

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



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

バージョン 4.3



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



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

バージョン 4.3

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、113ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM WebSphere Business Integration Server Express Plus バージョン 4.3、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

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

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

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

(URL は、変更になる場合があります)

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

原 典： WebSphere Business Integration Server Express and Plus  
Adapter for Siebel eBusiness Applications User Guide  
Version 4.3

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2004.8

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

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

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

© Copyright IBM Japan 2004

---

# 目次

本書について	v
対象読者	v
関連資料	v
表記上の規則	vi
<b>本リリースでの新機能</b>	<b>vii</b>
リリース 4.3 の新機能	vii
<b>第 1 章 アダプターの概要</b>	<b>1</b>
コネクターの定義	1
Siebel アプリケーション・アーキテクチャー	1
コネクター・アーキテクチャー	2
コネクターの動作方法	2
<b>第 2 章 コネクターのインストールと構成</b>	<b>9</b>
前提条件	9
アダプターのインストール	10
コネクターを使用するためにインストールされたファイル	10
イベント表およびアーカイブ表	11
コネクターの構成	18
複数のコネクター・インスタンスの作成	23
コネクターの始動	25
コネクターの停止	26
<b>第 3 章 ビジネス・オブジェクトの理解</b>	<b>27</b>
ビジネス・オブジェクトの構造および関係	27
ビジネス・オブジェクトのアプリケーション固有の情報	30
ビジネス・オブジェクトのサンプルの使用	40
<b>第 4 章 SiebelODA を使用したビジネス・オブジェクト定義の生成</b>	<b>45</b>
インストールと使用方法	45
Business Object Designer Express 内での SiebelODA の使用	48
生成される定義の内容	60
ビジネス・オブジェクト定義への情報の追加	63
<b>第 5 章 Siebel 接続 DLL の使用</b>	<b>65</b>
コネクター・アーキテクチャーおよびプロセス	65
Siebel 接続 DLL の構成	67
Siebel 接続 DLL との通信	70
IBM ビジネス・オブジェクトの生成	70
IBM 汎用ビジネス・サービスの構成	71
<b>第 6 章 Siebel ビジネス・サービスのサポート</b>	<b>73</b>
ビジネス・サービスの理解	73
ビジネス・サービスにおける動詞処理	75
ビジネス・サービスにおけるイベント検出	77
カスタム・ビジネス・サービスのサポート	77
<b>第 7 章 トラブルシューティング</b>	<b>79</b>
エラー処理	79

Siebel ログ・ファイルのサイズの削減	80
<b>付録 A. コネクターの標準構成プロパティ</b>	<b>81</b>
標準コネクタ・プロパティの構成	81
標準プロパティの要約	82
標準構成プロパティ	85
<b>付録 B. Connector Configurator Express</b>	<b>97</b>
Connector Configurator Express の概要	97
Connector Configurator Express の始動	98
System Manager からの Configurator Express の実行	98
コネクタ固有のプロパティ・テンプレートの作成	99
新しい構成ファイルを作成	101
既存ファイルの使用	102
構成ファイルの完成	104
構成ファイル・プロパティの設定	104
構成ファイルの保管	110
構成の完了	111
グローバル化環境における Connector Configurator Express の使用	111
<b>特記事項</b>	<b>113</b>
プログラミング・インターフェース情報	114
商標	115

---

## 本書について

製品 IBM<sup>(R)</sup>WebSphere Business Integration Server Express および IBM<sup>(R)</sup> 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。

本書では、IBM WebSphere Business Integration Server Express Plus Adapter for Siebel eBusiness Applications の構成、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。

---

## 対象読者

本書は、WebSphere Business Integration Server Express Plus 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

---

## 関連資料

本書の対象製品の一連の関連文書には、WebSphere Business Integration Server Express Plus のどのインストールにも共通する機能とコンポーネントの解説のほか、特定のコンポーネントに関する参考資料が含まれています。

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

**注:** 本書の発行後に公開されたテクニカル・サポートの技術情報や速報に、本書の対象製品に関する重要な情報が記載されている場合があります。これらの技術情報や速報は、WebSphere Business Integration のサポート Web サイト (<http://www.ibm.com/software/integration/websphere/support/>) で参照できます。適切なコンポーネント領域を選択し、「Technotes (技術情報)」セクションと「Flashes (速報)」セクションを参照してください。

上記のサイトには資料のダウンロード、インストール、および表示に関する簡単な説明が記載されています。

---

## 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 4.3 の新機能

本書の最初のリリースです。



---

## 第 1 章 アダプターの概要

この章では、IBM WebSphere Business Integration Server Express Plus の Adapter for Siebel eBusiness Applications コンポーネントについての概要を説明します。次のセクションで構成されています。

- 『コネクターの定義』
- 『Siebel アプリケーション・アーキテクチャー』
- 2 ページの 『コネクター・アーキテクチャー』
- 3 ページの 『ビジネス・オブジェクトの処理』
- 6 ページの 『イベント管理』
- 7 ページの 『Siebel アプリケーションへの接続逸失時の処理』

---

### コネクターの定義

コネクターは、コネクター・フレームワークとアプリケーション固有コンポーネントから構成されています。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に共通です。コネクター・フレームワークは、InterChange Server Express とアプリケーション固有のコンポーネントの間を仲介します。アプリケーション固有のコンポーネントには、特定のアプリケーションに対して調整されたコードが含まれています。コネクター・フレームワークは、InterChange Server Express とアプリケーション固有のコンポーネントの間で以下のサービスを提供します。

- ビジネス・オブジェクトの受信と送信
- 始動メッセージや管理メッセージの交換の管理

本書では、コネクター・フレームワークと、コネクターと呼ぶアプリケーション固有コンポーネントについて説明しています。

アダプターのコネクター・コンポーネントは、WebSphere Business Integration Server Express Plus Adapter フレームワークと Siebel eBusiness Applications コンポーネントを使用して、Siebel 6.2.x、7.0 または 7.5 (英語) のいずれかと Siebel 7 (日本語) のみへのコネクティブティティーを提供します。

コネクターは、Siebel Java Data Bean で提供される Java API を使用して、データ交換のために Siebel Object Manager と通信します。Siebel Java Data Bean は、Siebel アプリケーション・アーキテクチャーのビジネス・オブジェクト・レイヤーの一部である Siebel ビジネス・コンポーネントとビジネス・オブジェクトを公開します。

---

### 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ヨナル・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・アーキテクチャ

コネクタは、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・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・アーキテクチャを示したものです。統合ブローカーは InterChange Server Express です。

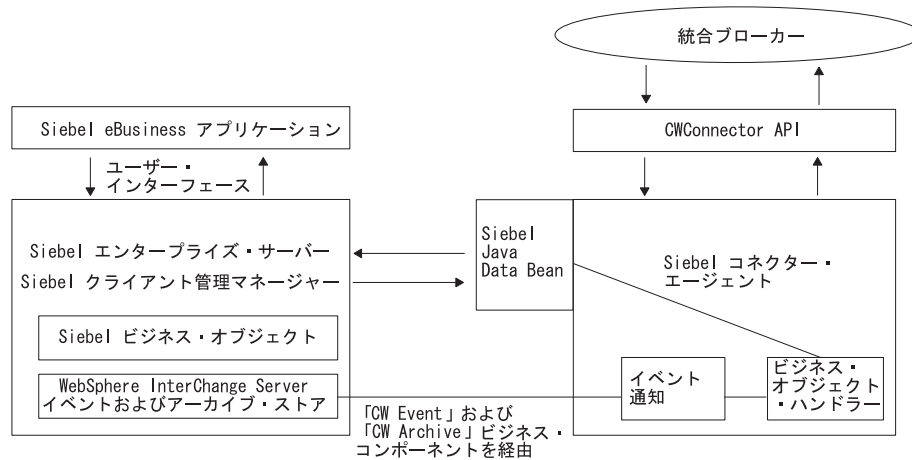


図 1. Siebel コネクタ・アーキテクチャ

## コネクタの動作方法

ここでは、メタデータでコネクタ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。

## コネクタおよびメタデータ

コネクタ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は、コネクタ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、73 ページの『第 6 章 Siebel ビジネス・サービスのサポート』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。コネクタはメタデータ主導型なので、コネクタ・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。WebSphere Business Integration Server Express Plus の統合ブローカーは、InterChange Server Express です。

### InterChange Server Express 要求の処理

コネクタがビジネス・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:** InterChange Server Express 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に要求すると、コネクタは 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 InterChange Server Express 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、InterChange Server Express 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、コネクタは InterChange Server Express 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):** InterChange Server Express 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に要求すると、コネクタ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:** InterChange Server Express 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に要求すると、コネクタ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 内の複数值リンクであり、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の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ません。

**ビジネス・オブジェクトの削除:** InterChange Server Express がレコードの削除をコネクタに要求すると、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 (UID)
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 READY_FOR_POLL 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。IN_PROGRESS=1 -- 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 IN_PROGRESS に変更します。UNSUBSCRIBED=2 -- イベントはサブスクライブされていません。isSubscribed 呼び出しが False を戻すと、コネクタは状況を UNSUBSCRIBED に設定します。SUCCESS=3 -- 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 SUCCESS に設定します。ERROR_PROCESSING_EVENT=-1 -- 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。ERROR_POSTING_EVENT=-2 -- 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、gotApp1Event への呼び出しが pollForEvents で失敗した場合に設定されます。ERROR_OBJECT_NOT_FOUND=-3 -- 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、doVerbFor 呼び出しで pollForEvents 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。
Description	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 InterChange Server Express 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 InterChange Server Express 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 InterChange Server Express 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 プロパティの詳細については、18 ページの『コネクタの構成』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



表 2. アーカイブ時の振る舞い (続き)

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
	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



---

## 第 2 章 コネクターのインストールと構成

この章では、WebSphere Business Integration Server Express Plus の Adapter for Siebel eBusiness Applications のコンポーネントをインストールして構成する方法について説明します。以下の項目について説明します。

- 『前提条件』
- 10 ページの『アダプターのインストール』
- 10 ページの『コネクターを使用するためにインストールされたファイル』
- 11 ページの『イベント表およびアーカイブ表』
- 18 ページの『コネクターの構成』
- 25 ページの『コネクターの始動』

**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。

---

### 前提条件

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.x、または Siebel 7.5 .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・アカウントは完全なアクセス権を持ち、ログイン名は ApplicationUserName 構成プロパティー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。

コネクター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` ディレクトリー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` ディレクトリー内の `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 7x または 7.5

- SiebelJI\_Common.jar
- SiebelJI\_enu.jar

---

## アダプターのインストール

アダプターのインストールについては、次のサイトの WebSphere Business Integration Server Express InfoCenter にある「*WebSphere Business Integration Server Express インストール・ガイド*」の、Adapter Capacity Pack for WebSphere Business Integration Server Express Plus からのインストールについての説明を参照してください。

<http://www.ibm.com/websphere/wbiserverexpress/infocenter>

---

## コネクタを使用するためにインストールされたファイル

以下のセクションでは、Windows 2000 システムにインストールされたコネクタ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 (Windows 2000 システムの場合)

表 3 では、コネクタで使用される Windows ファイル構造について説明します。

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

<b>%ProductDir%</b> のサブディレクトリー	説明
¥connectors¥Siebel	コネクタ用の CWSiebel.jar および start_Siebel.bat ファイルが含まれています。
¥connectors¥Siebel¥dependencies	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、Siebel.jar ファイルも含まれています。
¥connectors¥messages	関連するメッセージ・ファイル SiebelConnector.txt が含まれています。
¥connectors¥Siebel¥Samples¥Repository	以下の BO サンプルが含まれています。 Siebel_BCAccount Siebel_BCQuote Siebel_BCContact Siebel_BCInternalProduct Siebel_BCAsset
¥repository¥Siebel¥lib	CN_Siebel.txt ファイルが含まれています。
¥bin	WBIA.jar ファイルが含まれています。 CWConnEnv.bat ファイルが含まれています。

インストーラーは、コネクター・ファイルのアイコンを WebSphere Business Integration Server Express Plus のメニューに追加します。コネクター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: アダプターのインストール方法については、「*WebSphere Business Integration Server Express インストール・ガイド*」を参照してください。

---

## イベント表およびアーカイブ表

コネクターは、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、コネクター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。

コネクターはイベントごとに、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、コネクター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、コネクター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、コネクターは 1 回の取り出しで両方のデータ変更を取得します。

コネクター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。

コネクター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、不必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、コネクター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ー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ーの固有な ID

## Siebel バージョン 7.5 内で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。
  - 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 Server Express Plus 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\_Archive\_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\_ARCHIVE\_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\_ARCHIVE\_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。

## 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。
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 Server Express Plus 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、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、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 ConnectorId 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 分以上アイドル状態である場合、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。

---

## コネクタの構成

コネクタには、標準の構成プロパティとコネクタ固有の構成プロパティの 2 つのタイプの構成プロパティがあります。コネクタを実行する前に、これらのプロパティの一部の値を設定する必要があります。

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。構成情報の詳細については、97 ページの『付録 B. Connector Configurator Express』を参照してください。

