



BEA WebLogic Integration™

WebLogic Integration EDI ユーザーズ ガイド

著作権

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

限定的権利条項

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

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

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

商標または登録商標

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

その他の商標はすべて、関係各社が著作権を有します。

WebLogic Integration EDI ユーザーズ ガイド

パート番号	日付	ソフトウェアのバージョン
なし	2002年6月	7.0

目次

このマニュアルの内容

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

1. EDI の背景情報

EDI の概要.....	1-1
アーキテクチャの概要.....	1-2
メッセージ構造.....	1-2
他の電子商取引標準との相違.....	1-5
EDI 標準.....	1-6
X12.....	1-6
EDIFACT.....	1-7
TRADACOMS.....	1-7
BEA WebLogic EDI Integration のアーキテクチャ.....	1-7
BEA WebLogic Adapter for Power.Enterprise!.....	1-8
BEA EDI Connect for WebLogic Integration.....	1-8

2. アーキテクチャ

EDI Integration のアーキテクチャ.....	2-1
Power.Enterprise! のアーキテクチャ.....	2-2
機能の概要.....	2-3
サポートされる標準.....	2-3
制限事項.....	2-4
Power.Map!.....	2-4
マップの作成および保守.....	2-5
ドキュメントの作成および保守.....	2-5
Power.Manager!.....	2-5

トレーディング パートナの保守	2-6
接続.....	2-6
エクステンジブプロファイル	2-6
ドキュメントのトラッキング	2-6
管理.....	2-7
VAN および Network との接続.....	2-7

3. BEA WebLogic Adapter for Power.Enterprise! 3.1

Application Integration の概要	3-1
アプリケーション ビュー.....	3-2
BEA WebLogic Adapter for Power.Enterprise! のコンフィグレーション ...	3-3
イベント	3-6
イベントのコンフィグレーション	3-7
イベントのテスト	3-9
サービス.....	3-9
サービスのコンフィグレーション	3-10
サービスのテスト	3-11
BEA WebLogic Integration Studio へのアダプタ プラグインの使用.....	3-12
新しい WebLogic Integration ドメインでのアダプタのデプロイメント ...	3-12
Windows システムの場合	3-13
UNIX システムの場合	3-13
例外処理.....	3-14

4. Power.Enterprise! のコンフィグレーション

BEA EDI Connect for WebLogic Integration のインストール	4-1
ライセンスの取得	4-2
インストールの前提条件の確認	4-2
Power.Server! のインストール	4-3
Power.Client! のインストール	4-4
Get2Connect.Net への接続.....	4-4
非同期通信の設定	4-4
ソフトウェア アップデートのダウンロード	4-4
EDI ドキュメント名の更新	4-5
Power.Server! の起動.....	4-6

Windows での Power.Server! の起動	4-6
UNIX での Power.Server! の起動.....	4-6
サーバへの最初の接続.....	4-7
Power.Enterprise! のコンフィグレーション	4-9
トレーディング パートナのコンフィグレーション	4-9
XML と EDI データのマッピング	4-12
接続の設定および VAN との接続	4-14
RMI 接続の作成	4-15
トレーディング パートナ接続の作成.....	4-17
エクステンジプロファイルのコンフィグレーション	4-18

5. EDI Integration のコンフィグレーション

共通のコンフィグレーション	5-1
トレーディング パートナのコンフィグレーション	5-1
VAN/トレーディング パートナ接続のコンフィグレーション	5-2
EDI ドキュメントを受信する EDI Integration のコンフィグレーション	5-2
準備	5-2
Power.Enterprise! 内でのコンフィグレーション	5-3
BEA WebLogic Integration 内でのコンフィグレーション	5-4
EDI ドキュメントを送信する EDI Integration のコンフィグレーション	5-5
準備	5-5
Power.Enterprise! 内でのコンフィグレーション	5-5
BEA WebLogic Integration 内でのコンフィグレーション	5-6

6. EDI のサンプル

サンプルの概要	6-1
サンプルの設定と実行.....	6-3
要件	6-3
ハードウェアおよび OS の要件.....	6-4
EDI サンプルのコンフィグレーション	6-4
手順 1 : Power.Server! の起動	6-5
手順 2 : Power.Manager! の起動およびパートナーのコンフィグレーション	6-5
手順 3 : Power.Map! の起動およびマップとアダプタのロード	6-9
手順 4 : 接続の設定.....	6-13
手順 5 : エクステンジプロファイルの設定.....	6-18

手順 6: ワークフローの設定.....	6-21
手順 7: アプリケーション ビューのデプロイ.....	6-22
手順 8: サンプルの実行.....	6-23

7. Power.Enterprise! チューニング ガイド

Java 仮想マシン (JVM) のチューニング.....	7-1
データベースのチューニング.....	7-4
Power.Server! のチューニング.....	7-6
HTTP および HTTPS サーバのコンフィグレーション.....	7-7

このマニュアルの内容

このマニュアルでは、電子データ交換（EDI）の概要と、BEA WebLogic Adapter for Power.Enterprise! 3.1 を使用して、BEA WebLogic Integration 環境の EDI を用いたトレーディング パートナとの相互通信を実現する方法について説明します。

このマニュアルの内容は以下のとおりです。

- 第 1 章「EDI の背景情報」では、EDI システムおよび EDI 標準の背景情報について説明します。
- 第 2 章「アーキテクチャ」では、EDI Integration の構造と、それを構成するコンポーネントについて説明します。
- 第 3 章「BEA WebLogic Adapter for Power.Enterprise! 3.1」では、本製品をインストールおよびコンフィグレーションする方法について説明します。
- 第 4 章「Power.Enterprise! のコンフィグレーション」では、BEA WebLogic Integration および BEA WebLogic Adapter for Power.Enterprise! 3.1. と相互運用できるように BEA EDI Connect for WebLogic Integration（個別に購入できます）をコンフィグレーションする方法について説明します。
- 第 5 章「EDI Integration のコンフィグレーション」では、EDI Integration のエンド ツー エンドのコンフィグレーションを行うための要件について説明します。
- 第 6 章「EDI のサンプル」では、単純な Purchase Order ビジネス プロセスのサポートに EDI を使用したサンプル シナリオのコンフィグレーションおよび操作方法について説明します。
- 第 7 章「Power.Enterprise! チューニング ガイド」では、Power.Enterprise! ソフトウェアをチューニングして、デプロイメント環境におけるパフォーマンスを向上させる方法について説明します。

対象読者

このマニュアルは、**EDI Integration** をデプロイおよび**実装**するビジネスアナリストとシステム管理者を対象としています。読者が **EDI** およびネットワークの概念について基本的な知識を持ち、**WebLogic Integration** 環境に**精通**していることを前提として書かれています。

e-docs Web サイト

BEA 製品のドキュメントは、BEA Systems, Inc. の Web サイトで入手できます。BEA のホーム ページで [製品のドキュメント] をクリックするか、または「e-docs」という製品ドキュメント ページ (<http://edocs.beasys.co.jp/e-docs/index.html>) を直接表示してください。

このマニュアルの印刷方法

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

このマニュアルの PDF 版は、WebLogic Integration の Web サイトで入手できます。PDF を Adobe Acrobat Reader で開くと、マニュアルの全体（または一部分）を書籍の形式で印刷できます。PDF を表示するには、WebLogic Integration ドキュメントのホーム ページを開き、[PDF 版] をクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader がない場合は、Adobe の Web サイト (<http://www.adobe.co.jp/>) で無料で入手できます。

関連情報

EDI および EDI 標準の概要については、以下の Web サイトを参照してください。

- Accredited Standards Committee X12 の Web サイト
(<http://www.x12.org/>)
- UN EDIFACT の Web サイト
(<http://www.unece.org/trade/untddid/welcome.htm>)

サポート情報

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

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

本バージョンの BEA WebLogic Integration について不明な点がある場合、または BEA WebLogic Integration のインストールおよび動作に問題がある場合は、BEA WebSUPPORT (<http://websupport.bea.com/custsupp>) を通じて BEA カスタマサポートまでお問い合わせください。カスタマサポートへの連絡方法については、製品パッケージに同梱されているカスタマサポート カードにも記載されています。

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

- お名前、電子メール アドレス、電話番号、ファクス番号
- 会社の名前と住所
- お使いの機種とコード番号
- 製品の名前とバージョン

- 問題の状況と表示されるエラーメッセージの内容

表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
太字	用語集で定義されている用語を示す。
[Ctrl] + [Tab]	複数のキーを同時に押すことを示す。
<i>斜体</i>	強調または書籍のタイトルを示す。
等幅テキスト	コード サンプル、コマンドとそのオプション、データ構造体とそのメンバー、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。 <i>例</i> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
太字の等幅テキスト	コード内の重要な箇所を示す。 <i>例</i> <pre>void commit ()</pre>
<i>斜体の等幅テキスト</i>	コード内の変数を示す。 <i>例</i> <pre>String <i>expr</i></pre>

表記法	適用
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 <i>例</i> LPT1 SIGNON OR
{ }	構文の中で複数の選択肢を示す。実際には、この括弧は入力しない。
[]	構文の中で任意指定の項目を示す。実際には、この括弧は入力しない。 <i>例</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
	構文の中で相互に排他的な選択肢を区切る。実際には、この記号は入力しない。
...	コマンドラインで以下のいずれかを示す。 <ul style="list-style-type: none"> ■ 引数を複数回繰り返すことができる ■ 任意指定の引数が省略されている ■ パラメータや値などの情報を追加入力できる 実際には、この省略記号は入力しない。 <i>例</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
.	コード サンプルまたは構文で項目が省略されていることを示す。実際には、この省略記号は入力しない。



1 EDI の背景情報

この章では、電子データ交換（EDI）、各種の EDI 標準、および BEA の EDI Integration の概要について説明します。この章の内容は以下のとおりです。

- EDI の概要
- EDI 標準
- BEA WebLogic EDI Integration のアーキテクチャ

EDI の概要

電子データ交換（EDI）は、1970 年代末に米国および西欧で整備された共通のデータフォーマット標準のセットです。米国規格協会（American National Standards Institute: ANSI）は EDI 標準の X12 セットの作成開始を主催しましたが、その目的は、以下のような特徴を持つ国家的に認められるデータフォーマットの集合を確立することでした。

- ハードウェアに依存しない
- あいまいではなく、すべてのトレーディング パートナが使用できる
- データ交換のために要求される労働集約的な作業の量を減らす（たとえば、データ入力のを減らす）
- データの送信側で交換を制御でき、受信者によってトランザクションが受信されたかどうか、およびいつ受信されたかを知ることができる

この標準の初期の定義を主催した後、ANSI は X12 標準の保守および展開のために Accredited Standards Committee (ASC) を設立しました。結果として生まれた X12 標準では、受注および処理、発送および受領、請求、支払など多様なビジネス トランザクションで使用するためのデータフォーマットおよびエンコーディング ルールが定義されています。

単一の国際的な EDI 標準を作成する目的で、United Nations/Economic Commission for Europe (UN/ECE) の Working Party on Facilitation of International Trade Procedures により UN/EDIFACT 系の標準が作成されました。EDIFACT の構文は、1987 年に国際標準化機構 (ISO) により採用されました。

