメッセージが表示され、「ログ出力」タブには、発生したエラーや警告が表示されます。

統合コンポーネント・ライブラリーでの作業

統合コンポーネント・ライブラリーは、作成したコンポーネントを保管するために使用します。このセクションでは、新規統合コンポーネント・ライブラリーの作成方法について説明します。

統合コンポーネント・ライブラリーを作成すると、一般的に次の作業も実行するようになります。

- ライブラリーにコンポーネントをインポートする。この作業を行うための方法については、161 ページの『統合コンポーネント・ライブラリー内のコンポーネントの処理』を参照してください。
- ユーザー・プロジェクト内にコンポーネントへのショートカットを作成する。この作業の詳細については、156 ページの『ユーザー・プロジェクトでの作業』を参照してください。
- 統合ブローカーにコンポーネントを配置する。詳細については、99 ページの『メッセージ・ブローカー・ワークスペースへの配置』を参照してください。
- コンポーネントをサーバーや他のライブラリーにインポートしたり、開発をバックアップするために、コンポーネントをパッケージにエクスポートする。詳細については、168 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』を参照してください。

統合コンポーネント・ライブラリーの作成

ウィザードを使用して System Manager 内に新規統合コンポーネント・ライブラリーを作成するには、次の手順を実行します。

1. 次のいずれかを実行して、「新規統合コンポーネント・ライブラリー」ウィザードを始動します。
 - メニュー・バーから、「ファイル」 > 「新規」 > 「統合コンポーネント・ライブラリー」を選択します。
 - WebSphere Business Integration System Manager ビューで、統合コンポーネント・ライブラリー・フォルダーを右マウス・ボタン・クリックし、コンテキスト・メニューから「新規統合コンポーネント・ライブラリー」を選択します。
 - ツールバーの「新規ウィザードを開く (Open The New Wizard)」ボタンをクリックし、メニューから「新規統合コンポーネント・ライブラリー」を選択します。

155 ページの図 29 に、「新規統合コンポーネント・ライブラリー」ウィザードを示します。

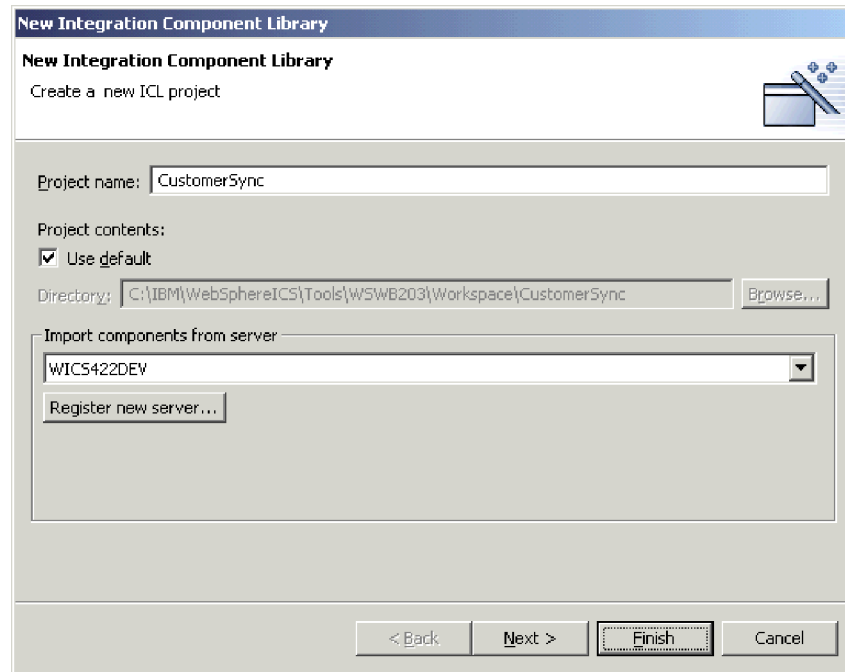


図 29. 統合コンポーネント・ライブラリーの作成

2. 「プロジェクト名」フィールドに、統合コンポーネント・ライブラリーの名前を入力します。

プロジェクト名には英数字および下線のみが使用でき、英語で指定する必要があります。

ライブラリーに名前を付ける際には、ライブラリーを対応するサーバーに関連付けるようにすることをお勧めします。例えば、サーバー名が SERV420DEV である場合は、ライブラリーに SERV420DEVICL と名前を付けます。

3. ライブラリーのフォルダーをデフォルトの場所 (ワークスペース) に作成し、ライブラリーに指定した名前と同じ名前を付ける場合は、「デフォルトを使用」チェック・ボックスをチェックしたままにします。

ライブラリー・フォルダーの名前と場所を指定する場合は、次の手順を実行します。

- a. 「デフォルトを使用」チェック・ボックスをクリアします。
- b. 「ディレクトリー」フィールドに、ライブラリー用に使用するディレクトリーの絶対パスと名前を入力するか、「参照」をクリックして既存のディレクトリーを選択します。

注: ワークスペースのパスにライブラリー用のフォルダーを作成するには、この「デフォルトを使用」チェック・ボックスを使用して System Manager に作成させる以外に方法はありません。

4. 「終了」をクリックして、ウィザードを終了します。

System Manager は、統合コンポーネント・ライブラリー・フォルダーの下にユーザーが指定した名前のフォルダーを作成します。

ユーザー・プロジェクトでの作業

ユーザー・プロジェクトには、1 つ以上のライブラリー内で作業を行う統合コンポーネントへのショートカットを作成します。ユーザー・プロジェクトを使用すると、コンポーネントのビューをインターフェースとして編成できます。System Manager からサーバーにコンポーネントを配置するには、ユーザー・プロジェクトにコンポーネントのショートカットを追加する必要があります。

このセクションの内容は次のとおりです。

- 『ユーザー・プロジェクト用に統合ブローカー設定を構成する』
- 『ユーザー・プロジェクトの作成』
- 158 ページの『ユーザー・プロジェクトへのショートカットの追加』
- 165 ページの『ソリューションのエクスポート』

ユーザー・プロジェクト用に統合ブローカー設定を構成する

System Manager を使用して、複数の統合ブローカー用にユーザー・プロジェクトを作成することができます。さまざまな統合ブローカーに対して System Manager を使用可能にするには、次の手順を実行します。

1. テキスト・エディターで、`ProductDir\bin` の `cwtools.cfg` というファイルを開きます。
2. System Manager を WebSphere Application Server プロジェクト用に使用可能にするには、`WAS_PROJECT` セクションの `Installed` プロパティの値を `true` に設定します。

System Manager を WebSphere MQ Integrator Broker プロジェクト用に使用可能にするには、`WMQI_PROJECT` セクションの `Installed` プロパティの値を `true` に設定します。

System Manager を WebSphere InterChange Server プロジェクト用に使用可能にするには、`ICS_PROJECT` セクションの `Installed` プロパティの値を `true` に設定します。

3. ファイルを保管して閉じます。

ユーザー・プロジェクトの作成

注: 以下は、ユーザー・プロジェクトの作成方法の一般的な説明です。メッセージ・ブローカー・プロジェクトの新規作成についての具体的な説明は、94 ページの『メッセージ・ブローカー・プロジェクトの作成』を参照してください。

System Manager でウィザードを使用して新規ユーザー・プロジェクトを作成するには、以下の手順を実行します。

1. 以下のいずれかを実行し、「新規ユーザー・プロジェクト」ウィザードを開始します。
 - メニュー・バーから「ファイル」>「新規」>「ユーザー・プロジェクト」を選択する。
 - WebSphere Business Integration System Manager ビューで、「ユーザー・プロジェクト」フォルダーを右マウス・ボタンでクリックし、次に「新規ユーザ

「ユーザー・プロジェクト」を選択し、コンテキスト・メニューから「メッセージ・ブローカー・プロジェクト (Message Broker Projects)」を選択する。

- WebSphere Business Integration System Manager ビューで、「ユーザー・プロジェクト」フォルダーを展開し、次に「メッセージ・ブローカー・プロジェクト (Message broker Projects)」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規メッセージ・ブローカー・プロジェクト (New Message broker project)」を選択する。
 - ツールバーの「新規ウィザードを開く (Open The New Wizard)」ボタンをクリックし、メニューから「新規ユーザー・プロジェクト」を選択する。
2. 「プロジェクト名」フィールドにユーザー・プロジェクト名を入力します。

プロジェクト名には英数字および下線のみが使用でき、英語で指定する必要があります。

このタイプのユーザー・プロジェクトには、対応するサーバーに関連する方法で名前を付けることをお勧めします。例えば、サーバー名が SERV420DEV である場合は、ユーザー・プロジェクトに SERV420DEVUP と名前を付けます。

3. デフォルトのロケーション (ワークスペース) にユーザー・プロジェクト用フォルダーを作成し、ユーザー・プロジェクトに指定した名前と同じ名前を付けるには、「プロジェクト内容」ペインの「デフォルトの使用」チェック・ボックスを有効にしておきます。

ユーザー・プロジェクト・フォルダーの名前およびロケーションを指定する場合は、以下の手順を実行します。

- a. 「プロジェクト内容」ペインの「デフォルトの使用」チェック・ボックスをクリアします。
- b. ユーザー・プロジェクトに使用するディレクトリーの絶対パスおよび名前を「ディレクトリー」フィールドに入力するか、「ブラウズ」をクリックして既存のディレクトリーを選択します。

注: 「デフォルトを使用」チェック・ボックスを使用して System Manager で作成する以外に、ワークスペースのパスにユーザー・プロジェクト用フォルダーを作成する方法はありません。

4. この時点で既存の統合コンポーネントへのショートカットを作成しない場合は、ステップ 5 (158 ページ) に進みます。

既存の統合コンポーネントへのショートカットを作成する場合は、統合コンポーネント・ライブラリーの横にあるチェック・ボックスを有効にしてその中にあるすべてのコンポーネントへのショートカットを作成するか、統合コンポーネント・ライブラリー・フォルダーを展開してコンポーネント・グループの横にあるチェック・ボックスを有効にするか、グループのフォルダーを展開して個々のコンポーネントのチェック・ボックスを有効にします。

注: 複数の統合コンポーネント・ライブラリーから同じ名前のコンポーネントを選択する場合は、選択内容に重複する参照が存在することを通知するプロンプトは出されません。重複するコンポーネントを選択した場合は、ウィザードで選択したときにライブラリーの最下部にあった統合コンポーネント・ライブラリーのコンポーネントに対してショートカットが作成されます。

図 30 に、「新規ユーザー・プロジェクト」ウィザードを示します。

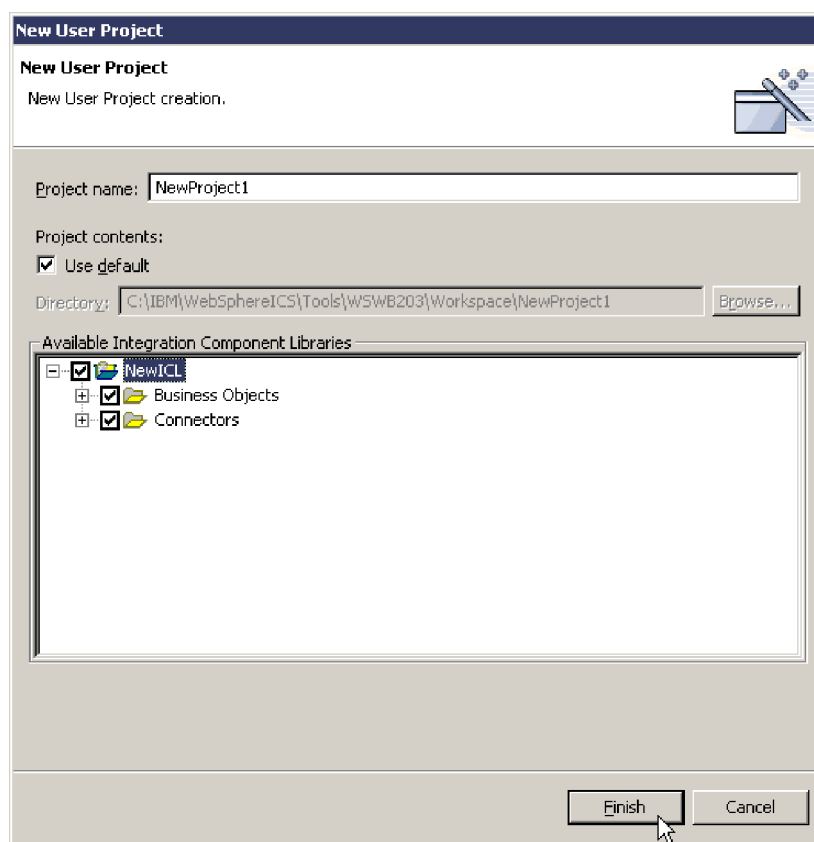


図 30. ユーザー・プロジェクトの作成

5. 「終了」をクリックして、ウィザードを終了します。

System Manager は、「ユーザー・プロジェクト」フォルダー内の統合ブローカーに対応するフォルダー (メッセージ・ブローカー・プロジェクト (Message broker Projects)) に、ユーザーが指定した名前のフォルダーを作成します。

ユーザー・プロジェクトへのショートカットの追加

作業しているインターフェースを表示できるようにするには、ユーザー・プロジェクトにショートカットを追加します。ユーザー・プロジェクトにショートカットを追加する方法はいくつかあります。以下のセクションではこれらの方法について説明します。

- 159 ページの『依存関係ツリーの使用』
- 160 ページの『「プロジェクトを更新」ウィザードの使用』
- 161 ページの『コンポーネントのドラッグ・アンド・ドロップ』
- 167 ページの『ソリューションのインポート』
- 167 ページの『ソリューションのインポート』

依存関係ツリーの使用

「依存関係ツリー」ウィザードは、コンポーネント・ショートカットをユーザー・プロジェクトに追加するための最も便利なインターフェースです。ユーザー・プロジェクトは、主にインターフェースを表す目的で設計されており、インターフェースは一般にコネクタ・オブジェクトを中心に置きます。さらに、ユーザー・プロジェクトは一般にコネクタ・オブジェクトを中心に置きます。通常、ユーザー・プロジェクトに必要なショートカットを作成するには、コネクタ・オブジェクトの依存関係を見つけます。

依存関係の概念については、170 ページの『依存関係および参照』を参照してください。

「依存関係ツリー」ウィザードを使用してユーザー・プロジェクトにショートカットを追加するには、以下の手順を実行します。

1. ライブラリーにあるコネクタ・オブジェクトなどの統合コンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「**依存関係を表示**」を選択します。
2. 「**プロジェクトに追加**」ドロップダウン・メニューから、ショートカットの追加先ユーザー・プロジェクトを選択します。
3. ウィザードの左側のペインで、ショートカットを作成するコンポーネントを選択します。

キーボード・ショートカットを使用すると作業しやすくなります。例えば、オブジェクトの範囲を選択するには **Shift** を押しながら操作し、不連続な単一オブジェクトを選択するには **Ctrl** を押しながら操作します。

4. 右向きの矢印をクリックし、ウィザードの右側のペインにコンポーネントを追加します。

160 ページの図 31 に、「依存関係ツリー」ウィザードを示します。

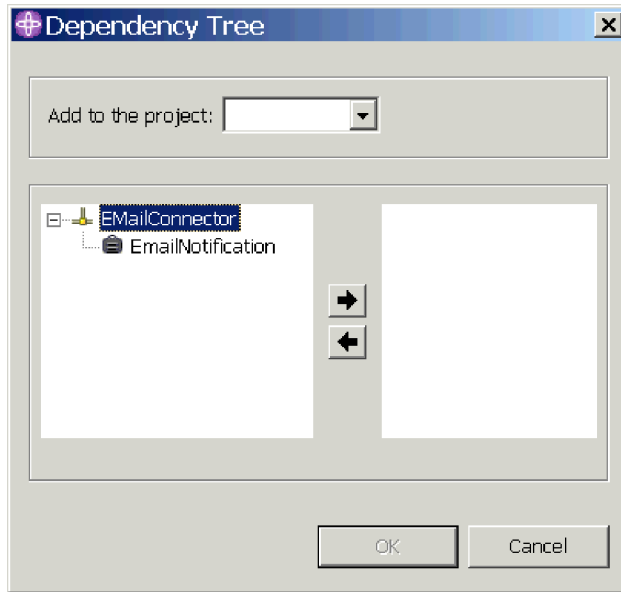


図 31. 「依存関係ツリー」を使用したユーザー・プロジェクトへのショートカットの追加

5. 「OK」をクリックする。

System Manager は、指定のユーザー・プロジェクトに選択されたコンポーネントへのショートカットを作成します。

「プロジェクトを更新」ウィザードの使用

「プロジェクトを更新」ウィザードが提供するインターフェースにより、最初にユーザー・プロジェクトを作成するときに表示されたユーザー・プロジェクトに類似したユーザー・プロジェクトへのショートカットを追加できます。「プロジェクトを更新」ウィザードを使用するには、以下の手順を実行します。

1. WebSphere Business Integration System Manager ビューで任意のユーザー・プロジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロジェクトを更新」を選択します。
2. 統合コンポーネント・ライブラリーの横にあるチェック・ボックスを有効にしてその中にあるすべてのコンポーネントへのショートカットを作成するか、統合コンポーネント・ライブラリー・フォルダーを展開してコンポーネント・グループの横にあるチェック・ボックスを有効にするか、グループのフォルダーを展開して個々のコンポーネントのチェック・ボックスを有効にします。
3. 「終了」をクリックします。