### 標準コネクタ・プロパティ

標準の構成プロパティにより、すべてのコネクタによって使用される情報が提供されます。これらのプロパティの詳細については、81 ページの『付録 A. コネクタの標準構成プロパティ』を参照してください。

表 4 では、付録に記載された構成プロパティのうち、このプロパティに固有の情報を示しています。

表 4. このコネクタに固有のプロパティ情報

プロパティ	注
Character Encoding	このコネクタはこのプロパティを使用しません。

表 4. このコネクタに固有のプロパティ情報 (続き)

プロパティ	注
Locale	このコネクタは国際化対応されていないため、このプロパティの値を変更することはできません。

**注:** インストールされているすべてのコンポーネント (例えば、すべてのアダプター、アプリケーション、および InterChange Server Express 自体) に、同一のロケールがセットされていることを確認します。

## 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。

表 5 に、そのコネクタの、コネクタ固有構成プロパティをリストします。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 5. 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
ConnectString	新しい Siebel アーキテクチャでは、Siebel Java Data Bean は接続ストリングを使用して Siebel Object Manager に接続することができます。Siebel Java Data Bean が構成ファイルを認識する必要はなく、単にファイルを読み取り、Object Manager に接続します。	protocol://machinename/enterprisename/objectmanager/servername	該当なし
ConnectString (Siebel バージョン 7.5 用)	新しい Siebel アーキテクチャでは、Siebel Java Data Bean は接続ストリングを使用して Siebel Object Manager に接続することができます。Siebel Java Data Bean が構成ファイルを認識する必要はなく、単にファイルを読み取り、Object Manager に接続します。	protocol://machinename/enterprisename/objectmanager_languageCode/ servername	該当なし
ApplicationUserName	Siebel アプリケーションのユーザー・アカウント		CWCONN
ApplicationPassword	Siebel ユーザー・アカウントのパスワード		CWCONN

表 5. コネクター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
ConnectErrors	Siebel から戻され、コネクター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、コネクターは終了します。	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ーを処理するように構成されている場合に使用されます。	コネクターを示す整数値。	
ViewMode	後方互換性を保つために維持されています。ビジネス・コンポーネントに対して指定された VM asi がない場合にこの値が使用されます。	整数値。詳細については、VM asi を参照してください。	
DataBeanRefreshInterval	この値は、コネクターが 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。コネクターは、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ーで処理される要求数に対応する DataBeanRefreshInterval を示す整数値。	
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。	
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	
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

表 5. 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
SupportNameValuePairs	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
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	;
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
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
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

## 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ー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、コネクターはアーカイブ処理を行いません。`ArchiveProcessed` が `false` に設定されると、コネクター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、コネクターは `Event` ビジネス・コンポーネントからイベントを削除します。
- コネクター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、コネクターはイベントを `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、コネクター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、コネクター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、コネクターは値を `true` と見なします。

`ArchiveTableName` プロパティにも値がない場合、コネクターはアーカイブ表の名前が `xworlds_archive_events` であると見なします。

デフォルト値は `true` です。

## ConnectorId

コネクターの固有の ID。この ID は、コネクター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。

## 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。許容値は 1 から 500 です。

デフォルトは、1 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。

## 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。

## 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。

---

## 複数のコネクター・インスタンスの作成

コネクターの複数のインスタンスを作成する作業は、いろいろな意味で、カスタム・コネクターの作成と同じです。

**注:** WebSphere Business Integration Server Express に付属するアダプターの追加インスタンスは、配置できるアダプターの総数を制限する機能では、それぞれ別個のアダプターとして取り扱われます。

以下に示すステップを実行することによって、コネクターの複数のインスタンスを作成して実行するように、ご使用のシステムを設定することができます。次のようにする必要があります。

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

## 新規ディレクトリーの作成

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

```
ProductDir¥connectors¥connectorInstance
```

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

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

```
ProductDir¥repository¥connectorInstance
```

### ビジネス・オブジェクト定義の作成

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

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

```
ProductDir¥repository¥initialConnectorInstance
```

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

### コネクタ定義の作成

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

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

### 始動スクリプトの作成

始動スクリプトは以下のように作成します。

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

```
dirname
```

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

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

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

---

## コネクターの始動

コネクターは、**コネクター始動スクリプト**を使用して明示的に始動する必要があります。始動スクリプトは、次に示すようなコネクターのランタイム・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* はコネクターを示します。始動スクリプトの名前は、表 6 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 6. コネクターの始動スクリプト

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

コネクター始動スクリプトは、以下に示すいずれかの方法で起動することができます。

- Windows システムで「スタート」メニューから。

「プログラム」>「IBM WebSphere Business Integration Express」>「アダプター」>「コネクター」>「ご使用のコネクター名 (*your\_connector\_name*)」を選択します。

デフォルトでは、プログラム名は「IBM WebSphere Business Integration 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、ご使用のコネクターへのデスクトップ・ショートカットを作成することもできます。

- コマンド行から。

– Windows システム:

```
start_connName connName ICS [-cconfigFile ]
```

ここで、*connName* はコネクターの名前です。

- System Monitor から。

このツールを使用して、コネクターのロード、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ー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、コネクターが始動します。

コマンド行の始動オプションなどのコネクター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。

---

## コネクターの停止

コネクターを停止する方法は、以下に示すように、コネクター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ーを始動した場合は、コネクター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、そのコネクター用の別個の「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、コネクターが停止します。
- System Monitor を使用します。

このツールを使用して、コネクターのロード、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ー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、コネクターは停止します。

---

## 第 3 章 ビジネス・オブジェクトの理解

この章では、コネクターがビジネス・オブジェクトを処理する方法について説明し、ビジネス・オブジェクトの変更に関する提案を示します。

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。

以下のトピックで構成されています。

- 『ビジネス・オブジェクトの構造および関係』
- 30 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』
- 40 ページの『ビジネス・オブジェクトのサンプルの使用』

**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

コネクターは、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。コネクター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ー・ロジックはありません。例えば、コネクターは特定のビジネス・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。コネクターは、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づけることができます。

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。

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。

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 7. 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。

表7. 単純属性 (続き)

名前	属性の名前
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。
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8. 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。

表 8. 子オブジェクト属性 (続き)

名前	子オブジェクトの名前
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_B0InternalProduct
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

ーメ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 を持っているはなりません。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ったら、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。

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ス命令です。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

子ビジネス・オブジェクトを表すコンテナ属性のアプリケーション固有の情報では、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、WebSphere Business Integration Server Express Plus システムのビジネス・オブジェクト Siebel\_BCContact 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 パラメータに対してパスが無効であれば、To パラメータはヌルに設定される。To パラメータ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。In WebSphere Business Integration Server Express Plus 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_BCOppportunity
Type = Siebel_BCOppportunity
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。コネクターは、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、コネクター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ーへのビジネス・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。

## 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 eBusiness Applications は、ラッパー・ビジネス・オブジェクトをサポートします。27 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。