X12 標準および EDIFACT 標準は、同様な機能を提供します。X12 標準の方が古く、より成熟しています。X12 には、EDIFACT にはまだない機能が用意されています。しかし、こうした用意されていない機能のほとんどは、EDIFACT 標準でも開発中です。

X12 と EDIFACT の主な相違は、基盤となるデータ構造にあります (たとえば、データ要素やデータセグメント)。X12 および EDIFACT の間に、データ要素の 1 対 1 対応はありません。1 つの EDIFACT データ要素を表現するために複数の X12 データ要素が必要な場合もあります。

アーキテクチャの概要

X12 標準および EDIFACT 標準は、メッセージ要素構造の階層のセットです。各階層の下に、簡単なデータ構造が結合されてさらに複雑なデータ構造を形成しています。各メッセージ構造の階層の上部には、トレーディング パートナ同士で交換される情報単位が配置されます。

データセットごとに、異なる階層およびデータ構造が定義されています。たとえば、X12 997 メッセージ (Functional Acknowledgement) と X12 810 メッセージ (Invoice) は大幅に異なり、また両者の要件は X12 130 メッセージ (Student Educational Record、つまり成績証明書) の要件とは異なります。

メッセージ構造

各 EDI 標準の基盤には、シンプルデータ要素の辞書があります。シンプルデータ要素は X12 または EDIFACT 標準で最も小さな名前付き項目を表します。たとえば修飾子、値、あるいは説明などです。

シンプルデータ要素には、たとえば次のものがあります。

- インボイス発行日
- 許容重量

- 為替レート
- 危険物分類
- 測定単位コード（ポンド、ダース、立方フィート、ガロンなど）

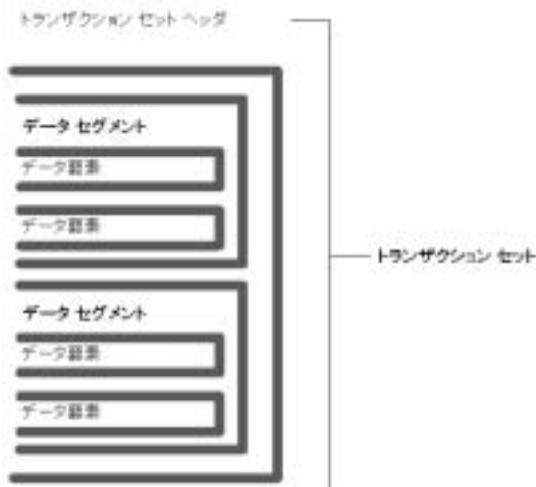
また、X12 標準はコンポジット データ要素も定義します。コンポジット データ要素はシンプル データ要素のセットで、単一の名前付き項目を表します。たとえば、X12 130 のメッセージでは、1 枚の書類について実行されるアクションの識別にコンポジット データ要素 C002 が使用されます。C002 は 5 つのシンプル データ要素から成ります。各シンプル データ要素は、1 つの必要なアクションを識別するコードです。

データ要素が機能ごとに関連する単位にグループ化されると、データ セグメントと呼ばれます。データ セグメントの例として地理的な位置のアドレスを挙げると、このセグメントには都市名、州または郡の名前、郵便番号、および国コードのデータ要素があります。データ セグメントは X12 のセグメント ディレクトリで定義されています。このディレクトリには、各データ セグメントを構成するデータ要素が必要順にリストされています。

一方、データ セグメントはトランザクション セットにグループ化されます。トランザクション セットは、トレーディング パートナによって交換される情報のセットの最小単位です。これによって、たとえば発注書や送付状のような共通のビジネス ドキュメントが表されます。ほとんどのトランザクション セットは 3 つの領域（テーブルと呼ばれる）に区分されます。各領域が、出力されるドキュメントの一部と対応します。

- テーブル 1 であるトランザクション セット ヘッダはヘッダ領域で、トランザクション全体に関する情報が入ります。
- テーブル 2 は詳細の領域で、1 つ以上のデータ セグメントから成ります。たとえば、発注書の行の各項目が、詳細の領域に入ります。
- テーブル 3 であるトランザクション セット トレーラは集計の領域で、1 つのトランザクション セットに使用されているデータ セグメントの数といった情報が入ります。

図 1-1 トランザクション セットの構造



トランザクション セットは出力されるドキュメントを表しますが、EDI トランザクションで交換される情報単位ではありません。類似しているトランザクション セットは、機能グループに分類されます。たとえば、A 社が B 社に 2 つの見積もり要求書 (RFQ) と 5 つの発注書 (PO) を送信する場合は、2 つの RFQ が 1 つの機能グループにまとめられ、5 つの PO は別の機能グループにまとめられます。その後で、特定のトレーディング パートナ向けのすべての機能グループが、EDI インターチェンジと呼ばれる情報単位にまとめられます。トレーディング パートナ間で転送されるのは、このインターチェンジです。

いつでも利用できる高速ネットワークの到来により、多くの EDI システム リアルタイム EDI インターチェンジへと移行し、各トランザクション セットをトレーディング パートナへ転送する処理は、可能であれば、トランザクション セットが EDI システムに個別の EDI インターチェンジとして到着した時点で行われるようになりました。旧来のシステムでは、トランザクション セットがまとめられて、1 つまたは複数のコンポジット EDI インターチェンジとして転送されました。EDI は電子商取引市場に幅広くそして深く浸透しているので、トレーディング パートナとの間で新旧どちらの状況にも遭遇する可能性があります。

インターチェンジの機能グループは、インターチェンジのヘッダおよびトレーラの内部にラップされます。ヘッダには次の制御情報が入ります。

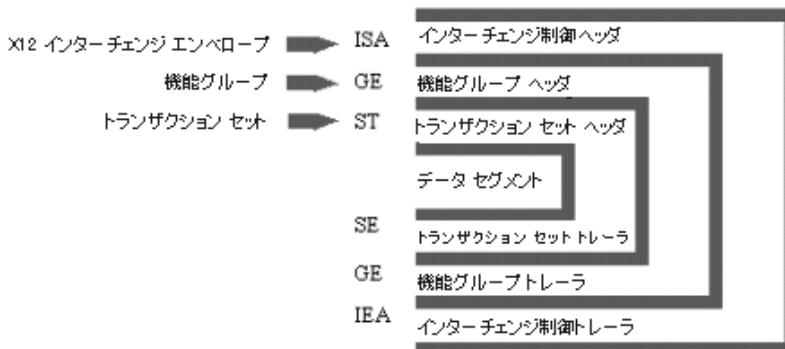
- インターチェンジの送信者および受信者

- インターチェンジの日付および時刻
- インターチェンジの標準およびバージョン
- 認証およびセキュリティの情報
- インターチェンジを追跡するための一意な制御番号

トレーラによりインターチェンジの末尾が示されます。トレーラにはインターチェンジ内の機能グループの総数が示されるほかに、ヘッダで指定された一意なインターチェンジ制御番号が繰り返されます。

次の図は、以上のすべてが合わさって X12 EDI メッセージを形成する様子を示します。EDIFACT および TRADACOMS のメッセージも、同一ではないものの類似したメッセージ構造を使用します。

図 1-2 X12 のメッセージ構造



他の電子商取引標準との相違

EDI 以外の電子商取引標準、たとえば RosettaNet や BizTalk は最新のテクノロジーを活用できるように設計されていますが、それらにも制限事項はあります。

一般に、こうした電子商取引標準には以下の要件があります。

- 標準の転送プロトコルとして HTTP または HTTPS、あるいはその両方を使用する
- データをパッケージする方法として XML を使用する

- トレーディング パートナへのリアルタイム アクセス
- データグラムあたり 1 つのトランザクションの配信をサポートする構造

さらに、これらの標準は比較的新しいために、一般的には小さな業界セグメントに適用が限定され、使用できるトランザクション メッセージの数も比較的少なくなります。しかし、これらの電子商取引標準は発展の速度が比較的速く、新しい機能も次々と追加されています。

対照的に、最も一般的な X12、EDIFACT、および TRADACOMS などがある EDI は、確立された多様な標準を提供します。これらの標準は多様なトランザクションタイプを提供して、幅広い業界および政府機関のセグメントに対応します。

EDI 標準

現在、主要な EDI 標準は 3 つあります。X12、EDIFACT、および TRADACOMS です。

X12

X12 標準は、ANSI による Accredited Standards Committee (<http://www.x12.org/>) によって保守されています。この標準は、主に米国およびカナダで使用されています。米国連邦政府は、すべての EDI 実装標準の Federal レジストリを連邦政府で使用されているとおりに保守する目的で、National Institute of Standards and Technology を設立しました。詳細については、Web サイト (<http://snad.ncsl.nist.gov/dartg/edi/fededi.html>) を参照してください。

X12 標準の新規リリースは、技術的なアップデートと新しい EDI トランザクションが追加された上で定期的に公開されています。

EDIFACT

EDIFACT 標準は United Nations/Economic Commission for Europe (UN/ECE) および国際標準化機構 (ISO) によって保守されています (<http://www.unece.org/trade/untdid>)。EDIFACT 標準は、主に欧州で使用されています。

TRADACOMS

TRADACOMS 標準は、ANA (Article Numbering Association) によって開発および保守されています。TRADACOMS 標準は、主に英国の小売業界で使用されています。

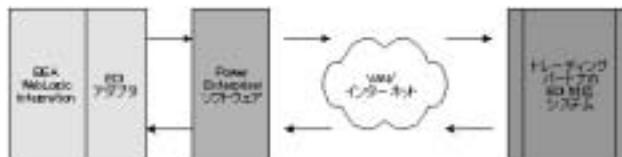
BEA WebLogic EDI Integration のアーキテクチャ

BEA では、BEA WebLogic Adapter for Power.Enterprise! (BEA WebLogic Platform に付属) および BEA EDI Connect for WebLogic Integration (Power.Enterprise! として個別購入が可能) という 2 つのコンポーネントによるオプションの EDI サポートを用意しています。

注意： BEA WebLogic Adapter for Power.Enterprise! は、Power.Enterprise! 3.0 と 3.1 バージョンの両方をサポートしています。

図 1-3 は、BEA WebLogic Adapter for Power.Enterprise! および BEA EDI Connect for WebLogic Integration が協調して WebLogic Integration をトレーディング パートナの EDI システムで使用できるようにする図式を示します。

図 1-3 BEA EDI Integration のアーキテクチャ



BEA WebLogic Adapter for Power.Enterprise!

BEA WebLogic Adapter for Power.Enterprise! は、WebLogic Integration と Power.Enterprise! ソフトウェアの間のゲートウェイを提供します。BEA WebLogic のテクノロジーで定義する標準的なアプリケーション統合イベントおよびサービスを使用すれば、BEA WebLogic Adapter for Power.Enterprise! では以下のことができます。

- Power.Enterprise! を使用して、トレーディング パートナと XML ドキュメントを交換する
- Power.Enterprise! を使用して、トレーディング パートナからドキュメントを受信する

BEA EDI Connect for WebLogic Integration

BEA EDI Connect for WebLogic Integration（個別購入が可能です）は 3 つのコンポーネントから成る一式です。

- Power.Server!
- Power.Map!
- Power.Manager!