ウィザードで選択したコンポーネントと同じ名前のコンポーネントへのショートカットがユーザー・プロジェクトに存在する場合は、プロンプトが表示され、以下の操作が可能になります。

- 表示されたコンポーネントを上書きする。
- 重複するコンポーネントをすべて上書きする。
- コンポーネントを上書きしない。
- 更新操作をキャンセルする。

ユーザー・プロジェクトにショートカットが存在するコンポーネントと同じ名前のコンポーネントを選択しなかった場合は、プロジェクトにショートカットが追加され、ウィザードが終了します。

コンポーネントのドラッグ・アンド・ドロップ

統合コンポーネント・ライブラリー・フォルダーからコンポーネントを選択し、ユーザー・プロジェクトにドラッグ・アンド・ドロップすると、そのユーザー・プロジェクトにコンポーネントへのショートカットを追加できます。

ユーザー・プロジェクトのフォルダーにまだショートカットが存在しない場合は、フォルダー自体にコンポーネントをドラッグ・アンド・ドロップしてください。フォルダーにコンポーネントをドラッグ・アンド・ドロップし、マウス・ポインターの下に四角形が表示されたらマウス・ボタンを放します。

ユーザー・プロジェクトのフォルダーにすでにショートカットが存在する場合は、フォルダー自体にコンポーネントをドラッグ・アンド・ドロップすることはできません。線が表示されるまでフォルダーの既存のショートカットの間にコンポーネントをドラッグ・アンド・ドロップし、マウス・ボタンを放します。

統合コンポーネント・ライブラリー内のコンポーネントの処理

WebSphere Business Integration システムを実装する場合、大半の時間が統合コンポーネントの処理に費やされます。本書では個々のコンポーネントの作成方法の詳細については説明しません。このセクションでは、Designer ツールの起動方法、新規コンポーネントの作成方法、既存のコンポーネントの変更方法、および System Manager で作成する一部のコンポーネントの処理方法について説明します。

統合コンポーネントの作成方法について詳しくは、以下のマニュアルを参照してください。

- ビジネス・オブジェクト開発ガイド
- コネクター開発ガイド (Java 用または C++ 用)

Designer ツールの起動

このセクションでは、各 Designer ツールの起動方法について説明します。Designer ツールを使用すると、新規コンポーネントの作成や、既存のコンポーネントのオープンおよび変更を行うことができます。

注: デザイナー・ツールのどれかを起動しようとしたときに「クラスが見つかりません」という内容のエラーが発生した場合は、System Manager を起動してから、デザイナー・ツールの起動を再試行する必要があります。ただし、いったんツールが起動してしまえば、System Manager を稼働状態のままにしておく必要はありません。

Business Object Designer

Business Object Designer を起動するには、以下のいずれかを実行します。

- WebSphere Business Integration システム・ビューで「ビジネス・オブジェクト」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規ビジネス・オブジェクトの作成」を選択する。

- WebSphere Business Integration System Manager ビューで任意のフォルダーを選択し、以下のいずれかを実行する。
 - メニュー・バーから「ツール」>「**Business Object Designer**」を選択する。
 - 「**Business Object Designer**」 ツールバー・ボタンをクリックする。
 - キーボード・ショートカット **Ctrl+4** を使用する。
- 「スタート」>「プログラム」>「**IBM WebSphere Business Integration Adapters**」>「ツール」>「**Business Object Designer**」を選択する。

Business Object Designer の詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

Connector Configurator

Connector Configurator を起動するには、以下のいずれかの操作を実行します。

- WebSphere Business Integration システム・ビューで「コネクター」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「**新規コネクターの作成**」を選択する。
- WebSphere Business Integration System Manager ビューで任意のフォルダーを選択し、以下のいずれかを実行する。
 - メニュー・バーから「ツール」>「**Connector Configurator**」を選択する。
 - 「**Connector Configurator**」 ツールバー・ボタンをクリックする。
 - キーボード・ショートカット **Ctrl+1** を使用する。
- 「スタート」>「プログラム」>「**IBM WebSphere Business Integration Adapters**」>「ツール」>「**Connector Configurator**」を選択する。

Connector Configurator について詳しくは、いずれかの WebSphere Business Integration Adapters のユーザーズ・ガイドまたは コネクター開発ガイド (*Java* 用または *C++* 用) を参照してください。

新規コンポーネントの作成

161 ページの『Designer ツールの起動』で説明している各 Designer ツールを起動すると、新規コンポーネントを作成できます。例えば、Business Object Designer を起動すると新規ビジネス・オブジェクトが作成され、Connector Configurator を起動すると新規コネクターが作成されます。詳細については、これらの各ツールのマニュアルを参照してください。

既存のコンポーネントの変更

- ライブラリーのコンポーネントまたはユーザー・プロジェクトのコンポーネントへのショートカットをダブルクリックします。
- ライブラリーのコンポーネントまたはユーザー・プロジェクトのコンポーネントへのショートカットを選択し、以下のいずれかの操作を実行します。
 - 対応する Designer ツールを起動する。詳細については、161 ページの『Designer ツールの起動』を参照してください。
 - **Enter** を押す。
 - **Ctrl+E** を押す。
 - メニュー・バーから「コンポーネント」>「**定義の編集**」を選択する。

- ライブラリーのコンポーネントまたはユーザー・プロジェクトのコンポーネントへのショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「定義の編集」を選択します。
- コンポーネントの Designer ツールを起動し（161 ページの『Designer ツールの起動』を参照）、ツールが起動したらコンポーネントを開きます。

パッケージからライブラリーへのコンポーネントのインポート

168 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』の説明に従い、.jar ファイル・パッケージに統合コンポーネントをエクスポートできます。これにより、コンポーネントの環境間での移行、他の開発者との共用、およびテクニカル・サポートへのサブミットが容易になります。

パッケージから統合コンポーネント・ライブラリーにコンポーネントをインポートするには、以下の手順を実行します。

重要: インポートするパッケージのコンポーネントと同じ名前のコンポーネントが存在する場合、System Manager は警告を出さずに既存のコンポーネントを上書きします。

1. 統合コンポーネント・ライブラリーを右マウス・ボタンでクリックし、コンテキスト・メニューから「リポジトリ・ファイルからインポート (**Import from Repository File**)」を選択します。

System Manager により、「リポジトリ・ファイルのインポート」ウィザードが表示されます（164 ページの図 32 を参照）。

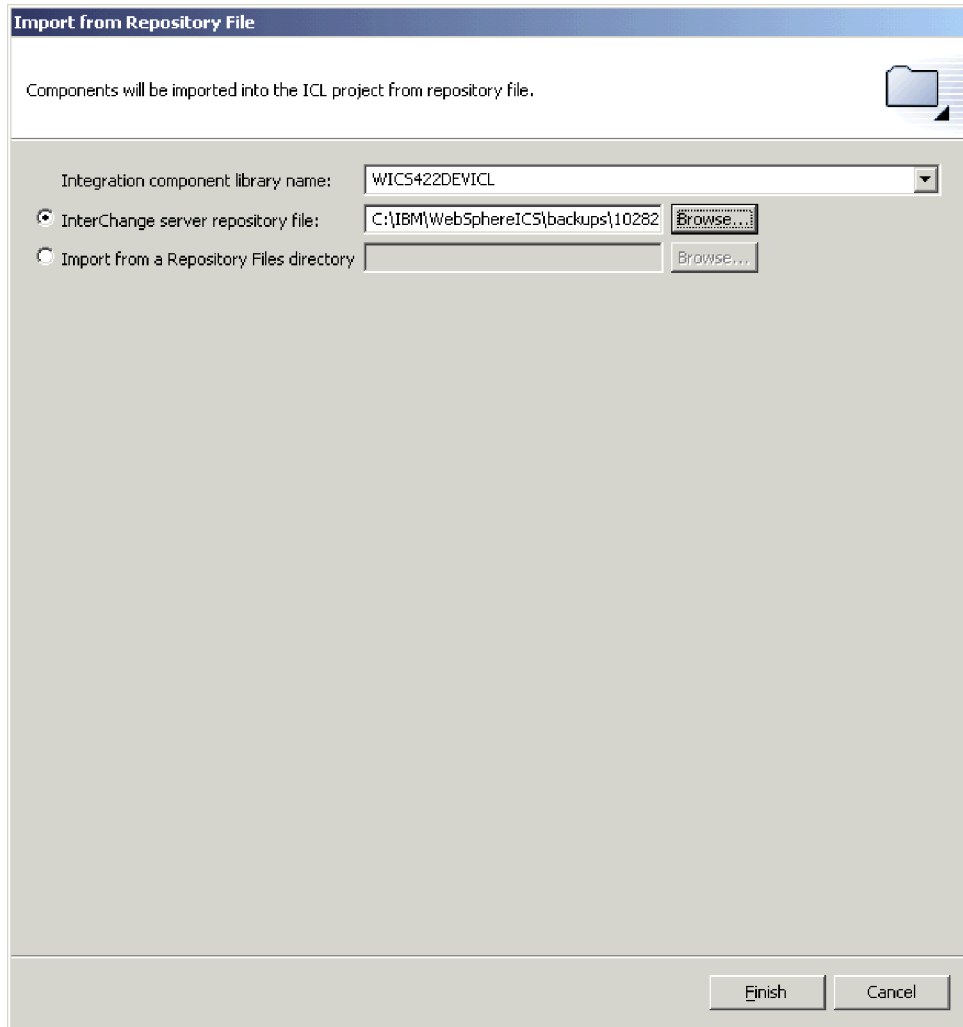


図 32. パッケージのインポート

2. 「リポジトリ・ファイルのインポート」画面で、「統合コンポーネント・ライブラリー名」ドロップダウン・メニューに、コンポーネントのインポート先のライブラリー名が含まれていることを確認します。

コンポーネントのインポート先のライブラリー以外のライブラリーから「リポジトリ・ファイルからインポート (Import from Repository File)」ウィザードを起動した場合は、ウィザードを閉じて再起動しなくても、この方法でインポート先を変更できます。

3. 以下のいずれかを実行して、インポートするコンポーネントを指定します。
 - 単一のパッケージ・ファイルをインポートするには、インポートする .jar ファイルの絶対パスおよび名前を「**統合リポジトリ・ファイル (Integration repository file)**」フィールドに入力するか、「**ブラウズ**」をクリックしてファイルを選択します。
 - パッケージ・ファイルのディレクトリー全体をインポートするには、「**リポジトリ・ファイルからインポートするディレクトリー (Import from a Repository Files directory)**」フィールドにディレクトリーの絶対パスを入力するか、「**ブラウズ**」をクリックしてファイルを選択します。

4. 「終了」をクリックします。

注: パッケージ・ファイルをインポートするには、ワークベンチの「ファイル」>「インポート」メニュー項目は使用しないでください。「Zip ファイル」ウィザードは .jar 拡張子を持つアーカイブを処理でき、WebSphere Business Integration Adapter パッケージ・ファイルには .jar 拡張子が付いていますが、「Zip ファイル」ウィザードはパッケージ・ファイルを適切に処理しません。

ソリューションの処理

ユーザー・プロジェクトはソリューションとしてエクスポートできます。このアクションでは、ショートカットをユーザー・プロジェクトからコピーする他、統合コンポーネント・ライブラリーでショートカットが参照するコンポーネント定義もコピーします。これにより、ある環境から別の環境に全インターフェースまたはビジネス・インテグレーション・システムを容易に移行できます。

ソリューションのエクスポート

参照するユーザー・プロジェクトおよび統合コンポーネントをソリューションとしてエクスポートするには、以下の手順を実行します。

1. WebSphere Business Integration System Manager ビューで、「ユーザー・プロジェクト」フォルダーを展開し、次に「メッセージ・ブローカー (Message Brokers)」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「ソリューションのエクスポート」を選択します。

System Manager により、「ソリューションのエクスポート」ウィザードが表示されます (166 ページの図 33 を参照)。

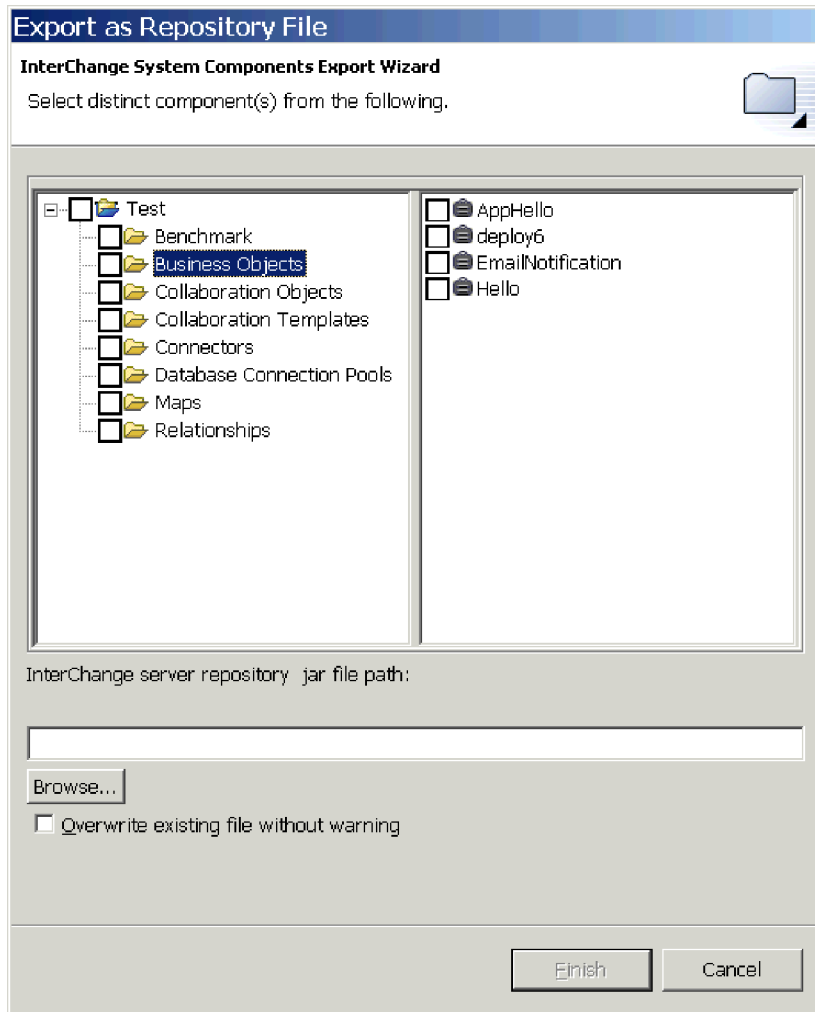


図33. ソリューションのエクスポート

2. 以下の方法で、エクスポートするコンポーネントを選択します。
 - ユーザー・プロジェクトの横のチェック・ボックスを有効にして、プロジェクト内のすべてのコンポーネントを選択します。
 - コンポーネント・グループの横のチェック・ボックスを有効にして、グループ内のすべてのコンポーネントを選択します。
 - コンポーネント・グループを強調表示し、右側のペインで個々のコンポーネントの横にあるチェック・ボックスを有効にして、それらのコンポーネントを選択します。
3. ウィザード画面の下部にあるテキスト・フィールドにソリューションのエクスポート先ディレクトリーの絶対パスおよび名前を入力するか、「ブラウズ」をクリックして所定のディレクトリーにナビゲートします。
4. 「終了」をクリックします。

System Manager が以下の処理を実行し、ステップ 3 で指定されたディレクトリーにソリューションをエクスポートします。

- ソリューションのエクスポート時に選択したユーザー・プロジェクトに、ショートカットを含む User ディレクトリーを作成します。

- ソリューションのエクスポート時に選択したユーザー・プロジェクトに、ショートカットによって参照される統合コンポーネント・ライブラリーのディレクトリーを含む System ディレクトリーを作成します。
5. エクスポート処理が正常に完了したことを示すプロンプトが出されたら、「OK」をクリックします。

ソリューションのインポート

ソリューションをインポートするには、以下の手順を実行します。

1. WebSphere Business Integration System Manager ビューで、「ユーザー・プロジェクト」フォルダーを展開し、次に「メッセージ・ブローカー (Message Brokers)」を右マウス・ボタンでクリックし、コンテキスト・メニューから「ソリューションのインポート」を選択します。

