

**IBM WebSphere Business Integration Server
Express and Express Plus**



**Adapter for Siebel eBusiness Applications
ユーザース・ガイド**

Adapter バージョン 4.6.x

**IBM WebSphere Business Integration Server
Express and Express Plus**



**Adapter for Siebel eBusiness Applications
ユーザース・ガイド**

Adapter バージョン 4.6.x

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、121 ページの『特記事項』に記載されている情報をお読みください。

本書は、WebSphere Business Integration Adapter for Siebel eBusiness Applications (5724-H43) バージョン 4.6 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 Server Express
and Express Plus
Adapter for Siebel eBusiness Applications
User Guide
Adapter Version 4.6.x

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2005.8

この文書では、平成明朝体™W3、平成明朝体™W7、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™W7、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 2004, 2005. All rights reserved.

© Copyright IBM Japan 2005

目次

本書について	v
本書で扱う項目	v
本書で扱わない項目	v
対象読者	v
関連資料	vi
表記上の規則	vi
本リリースでの新機能	vii
リリース 4.6.x での新機能	vii
第 1 章 概要	1
用語	1
Siebel アプリケーション・アーキテクチャー	2
コネクタ・アーキテクチャー	3
コネクタの動作方法	3
第 2 章 アダプターのインストール	9
アダプター環境	9
アダプターと関連ファイルのインストール	10
インストールの検証	10
イベント表およびアーカイブ表	11
第 3 章 コネクタの構成	19
Connector Configurator Express の概要	19
Connector Configurator Express の始動	20
System Manager からの Configurator の実行	20
コネクタ固有のプロパティ・テンプレートの作成	21
新しい構成ファイルを作成	24
既存ファイルの使用	26
構成ファイルの完成	27
構成ファイル・プロパティの設定	28
構成ファイルの保管	36
構成の完了	36
グローバル化環境における Connector Configurator Express の使用	37
第 4 章 ビジネス・オブジェクトの理解	39
ビジネス・オブジェクトの構造および関係	39
ビジネス・オブジェクトのアプリケーション固有の情報	42
第 5 章 ビジネス・オブジェクトの作成	53
ビジネス・オブジェクトのサンプルの変更	53
Siebel ODA の概要	57
Siebel ODA のインストールおよび使用	57
Business Object Designer Express 内での SiebelODA の使用	61
生成された定義の検討	74
ビジネス・オブジェクト定義への情報の追加	77
第 6 章 Siebel ビジネス・サービスでのアダプターの使用	79
ビジネス・サービスの理解	79
ビジネス・サービスにおける動詞処理	82

ビジネス・サービスにおけるイベント検出	83
カスタム・ビジネス・サービスのサポート	83
第 7 章 コネクタの実行	85
コネクタの始動	85
コネクタの停止	86
第 8 章 トラブルシューティング	87
Siebel ログ・ファイルのサイズの削減	87
結果セットのサポートでのメモリー制限	87
付録 A. コネクタの標準構成プロパティ	89
新規プロパティ	89
標準コネクタ・プロパティの概要	89
標準プロパティの早見表	91
標準プロパティ	96
付録 B. コネクタ固有のプロパティ	113
ApplicationPassword	115
ApplicationUserName	115
ArchiveProcessed	116
ConnectErrors	116
ConnectorID	116
ConnectString	116
DataBeanPoolSize	116
DataBeanRefreshInterval	117
EventProcessingSupport	117
PollAttributeDelimiter	117
PollQuantity	117
ResonateSupport	117
SiebelLanguageCode	117
SiebelVersion	118
SupportNameValuePair	118
UseDefaults	118
ViewMode	118
索引	119
特記事項	121
プログラミング・インターフェース情報	122
商標	123

本書について

製品 IBM® WebSphere Business Integration Server Express および IBM® WebSphere Business Integration Server Express Plus は、InterChange Server Express、関連する Toolset Express、CollaborationFoundation、およびソフトウェア統合アダプターのセットで構成されています。Toolset Express に含まれるツールは、ビジネス・オブジェクトの作成、変更、および管理に役立ちます。プリパッケージされている各種アダプターは、お客様の複数アプリケーションにまたがるビジネス・プロセスに応じて、いずれかを選べるようになっています。標準的な処理のテンプレートである CollaborationFoundation は、カスタマイズされたプロセスを簡単に作成できるようにするためのものです。

IBM® WebSphere® Business Integration Server Express 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、レガシー・システム、およびメインフレーム・システムに統合コネクティビティーを提供します。この製品セットには、ビジネス統合の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やテンプレートが含まれています。

本書で扱う項目

本書では、IBM WebSphere Business Integration Server Express Adapter for Siebel eBusiness Applications のインストール、コネクター・プロパティー構成、ビジネス・オブジェクト開発、およびトラブルシューティングについて説明します。

本書で扱わない項目

本書では、サーバー・ロード・バランシング、アダプター処理スレッドの数、最大および最小スループット、および許容度しきい値などの、配置メトリックおよびキャパシティー・プランニングの問題については説明しません。

このような問題は、お客様の配置ごとに固有であり、アダプターが配置される環境内またはそれに近い環境で評価する必要があります。IBM サービス技術員に連絡を取り、お客様の配置サイトの構成について、およびお客様固有の構成を前提としたこのようなメトリックの計画および評価の詳細について、ご相談ください。

対象読者

本書は、WebSphere Business Integration システムのコンサルタントおよびお客様を対象としています。本書の内容を利用するには、以下の分野の知識が必要です。

- コネクターの開発
- ビジネス・オブジェクトの開発
- Siebel アプリケーション・アーキテクチャー
- Siebel ツール
- Visual Basic

注: 日本のコンサルタントまたはお客様で、Siebel 2000 を使用している場合は、「Adapter for Siebel 2000 ユーザーズ・ガイド」を使用する必要があります。

関連資料

この製品に付属する資料の完全セットでは、すべてのアダプターのインストールに共通する機能とコンポーネントについて説明しています。また、特定のコンポーネントに関する参照資料も含まれています。

本書では、随所で「WebSphere Business Integration Server Express インストール・ガイド (Windows® 版)」および「システム・インプリメンテーション・ガイド」の2つの資料を参照しています。本書を印刷する場合は、これらの資料も印刷すると便利です。

関連文書は、次のサイトからダウンロード、インストール、および表示することができます。 <http://www.ibm.com/websphere/wbiserverexpress/infocenter>

表記上の規則

本書では、以下のような規則を使用しています。

Courier フォント	コマンド名、ファイル名、ユーザーが入力する情報、システムが画面上に出力する情報などのリテラル値を示します。
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示します。
青のアウトライン	青のアウトラインは、マニュアルをオンラインで表示するときのみ見られるもので、相互参照用のハイパーリンクを示します。アウトラインの内側をクリックすると、参照先オブジェクトにジャンプします。
<i>ProductDir</i>	製品のインストール先ディレクトリーを表します。

本リリースでの新機能

リリース 4.6.x での新機能

2005 年 7 月 14 日更新

本書は、以下の変更を含むように更新されました。

- このアダプターは以下のプラットフォームをサポートします。
WebSphere Business Integration Adapter Framework (管理ツールのみ) の場合、
Windows XP (Service Pack 1A を導入済みのもの)
Windows 2003 (Standard Edition または Enterprise Edition)
- アダプターは統合ブローカーの InterChange Server Express をサポートします。

第 1 章 概要

- 『用語』
- 2 ページの『Siebel アプリケーション・アーキテクチャー』
- 3 ページの『コネクター・アーキテクチャー』
- 4 ページの『ビジネス・オブジェクトの処理』
- 7 ページの『イベント管理』
- 8 ページの『Siebel アプリケーションへの接続逸失時の処理』
- 8 ページの『ロケール依存データの処理』

本章では、アダプター用語および WebSphere Business Integration Adapter for Siebel eBusiness Applications の概要を説明します。アダプターをインストールおよび構成してご使用になる前に、本章のトピックをお読みください。

注: この章では、Event ビジネス・コンポーネントと Archive ビジネス・コンポーネント、Event ビジネス・オブジェクトと Archive ビジネス・オブジェクト、およびイベント表とアーカイブ表を参照しています。これらの参照は、以前のバージョンで見られる CW Event と CW Archive への参照、および Siebel 7.5 で見られる IBM Event と IBM Archive への参照と同義です。

用語

本書では、以下の用語を使用します。

アダプター

統合ブローカーとアプリケーション (またはテクノロジー) との間の通信をサポートするコンポーネントを備えている、WebSphere Business Integration システムのコンポーネント。アダプターには、コネクター、メッセージ・ファイル、および構成ツールが必ず含まれています。また、Object Discovery Agent (ODA) やデータ・ハンドラーも含まれていることがあります。

アダプター・フレームワーク

IBM が提供する、アダプターの構成と実行のためのソフトウェア。アダプター・フレームワークには、ランタイム・コンポーネントとして、Java™ ランタイム環境、コネクター・フレームワーク、および Object Discovery Agent (ODA) ランタイムが組み込まれています。コネクター・フレームワークには、コネクターを新規開発するときに必要なコネクター・ライブラリー (C++ および Java) が含まれています。ODA ランタイムには、ODA を新規開発するときに必要な Object Development Kit (ODK) のライブラリーが含まれています。構成用コンポーネントとしては、以下のツールが用意されています。

- Business Object Designer Express
- Connector Configurator Express
- Log Viewer
- System Manager
- System Monitor

- Test Connector
- アダプターに関連付けられた Object Discovery Agent (ODA) (用意されていない場合もあります)

Adapter Development Kit (ADK)

アダプター開発用のサンプルをいくつか備えた開発キット。サンプルには、コネクターと Object Discovery Agent (ODA) のサンプルも含まれます。

コネクター

ビジネス・オブジェクトを使用して、統合ブローカーにイベント関連の情報を送信し (イベント通知)、統合ブローカーから要求関連の情報を受信する (要求処理)、アダプターのコンポーネント。コネクターは、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で構成されています。

コネクター・フレームワーク

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相互作用を管理する、コネクターのコンポーネント。このコンポーネントは、必要な管理サービスをすべて備えており、コネクターが必要とするメタデータをリポジトリから取得します。コネクター・フレームワークは Java で記述されており、C++ で記述されたアプリケーション固有のコンポーネントをサポートできるように C++ 拡張が組み込まれています。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で共通です。

コネクター・コントローラー

コラボレーションと相互作用する、コネクター・フレームワークのサブコンポーネント。コネクター・コントローラーは、InterChange Server Express 内で動作するもので、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・オブジェクトの間のマッピングを開始し、コラボレーションのビジネス・オブジェクト定義に対するサブスクリプションを管理します。

統合ブローカー

異種アプリケーションの間でデータを統合する、WebSphere Business Integration システムのコンポーネント。通常、InterChange Server Express などの統合ブローカーはさまざまなサービスを備えています。このサービスには、データをルーティングする機能、統合プロセスを決定する規則のリポジトリ、各種アプリケーションに接続する機能、および統合を容易にする管理機能が含まれます。

WebSphere Business Integration システム

多様なソースの間で情報を移動してビジネス関連の情報を交換し、エンタープライズ環境内の異種アプリケーションの間で情報の処理とルーティングを行う、エンタープライズ・ソリューション。このビジネス・インテグレーション・システムは、統合ブローカーと 1 つ以上のアダプターで構成されています。

Siebel アプリケーション・アーキテクチャー

Siebel アプリケーション・アーキテクチャーには、以下のように 3 つのレイヤーが含まれています。

- ユーザー・インターフェース・オブジェクト・レイヤー -- このレイヤーには、ユーザーが対話する表示要素が含まれています。

- ビジネス・オブジェクト・レイヤー -- このレイヤーには、ビジネス・コンポーネントとビジネス・オブジェクトの両方が含まれています。ビジネス・コンポーネントは基本的なビジネス・エンティティーで、それを表現する複数のフィールドからなります。ビジネス・オブジェクトは、関連するビジネス・コンポーネントの集合です。Siebel コネクターは、Siebel Java Data Bean を使用してこのレイヤーと通信します。
- データ・オブジェクト・レイヤー -- このレイヤーには、基礎となる物理データベースの論理的な表現を行うオブジェクト定義が含まれています。このレイヤーはインストールされたりレシーショナル・データベース管理システムから独立しており、Siebel Java Data Bean からはアクセスできません。

コネクター・アーキテクチャー

コネクターは、「コネクター開発ガイド (Java 用)」で概説されているメタデータ設計原則に従って設計されています。つまり、既存のアプリケーション固有のビジネス・オブジェクトを拡張したりカスタマイズすることができるほか、コネクター・コードでさらにコーディングまたはカスタマイズを行わなくても新しいビジネス・オブジェクトを定義することができます。

以下の図は、Siebel コネクター・アーキテクチャーを示したものです。

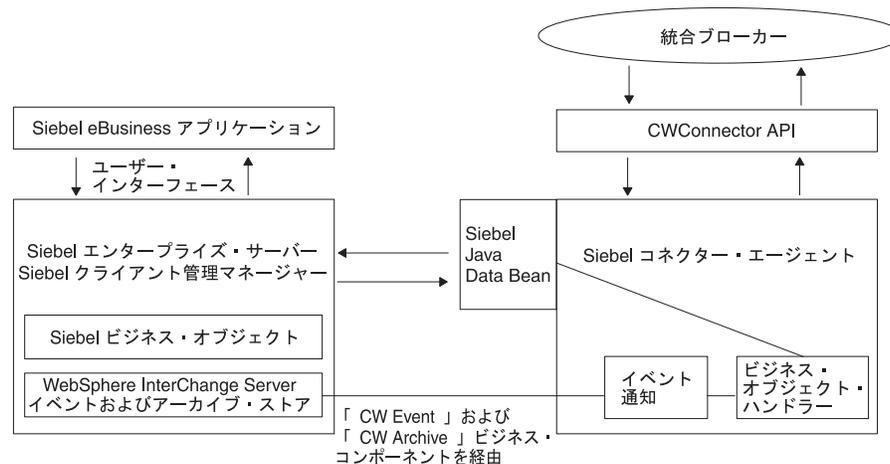


図 1. Siebel コネクター・アーキテクチャー

コネクターの動作方法

ここでは、メタデータでコネクターの柔軟性を向上する方法について説明し、ビジネス・オブジェクトの処理とイベント通知に関する高度な説明を行います。

コネクタールおよびメタデータ

コネクタールはメタデータ主導型です。メタデータはビジネス・オブジェクトに格納されるアプリケーション固有のデータで、コネクタールとアプリケーションの対話を支援します。メタデータ主導型コネクタールは、コネクタールでハードコーディングされた命令でなくビジネス・オブジェクト定義でエンコードされたメタデータに基づいて、サポートする各ビジネス・オブジェクトを処理します。ビジネス・オブジェクトは、Siebel ビジネス・コンポーネントに対応します。ビジネス・オブジェクトの詳細については、79 ページの『第 6 章 Siebel ビジネス・サービスでのアダプターの使用』を参照してください。

ビジネス・オブジェクトのメタデータには、ビジネス・オブジェクトの構造、属性プロパティの設定、アプリケーション固有の情報の内容などがあります。コネクタールはメタデータ主導型なので、コネクタール・コードを変更しなくても、新しいビジネス・オブジェクトまたは変更されたビジネス・オブジェクトを処理することができます。

ビジネス・オブジェクトの処理

ここでは、コネクタールが統合ブローカーの要求とアプリケーション・イベントを処理する方法について概説します。

統合ブローカー要求の処理

コネクタールがビジネス・オブジェクトからの要求を受信してアプリケーションの操作を行う場合、コネクタールは階層型ビジネス・オブジェクトを反復処理します。すなわち、個々のビジネス・オブジェクトをすべて処理するまでそれぞれの子ビジネス・オブジェクトに対して同じ手順を実行します。

注: **階層型**ビジネス・オブジェクトという用語は、どのようなレベルでも収容できるすべての子ビジネス・オブジェクトを持つ 1 つの完全なビジネス・オブジェクトを指します。**個々の**ビジネス・オブジェクトという用語は、それが収容している、あるいは収容されている、どの子ビジネス・オブジェクトからも独立した 1 つのビジネス・オブジェクトを指します。**トップレベルの**ビジネス・オブジェクトという用語は、それ自体が親ビジネス・オブジェクトを持たない階層のトップにある個々のビジネス・オブジェクトを指します。

ビジネス・オブジェクトの検索: 統合ブローカーが Siebel アプリケーションから階層型ビジネス・オブジェクトの検索をコネクタールに要求すると、コネクタールは Siebel ビジネス・コンポーネント・インスタンスの現行の表現に正確に一致する統合ブローカーにビジネス・オブジェクトを戻そうとします。言い換えると、統合ブローカーに戻された個々のビジネス・オブジェクトの単純な属性はすべて、Siebel ビジネス・コンポーネント内の対応するフィールドの値に一致します。

完全なビジネス・コンポーネントを検索するために、コネクタールは統合ブローカーから受信したトップレベル・ビジネス・オブジェクト内の基本キー値を使用して、データベースの対応するデータ内を再帰的に下位レベルに向かって検索します。

ビジネス・オブジェクトの内容による検索 (RetrievalByContent): 統合ブローカーがトップレベル・ビジネス・オブジェクト内の非キー属性の値に基づいた階層型ビジネス・オブジェクトの検索をコネクタールに要求すると、コネクタールはデータ検索のための基準としてすべての非ヌル属性の値を使用します。

ビジネス・オブジェクトの作成: 統合ブローカーが Siebel アプリケーション内の階層型ビジネス・オブジェクトの作成をコネクターに要求すると、コネクターは親を作成する前にトップレベル・ビジネス・オブジェクトのすべての子を作成します。この規則の例外は、親と子の関係が Siebel 内の複数値リンクであり、そのリンクが非アクティブである場合です。この場合、親の後に子が作成され、キーは Siebel アプリケーションによって作成されます。

ビジネス・オブジェクトの変更: ビジネス・オブジェクトの変更または更新では、Siebel からのビジネス・オブジェクトの検索後イメージをインバウンド・ビジネス・オブジェクトと比較します。このプロセスでは、子オブジェクトに正しい動詞を設定します。キーが親に設定され、他のすべての属性が CxIgnore に設定されると、親の更新は省略されます。

デフォルトの振る舞いでは、Siebel アプリケーションからの検索後イメージをインバウンド・ビジネス・オブジェクトと比較し、次に子のコンテナ・オブジェクト上の動詞を変更します。このプロセスでは、Siebel アプリケーション内のすべての子がインバウンド・ビジネス・オブジェクトと同じものになるようにします。動詞が子に設定されない場合、デフォルトは Update に設定されます。

重要: 子のいくつかを保存しなければならない場合は、インバウンド・オブジェクト動詞を DeltaUpdate に設定し、動詞をそれぞれの子コンテナ・オブジェクトに設定する必要があります。この場合、Siebel アプリケーション内のオブジェクトのみ処理され、その他のオブジェクトは処理されないままになります。

ビジネス・オブジェクトの削除: 統合ブローカーがレコードの削除をコネクターに要求すると、レコードは基礎となるデータベースから削除されます。Siebel DeleteCascade 機能がすべての子を削除するので、親のみを削除してください。必要な属性がすべてインバウンド・ビジネス・オブジェクトにないと、削除は行えません。

Exists 動詞: 基本ビジネス・コンポーネント名は一般に、Siebel 内のビジネス・オブジェクト名と同じです。ObjectName と ComponentName のアプリケーション固有の情報が一致すると、キーはこのビジネス・コンポーネントに設定され、照会が実行されます。レコードが存在する場合は True を戻します。レコードが存在しない場合は False を戻します。

アプリケーション・イベントの処理

コンポーネント: イベント通知では、Siebel データベースにイベント表とアーカイブ表を作成する必要があります。これらの 2 つの表に対応する新しい Siebel ビジネス・コンポーネントである Event と Archive を作成する必要があります。

トリガー: Siebel eBusiness アプリケーションにおけるレコードの作成、更新、または削除の操作は、イベントとして処理できます。Siebel は、イベント表を取り込むために、Siebel ビジネス・コンポーネント・イベント・ハンドラーへの組み込み Visual Basic スクリプトと組み込み Siebel eScripts をサポートします。pollForEvents への呼び出し時に、これらのレコードが取得、処理されます。イベント・ビジネス・コンポーネントは、表 1 にリストされているように、イベントに関する情報を格納します。

注: 表 1 の情報は、対応するビジネス・オブジェクトを構築し、処理のためにこれらのオブジェクトをコネクタ・フレームワークに送る目的で、コネクタがイベント・サブスクリプション時に使用します。

表 1. イベント・ビジネス・コンポーネント構造

フィールド	説明
Object Key	イベントが作成されたビジネス・オブジェクト行を識別する固有 ID
Object Name	イベントが削除された Siebel ビジネス・オブジェクト
Object Verb	イベントの動詞
Priority	イベントの優先順位
Status	イベントの初期状況は <code>READY_FOR_POLL</code> に設定されます。その他の状況値には以下のものがあります。 <code>IN_PROGRESS=1</code> -- イベントが選択されてコネクタ・フレームワークに送られます。コネクタは、処理のためにイベントを選択した後にイベント状況を <code>IN_PROGRESS</code> に変更します。 <code>UNSUBSCRIBED=2</code> -- イベントはサブスクライブされていません。 <code>isSubscribed</code> 呼び出しが <code>False</code> を戻すと、コネクタは状況を <code>UNSUBSCRIBED</code> に設定します。 <code>SUCCESS=3</code> -- イベントは、コネクタ・フレームワークによって正常に処理されました。コネクタ・フレームワークによりイベントが正常に処理されると、コネクタは状況を <code>SUCCESS</code> に設定します。 <code>ERROR_PROCESSING_EVENT=-1</code> -- イベントの処理でエラーが発生しました。この状況は、イベント処理中にエラーがあった場合に設定されます。 <code>ERROR_POSTING_EVENT=-2</code> -- コネクタ・フレームワークにイベントを転送する際にエラーがありました。この状況は、 <code>gotApp1Event</code> への呼び出しが <code>pollForEvents</code> で失敗した場合に設定されます。 <code>ERROR_OBJECT_NOT_FOUND=-3</code> -- イベントが作成されたオブジェクトを見つけることができませんでした。この状況は、 <code>doVerbFor</code> 呼び出しで <code>pollForEvents</code> 内にオブジェクトを見つけることができなかった場合に設定されます。
説明	イベントに関連したコメント
Event Id	イベント行の ID
ConnectorId	多重結合子構成でコネクタを識別します。
Event Ts	イベント作成のタイム・スタンプ

Create 通知: Create イベントが発生した場合、コネクタはそのイベントで指定されたタイプのビジネス・オブジェクトを作成し、ビジネス・オブジェクトにキー値を設定して (Event ビジネス・コンポーネントで指定されたオブジェクト・キーを使用して行う)、Siebel アプリケーションからビジネス・オブジェクトを検索します。コネクタはビジネス・オブジェクトを検索した後、Create 動詞とともに統合ブローカーに送ります。

Update 通知: Update イベントが発生した場合、コネクタはそのイベントで指定されたタイプのビジネス・オブジェクトを作成し、ビジネス・オブジェクトにキー値を設定して (Event ビジネス・コンポーネントで指定されたオブジェクト・キーを使用して行う)、データベースからビジネス・オブジェクトを検索します。コネクタはビジネス・オブジェクトを検索した後、Update 動詞とともに統合ブローカーに送ります。

Delete 通知: Delete イベントが発生した場合、コネクタはそのイベントで指定されたタイプのビジネス・オブジェクトを作成し、ビジネス・オブジェクトにキー値

を設定して (Event ビジネス・コンポーネントで指定されたオブジェクト・キーを使用して行う)、Delete 動詞とともに統合ブローカーに送ります。キー値以外のすべての値は、CxIgnore に設定されます。

イベント処理用ビジネス・オブジェクトの検索: イベント処理のためのオブジェクトの検索は、キー属性と非キー属性の両方に基づいて行います。必要なのは、ビジネス・オブジェクトが RetrieveByContent 動詞をサポートすることです。

イベント管理

コネクタのイベント検出機構では、Event ビジネス・コンポーネントと Archive ビジネス・コンポーネントを使用します。イベント処理に関連する潜在的な障害点があるため、イベント管理プロセスでは、イベントが Archive ビジネス・コンポーネントに挿入されるまで Event ビジネス・コンポーネントからイベントを削除しません。

コネクタは、設定可能な一定間隔で Event ビジネス・コンポーネントをポーリングしてイベントを検索し、最初は優先順位に従ってイベントを処理し、それ以降はイベントを順次処理します。コネクタがイベントを処理すると、イベントの状況がそれに応じて更新されます。

ArchiveProcessed プロパティの設定によって、コネクタがイベント状況を更新した後にイベントを Archive ビジネス・コンポーネントにアーカイブするかどうかが決まります。ArchiveProcessed プロパティに関する詳細については、116 ページの『ArchiveProcessed』を参照してください。

表 2 は、ArchiveProcessed プロパティの設定に基づいたアーカイブの振る舞いを示しています。

表 2. アーカイブ時の振る舞い

アーカイブ処理設定	イベント処理の状況	コネクタの振る舞い
true または値なし	成功	イベントが Events ビジネス・コンポーネントから削除され、Success の状況でアーカイブされる
	失敗 ビジネス・オブジェクトのサブスクリプションなし	Error の状況でアーカイブされる イベントが Events ビジネス・コンポーネントから削除され、Error Processing Event Error Posting Event Error Object Not Found のいずれかの状況でアーカイブされる
false	成功	アーカイブされず、Success の状況で Events ビジネス・コンポーネント内に残る
	失敗	イベントがアーカイブされず、Error Processing Event Error Posting Event Error Object Not Found のいずれかの状況で Events ビジネス・コンポーネント内に残る
	ビジネス・オブジェクトのサブスクリプションなし	Unsubscribed の状況でイベント表に残る

スマート・フィルター

重複したイベントはイベント・ストアに保存されません。イベント・ストアに新しいイベントをレコードとして格納する前に、VB Script または eScript が新しいイベントに一致する既存のイベントの有無についてイベント・ストアに照会する必要があります。イベント検出機構は、以下の場合に新しいイベントのレコードを生成しません。

- 新しいイベント内のビジネス・オブジェクト名、動詞、状況、および ConnectorId (適用可能な場合) がイベント・ストア内の別の未処理イベントのものと一致する場合。
- 新しいイベントのビジネス・オブジェクト名、キー、および状況がイベント表内の未処理イベントに一致し、新しいイベントの動詞が Update で未処理イベントの動詞が Create である場合。
- 新しいイベントのビジネス・オブジェクト名、キー、および状況がイベント表内の未処理イベントに一致し、イベント表内の未処理イベントの動詞が Create で新しい動詞が Delete である場合。この場合、イベント・ストアから Create レコードを除去します。

Siebel アプリケーションへの接続逸失時の処理

コネクタが終了するのは、ConnectErrors コネクタ・プロパティーで指定されたエラー・メッセージが検出された場合です。ConnectErrors からのテキストは、Siebel エラー・メッセージと比較されます。一致したものが見つかり、コネクタが AppResponseTimeout を戻し、コネクタが終了します。

接続が失われているのにコネクタが以下のことを試行すると、Siebel アプリケーションによって ConnectErrors メッセージが戻されることがあります。

- Event および Archive ビジネス・コンポーネントにアクセスする
- イベントに関連するビジネス・オブジェクトを検索する
- ビジネス・オブジェクトに関連するレコードを作成または更新する

ロケール依存データの処理

コネクタは、2 バイト文字セットをサポートし、指定された言語でメッセージ・テキストを送達できるように国際化されています。ある文字コードを使用するロケーションから別の文字コード・セットを使用するロケーションへデータを転送する場合、コネクタは、そのデータの意味が伝わるように文字変換を実行します。

Java 仮想マシン (JVM) 内での Java ランタイム環境は、Unicode 文字コード・セットでデータを表します。Unicode には、ほとんどの既知の文字コード・セット (単一バイトおよびマルチバイトの両方) の文字エンコード方式が組み込まれています。IBM WebSphere Business Integration システムのほとんどのコンポーネントは Java で作成されています。そのため、Server Access コンポーネント間でデータを転送するときは、ほとんどの場合文字変換は必要ありません。エラー・メッセージと通知メッセージを適切な言語で適切な国および地域に対して記録するには、該当する環境の Locale 標準構成プロパティーを設定します。これらのプロパティーの詳細については、89 ページの『付録 A. コネクタの標準構成プロパティー』を参照してください。

第 2 章 アダプターのインストール

- 『アダプター環境』
- 10 ページの『アダプターと関連ファイルのインストール』
- 10 ページの『インストールの検証』
- 11 ページの『イベント表およびアーカイブ表』

この章では、WebSphere® Business Integration Adapter for Siebel eBusiness Applications をインストールする方法について説明します。

注: この章では、Event ビジネス・コンポーネントと Archive ビジネス・コンポーネント、Event ビジネス・オブジェクトと Archive ビジネス・オブジェクト、およびイベント表とアーカイブ表を参照しています。これらの参照は、以前のバージョンで見られる CW Event と CW Archive への参照、および Siebel 7.5 で見られる IBM® Event と IBM Archive への参照と同義です。

アダプター環境

アダプターをインストール、構成、および実行する前に、環境要件を理解する必要があります。

アダプターのプラットフォーム

このアダプターは、以下のオペレーティング・システムのいずれかを必要とします。

- **Windows:**
WebSphere Business Integration Adapter Framework (管理ツールのみ) の場合、Windows XP (Service Pack 1A を導入済みのもの)
Windows 2003 (Standard Edition または Enterprise Edition)

アダプターの依存関係

コネクターを使用する前に、以下のことを行ってください。

- 使用する Siebel 6.2.x、Siebel 7.0.x、Siebel 7.5.x、または Siebel 7.7x .jar ファイルをインストールする。
- アプリケーションにユーザー・アカウントが存在するかどうかを確認する。このユーザー・アカウントは、Siebel Tools でイベント作成用の Siebel スクリプトに指定したユーザーと同じでなければなりません。
- Siebel の Connector.txt ファイルを %ProductDir%/connectors/messages/Siebel ディレクトリーから %ProductDir%/connectors/messages ディレクトリーにコピーする。

ユーザーのセットアップ

コネクターをインストールする前に、Siebel にコネクターのユーザー・アカウントを作成してください。このユーザー・アカウントは全アクセス権限を持ち、ログイ

ン名は ApplicationUserName 構成プロパティと同じでなければなりません。ユーザー・アカウントのログイン名とパスワードのデフォルト値は CWCONN です。

コネクタをインストールする場合は、必ず、該当する以下のいずれか 1 つのリストに記載されているファイルを %ProductDir%/Connectors/Siebel/dependencies ディレクトリにインストールしてください。ファイルは、Siebel 6 または Siebel 7 サーバー上にあります。

重要: %ProductDir%/Connectors/Siebel ディレクトリ内の start_Siebel.bat ファイルには、現在、JCLASSES 変数内に英語と日本語の Siebel.jar ファイルがあります。このファイルは CLASSPATH に追加されます。Siebel でサポートされるその他の言語については、対応する.jar ファイルを JCLASSES 変数に追加する必要があります。