これらのコンポーネントを使用して、EDI から XML へのドキュメント マップを定義したり、トレーディング パートナとの関係を管理したり、EDI メッセージの送信および受信を処理したりできます。

2 アーキテクチャ

以下の節では、ED I Integration のアーキテクチャについて、BEA WebLogic Adapter for Power.Enterprise!、Power.Enterprise! ソフトウェア、および VAN/ ネットワーク接続に関する情報も含めて説明します。

- ED I Integration のアーキテクチャ
- Power.Enterprise! のアーキテクチャ
- VAN および Network との接続

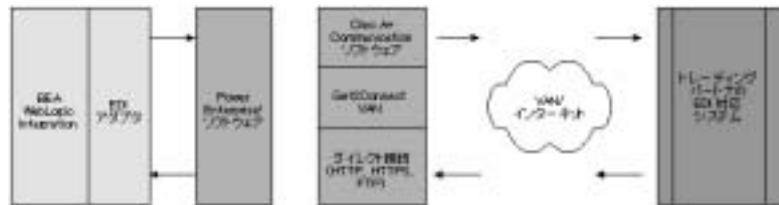
ED I Integration のアーキテクチャ

BEA では、BEA WebLogic Adapter for Power.Enterprise! (BEA WebLogic Platform に付属) および BEA ED I Connect for WebLogic Integration (Power.Enterprise! として個別購入が可能) という 2 つのコンポーネントによるオプションの ED I サポートを用意しています。

注意： BEA WebLogic Adapter for Power.Enterprise! は、Power.Enterprise! 3.0 と 3.1 バージョンの両方をサポートしています。

図 2-1 は、BEA WebLogic Adapter for Power.Enterprise! および Power.Enterprise! ソフトウェアが協調して WebLogic Integration をトレーディング パートナの ED I システムで使用できるようにする図式を示します。

図 2-1 BEA EDI Integration のアーキテクチャ



EDI アダプタはアプリケーション統合を使用して WebLogic Integration で XML ドキュメントを Power.Enterprise! と交換できるようにしますが、この交換では RMI メッセージ渡しを使用されます。

Power.Enterprise! の Power.Server! コンポーネントは、EDI Integration に EDI メッセージを変換、送信、および受信する機能を提供します。Power.Enterprise! は、個別のトレーディング パートナとのすべての EDI トランザクションをリアルタイムに解決するために、ランタイム、管理、およびマッピングのツールを提供します。

Power.Enterprise! のアーキテクチャ

Power.Enterprise! は、統一された、Java ベースの作業環境を提供して、ドキュメントの送信先を適切な EDI ドキュメントにマップしたり、トレーディング パートナとの関係を管理したりできるようにします。この環境は 3 つのコンポーネントから成ります。

- **Power.Server!:** この Java ベースの実行時サーバは、EDI メッセージトラフィックのためのディスパッチおよび受信のサービスを提供します。Power.Server! は、Power.Manager! で定義されているトレーディング パートナとのトランザクションを実行します。受信および送信メッセージは、Power.Map! で指定された定義に従って変換されます。
- **Power.Manager!:** このアプリケーションを使用すると、複数のマシンにわたる複数の Power.Server! インスタンスの管理が、すべてのトレーディング パートナについてトレーディング パートナの関係および接続情報を定義することで可能になります。

- **Power.Map!**: このアプリケーションを使用すると、使用している各種のデータフォーマットと一般的な EDI データ構造の間でマップをロード、作成、および格納できます。**Power.Map!** を使用することで、簡単な変換の場合に簡潔なマッピング ルールを作成したり、より複雑で高度な変換のために式や JavaScript を使用したりできます。

機能の概要

Power.Enterprise! は、メッセージのトランスフォーメーションおよび配信のための完備したソリューションを提供します。

- **Power.Server!** は、リアルタイム データ トランスフォーメーション エンジンを提供し、ANSI X12、UN/EDIFACT、および TRADACOMS のデータフォーマットをサポートします。
- **Power.Map!** は、マッピングにドラッグアンドドロップ機能を提供する拡張可能なマッピング ツールです。また、**Power.Map!** では式または JavaScript コードを使用してより複雑な変換ができます。
- **Power.Manager!** はコンフィグレーションおよび管理のツールで、トレーディング パートナとの関係および単一方向接続のコンフィグレーションを可能にします。

Power.Enterprise! には IP レベルで Get2Connect VAN に接続する機能も備わっているほか、さまざまな標準のネットワーク接続プロトコルもサポートしています。

サポートされる標準

Power.Enterprise! は、以下の EDI 標準およびデータ標準をサポートしています。

- X12
- EDIFACT
- TRADACOMS
- XML
- フラットファイル

制限事項

現在、Power.Enterprise! 一式は以下のシステムで使用可能です。

- Power.Server! は、Windows NT/2000、Solaris、HPUX、および AIX システムで使用可能です。
- Power.Enterprise! が動作するには SQL データベースが必要です。
Power.Enterprise! は、Oracle 8i (8.1.7 以降)、Microsoft SQL Server 7.0 または 2000、および IBM DB2 データベースをサポートします。
- Power.Map! および Power.Management! は、Windows NT/2000 システムでのみ使用可能です。

Power.Map!

Power.Map! は、Power.Enterprise! で使用可能なデータ フォーマット マッピング機能を定義するツールです。Power.Map! は、データを EDI フォーマットに変換したり、その逆の変換をすることが可能なため、Power.Map! でサポートされているあらゆるドキュメント構造標準を選択できる一方で、各種の標準およびトレーディング パートナに独自の EDI フォーマットとやりとりを行う機能も保持しています。

Power.Map! には次の 3 つの項目が必要です。

- ソースドキュメントのデータ定義
- 対象のドキュメントのデータ定義
- ソースドキュメントのデータと対象のドキュメントのデータ間の、リンクのマップ

Power.Map! に定義されていないデータ フォーマットについては、たとえば DTD や XSD などのデータ定義を個別に提供する必要があります。Power.Map! を使用すれば、アプリケーション内で XML データ構造を構築したり、外部で作成された DTD およびスキーマをインポートしたりできます。こうしたドキュメントを Power.Map! にインポートして、必要なマップを定義することもできます。

Power.Map! ではデータ定義のために XML の DTD およびスキーマのインポートがサポートされていますが、データ フォーマットをエクスポートする機能はありません。ドキュメントのフォーマットのインポートのみができます。このた

め、データ インターチェンジドキュメントの設計は **Power.Map!** の外部で行い、そのドキュメントのコピーを **Power.Map!** にインポートしてマップを作成する必要があります。

Power.Map! には標準的な EDI メッセージのほとんどが組み込みの定義として用意されていて、**X12**、**EDIFACT**、および **TRADACOMS** のメッセージのサポートも含まれています。

マップの作成および保守

Power.Map! を使用すると、サーバに接続しないオフライン状態でマップを作成、保守、およびテストできます。これらのマップは、インポートするドキュメント定義、**Power.Map!** で提供される EDI ドキュメント定義、およびユーザが定義するカスタム定義に基づきます。**Power.Map!** を使用すると、定義したマップをエクスポートしたり、別の方法で作成されたマップをインポートしたりできます。

ドキュメントの作成および保守

Power.Server! で使用するドキュメント定義は、別個に保守される必要があります。**Power.Map!** は、XML の DTD および XSD のインポートのみをサポートしています。これらのドキュメントは、**Power.Map!** の外部では使用できません。たとえば、これらのドキュメントは **WebLogic Integration** では使用できません。**Power.Map!** 以外で作成されたドキュメントである必要があります。

さらに、使用する EDI ドキュメント標準に合致しているかを検証する必要があります。**Power.Enterprise!** では多様な標準およびその各バージョンがサポートされていますが、使用する特定のトランザクション、標準、およびバージョンがサポートされていない可能性もあります。サポートされていないトランザクション、標準、またはバージョンを使用する場合は、BEA カスタマ サポートまでお問い合わせください。

Power.Manager!

Power.Manager! は、サーバおよびトレーディング パートナとの関係の保守および管理に使用されます。**Power.Manager!** を使用すると、**Power.Server!** の複数のインスタンスを、単一の、リモートの場所から管理できます。

トレーディング パートナの保守

Power.Manager! には、トレーディング パートナ定義およびそれと関連付けられているデータを保守する機能があります。こうしたトレーディング パートナ定義は、EDI Integration に特有のものです。BEA WebLogic Integration システムの他のコンポーネントで共有されたり、使用されたりすることはありません。

接続

Power.Manager! を使用すると、複数の同時接続を保守できます。各接続は、Power.Server! に入る方向または出る方向、どちらか一方の情報の流れを定義します。このため、一方の接続は、Power.Server! から EDI アダプタへのドキュメントの RMI 転送を定義することになります。他方の接続は、トレーディング パートナによって Get2Connect VAN を通じて送信された EDI ドキュメントを Power.Server! で受信することを定義することになります。

エクステンジ プロファイル

Power.Manager! を使用すると、トレーディング パートナとやり取りするドキュメントの流れを管理する複数のエクステンジ プロファイルを定義できます。各エクステンジ プロファイルは、以下のような特徴を持つドキュメントの単一のセットを管理します。

- 送り元から送り先へという一方向に流れる
- 2つの接続を含む
- マップを含む

1つのエクステンジ プロファイルは、たとえば、Get2Connect VAN を経由するトレーディング パートナからの EDI ドキュメントの受信を管理する接続、EDI ドキュメントを XML ドキュメントに変換するマップ、および生成された XML ドキュメントを RMI 経由で EDI アダプタへ送信する 2つめの接続定義などを含むことがあります。

ドキュメントのトラッキング

Power.Manager! は、ドキュメントをリアルタイムでモニタできるロギング、レポート、およびトラッキングのためのツールを提供しています。

管理

Power.Manager! を使用すると、複数のサーバを管理できます。ドキュメントの管理、ログの表示とコンフィグレーション、およびサーバ上のエラー処理手順の指定が、単一のリモート アプリケーションで行えます。

VAN および Network との接続

EDI トランザクションの多くは、付加価値ネットワーク (VAN) と呼ばれる 1 つまたは複数の専用ネットワークを通じて実行されます。VAN は縦割りまたは横割りの業界セグメントに合わせて作成およびカスタマイズされます。たとえば、以下のようになります。

- **General Electric Global Exchange Services (GE GXS)** は、世界最大級の VAN です。GXS は世界中で幅広い VAN サービスを提供し、100,000 を越えるトレーディング パートナが毎年 1 兆ドルに上る商品およびサービスを交換しています。
- **Telefonica Servicios Avanzados de Informacion (TSAI)** は、横割りの業界セグメントによる VAN の一例です。TSAI は主にスペイン語圏の VAN で、多様な業界にサービスを提供しています。およそ 12.5 億ドルに上る取引が TSAI VAN によって毎年処理され、これはスペインの EDI ビジネスの 60% に相当します。
- **PaperVan** は、縦割りの業界セグメントによる VAN の一例で、紙業界を扱う VAN です。ネットワーキング サービスに加えて、PaperVan はコンサルティングと実装のサポートも提供しています。
- **WalMart** は EDI 戦略の一環として自社専用の VAN を立ち上げ、製品の購買と流通の管理を目指しました。この戦略を実現するために、WalMart はサプライヤに同社の EDI の実装および VAN への接続のための詳細な標準を提供しました。