System Manager により、「ソリューションのインポート」ウィザードが表示されます (図 34 を参照)。

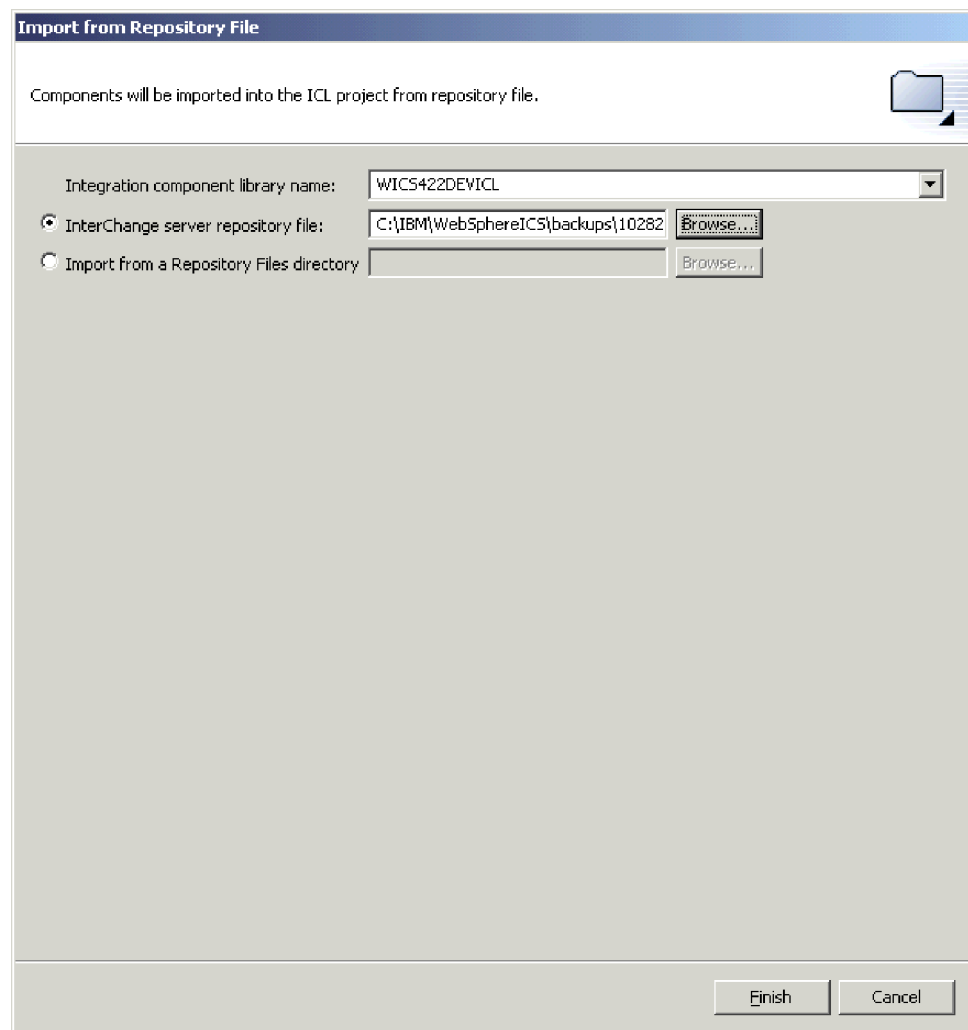


図 34. ソリューションのインポート

2. エクスポートしたソリューションが存在するディレクトリーの絶対パスおよび名前を「ソリューション・ディレクトリー名 (Solution directory name)」フィールドに入力するか、「ブラウズ」をクリックして所定のディレクトリーにナビゲートします。
3. 「終了」をクリックします。

System Manager が、エクスポートされたソリューションで定義されている統合コンポーネント・ライブラリーおよびユーザー・プロジェクトを環境に作成します。

System Manager を使用したパッケージへのコンポーネントのエクスポート

統合コンポーネントはパッケージ・ファイルにエクスポートできます。統合コンポーネントはリソースであり、最終的には、ファイル・システムに格納されるファイルとなります。詳細については、149 ページの『リソース』を参照してください。System Manager は、コンポーネントをパッケージにエクスポートするときに以下のリソースを .jar (Java アーカイブ) ファイルに圧縮します。

- 定義ファイル (コンポーネント・タイプに応じた拡張子を付け、XML 形式で格納します)
- マップおよびコラボレーション・テンプレートの Java ソース・ファイル
- メッセージ・ファイル

コンポーネントをパッケージにエクスポートするには、以下の手順を実行します。

1. 統合コンポーネント・ライブラリー、またはエクスポートするコンポーネントが含まれるユーザー・プロジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「リポジトリー・ファイルとしてエクスポート (Export as Repository File)」を選択します。

System Manager により、「リポジトリー・ファイルのエクスポート」ウィザードが表示されます (169 ページの図 35 を参照)。

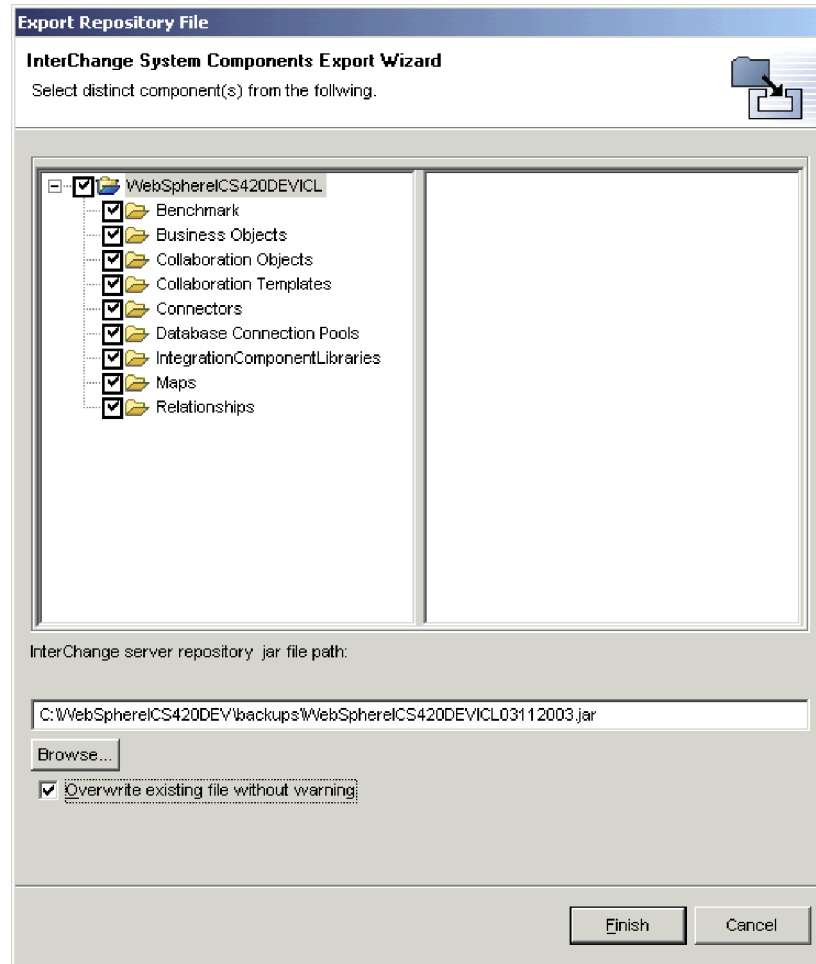


図 35. パッケージのエクスポート

2. 以下の方法で、エクスポートするコンポーネントを選択します。
 - 統合コンポーネント・ライブラリーまたはユーザー・プロジェクトの横のチェック・ボックスを有効にして、ライブラリーまたはプロジェクト内のすべてのコンポーネントを選択します。
 - コンポーネント・グループの横のチェック・ボックスを有効にして、グループ内のすべてのコンポーネントを選択します。
 - コンポーネント・グループを強調表示し、右側のペインで個々のコンポーネントの横にあるチェック・ボックスを有効にして、それらのコンポーネントを選択します。
3. コンポーネントのエクスポート先として既存の .jar ファイルを指定し、プロンプトを受け取ることなくファイルを上書きする場合は、「警告せずに既存のファイルを上書き」チェック・ボックスを有効にします。

注: 「警告せずに既存のファイルを上書き」チェック・ボックスを機能させるには、使用するファイルを指定する前に、このチェック・ボックスを有効にする必要があります (ステップ 4 (170 ページ) を参照)。System Manager は、既存のファイルが指定されていることを検出すると直ちにファイルを上書きするようプロンプトを出し、ウィザードが終了するまで待機しません。このため、このオプションはあらかじめ有効にしておく必要があります。

- コンポーネントのエクスポート先の `.jar` ファイルの名前およびパスを「リポジトリ `jar` ファイル・パス (Repository jar file path)」フィールドに入力するか、「ブラウズ」をクリックして上書きするファイルを選択するか、またはディレクトリーをナビゲートしてファイル名を指定します。

既存ファイルの名前およびパスを指定するときに「警告せずに既存ファイルを上書き」チェック・ボックスを有効にしなかった場合、既存のファイルを上書きするには、プロンプトが表示されたときに「はい」をクリックします。

注: ファイルの名前およびパスをフィールドに入力する場合は、`.jar` 拡張子を含めなければ「終了」ボタンは使用可能になりません。

- 「終了」をクリックして、ウィザードを終了します。

依存関係および参照

統合コンポーネントは相互に依存し合ってビジネス・インテグレーション・システムでの役割を果たします。例えば、ビジネス・オブジェクト定義には他のビジネス・オブジェクト定義を子として含めることができ、コネクターのマップはサポートされるビジネス・オブジェクトに関連付けられます。システムを正常に機能させるには、これらの依存関係が満たされている必要があります。

コンポーネント間の関係について説明するときには、状況に応じて**依存関係**および**参照**という用語を使用します。例えば、コネクター定義が統合ブローカーとデータを交換するには、サポートされるビジネス・オブジェクト定義が存在している必要があります。この場合には、ビジネス・オブジェクト定義はコネクター定義の**依存関係**です。ビジネス・オブジェクト定義とコネクター定義の間に同じ関係がある場合でも、ビジネス・オブジェクトの文脈では、コネクターはビジネス・オブジェクト定義の**参照**の 1 つであり、そのビジネス・オブジェクトを子として含む他のビジネス・オブジェクト、そのビジネス・オブジェクトを変換するマップ、ポート定義用にビジネス・オブジェクトをサポートするコラボレーション・テンプレートなどを持ちます。

表 26 に、依存関係および参照となるコンポーネントをコンポーネント・タイプごとに示します。

表 26. 統合コンポーネントの依存関係および参照

コンポーネント	依存関係	参照
ビジネス・オブジェクト	<ul style="list-style-type: none"> ビジネス・オブジェクト 	<ul style="list-style-type: none"> ビジネス・オブジェクト マップ コネクター コラボレーション・テンプレート コラボレーション・オブジェクト
コネクター	<ul style="list-style-type: none"> ビジネス・オブジェクト マップ 	<ul style="list-style-type: none"> コラボレーション・オブジェクト

表 26. 統合コンポーネントの依存関係および参照 (続き)

コンポーネント	依存関係	参照
マップ	<ul style="list-style-type: none"> ビジネス・オブジェクト マップ 	<ul style="list-style-type: none"> コネクタ コラボレーション・オブジェクト
コラボレーション・テンプレート	<ul style="list-style-type: none"> ビジネス・オブジェクト 	なし
コラボレーション・オブジェクト	<ul style="list-style-type: none"> ビジネス・オブジェクト マップ コネクタ コラボレーション・テンプレート コラボレーション・オブジェクト 	<ul style="list-style-type: none"> コラボレーション・オブジェクト

依存関係および参照の表示

System Manager を使用すると、統合コンポーネントの依存関係および参照を表示することができます。

コンポーネントの依存関係を表示するには、System Manager でコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「依存関係を表示」を選択します。「依存関係ツリー」ウィザードが表示されます。詳細については、159 ページの『依存関係ツリーの使用』を参照してください。

コンポーネントの参照を表示するには、System Manager でコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「参照を表示」を選択します。「オブジェクト参照」ウィンドウが表示されます。

複数のワークベンチ・リソースで使用可能な標準の操作

ワークベンチで行う多くの作業は処理している特定のリソースまたは作業している状況に依存しますが、すべてのリソースに同じように影響を及ぼす多くの操作も用意されています。このセクションでは、ワークベンチで実行可能な作業のうち、すべてのリソースに対して同様に機能するものについて説明します。

ソース・コード制御からワークスペースへのプロジェクトの追加

Rational ClearCase などのソース・コード制御システムから統合コンポーネント・ライブラリおよびユーザー・プロジェクトをワークスペースに追加できます。

追加方法については、ソース・コード制御システム・プラグインの資料を参照してください。

ClearCase を使用したこの追加方法の例については、「*WebSphere Interchange Server インプリメンテーション・ガイド*」を参照してください。

リソースの切り取り、コピー、および貼り付け

リソースの切り取り、コピー、および貼り付けは、System Manager とファイル・システムの両方で可能です。

System Manager でユーザー・プロジェクト、統合コンポーネント・ライブラリー、統合コンポーネント、ショートカット、またはフォルダーの切り取り、コピー、または貼り付けを行うには、リソースを右マウス・ボタンでクリックし、所定のメニュー項目を選択します。統合コンポーネントをコピーする場合は、同じライブラリーに貼り付ける (例えば、ビジネス・オブジェクト定義をコピーして同じライブラリーに貼り付け、別の名前を指定してテンプレートとして使用する) ことはできません。ただし、Designer ツールでコンポーネント定義を開き、「別名保管」操作を実行して新しい名前と同じライブラリーに保管することはできます。

ファイル・システムで統合コンポーネントまたはショートカットの切り取り、コピー、および貼り付けを行うには、Windows のエクスプローラーを起動し、プロジェクト・ディレクトリーの該当するサブディレクトリーにナビゲートし、コンポーネントの名前を共用するファイルをコピーし、宛先プロジェクト・ディレクトリー内の適切なサブディレクトリーに貼り付けます。

ファイル・システムで切り取り、コピー、および貼り付け操作を実行する場合は、必ず System Manager で統合コンポーネント・ライブラリーまたはユーザー・プロジェクトを最新表示させ、新しく追加したリソースが表示されるようにしてください。詳細については、『リソースの最新表示』を参照してください。

存在するユーザー・プロジェクトおよびライブラリーを指定するメタデータ参照がワークベンチで管理されているため、ユーザー・プロジェクト全体または統合コンポーネント・ライブラリー全体の切り取り、コピー、および貼り付けを行うことはできません。ワークスペース・ディレクトリーにフォルダーをコピーしても、メタデータ参照は更新されません。ただし、メタデータ・エントリーを満たすように System Manager で新規ユーザー・プロジェクトまたはライブラリーを作成してから、新規ライブラリーまたはユーザー・プロジェクトに対応するディレクトリーにコンポーネント定義のフォルダーを貼り付けることはできます。

リソースの最新表示

ファイル・システムでのファイルの切り取り、コピー、および貼り付けによってコンポーネント定義をライブラリーに追加したりショートカットをユーザー・プロジェクトに追加したりする場合は、必ず System Manager でライブラリーを最新表示させ、変更内容を反映させてください。

統合コンポーネント・ライブラリーまたはユーザー・プロジェクトを最新表示するには、System Manager でライブラリーまたはユーザー・プロジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「最新表示」を選択します。

リソースの削除

ワークベンチ・リソースを削除するには、以下の手順を実行します。

1. System Manager でリソースを右マウス・ボタンでクリックしてコンテキスト・メニューから「削除」を選択するか、または System Manager でリソースを選択して「削除」キーを押します。

2. 「コンポーネントの削除」ダイアログが表示された場合は、「OK」をクリックします。

注: コンポーネントに依存関係が存在する場合は、そのコンポーネントを削除することはできません。

Eclipse ベースのワークベンチの使用

このセクションでは、ツールを効率的に使用するためにツール・フレームワークで実行する一部の操作について説明します。

パースペクティブのオープンおよびクローズ

このセクションでは、パースペクティブのオープン方法、およびクローズ方法について説明します。

パースペクティブのオープン

ワークベンチでパースペクティブを開くには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから、「ウィンドウ」>「パースペクティブを開く」>「その他」を選択します。
2. 「パースペクティブの選択」ダイアログで開くパースペクティブを選択し、「OK」をクリックします。

System Manager パースペクティブを使用して統合コンポーネントを処理する他に、以下のセクションで説明しているパースペクティブも使用できます。

「リソース」パースペクティブ: このパースペクティブにより、統合コンポーネント・ライブラリーなどのプロジェクト・データを表すメタデータ・ファイルを直接処理できます。174 ページの図 36に、「リソース」パースペクティブが開かれた状態を示します。このパースペクティブでは、ビジネス・オブジェクトを表す .xsd ファイルを「ナビゲーター」ビューから開き、その内容を「エディター」ビューに示しています。