## 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では、コネクター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。オブジェクトの一部を検索するためのアプリケーション固有の情報は max=n です。5 つのオブジェクトだけを指定する Retrieve 動詞の例を以下に示します。

```
[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ーは使用せず、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。

## 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。

---

## ビジネス・オブジェクトのサンプルの使用

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。以下のオブジェクトは、これらの作業の実行方法を示すサンプルです。

- 『Siebel\_BCAccount』
- 41 ページの 『Siebel\_BCQuote』
- 42 ページの 『Siebel\_BCContact』
- 42 ページの 『Siebel\_BCInternalProduct』
- 43 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、および Business\_Address\_Write.svb

### Siebel 7

Account.sbl または 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 をピックアップして 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 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 をピックアップして 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. 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、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 プロシージャーに 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. 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、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 プロシージャー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 プロシージャーに 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、WebSphere Business Integration Server Express Plus 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 Server Express Plus 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。

---

## 第 4 章 SiebelODA を使用したビジネス・オブジェクト定義の生成

この章では、SiebelODA について説明します。これはオブジェクト・ディスカバリー・エージェント (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。

以下のトピックで構成されています。

- 『インストールと使用法』
- 48 ページの『Business Object Designer Express 内での SiebelODA の使用』
- 60 ページの『生成される定義の内容』
- 63 ページの『ビジネス・オブジェクト定義への情報の追加』

---

### インストールと使用法

ここでは、以下のことについて説明します。

- 『SiebelODA のインストール』
- 46 ページの『SiebelODA を使用する前に』
- 46 ページの『SiebelODA の起動』
- 47 ページの『複数のマシンにおける SiebelODA の実行』
- 47 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、WebSphere Business Integration Server Express Adapter インストーラーを使用します。「*WebSphere Business Integration Server Express インストール・ガイド*」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.txt
- ODA¥Siebel¥start\_SiebelODA.bat
- bin¥CWODAEEnv.bat

**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 WebSphere Business Integration Server Express Plus 製品のパス名は、使用システムで製品がインストールされたディレクトリーを基準と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.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 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。

## SiebelODA の起動

この起動スクリプトを使用して SiebelODA を起動できます。

### 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、メニュー「プログラム」>「IBM WebSphere Business Integration Express」>「アダプター」>「Object Discovery Agents」に始動用ショートカットがあります。

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・メッセージおよびトレース・メッセージのファイル (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方法は 2 つあります。

- 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 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 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、50 ページの表 10 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 9 で、これらの値を説明します。

表 9. 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"><li>• ODA のすべてのプロパティの初期値を示す</li><li>• 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</li><li>• ビジネス・オブジェクト定義ダンプをトレースする</li></ul>

これらの値を設定する場所については、50 ページの表 10 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。49 ページの図 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. 49 ページの『ODA の選択』
2. 49 ページの『構成プロパティの指定』
3. 51 ページの『ソースの選択』
4. 56 ページの『オブジェクトの選択の確認』
5. 56 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、57 ページの『追加情報の入力』
6. 60 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。

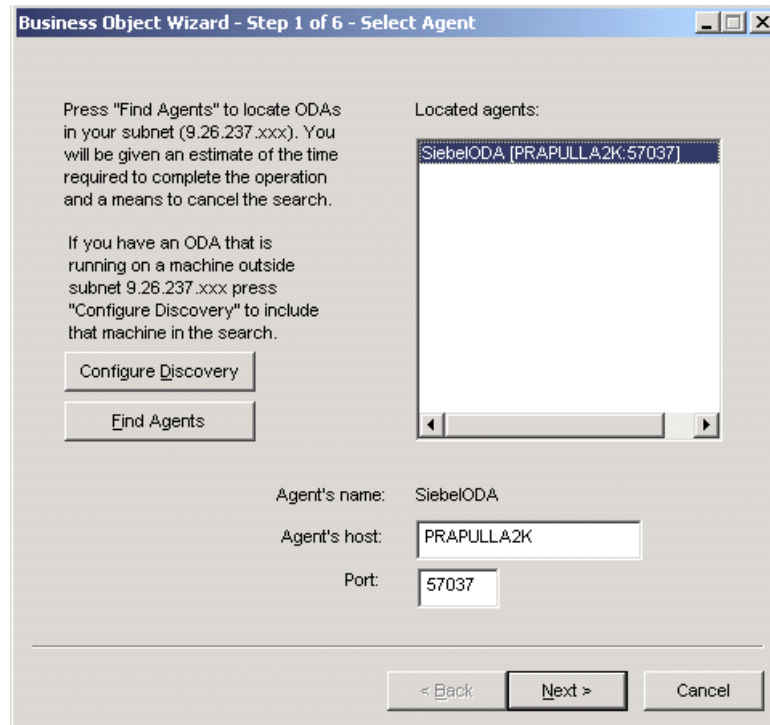


図 2. ビジネス・オブジェクト・ウィザードの「エージェントの選択」画面

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。

2. 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。

3. 「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。

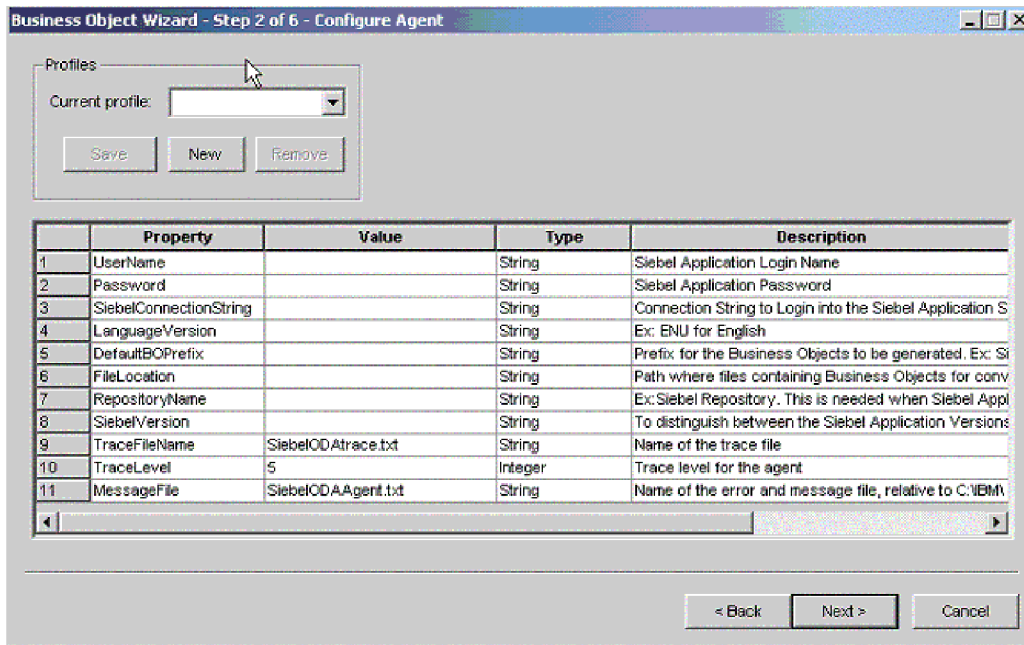


図3. ビジネス・オブジェクト・ウィザードの「エージェントの構成」画面

表 10 に示された SiebelODA プロパティを構成します。

表 10. 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
3	SiebelConnectionString	String	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。  例:  Siebel 7.x の場合: //machinename/enterprisename/objectmanager/servername  Siebel 7.5 の場合: //machinename/enterprisename/objectmanager_ languageCode/servername
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
5	DefaultBOPrefix	String	す。 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。

表 10. 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
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 の場合は設定しません。
9	TraceFileName	String	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。  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。  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。
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、47 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。
11	MessageFile	String	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。  <b>重要:</b> 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。  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。

**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 (52 ページの図 4 を参照)。

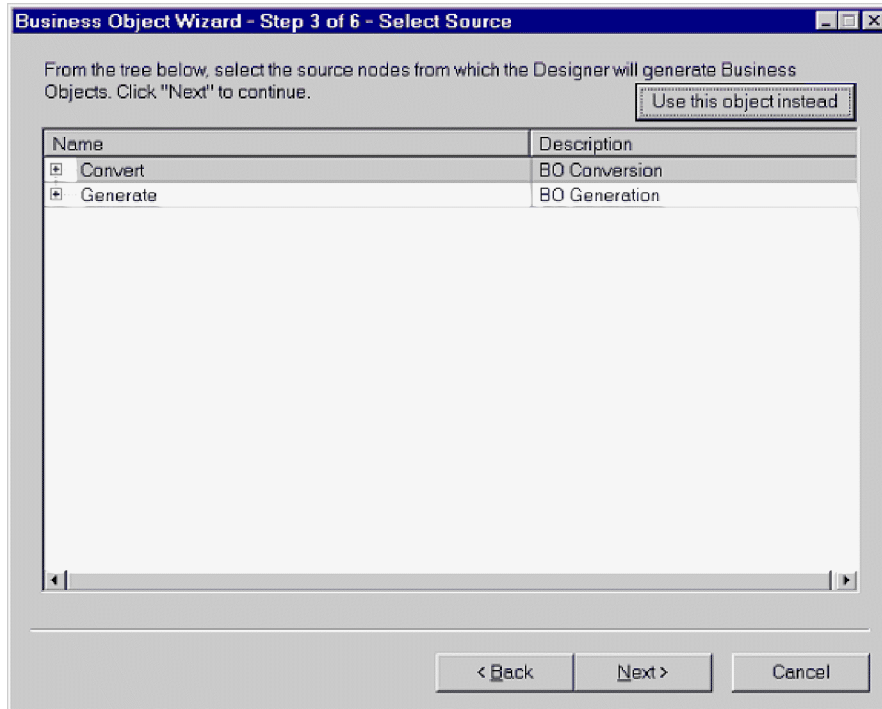


図4. ビジネス・オブジェクト・ウィザードの「ソースの選択」画面

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 (53 ページの図5 を参照)。