EDI を実装する企業の多くが、専用ネットワークまたはインターネットを経由したトランザクションの直接処理も試みています。こうした実装の実地試験により、インターネット上のセキュリティが問題にはならないことが示されています。

Power.Enterprise! は、以下の通信オプションをすべてサポートしています。以下の接続方式のいずれかを使用しているパートナーとトランザクションを実行できません。

- 専用 VAN、Cleo A+ ソフトウェアを使用
- 直接ネットワーク接続を経由
- インターネットを経由（HTTP、HTTPS、および FTP 接続のすべてがサポートされている）
- Get2Connect VAN を使用（組み込みでサポート）

Power.Enterprise! は、特定の業界向けではない Get2Connect.net という汎用 VAN への接続を提供しています。

他の VAN へ接続するために、Power.Enterprise! には Cleo A+ の試用ライセンスも含まれています。Cleo A+ は Windows システム用の通信パッケージで、非同期通信をサポートするマイクロコンピュータ、ミニコンピュータ、およびメインフレームコンピュータとのファイル交換を可能にします。

3 BEA WebLogic Adapter for Power.Enterprise! 3.1

BEA WebLogic Adapter for Power.Enterprise! 3.1 は、BEA WebLogic Integration でのアプリケーション統合機能の実装です。BEA WebLogic Adapter for Power.Enterprise! によって、XML メッセージの送受信が可能になります。この XML メッセージは、Power.Enterprise! で EDI メッセージに変換されたり、EDI メッセージから XML メッセージに戻されたりします。この付録のトピックは以下のとおりです。

- Application Integration の概要
- BEA WebLogic Integration Studio へのアダプタ プラグインの使用
- 例外処理

Application Integration の概要

BEA WebLogic Integration の説明については、『*Application Integration 入門*』を参照してください。アプリケーション統合機能の実装および使用の詳細については、『*Application Integration ユーザーズガイド*』を参照してください。

BEA WebLogic Integration の統合ソリューションは、エンタープライズの内部およびエンタープライズ間のどちらの接続アプリケーションについても、現行および将来の標準をサポートしています。BEA WebLogic Integration は、通信のエンドポイントを定義する方法を提供します。通信のエンドポイントをプロセスフローに追加するには、BEA WebLogic Integration の Business Process Management (BPM) 機能を使用するか、完全な統合ソリューションを形成する独自のプログラムを使用することになります。

以前に説明したとおり、BEA WebLogic Adapter for Power.Enterprise! は、BEA WebLogic Integration で提供される統合機能を使用して実装されています。この章では、BEA WebLogic Adapter for Power.Enterprise! に適用されるこうした機能について説明します。

アプリケーション ビュー

メッセージレベルのエンタープライズ アプリケーション統合を実現するためには、各アダプタごとにばく大な数のアプリケーション ビューを作成することになる可能性があります。1つのアプリケーションビューは、1つのアダプタ上のビジネス機能のセットを定義します。このアプリケーションビューは、関係するアダプタを介して、特定のビジネス用途のイベントおよびサービスをサポートします。

- イベントは、アプリケーションによって生成されるメッセージをパブリッシュおよびサブスクライブのモデルに合わせて管理できるようにします。
- サービスは、ユーザによって呼び出されるビジネス機能です。サービスの呼び出しによって、メッセージは要求 / 応答モデルに合わせて送信されるようになります。

イベントおよびサービスの両方について、要求および応答は XML ドキュメントの形式でシステムまで送られます。アプリケーションビューの詳細については、『*Application Integration 入門*』の「統合フレームワーク」を参照してください。

BEA WebLogic Adapter for Power.Enterprise! のビューは、複数のアプリケーションビューを提供し、各ビューにはユーザ定義が可能な複数のサービスおよびイベントがあります。サービスおよびイベントの両方もが XML スキーマによって定義され、データを XML の形式で送信します。この方式により、任意の XML 対応コンポーネント、たとえば WebLogic Integration であれば、BEA WebLogic Adapter for Power.Enterprise! を使用して Power.Enterprise! と通信することができます。BEA WebLogic Adapter for Power.Enterprise! のアプリケーションビュー、サービス、およびイベントの管理には、WebLogic Application Integration アダプタのホームページを使用します。

BEA WebLogic Adapter for Power.Enterprise! が Power.Enterprise! と共に正常に動作するためには、そのようにコンフィグレーションする必要があります。また、アダプタのサービスおよびイベントをコンフィグレーションする必要もあります。

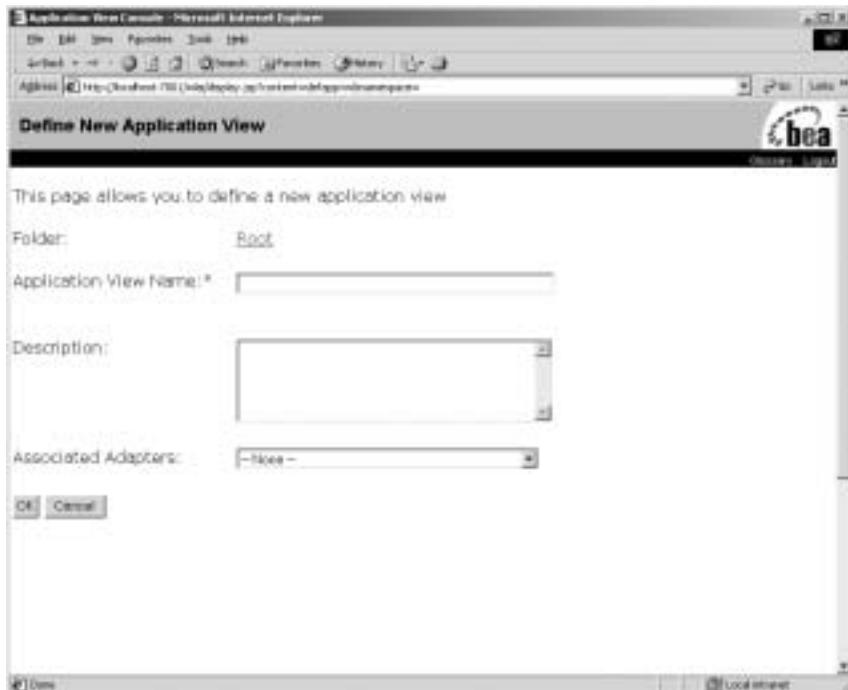
こうしたコンフィグレーション作業を支援するために、BEA WebLogic Adapter for Power.Enterprise! のアプリケーションビューには3つの Java Server Pages (JSP) が用意されています。

BEA WebLogic Adapter for Power.Enterprise! の コンフィグレーション

BEA WebLogic Adapter for Power.Enterprise! が Power.Enterprise! と通信するためには、適切な通信情報をコンフィグレーションする必要があります。手順は、次のとおりです。

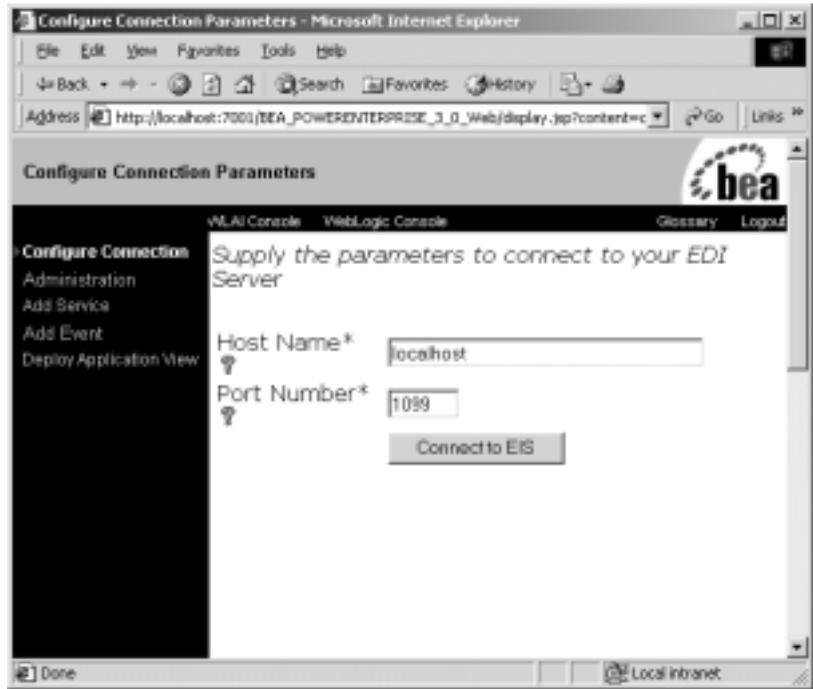
1. BEA WebLogic Integration Application View Console で、[Define New App View] を選択します。[Define New Application View] 画面が表示されます。

図 3-1 [Define New App View] の JSP



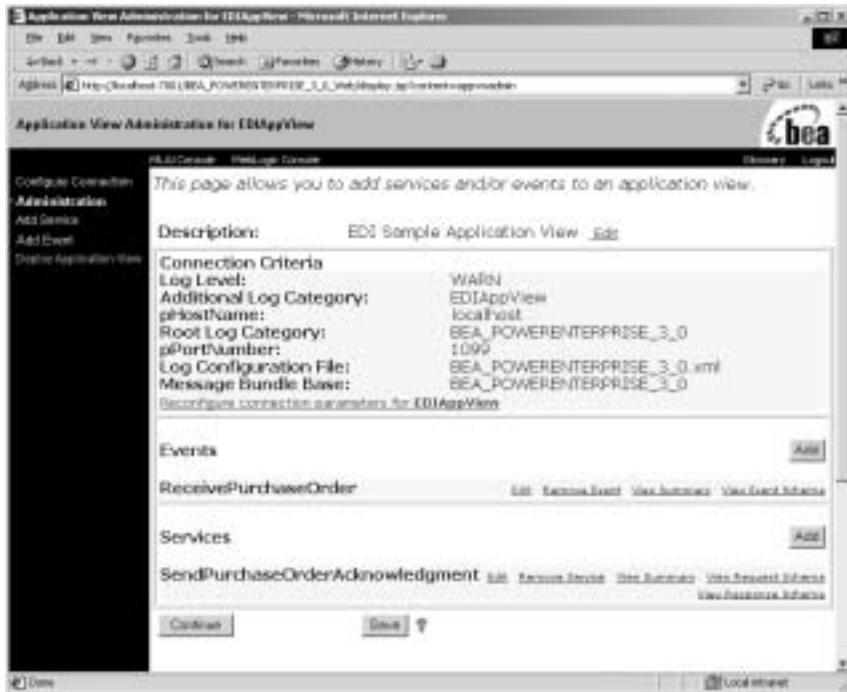
2. [Configure Connection] を選択します。[Configure Connection Parameters] の JSP が表示されます。