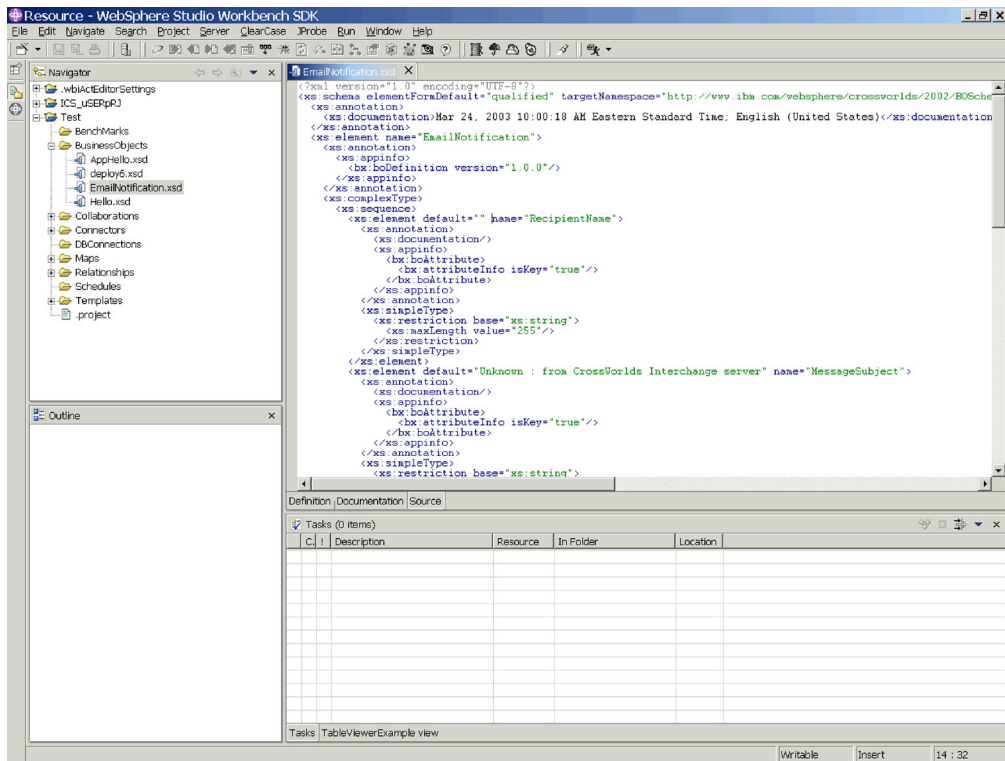


図 36. 「リソース」 パースペクティブ

重要: 「リソース」 パースペクティブから公開されるメタデータ・ファイルは、WebSphere Business Integration コンポーネントを定義します。System Manager パースペクティブには、これらのコンポーネントをインターフェース経由で安全に処理するための方法が用意されています。直接メタデータ・ファイルを操作すると、コンポーネント定義を破壊してしまう危険があります。コンポーネント・メタデータ・ファイルを操作するのは、構造を熟知している場合、または定義のトラブルシューティングを行うためにテクニカル・サポートに連絡し、指示があった場合に限定してください。

Java パースペクティブ: このパースペクティブにより、Java ファイルのオーサリングに役立つエディターおよびビューを使用できます。Java プログラミングの大部分は Designer ツールで実行しますが、データ・ハンドラーなどの外部コンポーネントを記述したり、ユーティリティ・クラスを記述しなければならない場合があります。このような場合に Java パースペクティブは非常に便利です。以下の図に、Java パースペクティブを示します。

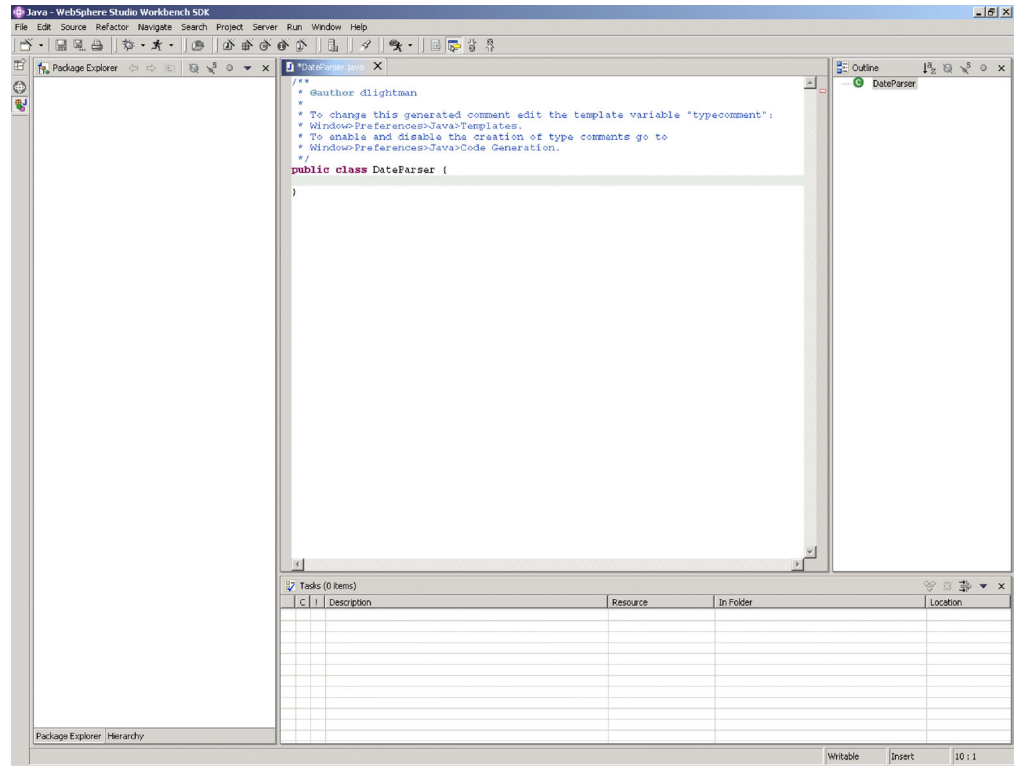


図 37. Java パースペクティブ

パースペクティブのクローズ

パースペクティブを閉じるには、以下の手順を実行します。

- 現在アクティブなパースペクティブを閉じるには、「ウィンドウ」>「パースペクティブを閉じる」を選択します。
- 現在開かれているパースペクティブをすべて閉じるには、「ウィンドウ」>「すべてのパースペクティブを閉じる」を選択します。
- パースペクティブ・ショートカット・バーにあるパースペクティブのアイコンを右マウス・ボタンでクリックし、コンテキスト・メニューから「閉じる」を選択すると、そのパースペクティブが閉じます。
- パースペクティブ・ショートカット・バーにあるパースペクティブのアイコンを右マウス・ボタンでクリックし、コンテキスト・メニューから「すべて閉じる」を選択すると、開いているすべてのパースペクティブが閉じます。

ビューの表示およびクローズ

WebSphere WorkBench パースペクティブおよび WebSphere Studio Application Developer Integration Edition パースペクティブに表示されるペインを制御できます。

ビューの表示

ビューを表示するには、以下の手順を実行します。

1. 「ウィンドウ」>「ビューの表示」>「その他」を選択します。

2. ビュー・グループのフォルダー（「**WebSphere Business Integration Adapter Monitor** のカテゴリ（**WebSphere Business Integration Adapter Monitor Category**）」など）を展開します。
3. 「障害の発生しているキュー・マネージャー」など、特定のビューを選択します。
4. 「OK」をクリックする。

ビューのクローズ

ビューを閉じるには、以下の手順を実行します。

- ビューのタイトル・バーを右マウス・ボタンでクリックし、コンテキスト・メニューから「閉じる」を選択します。
- ビューのタイトル・バーにある「閉じる」ボタンをクリックします。

パースペクティブのカスタマイズ

パースペクティブをカスタマイズし、必要なパースペクティブ、ビュー、ウィザード、およびプラグイン・インターフェースをパースペクティブに組み込むことにより、頻繁に使用するエレメントを開いたり不要なエレメントを閉じたりする回数を最小限に抑えることができます。現在アクティブなパースペクティブをカスタマイズするには、以下の手順を実行します。

1. メニュー・バーから、「ウィンドウ」>「パースペクティブのカスタマイズ」を選択します。
2. カスタマイズするノードをクリックして展開します。
3. ノード・エレメントのチェック・ボックスを使用可能または使用不可に設定します。表 27 に、カスタマイズ可能なパースペクティブ・ノードおよびそのノードを使用可能化した場合の効果を示します。

表 27. カスタマイズ可能なパースペクティブ・ノード

パースペクティブ・ノードのカスタマイズ	結果
「ファイル」>「新規」	「ファイル」>「新規」メニューから項目を追加または除去します。
「ウィンドウ」>「パースペクティブを開く」	「ウィンドウ」>「パースペクティブを開く」メニューからパースペクティブを追加または除去します。
「ウィンドウ」>「ビューの表示」	「ウィンドウ」>「ビューの表示」メニューからビューを追加または除去します。
「その他」	メニュー・バーおよびツールバーから追加または除去します。 例えば、「ClearCase」メニューを表示させるには、「ClearCase」チェック・ボックスを使用可能に設定する必要があります。

パースペクティブの保管

パースペクティブ構成を保管し、カスタマイズ内容を保存できます。パースペクティブを保管するには、以下の手順を実行します。

1. ツール・フレームワークのメニュー・バーから、「ウィンドウ」>「パースペクティブの別保管」を選択します。

2. 「名前」フィールドに、パースペクティブの名前を入力します。
3. 「OK」をクリックする。

デフォルトでのパースペクティブの設定

デフォルトでは、ツール・フレームワークは「リソース」パースペクティブを開きます。主にツール・フレームワークを使用して統合コンポーネントを処理する場合は、System Manager をデフォルトのパースペクティブにすることができます。これを行うには、以下の手順を実行します。

1. ツール・フレームワークのメニュー・バーから、「ウィンドウ」>「設定」を選択します。
2. 「ワークベンチ」ノードを展開します。
3. 「ワークベンチ」ノードの下にある「パースペクティブ」ノードを選択します。
4. 「使用可能なパースペクティブ (Available Perspectives)」リストから「System Manager」を選択します。
5. 「デフォルトにする (Make Default)」をクリックします。
6. 「OK」をクリックする。

System Manager の設定の構成

System Manager の設定を構成するには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから、「ウィンドウ」>「設定」を選択します。
2. 「System Manager の設定 (System Manager Preferences)」を選択し、以下の手順を実行して使用可能な設定オプションを構成します。
 - 統合コンポーネント・ライブラリーからコンポーネントを削除するときにプロンプトが表示されないようにするには、「オブジェクトの削除について確認しない」ペインでコンポーネント・タイプに対応するチェック・ボックスを使用可能に設定します。
 - コピー操作時にコンポーネントとともにコンポーネントの依存関係もコピーする場合は、「ディープ・コピー」チェック・ボックスを使用可能に設定します。

「ディープ・コピー」を使用可能に設定すると、ライブラリー間でビジネス・オブジェクト定義をコピーする場合に、例えば、含んでいる子ビジネス・オブジェクトもすべてコピーされます。しかし、「ディープ・コピー」を使用可能に設定しないと、ライブラリー間でビジネス・オブジェクト定義をコピーする場合に、そのビジネス・オブジェクト自体しかコピーされません。

依存関係については、170 ページの『依存関係および参照』を参照してください。

- ファイルの名前およびパスを「ログ・ファイル」フィールドに入力するか、または「参照」をクリックしてファイルを選択します。System Manager でエラーが発生すると、指定したファイルにエラー情報が書き込まれます。メガバイト単位でログ・ファイルの最大サイズを指定するには、「最大サイズ」フィールドに数値を入力します。
- 設定エレメントをデフォルト値に設定するには、「デフォルトの復元」をクリックします。

図 38 に、System Manager の設定インターフェースを示します。

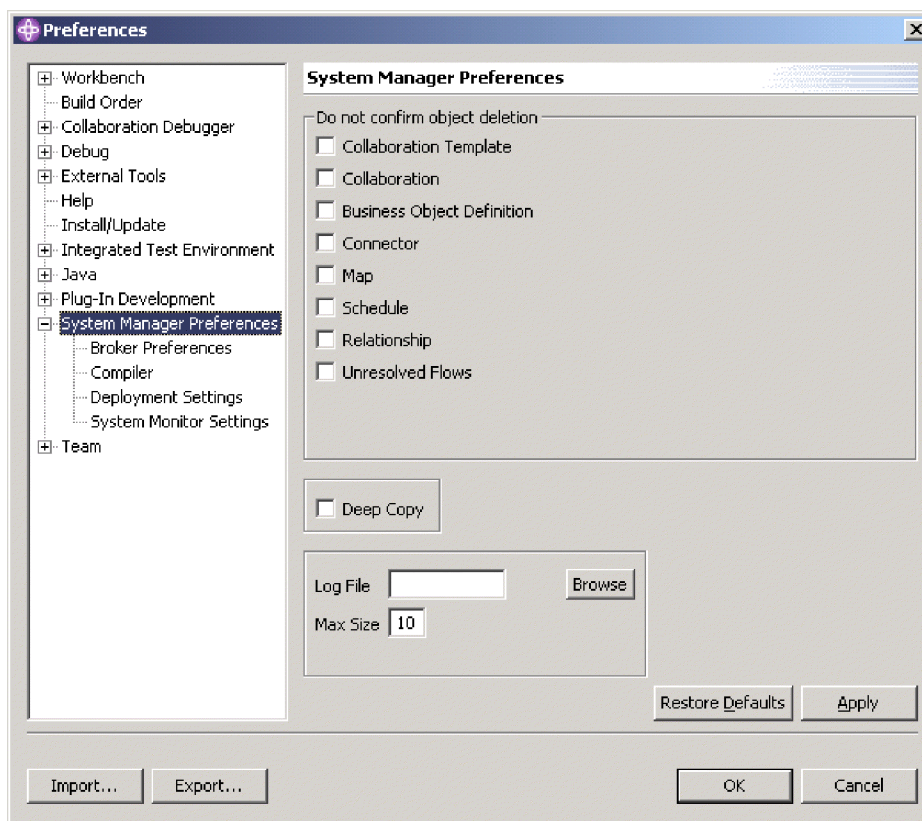


図 38. System Manager の設定

3. 「ブローカーの設定 (Broker Preferences)」インターフェースを使用すると、サポートされる WebSphere Message Broker と連携するように System Manager を構成できます。

ブローカーの設定を行うには、以下の手順を実行します。

- a. 「**System Manager の設定 (System Manager Preferences)**」を展開して、「**ブローカーの設定 (Broker Preferences)**」を選択します。
- b. WebSphere MQ Integrator Broker インポーター・ユーティリティーの絶対パスを「**Integrator Broker インポーター・パスを指定 (Specify the Integrator broker importer path)**」フィールドに入力するか、または「**参照**」をクリックしてディレクトリーを選択します。
- c. WebSphere Business Integration Message Broker インポーター・ユーティリティーの絶対パスを「**Message Broker インポーター・パスを指定 (Specify the Message broker importer path)**」フィールドに入力するか、または「**参照**」をクリックしてディレクトリーを選択します。
- d. WebSphere Business Integration Message Broker の Eclipse ワークスペース・ディレクトリーの絶対パスを「**Message Broker のワークスペース・ディレクトリーを指定 (Specify the Message broker workspace directory)**」フィールドに入力するか、または「**参照**」をクリックしてディレクトリーを選択します。

図 39 は、「ブローカーの設定 (Broker Preferences)」インターフェースを示しています。

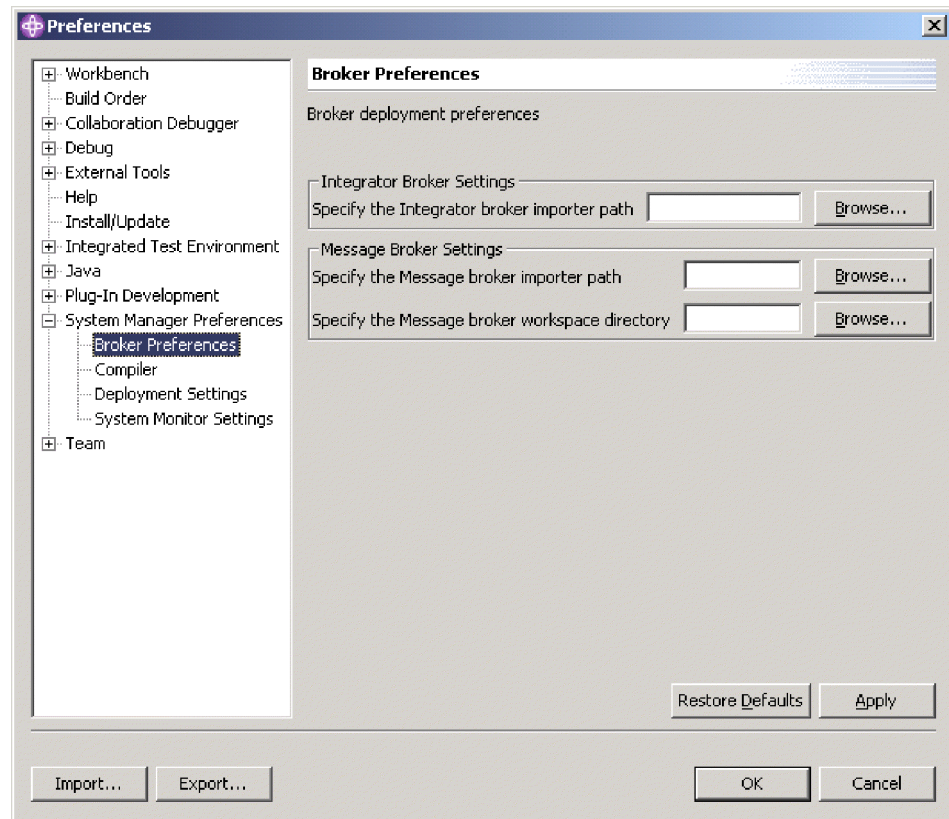


図 39. ブローカーの設定 (Broker Preferences)

注: 「コンパイラ (Compiler)」設定インターフェース、「配置設定 (Deployment Settings)」インターフェース、および「System Monitor の設定 (System Monitor Settings)」インターフェースは、統合ブローカーとして WebSphere Interchange Server を使用する場合に使用します。これらのインターフェースはその他の統合ブローカーには関係しません。