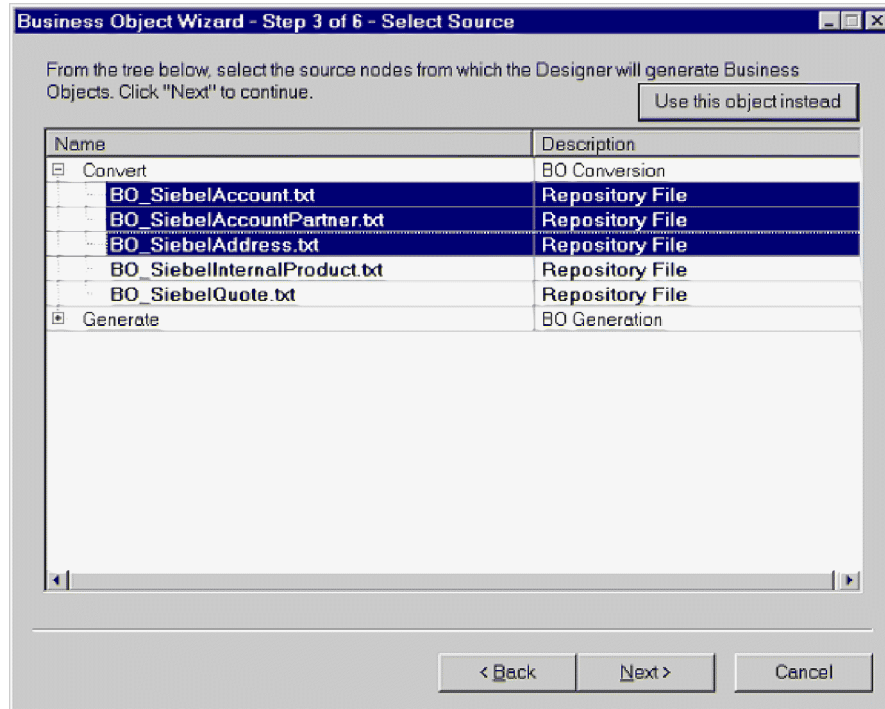


図5. ビジネス・オブジェクト・ウィザードで変換対象のビジネス・オブジェクトを表示している画面

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、54 ページの図 6、54 ページの図 7、および 55 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 Objects)」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。

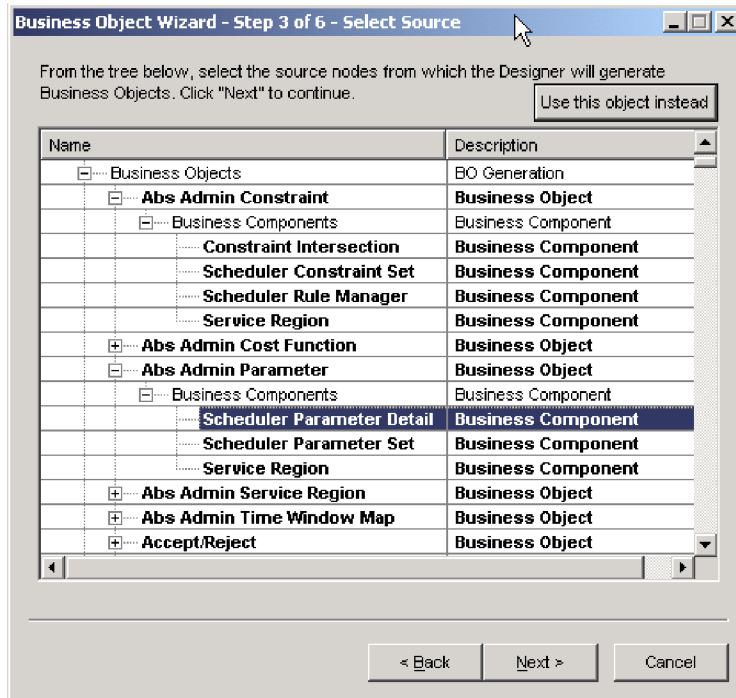


図6. ビジネス・オブジェクト・ウィザードで「ビジネス・オブジェクト (Business Objects)」を展開した状態

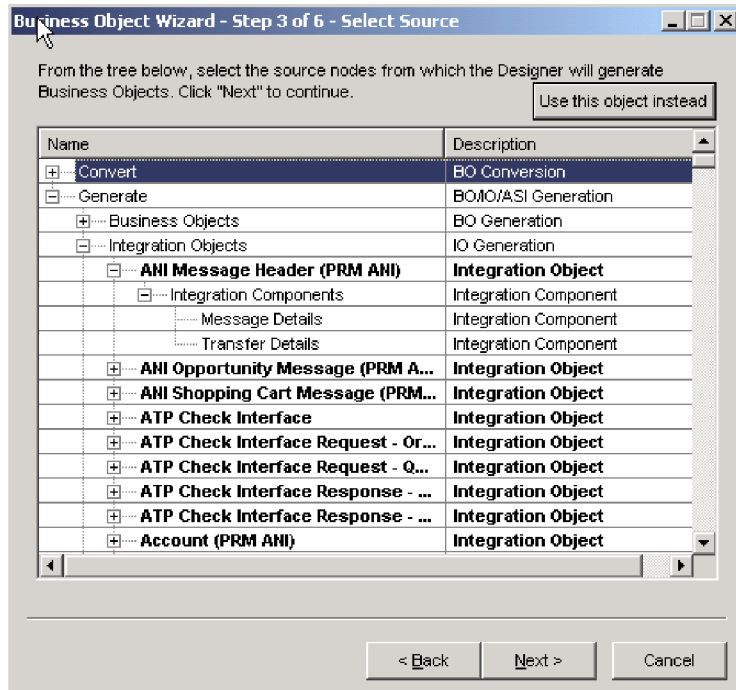


図7. ビジネス・オブジェクト・ウィザードで「統合オブジェクト (Integration Objects)」を展開した状態



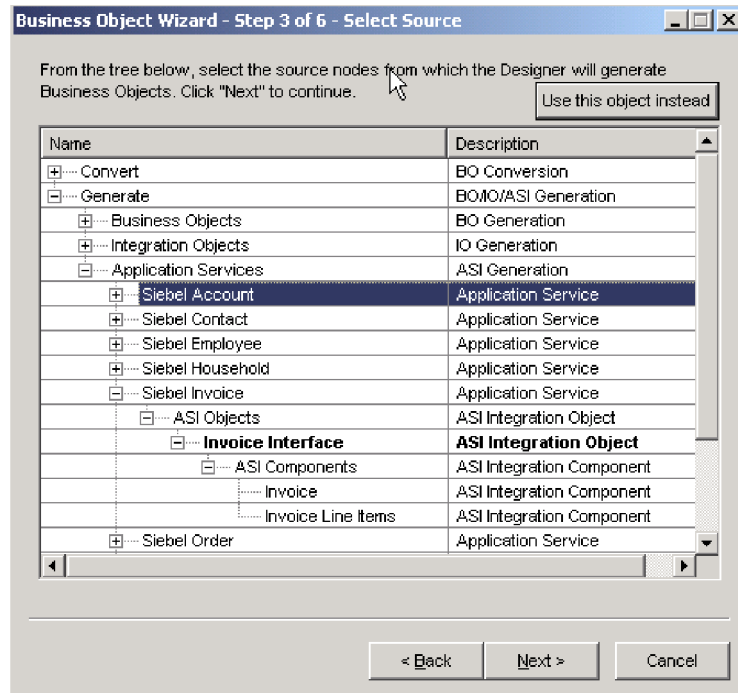


図8. ビジネス・オブジェクト・ウィザードで「アプリケーション・サービス (Application Services)」を展開した状態

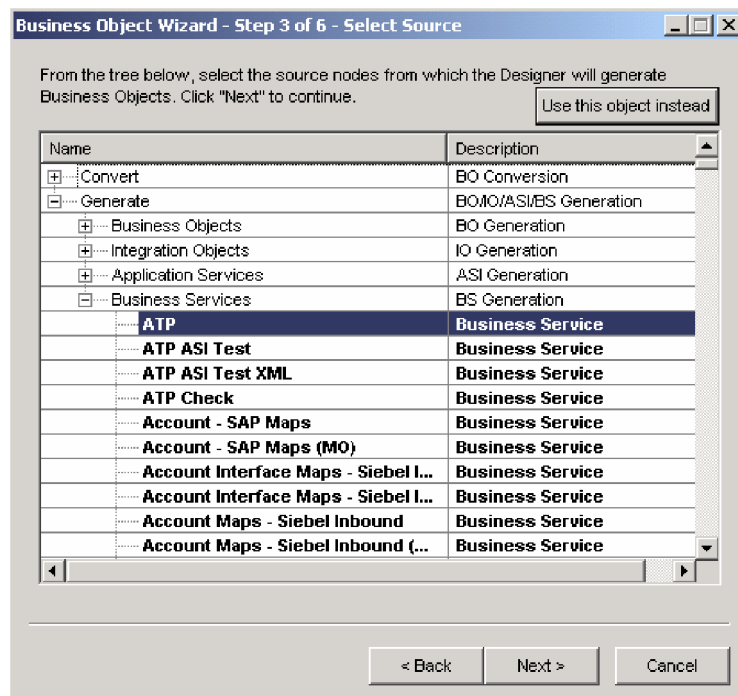


図9. ビジネス・オブジェクト・ウィザードで「ビジネス・サービス (Business Services)」を展開した状態

## 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。

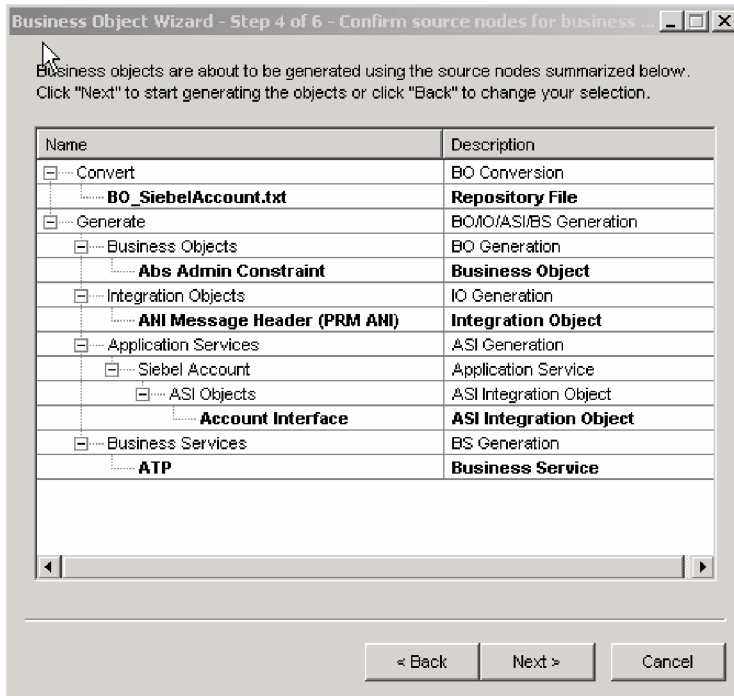


図 10. ビジネス・オブジェクト・ウィザードでのオブジェクトおよびコンポーネントの選択の確認

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。

57 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。

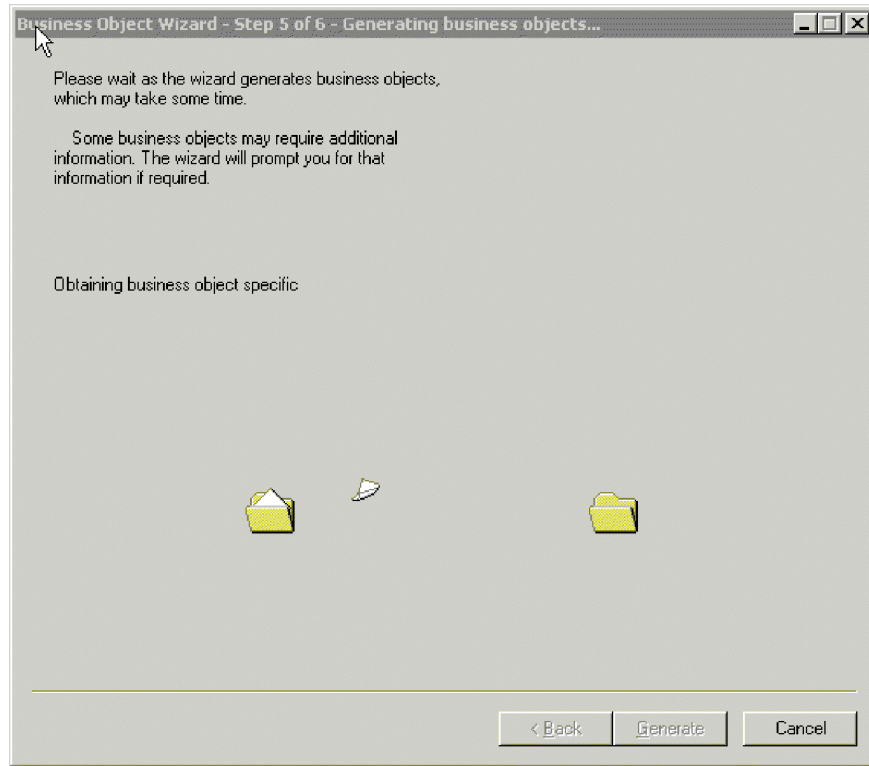


図 11. ビジネス・オブジェクト定義の生成

## 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。58 ページの図 12 (および図 12 と 13) は、これらの画面を示しています。

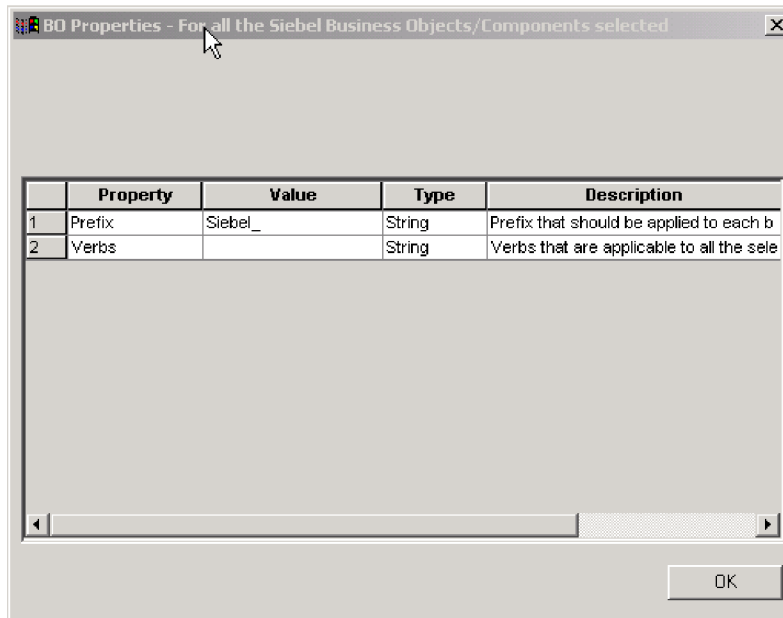


図 12. ビジネス・オブジェクトの追加情報の入力

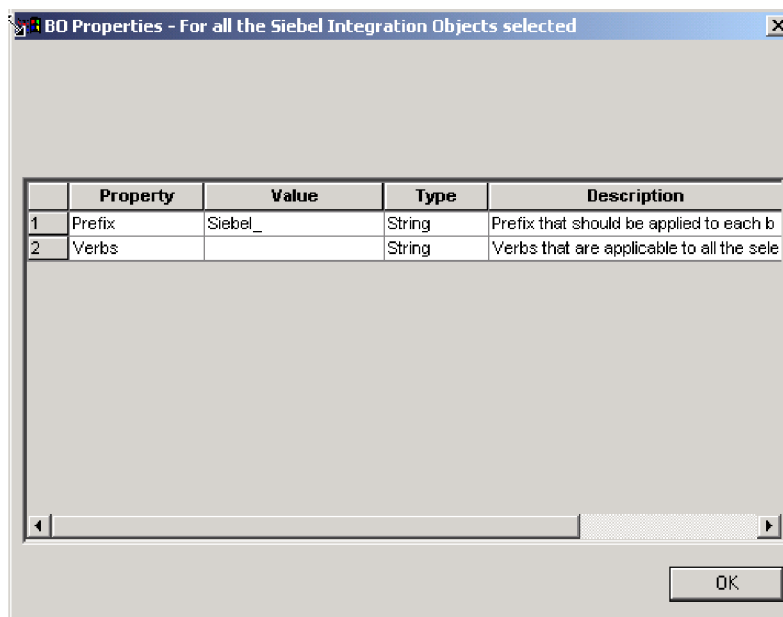


図 13. 統合ビジネス・オブジェクトの追加情報の入力

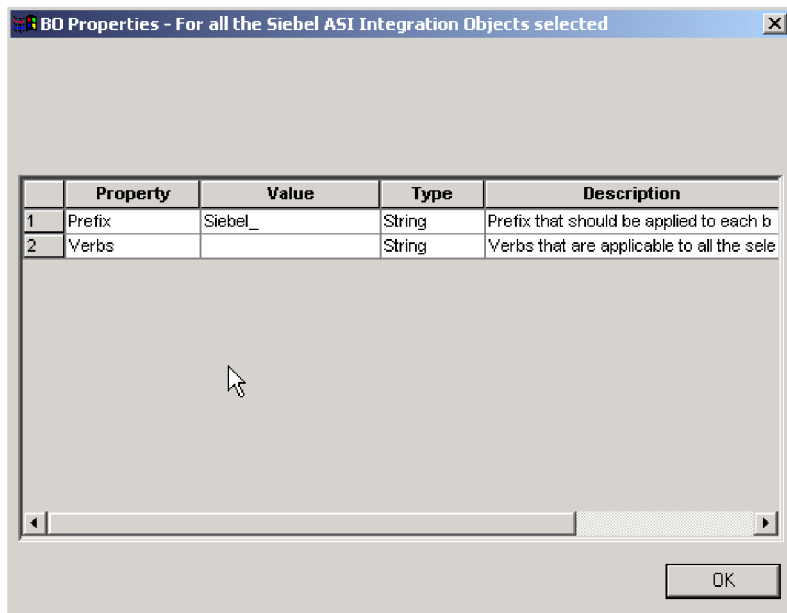


図 14. アプリケーション・サービスの追加情報の入力

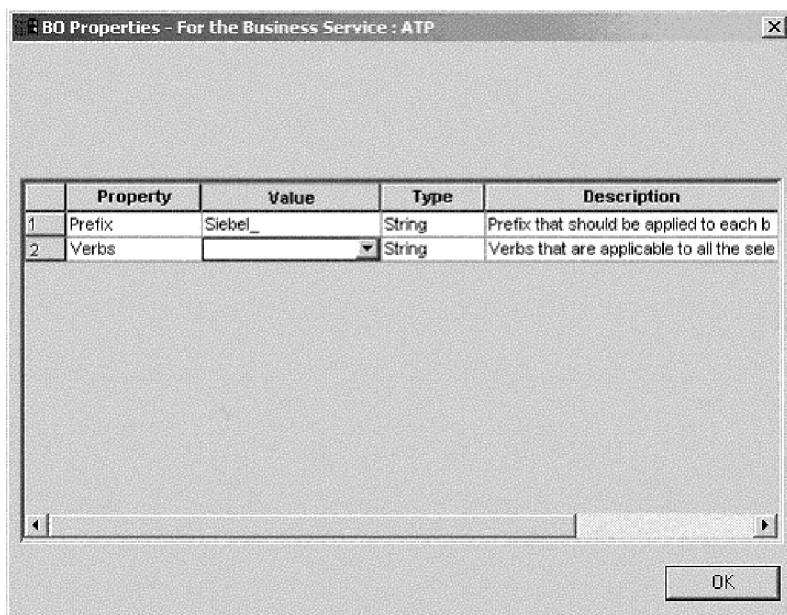


図 15. カスタム仕様で作成されたビジネス・サービスの追加情報の入力

「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します。
- 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。

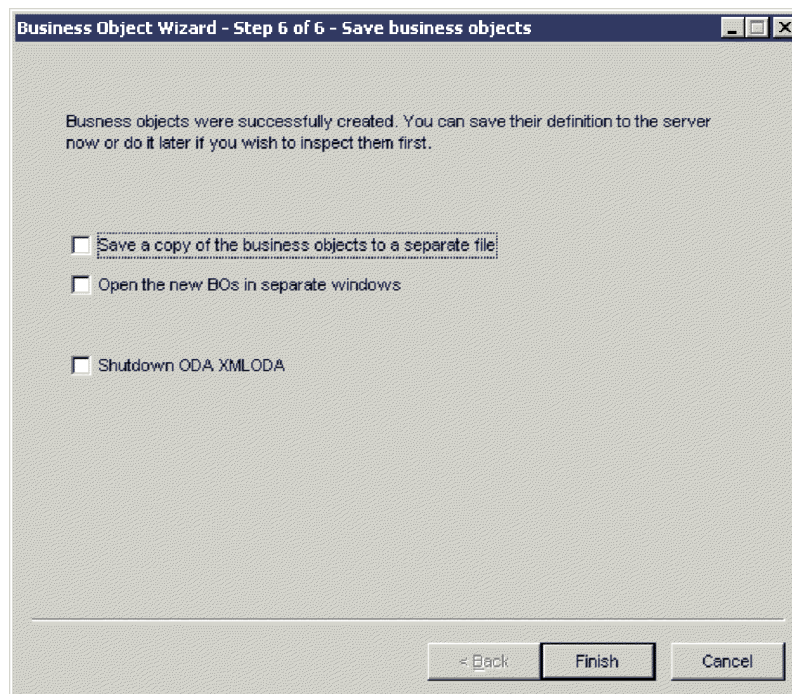


図 16. ビジネス・オブジェクトの定義の保管

---

## 生成される定義の内容

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、アプリケーション固有の情報が含まれるかどうかを確認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。

- 『ビジネス・オブジェクト・レベルのプロパティ』
- 『属性プロパティ』
- 63 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

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

ここでは、SiebelODA が属性ごとに生成するプロパティ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

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

属性の型を設定する際、表 11 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 11. 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
DTYPE_BOOL	BOOLEAN	

表 11. データ型の対応関係 (続き)

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 11 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ーを 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ー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ーを 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。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もできます。

**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 5 章 Siebel 接続 DLL の使用

Siebel Enterprise Application Integration (EAI) フレームワークは、Siebel アプリケーションが外部アプリケーションとデータ交換をできるようにするトランスポートを提供します。これを行うために、Siebel EAI ダイナミック・リンク・ライブラリー (DLL) トランスポートは、InterChange Server Express 統合ブローカーが提供するアクセス・フレームワーク・インターフェースを起動する WebSphere Integration Adapter (WIA) Siebel 接続 DLL を起動します。WebSphere Integration Adapter Suite は、さまざまなアプリケーション向けのアダプターから構成されているため、InterChange Server Express および適切な WebSphere Business Integration Server Express Plus アダプターは、DLL によって送信される Siebel 生成要求を処理できます。DLL は単一方向に動作します。Siebel からブローカー (InterChange Server Express) に送信される要求は処理されますが、その逆の要求は処理されません。

本章には、以下のトピックが記載されています。

- 『コネクター・アーキテクチャーおよびプロセス』
- 67 ページの『Siebel 接続 DLL の構成』
- 70 ページの『Siebel 接続 DLL との通信』
- 70 ページの『IBM ビジネス・オブジェクトの生成』
- 71 ページの『IBM 汎用ビジネス・サービスの構成』

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。

---

### コネクター・アーキテクチャーおよびプロセス

Siebel EAI のコンテキストでは、データは統合メッセージ形式でトランスポートされます。このコンテキストでは、メッセージは通常、メッセージ・タイプと構造を示すヘッダー・データと、注文や会計、従業員レコードなどのデータの 1 つ以上のインスタンスからなる本文から構成されています。

以下の図は、Siebel のコネクターおよびアーキテクチャーのプロセスを示しています。

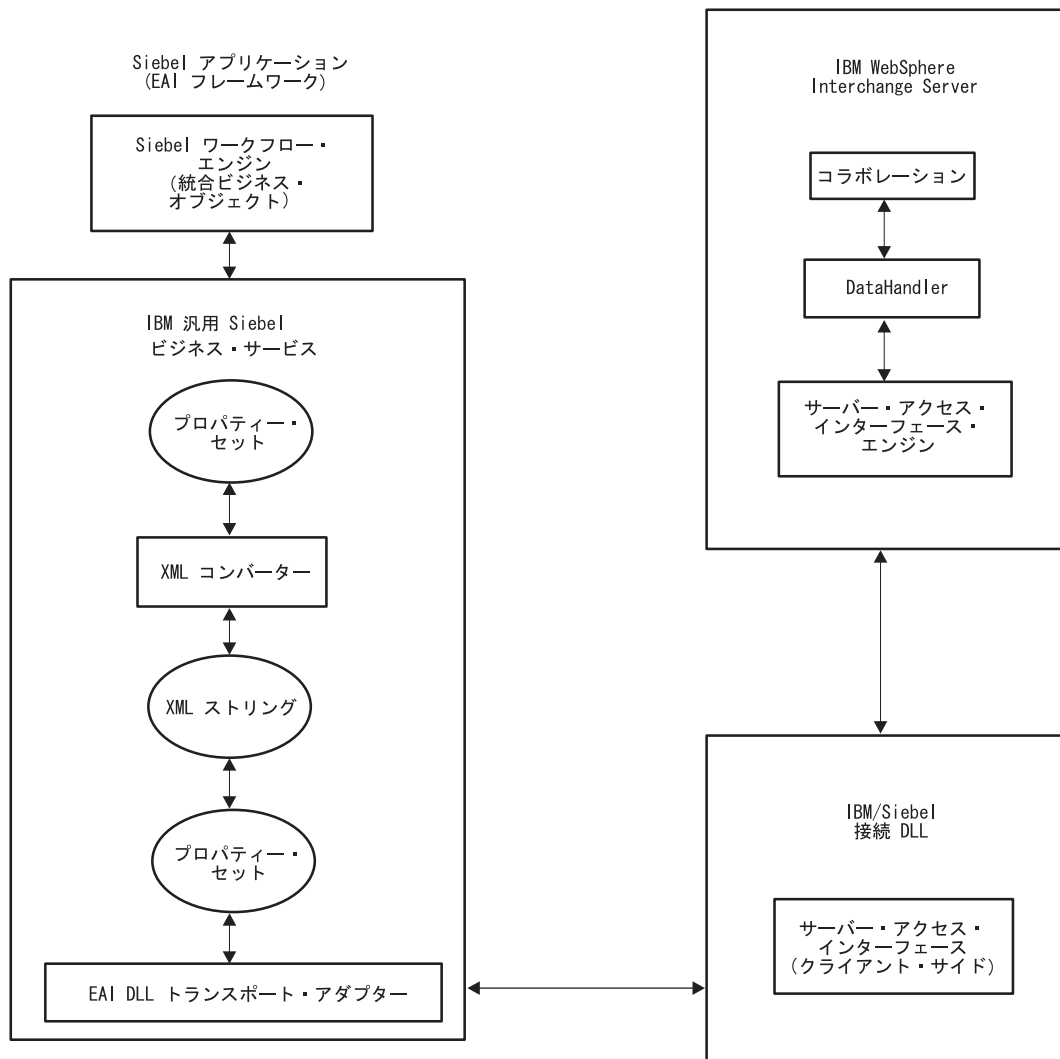


図 17. Siebel コネクターおよびアーキテクチャーのプロセス

IBM 汎用 Siebel ビジネス・サービスは、Siebel 接続 DLL ソリューションの一部として提供されています。ビジネス・サービスは、データの移動タスクやデータ・フォーマットの変換タスク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オブジェクトです。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ます。また、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ず、オブジェクトを操作して働きかけ、特有の目的を果たします。

IBM 汎用ビジネス・サービスは、ワークフローや他のビジネス・サービスによって起動することができます。汎用ビジネス・サービスは、プロパティー・セットの形式で要求を受信します。XML コンバーターは、データをプロパティー・セットから XML ストリングに変換します。変換の際に、DataHandler ではサポートされないストリングのタグは除去されます。タグが除去された XML ストリングは、プロパティー・セットとして EAI DLL トランスポートに送信され、受信メソッドまたは送信メソッドを起動します。EAI DLL トランスポート・アダプターは、IBM Siebel 接続 DLL との間で統合メッセージの送受信を行います。

EAI DLL トランSPORT・アダプターは以下を行います。

- Siebel アプリケーションが、同期通信プロトコルの標準テクノロジーを使用して、外部アプリケーションとのデータのやり取りできるようにする
- アウトバウンド・メッセージのみをサポートする
- IBM Siebel 接続 DLL を起動する
- 同期処理を行い、メッセージを送信した後、処理を続行する前に返答を待機する

コラボレーションを実行するために、IBM Siebel 接続 DLL はサーバー・アクセス・インターフェースを呼び出します。コラボレーションは、複数のアプリケーションを表すことができるビジネス・プロセスを表します。Siebel からの XML 要求は、XML DataHandler を使用してビジネス・オブジェクトへと変換されます。

InterChange Server Express は、サーバー・アクセス・インターフェースを介して XML DataHandler を起動します。このインスタンスでは、DataHandler によって Siebel からの XML 入力が入用ビジネス・オブジェクトへと変換されます。その結果生成されたメッセージは、InterChange Server Express によって Siebel 接続 DLL へと戻されます。接続 DLL は、メッセージを IBM Siebel 汎用ビジネス・サービスへと渡します。汎用ビジネス・サービスでは、すでに除去された Siebel 固有のタグがメッセージに再び付加され、XML コンバーター・サービスを使用して、元の Siebel プロパティ・セットに変換されます。メッセージは汎用 Siebel ビジネス・サービスの出力プロパティ・セットに設定されます。この時点で、Siebel ワークフローはメッセージを取得します。

---

## Siebel 接続 DLL の構成

WebSphere Business Integration Server Express Plus Siebel 接続 DLL は、製品の Siebel スイートの一部としてインストールされています。WebSphere Business Integration Server Express Plus CD インストール・プログラムを実行している場合は、Siebel を選択してインストールします。インストールによって、ご使用の Siebel のバージョンに応じて Siebel 接続 DLL が以下のフォルダーに置かれます。

### Siebel 7.0.x