Siebel 6

- SiebelDataBean.jar
- SiebelTC_enu.jar
- SiebelTcCommon.jar
- SiebelTcOM.jar

Siebel 7.0x または 7.5x

- SiebelJI_Common.jar
- SiebelJI_enu.jar

Siebel 7.7

- Siebel.jar
- SiebelJI_enu.jar

アダプターと関連ファイルのインストール

WebSphere Business Integration Server Express Adapter 製品のインストールについては、「*WebSphere Business Integration Server Express インストール・ガイド (Windows 版)*」を参照してください。このガイドは、WebSphere Business Integration Server Express Adapters Infocenter (<http://www.ibm.com/websphere/wbiserverexpress/infocenter>) にあります。

インストールの検証

このセクションでは、本製品が Windows システムにインストールされた後のファイル構造について説明します。

インストールされるファイルの検証 (Windows システムの場合)

Windows システムでインストールを検証するには、アダプターのインストール先ディレクトリにあるファイルと、表 4 にリストされたファイルを比較します。表 3 は、コネクタが使用する Windows ファイル構造を示しています。

表3. コネクタ用としてインストールされた Windows ファイルのファイル構造

<code>%ProductDir%</code> のサブディレクトリ	説明
<code>¥connectors¥Siebel</code>	コネクタ用の <code>CWSiebel.jar</code> および <code>start_Siebel.bat</code> ファイルが含まれています。
<code>¥connectors¥Siebel¥dependencies</code>	Siebel eBusiness アプリケーションにおけるイベント管理用のパッチ・ファイルが含まれています。このフォルダーには、 <code>Siebel.jar</code> ファイルも含まれています。
<code>¥connectors¥messages¥Siebelconnector.txt</code>	関連するメッセージ・ファイル <code>SiebelConnector.txt</code> が含まれています。
<code>¥connectors¥Siebel¥Samples¥Repository</code>	以下の BO サンプルが含まれています。 <code>Siebel_BCAccount</code> <code>Siebel_BCQuote</code> <code>Siebel_BCContact</code> <code>Siebel_BCInternalProduct</code> <code>Siebel_BCAsset</code>
<code>¥repository¥Siebel¥BIA_CN_Siebel.txt</code>	リポジトリ/ <code>Siebel</code> 。 <code>CN_Siebel.txt</code> ファイルが含まれています。
<code>¥lib</code>	<code>WBIA.jar</code> ファイルが含まれています。
<code>¥bin</code>	<code>CWConnEnv.bat</code> ファイルが含まれています。
<code>¥connectors¥Siebel¥start_Siebel_service.bat</code>	コネクタ・サービスの始動スクリプト。

インストーラーは、コネクタ・ファイルのアイコンを WebSphere Business Integration のメニューに追加します。コネクタをすばやく始動するには、このファイルへのショートカットをデスクトップに作成してください。

注: WebSphere Business Integration のインストーラーの詳細については、「システム・インストール・ガイド (Windows 版)」を参照してください。

イベント表およびアーカイブ表

コネクタは、イベント表を使用してピックアップのためのイベントをキューに登録します。ArchiveProcessed プロパティを true または値なしに設定した場合、コネクタはアーカイブ表を使用してイベント表でイベントの状況を更新した後にイベントを格納します。

コネクタはイベントごとに、Event ビジネス・コンポーネントからビジネス・オブジェクトの名前、動詞、およびキーを取得します。また、この情報を使用してアプリケーションからエンティティ全体を取り出します。イベントが最初にログに記録された後でエンティティが変更されると、コネクタは初期イベントとその後のすべての変更を取得します。つまり、コネクタがイベント表からエンティティを取得する前にエンティティが変更および更新されると、コネクタは 1 回の取り出しで両方のデータ変更を取得します。

コネクタで処理された各イベントに関して、次の 3 種類の結果が発生する可能性があります。

- イベントが正常に処理された。
- イベントが正常に処理されなかった。
- イベントがサブスクライブされなかった。

コネクタがイベントを取り出した後でイベント表からイベントが削除されないと、不要なスペースを占有することになります。ただし、イベントが削除されると、処理されていないすべてのイベントが失われ、イベントの処理を監査することができません。したがって、アーカイブ表を作成し、ArchiveProcessed プロパティを true に設定したままにしてください。イベントがイベント表から削除されると、コネクタはイベントをアーカイブ表に挿入します。

イベントおよびアーカイブ処理の構成

イベントおよびアーカイブ処理を構成する場合は、構成プロパティを使用して以下の情報を指定してください。

- 間隔頻度
- ポーリング間隔ごとのイベント数
- アンサブスクライブされた未処理のイベントをコネクタがアーカイブするかどうか
- 多重結合子が同じ表をポーリングするときに重要であるコネクタの固有な ID

Siebel バージョン 7.5 および 7.7 のイベント表とアーカイブ表の作成

この手順では、例として Siebel Sales Enterprise アプリケーションを使用しています。参照先が Siebel Sales Enterprise となっている箇所は、すべて、ご使用の Siebel アプリケーションの名前で置き換えてください。

イベント表とアーカイブ表を作成し、ビジネス・オブジェクトを起動するには、以下の手順を実行します。

1. すべての現行プロジェクトがチェックインされたことを確認します。
 - Siebel Sales Enterprise プロジェクト
 - Account プロジェクトなど、変更したいオブジェクトを含むプロジェクト

注: プロジェクトがローカル・サーバーと開発サーバーの両方でロックされていることを確認してください。

2. 以下の順序で 6 つのパッチ・ファイルをローカル・データベースに適用します。
 - ibmtable.sif
 - ibmview.sif
 - ibmapplet.sif
 - ibmbo.sif
 - ibmbc.sif
 - ibmcreen.sif

WebSphere Business Integration システムのパッチ・ファイルを日本語環境に適用する場合は、すべてのパッチ・ファイルを以下のように編集します。

各ファイルの最初の行

```
<xml version="1.0" encoding="windows-1252"?>
```

を、次のように編集してください。

```
<xml version="1.0" encoding="Shift_JPN"?>
```

「ENU」言語設定のインスタンスはすべて「JPN」に置き換えます。テキスト・エディターの検索/置換機能を使用する場合には、言語設定を引用符で囲み、似たワード (MENU など) が置換されないようにしてください。

3. プロンプトが表示されたら、ローカル・データベースで IBM Audit プロジェクトをロックします。
4. 以下が作成されていることを確認します。
 - 2 つの新しい表 CX_IBM_ARCH_Q と CX_IBM_Event_Q
 - 1 つの新しいビジネス・オブジェクト IBM Events
 - 1 つの新しいビジネス・オブジェクト Schema version
 - 2 つの新しいビジネス・コンポーネント IBM Archive と IBM Events
 - 1 つの新しいビュー IBM Event List View
 - 2 つの新しいアプレット IBM Archive List Applet と IBM Event List Applet
 - 1 つの新しい画面 IBM Events と 1 つの新しい画面ビュー IBM Event List ビュー
5. 以下のようにページ・タブを作成します。
 - a. 「アプリケーション (Application)」 > 「Siebel Sales Enterprise」 > 「ページ (Page)」 タブの順にアクセスします。
 - b. 右マウス・ボタン・クリックして、メニューから「新規レコード (New Record)」を選択します。
 - c. 画面名に IBM Events を入力し、テキスト名に IBM Events を入力します。
 - d. シーケンスの場合は、シーケンス番号以外の部分より大きい番号を入力します。この選択で、タブがアプリケーション内のどこに表示されるかが決まります。
 - e. 非アクティブなフィールドのチェックを外したままにします。
 - f. 「ページ」タブのロケールに移動し、IBM Events の新しいレコードを作成します。言語コードとして ENU を追加し、テキストがない場合はテキストとして IBMEvents を追加します。
6. 以下のように画面メニュー項目を作成します。
 - a. 「アプリケーション (Application)」 > 「Siebel Sales Enterprise」 > 「画面 (Screen)」メニュー項目の順にアクセスします。
 - b. 右マウス・ボタン・クリックして「新規レコード (New Record)」を選択します。
 - c. 画面名に IBM Events を入力し、テキスト名に IBM Events を入力します。
 - d. シーケンスの場合は、シーケンス番号以外の部分より大きい番号を入力します。この選択で、タブがプルダウン・メニューのどこに表示されるかが決まります。
 - e. 非アクティブなフィールドのチェックを外したままにします。

- f. 画面メニュー項目のロケールに移動し、IBMEvents の新しいレコードを作成します。言語として ENU を追加し、テキストがない場合はテキストとして IBMEvents を追加します。
7. サイトで使用するビジネス・オブジェクトに対応するビジネス・コンポーネントとして Siebel VB スクリプトを追加または変更します。Siebel VB スクリプトは、ビジネス・オブジェクトのイベント通知を起動します。
 - 優先順位別にイベントをソートしたい場合は、コンパイルする前にビジネス・オブジェクト VB スクリプト内で優先順位の値を編集します。
 - 複数のコネクタをインストールする場合は、VB スクリプト内に Connector Id を設定し、アクティブにします。
8. 新しい表の物理スキーマをローカル・データベースに適用します。これを行うには、2 つの新しい表 CX_IBM_ARCH_Q と CX_IBM_EVENT_Q を照会し、現行の照会を選択して物理スキーマを作成します。表スペースと索引スペースをブランクのままにします。
9. 活動化ボタンを使用して新しいスキーマをアクティブにします。
10. ローカル・データベース上で更新済みプロジェクトとロックされたプロジェクトをコンパイルして、新しい Siebel リポジトリ (.srf) ファイルを作成します。
11. ローカル・データベースで Siebel Sales Enterprise を開きます。以下の操作を行うには、管理特権が必要です。
 - a. IBM Event List View という新しいビューを作成します。ヒント: ツールからビュー名をコピーし、View Name フィールドに貼り付けます。
 - b. IBM Event List View に対し、IBM Responsibility という新しい責任を作成します。
 - c. イベントを検討する責任を持つ社員またはチームを、新たに作成された IBM Responsibility に追加します。
 - d. CWCONN ユーザーを作成し、IBM Responsibility および Administrative Responsibility に追加します。
12. ローカル環境でアプリケーションをテストします。IBM Event List View に対する可視性があるかどうか、また、サポートされるオブジェクトを作成した後にイベントがビューに作成されるかどうかを確認します。例えば、Siebel に新しいアカウントを作成し、新しいアカウント・イベントが IBM Event List View に表示されるかどうかを確認します。
13. 以下の更新済みプロジェクトとロックされたプロジェクトを開発サーバーにチェックインします。
 - IBM Audit
 - Siebel Sales Enterprise
 - 使用するビジネス・オブジェクトのプロジェクト

注: 照会によってのみロックされたプロジェクトをチェックインしてください。

14. 開発データベースに物理スキーマを適用します。これを行うには、2 つの新しい表 CX_IBM_ARCH_Q と CX_IBM_EVENT_Q を照会し、現行の照会を選択して物理スキーマを作成します。表スペースと索引スペースをブランクのままにします。

15. 開発データベースで照会済み表をアクティブにします。
16. 適宜、テストおよび実稼働環境に移動します。
17. 新たにコンパイルした Siebel.srf ファイルをサーバーに移動します。

注: 「サイト・マップ (Sitemap)」 > 「サーバー管理 (Server Administration)」 > 「コンポーネント・グループ (Component Group)」
と移動して「使用可能」を選択して、Enterprise Application Integration を使用可能にします。

Siebel JAVABean の設定方法は、以下のとおりです。

1. 「サイト・マップ (Site Map)」 -> 「サーバー管理 (Server Administration)」 -> 「コンポーネント (セールス・オブジェクト・マネージャー) (Components (Sales Object Manager))」 と選択します。
2. 下部アプレットの「コンポーネント・パラメーター (Component Parameter)」 に移動して、タイムアウト値を入力します。

注: 「要求タイムアウト (Request Timeout)」の現行値は 600 に設定されています。これは、コネクタが 10 分で停止するというを示します。Siebel に基づいて、この値を必要に応じて任意の値に変更できます。

Siebel バージョン 7.5 より前のイベント表とアーカイブ表の作成

この手順では、例として Siebel Sales Enterprise アプリケーションを使用しています。参照先が Siebel Sales Enterprise となっている個所は、すべて、ご使用の Siebel アプリケーションの名前で置き換えてください。

イベント表とアーカイブ表を作成し、ビジネス・オブジェクトを起動するには、以下の手順を実行します。

1. すべての現行プロジェクトがチェックインされたことを確認します。
2. ローカル・データベースで、以下のファイルをチェックアウトし、ロックします。
 - 新規表プロジェクト
 - Siebel Sales Enterprise プロジェクト
 - Account プロジェクトなど、変更したいオブジェクトを含むプロジェクト
 - Dock プロジェクト

注: プロジェクトがローカル・サーバーと開発サーバーの両方でロックされていることを確認してください。

3. 以下の順序で 7 つのパッチ・ファイルをローカル・データベースに適用します。
 - cwtable.sif
 - cwview.sif
 - cwapplet.sif
 - cwbo.sif
 - cwbc.sif
 - cwdo.sif

- cwscreen.sif
- schemabo.sif

WebSphere Business Integration システムのパッチ・ファイルを日本語環境に適用する場合は、すべてのパッチ・ファイルを以下のように編集します。

各ファイルの最初の行

```
<xml version="1.0" encoding="windows-1252"?>
```

を、次のように編集してください。

```
<xml version="1.0" encoding="Shift_JPN"?>
```

「ENU」言語設定のインスタンスはすべて「JPN」に置き換えます。テキスト・エディターの検索/置換機能を使用する場合には、言語設定を引用符で囲み、似たワード (MENU など) が置換されないようにしてください。

4. プロンプトが表示されたら、ローカル・データベースで CW Audit プロジェクトをロックします。
5. 以下が作成されていることを確認します。
 - 2 つの新しい表 CX_CW_Archive_Q と CX_CW_Event_Q
 - 1 つの新しいビジネス・オブジェクト Events
 - 1 つの新しいビジネス・オブジェクト Schema version
 - 2 つの新しいビジネス・コンポーネント Archive と Events
 - 1 つの新しいビュー Event List View
 - 2 つの新しいアプレット Archive List Applet と Event List Applet
 - 1 つの新しい画面 Events と 1 つの新しい画面ビュー Event List ビュー
 - 2 つの新しいドック・オブジェクト CX_CWArchive と CX_CWEvent
6. 以下のようにページ・タブを作成します。
 - a. 「アプリケーション (Application)」 > 「Siebel Sales Enterprise」 > 「ページ (Page)」 タブの順にアクセスします。
 - b. 右マウス・ボタン・クリックして、メニューから「新規レコード (New Record)」を選択します。
 - c. 画面名に CW Events を入力し、テキスト名に IBM Events を入力します。
 - d. シーケンスの場合は、シーケンス番号以外の部分より大きい番号を入力します。この選択で、タブがアプリケーション内のどこに表示されるかが決まります。
 - e. 非アクティブなフィールドのチェックを外したままにします。
 - Siebel 6 を使用している場合は、ステップ 7 に進みます。
 - Siebel 7 を使用している場合は、「ページ (Page)」タブのロケールに移動し、CW Events の新しいレコードを作成します。言語コードとして ENU を追加し、テキストがない場合はテキストとして IBMEvents を追加します。
7. 以下のように画面メニュー項目を作成します。
 - a. 「アプリケーション (Application)」 > 「Siebel Sales Enterprise」 > 「画面 (Screen)」メニュー項目の順にアクセスします。

- b. 右マウス・ボタン・クリックして「新規レコード (New Record)」を選択します。
 - c. 画面名に **Events** を入力し、テキスト名に **IBM Events** を入力します。
 - d. シーケンスの場合は、シーケンス番号以外の部分より大きい番号を入力します。この選択で、タブがプルダウン・メニューのどこに表示されるかが決まります。
 - e. 非アクティブなフィールドのチェックを外したままにします。
 - Siebel 6 を使用している場合は、ステップ 8 に進みます。
 - Siebel 7 を使用している場合は、画面メニュー項目のロケールに移動し、**CWEvents** の新しいレコードを作成します。言語として **ENU** を追加し、テキストがない場合はテキストとして **IBMEvents** を追加します。
8. サイトで使用するビジネス・オブジェクトに対応するビジネス・コンポーネントとして **Siebel VB** スクリプトを追加または変更します。**Siebel VB** スクリプトは、ビジネス・オブジェクトのイベント通知を起動します。
- 優先順位別にイベントをソートしたい場合は、コンパイルする前にビジネス・オブジェクト **VB** スクリプト内で優先順位の値を編集します。
 - 複数のコネクタをインストールする場合は、**VB** スクリプト内に **Connector Id** を設定し、アクティブにします。

Siebel 6

Additional Object Key フィールドを使用したい場合は、**VB** スクリプト内にそのフィールドを設定します。

- 9. 新しい表の物理スキーマをローカル・データベースに適用します。これを行うには、2 つの新しい表 **CX_CW_ARCHIVE_Q** と **CX_CW_EVENT_Q** を照会し、現行の照会选择して物理スキーマを作成します。表スペースと索引スペースをブランクのままにします。
- 10. 活動化ボタンを使用して新しいスキーマをアクティブにします。
- 11. ローカル・データベース上で更新済みプロジェクトとロックされたプロジェクトをコンパイルして、新しい **Siebel** リポジトリ (**.srf**) ファイルを作成します。
- 12. ローカル・データベースで **Siebel Sales Enterprise** を開きます。以下の操作を行うには、管理特権が必要です。
 - a. **Event List View** という新しいビューを作成します。ヒント: ツールからビュー名をコピーし、**View Name** フィールドに貼り付けます。
 - b. **Event List View** に対し、**CW Responsibility** という新しい責任を作成します。
 - c. イベントを検討する責任を持つ社員またはチームを、新たに作成された **CW Responsibility** に追加します。
 - d. **CWCONN** ユーザーを作成し、**CW Responsibility** および **Administrative Responsibility** に追加します。
- 13. ローカル環境でアプリケーションをテストします。**Event List View** に対する可視性があるかどうか、また、サポートされるオブジェクトを作成した後にイ

イベントがビューに作成されるかどうかを確認します。例えば、Siebel に新しいアカウントを作成し、新しいアカウント・イベントが Event List View に表示されるかどうかを確認します。

14. 以下の更新済みプロジェクトとロックされたプロジェクトを開発サーバーにチェックインします。
 - 新しい表
 - CW Audit
 - Dock
 - Siebel Sales Enterprise
 - 使用するビジネス・オブジェクトのプロジェクト

注: 照会によってのみロックされたプロジェクトをチェックインしてください。

15. 開発データベースに物理スキーマを適用します。これを行うには、2 つの新しい表 CX_CW_ARCHIVE_Q と CX_CW_EVENT_Q に照会を行い、現行の照会を選択して物理スキーマを作成します。表スペースと索引スペースをブランクのままにします。
16. 開発データベースで照会済み表をアクティブにします。
17. 適宜、テストおよび実稼働環境に移動します。
18. 新たにコンパイルした Siebel.srf ファイルをサーバーに移動します。

注: 「サイト・マップ (Sitemap)」 > 「サーバー管理 (Server Administration)」 > 「コンポーネント・グループ (Component Group)」と移動して「使用可能」を選択して、Enterprise Application Integration を使用可能にします。

Siebel JAVABean の設定方法は、以下のとおりです。

1. 「サイト・マップ (Site Map)」 -> 「サーバー管理 (Server Administration)」 -> 「コンポーネント (セールス・オブジェクト・マネージャー) (Components (Sales Object Manager))」と選択します。
2. 下部アプレットの「コンポーネント・パラメーター (Component Parameter)」に移動して、タイムアウト値を入力します。

注: 「要求タイムアウト (Request Timeout)」の現行値は 600 に設定されています。これは、コネクタが 10 分で停止するということを示します。Siebel に基づいて、この値を必要に応じて任意の値に変更できます。

第 3 章 コネクタの構成

この章では、Connector Configurator Express を使用してアダプターをインストールおよび構成する方法について説明します。

- 『Connector Configurator Express の概要』
- 20 ページの『Connector Configurator Express の始動』
- 20 ページの『System Manager からの Configurator の実行』
- 21 ページの『コネクタ固有のプロパティ・テンプレートの作成』
- 24 ページの『新しい構成ファイルを作成』
- 26 ページの『既存ファイルの使用』
- 27 ページの『構成ファイルの完成』
- 28 ページの『構成ファイル・プロパティの設定』
- 36 ページの『構成ファイルの保管』
- 36 ページの『構成の完了』
- 37 ページの『グローバル化環境における Connector Configurator Express の使用』

Connector Configurator Express の概要

Connector Configurator Express では、統合ブローカーである InterChange Server Express で使用するアダプターのコネクタ・コンポーネントを構成できます。

Connector Configurator Express を使用して次の作業を行います。

- コネクタを構成するための**コネクタ固有のプロパティ・テンプレート**を作成する。
- **コネクタ構成ファイル**を作成します。インストールするコネクタごとに構成ファイルを 1 つ作成する必要があります。
- 構成ファイル内のプロパティを設定する。
場合によっては、コネクタ・テンプレートでプロパティに対して設定されているデフォルト値を変更する必要があります。また、サポートされるビジネス・オブジェクト定義と、InterChange Server Express の場合はコラボレーションとともに使用するマップを指定し、必要に応じてメッセージング、ロギング、トレース、およびデータ・ハンドラー・パラメーターを指定する必要があります。

Connector Configurator Express の実行モードと使用する構成ファイルのタイプは、実行する統合ブローカーによって異なります。例えば、使用している統合ブローカーが InterChange Server Express の場合、Connector Configurator Express を System Manager から実行するのではなく、直接実行します (20 ページの『スタンドアロン・モードでの Configurator の実行』を参照)。

コネクタ構成プロパティには、標準の構成プロパティ (すべてのコネクタがもつプロパティ) と、コネクタ固有のプロパティ (特定のアプリケーションまたはテクノロジーのためにコネクタに必要なプロパティ) とが含まれます。

標準プロパティは、すべてのコネクタで使用されるので、最初から定義する必要はありません。構成ファイルを作成すると、Connector Configurator Express によって標準プロパティがそのファイルに挿入されます。ただし、Connector Configurator Express で各標準プロパティの値を設定する必要があります。

標準プロパティの範囲は、ブローカーと構成によって異なる可能性があります。特定のプロパティに特定の値が設定されている場合にのみ使用できるプロパティがあります。Connector Configurator Express の「標準のプロパティ」ウィンドウには、現在ご使用の特定の構成で設定可能なプロパティが表示されます。

ただしコネクタ固有プロパティの場合は、最初にプロパティを定義し、その値を設定する必要があります。このため、特定のアダプタのコネクタ固有プロパティのテンプレートを作成します。システム内で既にテンプレートが作成されている場合には、作成されているテンプレートを使用します。システム内でまだテンプレートが作成されていない場合には、21 ページの『新規テンプレートの作成』のステップに従い、テンプレートを新規に作成します。

Connector Configurator Express の始動

Connector Configurator Express は、以下の 2 種類のモードで始動し、実行することができます。

- スタンドアロン・モードで個別に実行
- System Manager から

スタンドアロン・モードでの Configurator の実行

Connector Configurator Express をブローカーと連携させずに別個に実行して、コネクタ構成ファイルを編集することができます。

これを行うには、以下のステップを実行します。

- 「スタート」>「プログラム」から、「IBM WebSphere Business Integration Express」>「Toolset Express」>「開発」>「Connector Configurator Express」をクリックします。
- 「ファイル」>「新規」>「コネクタ構成」を選択します。
- 「システム接続: Integration Broker」の隣のプルダウン・メニューをクリックし、「InterChange Server Express」を選択します。

Connector Configurator Express を個別に実行して構成ファイルを生成してから、System Manager に接続してこの構成ファイルを System Manager プロジェクトに保存する方法が便利です (27 ページの『構成ファイルの完成』を参照)。

System Manager からの Configurator の実行

System Manager から Connector Configurator Express を実行できます。

Connector Configurator Express を実行するには、以下のステップを実行します。

1. System Manager を開きます。
2. 「System Manager」ウィンドウで、「統合コンポーネント・ライブラリー」アイコンを展開し、「コネクタ」を強調表示します。

3. System Manager メニュー・バーから、「ツール」>「**Connector Configurator Express**」をクリックします。「Connector Configurator Express」ウィンドウが開き、「**新規コネクタ**」ダイアログ・ボックスが表示されます。
4. 「**システム接続: Integration Broker**」の隣のプルダウン・メニューをクリックし、「InterChange Server Express」を選択します。

既存の構成ファイルを編集するには、以下のステップを実行します。

- 「System Manager」ウィンドウの「コネクタ」フォルダーでいずれかの構成ファイルを選択し、右クリックします。Connector Configurator Express が開き、この構成ファイルの統合ブローカー・タイプおよびファイル名が上部に表示されます。
- Connector Configurator Express で「ファイル」>「開く」を選択します。プロジェクトまたはプロジェクトが保管されているディレクトリーからコネクタ構成ファイルを選択します。
- 「標準のプロパティ」タブをクリックし、この構成ファイルに含まれているプロパティを確認します。

コネクタ固有のプロパティ・テンプレートの作成

コネクタの構成ファイルを作成するには、コネクタ固有プロパティのテンプレートとシステム提供の標準プロパティが必要です。

コネクタ固有プロパティのテンプレートを新規に作成するか、または既存のコネクタ定義をテンプレートとして使用します。

- テンプレートの新規作成については、『**新規テンプレートの作成**』を参照してください。
- 既存のファイルを使用する場合には、既存のテンプレートを変更し、新しい名前でのこのテンプレートを保管します。既存のテンプレートは
¥WebSphereAdapters¥bin¥Data¥App ディレクトリーにあります。

新規テンプレートの作成

このセクションでは、テンプレートでプロパティを作成し、プロパティの一般特性および値を定義し、プロパティ間の依存関係を指定する方法について説明します。次にそのテンプレートを保管し、新規コネクタ構成ファイルを作成するためのベースとして使用します。

Connector Configurator Express でテンプレートを作成するには、以下のステップを実行します。

1. 「ファイル」>「新規」>「コネクタ固有プロパティ・テンプレート」をクリックします。
2. 「コネクタ固有プロパティ・テンプレート」ダイアログ・ボックスが表示されます。
 - 「**新規テンプレート名を入力してください**」の下の「名前」フィールドに、新規テンプレートの名前を入力します。テンプレートから新規構成ファイルを作成するためのダイアログ・ボックスを開くと、この名前が再度表示されます。

- テンプレートに含まれているコネクタ固有のプロパティ定義を調べるには、「**テンプレート名**」表示でそのテンプレートの名前を選択します。そのテンプレートに含まれているプロパティ定義のリストが「**テンプレートのプレビュー**」表示に表示されます。
3. テンプレートを作成するときには、ご使用のコネクタに必要なプロパティ定義に類似したプロパティ定義が含まれている既存のテンプレートを使用できます。ご使用のコネクタで使用するコネクタ固有のプロパティが表示されるテンプレートが見つからない場合は、自分で作成する必要があります。
 - 既存のテンプレートを変更する場合には、「**変更する既存のテンプレートを選択してください: 検索テンプレート**」の下の「**テンプレート名**」テーブルのリストから、テンプレート名を選択します。
 - このテーブルには、現在使用可能なすべてのテンプレートの名前が表示されます。テンプレートを検索することもできます。

一般特性の指定

「次へ」をクリックしてテンプレートを選択すると、「**プロパティ: コネクタ固有プロパティ・テンプレート**」ダイアログ・ボックスが表示されます。このダイアログ・ボックスには、定義済みプロパティの「**一般**」特性のタブと「**値**」の制限のタブがあります。「**一般**」表示には以下のフィールドがあります。

- **一般:**
 - プロパティ・タイプ
 - プロパティ・サブタイプ
 - 更新されたメソッド
 - 説明
- **フラグ**
 - 標準フラグ
- **カスタム・フラグ**
 - フラグ

「**プロパティ・タイプ**」がストリングの場合、「**プロパティ・サブタイプ**」を選択できます。これは、構成ファイルの保管時に構文検査を提供するオプションの値です。デフォルトはブランク・スペースで、プロパティのサブタイプが指定されていないことを意味します。

プロパティの一般特性の選択を終えたら、「**値**」タブをクリックします。

値の指定

「**値**」タブを使用すると、プロパティの最大長、最大複数値、デフォルト値、または値の範囲を設定できます。編集可能な値も許可されます。これを行うには、以下のステップを実行します。

1. 「**値**」タブをクリックします。「**一般**」のパネルに代わって「**値**」の表示パネルが表示されます。
2. 「**プロパティを編集**」表示でプロパティの名前を選択します。
3. 「**最大長**」および「**最大複数値**」のフィールドに値を入力します。

新規プロパティ値を作成するには、以下のステップを実行します。

1. 「**値**」列見出しの左側の正方形を右クリックします。

2. ポップアップ・メニューから「追加」を選択して、「プロパティ値」ダイアログ・ボックスを表示します。このダイアログ・ボックスではプロパティのタイプに応じて、値だけを入力できる場合と、値と範囲の両方を入力できる場合があります。
3. 新規プロパティ値を入力し、「OK」をクリックします。右側の「値」パネルに値が表示されます。

「値」パネルには、3つの列からなるテーブルが表示されます。

「値」の列には、「プロパティ値」ダイアログ・ボックスで入力した値と、以前に作成した値が表示されます。

「デフォルト値」の列では、値のいずれかをデフォルトとして指定することができます。

「値の範囲」の列には、「プロパティ値」ダイアログ・ボックスで入力した範囲が表示されます。

値が作成されて、グリッドに表示されると、そのテーブルの表示内から編集できるようになります。

テーブルにある既存の値の変更を行うには、その行の行番号をクリックして行全体を選択します。次に「値」フィールドを右マウス・ボタンでクリックし、「値の編集 (Edit Value)」をクリックします。

依存関係の設定

「一般」タブと「値」タブで変更を行ったら、「次へ」をクリックします。「依存関係: コネクター固有プロパティ・テンプレート」ダイアログ・ボックスが表示されます。

依存プロパティは、別のプロパティの値が特定の条件に合致する場合にのみ、テンプレートに組み込まれて、構成ファイルで使用されるプロパティです。例えば、テンプレートに PollQuantity が表示されるのは、トランスポート機構が JMS であり、DuplicateEventElimination が True に設定されている場合のみです。プロパティを依存プロパティとして指定し、依存する条件を設定するには、以下のステップを実行します。

1. 「使用可能なプロパティ」表示で、依存プロパティとして指定するプロパティを選択します。
2. 「プロパティを選択」フィールドで、ドロップダウン・メニューを使用して、条件値を持たせるプロパティを選択します。
3. 「条件演算子」フィールドで以下のいずれかを選択します。

== (等しい)

!= (等しくない)

> (より大)

< (より小)

>= (より大か等しい)

<= (より小か等しい)