4. 「適用」をクリックして設定を保管し、「設定」ダイアログでの作業を継続するか、または「OK」をクリックして設定を保管し、ダイアログを終了します。

System Manager で統合ブローカーに接続する際の問題のトラブルシューティング

接続に関する問題のトラブルシューティングを行うには、以下の可能性について調査します。

- インポーター・パスが指定されていることを確認してください。
- System Manager が正しいワークスペースに配置されていることを確認してください。

付録 F. コネクタ・スクリプト生成ツールの使用

コネクタ・スクリプト生成ユーティリティは、UNIX プラットフォームで実行されるコネクタ用のコネクタ・スクリプトを作成または変更します。このツールを使用して以下のいずれかを実行します。

- WebSphere Business Integration Adapters インストーラーを使用せずに追加したコネクタに対応する、新規のコネクタ開始スクリプトを生成する。
- 正しい構成ファイルパスを組み込むために、既存のコネクタ開始スクリプトを修正する。

コネクタ・スクリプト生成プログラムを実行するには、以下の手順を実行します。

1. `ProductDir/bin` ディレクトリーにナビゲートする。
2. コマンド `./ConnConfig.sh` を入力する。

コネクタ・スクリプト生成プログラム画面が、図 40 のように表示されます。

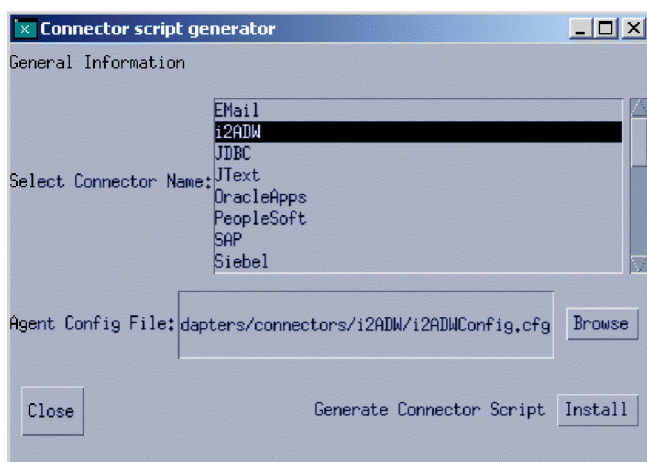


図 40. コネクタ・スクリプト生成プログラム

3. 「コネクタ名を選択」リストから、生成する開始スクリプトに対応するコネクタを選択する。
4. Agent Config File では、コネクタの構成ファイルの絶対パス名を入力するか、「参照」をクリックしてファイルを選択し、その構成ファイルを指定する。
5. コネクタ・スクリプトを生成または更新するには、「インストール」をクリックする。

`connector_manager_ConnectorName` ファイル (ここで、`ConnectorName` は構成するコネクタの名前) が `ProductDir/bin` ディレクトリーに作成されます。

6. 「閉じる」をクリックする。

付録 G. Visual Test Connector の使用

Visual Test Connector は、コネクターの活動をシミュレートすることで、実際にコネクターを実行するという手間をかけずに、統合コンポーネントをテストできます。本章は、以下のセクションから構成されています。

- 『推奨されるテスト手順』
- 185 ページの『Test Connector の始動』
- 186 ページの『Test Connector のシャットダウン』
- 186 ページの『コネクター・プロファイルの作成と編集』
- 188 ページの『コネクターのエミュレート』
- 189 ページの『ビジネス・オブジェクトの処理』

推奨されるテスト手順

WebSphere Business Integration システムでコンポーネントをテストするための、推奨されるテスト手順は次のとおりです。

1. ご使用の統合ブローカーが InterChange Server の場合、System View ビューの使用を検討してください。送信したフローが成功したか失敗したかを判断するのに大変便利です。

詳しくは、「システム管理ガイド」を参照してください。

2. Test Connector を、ソース・コネクターをエミュレートするようにセットアップする。
 - a. 185 ページの『Test Connector の始動』の説明に従って、Test Connector を起動する。
 - b. インターフェース内のソース・コネクター用プロファイルを、186 ページの『新規プロファイルの作成』の説明に従って作成する。
 - c. 188 ページの『コネクターのエミュレート』の説明に従って、Test Connector をエージェントに接続し、ソース・コネクターのエミュレーションを開始する。
3. Test Connector のインスタンスを、インターフェースに接続された各宛先コネクターをエミュレートするようにセットアップする。
 - a. 185 ページの『Test Connector の始動』の説明に従って、Test Connector を起動する。
 - b. 宛先コネクター用のプロファイルを、186 ページの『新規プロファイルの作成』に示す方法で作成する。
 - c. 188 ページの『コネクターのエミュレート』の説明に従って、Test Connector をエージェントに接続し、宛先コネクターのエミュレーションを開始する。
 - d. インターフェースに接続されたすべての宛先コネクターについて、上記の 3a から 3c を繰り返す。
4. 各「Test Connector」ウィンドウで、エミュレートしているコネクターを識別しやすいように、画面上の Test Connector のインスタンスを配置する。例えば

図 41 では、ソース Test Connector は宛先 Test Connector の左側に配置されています。

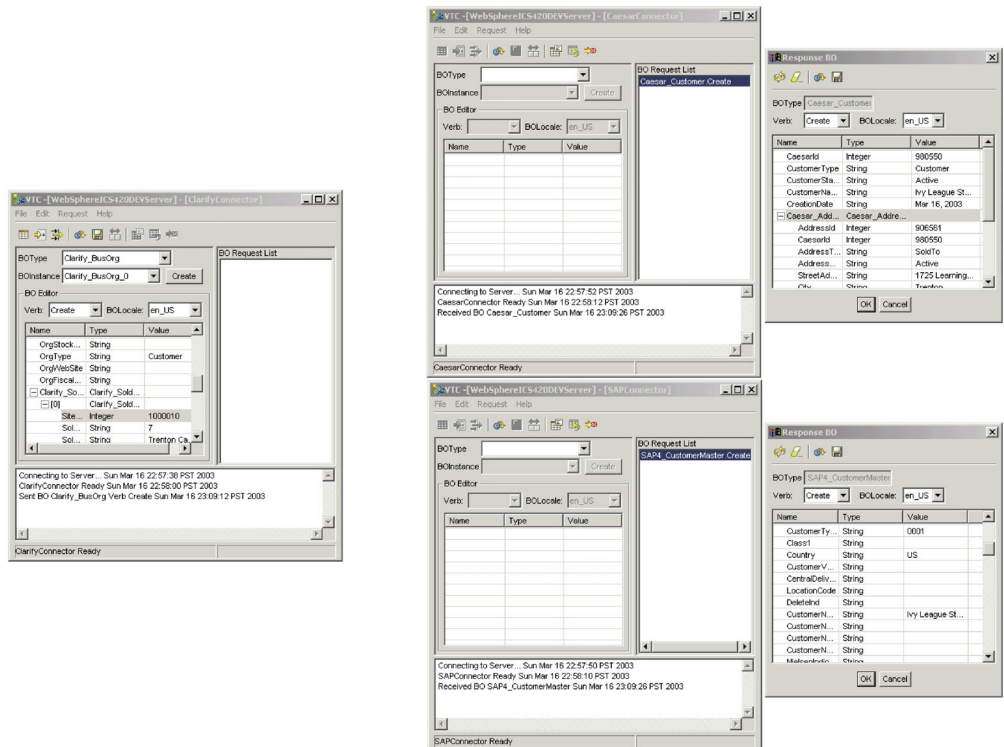


図 41. Test Connector のソース・インスタンスおよび宛先インスタンス

5. ソース・コネクタから要求ビジネス・オブジェクトを送信する。ソース Test Connector で、以下の手順を実行します。
 - a. テストを必要とするインターフェースにより管理されるビジネス・オブジェクトを、189 ページの『要求ビジネス・オブジェクトの作成』に示す方法により作成する。
 - b. 194 ページの『ビジネス・オブジェクトの保管』の説明に従って、このビジネス・オブジェクトをファイルに保管し、以降のテストでも使用できるようにする。
 - c. ビジネス・オブジェクトを、190 ページの『要求ビジネス・オブジェクトの送信』に示す方法により送る。
6. 要求ビジネス・オブジェクトに対する宛先コネクタからの応答をシミュレートする。宛先「Test Connector」ウィンドウで、以下の手順を実行します。
 - a. ビジネス・オブジェクトを、195 ページの『要求ビジネス・オブジェクトの受け入れ』に示す方法により受け付ける。
 - b. ビジネス・オブジェクトを、196 ページの『応答ビジネス・オブジェクトの送信』に示す方法により応答として送る。
7. ステップ 5 から 6 を、各インターフェースのテストに必要な回数だけ繰り返す。

Test Connector の始動

Test Connector を始動するには、統合ブローカーの種類に応じて、次のいずれかの方法を実行します。

- 統合ブローカーが InterChange Server の場合は、「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「開発」>「Test Connector」を選択する。
- ご使用の統合ブローカーが WebSphere Application Server または WebSphere message broker (WebSphere MQ Integrator、WebSphere MQ IntegratorBroker、または WebSphere Business Integration Message Broker) の場合、「スタート」>「プログラム」>「IBM WebSphere Business Integration Adapters」>「ツール」>「Test Connector」を選択します。

図 42 に、始動後の Test Connector を示します。

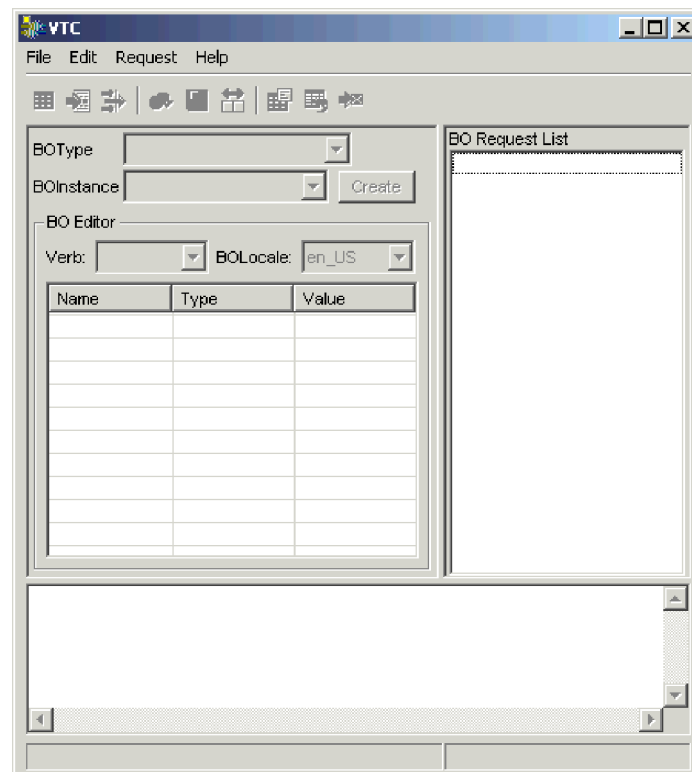


図 42. Test Connector

「Test Connector」ウィンドウには、以下のペインがあります。

- 「サポートされているビジネス・オブジェクト」ペインでは、送信するビジネス・オブジェクトのインスタンスを作成できます。
- 「ビジネス・オブジェクト要求リスト」ペインには、コネクターが受信したすべてのビジネス・オブジェクト要求が表示されます。
- 「出力」ペインには、ビジネス・オブジェクトが送信された時刻などの、Test Connector の操作に関するメッセージが表示されます。

Test Connector のシャットダウン

Test Connector をシャットダウンし、コネクタ・エージェントのエミュレーションを停止させるには、メニュー・バーから「ファイル」>「終了」を選択します。「シャットダウン」プロンプトが表示された場合は、「はい」をクリックします。

コネクタ・プロファイルの作成と編集

Test Connector は、コネクタをエミュレートするために必要となる情報を保管するために、プロファイルを使用します。エミュレートするコネクタごとにプロファイルを作成する必要があります。既存のプロファイルを編集したり、削除したりすることも可能です。

ファイルへのコネクタ定義の保管

Test Connector を使用してコネクタをエミュレートするには、コネクタ定義をファイルに保管しておく必要があります。コネクタ定義をファイルに保管するには、以下の手順を実行します。

1. Connector Configurator で、コネクタ定義を開く。
2. メニュー・バーから「ファイル」>「別名保管」>「ファイルに」を選択する。
3. ファイルの保管先とするディレクトリに移動し、「ファイル名」フィールドに名前を入力する。「ファイルの種類 (Save as type)」ドロップダウン・メニューに値「構成ファイル (*.cfg)」が表示されていることを確認して、「保管」をクリックする。

Connector Configurator は、指定した名前でもコネクタ定義をファイルに保管します。

新規プロファイルの作成

プロファイルは、Test Connector でエミュレートするすべてのコネクタについて作成する必要があります。プロファイルでは、コネクタの名前、使用する構成ファイル、コネクタが通信する統合ブローカーのタイプなどの情報を指定します。新規コネクタ・プロファイルを作成する手順は次のとおりです。

1. メニュー・バーから「ファイル」>「プロファイルを作成/選択」を選択し、「コネクタ・プロファイル」ウィンドウを表示する。
2. 「コネクタ・プロファイル」ウィンドウで、メニュー・バーから「ファイル」>「新規プロファイル」を選択する。
3. 「新規プロファイル」ウィンドウで「参照」をクリックし、『ファイルへのコネクタ定義の保管』で用意したコネクタの構成ファイルに移動する。
4. 「コネクタ名」フィールドに、コネクタの名前を入力する。コネクタ定義の名前は、統合ブローカーのリポジトリに存在するものと完全に同じものを入力する必要があります。例えば、Adapter for JText の場合は、JTextConnector と入力します。その際、JText という単語と Connector という単語の間にスペースを入れないようにし、すべての文字の大文字と小文字を正しく入力する必要があります。
5. 「ブローカー・タイプ」ドロップダウン・メニューで、適切な統合ブローカー (ICS、WMQI または WAS) を選択する。

注: ご使用のブローカーが WebSphere message broker の場合、WMQI を選択してください。

6. ステップ 5 (186 ページ) で、ブローカーのタイプとして ICS を選択した場合は、以下の作業も行う必要があります。
 - a. 「サーバー」フィールドに、InterChange Server インスタンスの名前を入力する。

名前は正確に入力してください。この名前では大文字小文字が区別されません。名前が不正確な場合には、Test Connector は InterChange Server と通信できません。

- b. 「パスワード」フィールドに、admin ユーザー・アカウントのパスワードを入力する。デフォルトのパスワードは null です。

図 43 に、「新規プロファイル」ウィンドウを示します。

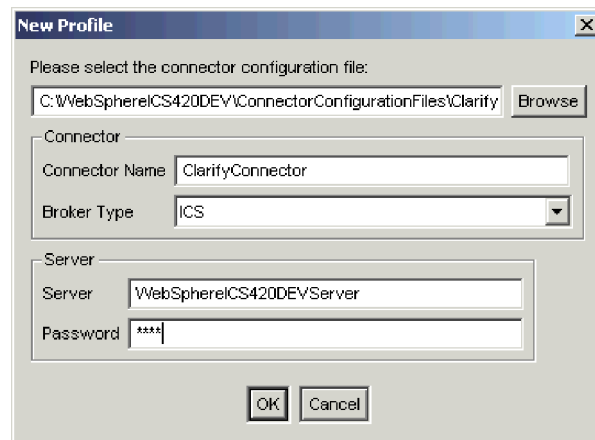


図 43. 新規コネクタ・プロファイルの作成

7. 「OK」をクリックして、「新規プロファイル」ウィンドウを閉じる。

「コネクタ・プロファイル」ウィンドウの「コネクタ」列にはコネクタの名前が、「サーバー」列には InterChange Server インスタンスの名前 (統合ブローカーが ICS の場合) が、「構成ファイル」列にはコネクタ構成ファイルのパスと名前が表示されます。

188 ページの図 44 に、InterChange Server インスタンスと通信するように構成された ClarifyConnector のプロファイルと、WMQIB サーバーと通信するように構成された JTextConnector のプロファイルが表示された「コネクタ・プロファイル」ウィンドウを示します。

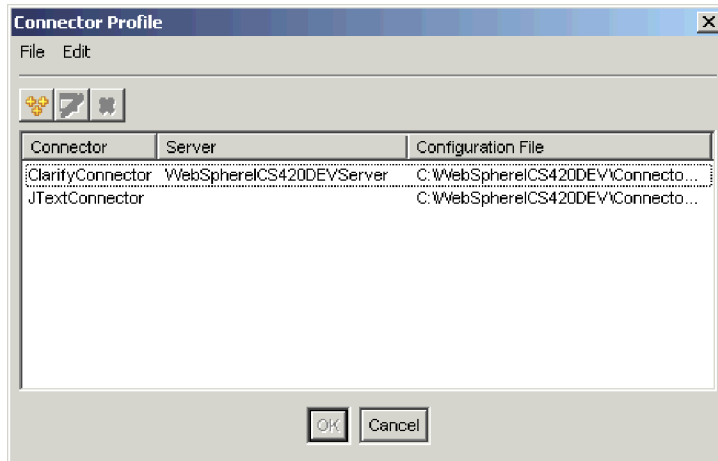


図 44. 「コネクタ・プロファイル」 ウィンドウ

8. 「OK」をクリックして、「コネクタ・プロファイル」ウィンドウを閉じる。

プロファイルの編集

既存のコネクタ・プロファイルに変更を加える手順は次のとおりです。

1. Test Connector のメニュー・バーから「ファイル」>「プロファイルを作成/選択」を選択するか、またはキーボード・ショートカット **Ctrl+N** を使用して、「コネクタ・プロファイル」ウィンドウを表示する。
2. 「コネクタ・プロファイル」ウィンドウで、編集するプロファイルを選択し、次にメニュー・バーから「編集」>「プロファイルを編集」を選択する。
3. 「新規プロファイル」ウィンドウの各フィールドに新しい値を入力し、「参照」ボタンを使用して、編集作業の必要に応じて構成ファイルを切り替える。
4. 「OK」をクリックして、「新規プロファイル」ウィンドウを閉じる。