図 3-2 [Configure Connection Parameters] の JSP



3. Power.Server! があるシステムのホスト名または IP アドレスを入力します。
4. Power.Enterprise! での RMI 通信に使用されるポートの番号を入力します。デフォルトは 1099 です。
5. [Connect to EIS] を選択します。
6. [Add Service] をクリックし、「サービスのコンフィグレーション」の手順に従って、エクスポートする必要があるすべてのサービスを定義します。
7. [Add Event] をクリックし、「イベントのコンフィグレーション」の手順に従って、エクスポートする必要があるすべてのイベントを定義します。

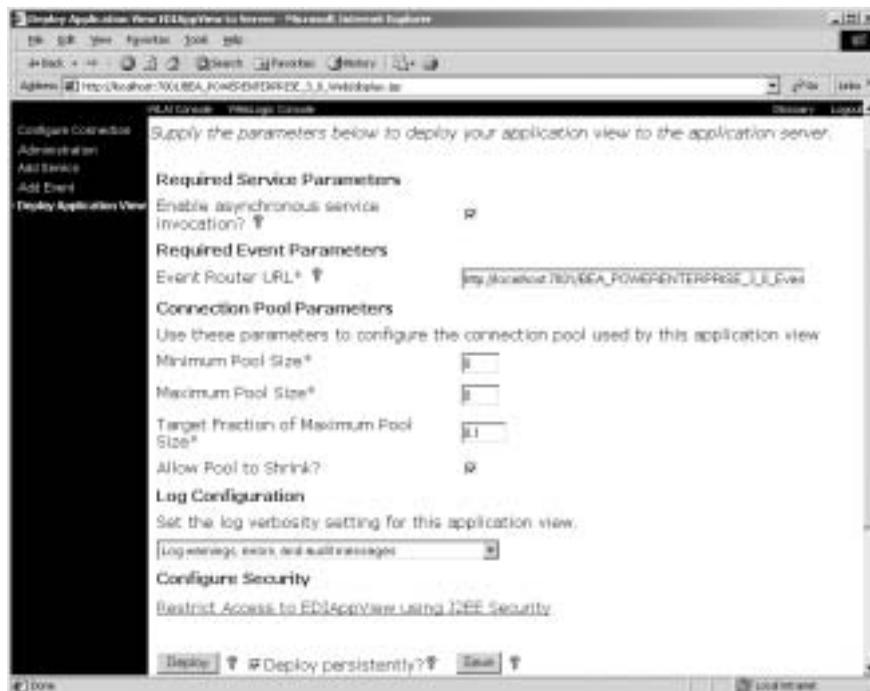
図 3-3 [Configuration Administration] の JSP



- 要件に基づいて接続条件を設定します。こうした条件の多くは、システムコンフィギュレーションと既に入力された情報に基づいて自動的に生成されます。

手順を済ませたら、[Continue] を選択します。[Deploy Application View] ウィンドウが表示されます。

図 3-4 [Deploy Application View] の JSP



9. 必要なパラメータをすべて設定してから、[Deploy] を選択します。コンフィグレーションされたアプリケーションビューがデプロイされます。

注意： 必ず、Power.Server! を起動してから WebLogic Integration または WebLogic Server を起動してください。WebLogic Integration のコネクタは初期化の時点で Power.Server! のアクティブなインスタンスがあることを前提にしているため、初期化の際に Power.Server! が実行中でないと例外を送出します。

イベント

イベントには、発生したイベントに関心があるアプリケーションビューによってパブリッシュされる XML ドキュメントが含まれます。イベントの通知が必要なクライアントは、アプリケーションビューにイベントへの関心を登録します。

アプリケーションビューによってパブリッシュされたイベントにクライアントがサブスクライブすると、サブスクライブされたイベントが対象のアプリケーションの内部で発生したときにアプリケーションビューは必ずクライアントに通知をし、イベントの詳細を示す XML ドキュメントをクライアントに渡します。アプリケーションビューへのイベントの追加に関する詳細については、『*Application Integration ユーザーズガイド*』の「アプリケーションビューの定義」の「手順 4B: アプリケーションビューへのイベントの追加」を参照してください。

BEA WebLogic Adapter for Power.Enterprise! は、ユーザ定義可能なイベントを提供します。こうしたイベントは Java Server Page を使用して作成および管理されます。Java Server Page は、WebLogic Integration Application Interface の一部としてパブリッシュされます。Power.Enterprise! がトレーディング パートナから EDI メッセージを受信して、それを BEA WebLogic Adapter for Power.Enterprise! へ転送すると、アダプタによって適切なイベントがパブリッシュされます。このプロセスは、次のように動作します。

1. トレーディング パートナから EDI ドキュメントを受信した時点で、Power.Enterprise! はそのドキュメントを XML へ変換します。
2. RMI を使用して、Power.Enterprise! は以前に登録済みのイベントがある BEA WebLogic Adapter for Power.Enterprise! を呼び出します。
3. WebLogic Integration は、定義されているイベントをパブリッシュします。

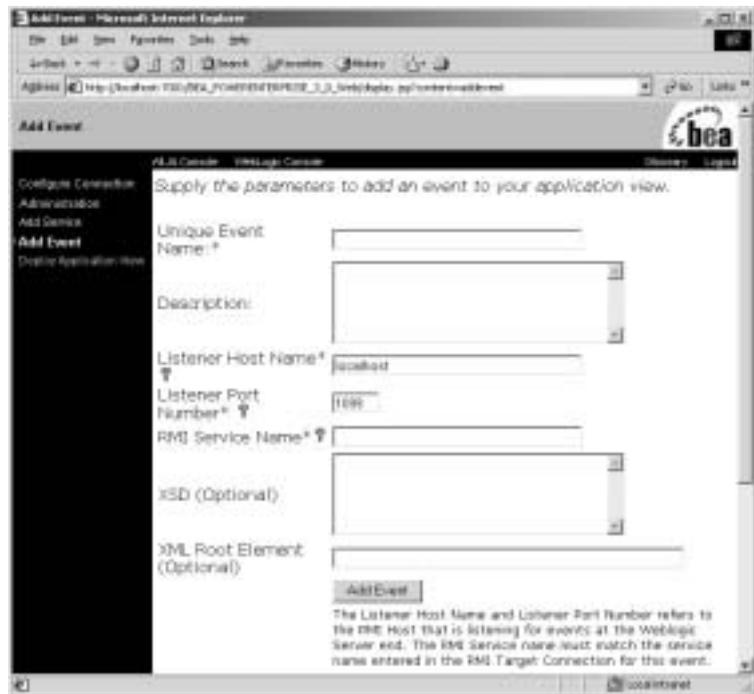
イベントのコンフィグレーション

イベントは BEA WebLogic Adapter for Power.Enterprise! がワークフローからの情報を Power.Enterprise! へ送信するために使用され、Power.Enterprise! はその情報をトレーディング パートナへ転送します。BEA WebLogic Adapter for Power.Enterprise! では、複数のイベントを定義できます。通常は、送信する XML/EDI ドキュメントの種類（たとえば発注書）ごとに 1 つのイベントを定義することになります。

BEA WebLogic Adapter for Power.Enterprise! 上のイベントを定義するには、次の作業が必要です。

1. BEA WebLogic Adapter for Power.Enterprise! のホームページで、[Add Event] をクリックします。[Add Event] ページが表示されます。

図 3-5 [Add Event] の JSP



2. BEA WebLogic Adapter for Power.Enterprise! の内部では一意なイベント名を入力します。
3. 必要ならば、説明を入力します。この手順は省略可能です。
4. Power.Server! のプロセスがあるシステムのホスト名または IP アドレスを入力します。
5. RMI リスナプロセスによって使用されるポートの番号を入力します。
6. Power.Enterprise! に定義されているとおりに、リスナプロセスの RMI サービス名を入力します。

イベントのテスト

イベントおよびワークフローの両方のコンフィグレーションを済ませたら、これらのイベントを使用して正常にメッセージを送信できることを確認する必要があります。Application View Console 内でイベントをテストするには、[Test Event] リンクを使用します。

次の手順は、イベントがメッセージを受信することを確認するための簡単なテスト方式です。

1. Power.Enterprise! のコンフィグレーションを一時的に変更して、トレーディング パートナへのメッセージの送信にファイル接続を使用するようにします。
2. システムを介してメッセージを送信し、結果として生成された EDI メッセージをファイルにダンプします。
3. メッセージが格納されているファイルを調べて、メッセージが正しく作成および送信されたことを確認します。
4. テストが終了したら、接続のコンフィグレーションをやり直して、トレーディング パートナに適した送信方法を使用するようにします。

サービス

サービスは、アプリケーション内のビジネス オペレーションのうち、アプリケーション ビューによってエクスポートされるものです。サービスは要求 / 応答のメカニズムとして機能します。ビジネス サービスを呼び出すための要求をアプリケーションが受け取ると、アプリケーション ビューによってサービスの機能が対象のアプリケーションで呼び出されて、その結果を説明する XML ドキュメントが応答として返されます。アプリケーション ビューへのサービスの追加に関する詳細については、『Application Integration ユーザーズ ガイド』の「アプリケーション ビューの定義」の「手順 4A: アプリケーション ビューへのイベントの追加」を参照してください。

BEA WebLogic Adapter for Power.Enterprise! は、ユーザがコンフィグレーション可能な多数のサービスを提供します。こうしたサービスは Java Server Page を使用して作成および管理されます。Java Server Page は、WebLogic Integration Application Interface の一部としてパブリッシュされます。サービスは

Power.Enterprise! に XML を送信するために使用されます。Power.Enterprise! は XML メッセージを受信した時点で、そのメッセージを EDI メッセージに変換します。その後、EDI メッセージをトレーディング パートナに送信します。

EDI メッセージを送信するには、BEA WebLogic Adapter for Power.Enterprise! 用に定義されたサービスに、適切にフォーマットされた XML ドキュメントを送信します。サービスは RMI を使用して Power.Enterprise! にメッセージを渡し、Power.Enterprise! は XML メッセージを EDI メッセージのフォーマットに変換してトレーディング パートナへ送信します。

サービスのコンフィグレーション

サービスは BEA WebLogic Adapter for Power.Enterprise! がワークフローからの情報を受信するために使用され、その情報は Power.Enterprise! によってトレーディング パートナへ転送されます。BEA WebLogic Adapter for Power.Enterprise! では複数のサービスを定義できますが、ほとんどの状況では、受信する予定の XML/EDI ドキュメント タイプ（たとえば EDI 855 メッセージ）につき 1 つのイベントを定義すれば十分です。

BEA WebLogic Adapter for Power.Enterprise! 上のサービスを定義するには、次の作業が必要です。

1. BEA WebLogic Adapter for Power.Enterprise! のホーム ページで、[Add Service] をクリックします。[Add Service] の JSP が表示されます。

図 3-6 [Add Service] の JSP



2. BEA WebLogic Adapter for Power.Enterprise! の内部では一意なサービス名を入力します。
3. 説明を入力します。この手順は省略可能です。
4. Power.Enterprise! で定義された RMI ソース接続の名前を入力します。このサービスは、Power.Enterprise! からメッセージを受信することになります。