```
ProductDir¥connectors¥Siebel¥SiebelEAIAdapter¥Siebel7.0.x
```

### Siebel 7.5.x

```
ProductDir¥connectors¥Siebel¥SiebelEAIAdapter¥Siebel7.5.x
```

その他のコンポーネントは、以下のフォルダーにインストールされます。

```
ProductDir¥connectors¥Siebel¥SiebelEAIAdapter
```

このセクションでは、WebSphere Business Integration Server Express Plus Siebel 接続 DLL の構成およびセットアップ方法について説明します。

## IBMCrossWorlds.cfg ファイルの編集

[collabdetails] (コラボレーションの詳細) セクション (コラボレーション名、汎用 BO 名、ポート名など) を追加する場合は、IBMCrossWorlds.cfg ファイルを編集する必要があります。ファイルのコピーは

`%ProductDir%\connectors\Siebel\EAIAdapter\Samples` ディレクトリーにあります。このファイルには、表 12 にリストされている項目が含まれています。

表 12. `IBMCrossWorlds.cfg` ファイルの内容

項目	説明
SERVERNAME	使用する InterChange Server の名前。
USERNAME	Siebel アプリケーションのユーザー名。
PASSWORD	Siebel アプリケーションのパスワード。
[CollabDetails]	<p>使用している各コラボレーションの詳細。例えば、アカウント・オブジェクトには、以下が含まれます。</p> <pre>[CollabDetails] IntObjName=Account BOName=NewSiebel_Account CollabName=SiebelConnectivityObject PortName=From [EndCollabDetails]</pre>

## ファイルのコピー

以下のファイルをインストール・フォルダーから `%Siebel%/Client/bin` ディレクトリーにコピーします。

- SiebelConnectivity.DLL
- IBMCrossWorlds.cfg
- SiebelConnectivityMessages.txt

## 永続 .ior ファイルの生成

InterChange Server Express を起動すると、`.ior` ファイルが生成されます。これは InterChange Server Express のアクセス・エンジン・オブジェクトへの永続的なオブジェクト参照です。ファイル名は `[Server Name] InterChangeServer.ior` となり、`%WBIA%` にあります。このファイルを `%Siebel%\Client\bin` ディレクトリーにコピーします。InterChange Server Express が開始するたびに、新たな `.ior` ファイルが生成されます。この `.ior` ファイルの内容を、InterChange Server をブートするたびに同じ状態に保つために、永続 `.ior` ファイルが生成されます。サーバーに永続 `.ior` ファイルを生成させるには、`InterchangeSystem.cfg` ファイルを編集して、CORBA のサブセクションがない場合には追加します。

CORBA サブセクションには、ポート番号が `OApport=portnumber` 形式で含まれています。例えば、ポート番号が 15000 の場合、以下の行を `InterchangeSystem.cfg` ファイルに追加します。