プロファイルの削除

コネクタ・プロファイルを削除するには、以下の手順を実行します。

1. Test Connector のメニュー・バーから「ファイル」>「プロファイルを作成/選択」を選択するか、またはキーボード・ショートカット **Ctrl+N** を使用して、「コネクタ・プロファイル」ウィンドウを表示する。
2. 「コネクタ・プロファイル」ウィンドウで、削除するプロファイルを選択し、次にメニュー・バーから「編集」>「プロファイルを削除」を選択する。

コネクタのエミュレート

コネクタ用のプロファイルを作成した後、そのプロファイルを使用して、Test Connector をエージェントに接続できます。Test Connector をエージェントに接続すると、Test Connector は、選択したプロファイルで定義されているコネクタのエミュレーションを開始します。

Test Connector をエージェントに接続するには、以下の手順を実行します。

1. Test Connector のメニュー・バーから、「ファイル」>「プロファイルを作成/選択」を選択する。

2. 「コネクタ・プロファイル」ウィンドウで、これから開くコネクタ用プロファイルのコネクタ名を選択する。
3. 「OK」をクリックする。
4. メニュー・バーから、「ファイル」>「接続」を選択する。

Test Connector は、コネクタのエミュレーションを試行する際に、「出力」ペインにメッセージを表示します。接続が完了すると、準備が完了したことを示すメッセージを「出力」ペインに表示し、「サポートされているビジネス・オブジェクト」ペインに「BOType」リストを取り込みます。

ビジネス・オブジェクトの処理

ビジネス・プロセス・インターフェースが正しく開発されたことを検証するには、ビジネス・オブジェクトが正常に交換および処理されることを確認する必要があります。このセクションでは、以下の方法について説明します。

- ビジネス・オブジェクト・テスト・データの作成、変更、削除、および保管
- ビジネス・オブジェクトの属性値を比較しての、処理中に行われた変更の迅速かつ容易な確認
- ビジネス・オブジェクトの送受信

要求ビジネス・オブジェクトの処理

要求ビジネス・オブジェクトは、インターフェースを起動するイベントのソース・コネクタを Test Connector がエミュレートしている際に、Test Connector から送信されるビジネス・オブジェクトです。要求ビジネス・オブジェクトの処理では、ビジネス・オブジェクト・インスタンスを作成し、そのインスタンスにデータを取り込み、要求を送信します。

要求ビジネス・オブジェクトの作成

Test Connector で新しいビジネス・オブジェクトを作成するには、以下の手順を実行します。

1. 「サポートされているビジネス・オブジェクト」ペインで、作成するビジネス・オブジェクトの名前を「BOType」ドロップダウン・メニューから選択する。
2. 「BOInstance」フィールドの隣にある「作成」をクリックする。
3. 「新規インスタンス (New Instance)」ダイアログが表示されたら、インスタンスの名前を「名前を入力してください」フィールドに入力する。
4. 「動詞」ドロップダウン・メニューから、必要な動詞を選択する。
5. 「BOLocale」ドロップダウン・メニューから、必要なロケールを選択する。
6. 193 ページの『ビジネス・オブジェクト属性の値の設定』に示す方法により、トップレベル・オブジェクト内の単純属性および子ビジネス・オブジェクトの値を指定する。

190 ページの図 45 に、Caesar_Customer という名前のビジネス・オブジェクトを示します。このビジネス・オブジェクトの動詞は Create、ロケールは en_US で、そのそれぞれの単純属性に対して値が指定されており、Caesar_Address 子ビジネス・オブジェクトという単一インスタンスを持っています。

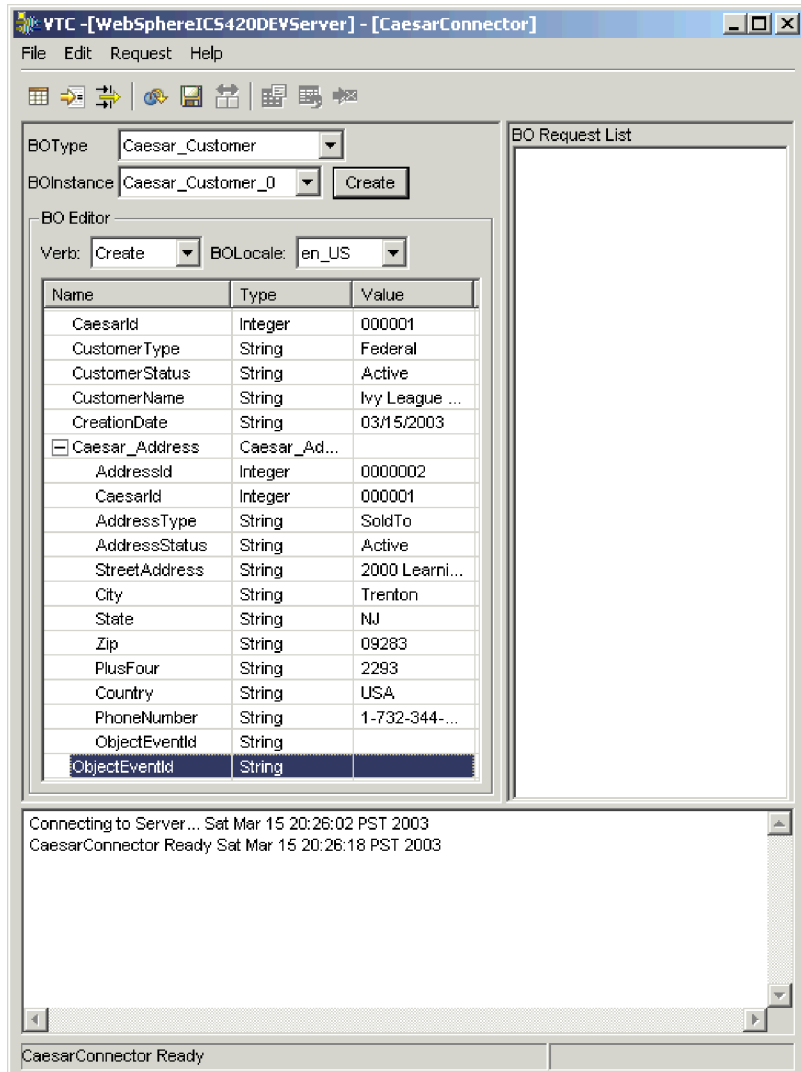


図 45. ビジネス・オブジェクトへのデータの取り込み

7. 「OK」をクリックする。

要求ビジネス・オブジェクトの送信

ビジネス・オブジェクトを作成またはロードし、その属性の値を指定した後、このビジネス・オブジェクトを要求として統合ブローカーに送信するには、いくつかの方法があります。

非同期モードでの要求ビジネス・オブジェクトの送信: ソース・コネクタは、要求ビジネス・オブジェクトを非同期モードで送ったとき、応答ビジネス・オブジェクトが戻ることを期待しません。要求ビジネス・オブジェクトがディスパッチされてしまえば、トランザクションにおけるソース・コネクタの役割は終了します。応答ビジネス・オブジェクトは、通常は統合ブローカーが処理します。Test Connector のデフォルトのモードは非同期です。

ビジネス・オブジェクトを非同期モードで送信する手順は次のとおりです。

1. メニュー・バーから、「要求」>「モード」>「非同期」を選択する。

注: Test Connector は、デフォルトでは「非同期」モードで作動するため、このステップは、以前にコネクタから同期要求を送信していた場合にのみ行う必要があります。また、各要求を送信する前にモードを設定する必要はありません。

2. メニュー・バーから、「要求」>「送信」を選択する。

コネクタ定義で指定されているブローカーが InterChange Server の場合、ビジネス・オブジェクト要求が処理のためにサーバーに送信されます。

コネクタ定義で指定したブローカーが、サポートされるメッセージ・ブローカーか WebSphere Application Server の場合には、ビジネス・オブジェクトは RequestQueue 標準プロパティで指定しているキューに置かれます。

同期モードでの要求ビジネス・オブジェクトの送信: ソース・コネクタは、要求ビジネス・オブジェクトを同期モードで送信すると、宛先アプリケーションが要求を処理した後、統合ブローカーから応答ビジネス・オブジェクトが戻ることを期待します。同期モードでは、Test Connector は応答ビジネス・オブジェクトを、ソース・コネクタの Synchronous Request Queue プロパティで指定されたキューに入れます。Test Connector のデフォルトのモードは非同期です。

1. メニュー・バーから「要求」>「モード」>「同期」を選択して、Test Connector を同期モードに設定する。
2. メニュー・バーから、「要求」>「送信」を選択する。
3. コネクタ定義で指定されているブローカーが InterChange Server の場合は、「コラボレーションの選択」ダイアログが表示される。ビジネス・オブジェクトの送信先とするコラボレーションを、「コラボレーション」ドロップダウン・メニューから選択し、「OK」をクリックする。

コネクタ定義で指定されているブローカーが InterChange Server の場合、ビジネス・オブジェクト要求が、処理のために、選択されたコラボレーション・オブジェクトの構成済みポートに送信されます。

コネクタ定義で指定したブローカーが サポートされるメッセージ・ブローカーか、WebSphere Application Server の場合には、ビジネス・オブジェクトは SynchronousRequestQueue 標準プロパティで指定しているキューに置かれます。

バッチ・モードでの要求ビジネス・オブジェクトの送信: バッチ・モードの場合、Test Connector では、送信する特定のビジネス・オブジェクトのインスタンス数を指定できます。また、それぞれのインスタンスごとに固有値に設定する必要のあるトップレベル・オブジェクトの 1 つの属性 (例えば、基本キー属性) も指定できます。Test Connector は、指定された回数だけビジネス・オブジェクトをコピーし、指定された 1 つの属性の値を増分し、それぞれのビジネス・オブジェクトを送信します。このオプションにより、多数のビジネス・オブジェクトを迅速かつ容易に作成することができます。

選択された属性が、識別関係の一環として動的相互参照に関与しているキー・フィールドである場合は、初期値も後続する値も固有であることを保証する必要があります。この条件が満たされていないと、相互参照のロジックが成立せず、要求ビジネス・オブジェクトは失敗します。

値が固有であることを確認するためには、関係マネージャーを使用するか、次の手順により関係参加項目のテーブルに対して SQL ステートメントを実行します。

- 参加項目の現在の最高値を判別し、それよりも大きな値を初期値フィールドに設定する。バッチ送信の最初のビジネス・オブジェクト・インスタンスおよび後続のインスタンスはすべて固有となります。
- 参加項目の既存テーブル・エントリを削除することにより、どのテーブル・エントリもバッチ・ビジネス・オブジェクトのどれとも属性値が同じにならないようにする。

バッチ・モードでビジネス・オブジェクトを送信する手順は次のとおりです。

1. 送信するビジネス・オブジェクトの名前を、「**BOType**」ドロップダウン・メニューから選択する。
2. メニュー・バーから、「**要求**」>「**バッチを送信**」を選択する。
3. 「バッチ・モード」ウィンドウで、「**動詞**」ドロップダウン・メニューから、必要な動詞を選択する。
4. 「**BOLocale**」ドロップダウン・メニューから、必要なロケールを選択する。
5. 「**属性**」リストから、バッチ内の各ビジネス・オブジェクト要求ごとに増分させる、トップレベル・ビジネス・オブジェクトの属性を選択する。

通常は、基本キーなど、ビジネス・オブジェクトを一意的に特定する属性を選択します。

6. 「**初期値**」フィールドに、増加させる属性の初期値を入力する。
7. 「**ビジネス・オブジェクト数**」フィールドに、生成して送信するビジネス・オブジェクト・インスタンスの数を入力する。
8. 「**OK**」をクリックする。

Test Connector は、指定された数のビジネス・オブジェクトを生成します。これらのビジネス・オブジェクトは、インスタンスごとに値が増分されるように指定された 1 つの属性を除いて、すべて同一です。

コネクタ定義で指定されているブローカーが InterChange Server の場合、ビジネス・オブジェクト要求が処理のためにサーバーに送信されます。

コネクタ定義で指定したブローカーが、サポートされるメッセージ・ブローカーか WebSphere Application Server の場合には、ビジネス・オブジェクトは RequestQueue 標準プロパティで指定しているキューに置かれます。

193 ページの図 46 に、バッチ・モード構成の例を示します。この例では、次のように指定されています。

- 送信対象のビジネス・オブジェクトは 50 個。
- 属性 OrgObjid の値を増分する。
- 属性の初期値は 100001。

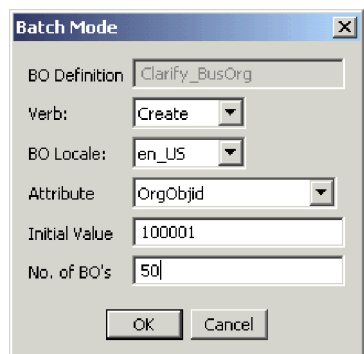


図 46. 「バッチ・モード」ウィンドウ

ビジネス・オブジェクト属性の値の設定

以下の各セクションでは、ビジネス・オブジェクト・インスタンスの単純属性および複合属性の値を設定するためのさまざまな方法について説明します。

- 『単純属性の値の設定』
- 『子ビジネス・オブジェクトの追加』
- 194 ページの『子ビジネス・オブジェクトの除去』
- 194 ページの『子ビジネス・オブジェクトの動詞の設定』

単純属性の値の設定

単純属性の値を設定するには、「値」列でその属性のセルをクリックして、値を入力します。

子ビジネス・オブジェクトの追加

子ビジネス・オブジェクトのインスタンスを追加するには、その子ビジネス・オブジェクトを表す属性上で右マウス・ボタンをクリックし、コンテキスト・メニューから「インスタンスを追加」を選択します。

子ビジネス・オブジェクトを表す属性の隣に正符号 (+) が追加され、少なくとも 1 つの子ビジネス・オブジェクト・インスタンスが存在することを示します。子オブジェクト属性を展開すると、各インスタンスに対応した番号付き項目が表示されます。個々のインスタンスにもその隣に正符号 (+) が付いているため、それらを展開して属性に値を設定できます。

子ビジネス・オブジェクトのインスタンスをさらに追加するには、子オブジェクトを表す属性を右マウス・ボタン・クリックし、コンテキスト・メニューから「インスタンスを追加」を選択します。

注: 子ビジネス・オブジェクトを参照する属性の「カード」プロパティが値 1 (カーディナリティが単一であることを示す) に設定されている場合、追加できる子オブジェクトのインスタンスは 1 つのみです。

子ビジネス・オブジェクトの除去

子ビジネス・オブジェクトのインスタンスを削除するには、そのインスタンスの上で右マウス・ボタンをクリックし、コンテキスト・メニューから「インスタンスを削除」を選択します。

子ビジネス・オブジェクトのすべてのインスタンスを削除するには、子ビジネス・オブジェクトを表す属性を右マウス・ボタン・クリックし、コンテキスト・メニューから「全インスタンスを削除」を選択します。

子ビジネス・オブジェクトの動詞の設定

子ビジネス・オブジェクトの動詞を設定することにより、設定された値がビジネス・オブジェクトに及ぼす影響をテストすることができます。これは、子オブジェクトの相互参照が絡んでいるロジックのトラブルシューティングを実行するときに役立ちます。

子ビジネス・オブジェクト・インスタンスの動詞を設定するには、そのインスタンスを右マウス・ボタン・クリックし、コンテキスト・メニューから「動詞を設定」を選択します。「動詞の選択」プロンプトが表示されたら、必要な動詞を選択し、「OK」をクリックします。

「応答ビジネス・オブジェクト」ツールバーの使用

宛先コネクタが受信したビジネス・オブジェクトの属性は、応答として送信する前に編集することができます。編集する際に使用する「応答ビジネス・オブジェクト」ダイアログのツールバーには、ビジネス・オブジェクトの値を設定するために使用できる、いくつかのツールバー・ボタンがあります。詳しくは、196 ページの『応答ビジネス・オブジェクトの編集』を参照してください。

ビジネス・オブジェクトの保管

Test Connector では、ビジネス・オブジェクトを保管して、後のテストで使用したり、テクニカル・サポートに提供したり (問題のトラブルシューティングに役立てるため)、応答データとして使用したりすることができます。ユーザーが作成したビジネス・オブジェクトや、宛先コネクタの「Test Connector」ウィンドウに要求として表示されたビジネス・オブジェクトを含め、すべてのビジネス・オブジェクトを保管できます。デフォルトでは、ビジネス・オブジェクトはビジネス・オブジェクト拡張子 (.bo) の付いたファイルに保管されます。