サービスのテスト

サービスのコンフィグレーション、およびこうしたサービスを使用することになるワークフローのコンフィグレーションを終えたら、これらからメッセージを正常に受信できるかどうかを確認する必要があります。Application View Console 内でイベントをテストするには、[Test Service] リンクを使用します。

次の手順は、サービスがメッセージを配信できることを確認するための簡単なテスト方式です。

1. Power.Enterprise! のコンフィグレーションを一時的に変更して、着信のファイル接続を使用するようにします。
2. テストのタスク順序を開始するには、ファイルを 1 つ送信ボックスにドロップします。結果として生成されるメッセージは Power.Enterprise! によって変換されてからワークフローへ送信されます。
3. 使用したワークフローでメッセージを調べて、メッセージが正しく作成および受信されたことを確認します。
4. テストが終了したら、接続のコンフィグレーションをやり直して、トレーディング パートナに適した送信方法を使用するようにします。

BEA WebLogic Integration Studio へのアダプタ プラグインの使用

BEA WebLogic Adapter for Power.Enterprise! には、ビジネス プロセスのモデル化に使用できるグラフィカル ツールである BEA WebLogic Integration Studio が含まれています。アダプタは BEA WebLogic Integration の一部として統合されているので、このプラグインは他の Application Integration アダプタとまったく同様にして EDI メッセージの処理および統合に使用できます。

アダプタおよび BEA WebLogic Integration Studio を使用した BPM ワークフロー デザインの詳細については、『*Application Integration 入門*』の「BPM の Application Integration プラグインについて」および「ADK について」を参照してください。

新しい WebLogic Integration ドメインでのアダプタのデプロイメント

BEA WebLogic Adapter for Power.Enterprise! 3.1 のインスタンスを新しいドメインでデプロイする必要がある場合は、プラットフォームに合わせて適切な手順を実行します。

Windows システムの場合

作業を開始する前に、アダプタの EAR ファイルおよび JAR ファイルをコンピュータのどこに置くかを決めておく必要があります。デフォルトでは、これらのファイルは `WLI_HOME\adapters\powerenterprise\lib` に配置されます。

1. 以下のリストで示すように、アダプタの EAR ファイルをドメインの `config.xml` ファイルで宣言します。

コード リスト 3-1 Windows システムの `config.xml` ファイルでのドメイン宣言

```
<Application Deployed="true" Name="BEA_POWERENTERPRISE_3_0"
  Path="WLI_HOME\adapters\powerenterprise\lib\BEA_POWERENTERPRISE_3_0_EAR.ear">
  <ConnectorComponent Name="BEA_POWERENTERPRISE_3_0"
    Targets="myserver" URI="BEA_POWERENTERPRISE_3_0.rar" />
  <WebAppComponent Name="BEA_POWERENTERPRISE_3_0_EventRouter"
    Targets="myserver"
    URI="BEA_POWERENTERPRISE_3_0_EventRouter.war" />
  <WebAppComponent Name="BEA_POWERENTERPRISE_3_0_Web"
    Targets="myserver" URI="BEA_POWERENTERPRISE_3_0_Web.war" />
</Application>
```

2. 以下の JAR ファイルを WebLogic Server のクラスパスに追加します。

- `WLI_HOME\lib\hlcommon.jar`
- `WLI_HOME\lib\mekshared.jar`
- `WLI_HOME\lib\powerapi.jar`
- `WLI_HOME\adapters\powerenterprise\lib\BEA_POWERENTERPRISE_3_0.jar`

UNIX システムの場合

作業を開始する前に、アダプタの EAR ファイルおよび JAR ファイルをコンピュータのどこに置くかを決めておく必要があります。デフォルトでは、これらのファイルは `WLI_HOME/adapters/powerenterprise/lib` に配置されます。

1. 以下のリストで示すように、アダプタの EAR ファイルをドメインの `config.xml` ファイルで宣言します。

コード リスト 3-2 UNIX システムの config.xml ファイルでのドメイン宣言

```
<Application Deployed="true" Name="BEA_POWERENTERPRISE_3_0"
Path="WLI_HOME/adapters/powerenterprise/lib/BEA_POWERENTERPRISE_3_0_EAR.ear">
  <WebAppComponent Name="BEA_POWERENTERPRISE_3_0_EventRouter"
    Targets="myserver" URI="BEA_POWERENTERPRISE_3_0_EventRouter.war"/>
  <ConnectorComponent Name="BEA_POWERENTERPRISE_3_0" Targets="myserver"
    URI="BEA_POWERENTERPRISE_3_0.rar"/>
  <WebAppComponent Name="BEA_POWERENTERPRISE_3_0_Web"Targets="myserver"
    URI="BEA_POWERENTERPRISE_3_0_Web.war"/>
</Application>
```

2. 以下の JAR ファイルを WebLogic Server のクラスパスに追加します。

- *WLI_HOME*/lib/hlcommon.jar
- *WLI_HOME*/lib/mekshared.jar
- *WLI_HOME*/lib/powerapi.jar
- *WLI_HOME*/adapters/powerenterprise/lib/BEA_POWERENTERPRISE_3_0.jar

例外処理

サービス メソッドの応答スキーマは、Code ステータスおよび Message ステータスの両方を含んでいる必要があります。

- Code は Success または Failure を返します。
- Message は、サービス呼び出しが開始されたときに Power.Enterprise! によって返されるメッセージと対応します。

発生する可能性があるすべての例外を処理するためには、すべての応答 XML からステータス コードを取得する XPath 式を作成します。

次に、サービス呼び出しに対する応答スキーマの例を示します。

```
<Status>
  <Code>Success </Code>
  <Message>Translation Request Queued Successfully.</Message>
</Status>
```

Code は、次の XPath 式を使用して応答スキーマから抽出できます。

```
"/Status/Code/text()"
```

4 Power.Enterprise! のコンフィグレーション

この節では、BEA EDI Connect for WebLogic Integration（個別に購入できます）および関連するその他のソフトウェアのインストールと設定方法について説明します。

- BEA EDI Connect for WebLogic Integration のインストール
- Power.Enterprise! のコンフィグレーション
- トレーディング パートナのコンフィグレーション
- XML と EDI データのマッピング
- 接続の設定および VAN との接続
- エクスチェンジ プロファイルのコンフィグレーション

BEA EDI Connect for WebLogic Integration のインストール

BEA EDI Connect for WebLogic Integration は Power.Enterprise! ソフトウェアとして提供され、個別にインストールする Power.Server! および Power.Client! という 2 つのパッケージから成ります。また、VAN 接続を管理する必要がある場合は Cleo A+ をインストールする必要があります。

この節では、次のタスクのための手順を説明します。

- ライセンスの取得
- インストールの前提条件の確認
- Power.Server! のインストール

- Power.Client! のインストール
- Get2Connect.Net への接続
- 非同期通信の設定
- ソフトウェア アップデートのダウンロード
- EDIドキュメント名の更新

ライセンスの取得

BEA EDI Connect for WebLogic Integration（個別に購入できます）をダウンロードした場合も CD からインストールした場合も、次の URL の BEA Web サイトから 30 日間の Power.Enterprise! 3.1 評価ライセンスを取得する必要があります。

http://commerce.bea.com/downloads/weblogic_integration.jsp

Power.Enterprise! 製品ライセンスは、BEA EDI Connect for WebLogic Integration の購入時に提供されます。詳細については、BEA 販売代理店にお問い合わせください。

また、VAN 接続を扱う場合には、Cleo A+ の 30 日間試用版をダウンロードしてインストールできます。現在サポート契約をお持ちの WebLogic Platform カスタマは、Cleo A+ のトライアルライセンスを BEA Customer Support Web サイトから変換できます。

http://support.bea.com/managed_content/third_party_product_support.jsp

注意： この Web サイトをご利用になる場合は登録が必要です。

BEA WebLogic Adapter for Power.Enterprise! には、Get2Connect.Net VAN へのアクセスが含まれています。Get2Connect.Net を使用するには、Peregrine カスタマサポート (<http://www.peregrine.com/>) にお問い合わせください。

インストールの前提条件の確認

Power.Server! および Power.Client! の両方を同一のマシンにインストールしなければならない場合は、これらを別々のディレクトリにインストールする必要があります。

- **Power.Server!** は実行時サーバです。このパッケージをインストールする前に、ご使用の環境が以下の要件を満たすことを確認してください。
 - オペレーティング システム : Microsoft Windows NT 4.0 SP6、Windows 2000 SP1、Sun Solaris 7、HP UX 11、または IBM AIX 4.3.3
 - Java : Sun JRE/SDK 1.3 以降
 - メモリ : 最小 512 MB、1 GB を推奨
 - ディスクの空き容量 : 100 MB
 - SQL データベース : **Power.Server!** は SQL データベースにアクセスする必要があります。**Power.Server!** は Oracle 8i、Microsoft SQL Server 2000、および IBM DB2 version 7.1 をサポートしています。インストール作業中に、データベースへのアクセスに必要なホスト名、接続ポート、データベース名、およびログイン情報を指定する必要があります。
- **Power.Client!** には **Power.Map!** および **Power.Manager!** が含まれています。このパッケージをインストールする前に、ご使用の環境が以下の要件を満たすことを確認してください。
 - オペレーティング システム : Microsoft Windows NT 4.0 SP4 または Windows 2000
 - Java : Sun JRE/SDK 1.3 以降
 - メモリ : 256 MB
 - ディスクの空き容量 : 50 MB
 - Internet Explorer 5.0
- 通信環境に非同期接続が必要であれば、次の URL の Cleo Communications から Cleo A+ ソフトウェアおよび評価用のスクリプトを入手してください。
http://www.cleo.com/partnerpages/bea_trial.asp

Power.Server! のインストール

Power.Server! ソフトウェアをインストールするには、このマニュアルの前の節にリストされている要件を満たしていなければなりません。**Power.Server!** のインストール概要については、次の URL にある『*WebLogic Platform インストールガイド*』の付録 A「BEA EDI Connect for WebLogic Integration のインストール」を参照してください。

<http://edocs.beasys.co.jp/e-docs/platform/docs70/install/apaedi.html>

Power.Client! のインストール

Power.Client! ソフトウェア（特に、Power.Map! および Power.Manager!）をインストールするには、4-1 ページの「BEA EDI Connect for WebLogic Integration のインストール」にリストされている要件を満たしていなければなりません。Power.Client! ソフトウェア インストールの概要については、次の URL にある『WebLogic Platform インストールガイド』の付録 A 「BEA EDI Connect for WebLogic Integration のインストール」を参照してください。

<http://edocs.beasys.co.jp/e-docs/platform/docs70/install/apaedi.html>

Get2Connect.Net への接続

Get2Connect.Net を使用するには、Power.Enterprise! ソフトウェアがインストール済みでなければなりません。また、Get2Connect.Net のトレーディング パートナとして登録するには、有効な製品シリアル番号（PSN）が必要です。有効な PSN を取得するには、BEA カスタマ サポートにお問い合わせください。

非同期通信の設定

通信環境に非同期接続が必要であれば、次の URL の Cleo Communications から Cleo A+ ソフトウェアおよび評価用のスクリプトを入手してください。

http://www.cleo.com/partnerpages/bea_trial.asp

ソフトウェア アップデートのダウンロード

Power.Enterprise! は、BEA カスタマ サポート Web サイトで提供されるダウンロード可能なソフトウェア パッチを使用してアップデートできるように設計されています。