4. 「条件値」フィールドで、依存プロパティをテンプレートに組み込むために必要な値を入力します。
5. 「使用可能なプロパティ」表示で依存プロパティを強調表示させて矢印をクリックし、「依存プロパティ」表示に移動させます。
6. 「完了」をクリックします。入力した情報が、Connector Configurator Express によって、Connector Configurator Express がインストールされている %bin ディレクトリーの %data%app の下に XML 文書として保管されます。

パス名の設定

パス名の設定の一般的な規則のいくつかを以下に示します。

- Windows でのファイル名の最大長は 255 文字です。
- Windows では、絶対パス名は [Drive:][Directory]%filename の形式に従う必要があります。例えば、C:%WebSphereAdapters%bin%Data%Std%StdConnProps.xml のようにします。
- キュー名では、先頭または途中にスペースを使用することはできません。

新しい構成ファイルを作成

構成ファイルを新規に作成するには、構成ファイルの名前を指定し、統合ブローカーを選択する必要があります。

ファイルの拡張検証のために、オペレーティング・システムも選択します。ツールバーには「ターゲット・システム」というドロップ・リストがあり、ここで、プロパティの拡張検証用のターゲット・オペレーティング・システムを選択できます。選択可能なオプションは「Windows」で、これが始動時のデフォルトです。

Connector Configurator Express を始動するには、以下のステップを実行します。

- 「System Manager」ウィンドウで、「ツール」メニューから「**Connector Configurator Express**」を選択します。Connector Configurator Express が開きます。
- スタンドアロン・モードで、Connector Configurator Express を起動します。

構成ファイルの拡張検証用のオペレーティング・システムを設定するには、以下のステップを実行します。

- メニュー・バーの「ターゲット・システム:」ドロップ・リストをプルダウンします。
- 使用中のオペレーティング・システムを選択します。

次に、「ファイル」>「新規」>「コネクタ構成」を選択します。「新規コネクタ」ウィンドウで、新規コネクタの名前を入力します。

また、統合ブローカーも選択する必要があります。選択したブローカーによって、構成ファイルに記述されるプロパティが決まります。ブローカーを選択するには、以下のステップを実行します。

- 「**Integration Broker**」フィールドで、InterChange Server Express を選択します。

- この章で後述する説明に従って「新規コネクター」ウィンドウの残りのフィールドに入力します。

コネクター固有のテンプレートからの構成ファイルの作成

コネクター固有のテンプレートを作成すると、テンプレートを使用して構成ファイルを作成できます。

1. メニュー・バーの「ターゲット・システム:」ドロップ・リストを使用して、構成ファイルの拡張検証用のオペレーティング・システムを設定します (前述の『新規構成ファイルの作成』を参照してください)。
2. 「ファイル」>「新規」>「コネクター構成」をクリックします。
3. 以下のフィールドを含む「新規コネクター」ダイアログ・ボックス表示されず。

• 名前

コネクターの名前を入力します。名前では大文字と小文字が区別されます。入力する名前は、システムにインストールされているコネクターのファイル名に対応した一意の名前でなければなりません。

重要: Connector Configurator Express では、入力された名前のスペルはチェックされません。名前が正しいことを確認してください。

• システム接続

「InterChange Server Express」をクリックします。

- 「コネクター固有プロパティ・テンプレート」を選択します。

ご使用のコネクター用に設計したテンプレートの名前を入力します。「テンプレート名」表示に、使用可能なテンプレートが表示されます。「テンプレート名」表示で名前を選択すると、「プロパティ・テンプレートのプレビュー」表示に、そのテンプレートで定義されているコネクター固有のプロパティが表示されます。

使用するテンプレートを選択し、「OK」をクリックします。

4. 構成しているコネクターの構成画面が表示されます。タイトル・バーに統合ブローカーとコネクターの名前が表示されます。ここですべてのフィールドに値を入力して定義を完了するか、ファイルを保管して後でフィールドに値を入力するかを選択できます。
5. ファイルを保管するには、「ファイル」>「保管」>「ファイルに」をクリックするか、「ファイル」>「保管」>「プロジェクトに」をクリックします。プロジェクトに保管するには、System Manager が実行中でなければなりません。ファイルとして保管する場合は、「ファイル・コネクターを保管」ダイアログ・ボックスが表示されます。*.cfg をファイル・タイプとして選択し、「ファイル名」フィールド内に名前が正しいスペル (大文字と小文字の区別を含む) で表示されていることを確認してから、ファイルを保管するディレクトリーにナビゲートし、「保管」をクリックします。Connector Configurator Express のメッセージ・パネルの状況表示に、構成ファイルが正常に作成されたことが示されます。

重要: ここで設定するディレクトリー・パスおよび名前は、コネクターの始動ファイルで指定するコネクター構成ファイルのパスおよび名前に一致している必要があります。

6. この章で後述する手順に従って、「Connector Configurator Express」ウィンドウの各タブにあるフィールドに値を入力し、コネクター定義を完了します。

既存ファイルの使用

使用可能な既存ファイルは、以下の 1 つまたは複数の形式になります。

- コネクター定義ファイル。
コネクター定義ファイルは、特定のコネクターのプロパティと、適用可能なデフォルト値がリストされたテキスト・ファイルです。コネクターの配布パッケージの `¥repository` ディレクトリー内には、このようなファイルが格納されていることがあります (通常、このファイルの拡張子は `.txt` です。例えば、XML コネクターの場合は `CN_XML.txt` です)。
- InterChange Server Express リポジトリー・ファイル。
以前にコネクターの InterChange Server Express インプリメンテーションの際に使用された定義が、そのコネクターの構成に使用されたりポジトリー・ファイルに残されていることがあります。そのようなファイルの拡張子は、通常 `.in` または `.out` です。
- コネクターの以前の構成ファイル。
これらのファイルの拡張子は、通常 `*.cfg` です。

これらのいずれのファイル・ソースにも、コネクターのコネクター固有プロパティのほとんど、あるいはすべてが含まれますが、この章内の後で説明するように、コネクター構成ファイルは、ファイルを開いて、プロパティを設定しない限り完成しません。

既存ファイルを使用してコネクターを構成するには、Connector Configurator Express でそのファイルを開き、構成を修正してから、再度保管する必要があります。

以下のステップを実行して、ディレクトリーから `*.txt`、`*.cfg`、または `*.in` ファイルを開きます。

1. Connector Configurator Express で、「ファイル」>「開く」>「ファイルから」をクリックします。
2. 「ファイル・コネクターを開く」ダイアログ・ボックス内で、以下のいずれかのファイル・タイプを選択して、使用可能なファイルを調べます。
 - 構成 (`*.cfg`)
 - InterChange Server Express リポジトリー (`*.in`、`*.out`)(InterChange Server Express Repository (`*.in`、`*.out`))

InterChange Server Express 環境でのコネクターの構成にリポジトリー・ファイルが使用された場合には、このオプションを選択します。リポジトリー・ファイルに複数のコネクター定義が含まれている場合は、ファイルを開くとすべての定義が表示されます。

- すべてのファイル (`*.*`)

コネクタのアダプター・パッケージに *.txt ファイルが付属していた場合、または別の拡張子で定義ファイルが使用可能である場合は、このオプションを選択します。

3. ディレクトリー表示内で、適切なコネクタ定義ファイルへ移動し、ファイルを選択し、「開く」をクリックします。

System Manager プロジェクトからコネクタ構成を開くには、以下のステップを実行します。

1. System Manager を始動します。System Manager が開始されている場合にのみ、構成を System Manager から開いたり、System Manager に保管したりできます。
2. Connector Configurator Express を始動します。
3. 「ファイル」>「開く」>「プロジェクトから」をクリックします。

構成ファイルの完成

構成ファイルを開くか、プロジェクトからコネクタを開くと、「Connector Configurator Express」ウィンドウに構成画面が表示されます。この画面には、現在の属性と値が表示されます。

構成画面のタイトルには、ファイル内で指定された統合ブローカーとコネクタの名前が表示されます。InterChange Server Express ブローカーが設定されていることを確認してください。正しいブローカーが設定されていない場合、コネクタを構成する前にブローカー値を変更してください。これを行うには、以下のステップを実行します。

1. 「標準のプロパティー」タブで、BrokerType プロパティーの値フィールドを選択します。ドロップダウン・メニューで、値 Interchange Server Express を選択します。
2. 選択したブローカーに関連付けられているコネクタ・プロパティーが「標準のプロパティー」タブに表示されます。表に、「プロパティー名」、「値」、「タイプ」、「サブタイプ」（「タイプ」がストリングである場合）、「説明」、および「更新メソッド」が表示されます。
3. ここでファイルを保管するか、または 31 ページの『サポートされるビジネス・オブジェクト定義の指定』の説明に従い残りの構成フィールドに値を入力することができます。
4. 構成が完了したら、「ファイル」>「保管」>「プロジェクトに」を選択するか、または「ファイル」>「保管」>「ファイルに」を選択します。

ファイルに保管する場合は、*.cfg を拡張子として選択し、ファイルの正しい格納場所を選択して、「保管」をクリックします。

複数のコネクタ構成を開いている場合、構成をすべてファイルに保管するには「すべてファイルに保管」を選択し、コネクタ構成をすべて System Manager プロジェクトに保管するには「すべてプロジェクトに保管」をクリックします。

構成ファイルを作成する前に、プロパティーの拡張検証用のターゲット・オペレーティング・システムを選択することができる「ターゲット・システム」ドロップ・リストを使用します。

Connector Configurator Express では、ファイルを保管する前に、必須の標準プロパティーすべてに値が設定されているかどうかを確認されます。必須の標準プロパティーに値が設定されていない場合、Connector Configurator Express は、検証が失敗したというメッセージを表示します。構成ファイルを保管するには、そのプロパティーの値を指定する必要があります。

「ターゲット・システム」ドロップ・リストから「Windows」を選択することによって拡張検証機能を使用する場合、システムはタイプだけでなくプロパティー・サブタイプを検証し、検証に失敗した場合は警告メッセージを表示します。

構成ファイル・プロパティーの設定

新規のコネクター構成ファイルを作成して名前を付けると、または既存のコネクター構成ファイルを開くと、Connector Configurator Express に構成画面が表示されます。構成画面には、必要な構成値のカテゴリに対応する複数のタブがあります。

Connector Configurator Express では、すべてのブローカーで実行されているコネクターで、以下のカテゴリのプロパティーに値が設定されている必要があります。

- 標準プロパティー
- コネクター固有のプロパティー
- サポートされるビジネス・オブジェクト
- トレース/ログ・ファイルの値
- データ・ハンドラー (保証付きイベント・デリバリーで JMS メッセージングを使用するコネクターの場合に該当する)

注: JMS メッセージングを使用するコネクターの場合は、データをビジネス・オブジェクトに変換するデータ・ハンドラーの構成に関して追加のカテゴリが表示される場合があります。

InterChange Server Express で実行されているコネクターの場合、以下のプロパティーの値も設定されている必要があります。

- 関連付けられたマップ
- セキュリティー

重要: Connector Configurator Express では、英語文字セットまたは英語以外の文字セットのいずれのプロパティー値も設定可能です。ただし、標準のプロパティーおよびコネクター固有プロパティー、およびサポートされるビジネス・オブジェクトの名前では、英語文字セットのみを使用する必要があります。

標準プロパティーとコネクター固有プロパティーの違いは、以下のとおりです。

- コネクターの標準プロパティーは、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ブローカー・コンポーネントの両方によって共有されます。すべてのコネクターが同じ標準プロパティーのセットを使用します。これらのプロパティーについては、付録『コネクターの標準構成プロパティー』で説明されています。変更できるのはこれらの値の一部のみです。
- アプリケーション固有のプロパティーは、コネクターのアプリケーション固有コンポーネント (アプリケーションと直接対話するコンポーネント) のみに適用されます。各コネクターには、そのコネクターのアプリケーションだけで使用される

アプリケーション固有のプロパティがあります。これらのプロパティには、デフォルト値が用意されているものもあれば、そうでないものもあります。また、一部のデフォルト値は変更することができます。各アダプター・ガイドのインストールおよび構成の章に、アプリケーション固有のプロパティおよび推奨値が記述されています。

「標準プロパティ」と「コネクタ固有プロパティ」のフィールドは、どのフィールドが構成可能であるかを示すために色分けされています。

- 背景がグレーのフィールドは、標準のプロパティを表します。値を変更することはできますが、名前の変更およびプロパティの除去はできません。
- 背景が白のフィールドは、アプリケーション固有のプロパティを表します。これらのプロパティは、アプリケーションまたはコネクタの特定のニーズによって異なります。値の変更も、これらのプロパティの除去も可能です。
- 「値」フィールドは構成できます。
- プロパティごとに「更新メソッド」フィールドが表示されます。これは、変更された値をアクティブにするためにコンポーネントまたはエージェントの再始動が必要かどうかを示します。この設定を構成することはできません。

標準コネクタ・プロパティの設定

標準のプロパティの値を変更するには、以下の手順を実行します。

1. 値を設定するフィールド内でクリックします。
2. 値を入力するか、ドロップダウン・メニューが表示された場合にはメニューから値を選択します。

注: プロパティの「タイプ」が「string」である場合、「サブタイプ」列にサブタイプ値が含まれている場合があります。このサブタイプは、プロパティの拡張検証に使用されます。

3. 標準のプロパティの値をすべて入力後、以下のいずれかを実行することができます。
 - 変更内容を破棄し、元の値を保持したままで Connector Configurator Express を終了するには、「ファイル」>「終了」をクリックし（またはウィンドウを閉じ）、変更内容を保管するかどうかを確認するプロンプトが出されたら「はいえ」をクリックします。
 - Connector Configurator Express 内の他のカテゴリーの値を入力するには、そのカテゴリーのタブを選択します。「標準のプロパティ」（またはその他のカテゴリー）で入力した値は、次のカテゴリーに移動しても保持されます。ウィンドウを閉じると、すべてのカテゴリーで入力した値を一括して保管するかまたは破棄するかを確認するプロンプトが出されます。
 - 修正した値を保管するには、「ファイル」>「終了」をクリックし（またはウィンドウを閉じ）、変更内容を保管するかどうかを確認するプロンプトが出されたら「はい」をクリックします。「ファイル」メニューまたはツールバーから「保管」>「ファイルに」をクリックする方法もあります。

特定の標準プロパティに関する詳細を参照するには、「標準のプロパティ」タブ付きシート内のそのプロパティの「説明」列内の項目を左クリックします。全

般ヘルプをインストール済みの場合は、右側に矢印ボタンが表示されます。ボタンをクリックすると、「ヘルプ」ウィンドウが開き、標準プロパティの詳細が表示されます。

注: ホット・ボタンが表示されない場合、そのプロパティについては全般ヘルプが見つかりません。

インストール済みの場合、全般ヘルプ・ファイルは
<ProductDir>%bin%Data%Std%Help%<RegionalSetting>% にあります。

コネクタ固有の構成プロパティの設定

コネクタ固有の構成プロパティの場合、プロパティ名の追加または変更、値の構成、プロパティの削除、およびプロパティの暗号化が可能です。プロパティのデフォルトの長さは 255 文字です。

1. グリッドの左上端の部分で右マウス・ボタンをクリックします。ポップアップ・メニュー・バーが表示されます。プロパティを追加するときは「追加」をクリックします。子プロパティを追加するには、親の行番号で右マウス・ボタンをクリックし、「子を追加」をクリックします。
2. プロパティまたは子プロパティの値を入力します。

注: プロパティの「タイプ」が「ストリング」である場合、「サブタイプ」ドロップ・リストからサブタイプを選択できます。このサブタイプは、プロパティの拡張検証に使用されます。

3. プロパティを暗号化するには、「暗号化」ボックスを選択します。
4. 特定のプロパティに関する詳細を参照するには、そのプロパティの「説明」列内の項目を左クリックします。全般ヘルプをインストール済みの場合は、ホット・ボタンが表示されます。ホット・ボタンをクリックすると、「ヘルプ」ウィンドウが開き、標準プロパティの詳細が表示されます。

注: ホット・ボタンが表示されない場合、そのプロパティについては全般ヘルプが見つかりません。

5. 29 ページの『標準コネクタ・プロパティの設定』の説明に従い、変更内容を保管するかまたは破棄するかを選択します。

全般ヘルプがインストール済みで、AdapterHelpName プロパティがブランクである場合、Connector Configurator Express は、
<ProductDir>%bin%Data%App%Help%<RegionalSetting>% にあるアダプター固有の全般ヘルプ・ファイルを指します。それ以外の場合、Connector Configurator Express は、
<ProductDir>%bin%Data%App%Help%<AdapterHelpName>%<RegionalSetting>%にあるアダプター固有の全般ヘルプ・ファイルを指します。付録『コネクタの標準構成プロパティ』で説明されている AdapterHelpName プロパティを参照してください。

各プロパティごとに表示される「更新メソッド」は、変更された値をアクティブにするためにコンポーネントまたはエージェントの再始動が必要かどうかを示します。

重要: 事前設定のアプリケーション固有のコネクター・プロパティ名を変更すると、コネクターに障害が発生する可能性があります。コネクターをアプリケーションに接続したり正常に実行したりするために、特定のプロパティ名が必要である場合があります。

コネクター・プロパティの暗号化

「コネクター固有プロパティ」ウィンドウの「暗号化」チェック・ボックスにチェックマークを付けると、アプリケーション固有のプロパティを暗号化することができます。値の暗号化を解除するには、「暗号化」チェック・ボックスをクリックしてチェックマークを外し、「検証」ダイアログ・ボックスに正しい値を入力し、「OK」をクリックします。入力された値が正しい場合は、暗号化解除された値が表示されます。

各プロパティとそのデフォルト値のリストおよび説明は、各コネクターのアダプター・ユーザズ・ガイドにあります。

プロパティに複数の値がある場合には、プロパティの最初の値に「暗号化」チェック・ボックスが表示されます。「暗号化」を選択すると、そのプロパティのすべての値が暗号化されます。プロパティの複数の値を暗号化解除するには、そのプロパティの最初の値の「暗号化」チェック・ボックスをクリックしてチェックマークを外してから、「検証」ダイアログ・ボックスで新規の値を入力します。入力値が一致すれば、すべての複数值が暗号化解除されます。

更新メソッド

付録『コネクターの標準構成プロパティ』内の更新メソッドの説明を参照してください。

サポートされるビジネス・オブジェクト定義の指定

コネクターで使用するビジネス・オブジェクトを指定するには、Connector Configurator Express の「サポートされているビジネス・オブジェクト」タブを使用します。汎用ビジネス・オブジェクトと、アプリケーション固有のビジネス・オブジェクトの両方を指定する必要があり、またそれらのビジネス・オブジェクト間のマップの関連を指定することが必要です。

ご使用のブローカーが InterChange Server Express の場合

ビジネス・オブジェクト定義がコネクターでサポートされることを指定する場合や、既存のビジネス・オブジェクト定義のサポート設定を変更する場合は、「サポートされているビジネス・オブジェクト」タブをクリックし、以下のフィールドを使用してください。

ビジネス・オブジェクト名: ビジネス・オブジェクト定義がコネクターによってサポートされることを指定するには、System Manager を実行し、以下の手順を実行します。

1. 「ビジネス・オブジェクト名」リストで空のフィールドをクリックします。
System Manager プロジェクトに存在するすべてのビジネス・オブジェクト定義を示すドロップ・リストが表示されます。
2. 追加するビジネス・オブジェクトをクリックします。

3. ビジネス・オブジェクトの「エージェント・サポート」(以下で説明)を設定します。
4. 「Connector Configurator Express」ウィンドウの「ファイル」メニューで、「プロジェクトに保管」をクリックします。追加したビジネス・オブジェクト定義に指定されたサポートを含む、変更されたコネクタ定義が、System Manager の ICL (Integration Component Library) プロジェクトに保管されます。

サポートされるリストからビジネス・オブジェクトを削除する場合は、以下の手順を実行します。

1. ビジネス・オブジェクト・フィールドを選択するため、そのビジネス・オブジェクトの左側の番号をクリックします。
2. 「Connector Configurator Express」ウィンドウの「編集」メニューから、「行を削除」をクリックします。リスト表示からビジネス・オブジェクトが除去されません。
3. 「ファイル」メニューから、「プロジェクトの保管」をクリックします。

サポートされるリストからビジネス・オブジェクトを削除すると、コネクタ定義が変更され、削除されたビジネス・オブジェクトはコネクタのこのインプリメンテーションで使用不可になります。コネクタのコードに影響したり、そのビジネス・オブジェクト定義そのものが System Manager から削除されることはありません。

エージェント・サポート: ビジネス・オブジェクトがエージェント・サポートを備えている場合、システムは、コネクタ・エージェントを介してアプリケーションにデータを配布する際にそのビジネス・オブジェクトの使用を試みます。

一般に、コネクタのアプリケーション固有ビジネス・オブジェクトは、そのコネクタのエージェントによってサポートされますが、汎用ビジネス・オブジェクトはサポートされません。

ビジネス・オブジェクトがコネクタ・エージェントによってサポートされるよう指定するには、「エージェント・サポート」ボックスにチェックマークを付けます。「Connector Configurator Express」ウィンドウでは、「エージェント・サポート」を選択しても問題ないかどうかの検証は行われません。

最大トランザクション・レベル: コネクタの最大トランザクション・レベルは、そのコネクタがサポートする最大のトランザクション・レベルです。

ほとんどのコネクタの場合、選択可能な項目は「最大限の努力」のみです。

トランザクション・レベルの変更を有効にするには、サーバーを再始動する必要があります。

関連付けられたマップ - InterChange Server Express

各コネクタは、ビジネス・オブジェクト定義とそれらに関連付けられたマップのうち現在 InterChange Server Express でアクティブであるものを示すリストをサポートします。このリストは、「関連付けられたマップ」タブを選択すると表示されます。

ビジネス・オブジェクトのリストには、エージェントでサポートされるアプリケーション固有のビジネス・オブジェクトと、コントローラーがサブスクリプション・コラボレーションに送信する、対応する汎用オブジェクトが含まれます。マップの関連性によって、アプリケーション固有のビジネス・オブジェクトを汎用ビジネス・オブジェクトに変換したり、汎用ビジネス・オブジェクトをアプリケーション固有のビジネス・オブジェクトに変換したりするときに、どのマップを使用するかが決定されます。

特定のソースおよび宛先ビジネス・オブジェクトについて一意的に定義されたマップを使用する場合、表示を開くと、マップは常にそれらの該当するビジネス・オブジェクトに関連付けられます。ユーザーがそれらを変更する必要はありません (変更できません)。

サポートされるビジネス・オブジェクトで使用可能なマップが複数ある場合は、そのビジネス・オブジェクトを、使用する必要のあるマップに明示的にバインドすることが必要になります。

「関連付けられたマップ」タブには以下のフィールドが表示されます。

- **ビジネス・オブジェクト名**

これらは、「サポートされているビジネス・オブジェクト」タブで指定した、このコネクターでサポートされるビジネス・オブジェクトです。「サポートされているビジネス・オブジェクト」タブでビジネス・オブジェクトを追加指定した場合、その内容は、「Connector Configurator Express」ウィンドウの「ファイル」メニューから「プロジェクトに保管」を選択して変更を保管した後に、このリストに反映されます。

- **関連付けられたマップ**

この表示には、コネクターの、サポートされるビジネス・オブジェクトでの使用のためにシステムにインストールされたすべてのマップが示されます。各マップのソース・ビジネス・オブジェクトは、「ビジネス・オブジェクト名」表示でマップ名の左側に表示されます。

- **明示的バインディング**

場合によっては、関連付けられたマップを明示的にバインドすることが必要になります。

明示的バインディングが必要なのは、特定のサポートされるビジネス・オブジェクトに複数のマップが存在する場合のみです。InterChange Server Express は、ブート時、各コネクターのサポートされるビジネス・オブジェクトのそれぞれにマップを自動的にバインドしようとしています。複数のマップでその入力データとして同一のビジネス・オブジェクトが使用されている場合、サーバーは、他のマップのスーパーセットである 1 つのマップを見つけて、バインドしようとしています。

他のマップのスーパーセットであるマップがないと、サーバーは、ビジネス・オブジェクトを単一のマップにバインドすることができないため、バインディングを明示的に設定することが必要になります。

以下の手順を実行して、マップを明示的にバインドします。

1. 「明示的 (Explicit)」列で、バインドするマップのチェック・ボックスにチェックマークを付けます。
2. ビジネス・オブジェクトに関連付けるマップを選択します。
3. 「Connector Configurator Express」ウィンドウの「ファイル」メニューで、「プロジェクトに保管」をクリックします。
4. プロジェクトを InterChange Server Express に配置します。
5. 変更を有効にするため、サーバーをリブートします。

セキュリティ - InterChange Server Express

Connector Configurator Express 内の「セキュリティ」タブを使用して、メッセージにさまざまなプライバシー・レベルを設定することができます。DeliveryTransport プロパティが JMS に設定されている場合のみ、この機能を使用できます。

デフォルトでは、「プライバシー」はオフになっています。使用可能にするには、「プライバシー」ボックスにチェック・マークを付けます。

「鍵ストア・ターゲット・システムの絶対パス名」は、以下の値です。

- Windows の場合:
`<ProductDir>%connectors%security%<connectorname>.jks`

このパスおよびファイルは、コネクタを始動するシステム、すなわちターゲット・システム上に存在していなければなりません。

ターゲット・システムが現在実行中のシステムである場合のみ、右側の「参照」ボタンを使用できます。「プライバシー」が使用可能であり、メニュー・バーの「ターゲット・システム」が Windows に設定されている場合を除き、これはグレーアウトされています。

「メッセージのプライバシー・レベル」は、3 つのメッセージ・カテゴリ (全メッセージ、全管理メッセージ、および全ビジネス・オブジェクト・メッセージ) で以下のように設定されます。

- “”: がデフォルトです。メッセージ・カテゴリにプライバシー・レベルが設定されていない場合に使用します。
- none
 デフォルトと同じではありません。メッセージ・カテゴリにプライバシー・レベルなしと故意に設定する場合にこれを使用します。
- integrity
- privacy
- integrity_plus_privacy

「鍵の保守」機能によって、サーバーおよびアダプターの公開鍵を生成、インポート、およびエクスポートすることができます。

- 「鍵の生成」を選択すると、鍵を生成する keytool のデフォルトを含む「鍵の生成」ダイアログ・ボックスが表示されます。
- 「セキュリティ」タブの「鍵ストア・ターゲット・システムの絶対パス名」で入力した値が、鍵ストア値のデフォルトになります。

- 「OK」を選択すると、記入項目が検証され、鍵証明書が生成され、「Connector Configurator Express」ログ・ウィンドウに出力が送られます。

証明書をアダプター鍵ストアにインポートする前に、サーバー鍵ストアからエクスポートする必要があります。「**アダプター公開鍵のエクスポート**」を選択すると、「アダプター公開鍵のエクスポート」ダイアログ・ボックスが表示されます。

- エクスポート証明書のデフォルトは、ファイル拡張子が <filename>.cer であることを除き、鍵ストアと同じ値です。

「**サーバー公開鍵のインポート**」を選択すると、「サーバー公開鍵のインポート」ダイアログ・ボックスが表示されます。

- インポート証明書のデフォルトは、<ProductDir>%bin%ics.cer になります (システムにファイルが存在する場合)。
- インポート証明書関連はサーバー名でなければなりません。サーバーが登録されていれば、ドロップ・リストからそれを選択することができます。

DeliveryTransport の値が IDL の場合のみ、「**アダプター・アクセス制御**」機能が使用可能です。デフォルトでは、アダプターはゲスト ID を使用してログインします。「**ゲスト ID の使用**」ボックスにチェック・マークが付けられていない場合は、「**アダプター ID**」および「**アダプター・パスワード**」フィールドが使用可能です。

トレース/ログ・ファイル値の設定

コネクタ構成ファイルまたはコネクタ定義ファイルを開くと、Connector Configurator Express は、そのファイルに含まれるロギングとトレースに関する値をデフォルト値として使用します。これらの値は、Connector Configurator Express 内で変更できます。

ログとトレースの値を変更するには、以下の手順を実行します。

1. 「**トレース/ログ・ファイル**」タブをクリックします。
2. ログとトレースのどちらでも、以下のいずれかまたは両方へのメッセージの書き込みを選択できます。

- コンソールに (STDOUT):
ログ・メッセージまたはトレース・メッセージを STDOUT ディスプレイに書き込みます。

注: STDOUT オプションは、Windows プラットフォームで実行しているコネクタの「**トレース/ログ・ファイル**」タブでのみ使用できます。

- ファイルに:
ログ・メッセージまたはトレース・メッセージを指定されたファイルに書き込みます。ファイルを指定するには、ディレクトリー・ボタン (省略符号) をクリックし、指定する格納場所へ移動し、ファイル名を指定し、「**保管**」をクリックします。ログ・メッセージまたはトレース・メッセージは、指定した場所の指定したファイルに書き込まれます。

注: ログ・ファイルとトレース・ファイルはどちらも単純なテキスト・ファイルです。任意のファイル拡張子を使用してこれらのファイル名を設定できます。ただし、トレース・ファイルの場合、拡張子として .trc ではなく

.trace を使用することをお勧めします。これは、システム内に存在する可能性がある他のファイルとの混同を避けるためです。ログ・ファイルの場合、通常使用されるファイル拡張子は .log および .txt です。

データ・ハンドラー

データ・ハンドラー・セクションの構成が使用可能となるのは、DeliveryTransport の値に JMS を、また ContainerManagedEvents の値に JMS を指定した場合のみです。すべてのアダプターでデータ・ハンドラーを使用できるわけではありません。

これらのプロパティーに使用する値については、付録 A 『コネクターの標準構成プロパティー』にある ContainerManagedEvents の下の説明を参照してください。

構成ファイルの保管

コネクターの構成が完了したら、コネクター構成ファイルを保管します。Connector Configurator Express では、構成中に選択したブローカー・モードでファイルを保管します。Connector Configurator Express のタイトル・バーには、現在のブローカー・モード (InterChange Server Express) が常に表示されます。

ファイルは XML 文書として保管されます。XML 文書は次の 3 通りの方法で保管できます。

- System Manager から、統合コンポーネント・ライブラリーに *.con 拡張子付きファイルとして保管します。または、
- ユーザーが指定したディレクトリーに保管します。
- スタンドアロン・モードで、ディレクトリー・フォルダーに *.cfg 拡張子付きファイルとして保管します。デフォルトでは、このファイルは %WebSphereAdapters%bin\Data%App に保管されます。

System Manager でのプロジェクトの使用法、および配置の詳細については、InterChange Server Express の「システム・インプリメンテーション・ガイド」を参照してください。

構成の完了

コネクターの構成ファイルを作成し、そのファイルを変更した後で、コネクターの始動時にコネクターが構成ファイルの位置を特定できるかどうかを確認してください。

これを行うには、コネクターが使用する始動ファイルを開き、コネクター構成ファイルに使用されている格納場所とファイル名が、ファイルに対して指定した名前およびファイルを格納したディレクトリーまたはパスと正確に一致しているかどうかを検証します。