必要に応じて、テスト・データ・ファイル専用のディレクトリーやディレクトリー構造 (各インターフェースまたは各コネクタに対応したサブディレクトリーを持つ) を作成することをお勧めします。この構造により、必要なファイルが見つけやすくなり、テストの効率が上がります。さらに、ビジネス・オブジェクトのテスト・データ・ファイルには、ビジネス・オブジェクト定義自体と同じ名前を付けることをお勧めします。

要求ビジネス・オブジェクトの保管

要求として作成したビジネス・オブジェクト・インスタンスを保管するには、以下の手順を実行します。

1. 保管したいビジネス・オブジェクトを選択する。

2. メニュー・バーから、「編集」>「ビジネス・オブジェクトを保管」を選択する。
3. 目的のディレクトリーにナビゲートし、「ファイル名」フィールドにファイルの名前を指定する。
4. 「保管」をクリックする。

応答ビジネス・オブジェクトの保管

Test Connector の宛先インスタンスが受信し、応答として送信されるビジネス・オブジェクト・インスタンスを保管するには、以下の手順を実行します。

1. 「ビジネス・オブジェクト要求リスト」ペインで、ビジネス・オブジェクト・インスタンスを選択する。
2. メニュー・バーから、「要求」>「応答を編集」を選択する。
3. 「ビジネス・オブジェクトを保管」をクリックする。
4. 目的のディレクトリーにナビゲートし、「ファイル名」フィールドにファイルの名前を指定する。
5. 「保管」をクリックする。

ビジネス・オブジェクトのロード

ファイルに保管されたビジネス・オブジェクトをロードする手順は次のとおりです。

1. Test Connector のメニュー・バーから、「編集」>「ビジネス・オブジェクトをロード」を選択する。
2. ビジネス・オブジェクトのテスト・データ・ファイルに移動し、それを開く。
3. 「新規インスタンス (New Instance)」ダイアログが表示されたら、インスタンスの名前を「名前を入力してください」フィールドに入力する。
4. 「OK」をクリックする。

ビジネス・オブジェクトの削除

ビジネス・オブジェクトを Test Connector から削除するには、メニュー・バーから「編集」>「ビジネス・オブジェクトを削除」を選択します。

注: この操作では、Test Connector からビジネス・オブジェクトのみが削除されません。コネクタによるビジネス・オブジェクト定義のサポートは削除されません。

要求ビジネス・オブジェクトの受け入れ

ビジネス・オブジェクトを要求として送信すると、そのビジネス・オブジェクトは、トランザクションに障害が発生しない限り、インターフェース内の宛先コネクタをエミュレートしているすべての Test Connector インスタンスの「ビジネス・オブジェクト要求リスト」ペインに表示されます。

受け入れた要求ビジネス・オブジェクトは、196 ページの『応答ビジネス・オブジェクトの編集』に示した方法で、必要に応じて編集できます。

応答ビジネス・オブジェクトの処理

応答ビジネス・オブジェクトは、インターフェース内のビジネス・オブジェクト要求の宛先のコネクターを Test Connector がエミュレートしている際に、Test Connector から送信されるビジネス・オブジェクトです。要求ビジネス・オブジェクトの処理では、ビジネス・オブジェクト・インスタンス内の値を編集し、応答をブローカーに返送します。

応答ビジネス・オブジェクトの編集

Test Connector の宛先インスタンスでビジネス・オブジェクト要求を受信すると、通常、属性の値を編集する必要が生じます。例えば、関係に参与している基本キー属性への固有値の設定や、ビジネス・オブジェクトの実際の値に応じて異なった応答をするマップやコラボレーション・ロジックをテストするために、その他の属性の値の変更を行うことが必要となります。ビジネス・オブジェクト属性の値を設定するには、以下の手順を実行します。

1. 「ビジネス・オブジェクト要求リスト」ペインで、ビジネス・オブジェクト・インスタンスを選択する。
2. メニュー・バーから、「要求」>「応答を編集」を選択する。
3. 以下の方法で、ビジネス・オブジェクトの属性を編集する。
 - 193 ページの『ビジネス・オブジェクト属性の値の設定』に示した方法の 1 つを使用して、ビジネス・オブジェクト属性の値を変更する。
 - ビジネス・オブジェクト属性の値を、ビジネス・オブジェクト定義で指定されたデフォルト値に設定するには、「ビジネス・オブジェクトをデフォルトにリセット」をクリックする。
 - ビジネス・オブジェクトのすべての属性の値をクリアするには、「ビジネス・オブジェクト値をクリア」をクリックする。
 - ビジネス・オブジェクトの属性に、ファイルからテスト・データを取り込むには、「ビジネス・オブジェクトをロード」をクリックする。

保管されているデータをビジネス・オブジェクト要求にロードする機能は、応答として送信する前に応答ビジネス・オブジェクトにデータを取り込む必要がある場合に非常に役立ちます。応答データを必要とする各属性のために値を手動で入力する代わりに、値を 1 度だけ入力し、ビジネス・オブジェクトを保管し (194 ページの『ビジネス・オブジェクトの保管』に示す方法で)、保管されたデータを以降のテスト時にロードすることができます。

応答ビジネス・オブジェクトの送信

要求ビジネス・オブジェクトを受け付けたら、必要に応じて編集し、応答として返送します。

表 28 に、Test Connector の応答オプションと、それに対応する C++ コネクターと Java コネクターの双方のコネクター戻りコードを示します。C++ または Java コネクターの戻りコードについての詳細は、「コネクター開発ガイド (Java 用または C++ 用)」を参照してください。

表 28. Test Connector の応答タイプとコネクター戻りコード

Test Connector の応答タイプ	C++ コネクターの戻りコード	Java コネクターの戻りコード
成功	BON_SUCCESS	SUCCESS

表 28. Test Connector の応答タイプとコネクタ戻りコード (続き)

Test Connector の応答タイプ	C++ コネクタの戻りコード	Java コネクタの戻りコード
失敗	BON_FAIL	FAIL
複数のヒット	BON_MULTIPLE_HITS	MULTIPLE_HITS
内容による検索に失敗	BON_FAIL_RETRIEVE_BY_CONTENT	RETRIEVEBYCONTENT_FAILED
検出なし	BON_BO_DOES_NOT_EXIST	BO_DOES_NOT_EXIST
重複する値	BON_VALDUPES	VALDUPES

要求ビジネス・オブジェクトに応答する手順は次のとおりです。

1. 「ビジネス・オブジェクト要求リスト」ペインで、ビジネス・オブジェクトを選択する。
2. メニュー・バーから、「要求」>「応答」を選択する。
3. 「応答」サブメニューから、項目を選択する。

オブジェクト・オブジェクト・インスタンスの比較

Test Connector では、同じタイプの 2 つのビジネス・オブジェクトを比較して、双方の値が異なっている属性を表示できます。この機能を使用すると、トランザクション実行中の異なる時点での、ビジネス・オブジェクトの変化を確認できます (例えば、同一のビジネス・オブジェクトについて、統合ブローカーに送信された時点のものと、統合ブローカーによって更新された後のものとを比較できます)。2 つのビジネス・オブジェクトを比較する手順は次のとおりです。

1. 189 ページの『要求ビジネス・オブジェクトの作成』または 195 ページの『ビジネス・オブジェクトのロード』の説明に従って、要求ビジネス・オブジェクトのインスタンスを作成する。
2. 要求ビジネス・オブジェクトのインスタンスと比較する応答ビジネス・オブジェクトのインスタンスを、「ビジネス・オブジェクト要求リスト」ペインで選択する。
3. メニュー・バーから、「編集」>「ビジネス・オブジェクトを比較」を選択する。

Test Connector で「ビジネス・オブジェクトの比較」ウィンドウが開き、2 つのビジネス・オブジェクトで値の異なる属性を示したテーブルが表示されます。図 47 に、2 つのビジネス・オブジェクト・インスタンスの比較を示します。

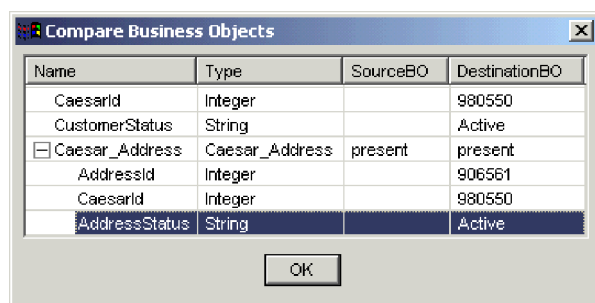


図 47. 「ビジネス・オブジェクトの比較」ウィンドウ

4. 「OK」をクリックして、ウィンドウを閉じる。

付録 H. WebSphere Business Integration Adapters のアップグレード

この付録では、WebSphere Business Integration Adapters の新リリースにアップグレードするためのプロセスについて説明します。本章は、以下のトピックから構成されています。

- 『前提事項』
- 『WebSphere Business Integration Adapters のインストール』
- 200 ページの『WebSphere Message Broker のアップグレード』
- 200 ページの『応答キューの作成』
- 201 ページの『ローカル・キュー・マネージャーのコード化文字セット ID 値の更新』
- 201 ページの『既存のアダプターのアップグレード』
- 204 ページの『新規アダプターの構成』
- 204 ページの『新規 WebSphere MQ バッチ・ファイルのカスタマイズ』。

前提事項

このアップグレード手順では、以下の条件であることが想定されています。

- WebSphere MQ Integrator Broker を統合ブローカーとして使用するシステム上でアップグレードを行っている。(現行リリースの前は、他のメッセージ・ブローカーはサポートされていませんでした。)
- 現在、システムに WebSphere Business Integration Adapters バージョン 2.0、2.0.1、2.1.0、2.2.0、または 2.3.1 がインストールされている。
- 必要なファイルはすべてバックアップしている。
- 実行中のシステムおよびすべてのコネクタが停止している。
- リリース情報を読み、インストールに影響する可能性があるリリース固有のアップグレード情報を理解している。オンラインのリリース情報は、WebSphere Business Integration Adapters InfoCenter (<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>) で参照できます。
- 開発環境でアップグレードし、テストが完了してから実稼働環境のアップグレードに移行する。

WebSphere Business Integration Adapters のインストール

WebSphere Business Integration Adapters をインストールするには、「*WebSphere Business Integration Adapters インストール・ガイド*」に記載されている説明に従ってください。この資料は、Web アドレス

<http://www.ibm.com/software/websphere/wbiadapters/infocenter> で入手できます。

WebSphere Message Broker のアップグレード

注: WebSphere MQ Integrator Broker または WebSphere MQ Integrator から WebSphere Business Integration Message Broker へのアップグレードを行う場合は、アップグレード処理の一環として、現在のメッセージ・ブローカー (WebSphere MQ Integrator Broker または WebSphere MQ Integrator) で生成したメッセージ・セットとメッセージ・フローを WebSphere Business Integration Message Broker に必ずインポートしてください。

WebSphere MQ Integrator Broker

WebSphere Business Integration Adapters バージョン 2.3.1 からアップグレードする場合は、WebSphere MQ Integrator Broker CSD05 Service Pack を適用してください。このパッケージは以下の URL から入手できます。

<http://www.ibm.com/software/integration/mqfamily/support/summary/mqsib.html>

WebSphere MQ Integrator

WebSphere MQ Integrator のアップグレード方法の詳細については、お客様の IBM 担当員にご連絡ください。

WebSphere Business Integration Message Broker

今回が、WebSphere Business Integration Message Broker をサポートする最初のリリースであるため、アップグレードの必要はありません。

WebSphere MQ 用 Service Pack のインストール

WebSphere Business Integration Adapters バージョン 2.3.1 からアップグレードする場合は、WebSphere MQ CSD05 Service Pack を適用してください。このパッケージは以下の URL から入手できます。

<http://www.ibm.com/software/integration/mqfamily/support/summary/>

応答キューの作成

2.2.0 より前のリリースの WebSphere Business Integration Adapters からアップグレードする場合は、以下のようにして新規応答キューを定義する必要があります。リリース 2.2.0 以降をすでにお持ちの場合は、この作業を実行する必要はありません。

WebSphere Business Integration Adapters バージョン 2.2.0 では、追加の WebSphere MQ キューである応答キューが必要です。このキューを、90 ページの『キューを定義する方法』にリストされている方法を使用して定義してください。202 ページの『コネクタ構成ファイルの更新』では、このキューの名前を、新しいコネクタ構成標準プロパティの ResponseQueue に割り当てる必要があります。

ローカル・キュー・マネージャーのコード化文字セット ID 値の更新

WebSphere Business Integration Adapters バージョン 2.0 からアップグレードする場合は、各キュー・マネージャーの CCSID を更新するために、次のコマンドを実行してください。

```
runmqsc alter qmgr ccsid(1208)
```

このコマンドの詳細については、「*WebSphere MQ: MQSC コマンド・リファレンス*」を参照してください。

既存のアダプターのアップグレード

アップグレードする既存の各アダプターについて、以下のトピックにリストされている作業を行ってください。

- 『ビジネス・オブジェクトのアップグレード』
- 202 ページの『コネクタ構成ファイルの更新』
- 204 ページの『新しいコネクタ始動スクリプトの構成』

ビジネス・オブジェクトのアップグレード

リリース 2.0.1、2.1.0、2.2.0、2.3.1 からのアップグレード

WebSphere Business Integration Adapters バージョン 2.0.1、2.1.0、2.2.0、2.3.1 からアップグレードする場合、ビジネス・オブジェクトのアップグレードは必要ありません。

リリース 2.0 からのアップグレード

注: WebSphere Business Integration Adapters バージョン 2.0 から既存のアダプターをアップグレードする場合にのみ、このステップを実行してください。

アップグレード対象アダプターごとに、以下の作業を行います。

1. インストールした新しいバージョンの Business Object Designer を使用し、アダプターの既存のビジネス・オブジェクト定義をロードし、同じ名前を使用して保管します。ビジネス・オブジェクト定義は、WebSphere Business Integration Adapters で使用される最新形式で保管されます。
- 2.

注: 統合ブローカーとして WebSphere Business Integration Message Broker 5.0 を使用してアップグレードする場合は、このステップは省略してください。その代わりに、99 ページの『メッセージ・ブローカー・ワークスペースへの配置』の説明に従って、System Manager を使用して再インポートすることで、ビジネス・オブジェクトを更新します。

リセット・モードの XML Schema Importer を使用して、更新済みビジネス・オブジェクト定義を WebSphere MQ Integrator Broker の MRM にインポートし、メッセージ・セットから既存の情報をすべて除去してから、更新した XMLスキーマのインポートを開始します。ビジネス・オブジェクト定義を再インポートするには、定義ごとに次のコマンドを実行します。

```
mqsiimpxmlschema -c -f parameter_file -o filename
```

ここで、

parameter_file

は XML スキーマ・インポーターのパラメーター・ファイルの名前です。

filename

は、インポーターがメッセージ・セット名とメッセージ・セット ID を、= 文字で区切って書き込むファイルです。ファイルがすでに存在する場合は、この行がそのファイルに追加されます。

XML スキーマのインポート、および `mqsiiimportxmlschema` コマンドの使用についての詳細は、「*WebSphere MQ Integrator Broker: メッセージの作業*」を参照してください。

3. Business Object Designer と WebSphere MQ Integrator Broker Control Center が、ファイル共有システムを使用しない別々のマシンに存在している場合は、生成されたファイルを、統合ブローカーが動作しているマシンにコピーしてください。
4. WebSphere MQ Integrator Broker の Control Center から、ワークスペースの既存のメッセージ・セットとメッセージを置き換えます。
5. 既存の割り当て済みメッセージ・セットを新しいメッセージ・セットで置換しません。
6. 統合ブローカーでメッセージ・セットを再展開します。

コネクタ構成ファイルの更新

WebSphere Business Integration Adapters の新しいリリースには、コネクタ構成ファイルに対するいくつかの変更が組み込まれています。どのリリースからアップグレードするかに応じて、以下のセクションを参照してください。

リリース 2.3.1 からのアップグレード

WebSphere Business Integration Adapters バージョン 2.3.1 からアップグレードする場合は、新しい標準構成プロパティ `XMLNamespaceFormat` があることに注意してください。このプロパティのデフォルト値は `short` です。

`XMLNamespaceFormat` を `long` にする場合は、コネクタ構成ファイルで、これを単独で指定する必要があります。使用しているメッセージ・ブローカーに応じて、`RFH2messagedomain` プロパティの値も変更する必要があります。正しい `XMLNamespaceFormat` の選択方法については、105 ページの『XML ネーム・スペースの長さの選択』を参照してください。

リリース 2.2.0 からのアップグレード

WebSphere Business Integration Adapters バージョン 2.3.0 以降には、コネクタ構成ファイルに対する以下の変更が組み込まれています。新しいプロパティ `RFH2MessageDomain` が標準構成プロパティに追加されました。このプロパティを使用すると、必要に応じて、メッセージ・ドメインを `xml` に設定することができます (デフォルトは `mrm` です)。詳細については、115 ページの『付録 A. WebSphere MQ のメッセージ・フォーマット』と 125 ページの『付録 C. コネクタの標準構成プロパティ』を参照してください。