Power.Enterprise! のソフトウェア パッチをインストールするには、次の作業が必要です。

1. Power.Manager! で、[Self Service] をクリックし、次に [Software Update] をクリックします。
2. 手順 1、[Select Software] をクリックします。Web ブラウザが起動されて、BEA カスタマ サポート Web ページに自動的に接続します。このページには、ダウンロードできるソフトウェア アップデート ファイルがリストされています。これらのアップデートのインストール手順は、この Web ページに掲載されています。

EDI ドキュメント名の更新

Power.Enterprise! 製品には、多くの標準的な EDI ドキュメントのための定義が含まれています。BEA は、BEA カスタマ サポート を通じて新しい定義も提供しています。

BEA カスタマ サポート Web サイトからダウンロードした新しい標準をインストールするには、次の作業が必要です。

1. Power.Manager! で、[Self Service] をクリックし、次に [Standards Update] をクリックします。
2. 手順 1、[Select Standards] をクリックします。Web ブラウザが起動されて、BEA カスタマ サポート Web ページに自動的に接続します。このページには、ダウンロードできる EDI 定義ファイルがリストされています。
3. 必要な EDI 定義ファイルを選択してダウンロードします。
4. ダウンロードしたファイルを次のディレクトリに置きます。
`Power_Server_install_dir\downloads\metadata\`
5. 定義ファイルがインストールされた Power.Server! を停止し、再起動します。
6. ダウンロードしたファイルを次のディレクトリから削除します。
`Power_Server_install_dir\downloads\metadata\`
7. Power.Map! 内からドキュメント名にアクセスするには、サーバからドキュメント名を確認してください。

Power.Server! の起動

この節では以下の方法について説明します。

- Windows での Power.Server! の起動
- UNIX での Power.Server! の起動
- サーバへの最初の接続

Windows での Power.Server! の起動

Windows プラットフォーム上で Power.Server! を起動する手順は次のとおりです。

1. [スタート | プログラム | Power.It | Power.Server!] の順に選択します。
2. システムによってはサーバの起動に 10 秒ほど時間がかかります。

UNIX での Power.Server! の起動

UNIX プラットフォーム上で Power.Server! を起動する手順は次のとおりです。

1. 次のディレクトリに移動します。

```
Power_Server_Install_Dir/jre/bin
```

パス名に含まれる *Power_Server_Install_DIR* が、Power.Server! ソフトウェアをインストールしたディレクトリです。

2. 次のコマンドを実行します。

```
rmiregistry
```

ポート 1099 で RMI レジストリが呼び出されます。

3. 次のディレクトリに移動します。

```
Power_Server_Install_Dir/bin
```

4. 次のコマンドを実行します。

```
start_server.sh
```

Power.Server! ソフトウェアが起動します。

注意： サーバが正常に起動したか確認するには、
`Power_Server_Install_Dir/logs/powerserver.log` ファイルの
`Translation server started on date` 行の内容を検証します。`date` に
は、サーバを起動した日付と時間が入力されています。

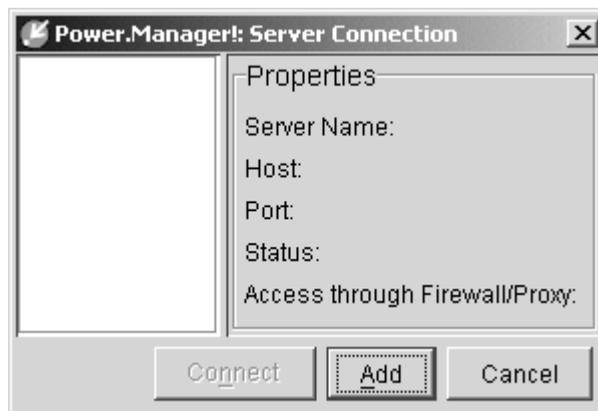
サーバへの最初の接続

Power.Manager! または Power.Map! にログインする前に、Power.Server! との接続を作成する必要があります。以降は、同じ手順を使用して Power.Server! の他のインスタンスに接続できます。複数の接続を作成することで、単一のコンソールから Power.Server! の複数のインスタンスを管理できます。すべての接続データは永続的です。いったん作成された接続は、Power.Manager! を起動すれば常に使用できるようになります。

接続を作成するには、次の作業が必要です。

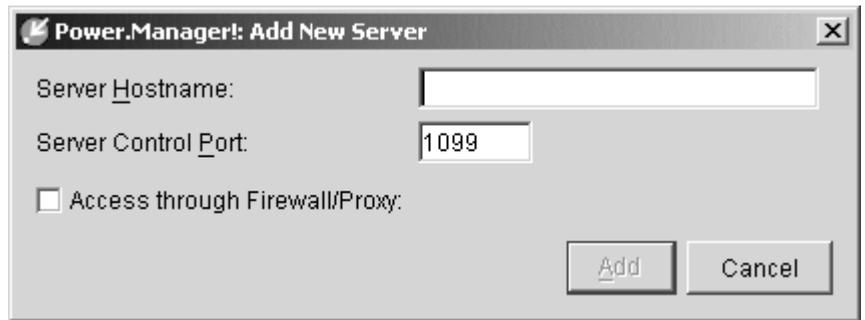
1. [スタート | プログラム | Power.It] を選択して、Power.Manager! または Power.Map にログインします。適切なアプリケーションを選択します。サーバ接続を作成するには、接続先のマシンで Power.Server! が実行中でなければなりません。

図 4-1 [Power.Manager! Server Connection] ダイアログ ボックス



2. [Add] をクリックして、新しい Power.Server! 接続を追加します。

図 4-2 Power.Manager! の [Add New Server] ダイアログ ボックス



3. 追加する Power.Server! の [Server Hostname] および [Server Control Port] を入力します。[Add] をクリックします。Power.Server! の定義済みインスタンスのリストが表示されます。
4. 定義済み接続のリストから、接続する Power.Server! のインスタンスを選択します。[Connect] をクリックします。[Power.Manager! Server Connection] ダイアログ ボックスが表示されます。

図 4-3 [Power.Manager! Server Connection] ダイアログ ボックス



Power.Enterprise! のコンフィグレーション

Power.Enterprise! では、BEA WebLogic Adapter for Power.Enterprise! への複数の接続をコンフィグレーションすることができます。個々の接続は、Power.Enterprise! へデータを送信するもの、あるいは Power.Enterprise! からデータを受信するもののどちらも、別々の RMI 接続として定義されます。Power.Enterprise! に必要な他のすべてのコンフィグレーションパラメータは、トレーディング パートナ、マップ、外部接続、およびエクステンジプロファイルを定義するように設計されています。

Power.Enterprise! のコンフィグレーションを最初に行うときは、以下のエンティティを列挙順にコンフィグレーションしてください。

- トレーディング パートナ
- マップ
- 接続
- エクステンジプロファイル

トレーディング パートナのコンフィグレーション

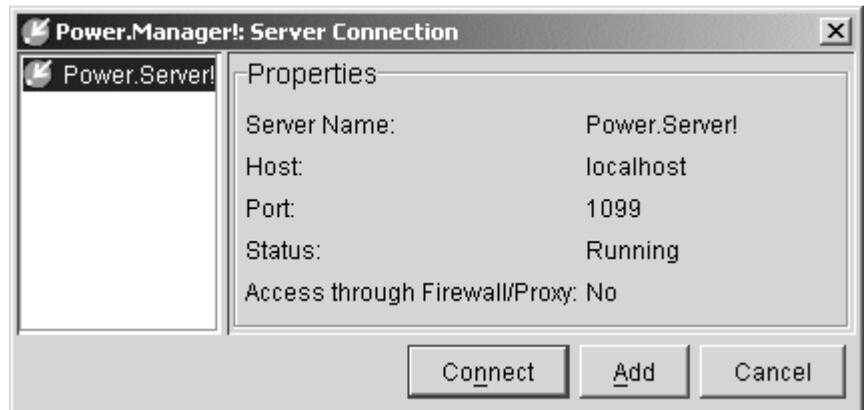
WebLogic Integration で EDI Integration を使用するには、Power.Enterprise! および WebLogic Integration の両方にトレーディング パートナのデータを入力する必要があります。トレーディング パートナのデータを Power.Enterprise! へ移行したり、Power.Enterprise! から読み出したりするメカニズムは存在しません。そのため、WebLogic Integration に登録されるすべてのトレーディング パートナのデータは、Power.Enterprise! にも入力される必要があります。

トランザクションに関係するすべてのトレーディング パートナおよび自身の会社について、Power.Enterprise! トレーディング パートナプロファイルを作成する必要があります。

新しいトレーディング パートナを作成するには、次の作業が必要です。

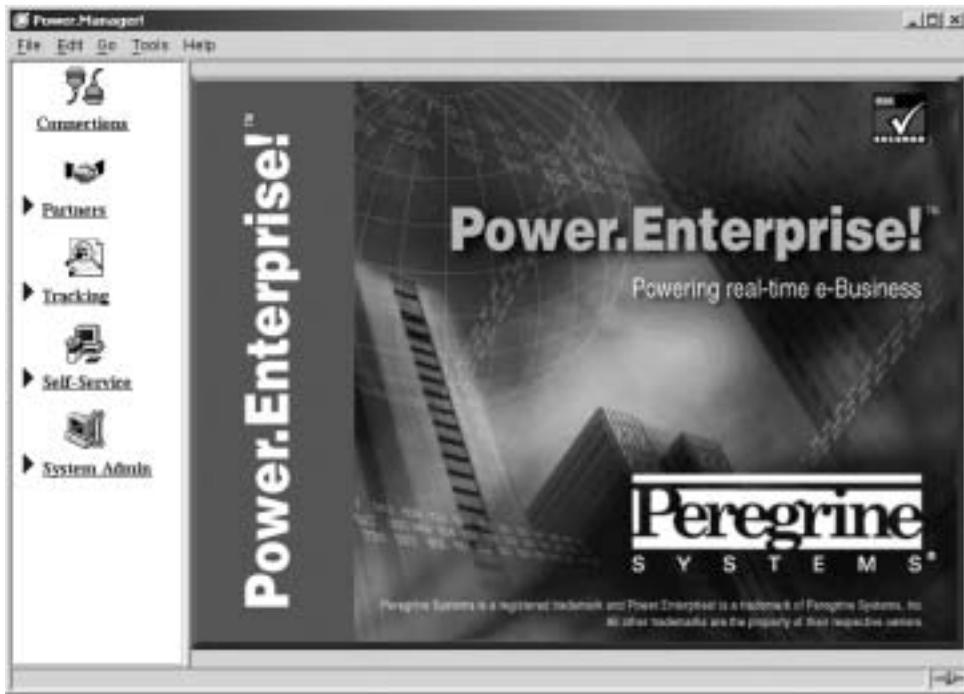
1. Power.Manager! を起動します。Power.Server! の定義済みインスタンスのリストが表示されます。
2. 接続する Power.Server! のインスタンスを選択します。Power.Server! 接続が定義されていない場合は、4-7 ページの「サーバへの最初の接続」を参照してください。[Power.Manager! Server Connection] ダイアログ ボックスが表示されます。

図 4-4 [Power.Manager! Server Connection] ダイアログ ボックス



3. [Connect] をクリックします。Power.Manager! のメイン ウィンドウが表示されます。

図 4-5 Power.Manager!