グローバル化環境における Connector Configurator Express の使用

Connector Configurator Express はグローバル化されており、構成ファイルと統合ブローカーの間での文字変換を処理できます。Connector Configurator Express では、ネイティブなエンコード方式を使用しています。構成ファイルに書き込む場合は UTF-8 エンコード方式を使用します。

Connector Configurator Express は、以下の場所で英語以外の文字をサポートします。

- すべての値のフィールド
- ログ・ファイルおよびトレース・ファイル・パス（「トレース/ログ・ファイル」タブで指定）

CharacterEncoding および Locale 標準構成プロパティのドロップ・リストに表示されるのは、サポートされる値の一部のみです。ドロップ・リストに、サポートされる他の値を追加するには、製品ディレクトリーの ¥Data¥Std¥stdConnProps.xml ファイルを手動で変更する必要があります。

例えば、Locale プロパティの値のリストにロケール en_GB を追加するには、stdConnProps.xml ファイルを開き、以下に太字で示した行を追加してください。

```
<Property name=""
isRequired="true"
updateMethod="component restart">
  <ValidType>String</ValidType>
  <ValidValues>
    <Value>ja_JP</Value>
    <Value>ko_KR</Value>
    <Value>zh_CN</Value>
    <Value>zh_TW</Value>
    <Value>fr_FR</Value>
    <Value>de_DE</Value>
    <Value>it_IT</Value>
    <Value>es_ES</Value>
    <Value>pt_BR</Value>
    <Value>en_US</Value>
    <Value>en_GB</Value>
  </ValidValues>
  <DefaultValue>en_US</DefaultValue>
</Property>
```

第 4 章 ビジネス・オブジェクトの理解

- 『ビジネス・オブジェクトの構造および関係』
- 42 ページの『ビジネス・オブジェクトのアプリケーション固有の情報』

この章では、コネクタによるビジネス・オブジェクトの処理方法について説明します。Siebel のビジネス・オブジェクトを正しく作成または変更するには、Siebel アーキテクチャー内のオブジェクトの関係について理解する必要があります。

注: この章では、Event ビジネス・コンポーネントと Archive ビジネス・コンポーネント、Event ビジネス・オブジェクトと Archive ビジネス・オブジェクト、およびイベント表とアーカイブ表を参照しています。これらの参照は、以前のバージョンで見られる CW Event と CW Archive への参照、および Siebel 7.5 で見られる IBM Event と IBM Archive への参照と同義です。

ビジネス・オブジェクトの構造および関係

コネクタは、Siebel アプリケーション固有のビジネス・オブジェクトの動詞 Create、Retrieve、Update、Delete、Exists、Retrieve By Content、および DeltaUpdate をサポートします。そのビジネス・オブジェクトのセマンティクスは、ビジネス・オブジェクト定義の中にカプセル化されています。コネクタ内のハードコーディングされた命令に従って Siebel アプリケーション固有のビジネス・オブジェクトを処理するコネクタ・ロジックはありません。例えば、コネクタは特定のビジネス・オブジェクトがある種の多数のエンティティから構成されているとは考えません。コネクタは、すべてのオブジェクトが 1 つ以上のエンティティから構成されていると考えます。

Siebel ビジネス・コンポーネントは、3 つの方法で関連づけることができます。一価のリンクで 1 対 1 の関係でビジネス・コンポーネントをリンクする方法、1 対多の関係を表す Multi-Value Link (MVL) フィールドをビジネス・コンポーネントに持たせる方法、単純なリンクを持たせる方法です。

ピック・リストを使用すれば、ビジネス・コンポーネントを多対 1 の関係で関連づけることができます。ビジネス・コンポーネント・メソッドは、PickList ビジネス・コンポーネントを検索して特定値を探したり、フィールド内にその値を設定したりする操作をサポートします。最後に、論理積テーブルで多対多の関係でビジネス・コンポーネントを関連付けることができます。

Siebel 内の同じビジネス・オブジェクトの下に 2 つの無関係な単一カーディナリティー・ビジネス・コンポーネントがある場合は、個別のビジネス・オブジェクト・ラッパーを作成する必要があります。

多数のビジネス・コンポーネントをカプセル化するビジネス・オブジェクト・コンテキストの Siebel の概念をサポートするために、トップレベル・ビジネス・オブジェクトを適切な Siebel ビジネス・オブジェクトに対応させる必要があります。トップレベル・ビジネス・オブジェクトのアプリケーション固有の情報には、対応する

Siebel ビジネス・オブジェクトの名前が含まれていなければなりません。次に、各トップレベル属性が Siebel ビジネス・コンポーネントに対応しなければなりません。

ビジネス・コンポーネントに対応するビジネス・オブジェクト定義内では、各属性が単純フィールドまたは Multi-Value Group (MVG) フィールドを指定します。単純属性内の属性データは、単純データ型を持っていなければなりません。MVG フィールドに対応する属性は、子 (コンテナ) ビジネス・オブジェクトとして処理する必要があります。

このビジネス・オブジェクトの構造は、コネクタが同じ方法ですべてのビジネス・オブジェクトを処理できるようにするメタデータの一部です。オブジェクトに対してビジネス・オブジェクト定義が指定されていれば、コネクタはさらに Siebel オブジェクトをサポートすることができます。

キー属性の指定

Siebel ビジネス・オブジェクトを開発する場合は、必ず、オブジェクトのトップにキー属性を配置してください。そうすると、コネクタはオブジェクトの残りの部分を処理する前にキー値を持つようになります。キー属性がオブジェクト内の他の場所に配置されていると、処理エラーが発生する恐れがあります。オブジェクトのキー属性は、Siebel 内の RowId です。

注: コネクタは、トップレベルのビジネス・オブジェクト (Siebel BO) の子を除いて、子ビジネス・オブジェクトまたはキー属性のような子ビジネス・オブジェクトの配列を表す属性の指定をサポートしません。

注: コネクタ用のビジネス・オブジェクトを開発するときは、ビジネス・オブジェクトと Siebel ビジネス・コンポーネントの間に 1 対 1 の対応関係があることを確認してください。

属性プロパティ

以下の表は、単純属性と子オブジェクト属性について説明したものです。

表 4. 単純属性

名前	属性の名前
Type	属性のデータ型。現在ではこの型を使用していませんが、前方互換性のために、SiebelODA はそのデータ型を boolean、String、Date、int、double のいずれかに設定します。すべての型はストリングとして処理されます。
MaxLength	String 型に適用され、属性に許される最大長を表します。これはコネクタでは使用されません。データが大きい場合は、そのデータをビジネス・プロセスで処理する必要があります。

表 4. 単純属性 (続き)

名前	属性の名前
IsKey	これが設定されている場合は、属性がキーであることを示します。この属性は、Siebel で特定レコードを更新するために Update とともに使用します。Retrieve とともに使用する場合は、この属性は Siebel からレコードを取得するために検索指定で使用します。Delete 時は、キーはトップレベル・ビジネス・コンポーネントに設定されます。
IsForeignKey	使用しません。
IsRequired	Siebel ビジネス・コンポーネントのフィールド属性で「Required」プロパティがチェックされている場合は、true に設定します。
AppSpecificInfo	アプリケーションとの通信に関する情報や、Siebel ビジネス・オブジェクトとこのビジネス・オブジェクトに関連するビジネス・コンポーネントの取得に関する情報から構成されているテキスト。
DefaultValue	この値を属性に設定した場合、インバウンド・ビジネス・オブジェクトに値が設定されておらず、コネクタ・プロパティ UseDefaults が True に設定されていれば、コネクタはこの値を使用します。

表 5. 子オブジェクト属性

名前	子オブジェクトの名前
Type	子のビジネス・オブジェクト・タイプ。
ContainedObjectVersion	子ビジネス・オブジェクトのバージョン。
Relationship	子がコンテナ属性である場合、これを Containment に設定します。
IsKey	この属性は、基本ビジネス・コンポーネントに設定する必要があります。
IsForeignKey	使用しません。
IsRequired	これを True に設定すると、子は親ビジネス・オブジェクト内に表現を持つことが期待されます。Create 動詞の処理中は、基本ビジネス・コンポーネントが存在しなければなりません。このコンポーネントがインバウンド・ビジネス・オブジェクト内に存在するかどうかを確認します。コンポーネントが見つかると、要求されたオブジェクトがインバウンド・ビジネス・オブジェクト内に見つからなかったことを示すエラーがスローされない限り、作成処理が進行します。
Cardinality	親レコードに対して選択できる子レコードの個数により、1 または N。

ビジネス・オブジェクトのアプリケーション固有の情報

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では、ビジネス・オブジェクトの処理方法に関するアプリケーション依存の指示をコネクターに提供します。メタデータ主導型コネクターはサポートするビジネス・オブジェクトの設計方法に関する仮定を行うため、ビジネス・オブジェクトに対する変更は、コネクターが変更されたビジネス・オブジェクトを正しく処理できるように、コネクターの規則に合致していなければなりません。したがって、Siebel アプリケーション固有のビジネス・オブジェクトを変更または作成する場合は、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がコネクターの期待する構文に合致しているかどうかを確認してください。

ここでは、Siebel ビジネス・オブジェクト、属性、および動詞に関するアプリケーション固有の情報について説明します。

ビジネス・オブジェクトのアプリケーション固有の情報

ビジネス・オブジェクトのトップレベルにあるアプリケーション固有の情報では、Siebel ビジネス・オブジェクトの名前を指定します。例えば、親ビジネス・オブジェクト Siebel_BCAccount のオブジェクト・アプリケーション固有の情報では、以下に示すように Siebel Account オブジェクトを指定します。

```
[BusinessObjectDefinition]
Name = Siebel_BCAccount
Version = 1.0.0
AppSpecificInfo = ON=Account;CN=Account
```

複数の無関係なビジネス・コンポーネントの例:

```
[BusinessObjectDefinition]
Name = Siebel_BCInternalProduct
Version = 1.0.0
AppSpecificInfo = CN=InternalProduct
```

```
[BusinessObjectDefinition]
Name = Siebel_BCProductDefect
Version = 1.0.0
AppSpecificInfo = CN=ProductDefect
```

```
[BusinessObjectDefinition]
Name = Siebel_BOInternalProduct
Version = 1.0.0
AppSpecificInfo = ON=InternalProduct
```

```
[Attribute]
Name = Siebel_BCInternalProduct
Type = Siebel_BCInternalProduct
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 0
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
```

```
[Attribute]
Name = Siebel_BCProductDefect
Type = Siebel_BCProductDefect
ContainedObjectVersion = 1.0.0
Relationship = Containment
```

```
Cardinality = 1
MaxLength = 0
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =
```

属性アプリケーション固有情報の形式

コネクタは、単純属性とコンテナ属性でアプリケーション固有の情報を使用します。ObjectEventId 属性については、アプリケーション固有のプロパティ・フィールドは空でなければなりません。

単純属性のアプリケーション固有情報

単純属性の場合、アプリケーション固有の情報は以下の表にリストされている名前値のペアから構成されています。名前値のペアは順序に依存せず、セミコロンで区切られます。

パラメーター	説明
FN =	Siebel ビジネス・コンポーネントの対応するフィールドの名前。
PLK = ...;Restrict=<field name>;Siebel GUI Name>;<field name>;Siebel GUI Name	Siebel のビジネス・コンポーネントは、ピック・リストを使用して多対 1 の関係に関連づけることができます。ピック・リストを検索し、特定値を選んでフィールドに設定することができます。PickListKey は、Siebel 内にフィールドに関連するピック・リストがある場合に設定されます。PickListKey は、ピック・リスト・レコードを取得するために検索指定で使用される、ピック・リスト内のフィールドを識別します。場合によっては、複数の属性に基づいて PickList 値を選択します。そのような場合、ピック・リストを複数のフィールドで制限することができます。

以下の各セクションでは、これらのパラメーターの使用例を示します。

単純属性のフィールド名

単純ビジネス・オブジェクト属性のアプリケーション固有の情報では、Siebel ビジネス・コンポーネント内の対応するフィールドの名前を指定する必要があります。このためのアプリケーション固有の情報は次のとおりです。

```
FN=fieldname
```

例えば、Siebel_BCAccount ビジネス・オブジェクトで、Main Phone 属性のアプリケーション固有の情報は、Main Phone Number が Siebel Account ビジネス・コンポーネント内の対応するフィールドであることを指定します。ビジネス・オブジェクト属性のアプリケーション固有の情報を以下に示します。

```
Name = Main Phone
Type = String
IsKey = false
AppSpecificInfo = FN=Main Phone Number
```

ピック・リストを使用する外部キー関係

Siebel で、2 つのビジネス・コンポーネント間の外部キー関係はピック・リストで定義されます。関連付けられたピック・リストがフィールドにある場合、そのフィールドの PickList プロパティと PickList の対応関係は 2 つのビジネス・コンポーネントの関係を定義します。PickList の対応関係における属性のうちの 1 つは通常、Account Id や Product Id などの ID です。

ビジネス・オブジェクトの単純属性では、関連づけられたピック・リストが Siebel ビジネス・コンポーネント・フィールドにある場合に、ビジネス・オブジェクト内の属性のアプリケーション固有の情報をコード化してこの情報をコネクターに与え、コネクターが属性を外部キーとして使用できるようにする必要があります。

ある属性にピック・リストを指定するには、ビジネス・オブジェクトに 2 つの属性を組み込む必要があります。第 1 の属性は関連するビジネス・コンポーネントの外部キー・フィールドを指定し、第 2 の属性はフィールド・プロパティとしてピック・リストを持つビジネス・コンポーネント内のフィールドに対応します。2 つの属性が必要なのは、ピック・リストの関係がオブジェクト ID でなくオブジェクト名に基づいているためです。

ピック・リスト属性のアプリケーション固有の情報で、テキスト PLK を使用してこの属性がピック・リストであることを指定します。次に、ピック・リスト内のどのレコードを選択すべきかを識別するために、テキスト `PLK=...;Restrict=<field name>:<Siebel GUI Name>,<field name>:Siebel GUI Name` を使用します。

例えば、Siebel_BCAsset ビジネス・オブジェクトを作成する際、このビジネス・オブジェクトに Siebel_BCInternalProduct ビジネス・オブジェクトへの外部キーとして属性を追加するとします。Siebel Asset Mgmt ビジネス・コンポーネント内の Product Name フィールドは Internal Product ビジネス・コンポーネントに対するピック・リストであるため、キーとピック・リストに追加する属性は異なる属性にします。属性の定義は、以下に示すようにビジネス・オブジェクト内で行えます。

```
[Key Attribute]
Name = Id
Type = String
Cardinality = 1
IsForeignKey = true
AppSpecificInfo = Product Id
```

```
[PickList Attribute]
Name = ProductName
Type = string
Cardinality = n
AppSpecificInfo = FN=ProductName;PLK=Id
```

場合によっては、複数の属性に基づいて PickList 値を選択します。例えば、同じ名前の Account が複数ある場合、Account 名が唯一のピック・リスト値として設定されていれば、Contact 検索ではその名前の付いた最初の Account を取得します。正しいデータが検索されるようにするために、複数のフィールドでピック・リストを制限することができます。以下の例では、Contact ビジネス・オブジェクトが Account、Site、および City で制限されます。

```
[Key Attribute]
Name = ContactId
Type = String
Cardinality = 1
```

```
AppSpecificInfo = FN=Id
```

```
Name = Last_Name  
Type = String  
Cardinality = 1  
AppSpecificInfo = FN=Last Name
```

```
Name = First_Name  
Type = String  
Cardinality = 1  
AppSpecificInfo = FN=First Name
```

```
Name = Site  
Type = String  
Cardinality = 1  
AppSpecificInfo = N/A
```

```
Name = City  
Type = String  
Cardinality = 1  
AppSpecificInfo = N/A
```

```
Name = Account  
Type = String  
Cardinality = 1  
AppSpecificInfo = FN=Account;PLK=Name;Restrict=Location:Site,City:City
```

ピック・リスト・フィールドを制限するための `AppSpecificInfo` は、以下の構文に従います。

```
Restrict=<field name>:<Siebel GUI name>,<field name>:<Siebel GUI name>
```

制限用フィールドの数は無制限です。Restrict パラメーターの後の属性間にはスペースを入れないでください。制限用フィールドはすべて、ビジネス・オブジェクトに対して属性として追加しなければならず、`AppSpecificInfo` を持っていないではありません。これらの属性は、制限用フィールドのプレースホルダーの役割を果たします。

Retrieve で、アプリケーション固有の情報 `PLK=Id` は `ProductName` 属性がピック・リスト・ビジネス・コンポーネントに対応することを指定し、set パラメーターはコネクタがどのレコードをピックすべきかを ID の値で示すことを指定します。

ピック・リスト関係の中には、ピックされた子の作成を必要とするものがあります。例えば、`Siebel_BCQuote` 内の `Account` と `Quote` の間のピック・リスト関係があります。Create では、以下のように `Quote` ビジネス・コンポーネントに対する包含関係を使用して新しい `Account` ビジネス・コンポーネントを作成する必要があります。

```
[Business Object Definition]  
Name = Siebel_BCQuote  
Version = 7.0.0  
Relationship = Containment  
AppSpecificInfo = ON=Quote;CN=Quote
```

```
Name = Account  
Type = String  
AppSpecificInfo = FN=AccountId
```

```
Name = Account  
Type = Siebel_BCAccount  
Relationship = Containment  
Cardinality = 1
```

```
IsForeignKey = false
AppSpecificInfo = LFN=Account;PL=true;From=AccountId;To=AccountId
```

```
[Siebel_BCAccount]
Name = AccountId
IsKey = true
AppSpecificInfo = ...
```

ピック・リスト属性については、以下のビジネス処理に関するヒントに注意してください。

- Retrieve 操作では、ピック・リスト属性の値をピック・リスト・ビジネス・コンポーネントの名前に対応させ、キー属性の値をキーに対応させます。
- Create または Update 操作では、ピック・リスト属性をキーに対応させ、キー属性をヌル値に対応させます。ピック・リストのリンクはフィールドの名前で定義されるため、コネクタはキー属性値を任意の値に設定することができ、Siebel は値を検証しません。ピック・リスト属性にキー値が含まれている場合、ピック・リスト・コンポーネントを使用してピック操作を行うと、キーの検証が行われます。ピック操作でフィールドが見つかると、ピック対応関係内のすべての属性が新しいオブジェクトに追加され、新しいオブジェクトが作成されます。
- ピック・リストからリンクを除去するには、ピック・リスト属性の値をヌルに対応させ、キー属性の値をブランクに対応させます。

コンテナ属性のアプリケーション固有情報

コンテナ属性の場合、アプリケーション固有の情報には以下の表にリストされている名前値のペアが含まれています。名前値のペアは順序は関係なく、セミコロンで区切られます。

パラメーター	説明
LFN = ...;	Siebel ビジネス・コンポーネントに関連する Multi Value Field Name。
MVL = ...;	MVL を Active に設定すると、1 対多の関係を指定したことになります。Inactive に MVL を設定することは、親オブジェクトと子オブジェクトの間に非アクティブな複数値リンクの関係があることを示します。つまり、親オブジェクトが複数値フィールドを持っていないということです。
PL = ...	PL を True に設定すると、それは多対 1 の関係があることを示します。
Assoc = ...	Assoc を True に設定すると、それは関係が交差テーブルを通じて多対多であることを示します。関係が 2 つのビジネス・コンポーネント間の関連であるときは、アクティブな複数値リンクが存在する場合と存在しない場合があります。複数値リンクが存在するかどうかに基づいて、アプリケーション固有情報に MVL=Active;Assoc=true が含まれます。アプリケーション固有情報に Assoc=true のみが含まれる場合、デフォルトではアクティブな複数値リンクが存在することになります。明示的に指定する場合は、アプリケーション固有情報には MVL=Active;Assoc=true が含まれます。複数値リンクが存在しない場合、アプリケーション固有情報には MVL=Inactive;Assoc=true が含まれます。

パラメーター	説明
From = ...; To = ...	これらは、To 属性を From 属性の値に設定するためのコネクタに対するプリプロセッシング命令です。From 属性は取り込む必要がありますが、True 属性はヌルのときのみ設定されます。これらの属性を含むオブジェクトには、1 対 1 の関係がなければなりません。これは Retrieve 操作で使用され、取り出す必要のある子レコードを指定します。
SF	ビジネス・オブジェクトの下にビジネス・コンポーネント構造のマスター詳細ビューを表示する単純リンク。SF はソース・フィールドを表します。
DF	ビジネス・オブジェクトの下にビジネス・コンポーネント構造のマスター詳細ビューを表示する単純リンク。DF は宛先フィールド (外部キー) を表します。

コンテナ属性のフィールド名

親オブジェクトと子オブジェクト間の関係が 1 対多である場合、子ビジネス・オブジェクトを参照するコンテナ属性に対するアプリケーション固有情報には、その親ビジネス・コンポーネントに関連する Multi-Value Field の名前を指定する必要があります。Multi-Value Field は、Siebel における親と子のビジネス・コンポーネント間の関係を定義する Multi-Value Link を表します。このためのアプリケーション固有の情報は次のとおりです。

LFN=multiValueFieldName

例えば、Siebel アプリケーションで Siebel ビジネス・コンポーネント Account に複数值フィールド Street Address がある場合、対応する WebSphere Business Integration システムのビジネス・オブジェクト Siebel_BCAccount には、子ビジネス・オブジェクト Siebel_BCBusinessAddress のコンテナ属性があります。このコンテナ属性のアプリケーション固有の情報では、Siebel Business Address ビジネス・コンポーネントへのリンクを含む Multi-Value Field として Street Address を指定します。

```
[Example of Container Attribute]
Name = PrimaryAddress
Type = Siebel_BCBusinessAddress
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = n
IsKey = false
IsForeignKey = false
AppSpecificInfo = MVL=Active;LFN=Street Address
```

親ビジネス・オブジェクトと子ビジネス・オブジェクトの関係

コンテナ属性のアプリケーション固有の情報には、フィールド名のほか、Siebel 内の親と子のビジネス・コンポーネント間の関係の型を定義するパラメーターを含めることができます。

関係型パラメーターを 0 に設定するのは無効です。関係型を false に設定する際は、そのパラメーターを含めないでください。

例えば、前のセクションで示した子ビジネス・オブジェクト Siebel_BCBusinessAddress のコンテナ属性には、Street Address が Siebel

Account ビジネス・コンポーネントを Siebel Business Address ビジネス・コンポーネントにリンクする Multi Value Field であることを示すパラメーターを含めることができます。

```
[Example of Container Attribute]
Name = PrimaryAddress
Type = Siebel_BCBusinessAddress
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = n
IsKey = false
IsForeignKey = false
AppSpecificInfo = MVL=Active;LFN=Street Address
```

ここに、子ビジネス・オブジェクト Siebel_BCOppportunity のもう一つの例を示します。このビジネス・オブジェクトには、Siebel_BCContact または Association に対して多対多の関係があります。この場合、Create 操作では、コネクタはコンテナに取り込まれたビジネス・オブジェクト・フィールドを使用してビジネス・コンポーネントを検索します。コネクタは一致するオブジェクトを見つけると、そのオブジェクトを親ビジネス・コンポーネントに関連づけます。オブジェクトが見つからないと、エラーが記録され、ビジネス・オブジェクト要求は失敗します。

```
Name = Siebel_BCContact
Version = 1.0.0
AppSpecificInfo = ON=Contact;CN=Contact
```

```
Name = ContactId
Type = String
Cardinality = 1
MaxLength = 10
IsKey = true
IsForeignKey = false
IsRequired = false
AppSpecificInfo = FN=Id
```

```
Name = ...
```

```
Name = ...
```

```
Name = Siebel_BCOppportunity
Type = Siebel_BCOppportunity
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = N
MaxLength = 0
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo =LFN=Oppportunity;Assoc=true
```

この例は、単純リンクの関係の場合です。この場合、Quote と Order Entry の間に単純リンクがあります。

```
Name = Siebel_BCQuote
Version = 1.0.0
AppSpecificInfo = ON=Quote;CN=Quote
Name = QuoteId
Type = String
Cardinality = 1
MaxLength = 10
IsKey = true
IsFireignKey = false
IsRequired - false
AppspecificInfo = FN=Id
```

```
Name = ...
```

```
Name = ...
```

```
Name = Siebel_BCOrderEntry  
Type = Siebel_BCOrderEntry  
ContainedObjectVersion = 1.0.0  
Relationship = Containment  
Cardinality = N  
MaxLength = 0  
IsKey = true  
IsFireignKey = false  
IsRequired = false  
AppspecificInfo = SF=QuoteId;DF=QuoteId
```

他の属性への属性値の割り当て

属性アプリケーション固有の情報をコード化して、コネクターが属性の値を取得し、第 2 の属性が処理される前に別の属性にその値を指定することができます。この機能は、Retrieve 操作で主に検索する子のレコードを指定するために、コンテナ属性で使用します。

この機能を使用するには、ビジネス・オブジェクト定義内の属性アプリケーション固有の情報を編集して以下のテキストを組み込みます。

```
From=attribute;To=attribute;
```

属性パス値は、現行のビジネス・オブジェクト内の属性名とすることができます。以下の規則に注意してください。

- From 属性は親からの属性であり、To 属性は子の属性である。
- 命令内の To 属性が処理される前に From 属性を取り込まなければならない。
- To 属性は、ヌル値を持っている場合のみ設定される。
- From パラメーターに対してパスが無効であれば、To パラメーターはヌルに設定される。To パラメーターに対してパスが無効であれば、エラーのフラグは付かない。
- From/To ディレクティブを指定できるのは、子ビジネス・オブジェクト上の属性のアプリケーション固有の情報内のみである。つまり、トップレベル・ビジネス・オブジェクトではそのディレクティブを指定できない。

例えば、Siebel_BCQuote ビジネス・オブジェクトに子ビジネス・オブジェクト Siebel_BCAccount が含まれている場合、Siebel_BCQuote オブジェクト内の属性は、ピック・リストのどのアドレスを検索するかを指定することができます。この例で、AccountId はキー属性、Siebel_BCAccount はピックされるオブジェクトです。コネクターは AccountId 属性の値を取得し、その値を使用して特定アカウントを検索します。子属性は、親ビジネス・オブジェクト内の属性の後で処理されます。次の例は、親ビジネス・オブジェクトから子ビジネス・オブジェクトへの属性の処理の流れを示しています。

```
[Siebel_BCQuote]  
Name = Account  
Type = String  
AppSpecificInfo = FN=Id
```

```
Name = Account  
Type = Siebel_BCAccount  
Relationship = Containment
```

```
Cardinality = 1
IsForeignKey = false
AppSpecificInfo = LFN=Address;PL=true;From=AccountId;To=AccountId
```

```
[Siebel_BCAccount]
Name = AccountId
IsKey = true
AppSpecificInfo = ...
```

ピック・リストの関係の指定

ピック・リストの関係の中には、同じトランザクション内のピックされた子オブジェクトの作成を必要とするものがあります。WebSphere Business Integration システムのビジネス・オブジェクトでは、親ビジネス・オブジェクトと子ビジネス・オブジェクトの間のピック・リスト関係は、ピックされたオブジェクトのキー属性および単一カーディナリティー・コンテナ属性の 2 つの属性で表されます。この属性集合を使用して、ピック・リスト・マップに含まれていないピック・リスト・ビジネス・コンポーネントのいくつかの属性またはすべての属性を検索することができます。

例えば、Quote と Opportunity との間のピック・リスト関係を指定するための 2 つの属性を含めるように、Siebel_BCQuote ビジネス・オブジェクトを設計することができます。以下に示すように、OpportunityId はキー属性、Opportunity はピック・リスト・オブジェクトです。

```
Name = Siebel_BCQuote
Version = 1.0.0
AppSpecificInfo = ON=Quote;CN=Quote
```

```
Name = ...
```

```
Name = ...
```

```
Name = OpportunityId
Type = String
Cardinality = 1
MaxLength = 10
IsKey = false
IsForeignKey = true
IsRequired = false
AppSpecificInfo = FN= OpportunityId
```

```
Name = Siebel_BCOpportunity
Type = Siebel_BCOpportunity
ContainedObjectVersion = 1.0.0
Relationship = Containment
Cardinality = 1
MaxLength = 0
IsKey = false
IsForeignKey = false
IsRequired = false
AppSpecificInfo = LFN=Opportunity;PL=true;From=OpportunityId;To=OpportunityId
```

アプリケーション固有の情報の PL=true はコンテナ属性がピック・リストを表すことを示しています。From= パラメーターはキー属性へのポインターであり、To= パラメーターは Siebel_BCOpprotunity ビジネス・オブジェクトのキー属性を指しています。

Retrieve 操作では属性の順序が重要です。これは、OpportunityId の値を検索してからでないと、子オブジェクト内の外部キーとして定義できないためです。Create または Update 操作では、OpportunityId の値が外部キーであり、オブジェクトの作成後に検索されます。

ピック・リスト・コンテナーとして完全なビジネス・オブジェクトを使用する必要はありません。必要なキーのみが設定されたオブジェクトで十分です。コネクタは、ピック・リスト・コンテナーを処理するために以下の規則を適用します。

- ピック・リスト・ビジネス・オブジェクトのキー属性を設定しないと、新しいオブジェクトが Siebel 内に作成され、ピックされる。
- ピック・リスト・ビジネス・オブジェクトのキー属性を設定すると、コンテナーはオブジェクトを検索し、ピックする。そのビジネス・オブジェクトのピック・リスト・オブジェクトが見つからないと、コネクタはエラーを記録する。オブジェクト・キーが無効な場合、エラーが発生することがある。

以下に、コンテナー・ビジネス・オブジェクトのピック・リスト属性のマップに関するガイドラインを示します。

- コラボレーションからコネクタへのビジネス・オブジェクト要求である場合、キー属性のマッピングは、上記の単純属性の場合と同じガイドラインに従わなければならない。
- コンテナー属性のマッピングは、キーが既知の場合のみキーでなければならない。
- ピック・リスト・オブジェクトを作成する場合、ピック・リスト・オブジェクトに関して指定されたとおりにすべての必須属性をマップする。
- Delete 操作で、キー属性をスペースに設定し、ピック・リスト・コンテナー属性を null に設定する。

動詞アプリケーション固有情報の形式

ビジネス・オブジェクトの Retrieve 動詞に関するアプリケーション固有の情報は、コネクタが検索ごとに一定数のオブジェクトを検索するように指定できます。オブジェクトのサブセットを検索するアプリケーション固有情報は、max=n です。5 つのオブジェクトのみを検索するように指定された Retrieve 動詞の例は、次のとおりです。[Verb] Name = Retrieve AppSpecificInfo = max=5

他の動詞の場合、アプリケーション固有のプロパティは使用せず、ブランクのままにするか、ビジネス・オブジェクト定義の作成時に省略してください。

複数のレコード検索

アダプターは、インバウンド IBM ビジネス・オブジェクトに対してキーがすべて設定されていることをチェックしないという点を除き、Retrieve 動詞を処理するのと同様の方法で RetrievebyContent 動詞を処理します。

Adapter for Siebel は、ラッパー・ビジネス・オブジェクトをサポートします。39 ページの『ビジネス・オブジェクトの構造および関係』を参照してください。アダプターは、RetrievebyContent 動詞とラッパー・ビジネス・オブジェクトを使用して、要求処理中に複数のレコードを検索できます。複数カーディナリティー・コン

テナーを持つラッパー・ビジネス・オブジェクトの場合、一致するレコードがすべて戻されます。単一カーディナリティーの子の場合は、1つのレコードのみが処理され、MULTIPLE_HITS が戻されます。

Create 動詞および Update 動詞のキー属性

Create または Update 要求では、Object Key 値が RowId と異なる場合、Siebel アプリケーションは、Object Key 属性を消去してそのレコード用の独自の RowId を作成します。

重要: Create 要求および Update 要求では、キー属性として RowId を使用してください。

第 5 章 ビジネス・オブジェクトの作成

- 『ビジネス・オブジェクトのサンプルの変更』
- 57 ページの『Siebel ODA の概要』
- 57 ページの『Siebel ODA のインストールおよび使用』
- 61 ページの『Business Object Designer Express 内での SiebelODA の使用』
- 74 ページの『生成された定義の検討』
- 77 ページの『ビジネス・オブジェクト定義への情報の追加』

ビジネス・オブジェクトのサンプルの変更

ビジネス・オブジェクトのサンプルには、アダプターのコネクター・コンポーネントが含まれています。すべてのオブジェクトは、ポーリング用の Siebel トリガーが設定されていなければなりません。場合によっては、オブジェクトを Siebel Tools でカスタマイズする必要があります。このセクションではサンプル・オブジェクトについて説明し、それらのカスタマイズ方法の例を示します。

- 53 ページの『Siebel_BCAccount』
- 54 ページの『Siebel_BCQuote』
- 55 ページの『Siebel_BCContact』
- 56 ページの『Siebel_BCInternalProduct』
- 56 ページの『Siebel_BCAsset』

Siebel_BCAccount

1. 以下の Siebel VB ファイルを見つけます。

Siebel 6

Account_Write.svb, Account_PreDelete.svb, Business_Address_PreDelete.svb, and Business_Address_Write.svb.