```
[CORBA]USERNAME=Siebel Application user name
OApport=15000
```

アクセス・クライアントが DMZ にあり、InterChange Server Express が別のサブネットにある場合は、`OApport` のポート番号が開かれていることを確認します。

**注:** システム・ガイドの詳細については、`[Server Name] InterChangeServer.ior` ファイルを、`%Siebel%\Client\bin` ディレクトリーにコピーします。

## IBM ビジネス・オブジェクトの生成とコピー

XMLborgen または XMLODA を使用して、各統合オブジェクトで WebSphere Business Integration Server Express Plus ビジネス・オブジェクトの定義を生成します。生成されたオブジェクトは、ワークフロー処理で使用されます。生成されたビジネス・オブジェクトは、InterChange Server Express リポジトリに自動的にロードされます。

## XML DataHandler の構成

XML DataHandler はデータ変換モジュールです。このモジュールは、主にビジネス・オブジェクトの XML 形式への変換 (またはその逆) を行います。XML 文書は text/xml MIME タイプで直列化されたデータです。この XML 資料はサーバー・アクセス・インターフェース内から呼び出されます。XML DataHandler を構成するには、構成情報を XML 子メタオブジェクトに提供する必要があります。XML DataHandler では、WebSphere Business Integration Server Express Plus システムにより、デフォルトのメタオブジェクト MO\_DataHandler\_DefaultXMLConfig が提供されています。表 13 に、構成の必要があるメタオブジェクトに含まれている属性を示します。

表 13. メタオブジェクトに含まれる属性

属性	説明
BOPrefix	適切な接頭部を持つ WebSphere Business Integration Server Express Plus ビジネス・オブジェクト名が、構成ファイルに指定されています。したがって、この属性はブランクのままとします。
ClassName	ClassName 属性は、デフォルトでは、サーバー・アクセス・インターフェースによって起動される XML DataHandler クラスを指しています。デフォルト値は com.crossworlds.DataHandlers.text.xml です。

## コラボレーション・テンプレートの定義

コラボレーション・テンプレートを定義して、ビジネス処理用のコラボレーション・オブジェクトを作成します。コラボレーション・テンプレートの定義およびコラボレーション・オブジェクトの作成についての詳細は、「コラボレーション開発ガイド」を参照してください。

## ビジネス・サービスのインポート

ビジネス・サービスをインポートするには、以下を行います。

1. Siebel クライアントを起動します。
2. 「表示 (View)」 → 「サイト・マップ (Site Map)」 → 「ビジネス・サービス管理 (Business Service Administration)」 → 「ビジネス・サービス・スクリプト (Business Service Scripts)」を選択します。
3. IBMSiebelConnectivityService.xml ファイルからビジネス・サービスをインポートします。このファイルは、  
`%ProductDir%connectors¥Siebel¥SiebelEAIAdapter¥dependencies` フォルダにあります。

## Siebel ワークフローの構成

Siebel 資料で説明されているワークフロー・プロセスのセットアップに従い、ビジネス・サービス・トリガーを追加します。このトリガーは、*IBMSiebelConnectivityService* を呼び出して活動させます。メソッド名は、実行の必要があるプロセスに応じてワークフローによって設定する必要があります。

---

## Siebel 接続 DLL との通信

Siebel 接続 DLL はサーバー・アクセス・クライアントであるため、CORBA を使用して InterChange Server Express との通信を行います。アクセス・クライアントが InterChange Server Express インスタンスを確実に見付けるためには、その InterChange Server Express インスタンスの相互運用オブジェクト参照ファイルへのアクセスが用意されている必要があります。アクセスの詳細については、9 ページの『ユーザーのセットアップ』を参照してください。

---

## IBM ビジネス・オブジェクトの生成

Siebel 統合オブジェクトは、Siebel ビジネス・オブジェクトの統合メタデータを表します。XML は、Siebel EAI インフラストラクチャーが認識する共通の構造です。これらの統合オブジェクトは一連の構造規則を順守するため、Siebel eScript オブジェクト、メソッド、および関数を使用して、プログラマチックに全探索や変換を行うことができます。あるいは、Siebel Data Mapper を用いて、宣言によって変換することができます。詳細については、「*Siebel eBusiness Application Integration, volume IV*」を参照してください。

Siebel ツールには、統合オブジェクト用のスキーマまたは DTD を生成する機能が用意されています。生成された DTD は、WebSphere Business Integration Server Express Plus ユーティリティ、XMLborgen または XMLODA 用の入力となります。WebSphere Business Integration Server Express Plus システムのビジネス・オブジェクトは XMLborgen および XMLODA DTD から生成されます。XMLborgen の実行中に、トップレベルのオブジェクトの名前に、ストリング *ListOf* が接頭部として付加されている名前を指定します。

例えば、DTD の *ListOfChargesData* は、ファイル *ListofChargerData.dtd* にあります。このファイルは、ディレクトリー

`%ProductDir%connectors/Siebel/Siebel/SiebelEAIAdapter/Samples` にあります。

**注:** スキーマまたは DTD の生成についての詳細は、「*Siebel Integration Platform Technologies*」を参照してください。

XMLborgen によって生成されたビジネス・オブジェクトは、ディレクトリー `%ProductDir%connectors/Siebel/SiebelEAIAdapter/Samples` にあります。

**注:** DTD を生成する際には、Siebel メッセージ・エンベロープは使用しないことに注意してください。ただし、これには「エンベロープなし (*No Envelope*)」オプションを選択する必要があります。実際のデータを保持するプロパティ・セットを渡す際には、Siebel メッセージ・タグ `<SiebelMessage>` と `</SiebelMessage>`、および `<?Siebel-Property-Set...>` タグとともにプロパティ・セットを送信する必要があります。



## IBM 汎用ビジネス・サービスの構成

IBM は、汎用 Siebel ビジネス・サービス IBMSiebelConnectivityService を、WebSphere Business Integration Server Express Plus の Adapter for Siebel eBusiness Applications コンポーネントの一部として提供します。このビジネス・サービスは、Siebel ワークフロー・プロセスに含まれている必要があります。Siebel ワークフローは、統合オブジェクト・インスタンス (データ) を含む Siebel プロパティ・セットを IBMSiebelConnectivityService に渡します。このプロパティ・セットには、<SiebelMessage> タグが含まれます。Siebel 統合オブジェクト名はこのタグから取得されます。また、IBMSiebelConnectivityService は、Siebel ビジネス・サービスのインスタンスである XML コンバーターを使用して、プロパティ・セットを XML に変換します。変換した XML ストリングから統合オブジェクト名を取得すると、<SiebelMessage> および </SiebelMessage> の 2 つのタグが除去されます。これらのタグは、WebSphere Business Integration Server Express Plus Data Handler ではサポートされていません。

Siebel 接続 DLL は、以下の各動詞につき 1 つのメソッドをサポートしています。

- Create
- Retrieve
- Update
- Delete

IBMSiebelConnectivityService は、Service\_PreInvokeMethod の MethodName というパラメータを取ります。この MethodName は、Siebel 接続 DLL のメソッドにマップされます。表 14 に、MethodName パラメータと DLL がサポートする動詞との間のマッピングを示します。

表 14. Siebel 接続 DLL がサポートするマッピング

メソッド名	動詞
Insert	Create
Update	Update
Retrieve	Retrieve
Delete	Delete

MethodName の値が汎用ビジネス・サービスによって検査されると、Siebel 要求は以下のように処理されます。

- 編集された XML ストリングが、別のプロパティ・セットの一部として設定される。
- EAI DLL トランスポート・ビジネス・サービスが起動され、以下のパラメータが設定される。
  - MethodName が ExternalFunction プロパティの値に設定される。
  - SiebelConnectivity DLL の名前が、同じサービスの DLLName プロパティに設定される。
- その後、EAI DLL トランスポート・サービスの SendReceive メソッドが呼び出され、プロパティ・セットが入力として渡される。入力プロパティ・セットが、XML ストリング形式で Siebel 接続 DLL に渡されます。次に、Siebel 接続

DLL は、サーバー・アクセス・インターフェース呼び出しを介して、必要なオペレーションを実行します。その後、XML ストリング形式の結果を汎用 Siebel ビジネス・サービスに戻します。

汎用 Siebel ビジネス・サービスでは、Siebel 固有タグが、接続 DLL から戻される XML 結果ストリングに戻されて追加されます。編集された XML ストリングは、再び XML コンバーター・サービスのインスタンスを使用して、プロパティ・セットへ再変換されます。その結果は、IBMSiebelConnectivityService の出力プロパティ・セットのパラメーターの一部として保管されます。Siebel ワークフローは、IBMSiebelConnectivityService から同じものを取得して、必要なオペレーションを実行します。

---

## 第 6 章 Siebel ビジネス・サービスのサポート

この章では、Siebel ビジネス・サービスの使用方法について説明します。この章のトピックは以下のとおりです。

- 『ビジネス・サービスの理解』
- 75 ページの『ビジネス・サービスにおける動詞処理』
- 77 ページの『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』

74 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

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オブジェクトです。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 Applications Services Interfaces (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、WebSphere Business Integration Server Express Plus 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、WebSphere 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、Adapter for Siebel eBusiness Applications は、統合オブジェクトを WebSphere 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。同様にこのアダプターは、統合オブジェクトを WebSphere ビジネス・オブジェクトによって表して実装することで、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、WebSphere Business Integration Server Express Plus 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。

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、WebSphere 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。45 ページの『第 4 章 SiebelODA を使用したビジネス・オブジェクト定義の生成』を参照してください。

注: オブジェクトにアプリケーション固有の情報 BSN が含まれる場合は、ビジネス・サービスとして扱われます。75 ページの表 15 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 WebSphere Business Integration Server Express Plus 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 15. 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。
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"> <li>• 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。</li> <li>• 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。</li> <li>• 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。</li> </ul>

表 16. 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 17 のすべての動詞において、戻りコードは VALCHANGE です。

表 17. 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 17 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。

WebSphere Business Integration Server Express Plus 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

カスタム・ビジネス・サービスを使用する場合、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。アプリケーション固有の情報に含まれる `Request` ビジネス・オブジェクトの単純属性は、プロパティを提供し、`SiebelMessage` コンテナ属性は、統合オブジェクトを提供します。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

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 WebSphere Business Integration Server Express Plus 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 WebSphere Business Integration Server Express Plus ビジネス・オブジェクト