新規バージョンの Connector Configurator を使用して既存のコネクタ構成ファイルを開くと、ファイルへの必要な変更が自動的に行われます。ただし、新しいプロパティが存在すること、および目的の値に設定されていることを確認することをお勧めします。これを行うには、以下の操作を実行します。

1. インストールした新しいバージョンの Connector Configurator を使用し、アダプターのコネクタ構成ファイルを開く。
2. 「標準のプロパティ」タブをクリックする。
3. アダプター・ユーザズ・ガイドの指定のとおり、コネクタ固有の追加の構成プロパティの変更を行う。
4. 既存の名前を使用してこのファイルを保管する。

2.2.0 より前のリリースからのアップグレード

WebSphere Business Integration Adapters バージョン 2.2.0 以降には、コネクタ構成ファイルに対する以下の変更が組み込まれています。

- ファイル・フォーマットは XML になりました。
- デリバリー・トランスポート機構は、WMQI-JMS ではなく、JMS と呼ぶことになりました。
- 新規のプロパティ ResponseQueue は、必要ですが、WebSphere MQ Integrator Broker では使用されません。このプロパティには、200 ページの『応答キューの作成』で定義した応答キューの名前を割り当ててください。
- 特定の標準コネクタ構成プロパティが名前変更されました。
- 追加の標準コネクタ構成プロパティとアプリケーション固有のコネクタ構成プロパティが新しく定義されました。
- 既存の標準プロパティを使用して、クライアント・モード・キュー構成を指定できるようになりました。コネクタ構成ファイルを手動で編集する必要はなくなりました。クライアント・モードを指定するために、コネクタ構成ファイルに手動で追加する必要があったメッセージング・セクションは、現行バージョンの Connector Configurator を使用してコネクタ構成ファイルを保管すると自動的に削除されます。

ヒント

Connector Configurator の使用法の詳細や、新規バージョンのコネクタ構成プロパティの変更内容を調べるには、それぞれのアダプターのユーザズ・ガイドを参照してください。

新規バージョンの Connector Configurator を使用して既存のコネクタ構成ファイルを開くと、ファイルへの必要な変更が自動的に行われます。ただし、以下に説明するとおり、変更が正しいかどうかを確認する必要があります。

アップグレード対象の各アダプターごとに、以下の作業を行います。

1. インストールした新しいバージョンの Connector Configurator を使用し、アダプターのコネクタ構成ファイルを開く。
2. 「標準のプロパティ」タブをクリックする。

3. 以下のプロパティの値が、ご使用のビジネス・インテグレーション・システムに合った正しい値であることを確認する。
 - a. CharacterEncoding
 - b. jms.MessageBrokerName (以前は QueueManagerName)
 - c. jms.Password (以前は QueueManagerPassword)
 - d. jms.UserName (以前は QueueManagerLogin)
 - e. jms.NumConcurrentRequests (以前は ConcurrentRequests)
 - f. Locale
 - g. SynchronousRequestTimeout (以前は Timeout)
4. アダプター・ユーザース・ガイドの指定のとおり、コネクタ固有の追加の構成プロパティの変更を行う。
5. 既存の名前を使用してこのファイルを保管する。

新しいコネクタ始動スクリプトの構成

コネクタ用の始動スクリプトは WebSphere Business Integration Adapters 2.2.0 で大幅に変更されたため、新しいバージョンを使用する必要があります。111 ページの『コネクタの始動ファイル、ショートカット、および環境変数の構成』の説明に従って、始動スクリプトをカスタマイズしてください。

重要

WBIA インストーラーの現行バージョンでは、Windows にインストールした場合にのみショートカットが作成されます。CDE ショートカットがすでに存在する場合、それらのショートカットは更新されません。

新規アダプターの構成

新しくインストールしたそれぞれのアダプターごとに、以下のトピックにリストされている構成作業を行ってください。

- 93 ページの『ビジネス・オブジェクト定義の作成』
- 94 ページの『メッセージ・ブローカー・プロジェクトの作成』
- 105 ページの『アプリケーションでコネクタを使用可能にする』
- 106 ページの『コネクタの構成』
- 112 ページの『メッセージ・フローの定義』

新規 WebSphere MQ バッチ・ファイルのカスタマイズ

WebSphere Business Integration Adapters バージョン 2.2.0 ではバッチ・ファイルが提供されています。このバッチ・ファイルを利用して、ご使用のアダプターに必要な WebSphere MQ キューの構成や、必要に応じて WebSphere MQ キューからのメッセージのクリアなどを行うことができます。これらのバッチ・ファイルを使用すると、1 つのコマンドを使ってビジネス・インテグレーション・システム内のすべてのアダプターの WebSphere MQ キューを構成したり、クリアしたりすることができます。これらのバッチ・ファイルのカスタマイズおよび実行の詳細について

は、91 ページの『WebSphere Business Integration Adapters バッチ・ファイルを使用して WebSphere MQ キューを構成する方法』および 78 ページの『WebSphere MQ キューからのメッセージのクリア』を参照してください。

用語集

[ア行]

アダプター (adapter). アプリケーション・ロジックの実行やデータの交換などのタスクを実行するために、統合ブローカー (およびアプリケーションまたはテクノロジー) と通信するソフトウェア・モジュールの集合。

IBM WebSphere Business Integration Adapter は、どのような場合でも、アダプター・フレームワークおよび 1 つのアプリケーション/テクノロジーに固有のコネクターで構成されている。また、アダプターには、アプリケーション/テクノロジーに固有のサンプル・ビジネス・オブジェクトと、アプリケーション/テクノロジーに固有のビジネス・オブジェクト定義を生成するために設計された Object Discovery Agent (ODA) のいずれか一方、または両方が含まれていることもある。

アプリケーション固有のコンポーネント

(application-specific component). 特定のアプリケーションまたはテクノロジーに合わせて調製されたコードを含むコネクター・コンポーネント。このコンポーネントは、要求に回答するためにビジネス・オブジェクト・ハンドラーを初期化する。また、必要に応じて、アプリケーションまたは外部のプログラム・エンティティが開始するイベントを検出してそのイベントに回答するために、イベント通知機構をインプリメントする。このコンポーネントのコードは、アプリケーションまたはテクノロジーが提供する API の言語に応じて、C++ または Java で記述される。

アプリケーション・コネクター (application connector). 特定のアプリケーションと対話するために設計されたコネクター。アプリケーション固有のコネクターは、統合ブローカーとアプリケーションの間を中継する。このコネクターは、アプリケーション固有のデータを、統合ブローカーのコンポーネントが操作できるビジネス・オブジェクトに変換する。また、コンポーネントからのビジネス・オブジェクトを、特定のアプリケーションが受信できるデータに変換する。

[カ行]

コネクター (connector). イベントについての情報を統合ブローカーに送信したり、要求についての情報を統合ブローカーから受信したりするために、ビジネス・オブジェクトを使用するソフトウェア・モジュールの集合 (コネクター・フレームワークとコネクターのアプリケーション固有のコンポーネント)。**アプリケーション・コネ**

クター (application connector) とテクノロジー・コネクター (technology connector) を参照。

コネクター・フレームワーク (connector framework)

(connector framework). コネクターのアプリケーション固有のコンポーネントと統合ブローカーの間の対話を管理するコネクター・コンポーネント。このコンポーネントは、必要な管理サービスをすべて提供し、コネクターが必要とするメタデータをリポジトリから検索する。コネクター・フレームワークのコードは、すべてのコネクターに共通である。コネクター・フレームワークは Java で記述されているが、C++ 拡張機能が含まれるので、アプリケーション固有のコンポーネントの開発を C++ で記述できる。

[タ行]

テクノロジー・コネクター (technology connector).

特定のテクノロジーに準拠する対話向けに設計されたコネクター。例えば、WebSphere Business Integration Adapter for XML を統合ブローカーとの間の中継コネクターとして使用すると、この統合ブローカーは、WebSphere Business Integration システムを実行していないネットワーク上にある Web サーバー (またはその他のプログラマチック・エンティティ) に対しても、XML 形式でデータを送信できる。

統合ブローカー (integration broker). 複数の異種アプリケーション間でデータを統合するプログラム。通常の場合、統合ブローカーが提供する各種のサービスには、データをルーティングする機能、統合プロセスを制御する規則のリポジトリ、各種のアプリケーションへの接続、統合を支援する管理機能などが含まれる。

[ハ行]

ビジネス・インテグレーション・システム (business integration system).

1 つの統合ブローカーと複数の統合アダプターから成り、ビジネス・オブジェクト形式での統合的な情報転送を通じて複数の異種ビジネス・アプリケーション間でのデータ交換を実現するシステム。

ビジネス・オブジェクト (business object). 1 つのビジネス・エンティティ (従業員など)、データに対する 1 つのアクション (作成操作、更新操作など)、およびデータを処理するための命令を表す属性の集合。ビジネス・インテグレーション・システムのコンポーネント

は、情報を交換したりアクションを起動 (トリガー) したりするためにビジネス・オブジェクトを使用する。

[マ行]

文字エンコード (character encoding). 文字 (アルファベット文字) から文字コード・セットの数値へのマッピング。例えば、ASCII 文字コード・セットでは、文字「A」を 65 としてエンコードするが、EBCDIC 文字コード・セットではこの文字を 43 にエンコードする。文字コード・セットには、1 つ以上の言語で使用されるすべての文字に対するエンコードが含まれる。

文字変換 (character conversion). ある文字コード・セットを使用する場所から別のコード・セットを使用する場所に転送した場合にも意味を保持できるようにするために、文字に適用されるエンコード。**文字エンコード (character encoding)** も参照。

[ラ行]

ローカル・リポジトリ (local repository). 複数のアプリケーション間で転送されるデータを持つビジネス・オブジェクトなど、ビジネス・インテグレーション・システム・コンポーネントを記述するメタデータの集合。これには、コネクタ・フレームワークおよびコネクタのアプリケーション固有のコンポーネントに関連する構成情報も含まれる。コネクタのローカル・リポジトリと呼ばれることもある。

ロケール (locale). エンド・ユーザーの特定の国、言語、または地域に固有のデータを処理する方法に関する情報を集約するユーザー環境の一部。通常、ロケールは、オペレーティング・システムまたは国際化されたソフトウェア・プロダクトの構成時に指定する。

A

Adapter Development Kit (ADK). カスタム・アダプターを作成するための開発環境。

Adapter Framework. アダプターをインストール、構成、および実行するために IBM が提供するソフトウェア。

O

Object Discovery Agent development kit (ODK). Object Discovery Agent (ODA) を作成するための API。

Object Discovery Agent (ODA). データ・ソースに固有のビジネス・オブジェクト要件を「発見」し、それ

らの要件からビジネス・オブジェクト定義を生成するために設計されたツール。Business Object Designer は、使用可能な ODA とのフォーム・ベースのインターフェースを表示して、ディスカバリー・プロセスおよび定義生成プロセスの管理を支援する。

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]

アプリケーション固有情報 23, 25
 ビジネス・オブジェクト属性内 23
 ビジネス・オブジェクト動詞内 37
アプリケーション固有のビジネス・オブジェクト 19
アプリケーション固有のプロパティ (コネクタ) 40
アプリケーション・エンティティ 15
イベント
 アーカイブ 34
 受信箱 29
 テキスト 33
イベント通知 27, 34
 コネクタの役割 28
 セットアップ 28
 ビジネス・オブジェクトの役割 15
イベントのアーカイブ 34
インストール
 IBM MQSeries 64
 IBM WebSphere MQ 60
 JDK (Java Development Kit) 60, 62
インテグレーター・ブローカー、配置 102
インポーター、メッセージ・ブローカー 95
エラー・メッセージ
 重大度 88
 タイプ 88
応答ビジネス・オブジェクト 16

[カ行]

階層ビジネス・オブジェクト 18
環境変数
 PATH 63
管理
 参照: CrossWorlds System Manager
基本属性タイプ 21
構成
 コネクタ 40, 41
コネクタ
 イベント通知の振る舞い 27, 34
 開発 41

コネクタ (続き)
 構成 28, 40, 41
 プログラミング 56
 プロパティ 40
 変更 41
 ポーリング 32
 要求処理の振る舞い 40
コネクタ・エージェント
 イベントの検出 32
 イベントの処理 33, 34
 ビジネス・オブジェクトの構成 37, 39
 ポーリング 32
 要求の処理 35, 40
 参照: コネクタ、コネクタ・コントローラ
子ビジネス・オブジェクト 17

[サ行]

システム・ファイル
 /etc/profile 64
指定 102
重大度 (メッセージの) 88
冗長モード 102
スクリプト
 CWSharedEnv.sh 64
属性 (ビジネス・オブジェクト) 17, 20, 21
 アプリケーション固有情報 23
 データ型 21
ソフトウェア要件 59, 61

[タ行]

データ型 (属性) 21
動詞 17, 22
 アプリケーション固有情報 25, 37

[ナ行]

ネーム・スペース認識 105
ネーム・スペース認識 (Namespace aware)、フィールド 102

[ハ行]

配置、インテグレーター・ブローカーへの 102
配置、ビジネス・オブジェクト定義 94

配置、プロジェクト 94
配置、WebSphere Business Integration Message Broker への 99
配置、WebSphere MQ Integrator または WebSphere MQ Integrator Broker への 102
パブリッシュ (パブリック化) およびサブスクライブ ix
汎用ビジネス・オブジェクト 19
ビジネス・オブジェクト 19
 アプリケーション固有情報 23
 アプリケーション固有の 19, 22, 25
 イベント通知として 15
 応答として 16
 階層 18
 子 17
 構成 37
 コンポーネント 16
 属性 17, 20, 21
 属性値 17
 タイプ 16
 デコンストラクション 37
 動詞 17
 の役割 15, 16
 汎用 19
 フラット 18
 マッピング ix
 要求として 16
 参照: ビジネス・オブジェクト定義
ビジネス・オブジェクト定義 19
 コネクタ・ダウンロード 28
ビジネス・オブジェクト定義、作成 93
ビジネス・オブジェクト定義、配置 94
ビジネス・オブジェクトの構成 37, 39
ビジネス・オブジェクトのデコンストラクション 37, 39
標準プロパティ (コネクタ) 40
複合属性タイプ 21
フラット・ビジネス・オブジェクト 18
プロジェクト、配置 94
プロジェクト、メッセージ・ブローカー 97
プロジェクト、ユーザー 97
プロパティ
 コネクタ 40, 41
ベース・メッセージ・セット 101
ベース・メッセージ・セット・プロジェクト 101
ポーリング
 構成可能なプロパティ 32
 コネクタ・エージェントによる 32

[マ行]

メッセージ (CrossWorlds)
 フォーマット 88
メッセージ・セット、プロジェクト 101
メッセージ・セット、ベース 101
メッセージ・セット・プロジェクト、名前
 の指定 101
メッセージ・ブローカー、定義 viii
メッセージ・ブローカー・プロジェクト
 97
メッセージ・ブローカー・プロジェクト、
 作成 94

[ヤ行]

ユーザー・プロジェクト 97
要求処理 40
要求処理 (コネクター) 37
要求ビジネス・オブジェクト 16
要件
 ソフトウェア 59, 61
 Web ブラウザー 60, 62

[ワ行]

ワークスペース、メッセージ・ブローカー
 94, 95
ワークフロー (通知の例) 30

C

CSM (CrossWorlds System Manager) 150
CWSharedEnv.sh スクリプト 64

E

e-business
 アプリケーション統合 4

H

HP-UX オペレーティング・システム 62

I

IBM MQSeries
 インストール 64
 ソフトウェア要件として 62
IBM WebSphere MQ
 インストール 60
 ソフトウェア要件として 60

IBM WebSphere MQSeries
 参照: IBM WebSphere MQ
IBM WebSphere MQ、定義 64

J

Java Database Connectivity
 参照: JDBC
Java Development Kit
 参照: JDK
Java コンパイラー
 参照: JDK
JDK (Java Development Kit) 60
 インストール 60, 62

L

LogViewer
 使用 81
long、XML ネーム・スペース長 105

M

MQSeries
 参照: IBM MQSeries

P

PATH 環境変数 63

R

RFH2messagedomain、XML ネーム・スベ
 ース形式 105

S

short、XML ネーム・スペース長 105
Solaris オペレーティング・システム 62
System Manager 94
System Manager、プロジェクトの配置に使
 用 94

W

Web ブラウザー要件 60, 62
WebSphere Business Integration Message
 Broker、配置 94
WebSphere MQ Integrator Broker、配置
 94
WebSphere MQ Integrator、配置 94
Windows 2000 60

X

XML ネーム・スペース、形式の選択
 105
XMLNamespaceFormat、構成プロパティー
 105

特記事項

本書に記載の製品、サービス、または機能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、サービス、および機能については、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、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、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

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。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更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 Burlingame Laboratory Director
IBM Burlingame Laboratory
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A

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せんが、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が、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。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。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。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

警告: 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

商標

以下は、IBM Corporation の商標です。

IBM
IBM ロゴ
AIX
CrossWorlds
DB2
DB2 Universal Database
Domino
Lotus
Lotus Notes
MQIntegrator
MQSeries
Tivoli
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。

MMX、Pentium および ProShare は、Intel 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。

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です。



IBM WebSphere Business Integration Adapter Framework V 2.4.0



Printed in Japan