Siebel 7

Account.sbl or Account.js

Siebel VB ファイルは、Common/Siebel/Dependencies/Siebel_VB ディレクトリ一内にあります。

2. ファイルを一時ディレクトリに保存します。例えば、ディレクトリ名として Temp を使用します。
3. CWCNN アカウントに Siebel 内のシステム管理者責任を与えます。
4. Siebel Tools 環境で、Account および Contact プロジェクトをチェックアウトし、ロックします。
5. 以下のように、VB スクリプトを Account ビジネス・コンポーネントに追加します。

- a. Account ビジネス・コンポーネントを右マウス・ボタン・クリックし、メニューから「Basic スクリプトの編集 (Edit Basic Scripts)」を選択します。スクリプト・エディターが起動されます。
 - b. 「ファイル」メニューから「インポート」を選択して Temp ディレクトリーを選択し、Account_Write.svb をピックして Bus Comp オブジェクトおよび BusComp_WriteRecord プロシージャに VB コードをインポートします。
 - c. 「ファイル」メニューから「インポート」を選択して Temp ディレクトリーを選択し、Account_PreDelete.svb をピックして Bus Comp オブジェクトおよび BusComp_PreDeleteRecord プロシージャに 2 番目の VB コードをインポートします。
 - d. オブジェクトに対する変更を保存します。
6. 以下のように、スクリプトを Business Address ビジネス・コンポーネントに追加します。
- a. Business Address ビジネス・コンポーネントを右マウス・ボタン・クリックし、メニューから「Basic スクリプトの編集 (Edit Basic Scripts)」を選択します。スクリプト・エディターが起動されます。
 - b. 「ファイル」メニューから「インポート」を選択して Temp ディレクトリーを選択し、Business_Address_Write.svb をピックして Bus Comp オブジェクトおよび BusComp_WriteRecord プロシージャに VB コードをインポートします。
 - c. 「ファイル」メニューから「インポート」を選択して Temp ディレクトリーを選択し、Business_Address_PreDelete.svb をピックして Bus Comp オブジェクトおよび BusComp_PreDeleteRecord プロシージャに 2 番目の VB コードをインポートします。
 - d. オブジェクトに対する変更を保存します。
 - e. Account Id 属性の Business Address ビジネス・コンポーネントで、Force Active フィールドを TRUE に設定します。
- 注: ビューが使用されているために、このフィールドが取り込まれないことがあります。
7. すべてのオブジェクトを保存し、結果をコンパイルするまでは、オブジェクトをテストすることはできません。

Siebel_BCQuote

1. 以下の Siebel VB ファイルを見つけます。

Siebel 6

Quote_Write.svb, Contact_PreDelete.svb

Siebel 7

Quote.sbl, Quote.js

Siebel VB ファイルは、Common/Siebel/Dependencies/Siebel_VB ディレクトリー内にあります。

2. ファイルを一時ディレクトリーに保存します。例えば、ディレクトリー名として Temp を使用します。

3. CWCONN アカウントに Siebel 内のシステム管理者責任を与えます。
4. Siebel Tools 環境で、Quote プロジェクトをチェックアウトし、ロックします。
5. 以下のように、スクリプトを Quote ビジネス・コンポーネントに追加します。
 - a. Quote ビジネス・コンポーネントを右マウス・ボタン・クリックし、メニューから「Basic スクリプトの編集 (Edit Basic Scripts)」を選択します。スクリプト・エディターが起動されます。
 - b. 「ファイル」メニューから「インポート」を選択して Temp ディレクトリーを選択し、Quote_Write.svb をピックして Bus Comp オブジェクトおよび BusComp_WriteRecord プロシージャに VB コードをインポートします。
 - c. オブジェクトに対する変更を保存します。
6. すべてのオブジェクトを保存し、結果をコンパイルするまでは、オブジェクトをテストすることはできません。

Siebel_BCContact

1. 以下の Siebel VB ファイルを見つけます。

Siebel 6

Contact_Write.svb, Contact_PreDelete.svb.

Siebel 7

Contact.sbl, Contact.js

Siebel VB ファイルは、Common/Siebel/Dependencies/Siebel_VB ディレクトリー内にあります。

2. Siebel VB ファイルを一時ディレクトリーに保存します。例えば、ディレクトリー名として Temp を使用します。
3. CWCONN アカウントに Siebel 内のシステム管理者責任を与えます。
4. Siebel Tools 環境で、Contact プロジェクトをチェックアウトし、ロックします。
5. 以下のように、Siebel VB スクリプトを Contact ビジネス・コンポーネントに追加します。
 - a. Contact ビジネス・コンポーネントを右マウス・ボタン・クリックし、メニューから「Basic スクリプトの編集 (Edit Basic Scripts)」を選択します。スクリプト・エディターが起動されます。
 - b. 「ファイル」メニューから「インポート」を選択して VB コードをインポートし、一時ディレクトリーを選択して、Bus Comp オブジェクトおよび BusComp_WriteRecord プロシージャにインポートするために Contact_Write.svb をピックします。
 - c. 「ファイル」メニューから「インポート」を選択して Temp ディレクトリーを選択し、Contact_PreDelete.svb をピックして Bus Comp オブジェクトおよび BusComp_PreDeleteRecord プロシージャに 2 番目の VB コードをインポートします。
 - d. オブジェクトに対する変更を保存します。
6. すべてのオブジェクトを保存し、結果をコンパイルするまでは、オブジェクトをテストすることはできません。

Siebel_BCInternalProduct

1. InternalProduct_Write.svb ファイルを見つけます。Siebel VB ファイルは、Common/Siebel/Dependencies/Siebel_VB ディレクトリー内にあります。
2. ファイルを一時ディレクトリーに保存します。例えば、ディレクトリー名として Temp を使用します。
3. CWCONN アカウントに Siebel 内のシステム管理者責任を与えます。
4. Siebel Tools 環境で、Product プロジェクトをチェックアウトし、ロックします。
5. Siebel VB スクリプトを Internal Product ビジネス・コンポーネントに追加します。
 - a. Internal Product ビジネス・コンポーネントを右マウス・ボタン・クリックし、メニューから「Basic スクリプトの編集 (Edit Basic Scripts)」を選択します。スクリプト・エディターが起動されます。
 - b. 「ファイル」メニューから「インポート」を選択して Temp ディレクトリーを選択し、InternalProduct_Write.svb をピックして Bus Comp オブジェクトおよび BusComp_WriteRecord プロシージャに VB コードをインポートします。
 - c. オブジェクトに対する変更を保存します。
6. 以下のように、ビジネス・コンポーネントのプロパティーを変更します。
 - a. ツールバーで、「表示 (View)」 > 「プロパティー・ウィンドウ (Property Window)」と選択します。
 - b. 「Business Component/Internal Product」に移動します。
 - c. 以下の属性を False に変更します。
No Insert = False
No Merge = False
No Update = False

注: 上記プロパティーを変更する目的は、Com Data Server Interface を使用して Siebel に対してインバウンドで製品の作成と更新を行えるようにすることです。
7. すべてのオブジェクトを保存し、結果をコンパイルするまでは、オブジェクトをテストすることはできません。

Siebel_BCAsset

1. Siebel VB ファイル Asset_Write.svb を見つけます。Siebel VB ファイルは、Common/Siebel/Dependencies/Siebel_VB ディレクトリー内にあります。
2. Siebel VB ファイルを一時ディレクトリーに保存します。例えば、ディレクトリー名として Temp を使用します。
3. CWCONN アカウントに Siebel 内のシステム管理者責任を与えます。
4. Siebel Tools 環境で、Asset Management プロジェクトをチェックアウトし、ロックします。
5. 以下のように、Siebel VB スクリプトを Asset Mgmt - Asset ビジネス・コンポーネントに追加します。

- a. Asset Mgmt - Asset ビジネス・コンポーネントを右マウス・ボタン・クリックし、メニューから「Basic スクリプトの編集 (Edit Basic Scripts)」を選択します。スクリプト・エディターが起動されます。
 - b. 「ファイル」メニューから「インポート」を選択して Temp ディレクトリーを選択し、Asset_Write.svb をピックアップして Bus Comp オブジェクトおよび BusComp_WriteRecord プロシージャに Write VB コードをインポートします。
 - c. オブジェクトに対する変更を保存します。
6. 以下のように、ビジネス・コンポーネントのプロパティーを変更します。
- a. ツールバーで、「表示 (View)」>「プロパティー・ウィンドウ (Property Windows)」と選択します。
 - b. 「Business Component/Asset Mgmt - Asset」に移動します。
 - c. フィールドに移動します。
 - d. 以下のように、フィールド・プロパティー値を変更します。
 - 「Account Id」を選択し、「Inactive」の値を「False」に設定します。(このフィールドは、IBM WebSphere Business Integration システム・オブジェクトで必要となります。)
 - 「Name」を選択し、値を「Not Required」に設定します。
 - 「Product Id」を選択し、値を「Required」に設定します。
- 注: WebSphere Business Integration コンポーネントが製品をトラッキングする際には、製品の名前ではなく、製品 ID を使用します。
7. すべてのオブジェクトを保存し、結果をコンパイルするまでは、オブジェクトをテストすることはできません。

Siebel ODA の概要

このセクションでは、SiebelODA について説明します。これは Object Discovery Agent (ODA) で、コネクターのビジネス・オブジェクト定義を作成します。SiebelODA は、Siebel Java API を使用して Siebel アプリケーション・サーバーから Siebel ビジネス・オブジェクトとビジネス・コンポーネントに関する情報を取得します。次に、この情報を使用して新しいビジネス・オブジェクト定義を構築します。SiebelODA は、既存のビジネス・オブジェクト定義をコネクターでサポートされる定義に変換できるようにします。

Siebel ODA のインストールおよび使用

ここでは、以下のことについて説明します。

- 58 ページの『SiebelODA のインストール』
- 58 ページの『SiebelODA を使用する前に』
- 59 ページの『SiebelODA の起動』
- 60 ページの『複数のマシンにおける SiebelODA の実行』
- 60 ページの『エラーおよびトレース・メッセージ・ファイルの処理』

SiebelODA のインストール

SiebelODA をインストールするには、アダプター・インストーラーを使用します。「システム・インストール・ガイド(Windows版)」に示す指示に従ってください。インストールが完了すると、システムの製品をインストールしたディレクトリーに以下のファイルがインストールされます。

- ODA¥Siebel¥SiebelODA.jar
- ODA¥messages¥SiebelODAAgent.xsd
- ODA¥Siebel¥start_SiebelODA.bat (Windows のみ)
- bin¥CWODAEV.bat (Windows のみ)

注: ご使用のブローカーが InterChange Server Express であるため、CWODAEV.bat を編集して InterChange Server Express のバージョンを反映させる必要があります。InterChange Server Express バージョン 4.2.x の場合は、CWVERSION を 4.2 に変更してください。InterChange Server Express バージョン 4.1.1 の場合は、CWVERSION を 4.1 に変更してください。

注: 本書では、特に断りのない限り、ディレクトリー・パスの規則として円記号 (¥) を使用します。すべての IBM 製品パス名は、製品がシステムにインストールされているディレクトリーを基準にした相対パス名です。

SiebelODA を使用する前に

SiebelODA を実行する前に、必要な Siebel アプリケーションの .jar ファイルを %ProductDir%/connectors/Siebel/dependencies ディレクトリーにコピーしてください。以下のファイルをこのディレクトリーにコピーしてください。

Siebel 7.7

SiebelJI_enu.jar
Siebel.jar

Siebel 7.x

SiebelJI_enu.jar
SiebelJI_Common.jar

Siebel のデフォルト・バージョンは 7.x に設定されています。以下の行の REM が除去されていないことを確認してください。

```
REM set SIEBELVERSION="6.x"
```

Siebel 6.2.x

SiebelDataBean.jar
SiebelTC_enu.jar
SiebelTcCommon.jar
SiebelTcOM.jar

start_SiebelODA.bat ファイルを編集して以下の行の REM を除去しなければなりません。

```
REM set SIEBELVERSION ="6.X"
```

SiebelODA をインストールした後、以下のことを行ってビジネス・オブジェクトを生成してください。

1. ODA を起動します。
2. Business Object Designer Express を起動します。
3. Business Object Designer Express の 6 ステップのプロセスに従って、ODA を構成して実行します。

以降のセクションで、これらのステップについて詳しく説明します。

データ型のマッピング

以下の表は、Siebel アプリケーションのデータ型、および対応する WBI ビジネス・オブジェクト定義のデータ型を示しています。これらのデータ型は WBI Adapter for Siebel ODA でのみ使用されます。アプリケーションはすべての属性値をストリングと見なします。

表 6. Siebel アプリケーションおよび WBI Adapter のデータ型

Siebel のデータ型	WBI Adapter for Siebel のデータ型
DTYPE_BOOL	Boolean
DTYPE_ID	String
DTYPE_PHONE	String
DTYPE_TEXT	String
DTYPE_NOTE	String
DTYPE_DATE	Date
DTYPE_TIME	Date
DTYPE_DATETIME	Date
DTYPE_UTCDATETIME	Date
DTYPE_INTEGER	Integer
DTYPE_NUMBER	Float
DTYPE_CURRENCY	Double

以下のデータ型マッピングは、「ビジネス・サービス (Business Services)」オプションを使用してビジネス・オブジェクト定義を作成するとき使用されます。

Hierarchy	Container または String。「統合オブジェクト (Integration Object)」フィールドに取り込まれた内容によって異なります。
Integration object	Container または String。「統合オブジェクト (Integration Object)」フィールドに取り込まれた内容によって異なります。
Number	Integer
String	String
Date	Date

SiebelODA の起動

SiebelODA は、オペレーティング・システムに応じた適切な始動スクリプトで起動できます。

Windows

```
start_SiebelODA.bat
```

注: Windows インストーラーは、インストールした ODA を始動するためのショートカットを提供します。このインストーラーを使用して SiebelODA をインストールした場合は、「プログラム」メニュー>「IBM WebSphere Business Integration Server Express Adapters」>「アダプター」>「Object Discovery Agents」に始動用ショートカットが作成されます。

Business Object Designer Express を使用して SiebelODA を構成し、実行します。Business Object Designer Express はビジネス・オブジェクト・ウィザードを起動します。ビジネス・オブジェクト・ウィザードは、各スクリプトまたはバッチ・ファイルの AGENTNAME 変数で指定された名前で各 ODA を探し出します。このコネクタのデフォルト ODA 名は SiebelODA です。

複数のマシンにおける SiebelODA の実行

ネットワーク内のローカル・ホストまたはリモート・ホストのいずれかで、ODA の複数インスタンスを実行することができます。各インスタンスは、固有のポートで実行されます。

エラーおよびトレース・メッセージ・ファイルの処理

エラー・メッセージ・ファイルおよびトレース・メッセージ・ファイル (デフォルトでは SiebelODAAgent.txt) は、製品ディレクトリーの下の %ODA%\messages サブディレクトリーにあります。これらのファイルには、次の命名規則が使用されます。

AgentNameAgent.txt

ODA スクリプト・ファイルまたはバッチ・ファイルの複数のインスタンスを作成し、各インスタンスに対応する ODA に固有の名前を指定した場合には、各 ODA インスタンスに対応するメッセージ・ファイルを持つことができます。異なる名前の付いた ODA インスタンスが複数存在しても、メッセージ・ファイルは共通にすることも可能です。有効なメッセージ・ファイルを指定する 2 つの方法があります。

- ODA の名前を変更し、それに対応するメッセージ・ファイルを作成しない場合には、ODA 構成の一部として、Business Object Designer Express でメッセージ・ファイルの名前を変更する必要があります。Business Object Designer Express はメッセージ・ファイルの名前を指定しますが、実際にファイルを作成するわけではありません。ODA 構成の一部として表示されたファイルが存在しない場合には、既存のファイルを指すように値を変更してください。
- 特定の ODA に対応する既存のメッセージ・ファイルをコピーし、必要に応じて変更することもできます。Business Object Designer Express は、各ファイルが命名規則に従って命名されることを前提としています。例えば、AGENTNAME 変数が SiebelODA1 を指定する場合、Business Object Designer Express は、対応するメッセージ・ファイルの名前が SiebelODA1Agent.txt であると想定します。し

たがって、Business Object Designer Express が確認のため ODA 構成の一部としてファイル名を提供するとき、このファイル名は ODA 名に基づいています。デフォルトのメッセージ・ファイルが正しく命名されていることを確認し、必要ならば訂正してください。

重要: ODA の構成時にメッセージ・ファイルの名前を正しく指定できなかった場合には、ODA はメッセージなしに稼働します。メッセージ・ファイル名の指定方法の詳細については、63 ページの表 8 を参照してください。

構成処理中は、以下の項目を指定します。

- SiebelODA がエラーおよびトレース情報を書き込むファイルの名前
- トレースのレベル。0 から 5 の範囲です。

表 7 で、これらの値を説明します。

表 7. トレース・レベル

トレース・レベル	説明
0	すべてのエラーを記録する
1	メソッドのすべての開始メッセージおよび終了メッセージをトレースする
2	ODA のプロパティとそれらの値をトレースする
3	すべてのビジネス・オブジェクトの名前をトレースする
4	作成されたすべてのスレッドの詳細をトレースする
5	<ul style="list-style-type: none">• ODA のすべてのプロパティの初期値を示す• SiebelODA により作成された各スレッドの詳細状況をトレースする• ビジネス・オブジェクト定義ダンプをトレースする

これらの値を設定する場所については、63 ページの表 8 を参照してください。

Business Object Designer Express 内での SiebelODA の使用

このセクションでは、Business Object Designer Express を使用し、SiebelODA を使用してビジネス・オブジェクト定義を生成する方法について説明します。Business Object Designer Express の起動については、「ビジネス・オブジェクト開発ガイド」を参照してください。Business Object Designer Express には、これらの各ステップをガイドするウィザードがあります。このウィザードをビジネス・オブジェクト・ウィザードと呼びます。ODA を起動した後、Business Object Designer Express を起動し、ビジネス・オブジェクト・ウィザードへのアクセスを取得する必要があります (ビジネス・オブジェクト・ウィザードが ODA を構成して実行します)。ビジネス・オブジェクト・ウィザードには、ODA を使用してビジネス・オブジェクト定義を作成するための 6 つのステップがあります。

ODA を起動した後、以下のことを行ってウィザードを起動してください。

1. Business Object Designer Express を開きます。
2. 「ファイル」メニューから、「ODA を使用して新規作成...」サブメニューを選択します。

ビジネス・オブジェクト・ウィザードは、ウィザードに「エージェントの選択」という名前の最初のウィンドウを表示します。図2は、このウィンドウを図示したものです。

ODA の選択、構成、実行を行うには、以下のステップに従います。

1. 『ODA の選択』
2. 63 ページの『構成プロパティの指定』
3. 65 ページの『ソースの選択』
4. 70 ページの『オブジェクトの選択の確認』
5. 70 ページの『ビジネス・オブジェクト定義の生成』および、オプションとして、71 ページの『追加情報の入力』
6. 74 ページの『ビジネス・オブジェクト定義の保管』

ODA の選択

図2に、ビジネス・オブジェクト・ウィザードの6つのステップのウィザードのうち、最初のダイアログ・ボックスを示します。このウィンドウから、実行する ODA を選択します。

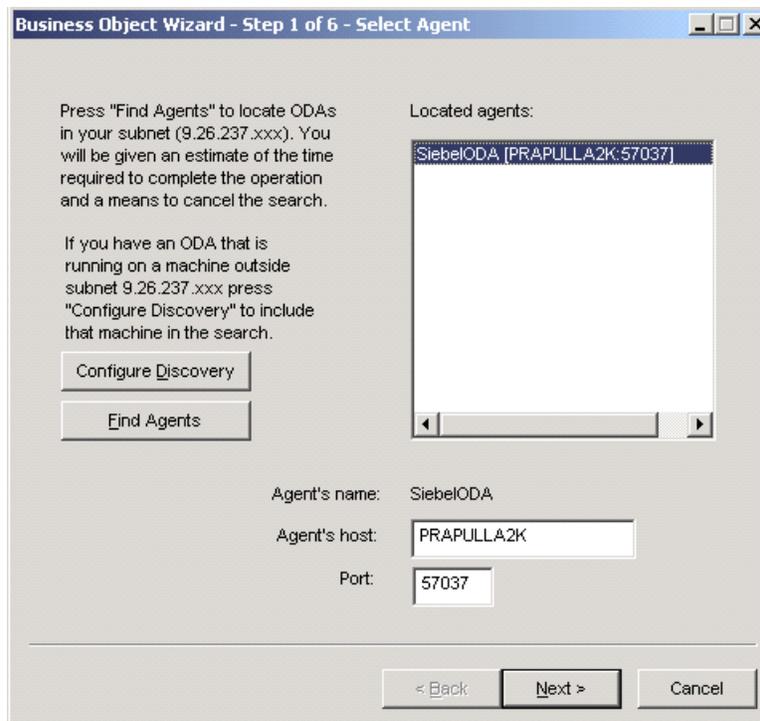


図2. ビジネス・オブジェクト・ウィザードの「エージェントの選択」画面

ODA を選択するには、次の操作を行います。

1. 「エージェントの検索」ボタンをクリックして、「検索されたエージェント」フィールド内にすべての登録済み ODA または現在実行中の ODA を表示します。代わりに、ホスト名とポート番号を使用して ODA を検索することもできます。

注: 希望する ODA がビジネス・オブジェクト・ウィザードで検索されない場合は、ODA のセットアップを検査してください。

- 表示されたリストから、必要な ODA を選択します。

ビジネス・オブジェクト・ウィザードは、エージェントの名前のフィールドにユーザーの選択項目を表示します。

- 「次へ」をクリックします。

構成プロパティの指定

ビジネス・オブジェクト・ウィザードが初めて SiebelODA と通信する場合、図 4 に示すように、一連の ODA 構成プロパティの入力が要求されます。

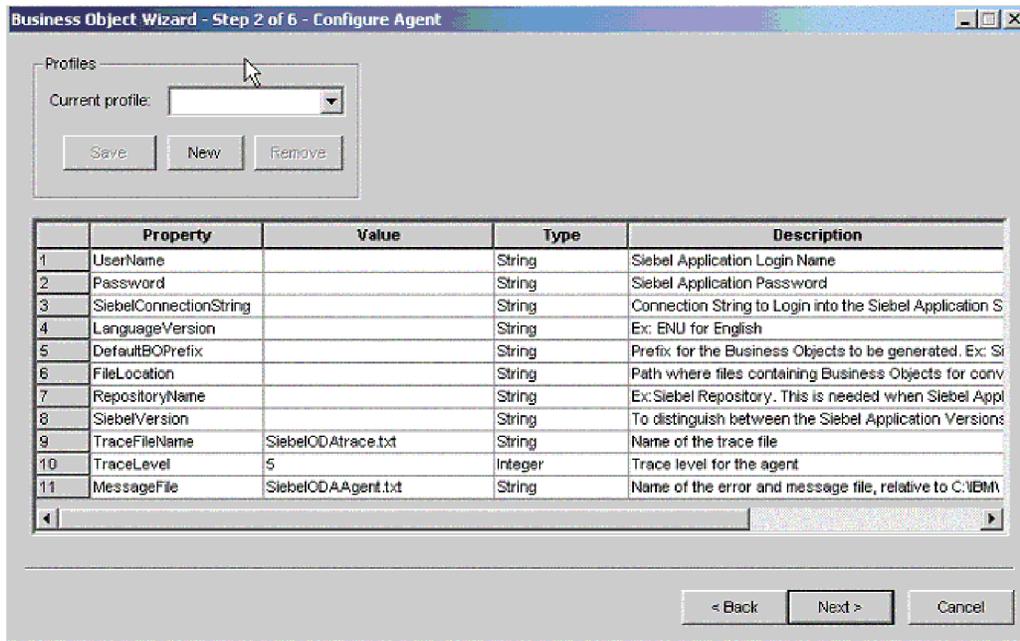


図 3. ビジネス・オブジェクト・ウィザードの「エージェントの構成」画面

表 8 に示された SiebelODA プロパティを構成します。

表 8. SiebelODA 構成プロパティ

行番号	プロパティ名	プロパティ・	
		タイプ	説明
1	UserName	String	Siebel アプリケーションのログイン名
2	Password	String	Siebel アプリケーションのパスワード

表 8. SiebelODA 構成プロパティ (続き)

行番号	プロパティ名	プロパティ・ タイプ	説明
3	SiebelConnection	String	<p>Siebel アプリケーションにログインするための接続ストリング。</p> <p>例:</p> <p>Siebel 7.x の場合: //machinename/enterprisename/objectmanager /servername</p> <p>Siebel 7.5 の場合: //machinename/enterprisename/objectmanager /servername</p> <p>Siebel 7.7 の場合: siebel://machinename:portno/enterprisename /objectmanager</p>
4	Language version	String	言語のバージョン。例えば、英語の場合は ENU を使用しません。
5	DefaultBOPrefix	String	Siebel 文書の各ビジネス・オブジェクト定義の名前に ODA が適用するプレフィックス。ビジネス・オブジェクト・プレフィックスを指定しない場合、ODA は、ビジネス・オブジェクト定義の名前の前にストリングを一切付加しません。
6	FileLocation	String	以前のバージョンのビジネス・オブジェクト定義を持つファイルを含む絶対パス。例えば、Windows の場合のパスは C:\SiebelBos です。
7	RepositoryName	String	Siebel アプリケーションの Siebel リポジトリの名前。
8	SiebelVersion	String	Siebel アプリケーションのバージョンを示します。Siebel バージョン 6.x の場合、このプロパティは 6.x に設定する必要があります。Siebel バージョン 7.x の場合は、設定しないでください。
9	TraceFileName	String	<p>SiebelODA がトレース情報を書き込むファイルの絶対パス名。ファイルが存在しない場合、SiebelODA はファイルを指定のディレクトリに作成します。ファイルがすでに存在している場合、SiebelODA はトレース情報をファイルの後に追加します。</p> <p>デフォルトでは、SiebelODA は、製品ディレクトリーの ODA\Siebel サブディレクトリーに SiebelODATrace.txt というトレース・ファイルを作成します。</p> <p>このプロパティを使用して、トレース・ファイルに別の名前を指定します。</p>
10	TraceLevel	Integer	SiebelODA に対して有効なトレースのレベルです。有効な値は 0 から 5 です。プロパティのデフォルト値は 5 (完全なトレースが使用可能) です。詳細については、60 ページの『エラーおよびトレース・メッセージ・ファイルの処理』を参照してください。

表 8. SiebelODA 構成プロパティ (続き)

行番号	プロパティ名	プロパティ・ タイプ	説明
11	MessageFile	String	<p>エラーおよびメッセージ・ファイルの絶対パス名。デフォルトでは、SiebelODA は SiebelODAAgent.txt というメッセージおよびエラー・ファイルを作成します。</p> <p>重要: エラーおよびメッセージ・ファイルは、製品ディレクトリーの ODA¥messages サブディレクトリーになければなりません。</p> <p>このプロパティを使用して、既存のファイルの確認や指定をします。</p>

重要: Business Object Designer Express で表示されているデフォルト値が存在しないファイルを指している場合には、メッセージ・ファイルの名前を訂正します。このダイアログ・ボックスから移動したときに、名前が不正確であった場合に、Business Object Designer Express は、ODA の起動元となったウィンドウにエラー・メッセージを表示します。このメッセージは、Business Object Designer Express ではポップアップしません。有効なメッセージ・ファイルの指定に失敗すると、ODA はメッセージなしに稼働します。

名前の付いたプロファイルにこれらのプロパティを保存すれば、SiebelODA を使用するたびにそれらのプロパティを再入力せずに済みます。ODA プロファイルの指定については、「ビジネス・オブジェクト開発ガイド」を参照してください。

ソースの選択

SiebelODA 用のすべての初期化プロパティの構成が終わると、「ソースの選択」画面が表示されます (66 ページの図 4 を参照)。

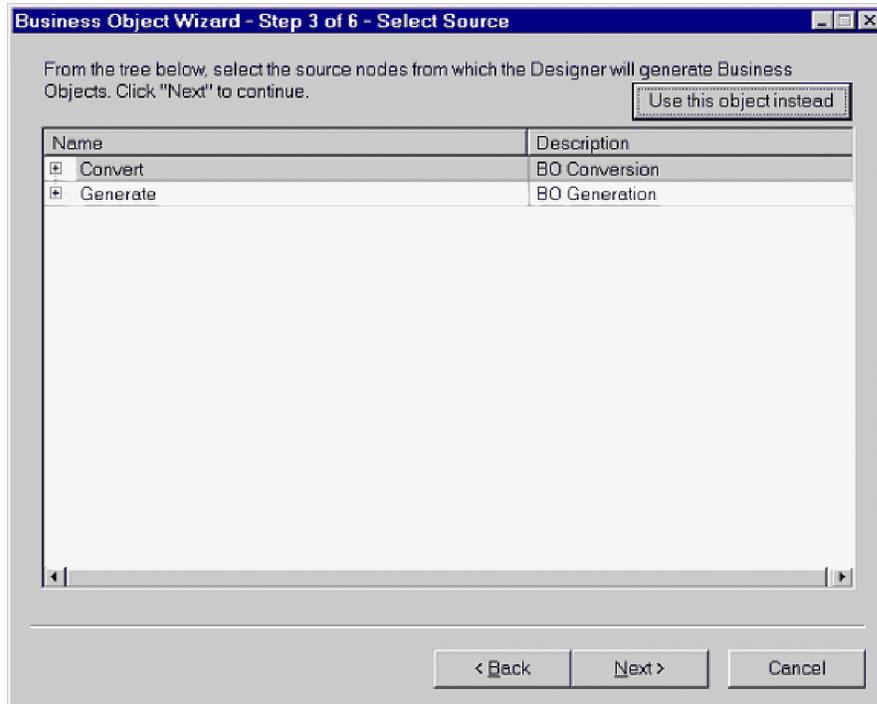


図4. ビジネス・オブジェクト・ウィザードの「ソースの選択」画面

この画面には、「変換」および「生成」の2つの展開可能オプションがあります。古いビジネス・オブジェクトを新しいビジネス・オブジェクトに変換する必要がある場合は、「変換」を展開します。これにより、変換の必要なリポジトリ・ファイルが表示されます(67ページの図5を参照)。