```
Siebel_<IntObjectName> (Parent IntegrationComponent)
```

Field1  
Field2  
+ChildIntegrationComponent



---

## 第 7 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。

---

### エラー処理

Adapter for Siebel eBusiness Applications の Siebel 接続 DLL コンポーネントは、以下の場合にエラーを戻します。

- InterChange Server Express に接続できない。
- ログまたはトレース・ファイルに書き込もうとしてもオープンできない。
- XML 入力メッセージから WebSphere Business Integration Server Express Plus システムのビジネス・オブジェクトを作成すると、エラーが発生する。これは、着信 XML データのデータ形式が誤っていることが原因である可能性があります。要求の中でデータ・エラーが発生しても、ワークフローはバッチ要求の処理を停止しません。障害が発生した要求を見付けるには、ワークフロー・ログを調べて失敗した要求を検索する必要があります。エラーを修正したら、もう一度要求を送信します。このシナリオは、要求の処理中に InterChange Server Express で障害が発生した場合にも発生します。ユーザーはワークフロー・ログを調べて、一連の要求が失敗したかどうかを確認する必要があります。その後、ユーザーは InterChange Server Express が稼働しているか検査して、失敗した要求を再送信する必要があります。
- コラボレーションの実行中に発生したエラーの原因は、誤った動詞の設定、誤った形式の属性値、または外部アプリケーションまたは InterChange Server Express で障害が発生していることなどが考えられます。これは、コラボレーションが存在していない、あるいは稼働していない場合にも発生します。ユーザーは、エラーの原因を分析する前に、外部アプリケーションおよび InterChange Server Express が稼働していることを確認する必要があります。その後、ユーザーは DLL ログおよびトレース・ファイルを調べて、エラーの原因を突き止め、エラーを修正して、要求を再送信する必要があります。
- ログまたはトレース・ファイルに書き込もうとしてもオープンできない。
- エラーのシナリオとして、Siebel 接続 DLL が制御できないという場合があります。例えば、WebSphere Business Integration Server Express Plus システムのビジネス・オブジェクトが作成されて処理を行うためにコラボレーションに送信され、その結果が DLL に戻された際に、ワークフローを起動する Siebel アプリケーションで障害が発生している、という場合などです。ここではさらに、要求がすでに送信されており、外部アプリケーションによって処理されています。この場合、ユーザーは、アプリケーションがシャットダウンする前に、処理された要求を検査しなければなりません。また、ユーザーは外部アプリケーションでの要求処理が正常に行われたかどうかを確認する必要があります。バッチ処理において何かを再送信する必要がある場合は、それを個別の要求単位で処理してください。

最後のケースを除く上記のすべてのケースで、接続 DLL によってエラー・メッセージが XML メッセージとして戻されます。エラー・メッセージは、<Error> および </Error> タグで囲まれています。汎用ビジネスによって Siebel 固有のタグがエラー・メッセージに追加され、XML コンバーター・サービスを呼び出すことによって、このメッセージがプロパティ・セットに変換されます。汎用ビジネス・サービスからのエラー・メッセージはすべて、そのサービスの呼び出し元のワークフローによって処理される必要があります。

Siebel 接続 DLL は、SiebelClient\bin フォルダに作成された SiebelConnectivityTrace.txt ファイルに、トレースおよびエラー・メッセージを記録します。

---

## 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、コネクタが 10 分以上アイドル状態である場合、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。

---

## 付録 A. コネクタの標準構成プロパティ

この付録では、WebSphere InterChange Server Express で動作する、WebSphere Business Integration Server Express のアダプターに含まれるコネクタ・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。

コネクタによっては、一部の標準プロパティが使用されないことがあります。Connector Configurator Express から統合ブローカーを選択すると、ご使用のアダプターに対して構成する必要がある標準プロパティのリストが表示されます。

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

---

### 標準コネクタ・プロパティの構成

アダプター・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
- コネクタ固有の構成プロパティ

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

### Connector Configurator Express の使用

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 の使用方法の詳細については、付録 B 『Connector Configurator Express』を参照してください。

### プロパティ値の設定と更新

プロパティ・フィールドのデフォルトの長さは 255 文字です。

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

1. デフォルト
2. リポジトリ
3. ローカル構成ファイル
4. コマンド行

コネクタは、始動時に構成値を取得します。実行時セッション中に 1 つ以上のコネクタ・プロパティの値を変更する場合は、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。標準コネクタ・プロパティには、以下の 4 種類の更新メソッドがあります。

- **動的**  
変更を System Manager に保管すると、変更が即時に有効になります。
- **コンポーネント再始動**  
System Manager でコネクターを停止してから再始動しなければ、変更が有効になりません。アプリケーション固有コンポーネントまたは統合ブローカーを停止、再始動する必要はありません。
- **サーバー再始動**  
アプリケーション固有のコンポーネントおよび統合ブローカーを停止して再始動しなければ、変更が有効になりません。
- **エージェント再始動**  
アプリケーション固有のコンポーネントを停止して再始動しなければ、変更が有効になりません。

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

## 標準プロパティの要約

表 18 は、標準コネクター構成プロパティの早見表です。標準プロパティの依存関係は RepositoryDirectory に基づいているため、コネクターによっては使用されないプロパティがあり、使用する統合ブローカーによってプロパティの設定が異なる可能性があります。

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

表 18. 標準構成プロパティの要約

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
AdminInQueue	有効な JMS キュー名	CONNECTORNAME /ADMININQUEUE	コンポーネント再始動	Delivery Transport は JMS
AdminOutQueue	有効な JMS キュー名	CONNECTORNAME/ADMINOUTQUEUE	コンポーネント再始動	Delivery Transport は JMS
AgentConnections	1 から 4	1	コンポーネント再始動	Delivery Transport は IDL
AgentTraceLevel	0 から 5	0	動的	
ApplicationName	アプリケーション名	コネクター・アプリケーション名として指定された値	コンポーネント再始動	
BrokerType	ICS	ICS		
CharacterEncoding	ascii7、ascii8、SJIS、Cp949、GBK、Big5、Cp297、Cp273、Cp280、Cp284、Cp037、Cp437 注: これは、サポートされる値の一部です。	ascii7	コンポーネント再始動	

表 18. 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
ConcurrentEventTriggeredFlows	1 から 32,767	1	コンポーネント再始動	Repository Directory は <REMOTE>
ContainerManagedEvents	値なしまたは JMS	値なし	コンポーネント再始動	Delivery Transport は JMS
ControllerStoreAndForwardMode	true または false	true	動的	Repository Directory は <REMOTE>
ControllerTraceLevel	0 から 5	0	動的	Repository Directory は <REMOTE>
DeliveryQueue		CONNECTORNAME/DELIVERYQUEUE	コンポーネント再始動	JMS トランスポートのみ
DeliveryTransport	IDL または JMS	IDL	コンポーネント再始動	
DuplicateEventElimination	true または false	false	コンポーネント再始動	JMS トランスポートのみ: Container Managed Events は <NONE> でなければならぬ
EnableOidForFlowMonitoring	true または false	false	コンポーネント再始動	
FaultQueue		CONNECTORNAME/FAULTQUEUE	コンポーネント再始動	JMS トランスポートのみ
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory または任意の Java クラス名	CxCommon.Messaging.jms.IBMMQSeriesFactory	コンポーネント再始動	JMS トランスポートのみ
jms.MessageBrokerName	crossworlds.queue.manager	crossworlds.queue.manager	コンポーネント再始動	JMS トランスポートのみ
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	JMS トランスポートのみ
jms.Password	任意の有効なパスワード		コンポーネント再始動	JMS トランスポートのみ
jms.UserName	任意の有効な名前		コンポーネント再始動	JMS トランスポートのみ
JvmMaxHeapSize	ヒープ・サイズ (メガバイト単位)	128m	コンポーネント再始動	Repository Directory は <REMOTE>

表 18. 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
JvmMaxNativeStackSize	スタックのサイズ (キロバイト単位)	128k	コンポーネント再始動	Repository Directory は <REMOTE>
JvmMinHeapSize	ヒープ・サイズ (メガバイト単位)	1m	コンポーネント再始動	Repository Directory は <REMOTE>
Locale	en_US、 ja_JP、 ko_KR、 zh_CN、 zh_TW、 fr_FR、 de_DE、 it_IT、 es_ES、 pt_BR 注: これは、サポートされるロケールの一部です。	en_US	コンポーネント再始動	
LogAtInterchangeEnd	true または false	false	コンポーネント再始動	
MaxEventCapacity	1 から 2147483647	2147483647	動的	Repository Directory は <REMOTE>
MessageFileName	パスまたはファイル名	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	JMS トランスポートのみ: DuplicateEvent Elimination は true でなければなりません。
OADAutoRestartAgent	true または false	false	動的	Repository Directory は <REMOTE>
OADMaxNumRetry	正数	1000	動的	Repository Directory は <REMOTE>
OADRetryTimeInterval	正数 (単位: 分)	10	動的	Repository Directory は <REMOTE>
PollEndTime	HH:MM (HH は 0 から 23、MM は 0 から 59)	HH:MM	コンポーネント再始動	
PollFrequency	正整数 (単位: ミリ秒)  no (ポーリングを使用不可にする)  key (コネクタのコマンド・プロンプト・ウィンドウで文字 p が入力された場合にのみポーリングする)	10000	動的	

表 18. 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
PollQuantity	1 から 500	1	エージェント再始動	JMS トランスポートのみ: Container Managed Events を指定
PollStartTime	HH:MM(HH は 0 から 23、MM は 0 から 59)	HH:MM	コンポーネント再始動	
RepositoryDirectory	メタデータ・リポジトリの場所		エージェント再始動	<REMOTE> に設定する
RequestQueue	有効な JMS キュー名	CONNECTORNAME/REQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
ResponseQueue	有効な JMS キュー名	CONNECTORNAME/RESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS
RestartRetryCount	0 から 99	3	動的	
RestartRetryInterval	適切な正数 (単位: 分): 1 から 2147483547	1	動的	
SourceQueue	有効な JMS キュー名	CONNECTORNAME/SOURCEQUEUE	エージェント再始動	Delivery Transport が JMS であり、かつ Container Managed Events が指定されている場合のみ
SynchronousRequestQueue	有効な JMS キュー名	CONNECTORNAME/ SYNCHRONOUSREQUESTQUEUE	コンポーネント再始動	Delivery Transport は JMS
SynchronousRequestTimeout	0 以上の任意の数値 (ミリ秒)	0	コンポーネント再始動	Delivery Transport は JMS
SynchronousResponseQueue	有効な JMS キュー名	CONNECTORNAME/ SYNCHRONOUSRESPONSEQUEUE	コンポーネント再始動	Delivery Transport は JMS
WireFormat	CwBO	CwBO	エージェント再始動	

## 標準構成プロパティ

このセクションでは、各標準コネクタ構成プロパティの定義を示します。

### AdminInQueue

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

デフォルト値は CONNECTORNAME/ADMININQUEUE です。

## AdminOutQueue

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

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

## AgentConnections

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

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

## AgentTraceLevel

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

## ApplicationName

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

## BrokerType

使用する統合ブローカーを指定します。ICS を指定する必要があります。

## CharacterEncoding

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

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

デフォルトでは、ドロップ・リストには、サポートされる文字エンコードの一部のみが表示されます。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 Connector Configurator Express の使用方法に関する付録を参照してください。

## ConcurrentEventTriggeredFlows

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できるビジネス・オブジェクトの数に設定します。例えば、この属性の値を 5 に設定すると、5 個のビジネス・オブジェクトが並行して処理されます。デフォルト値は 1 です。



このプロパティを 1 よりも大きい値に設定すると、ソース・アプリケーションのコネクターが、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、次のように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、コラボレーションが複数のスレッドを使用できるように構成します。
- 宛先アプリケーションのアプリケーション固有コンポーネントが複数の要求を並行して実行できることを確認します。つまり、このコンポーネントがマルチスレッド化されているか、またはコネクター・エージェント並列処理を使用でき、複数プロセスに対応するよう構成されている必要があります。**Parallel Process Degree** 構成プロパティに、1 より大きい値を設定します。

**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ーのポーリングでは無効です。

## ContainerManagedEvents

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

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

デフォルト値は No value です。

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

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

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

## ControllerStoreAndForwardMode

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

このプロパティを `true` に設定した場合、イベントが ICS に到達したときに宛先側のアプリケーション固有のコンポーネントが使用不可であれば、コネクター・コントローラーはそのアプリケーション固有のコンポーネントへの要求をブロックします。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、コネクター・コントローラーは、宛先側のアプリケーション固有のコンポーネントが使用不可であることを検出すると、ただちにすべてのサービス呼び出し要求を失敗させます。

デフォルト値は `true` です。

## ControllerTraceLevel

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

## DeliveryQueue

`DeliveryTransport` が JMS の場合のみ適用されます。

コネクターから WebSphere InterChange Server Express へビジネス・オブジェクトが送信されるときに使用されるキューです。

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

## DeliveryTransport

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。指定可能な値は、IDL (CORBA IIOP) または JMS (Java Messaging Service) です。デフォルトは IDL です。

`DeliveryTransport` プロパティに指定されている値が IDL である場合、コネクターは、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。

### JMS

Java Messaging Service (JMS) を使用して、コネクターとクライアント・コネクター・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すると、`jms.MessageBrokerName`、`jms.FactoryClassName`、`jms.Password`、`jms.UserName` などの追加の JMS プロパティ

イヤーが Connector Configurator Express に表示されます。このうち最初の 2 つは、このトランスポートの必須プロパティです。

**重要:** WebSphere InterChange Server Express で動作しているコネクタで 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があります。

この環境では、WebSphere MQ クライアント内でメモリーが使用されるため、(サーバー側の) コネクタ・コントローラーと (クライアント側の) コネクタの両方を始動するのは困難な場合があります。

## DuplicateEventElimination

このプロパティを true に設定すると、JMS 対応コネクタによるデリバリー・キューへの重複イベントのデリバリーが防止されます。この機能を使用するには、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。これはコネクタ開発時に設定されます。

このプロパティは、false に設定することもできます。

**注:** DuplicateEventElimination を true に設定する際は、MonitorQueue プロパティを構成して保証付きイベント・デリバリーを使用可能にする必要があります。

## EnableOidForFlowMonitoring

このプロパティを true に設定すると、アダプター・フレームワークは、フロー・モニターを使用できるようにするため、着信 **ObjectEventId** を外部キーとしてマークします。

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、そのメッセージを状況表示および問題説明とともにこのプロパティに指定されているキューに移動します。

デフォルト値は CONNECTORNAME/FAULTQUEUE です。

## JvmMaxHeapSize

エージェントの最大ヒープ・サイズ (メガバイト単位)。

デフォルト値は 128M です。

## JvmMaxNativeStackSize

エージェントの最大ネイティブ・スタック・サイズ (キロバイト単位)。

デフォルト値は 128K です。

## JvmMinHeapSize

エージェントの最小ヒープ・サイズ (メガバイト単位)。

デフォルト値は 1M です。

## jms.FactoryClassName

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。JMS をデリバリー・トランスポート機構 (DeliveryTransport) として選択する際は、このコネクター・プロパティを必ず 設定してください。

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

## jms.MessageBrokerName

JMS プロバイダー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選択するときは (DeliveryTransport を参照)、このコネクター・プロパティを必ず 設定してください。

デフォルト値は crossworlds.queue.manager です。

## jms.NumConcurrentRequests

コネクター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要求はブロックされ、既存のいずれかの要求が完了した後で処理されます。

デフォルト値は 10 です。

## jms.Password

JMS プロバイダー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。

デフォルトはありません。

## jms.UserName

JMS プロバイダーのためのユーザー名を指定します。このプロパティの値はオプションです。

デフォルトはありません。

## Locale

言語コード、国または地域、および、希望する場合には、関連した文字コード・セットを指定します。このプロパティの値は、データの照合やソート順、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。

<code>ll</code>	2 文字の言語コード (普通は小文字)
<code>TT</code>	2 文字の国または地域コード (普通は大文字)
<code>codeset</code>	関連文字コード・セットの名前。名前のこの部分は、通常、オプションです。

デフォルトでは、ドロップ・リストには、サポートされるロケールの一部のみが表示されます。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 **Connector Configurator Express** の使用方法に関する付録を参照してください。

デフォルト値は `en_US` です。コネクターがグローバル化に対応していない場合、このプロパティーの有効な値は `en_US` のみです。特定のコネクターがグローバル化に対応しているかどうかを判別するには、以下の **Web** サイトにあるコネクターのバージョン・リストを参照してください。

<http://www.ibm.com/software/websphere/wbiadapters/infocenter>、または  
<http://www.ibm.com/websphere/integration/wicsserver/infocenter>

## LogAtInterchangeEnd

統合ブローカーのログ宛先にエラーを記録するかどうかを指定します。ブローカーのログ宛先にログを記録すると、電子メール通知もオンになります。これにより、エラーまたは致命的エラーが発生すると、`InterchangeSystem.cfg` ファイルに指定された `MESSAGE_RECIPIENT` に対する電子メール・メッセージが生成されます。