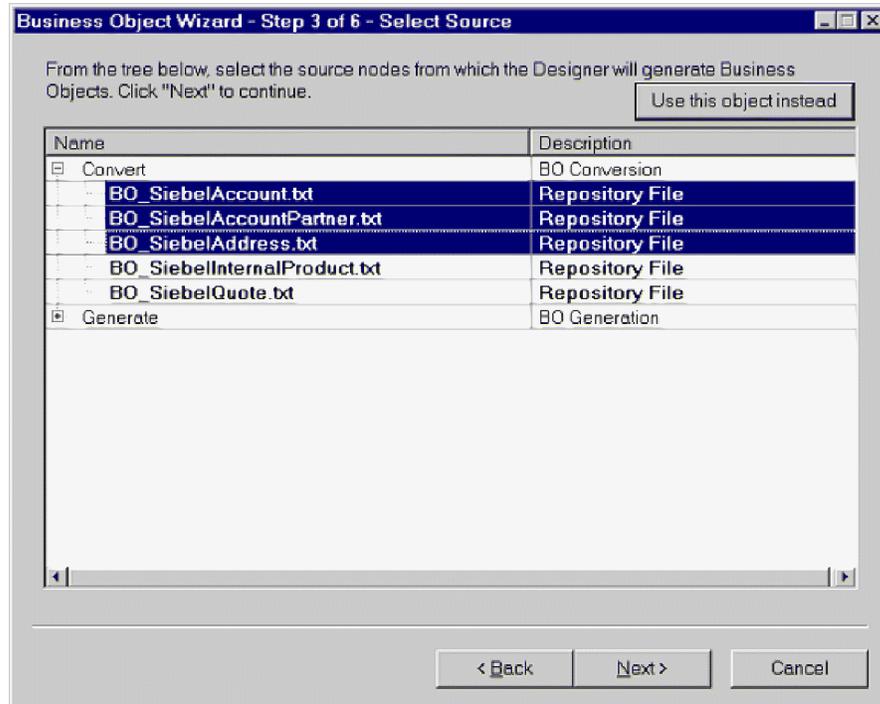


図5. ビジネス・オブジェクト・ウィザードで変換対象のビジネス・オブジェクトを表示している画面

新しいビジネス・オブジェクトを生成する必要がある場合は、「生成」を展開します。ここでは、「ビジネス・オブジェクト (Business Objects)」、「統合オブジェクト (Integration Objects)」、および「アプリケーション・サービス (Application Services)」という 3 つの展開可能なオプションがあります。これらのオプションの展開例については、68 ページの図 6、68 ページの図 7、および 69 ページの図 8 を参照してください。ビジネス・オブジェクトを展開すると、そのビジネス・オブジェクトのビジネス・コンポーネントを選択できます。同様に、統合オブジェクトを展開すると、そのオブジェクトの統合コンポーネントを選択できます。一方、アプリケーション・サービスを展開した場合、対応する統合オブジェクトはすでに選択状態です。

注: 統合コンポーネントが「アプリケーション・サービス (Application Services)」オプションおよび「統合オブジェクト (Integration Object)」オプションの両方にリストされた場合、この統合コンポーネントは「アプリケーション・サービス (Application Services)」を介してのみ生成できます。

注: 統合オブジェクトを生成すると、このオブジェクトに関してリストされたすべてのコンポーネントが生成されます。

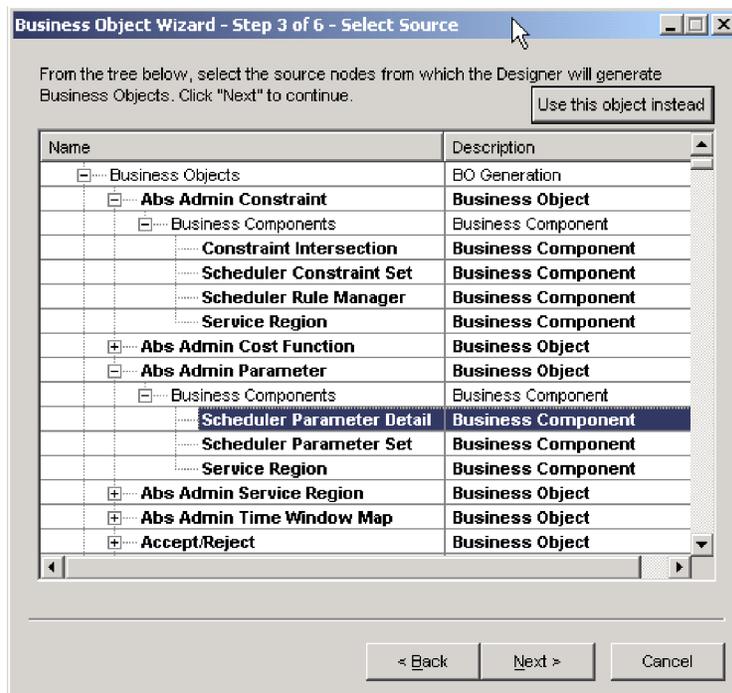


図6. ビジネス・オブジェクト・ウィザードで「ビジネス・オブジェクト (Business Objects)」を展開した状態

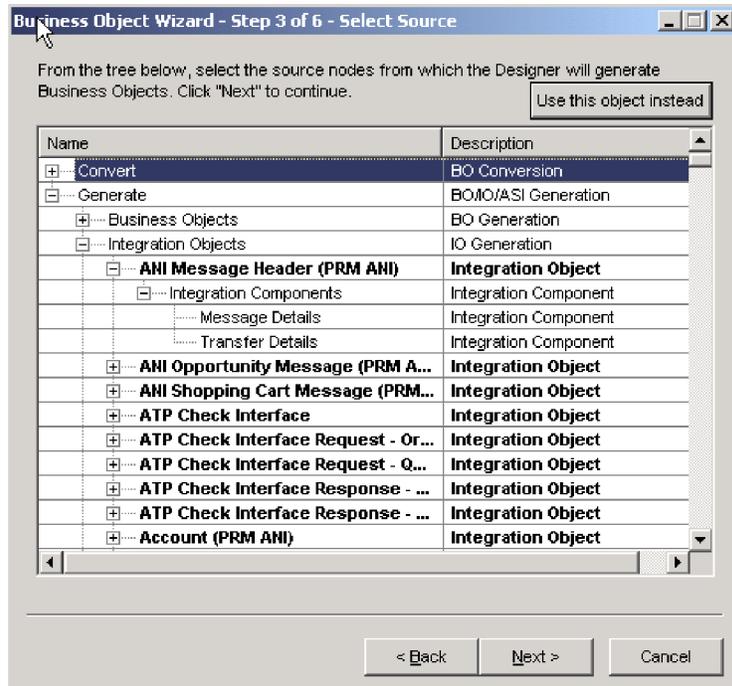


図7. ビジネス・オブジェクト・ウィザードで「統合オブジェクト (Integration Objects)」を展開した状態

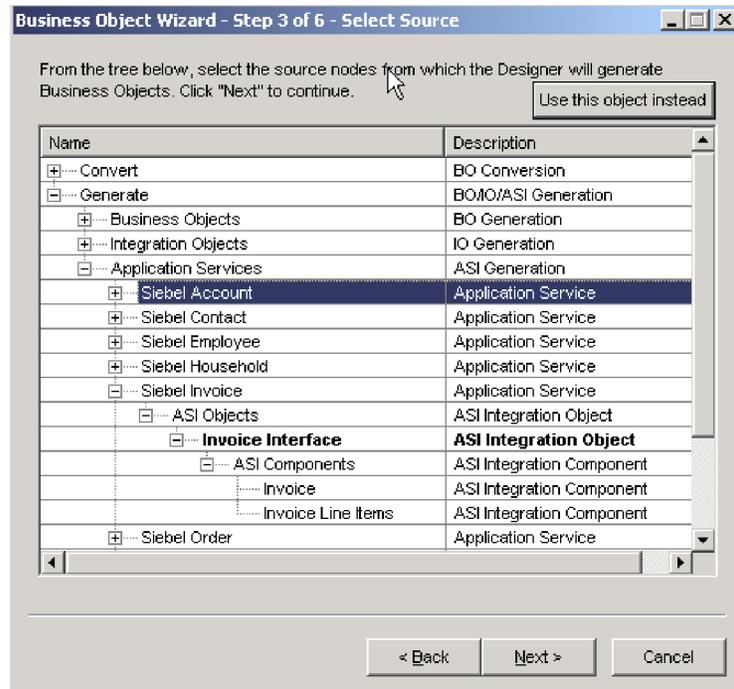


図8. ビジネス・オブジェクト・ウィザードで「アプリケーション・サービス (Application Services)」を展開した状態

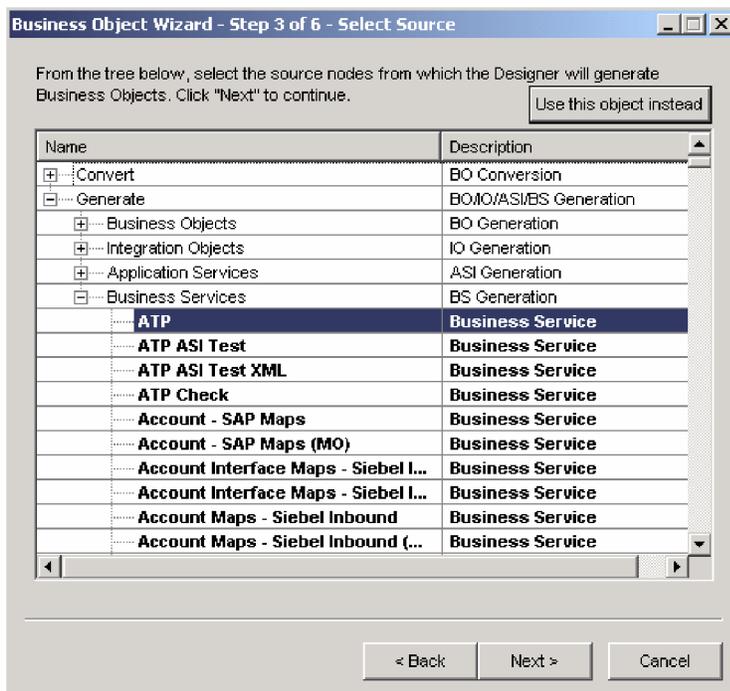


図9. ビジネス・オブジェクト・ウィザードで「ビジネス・サービス (Business Services)」を展開した状態

オブジェクトの選択の確認

生成するビジネス・オブジェクト定義に関連付ける Siebel 要素をすべて特定すると、Business Object Designer Express のダイアログ・ボックスには、選択したオブジェクトおよびコンポーネントのみが表示されます。図 10 は、このダイアログ・ボックスを図示したものです。

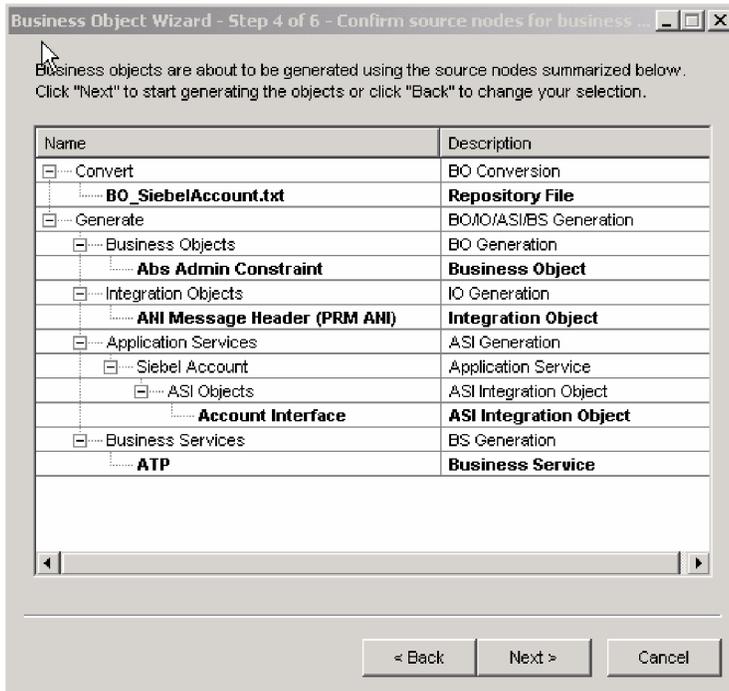


図 10. ビジネス・オブジェクト・ウィザードでのオブジェクトおよびコンポーネントの選択の確認

このウィンドウには、以下のオプションが表示されます。

- 選択を確定するには、「次へ」をクリックします。
- 選択が正しくない場合は、「戻る」をクリックして直前のウィンドウに戻り、必要な変更を行います。選択が正しい場合は、「次へ」をクリックします。

ビジネス・オブジェクト定義の生成

Siebel 要素を確認した後、次のダイアログ・ボックスで Business Object Designer Express がビジネス・オブジェクト定義を生成していることが通知されます。多くの数のコンポーネント・インターフェースが選択されると、この生成ステップは時間がかかることがあります。

71 ページの図 11 は、このダイアログ・ボックスを図示したものです。

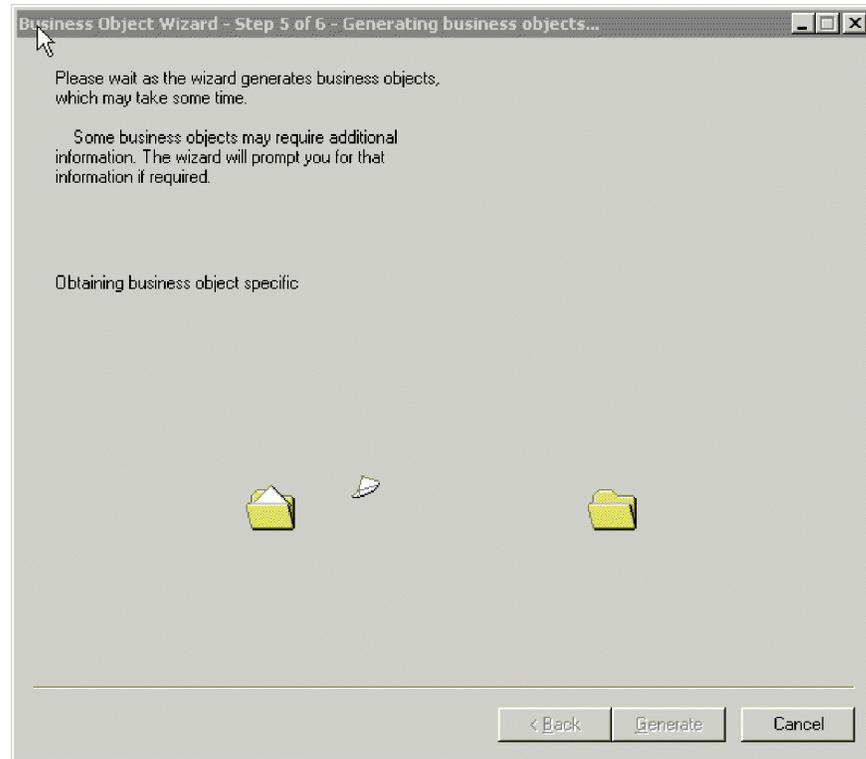


図 11. ビジネス・オブジェクト定義の生成

追加情報の入力

SiebelODA では動詞に関する追加情報が必要になるため、Business Object Designer Express は、ユーザーが選択したそれぞれの生成タイプ (ビジネス・オブジェクト、統合オブジェクト、およびアプリケーション・サービス) について「BO プロパティ」ウィンドウを表示し、情報の入力を求めるプロンプトを出します。72 ページの図 12 は、これらの画面を示しています。

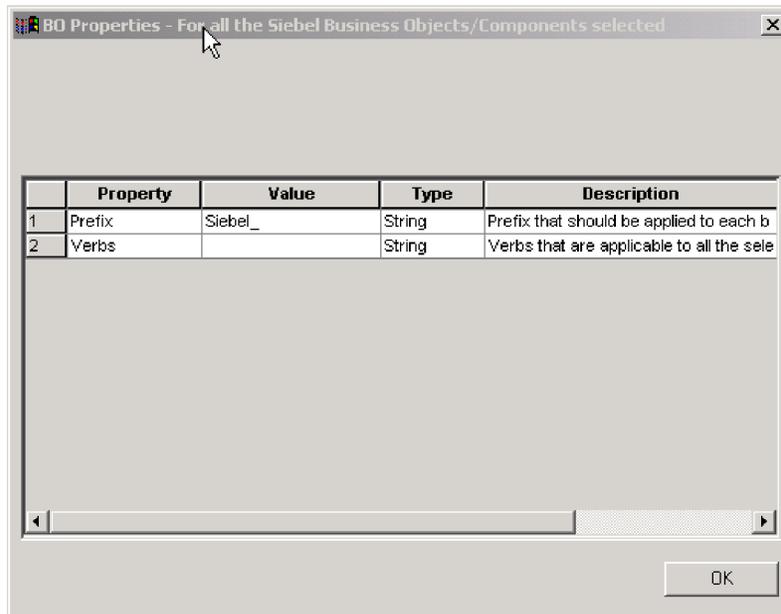


図 12. ビジネス・オブジェクトの追加情報の入力

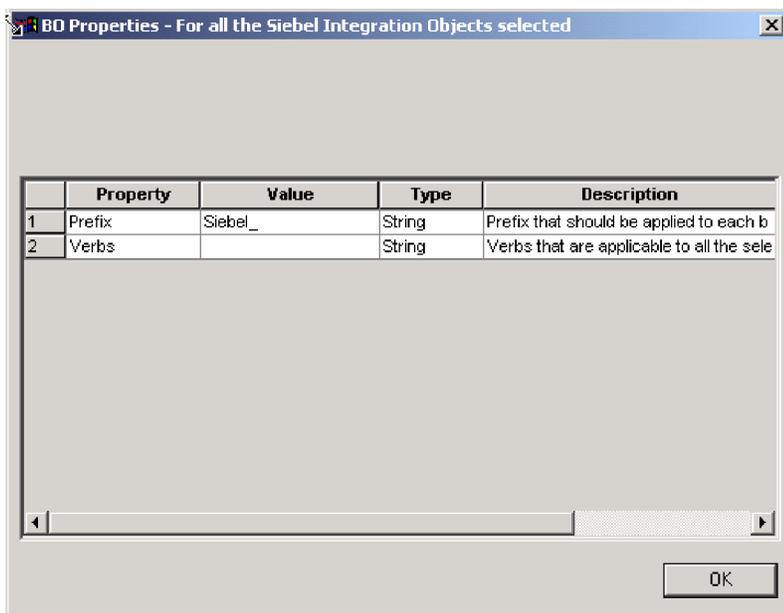


図 13. 統合ビジネス・オブジェクトの追加情報の入力

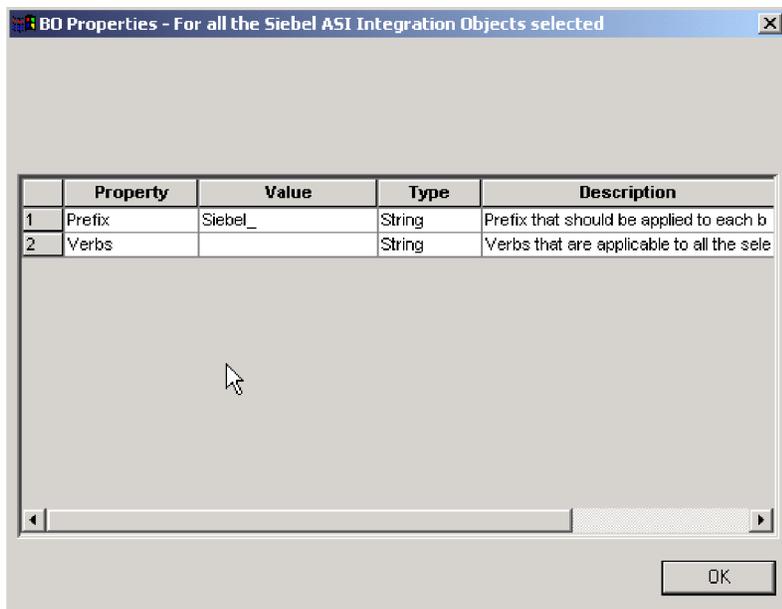


図 14. アプリケーション・サービスの追加情報の入力

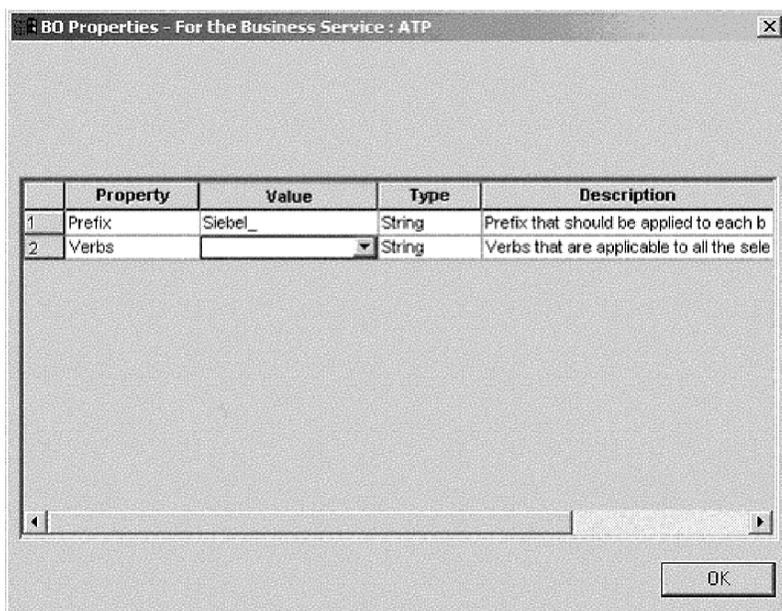


図 15. カスタム仕様で作成されたビジネス・サービスの追加情報の入力

「BO プロパティ」ウィンドウで、動詞の情報を入力または変更します。「値」フィールドをクリックし、ポップアップ・メニューから 1 つ以上の動詞を選択します。これらは、ビジネス・オブジェクトでサポートされる動詞です。

注: 「BO プロパティ」ダイアログ・ボックス内のフィールドに複数の値がある場合、このダイアログ・ボックスが最初に表示されたときに、このフィールドは空に見えます。フィールドをクリックして、値のドロップダウン・リストを表示します。

ビジネス・オブジェクト定義の保管

「BO プロパティ」ダイアログ・ボックスですべての必要な情報を指定し、「OK」をクリックすると、Business Object Designer Express はウィザードに最終のダイアログ・ボックスを表示します。このダイアログ・ボックスでは、以下の任意のアクションを実行できます。

- サーバーにビジネス・オブジェクト定義を保管します。
- ファイルにビジネス・オブジェクト定義を保管する (すべての統合ブローカーで可能)。
- ビジネス・オブジェクト定義を開いて Business Object Designer Express で編集します。

詳細について、およびさらに変更を行う場合は、「ビジネス・オブジェクト開発ガイド」を参照してください。

図 16 は、このダイアログ・ボックスを図示したものです。

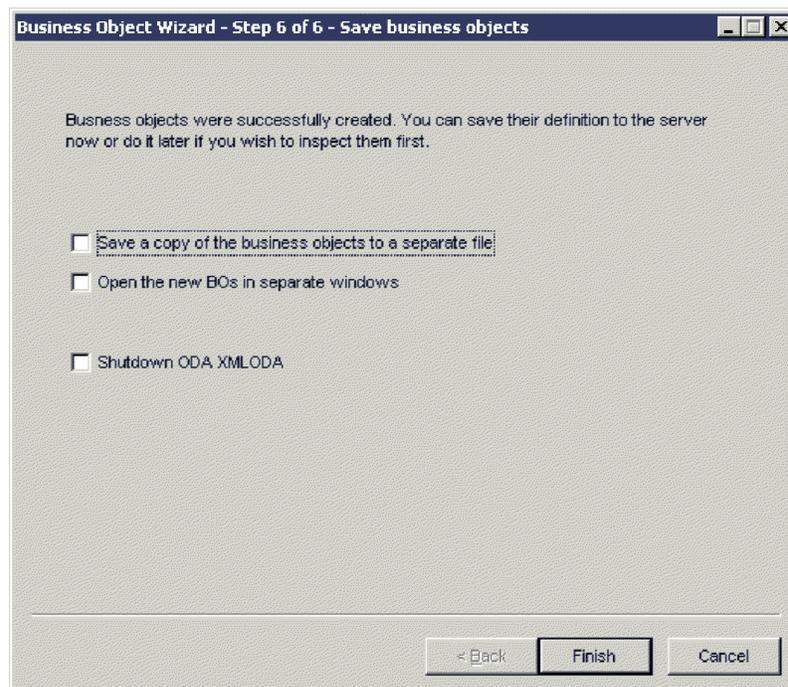


図 16. ビジネス・オブジェクトの定義の保管

生成された定義の検討

SiebelODA が生成するビジネス・オブジェクト定義には、以下の内容が含まれています。

- An attribute for each column in the specified Siebel objects
- 「BO プロパティ」ウィンドウで指定された動詞
- アプリケーション固有の情報:
 - ビジネス・オブジェクト・レベル
 - 属性ごと

- 動詞ごと

注: SiebelODA が Siebel ビジネス・オブジェクトまたはコンポーネントのビジネス・オブジェクトを生成する場合、アプリケーション固有の情報はコンテナ属性に対して正常に生成される場合とされない場合があります。生成されたビジネス・オブジェクトをチェックし、正しいアプリケーション固有情報が含まれているかどうかを確認して、必要な場合は情報を訂正する必要があります。

SiebelODA が Siebel 統合オブジェクト、コンポーネント、またはビジネス・サービスのビジネス・オブジェクトを生成する場合は、生成されるビジネス・オブジェクトには、コンテナ属性を含むすべての属性のアプリケーション固有情報が含まれます。

このセクションでは、以下のことについて説明します。

- 75 ページの『ビジネス・オブジェクト・レベルのプロパティー』
- 75 ページの『属性プロパティー』
- 77 ページの『動詞』

ビジネス・オブジェクト・レベルのプロパティー

SiebelODA は、ビジネス・オブジェクト・レベルで以下の情報を生成します。

- ビジネス・オブジェクトの名前
- バージョン - デフォルトでは 1.0.0 に設定
- アプリケーション固有の情報

ビジネス・オブジェクト・レベルのアプリケーション固有の情報には、対応する Siebel ビジネス・オブジェクトまたはビジネス・コンポーネントの名前が含まれています。

属性プロパティー

ここでは、SiebelODA が属性ごとに生成するプロパティーについて説明します。

重要: 以降のセクションで説明するユーザー・エディットとは、ビジネス・オブジェクトの変換でなく、ビジネス・オブジェクトの生成だけを指します。

Name プロパティー

SiebelODA は、Siebel ビジネス・コンポーネント内の対応する属性から属性の名前の値を取得します。

Data Type プロパティー

属性の型を設定する際、表 9 に示すように、SiebelODA は Siebel ビジネス・コンポーネント内の属性のデータ型を対応するデータ型に変換します。これはビジネス・オブジェクトの生成の場合に限られます。その理由は、ビジネス・オブジェクトの変換が既存のビジネス・オブジェクトのためのものであるためです。

表 9. データ型の対応関係

アプリケーション	WebSphere Business Integration システム	長さ
----------	-------------------------------------	----

表9. データ型の対応関係 (続き)

DTYPE_BOOL	BOOLEAN	
DTYPE_ID、DTYPE_PHONE	STRING	Siebel アプリケーション・サーバー内の対応する属性の長さ
DTYPE_TEXT DTYPE_NOTE		
DTYPE_DATE DTYPE_TIME DTYPE_DATETIME DTYPE_UTCDATETIME	DATE	
DTYPE_INTEGER DTYPE_NUMBER	INTEGER	
DTYPE_CURRENCY	DOUBLE	

注: 属性のデータ型が表9に示されているものでない場合、SiebelODAは列をスキップし、列を処理できないことを示すメッセージを表示します。

Cardinality プロパティ

SiebelODAは、すべての単純属性のカーディナリティを1に設定し、コンテナ属性をnに設定します。ユーザーは、必要に応じて、コンテナ属性のカーディナリティを変更する必要があります。例えば、コンテナ属性がピック・リストであることが判明した場合、ユーザーはカーディナリティを1に設定する必要があります。

MaxLength プロパティ

SiebelODAは、Siebel アプリケーション・サーバーから属性の長さを取得します。

IsKey プロパティ

列がテーブルまたはビューの1次キーである場合、SiebelODAはその列をキー属性としてマークします。ビジネス・オブジェクトの生成の場合は、ID属性が、デフォルトでkeyとしてマークされる唯一の属性です。

IsRequired プロパティ

フィールドがテーブルまたはビューで非ヌルに指定されている場合、SiebelODAはそのフィールドを必須属性としてマークします。ただし、SiebelODAはキー・フィールドを必須属性としてマークしません。これは、Siebel アプリケーションがレコード作成時に独自のID値を生成するためです。

AppSpecificInfo プロパティ

ユーザーは、コンテナ属性が生成されていない場合にはこのプロパティを編集する必要があり、また、コンテナ属性が生成されている場合にはその正確性を確認する必要があります。

PollQuantity

コネクタがポーリング間隔ごとに検索するデータベース表の行数。指定可能な値は1から500です。

デフォルトは、1です。

動詞

SiebelODA は、「BO プロパティ」ウィンドウで指定された動詞を生成します。各動詞の AppSpecificInfo プロパティを作成しますが、その取り込みは行いません。

ビジネス・オブジェクト定義への情報の追加

Siebel ビジネス・オブジェクトとビジネス・コンポーネントが、ビジネス・オブジェクトが要求するすべての情報を有しているとは限らないため、特に新しいビジネス・オブジェクトを生成する際には、SiebelODA が作成するビジネス・オブジェクト定義への情報の追加が必要となる場合があります。

ビジネス・オブジェクト定義を調べたり、変更された定義をリポジトリに再ロードしたりするには、Business Object Designer Express を使用します。あるいは、InterChange Server Express が統合ブローカーであるため、repos_copy コマンドを使用して定義をリポジトリにロードすることができます。

注: Siebel Application の計算フィールドは列に対応していないため、ODA はこれらのフィールドを生成しません。これらのフィールドは、ビジネス・オブジェクト定義に手動で追加できます。

第 6 章 Siebel ビジネス・サービスでのアダプターの使用

- 『ビジネス・サービスの理解』
- 82 ページの『ビジネス・サービスにおける動詞処理』
- 83 ページの『ビジネス・サービスにおけるイベント検出』

注: 本章では、Event ビジネス・コンポーネントと Archive ビジネス・コンポーネント、Event ビジネス・オブジェクトと Archive ビジネス・オブジェクト、およびイベント表とアーカイブ表を参照する場合があります。これらの参照は、以前のバージョンで見られる CW Event と CW Archive への参照、および Siebel 7.5 で見られる IBM Event と IBM Archive への参照と同義です。

ビジネス・サービスの理解

この節では、ビジネス・サービスとは何かについて説明し、ビジネス・サービスをサポートするビジネス・オブジェクトの作成方法を示します。以下のトピックで構成されています。

『ビジネス・サービスの説明』

80 ページの『ビジネス・サービスをサポートするビジネス・オブジェクトの処理』

ビジネス・サービスの説明

Siebel ビジネス・サービスは、Siebel アプリケーションと外部アプリケーションとの間でのデータの移動やデータ・フォーマットの変換といった一連の機能の使用をカプセル化して単純化する、Siebel 内のエンティティです。Siebel ビジネス・コンポーネントおよびビジネス・オブジェクトは、通常は Siebel データ・モデル内の特定のデータやテーブルに関連付けられているオブジェクトです。これに対し Siebel ビジネス・サービスは、特定のオブジェクトに関連付けられているのではなく、特定の目標を達成するためにオブジェクトに対して操作を実行します。

アダプターは、EAI Siebel Adapter (Siebel が提供する汎用ビジネス・サービス)、Siebel が定義する Application Service Interface (ASI)、およびカスタム仕様で作成されたビジネス・サービスをサポートします。

EAI Siebel Adapter および ASI は、IBM ビジネス・オブジェクトに関しては、同じように処理されます。これらは、IBM ビジネス・オブジェクト処理用の動詞として使用される、類似したメソッドを実装します。EAI Siebel Adapter は、Siebel ビジネス・オブジェクトに基づく統合オブジェクトであればすべて受け入れることができます。そのため、IBM WebSphere Business Integration Adapter for Siebel は、統合オブジェクトを IBM ビジネス・オブジェクトによって表すことで、EAI Siebel Adapter をサポートします。同様に、アダプターは、Siebel ASI を実装する統合オブジェクトを IBM ビジネス・オブジェクトによって表すことで、Siebel ASI をサポートします。

カスタム仕様で作成されたビジネス・オブジェクトは、異なる方法で処理されます。これらは任意のメソッドを実装できるため、IBM ビジネス・オブジェクトは、

統合オブジェクトではなく、サービス自体を表します。詳細については、83 ページの『カスタム・ビジネス・サービスのサポート』を参照してください。

EAI Siebel Adapter および ASI を、カスタム仕様で作成されたビジネス・サービスとしてアダプターによって処理することができ、IBM ビジネス・オブジェクトを、直接これらのサービスを表すように作成することができます。ただし、この方法は推奨されません。53 ページの『第 5 章 ビジネス・オブジェクトの作成』を参照してください。

注: アダプターは、Siebel に対してビジネス・オブジェクトを示す IBM オブジェクトと、アプリケーション固有の情報「BSN=」によって Siebel 統合オブジェクトを表す IBM ビジネス・オブジェクトを区別します。81 ページの表 10 を参照してください。

ビジネス・サービスをサポートするビジネス・オブジェクトの処理

アダプターは、統合オブジェクトの表現である着信ビジネス・オブジェクトのプロパティ・セットを作成します。以下の例では、アダプターが、統合オブジェクトを表す IBM ビジネス・オブジェクトからプロパティ・セットを作成する方法について説明します。

例:

- タイプ Siebel Message の新規プロパティ・セットのインスタンスを生成する
- プロパティ・セット・タイプが SiebelMessage およびプロパティ (IntObject Format = Siebel Hierarchy、MessageType = Integration Object、MessageId = "" など) として設定される
- IntObjectName がビジネス・オブジェクトのアプリケーション固有の情報から取得される
- 親コンポーネントの新規プロパティ・セットのインスタンスが作成される
- プロパティ・セットのタイプが ListOf<Parent component name> として設定される
- <Parent component name> はアプリケーション固有の情報から取得することができる
- タイプ <Parent Component> の新規プロパティ・セットのインスタンスを生成する
- タイプを <parent object type> として設定する
- 別のプロパティを設定する
- 子コンポーネントについて同じ作業を行う

例: Siebel 統合オブジェクト

```
Account (PRM ANI) (統合オブジェクト)
    +Account (統合コンポーネント)
    +Business address (統合コンポーネント)
```

例: Siebel 統合オブジェクトを表す IBM ビジネス・オブジェクト

```
Siebel <IntObjectName> (ParentIntegrationComponent)
    Attribute1 FN=<fieldname>
    Attribute2 FN=<fieldname>
```

```

Attribute3 FN=<fieldname>
+ChildIntegrationComponent
  childAttribute1 FN=<fieldname>
  childAttribute2 FN=<fieldname>

```

Parent Integration コンポーネントのオブジェクト・レベル ASI は、
BSN=<name>;IO=<Name>;IC=<Name> となります。

Child Integration コンポーネントでは、IO=<Name>;IC=<Name> となります。

以下の表には、統合オブジェクトを作成する際のビジネス・オブジェクト・レベル
のアプリケーション・テキストおよび単純属性レベルのアプリケーション・テキ
ストを示します。

表 10. ビジネス・オブジェクト・レベルのアプリケーション・テキスト

パラメーター	説明
IO=	このビジネス・オブジェクトに対応する Siebel 統合オブジェクトの名前
IC=	このビジネス・オブジェクトに対応する Siebel 統合コンポーネントの名前
BSN=	このビジネス・オブジェクトで使用するビジネス・サービスの名前。アプリケーション固有情報 (Siebel Account または Siebel Contact など) を使用する場合は、特定のビジネス・サービスが存在していなければなりません。他の統合オブジェクトを使用する場合は、Siebel Enterprise Applications Integration (EAI) が存在していなければなりません。
SiebASI=	(使用するべきでない) ビジネス・オブジェクトが ASI 統合オブジェクトを表す場合、SiebASI=true になります。アダプターおよび ODA の現行バージョンは、BSTYPE アプリケーション固有情報タグを代わりに使用します。
BSTYPE=	ビジネス・サービスのタイプを決定します。 <ul style="list-style-type: none"> • EAI Siebel Adapter で、IBM ビジネス・オブジェクトが Siebel 統合オブジェクトを表す場合、アプリケーション固有の情報内に、BSTYPE=GENERIC と記述する必要があります。 • Siebel ASI で、IBM ビジネス・オブジェクトが Siebel 統合オブジェクトを表す場合、アプリケーション固有の情報内に、BSTYPE=ASI と記述する必要があります。 • カスタム仕様で作成されたビジネス・サービスで、IBM ビジネス・オブジェクトがサービスを表す場合、アプリケーション固有の情報内に、BSTYPE=CUSTOM と記述する必要があります。

表 11. 単純属性オブジェクト・レベルのアプリケーション・テキスト

パラメーター	説明
FN=	この属性に対応する Siebel 統合コンポーネントのフィールドのフィールド名

ビジネス・サービスにおける動詞処理

ビジネス・サービスでは、次の動詞がサポートされます。

注: 表 12 のすべての動詞において、戻りコードは VALCHANGE です。

表 12. ビジネス・サービスでサポートされる動詞

動詞	説明
Delete	Siebel オブジェクトの削除には、親オブジェクトのキーを使用します。アダプターはすべての基本キーが存在することを検査します。
Insert	Insert 動詞には、完全な着信ビジネス・オブジェクトが使用されます。
InsertOrUpdate (Upsert)	入力オブジェクトと同じキーを持つオブジェクトが存在する場合、指定された入力オブジェクトを既存のオブジェクトにマージします。そうでない場合は、入力オブジェクトに基づいて新規オブジェクトを Siebel 内に作成します。 アダプターは、オブジェクトを処理する前に、すべての基本キーの存在検査を行います。
QueryByExample (EAI Siebel Adapter の場合は Query)	提供されたサンプル・オブジェクトに基づいてオブジェクトを照会します。この操作は、Retrieve by content 操作として扱われます。
QueryById	キーを持つオブジェクトが存在する場合、そのオブジェクトを照会または検索します。これらの操作は、Retrieve 操作として扱うことができます。
Update	入力オブジェクトと同じキーを持つオブジェクトが存在する場合、指定された入力オブジェクトを既存のオブジェクトにマージします。そうでない場合は、エラーになります。
Synchronize	入力オブジェクトと同じキーを持つオブジェクトが存在する場合、既存のオブジェクトを入力オブジェクトと同じ状態にします。そうでない場合は、入力オブジェクトに基づいて新規オブジェクトを Siebel 内に作成します。

以下の例は、表 12 の動詞を使用するためのプロセス・フローを示しています。

IBM ビジネス・オブジェクトの動詞は、ビジネス・サービスのメソッドを表します。

動詞処理:

- ビジネス・サービス名を取得する
- 入力に基づいてプロパティ・セットを作成する
- 指定されたビジネス・サービスに対して動詞を呼び出し、入力プロパティ・セットを渡す
- 出力プロパティ・セットからビジネス・オブジェクトを作成する

ビジネス・サービスにおけるイベント検出

ビジネス・サービスを使用する場合、ビジネス・オブジェクトの名前と動詞が変わることを除き、トリガーのためのスクリプトに変更はありません。トリガーは、統合オブジェクトのベースとなるビジネス・オブジェクトに基づいて作成する必要があります。イベントの作成時に、統合オブジェクトに新しい動詞および対応するビジネス・オブジェクトがトリガーによって読み込まれる必要があります。

アダプターはデフォルトの `getBO()` メソッドをオーバーライドするので、`doVerbFor` メソッドを呼び出す前に動詞 `RetrieveByContent` を設定する必要があります。このシナリオでは、ビジネス・オブジェクトが統合オブジェクトの場合は、動詞 `QueryByExample` を設定します。一方、ビジネス・オブジェクトがアプリケーション固有インターフェースの場合は、動詞 `Query` を設定します。`QueryByExample` (汎用ビジネス・サービス EAI Siebel Adapter の `RetrieveByContent` に相当) に対応するメソッドは `Query` です。

ビジネス・サービスを用いたイベント検出は、EAI Siebel Adapter および Application Services Interfaces のみをサポートします。統合サービスを 1 つのみ処理することができます。

カスタム・ビジネス・サービスのサポート

Siebel カスタム・ビジネス・サービスに対応する IBM ビジネス・オブジェクトを処理する場合、アダプターはアプリケーション固有の情報があるかどうかをチェックします。BSTYPE=Custom が検出された場合、トップレベル・ビジネス・オブジェクトの下に Request ビジネス・オブジェクトを取得します。アダプターが、着信ビジネス・サービスに対応するプロパティ・セットを構築するとき、Request ビジネス・オブジェクトの単純属性がプロパティを規定します。SiebelMessage コンテナ属性が存在する場合は、その属性が統合オブジェクトを規定します。アダプターは、SiebelMessage コンテナ属性が存在する場合も、しない場合も、ビジネス・オブジェクトを処理することができます。このアダプターでは、以下の処理が実行されます。

1. 新規プロパティ・セットのインスタンスを生成する
2. 単純属性値を新規プロパティ・セットのプロパティとして設定する
3. Siebel Message オブジェクトを子プロパティ・セットと見なす (存在する場合)

ビジネス・サービス・メソッドを実行すると、`OutputPropertySet` が取得され、`Response` ビジネス・オブジェクトに取り込まれます。

ビジネス・サービスに対応する IBM ビジネス・オブジェクト

```
Siebel_BS<Name>
  +Request
  +Response
```

ビジネス・サービスで提供されるさまざまなメソッドは、ビジネス・オブジェクト処理用の動詞として機能します。

ビジネス・オブジェクトは、それぞれのメソッド/操作によって判別され、入出力は異なります。

統合オブジェクトに対応する IBM ビジネス・オブジェクト
Siebel_<IntObjectName> (Parent IntegrationComponent)
Field1
Field2
+ChildIntegrationComponent

第 7 章 コネクタの実行

- 『コネクタの始動』
- 86 ページの『コネクタの停止』

コネクタの始動

コネクタは、**コネクタ始動スクリプト**を使用して明示的に始動する必要があります。Windows システムでは、始動スクリプトは、次に示すようなコネクタのランタイム・ディレクトリに存在していなければなりません。

```
ProductDir%\connectors%\connName
```

ここで、*connName* はコネクタを示します。

始動スクリプトの名前は、表 13 に示すように、オペレーティング・システム・プラットフォームによって異なります。

表 13. コネクタの始動スクリプト

オペレーティング・システム	始動スクリプト
Windows	start_ <i>connName</i> .bat

始動スクリプトが実行されると、始動スクリプトはデフォルトでは *Productdir* に構成ファイルがあるものと考えます (下記のコマンドを参照)。ここに構成ファイルを配置します。

注: アダプターが JMS トランSPORTを使用している場合、ローカル構成ファイルが必要です。

- **Windows システムでのコネクタの始動:**
 - 「スタート」メニューから、「プログラム」>「IBM WebSphere Business Integration Server Express」>「アダプター」>「コネクタ」を選択します。デフォルトでは、プログラム名は「IBM WebSphere Business Integration Server Express」となっています。ただし、これはカスタマイズすることができます。あるいは、ご使用のコネクタへのデスクトップ・ショートカットを作成することもできます。
 - Windows コマンド行から start_*connName* *connName* *brokerName* {-cconfigFile} を実行します。
 - Windows システムでは、Windows サービスとして始動するようにコネクタを構成することができます。この場合、Windows システムがブートしたとき (自動サービスの場合)、または Windows サービス・ウィンドウを通じてサービスを始動したとき (手動サービスの場合) に、コネクタが始動します。

コマンド行の始動オプションなどのコネクタの始動方法の詳細については、「システム管理ガイド」を参照してください。

コネクタの停止

コネクタを停止する方法は、以下に示すように、コネクタが始動された方法によって異なります。

- コマンド行からコネクタを始動した場合は、コネクタ始動スクリプトを用いて、以下の操作を実行します。
 - Windows システムでは、始動スクリプトを起動すると、そのコネクタ用の別個の「コンソール」ウィンドウが作成されます。このウィンドウで、「Q」と入力して Enter キーを押すと、コネクタが停止します。
- System Manager 始動時に起動される から (WebSphere Business Integration Server Express Adapters 製品のみ)

このツールを使用して、コネクタのロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除を行うことができます。

- System Monitor から (InterChange Server Express 製品のみ)

このツールを使用して、コネクタのロード、アクティブ化、非アクティブ化、休止、シャットダウン、または削除を行うことができます。

- Windows システムでは、Windows サービスとして始動するようにコネクタを構成することができます。この場合、Windows システムのシャットダウン時に、コネクタは停止します。

第 8 章 トラブルシューティング

この章では、コネクターの実行時に出される可能性のあるエラー・メッセージと、これらのエラーの考えられる修正方法について説明します。

Siebel ログ・ファイルのサイズの削減

Siebel JAVABean では、ロギング・タイムアウト値を変更することができます。

Siebel JAVABean でロギング・タイムアウト値をリセットするには、次のようにします。

1. 「サイト・マップ (Site Map)」 > 「サーバー管理 (Server Administration)」 > 「コンポーネント (セールス・オブジェクト・マネージャー) (Components (Sales Object Manager))」 と選択します。
2. 下部アプレットの「コンポーネント・パラメーター (Component Parameter)」に移動して、タイムアウト値を入力します。

注: 「要求タイムアウト (Request Timeout)」の現行値は 600 に設定されています。これは、コネクターが 10 分で停止するということを示します。Siebel に基づいて、この値を必要に応じて任意の値に変更できます。

結果セットのサポートでのメモリー制限

結果セットのサポートが DB2[®] で実行されている場合、アダプターでは、アダプターに付属する J2SE JRE 1.4.1、バージョン SR2 について 2 GB の JVM メモリー制限があります。結果セット・プロセスが 2 GB のメモリーを使用できるようにするには、ご使用の環境で、過剰なページングを行わない効率的なメモリー利用を容易にするハードウェアを使用していなければなりません。

付録 A. コネクターの標準構成プロパティ

この付録では、WebSphere Business Integration Server Express アダプターのコネクタ・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。説明は、InterChange Server Express が対象となります。

このコネクタに固有のプロパティについては、本書の該当するセクションを参照してください。

新規プロパティ

以下の標準プロパティは、本リリースで追加されました。

- AdapterHelpName
- ControllerEventSequencing
- jms.ListenerConcurrency
- jms.TransportOptimized
- TivoliTransactionMonitorPerformance

標準コネクタ・プロパティの概要

コネクタには 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。フレームワークが使用します。
- アプリケーション固有またはコネクタ固有の構成プロパティ。エージェントが使用します。

これらのプロパティは、アダプターのフレームワークおよびエージェントの実行時の振る舞いを決定します。

このセクションでは、Connector Configurator Express の始動方法について説明し、すべてのプロパティに共通する特性について説明します。コネクタ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ズ・ガイドを参照してください。

Connector Configurator Express の始動

コネクタ・プロパティの構成は Connector Configurator Express から行います。Connector Configurator Express には、System Manager からアクセスします。Connector Configurator Express の使用法の詳細については、本書の Connector Configurator Express に関するセクションを参照してください。

構成プロパティ値の概要

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。

1. デフォルト
2. InterChange Server Express 統合ブローカー用のリポジトリ

3. ローカル構成ファイル

4. コマンド行

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。STRING プロパティ・タイプの長さには制限はありません。INTEGER タイプの長さは、アダプターを実行しているサーバーによって決まります。

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する場合は、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定されます。

プロパティの更新特性 (すなわちコネクタ・プロパティへの変更を有効にする方法とタイミング) は、プロパティの性質によって異なります。

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、新しい値が即時に有効になります。ただし、コネクタがスタンドアロン・モードの場合 (System Manager に依存しない) です。

• エージェント再始動 (InterChange Server Express のみ)

コネクタ・エージェントを停止して再始動しなければ、新規の値が有効になりません。

• コンポーネント再始動

System Manager でコネクタを停止してから再始動しなければ、新しい値が有効になりません。エージェントまたはサーバー・プロセスを停止して再始動する必要はありません。

• システム再始動

コネクタ・エージェントおよびサーバーを停止して再始動しなければ、新規の値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 Express」ウィンドウ内の「更新メソッド」列を参照するか、91 ページの表 14 の「更新メソッド」列を参照してください。

標準プロパティが存在できる場所が 3 箇所あります。一部のプロパティは複数の場所にあってもかまいません。

• ReposController

このプロパティはコネクタ・コントローラ内にあり、その場所でのみ有効です。エージェント・サイドで値を変更した場合、コントローラには影響しません。

• ReposAgent

このプロパティはエージェント内にあり、その場所でのみ有効です。プロパティによっては、ローカル構成によってこの値をオーバーライドされることがあります。

• LocalConfig

このプロパティは、コネクタの構成ファイル内にあり、構成ファイルを通じてのみ機能することができます。コントローラはこのプロパティの値を変更

することができず、システムが再配置されてコントローラーが明示的に更新されなければ、構成ファイルに加えられた変更を認識しません。

標準プロパティの早見表

表 14 は、標準コネクタ構成プロパティの早見表です。すべてのコネクタでこれらのプロパティすべてを必要とするわけではなく、プロパティ設定は異なる場合があります。

各プロパティの説明については、表の次のセクションを参照してください。

注: 表 14 の注の欄で、「RepositoryDirectory が <REMOTE> に設定され」という句は、ブローカーが InterChange Server Express であることを示します。

表 14. 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
AdapterHelpName	有効な <Regional Setting> ディレクトリーを含む <ProductDir>%bin%Data%App%Help 内の有効なサブディレクトリーのいずれか	テンプレート名 (有効な場合) またはブランク・フィールド	コンポーネント再始動	サポートされる地域設定。chs_chn、cht_twn、deu_deu、esn_esp、fra_fra、ita_ita、jpn_jpn、kor_kor、ptb_bra、および enu_usa (デフォルト) を含む。
AdminInQueue	有効な JMS キュー名	<CONNECTORNAME>/ADMININQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
AdminOutQueue	有効な JMS キュー名	<CONNECTORNAME>/ADMINOUTQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
AgentConnections	1 から 4	1	コンポーネント再始動	このプロパティは、DeliveryTransport の値が MQ または IDL で、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
AgentTraceLevel	0 から 5	0	ICS では動的、その他の場合はコンポーネント再始動	
ApplicationName	アプリケーション名	コネクタのアプリケーション名に指定された値	コンポーネント再始動	
BrokerType	ICS	ICS	コンポーネント再始動	

表 14. 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
CharacterEncoding	サポートされる任意のコード。 次のリストはその一部です。ascii7、ascii8、SJIS、Cp949、GBK、Big5、Cp297、Cp273、Cp280、Cp284、Cp037、Cp437	ascii7	コンポーネント再始動	このプロパティは、C++コネクタでのみ有効です。
CommonEventInfrastructure	true または false	false	コンポーネント再始動	
CommonEventInfrastructureURL	URL スtring。例えば、corbaloc:iiop:host:2809。	デフォルト値はありません。	コンポーネント再始動	このプロパティは、CommonEventInfrastructure の値が true の場合のみ有効です。
ConcurrentEventTriggeredFlows	1 から 32,767	1	コンポーネント再始動	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
ContainerManagedEvents	ブランクまたは JMS	ブランク	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
ControllerEventSequencing	true または false	true	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
ControllerStoreAndForwardMode	true または false	true	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
ControllerTraceLevel	0 から 5	0	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
DeliveryQueue	任意の有効な JMS キュー名	<CONNECTORNAME>/DELIVERYQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
DeliveryTransport	IDL または JMS	RepositoryDirectory の値が <REMOTE> の場合は IDL。それ以外の場合は JMS。	コンポーネント再始動	RepositoryDirectory の値が <REMOTE> ではない場合、このプロパティの有効な値は JMS のみです。
DuplicateEventElimination	true または false	false	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

表 14. 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
EnableOidForFlowMonitoring	true または false	false	コンポーネント再始動	このプロパティは、BrokerType の値が ICS の場合のみ有効です。
FaultQueue	任意の有効なキュー名	<CONNECTORNAME>/FAULTQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory、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	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.ListenerConcurrency	1 から 32767	1	コンポーネント再始動	このプロパティは、jms.TransportOptimized の値が true の場合のみ有効です。
jms.MessageBrokerName	jms.FactoryClassName の値が IBM の場合は、crossworlds.queue.manager を使用します。	crossworlds.queue.manager	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.Password	任意の有効なパスワード		コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.TransportOptimized	true または false	false	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS で、BrokerType の値が ICS の場合のみ有効です。
jms.UserName	任意の有効な名前		コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
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	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
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	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。

表 14. 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
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	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
ListenerConcurrency	1 から 100	1	コンポーネント再始動	このプロパティは、DeliveryTransport の値が MQ の場合のみ有効です。
Locale	これは、サポートされるロケールの一部です。 en_US、ja_JP、ko_KR、zh_CN、zh_TW、fr_FR、de_DE、it_IT、es_ES、pt_BR	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	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
MaxEventCapacity	1 から 2147483647	2147483647	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
MessageFileName	有効なファイル名	InterchangeSystem.txt	コンポーネント再始動	
MonitorQueue	任意の有効なキュー名	<CONNECTORNAME>/MONITORQUEUE	コンポーネント再始動	このプロパティは、DuplicateEventElimination の値が true で、ContainerManagedEvents に値がない場合にのみ有効です。
OADAutoRestartAgent	true または false	false	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
OADMaxNumRetry	正整数	1000	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。

表 14. 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
OADRetryTimeInterval	正整数 (単位: 分)	10	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定され、BrokerType の値が ICS の場合のみ有効です。
PollEndTime	HH = 0 から 23 MM = 0 から 59	HH:MM	コンポーネント再始動	
PollFrequency	正整数 (単位: ミリ秒)	10000	ブローカーが ICS の場合は動的。そうでない場合は、コンポーネント再始動。	
PollQuantity	1 から 500	1	エージェント再始動	このプロパティは、ContainerManagedEvents の値が JMS の場合のみ有効です。
PollStartTime	HH = 0 から 23 MM = 0 から 59	HH:MM	コンポーネント再始動	
RepositoryDirectory	ブローカーが ICS の場合は <REMOTE>。それ以外の場合は任意の有効なローカル・ディレクトリー。	ICS の場合、値は <REMOTE> に設定されます。	エージェント再始動	
RequestQueue	有効な JMS キュー名	<CONNECTORNAME>/REQUESTQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
ResponseQueue	有効な JMS キュー名	<CONNECTORNAME>/RESPONSEQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
RestartRetryCount	0 から 99	3	ICS の場合は動的、その他の場合はコンポーネント再始動	
RestartRetryInterval	1 から 2147483647 までの値 (分単位)。	1	ICS の場合は動的、その他の場合はコンポーネント再始動	
RHF2MessageDomain	mrm または xml	mrm	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS で、WireFormat の値が CwXML の場合のみ有効です。

表 14. 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
SourceQueue	任意の有効な WebSphere MQ キュー名	<CONNECTORNAME>/SOURCEQUEUE	エージェント再始動	このプロパティは、ContainerManagedEvents の値が JMS の場合のみ有効です。
SynchronousRequest Queue	任意の有効なキュー名	<CONNECTORNAME>/SYNCHRONOUSREQUEST QUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
SynchronousRequest Timeout	0 から任意の数 (ミリ秒)	0	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
SynchronousResponse Queue	任意の有効なキュー名	<CONNECTORNAME>/SYNCHRONOUSRESPONSE QUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
TivoliMonitorTransaction Performance	true または false	false	コンポーネント再始動	
WireFormat	CwXML または CwB0	CwXML	エージェント再始動	RepositoryDirectory の値が <REMOTE> に設定されていない場合、このプロパティの値は、CwXML でなければなりません。RepositoryDirectory の値が <REMOTE> に設定されている場合、値は CwB0 でなければなりません。
WsifSynchronousRequest Timeout	0 から任意の数 (ミリ秒)	0	コンポーネント再始動	BrokerType の値が ICS の場合、このプロパティは無効です。
XMLNameSpaceFormat	short または long	short	エージェント再始動	BrokerType の値が ICS の場合、このプロパティは無効です。

標準プロパティ

このセクションでは、標準コネクタ構成プロパティについて説明します。

AdapterHelpName

AdapterHelpName プロパティは、コネクタ固有の全般ヘルプ・ファイルがあるディレクトリの名前です。ディレクトリは、<ProductDir>%bin%Data%App%Help 内に配置される必要があり、少なくとも言語ディレクトリ enu_usa が含まれていなければなりません。ロケールに応じて、その他のディレクトリが含まれることがあります。

デフォルト値は、テンプレート名が有効であればテンプレート名、有効でなければ空白です。

AdminInQueue

AdminInQueue プロパティは、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際に使用されるキューを指定します。

デフォルト値は `<CONNECTORNAME>/ADMININQUEUE` です。

AdminOutQueue

AdminOutQueue プロパティは、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際に使用されるキューを指定します。

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

AgentConnections

AgentConnections プロパティは、ORB (オブジェクト・リクエスト・ブローカー) が初期化するときにかかれる ORB 接続の数を制御します。

このプロパティのデフォルト値は 1 です。

AgentTraceLevel

AgentTraceLevel プロパティは、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を設定します。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デリバリーします。

デフォルト値は 0 です。

ApplicationName

ApplicationName プロパティは、コネクタ・アプリケーションの名前を一意的に識別します。この名前は、システム管理者が統合環境をモニターするために使用します。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。

デフォルトはコネクタの名前です。

BrokerType

BrokerType プロパティは、使用している統合ブローカーのタイプを識別します。値は ICS です。

CharacterEncoding

CharacterEncoding プロパティは、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++ ベースのコネクタでは、このプロパティに `ascii7` という値が使用されます。

デフォルトでは、サポートされる文字エンコードの一部のみが表示されます。リストに、サポートされる他の値を追加するには、製品ディレクトリー (<ProductDir>)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、本書の付録『Connector Configurator Express』を参照してください。

ConcurrentEventTriggeredFlows

ConcurrentEventTriggeredFlows プロパティは、コネクタがイベントの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。この属性の値を、並行してマップおよびデリバリーされるビジネス・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。

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が、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・オブジェクトの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、以下のプロパティを構成する必要があります。

- **Maximum number of concurrent events** プロパティの値を増加して、複数のスレッドを使用できるようにコラボレーションを構成する必要があります。
- 宛先アプリケーションのアプリケーション固有コンポーネントを、複数の要求を並行して処理できるように構成する必要があります。

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

このプロパティは、**RepositoryDirectory** プロパティの値が <REMOTE> に設定されている場合のみ有効です。

デフォルト値は 1 です。

ContainerManagedEvents

ContainerManagedEvents 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、1 つの JMS トランザクションとして宛先キューに配置されます。

このプロパティを 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設定する必要があります。

- **PollQuantity** = 1 から 500
- **SourceQueue** = /SOURCEQUEUE

また、MimeType および DHClass (データ・ハンドラー・クラス) 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。DataHandlerConfigMOName (オプションのメタオブジェクト名) を追加することもできます。これらのプロパティの値を設定するには、Connector Configurator Express の「データ・ハンドラー」タブを使用します。

これらのプロパティはアダプター固有ですが、以下に値の例をいくつか示します。

- MimeType = text/xml
- DHClass = com.crossworlds.DataHandlers.text.xml
- DataHandlerConfigMOName = MO_DataHandler_Default

「データ・ハンドラー」タブのこれらの値のフィールドは、ContainerManagedEvents プロパティを JMS という値に設定した場合にのみ表示されます。

注: ContainerManagedEvents を JMS に設定した場合、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は使用できなくなります。

ContainerManagedEvents プロパティは、DeliveryTransport プロパティの値が JMS に設定されている場合のみ有効です。

デフォルト値はありません。

ControllerEventSequencing

ControllerEventSequencing プロパティは、コネクター・コントローラーでイベント順序付けを使用可能にします。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType は ICS) のみ有効です。

デフォルト値は true です。

ControllerStoreAndForwardMode

ControllerStoreAndForwardMode プロパティは、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 InterChange Server Express (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。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・コントローラーはアプリケーション固有のコンポーネントにその要求を転送します。

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・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。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合 (`BrokerType` プロパティの値が `ICS`) のみ有効です。