例えば、`LogAtInterChangeEnd` を `true` に設定した場合にコネクターからアプリケーションへの接続が失われると、指定されたメッセージ宛先に、電子メール・メッセージが送信されます。デフォルト値は `false` です。

## MaxEventCapacity

コントローラー・バッファー内のイベントの最大数。このプロパティーは、フロー制御で使用されます。

値は 1 から 2147483647 の間の正整数です。デフォルト値は 2147483647 です。

## MessageFileName

コネクター・メッセージ・ファイルの名前です。メッセージ・ファイルの標準位置は `¥connectors¥messages` です。メッセージ・ファイルが標準位置に格納されていない場合は、メッセージ・ファイル名を絶対パスで指定します。

コネクター・メッセージ・ファイルが存在しない場合は、コネクターは `InterchangeSystem.txt` をメッセージ・ファイルとして使用します。このファイルは、製品ディレクトリーに格納されています。

**注:** 特定のコネクターについて、コネクター独自のメッセージ・ファイルがあるかどうかを判別するには、該当するアダプターのユーザーズ・ガイドを参照してください。

## MonitorQueue

コネクタが重複イベントをモニターするために使用する論理キューです。このプロパティは、DeliveryTransport プロパティ値が JMS であり、かつ DuplicateEventElimination が TRUE に設定されている場合にのみ使用されます。

デフォルト値は CONNECTORNAME/MONITORQUEUE です。

## OADAutoRestartAgent

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、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。MQ により起動される OAD 機能の構成方法については、「システム・インストール・ガイド (Windows 版)」を参照してください。

デフォルト値は false です。

## OADMaxNumRetry

異常シャットダウンの後で MQ によりトリガーされる OAD がコネクタ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。

デフォルト値は 1000 です。

## OADRetryTimeInterval

MQ によりトリガーされる OAD の再試行時間間隔の分数を指定します。コネクタ・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。

デフォルト値は 10 です。

## PollEndTime

イベント・キューのポーリングを停止する時刻です。形式は HH:MM です。ここで、HH は 0 から 23 時を表し、MM は 0 から 59 分を表します。

このプロパティには必ず有効な値を指定してください。デフォルト値は HH:MM ですが、この値は必ず変更する必要があります。

## PollFrequency

ポーリング・アクション間の時間の長さです。PollFrequency は以下の値のいずれかに設定します。

- ポーリング・アクション間のミリ秒数。
- ワード `key`。コネクタは、コネクタのコマンド・プロンプト・ウィンドウで文字 `p` が入力されたときにのみポーリングを実行します。このワードは小文字で入力します。
- 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。

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。このプロパティが使用されるかどうかを特定のコネクタについて判断するには、該当するアダプター・ガイドのインストールと構成についての章を参照してください。

## PollQuantity

コネクタがアプリケーションからポーリングする項目の数を指定します。アダプターにコネクタ固有のポーリング数設定プロパティがある場合、標準プロパティの値は、このコネクタ固有のプロパティの設定値によりオーバーライドされます。

## PollStartTime

イベント・キューのポーリングを開始する時刻です。形式は `HH:MM` です。ここで、`HH` は 0 から 23 時を表し、`MM` は 0 から 59 分を表します。

このプロパティには必ず有効な値を指定してください。デフォルト値は `HH:MM` ですが、この値は必ず変更する必要があります。

## RequestQueue

WebSphere InterChange Server Express からコネクタへビジネス・オブジェクトが送信されるときに使用されるキューです。

デフォルト値は `CONNECTOR/REQUESTQUEUE` です。

## RepositoryDirectory

コネクタが XML スキーマ文書を読み取るリポジトリの場所です。この XML スキーマ文書には、ビジネス・オブジェクト定義のメタデータが含まれています。

この値は `<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。

## ResponseQueue

`DeliveryTransport` が `JMS` の場合のみ適用されます。

`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へデリバリーします。WebSphere InterChange Server Express は、要求を送信した後、`JMS` 応答キューで応答メッセージを待機します。

## RestartRetryCount

コネクタ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コネクタに対して使用する場合、コネクタのマスター側のアプリケーション固有のコンポーネントがスレーブ側のアプリケーション固有のコンポーネントの再始動を試行する回数が指定されます。

デフォルト値は 3 です。

## RestartRetryInterval

コネクタ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コネクタに対して使用する場合、コネクタのマスター側のアプリケーション固有のコンポーネントがスレーブ側のアプリケーション固有のコンポーネントの再始動を試行する間隔が指定されます。指定可能な値の範囲は 1 から 2147483647 です。

デフォルトは 1 です。

## SourceQueue

DeliveryTransport が JMS で、ContainerManagedEvents が指定されている場合のみ適用されます。

JMS イベント・ストアを使用する JMS 対応コネクタ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、87 ページの『ContainerManagedEvents』を参照してください。

デフォルト値は CONNECTOR/SOURCEQUEUE です。

## SynchronousRequestQueue

DeliveryTransport が JMS の場合のみ適用されます。

同期応答を要求する要求メッセージを、コネクタ・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、SynchronousRequestQueue にメッセージを送信し、SynchronousResponseQueue でブローカーから戻される応答を待機します。コネクタに送信される応答メッセージには、元のメッセージの ID を指定する相関 ID が含まれています。

デフォルトは CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE です。

## SynchronousResponseQueue

DeliveryTransport が JMS の場合のみ適用されます。

同期要求に対する応答として送信される応答メッセージを、ブローカーからコネクタ・フレームワークに配信します。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。

デフォルトは CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE です。



## SynchronousRequestTimeout

DeliveryTransport が JMS の場合のみ適用されます。

コネクタが同期要求への応答を待機する時間を分単位で指定します。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をエラー・メッセージとともに障害キューに移動します。

デフォルト値は 0 です。

## WireFormat

トランスポートのメッセージ・フォーマットです。設定値は CwBO です。



---

## 付録 B. Connector Configurator Express

この付録では、Connector Configurator Express を使用してアダプターの構成プロパティ値を設定する方法について説明します。

この付録では、次のトピックについて説明します。

- 『Connector Configurator Express の概要』
- 98 ページの『Connector Configurator Express の始動』
- 99 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』
- 101 ページの『新しい構成ファイルを作成』
- 104 ページの『構成ファイル・プロパティの設定』
- 111 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、WebSphere 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、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、ロギングとトレース、およびデータ・ハンドラーに関するパラメーターを指定する必要があります。

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で必要なプロパティ) 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、新規に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、特定のアダプター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、99 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。

**注:** Connector Configurator Express は、Windows 環境でのみ実行できます。UNIX 環境でコネクタを実行する場合は、Windows で Connector Configurator Express を使用して構成ファイルを変更し、このファイルを UNIX 環境へコピーします。

---

## 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 Express 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 Server Express」>「Toolset Express」>「開発」>「Connector Configurator Express」をクリックします。
- 「ファイル」>「新規」>「構成ファイル」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 (104 ページの『構成ファイルの完成』を参照)。

---

## System Manager からの Configurator Express 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。

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。

1. 「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。

2. 「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、または既存のファイル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。

### 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。

テンプレートは以下のように作成します。

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. 以下のフィールドを含む「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。
  - 「テンプレート」、「名前」

このテンプレートが使用されるコネクタ（またはコネクタのタイプ）を表す固有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。

- 「旧テンプレート」、「変更する既存のテンプレートを選択してください」

「テンプレート名」表示に、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。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。
3. 「テンプレート名」表示からテンプレートを選択し、その名前を「名前の検索」フィールドに入力し（または「テンプレート名」で自分の選択項目を強調表示し）、「次へ」をクリックします。

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。

## 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。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
  - 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」のフィールドで、変更を行います。次のステップで説明するように、プロパティの「プロパティ値」ダイアログ・ボックスを開かない限り、そのプロパティの変更内容は受け入れられませんので、注意してください。
4. 値テーブルの左上の隅にあるボックスを右マウス・ボタンでクリックしてから、「追加」をクリックします。「プロパティ値」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。適切な値または範囲を入力し、「OK」をクリックします。
5. 「値」パネルが最新表示され、「最大長」および「最大複数値」で行った変更が表示されます。以下のような 3 つの列があ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。

---

## 新しい構成ファイルを作成

コネクター構成ファイルを作成するには、コネクター固有のテンプレートから作成するか、既存の構成ファイルを変更します。

### 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. 「ファイル」>「新規」>「コネクター構成」をクリックします。

- 以下のフィールドを含む「**新規コネクタ**」ダイアログ・ボックス表示されません。

- **名前**

コネクタ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**

デフォルトのブローカーは ICS です。この値は変更できません。

- 「**コネクタ固有プロパティ**・**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。

- 構成しているコネクタ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。

- 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、コネクタの始動ファイルで指定するコネクタ構成ファイルのパスおよび名前に一致する必要があります。

- 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、コネクタ定義を完了します。

---

## 既存ファイルの使用

既存ファイルを使用してコネクタを構成するには、Connector Configurator Express でそのファイルを開き、構成を修正してから、構成ファイル (\*.cfg) として保管する必要があります。

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。



- コネクタ定義ファイル。  
コネクタ定義ファイルは、特定のコネクタのプロパティと、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。コネクタの配布パッケージの `¥repository` ディレクトリ内には、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 コネクタの場合は `CN_XML.txt` です)。
- InterChange Server Express リポジトリ・ファイル。  
以前にコネクタの 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、そのコネクタの構成に使用されたりリポジトリ・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。
- コネクタの以前の構成ファイル。  
こ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、コネクタのコネクタ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。

ディレクトリから `*.txt`、`*.cfg` または `*.in` ファイルを開くには、以下のステップを実行します。

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 (`*.in`、`*.out`)(InterChange Server Express Repository (`*.in`、`*.out`))

これまでリポジトリ・ファイルを使用してコネクタを構成していた場合は、このオプションを選択します。リポジトリ・ファイルに複数のコネクタ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ーを開くと、「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。

Connector Configurator Express では、以下のセクションに記載されているプロパティの値を設定する必要があります。

- 105 ページの『標準コネクター・プロパティの設定』
- 105 ページの『アプリケーション固有の構成プロパティの設定』
- 106 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』
- 108 ページの『関連付けられたマップ』
- 110 ページの『トレース/ログ・ファイル値の設定』

**注:** コネクターが JMS メッセージングを使用するものである場合、データをビジネス・オブジェクトに変換するデータ・ハンドラーを構成できるように、追加のカテゴリが表示されることがあります。詳細については、110 ページの『データ・ハンドラー』を参照してください。

---

## 構成ファイル・プロパティの設定

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。

標準プロパティとコネクター固有プロパティの違いは、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のセットを使用します。これらのプロパティの説明は、各アダプター・ガイドの付録 A にあります。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」フィールドは構成可能です。
- 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。

## 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。
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「いいえ」をクリックします。
  - Connector Configurator Express 内の他のカテゴリの値を入力するには、そのカテゴリ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）で入力した値は、次のカテゴリ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。

## アプリケーション固有の構成プロパティの設定

アプリケーション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。
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. 105 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。

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・プロパティ名を変更すると、コネクタに障害が発生する可能性があります。コネクタ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・プロパティの暗号化

「プロパティを編集」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、各コネクタのアダプター・ユーザーズ・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 A 『コネクタの標準構成プロパティ』の 81 ページの『プロパティ値の設定と更新』にある更新メソッドの説明を参照してください。

コネクタ・プロパティはほとんどが静的なプロパティであり、それらの更新メソッドはコンポーネント再始動です。変更を有効にするには、変更したコネクタ構成ファイルを保管した後、コネクタを再始動する必要があります。

## 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。

サポートされるビジネス・オブジェクトを指定するときには、指定するビジネス・オブジェクトとそのオブジェクトに対応するマップが、システムに存在していなければなりません。ビジネス・オブジェクト定義 (データ・ハンドラー・メタオブジェクトのビジネス・オブジェクト定義を含みます) とマップ定義は、統合コンポー

ネット・ライブラリー (ICL) プロジェクトに保管されている必要があります。ICL プロジェクトの詳細については、「*WebSphere Business Integration Server Express ユーザーズ・ガイド*」を参照してください。

**注:** コネクタによっては、アプリケーションでイベント通知や (メタオブジェクトを使用した) 追加の構成を実行するために、特定のビジネス・オブジェクトをサポートされているものとして指定することが必要な場合もあります。詳細については、本書のビジネス・オブジェクトに関する章と、「*ビジネス・オブジェクト開発ガイド*」を参照してください。

ビジネス・オブジェクト定義がコネクタ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ダウン・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す。
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

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**

場合によっては、関連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。

## リソース

「リソース」タブでは、コネクター・エージェントが、コネクター・エージェント並列処理を使用して同時に複数のプロセスを処理するかどうか、またどの程度処理するかを決定する値を設定できます。

すべてのコネクターがこの機能をサポートしているわけではありません。複数のプロセスを使用するよりも複数のスレッドを使用する方が通常は効率的であるため、Java でマルチスレッドとして設計されたコネクター・エージェントを実行している場合、この機能を使用することはお勧めできません。

## 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。このタブは、アダプターが保証付きイベント・デリバリーを利用するものである場合に使用可能になります。

これらのプロパティに使用する値については、標準プロパティに関する付録の『ContainerManagedEvents』の説明を参照してください。

---

## 構成ファイルの保管

構成ファイルの作成とそのファイルに含まれるプロパティの設定が完了したら、使用するコネクタに応じた適切な場所にそのファイルを配置する必要があります。ICL プロジェクトに構成を保管し、保管されたファイルを System Manager から InterChange Server Express へロードしてください。

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。
- System Manager から、指定したディレクトリーに \*.con 拡張子付きファイルとして保管します。
- 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。

System Manager でのプロジェクトの使用方法と、配置の詳細については、「*User Guide for IBM WebSphere Business Integration Server Express*」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ーのドロップ・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 プロパティー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、以下に太文字で示した行を追加してください。

```
<Property name="Locale"
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>
<DefaultValue>en_US</DefaultValue>
</ValidValues>
</Property>
```

---

## 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。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

---

## プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

**警告:** 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

---

## 商標

以下は、IBM Corporation の商標です。

IBM  
IBM ロゴ  
AIX  
CrossWorlds  
DB2  
DB2 Universal Database  
Lotus  
Lotus Domino  
Lotus Notes  
MQIntegrator  
MQSeries  
Tivoli  
WebSphere

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

MMX、Pentium および ProShare は、Intel Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

System Manager には、Eclipse Project (<http://www.eclipse.org/>) により開発されたソフトウェアが含まれています。



WebSphere Business Integration Server Express Plus V4.3







Printed in Japan