デフォルト値は `true` です。

ControllerTraceLevel

`ControllerTraceLevel` プロパティは、コネクタ・コントローラーのトレース・メッセージのレベルを設定します。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合のみ有効です。

デフォルト値は `0` です。

DeliveryQueue

`DeliveryQueue` プロパティは、コネクタが統合ブローカーへビジネス・オブジェクトを送信するときに使用するキューを定義します。

このプロパティは、`DeliveryTransport` プロパティの値が `JMS` に設定されている場合のみ有効です。

デフォルト値は `<CONNECTORNAME>/DELIVERYQUEUE` です。

DeliveryTransport

`DeliveryTransport` プロパティは、イベントのデリバリーのためのトランスポート機構を指定します。Java Messaging Service の場合、値は `JMS` です。

- `RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合、`DeliveryTransport` プロパティの値には `IDL` または `JMS` を指定することができ、デフォルトは `IDL` です。
- `RepositoryDirectory` プロパティの値がローカル・ディレクトリーの場合、値に使用できるのは `JMS` のみです。

`RepositoryDirectory` プロパティの値が `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。

デフォルト値は `JMS` です。

JMS

`JMS` トランスポート機構は、Java Messaging Service (`JMS`) を使用した、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は、`jms.MessageBrokerName`、`jms.FactoryClassName`、`jms.Password`、`jms.UserName` などの追加の `JMS` プロパティが `Connector Configurator Express` 内にリストされま

す。jms.MessageBrokerName プロパティおよび jms.FactoryClassName プロパティは、このトランスポートの必須プロパティです。

InterChange Server Express (ICS) が統合ブローカーである場合、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、メモリー制限が発生することもあります。

この環境では、WebSphere MQ MQ クライアント内でメモリーが使用されるため、(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 768MB 未満である場合には、次の変数およびプロパティを設定してください。

- 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 (<ProductDir>) 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 プロパティの値を 11 または 12 に設定する。このプロパティの詳細については、「*WebSphere Business Integration Server Express インストール・ガイド (Windows 版)*」、または「*WebSphere Business Integration Server Express インストール・ガイド (OS/400 版)*」を参照してください。

DuplicateEventElimination

このプロパティの値が true の場合、JMS 対応コネクタでは重複イベントがデリバリー・キューヘデリバリーされないようにすることができます。この機能を使用するには、コネクタ開発時に、コネクタに対し、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。

注: このプロパティの値が true の場合、保証付きイベント・デリバリーを提供するには、MonitorQueue プロパティを使用可能にする必要があります。

デフォルト値は false です。

EnableOidForFlowMonitoring

このプロパティの値が true の場合、アダプター・ランタイムは、着信 ObjectEventID にフロー・モニターの外部キーのマークを付けます。

このプロパティは、BrokerType プロパティが ICS に設定されている場合のみ有効です。

デフォルト値は false です。

FaultQueue

コネクタでメッセージを処理中にエラーが発生すると、コネクタは、そのメッセージ (および状況標識と問題説明) を `FaultQueue` プロパティで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。

jms.FactoryClassName

`jms.FactoryClassName` プロパティは、JMS プロバイダーのためにインスタンスを生成するクラス名を指定します。`DeliveryTransport` プロパティの値が JMS に設定されている場合、このプロパティを設定する必要があります。

デフォルト値は `CxCommon.Messaging.jms.IBMMQSeriesFactory` です。

jms.ListenerConcurrency

`jms.ListenerConcurrency` プロパティは、JMS コントローラーの並行リスナーの数を指定します。コントローラー内部で、並行してメッセージを取り出して処理するスレッドの数を指定します。

このプロパティは、`jms.OptimizedTransport` プロパティの値が `true` の場合のみ有効です。

デフォルト値は 1 です。

jms.MessageBrokerName

`jms.MessageBrokerName` は、JMS プロバイダーのため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として (`DeliveryTransport` プロパティで) 指定する場合、このコネクタ・プロパティを設定する必要があります。

リモート・メッセージ・ブローカーに接続した場合、このプロパティでは以下の値を指定する必要があります。

QueueMgrName:Channel:HostName:PortNumber

ここで、以下のように説明されます。

QueueMgrName は、キュー・マネージャー名です。

Channel は、クライアントが使用するチャンネルです。

HostName は、キュー・マネージャーの配置先のマシン名です。

PortNumber は、キュー・マネージャーが `listen` に使用するポートの番号です。

例えば、次のようになります。

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

デフォルト値は `crossworlds.queue.manager` です。ローカル・メッセージ・ブローカーに接続する場合は、デフォルト値を使用します。

jms.NumConcurrentRequests

`jms.NumConcurrentRequests` プロパティは、コネクタに対して同時に送信することができる並行サービス呼び出し要求の数 (最大値) 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はブロックされ、処理を続行するには他のいずれかの要求が完了するのを待機する必要があります。

デフォルト値は 10 です。

jms.Password

`jms.Password` プロパティは、JMS プロバイダーのため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。

デフォルト値はありません。

jms.TransportOptimized

`jms.TransportOptimized` プロパティは、WIP (処理中の作業) が最適化されるかどうかを決定します。WIP を最適化するには、WebSphere MQ プロバイダーが必要です。最適化された WIP が作動するためには、メッセージング・プロバイダーが以下の操作を実行できなければなりません。

1. メッセージをキューから削除せずに読み取る。
2. メッセージ全体を受信側のメモリー空間に転送することなく、固有の ID を使用してメッセージを削除する。
3. 固有の ID を使用してメッセージを読み取る (リカバリーのために必要)。
4. 読み取られなかったイベントが現れるポイントを追跡する。

JMS API は、上記の条件 2 および 4 を満たさないため、最適化された WIP には使用できませんが、MQ Java API は 4 つの条件をすべて満たすため、最適化された WIP には必要です。

このプロパティは、`DeliveryTransport` の値が JMS で、`BrokerType` の値が ICS の場合のみ有効です。

デフォルト値は `false` です。

jms.UserName

`jms.UserName` プロパティは、JMS プロバイダー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。

デフォルト値はありません。

JvmMaxHeapSize

`JvmMaxHeapSize` プロパティは、エージェントの最大ヒープ・サイズ (メガバイト単位) を指定します。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合のみ有効です。

デフォルト値は 128M です。

JvmMaxNativeStackSize

`JvmMaxNativeStackSize` プロパティは、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) を指定します。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合のみ有効です。

デフォルト値は 128K です。

JvmMinHeapSize

`JvmMinHeapSize` プロパティは、エージェントの最小ヒープ・サイズ (メガバイト単位) を指定します。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合のみ有効です。

デフォルト値は 1M です。

ListenerConcurrency

`ListenerConcurrency` プロパティは、統合ブローカーとして ICS を使用する場合は `WebSphere MQ Listener` でのマルチスレッド化をサポートしています。このプロパティにより、データベースへの複数イベントの書き込み操作をバッチ処理できるので、システム・パフォーマンスが向上します。

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のみで有効です。`DeliveryTransport` プロパティの値は MQ でなければなりません。

デフォルト値は 1 です。

Locale

`Locale` プロパティは、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順、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、以下のように説明されます。

`ll` は、2 文字の言語コード (小文字を使用) です。

`TT` は、2 文字の国または地域コード (大文字を使用) です。

`codeset` は、関連文字コード・セットの名前です (オプションの場合があります)。

デフォルトでは、サポートされるロケールの一部のみがリストされます。サポートされる他の値をリストに追加するには、`<ProductDir>%bin` ディレクトリーにある `%Data%Std%stdConnProps.xml` ファイルを変更します。詳細については、本書の付録『Connector Configurator Express』を参照してください。

コネクタが国際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、そのアダプターのユーザズ・ガイドを参照してください。

デフォルト値は en_US です。

LogAtInterchangeEnd

LogAtInterchangeEnd プロパティは、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。

ログ宛先にログを記録すると、E 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 ファイルで MESSAGE_RECIPIENT の値として指定された宛先に対する E メール・メッセージが生成されます。例えば、LogAtInterChangeEnd の値を true に設定した場合に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、E メール・メッセージが送信されます。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

デフォルト値は false です。

MaxEventCapacity

MaxEventCapacity プロパティは、コントローラー・バッファ内のイベントの最大数を指定します。このプロパティは、フロー制御機能によって使用されます。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

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

MessageFileName プロパティは、コネクタ・メッセージ・ファイルの名前を指定します。メッセージ・ファイルの標準位置は、製品ディレクトリーの %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・メッセージ・ファイルが存在しない場合は、コネクタは 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: コネクタについて、コネクタ独自のメッセージ・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ズ・ガイドを参照してください。

デフォルト値は InterchangeSystem.txt です。

MonitorQueue

MonitorQueue プロパティは、コネクタ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を指定します。

このプロパティは、DeliveryTransport プロパティの値が JMS で、DuplicateEventElimination の値が true の場合のみ有効です。

デフォルト値は <CONNECTORNAME>/MONITORQUEUE です。

OADAutoRestartAgent

OADAutoRestartAgent プロパティは、コネクタが自動再始動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、WebSphere MQ により起動される Object Activation Daemon (OAD) を使用して、異常シャットダウン後にコネクタを再始動したり、System Monitor からリモート・コネクタ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、このプロパティを true に設定する必要があります。WebSphere MQ によりトリガーされる OAD 機能の構成方法については、「*WebSphere Business Integration Server Express インストール・ガイド (Windows 版)*」、または「*WebSphere Business Integration Server Express インストール・ガイド (OS/400 版)*」を参照してください。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

デフォルト値は false です。

OADMaxNumRetry

OADMaxNumRetry プロパティは、異常シャットダウンの後で WebSphere MQ によりトリガーされる Object Activation Daemon (OAD) がコネクタ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。このプロパティを有効にするためには、OADAutoRestartAgent プロパティを true に設定する必要があります。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

デフォルト値は 1000 です。

OADRetryTimeInterval

OADRetryTimeInterval プロパティは、WebSphere MQ によりトリガーされる Object Activation Daemon (OAD) の再試行時間間隔の分数を指定します。コネクタ・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・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ためには、OADAutoRestartAgent プロパティを true に設定する必要があります。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合 (BrokerType の値が ICS) のみ有効です。

デフォルト値は 10 です。

PollEndTime

PollEndTime プロパティは、イベント・キューのポーリングを停止する時刻を指定します。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必ず有効な値を指定してください。デフォルト値は、値を含まない *HH:MM* であるため、この値は必ず変更する必要があります。

アダプター・ランタイムが以下のことを検出した場合、

- **PollStartTime** が設定されて、**PollEndTime** が設定されていない、または
- **PollEndTime** が設定されて、**PollStartTime** が設定されていない

PollFrequency プロパティに構成された値を使用してポーリングします。

PollFrequency

PollFrequency プロパティは、あるポーリング・アクションの終了から次のポーリング・アクションの開始までの時間をミリ秒単位で指定します。これはポーリング・アクション間の間隔ではありません。この論理を次に説明します。

- ポーリングし、**PollQuantity** プロパティの値により指定される数のオブジェクトを取得します。
- これらのオブジェクトを処理します。一部のコネクタでは、これは個別のスレッドで部分的に実行されます。これにより、次のポーリング・アクションまで処理が非同期に実行されます。
- **PollFrequency** プロパティで指定された間隔にわたって遅延します。
- このサイクルを繰り返します。

このプロパティでは、以下の値が有効です。

- ポーリング・アクション間のミリ秒数 (正整数)。
- ワード **no**。コネクタはポーリングを実行しません。このワードは小文字で入力します。
- 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。このワードは小文字で入力します。

デフォルト値は 10000 です。

重要: 一部のコネクタでは、このプロパティの使用が制限されています。このようなコネクタが存在する場合には、アダプターのインストールと構成に関する章で制約事項が説明されています。

PollQuantity

PollQuantity プロパティは、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。

このプロパティは、DeliveryTransport プロパティの値が JMS で、ContainerManagedEvents プロパティに値がある場合のみ有効です。

電子メール・メッセージもイベントと見なされます。コネクタは、E メールに関するポーリングを受けたときには次のように動作します。

- 一度ポーリングされると、コネクタはメッセージの本文を検出し、それを添付ファイルとして読み取ります。本文の MIME タイプにはデータ・ハンドラーが指定されていないので、コネクタはメッセージを無視します。
- コネクタは最初の BO 添付ファイルを処理します。この MIME タイプには対応するデータ・ハンドラーがあるので、コネクタはビジネス・オブジェクトを Visual Test Connector に送信します。
- 2 回目のポーリングを受けると、コネクタは BO の 2 番目の添付を処理します。この MIME タイプには対応するデータ・ハンドラーがあるので、コネクタはビジネス・オブジェクトを Visual Test Connector に送信します。
- それが受け入れられると、3 番目の BO 添付ファイルが送信されます。

PollStartTime

PollStartTime プロパティは、イベント・キューのポーリングを開始する時刻を指定します。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必ず有効な値を指定してください。デフォルト値は、値を含まない HH:MM であるため、この値は必ず変更する必要があります。

アダプター・ランタイムが以下のことを検出した場合、

- PollStartTime が設定されて、PollEndTime が設定されていない、または
- PollEndTime が設定されて、PollStartTime が設定されていない

PollFrequency プロパティに構成された値を使用してポーリングします。

RepositoryDirectory

RepositoryDirectory プロパティは、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保管されています。

統合ブローカーが ICS の場合は、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 Express リポジトリからこの情報を取得するためです。

RequestQueue

RequestQueue プロパティは、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を指定します。

このプロパティは、DeliveryTransport プロパティの値が JMS の場合のみ有効です。

デフォルト値は <CONNECTORNAME>/REQUESTQUEUE です。

ResponseQueue

ResponseQueue プロパティは、JMS 応答キューを指定します。JMS 応答キューは、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ヘデリバリーします。統合ブローカーが InterChange Server Express (ICS) の場合、サーバーは要求を送信し、JMS 応答キューの応答メッセージを待ちます。

このプロパティは、DeliveryTransport プロパティの値が JMS の場合のみ有効です。

デフォルト値は <CONNECTORNAME>/RESPONSEQUEUE です。

RestartRetryCount

RestartRetryCount プロパティは、コネクタによるコネクタ自体の再始動の試行回数を指定します。このプロパティを並列に接続された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クライアント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

RestartRetryInterval プロパティは、コネクタによるコネクタ自体の再始動の試行間隔を分単位で指定します。このプロパティを並列にリンクされた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クライアント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。

プロパティに使用可能な値の範囲は 1 から 2147483647 です。

デフォルト値は 1 です。

RHF2MessageDomain

RHF2MessageDomain 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 WebSphere Message Broker 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。構成可能ドメイン名によって、WebSphere Message Broker がメッセージ・データを処理する方法を追跡できます。

ヘッダーの例を示します。

```
<mcd><Msd>mrm</Msd><Set>3</Set><Type>  
Retek_POPhyDesc</Type><Fmt>CwXML</Fmt></mcd>
```

BrokerType の値が ICS の場合、このプロパティは無効です。また、このプロパティは、DeliveryTransport プロパティの値が JMS で、WireFormat プロパティの値が CwXML の場合のみ有効です。

可能な値は、`mrm` および `xml` です。デフォルト値は `mrm` です。

SourceQueue

SourceQueue プロパティは、JMS イベント・ストアを使用する JMS 対応コネクタで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・フレームワークに、JMS ソース・キューを指定します。詳細については、98 ページの『ContainerManagedEvents』を参照してください。

このプロパティは、DeliveryTransport の値が JMS で、ContainerManagedEvents の値が指定されている場合のみ有効です。

デフォルト値は <CONNECTORNAME>/SOURCEQUEUE です。

SynchronousRequestQueue

SynchronousRequestQueue プロパティは、同期応答を要求する要求メッセージを、コネクタ・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、コネクタが同期実行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・フレームワークは、同期要求キューにメッセージを送信し、同期応答キューでブローカーからの応答を待機します。コネクタ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。

このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

デフォルト値は <CONNECTORNAME>/SYNCHRONOUSREQUESTQUEUE です。

SynchronousRequestTimeout

SynchronousRequestTimeout プロパティは、コネクタが同期要求への応答を待機する時間をミリ秒単位で指定します。コネクタは、指定された時間内に応答を受信できなかった場合、元の同期要求メッセージ (およびエラー・メッセージ) を障害キューに移動します。

このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

デフォルト値は 0 です。

SynchronousResponseQueue

SynchronousResponseQueue プロパティは、同期要求に対する応答メッセージを、ブローカーからコネクタ・フレームワークにデリバリーします。このキューは、コネクタが同期実行を使用する場合にのみ必要です。

このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

デフォルトは <CONNECTORNAME>/SYNCHRONOUSRESPONSEQUEUE です。

TivoliMonitorTransactionPerformance

TivoliMonitorTransactionPerformance プロパティは、IBM Tivoli Monitoring for Transaction Performance (ITMTP) を実行時に起動するかどうかを指定します。

デフォルト値は false です。

WireFormat

WireFormat プロパティは、トランスポートのメッセージ・フォーマットを指定します。

- RepositoryDirectory プロパティの値がローカル・ディレクトリーの場合、値は CwXML です。
- RepositoryDirectory プロパティの値がリモート・ディレクトリーの場合、値は CwB0 です。

付録 B. コネクタ固有のプロパティ

コネクタ固有の構成プロパティには、コネクタが実行時に必要とする情報が用意されています。コネクタ固有のプロパティを使用すると、エージェントの再コード化と再構築をしなくても、コネクタ内で静的な情報または論理を変更することもできます。

以下の表に、コネクタのコネクタ固有の構成プロパティを示します。プロパティの説明については、以下の各セクションを参照してください。

表 15. コネクタ固有の構成プロパティ

名前	意味	指定可能な値	デフォルト値
ApplicationPassword	Siebel ユーザー・アカウントのパスワード		CWCONN
ApplicationUserName	Siebel アプリケーションのユーザー・アカウント		CWCONN
ArchiveProcessed	現行のサブスクリプションがないイベントをコネクタがアーカイブするかどうかを指定します。	True または False	True
ConnectErrors	Siebel から戻され、コネクタ内で検査される一連のエラー。これらは重大なエラーと見なされ、検出されると、コネクタは終了します。	ネットワーク障害ネットワークまたは接続障害メッセージ。これらのメッセージは、「;」区切り文字で区切られます。	
ConnectorID	システムが複数のコネクタを処理するように構成されている場合に使用されます。	コネクタを示す整数値。	
ConnectString	Siebel Object Manager に接続するために Siebel Java Data Bean が使用するストリング。	protocol://machinename/enterprisename/objectmanager/servername	なし
ConnectString (Siebel バージョン 7.5 用)	Siebel Object Manager に接続するために Siebel Java Data Bean が使用するストリング。	protocol://machinename/enterprisename/objectmanager/servername	なし
ConnectString (Siebel バージョン 7.7 用)	Siebel Object Manager に接続するために Siebel Java Data Bean が使用するストリング。	protocol://machinename:portno/enterprisename/objectmanager	なし
DataBeanPoolSize	データ Bean プール内の最大 Bean 数を示します。	Bean プール・サイズを決定する整数。	

表 15. コネクタ固有の構成プロパティ (続き)

名前	意味	指定可能な値	デフォルト値
DataBeanRefreshInterval	この値は、コネクタが Siebel 6.2.x に対して動作しているときに Siebel データ Bean リソースをリフレッシュするために使用します。コネクタは、処理された要求がこの値に等しくなった後にログオフし、再度ログインします。	リフレッシュされるまでにコネクタで処理される要求数に対応する DataBeanRefreshInterval を示す整数値。	
EventProcessingSupport	アダプターがイベントを処理するかどうかを示します。必要に応じて、サブスクリプション・サービスをオフに切り替えるために使用できます。	Boolean	True
PollAttributeDelimiter	オブジェクト・キー内の複数の名前値ペアの場合、この値によりキー間の区切り文字が決定されます。この値を設定しない場合のデフォルトは ; (セミコロン) です。	文字	;
PollQuantity	pollForEvents 呼び出しで処理されるイベント数を決定します。	pollForEvents 呼び出しで処理されるイベントの数を表す整数。	1
ResonateSupport	Resonate が Siebel サーバーにインストールされているかどうかを示します。コネクタ Bean プールは、Resonate がインストールされている場合のみ、Attach/Detach 呼び出し (Siebel7) を使用します。Resonate がインストールされていない場合、このプールは特定の数の要求を処理した後にログオフします。	ブール値 (Bean からのログオフは DataBeanRefreshInterval で決定される) の設定。	false
SiebelLanguageCode	サポートされる言語に対して Siebel で使用される 3 文字の NLS 文字セット・コード。デフォルトは米国英語で、NLS 表記では ENU となります。	Siebel 7 の言語コードでサポートされている言語は、次のとおりです。Italian (Std) -- ITA Japanese -- JPN Korean -- KOR Norwegian -- NOR (Bokmal) Polish -- POL Portuguese -- PTB (Brazil) Portuguese -- PTG (Portugal) Russian -- RUS Spanish -- ESN (Modern Sort) Swedish -- SVE Turkish -- TUR English (US) -- ENU および NLS によってサポートされている他のすべての言語	

表 15. コネクタ固有の構成プロパティ (続き)

名前	意味	指定可能な値	デフォルト値
SiebelVersion	SchemeVersion Siebel ビジネス・オブジェクトにアクセスしてバージョンを取得しなくても、指定されたバージョンの Siebel に対してアダプターを実行できます。デフォルト値の使用を推奨します。	6、7、なし	なし
SupportNameValuePair	イベントのオブジェクト・キーのフォーマットを決定するために使用します。この値が設定されていないか、true に設定されている場合、オブジェクト・キー値は名前と値の間に「=」を持つ名前値ペアでなければなりません。False に設定されている場合は、rowId を 1 つのみ指定可能です。	True または False	False
UseDefaults	Create 操作において、コネクタが必要な各ビジネス・オブジェクト属性について有効な値やデフォルト値が設定されているかを検査するかどうかを決定します。	True または False	False
ViewMode	後方互換性を保つために維持されています。ユーザーのアクセス権を決定する整数値。VM ASI タグがビジネス・オブジェクト・レベルで指定されていないならば、このプロパティに指定された値が使用されます。	整数値。詳細については、VM asi を参照してください。	

ApplicationPassword

アプリケーション・ユーザー・アカウントのパスワード。

デフォルト値はありません。

ApplicationUserName

アプリケーション・ユーザー・アカウントの名前。

デフォルト値はありません。

ArchiveProcessed

現行のサブスクリプションがないイベントをコネクタがアーカイブするかどうかを指定します。

このプロパティを `true` に設定すると、イベントは `Event` ビジネス・コンポーネントから削除された後に `Archive` ビジネス・コンポーネントに挿入されます。

このプロパティを `false` に設定すると、コネクタはアーカイブ処理を行いません。`ArchiveProcessed` が `false` に設定されると、コネクタは以下のように動作します。

- イベントが正常に処理されると、コネクタは `Event` ビジネス・コンポーネントからイベントを削除します。
- コネクタがイベントのビジネス・オブジェクトにサブスクライブしない場合、コネクタはイベントを `Event` ビジネス・コンポーネント内に残し、イベント状況を `Unsubscribed` に変更します。
- 処理中にビジネス・オブジェクトに問題が発生すると、コネクタはイベントをイベント表に残し、イベント状況を `error` 状況に設定します。

このプロパティが `false` に設定されており、かつ、ポーリング量が少ない場合、コネクタはイベント表をポーリングしているように見えますが、単に同じイベントを繰り返しピックアップしているに過ぎません。

このプロパティに値がない場合、コネクタは値を `true` と見なします。

デフォルト値は `true` です。

ConnectErrors

Siebel から戻された接続エラーです。コネクタでこれらのエラーが発生すると、コネクタは強制終了します。

ConnectorID

コネクタの固有の ID。この ID は、コネクタの特定のインスタンスのイベントを検索する場合に役立ちます。

デフォルト値はヌルです。

ConnectString

Siebel Object Manager に接続するために Siebel Java Data Bean が使用するストリング。

設定する値は、使用している Siebel のバージョンによって異なります。このプロパティにはデフォルト値はありません。

DataBeanPoolSize

データ Bean プール内の最大 Bean 数を示す整数。

DataBeanRefreshInterval

Siebel データ Bean リソースをリフレッシュする呼び出しの前にコネクターで処理される要求数を示す整数値。コネクターを Siebel 6.x で実行した場合に、コネクターによって使用されます。

EventProcessingSupport

EventProcessingSupport が true に設定されている場合、アダプターはイベントを処理します。EventProcessingSupport が false に設定されている場合、アダプターはイベントを処理しません。

デフォルト値は true です。

PollAttributeDelimiter

イベント・テーブルのオブジェクト・キー欄で名前と値のペアが複数使用されているときは、この文字値によって複数のキーの間に置かれる区切り文字が決まります。

この値を設定しない場合のデフォルトは、 ; です。

PollQuantity

コネクターがポーリング間隔ごとに検索するデータベース表の行数。指定可能な値は 1 から 500 です。

デフォルトは、1 です。

ResonateSupport

Siebel 7.x で使用されます。Resonate サポートが Siebel サーバーにインストールされているかどうかを示すブール値。コネクターを Siebel 7.x で実行する場合に ResonateSupport を true に設定すると、コネクターは DataBeanRefreshInterval プロパティに設定された値とともにこのプロパティを使用して、データ Bean プールからのログオフを決定します。

ResonateSupport を true に設定すると、コネクターは Attach 呼び出しと Detach 呼び出しを使用して、要求の処理が完了するごとに既存のセッションと接続したりセッションから切断したりします。ResonateSupport を false に設定すると、コネクターは一定数の要求を処理した後にログオフします。

デフォルト設定は false です。

SiebelLanguageCode

サポートされる言語に対して Siebel で使用される 3 文字の NLS 文字セット・コード。デフォルトは米国英語で、NLS 表記では ENU となります。

SiebelVersion

Siebel ビジネス・オブジェクト Schema Version にアクセスしてバージョンを取得しなくても、指定されたバージョンの Siebel に対してアダプターを実行できます。Siebel バージョン 6 の場合は 6 を設定し、Siebel バージョン 7 の場合は 7 を設定します。

デフォルト値はなしです。デフォルト値を使用する場合、アダプターは Schema バージョンから Siebel バージョンを取得します。デフォルト値の使用を推奨します。

SupportNameValuePair

イベントのオブジェクト・キーのフォーマットを決定するために使用します。true に設定されている場合、または値が設定されていない場合は、オブジェクト・キー値は名前と値のペア (名前と値の間を「=」で区切った値) になります。

false に設定されている場合は、オブジェクト・キーに rowID を 1 つのみ指定可能です。複数のキーはサポートされません。

デフォルト設定は true です。

UseDefaults

UseDefaults が true に設定されているかまたは設定されていない場合、コネクターは必要な各ビジネス・オブジェクト属性について有効な値やデフォルト値が設定されているかどうかを検査します。値が設定されていると、Create は成功します。設定されていない場合は、失敗します。

UseDefaults が false に設定されている場合、コネクターは必要な各ビジネス・オブジェクト属性について有効な値が設定されているかどうかのみを検査します。有効な値が設定されていないと、Create 操作は失敗します。

デフォルト値は false です。

ViewMode

ユーザーのアクセス権を決定する整数。ViewMode アプリケーション固有情報タグがビジネス・オブジェクト・レベルで指定されていない場合、コネクター・プロパティーで指定された値が使用されます。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アーカイブ表 11
アダプター 1
 定義 1
アダプター環境 9
アダプターの依存関係 9
アダプターのプラットフォーム 9
アダプター・フレームワーク 1
アプリケーション固有の情報 42
アプリケーション・パスワード 115
イベント検出 83
イベント処理 11
イベント表 11
インストール 10
インストールされるファイル 10
失われた接続 8

[カ行]

外部キー関係 44
カスタム・ビジネス・サービス 79
キー属性 40
検索要求 4
更新要求 5
子オブジェクト 47
コネクタ固有のプロパティ 113
コネクタの構成 19
コネクタの定義 2
コネクタ・アーキテクチャー 3
コネクタ・コントローラー 2
コネクタ・フレームワーク 2
コンテナ属性 46
コンテナ属性のアプリケーション固有情報 46

[サ行]

削除要求 5
作成要求 5
スマート・フィルター 8
属性値の再割り当て 49
属性プロパティ 40

[タ行]

タイプ, 子オブジェクトの 41
単純属性のアプリケーション固有情報 43
データ型マッピング 59
統合ブローカー 2
動詞, キー属性 52
動詞, ビジネス・サービスでの 82
動詞アプリケーション固有情報 51
トラブルシューティング 87

[ハ行]

ビジネス・オブジェクト定義
 作成 62, 74
ビジネス・オブジェクト, 構造 39
ビジネス・オブジェクト, 作成 53
ビジネス・オブジェクト, 定義の生成 70
ビジネス・オブジェクト, 変更 53
ビジネス・オブジェクトの関連 39
ビジネス・サービス 79
ピック・リスト 50
複数のレコード 51

[マ行]

メモリー制限 87

[ヤ行]

ユーザー・アカウント 9
用語 1

[ラ行]

ログ・ファイル 87

A

Adapter Development Kit (ADK) 2
Application Service Interface 79
ApplicationUserName 115
AppSpecificInfo 41
ArchiveProcessed 116

B

Business Object Designer 61

C

Cardinality 41
ConnectErrors 116
ConnectorID 116
ConnectString 116
ContainedObjectVersion 41

D

DataBeanPoolSize 116
DataBeanRefreshInterval 117
DefaultValue 41

E

EAI Siebel Adapter 79
EventProcessingSupport 117
Exists 動詞 5

I

IsForeignKey 41
IsKey 41
IsRequired 41

M

MaxLength 40

O

Object Discovery Agent (ODA)
 構成プロパティ 63
 ソースの選択 65

P

PollAttributeDelimiter 117
PollQuantity 117

R

ResonateSupport 117
RetrieveByContent 要求 4

S

Siebel Object Discovery Agent 57
Siebel ODA のインストール 58
Siebel ODA の起動 59
Siebel ODA、依存関係 58
Siebel ODA、エラー 60
Siebel ODA、構成プロパティ 63
Siebel ODA、複数インスタンス 60
Siebel アプリケーション・アーキテクチャ
ー 2
SiebelLanguageCode 117
SiebelVersion 118
Siebel_BCAccount 53
Siebel_BCAsset 56
Siebel_BCContact 55
Siebel_BCInternalProduct 56
Siebel_BCQuote 54
SupportNameValuePair 118

T

Type 40

U

UseDefaults 118

V

ViewMode 118

W

WebSphere Business Integration システム
2
Windows vii, 9

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認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

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。

著作権使用許諾

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

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
CICS
CrossWorlds
DB2
DB2 Universal Database
IMS
Informix
iSeries
Lotus
Lotus Domino
Lotus Notes
MQIntegrator
MQSeries
MVS
OS/400
Passport Advantage
SupportPac
WebSphere
z/OS

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 および Pentium 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。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。



|

WebSphere Business Integration Adapter Framework V2.6.0



Printed in Japan