4. トレーディング パートナの企業プロフィールを設定するには、[Partners | Company Profiles] を選択します。[Company Profiles] ウィンドウが表示されます。

図 4-6 企業プロフィールのリスト



5. [New Company] をクリックします。トレーディング パートナについて必要な情報を入力します。

トレーディング パートナの作成および保守の詳細については、『*Power.Manager! User Reference Manual*』の第4章を参照してください。

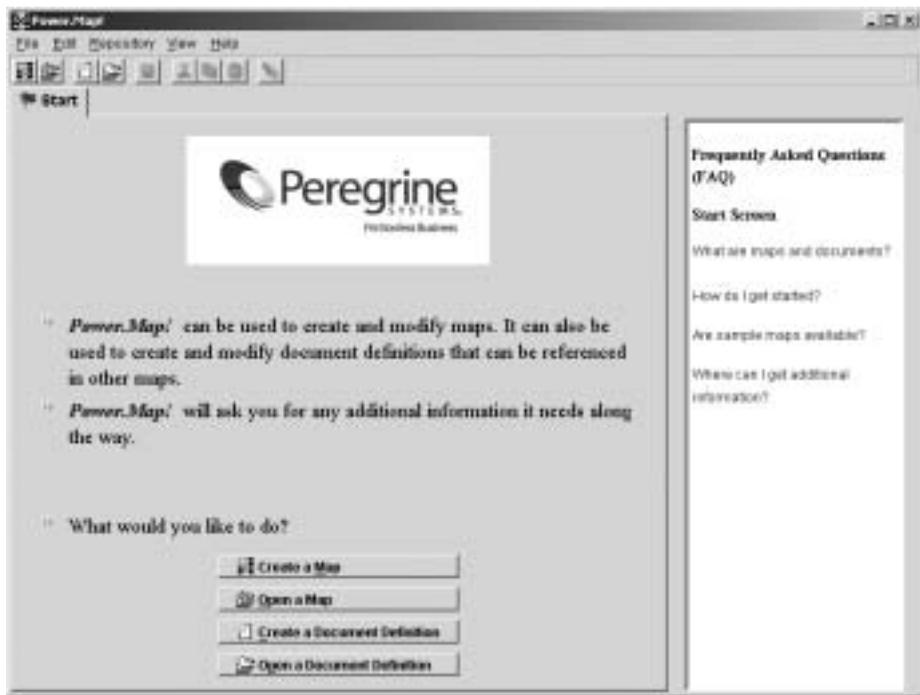
XML と EDI データのマッピング

Power.Enterprise! の主要な目的の1つは、メッセージを標準的なデータフォーマットから標準的な EDI メッセージフォーマットへ変換する処理を管理できるようにすることです。そのためには、選択しているデータフォーマットと対応する EDI メッセージとの間のマップを作成する必要があります。こうしたマップは単方向です。つまり、マップでは単方向の処理のみがサポートされます。こ

のため、特定の XML メッセージフォーマットと特定の EDI メッセージフォーマットの間でデータを双方向に変換できるようにする必要がある場合は、マップを2つ作成しなければなりません。1つは発信変換用（XML から EDI）、そしてもう1つは着信変換用（EDI から XML）です。

そのような複数のマップは簡単に作成できます。Power.Map! は標準的なドキュメント定義の多くに対応していて、フラットファイル、データベーステーブル、XML、およびすべての主要な EDI フォーマット（X12、EDIFACT、および TRADACOMS）などを扱えます。Power.Map! の内部で 사용되는ドキュメント定義は、ドキュメントの要素ごとの名前、その構造、および格納されるデータです。このため、標準的なドキュメント定義を使用してマップの枠組みを決めれば、2つのドキュメント間の個々のデータ要素は、ドラッグアンドドロップインタフェースを使用して簡単に接続できるようになります。

図 4-7 Power.Map! メイン ウィンドウ



マップ作成の詳細については、『Power.Map! User Reference Manual』を参照してください。

マップに対する作業は、Power.Server! のインスタンスに接続していなくても行うことができます。しかし、リポジトリ関係のコマンドを使用しようとするとき、[Server Connection] ダイアログ ボックスが表示されて、リポジトリのコマンドを実行するためにサーバへ接続するように求められます。サーバ接続の定義の詳細については、4-7 ページの「サーバへの最初の接続」を参照してください。

ローカル システム上で作成したマップについては、いつでもコピーを保存できます。[File | Save] を選択してから、ダイアログ ボックスにマップ ファイルのわかりやすい名前を入力してください。マップは JAR ファイルとして格納されます。

マップを実際に使用するには、マップをサーバにプロモートする必要があります。マップをサーバに送信するには、[Finish] タブを選択し、タブの下部にある [Promote] をクリックします。サーバへ接続していない場合は、接続を確立してからでないとマップをプロモートできません。

接続の設定および VAN との接続

接続は、Power.Server! を出入りするメッセージトラフィックがどのように配信されるかを定義します。BEA WebLogic Adapter for Power.Enterprise! を使用するには、最少で 4 種類の接続のコンフィグレーションを行う必要があります。

- BEA WebLogic Adapter for Power.Enterprise! からメッセージを受信するソース RMI 接続
- メッセージを Power.Enterprise! から BEA WebLogic Adapter for Power.Enterprise! へ送信するターゲット RMI 接続
- メッセージを Power.Server! からトレーディング パートナへ送信するターゲット接続
- トレーディング パートナからのメッセージを受信するソース接続

これらの接続のうち、ソース RMI 接続およびターゲット RMI 接続が BEA WebLogic Integration との通信に使用されます。これら 2 つの接続は、すべての EDI トランザクションで使用されます。

RMI 接続の作成

ソース RMI 接続およびターゲット RMI 接続を設定するには、次の作業が必要です。

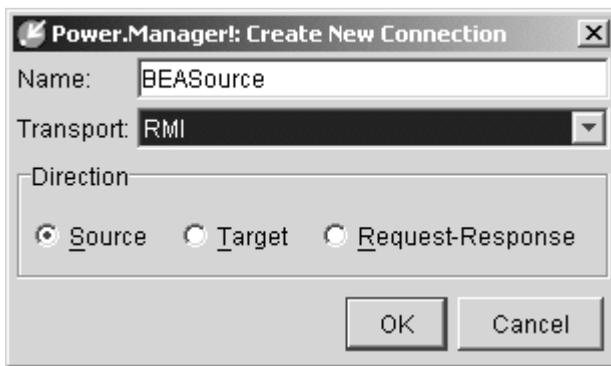
1. Power.Manager! で、[Connections] を選択します。接続のリストを表示できるウィンドウが表示されます。

図 4-8 Power.Manager! の接続リスト



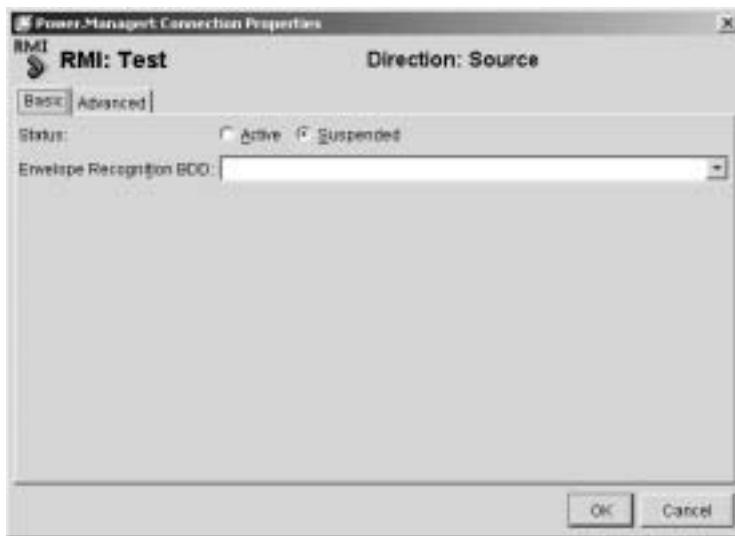
2. ソース RMI 接続を作成します。最初に、[Add New] をクリックします。[Create New Connection] ダイアログ ボックスが表示されます。

図 4-9 [Create New Connection] ダイアログ ボックス



3. 名前を BEASource とし、[Transport] は [RMI] を選択してから、[Direction] では [Source] をクリックします。[OK] をクリックします。[Connectrion Properties] ダイアログ ボックスが表示されます。

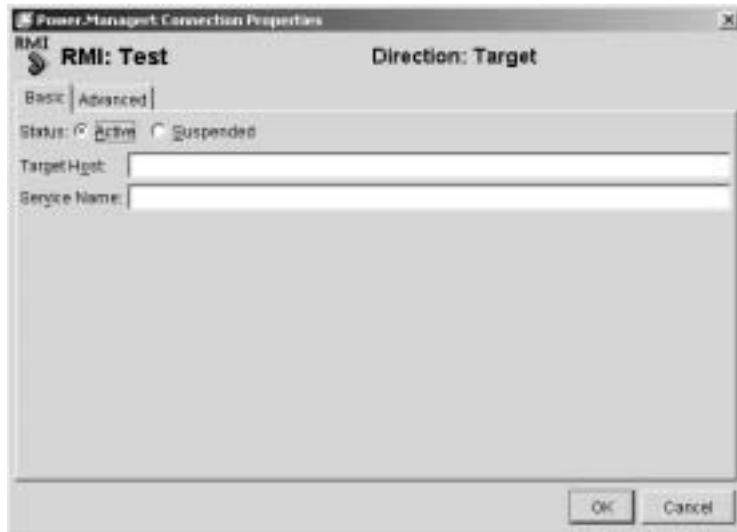
図 4-10 着信 RMI のプロパティ



4. [Status] を [Active] に設定し、[Envelope Recognition BDD] には XMLEnvelope と入力します。[OK] をクリックします。

5. 発信 RMI 接続を作成します。最初に、[Add New] をクリックします。
[Create New Connection] ダイアログ ボックスが表示されます。
6. 名前を BEATarget とし、[Transport] は [RMI] を選択してから、[Direction] では [Target] をクリックします。[OK] をクリックします。[Connection Properties] ダイアログ ボックスが表示されます。

図 4-11 発信 RMI のプロパティ



7. [Status] を [Active] に設定し、[Target] には localhost と入力してから、
[Service Name] には edi と入力します。[OK] をクリックします。

トレーディング パートナ接続の作成

他の 2 つの接続、つまり着信および発信のトレーディング パートナ接続は、各トレーディング パートナに固有の接続です。このため、各トレーディング パートナについて一意な接続のペアを定義する必要があります。各ペアを設定するときには、自分自身およびトレーディング パートナが使用する転送方式を選択する必要があります。推奨される転送方式オプションは、以下のとおりです。

- HTTP: トレーディング パートナと EDI メッセージのエクステンションを行う上で標準的な HTTP/HTTPS 転送プロトコル

- **File:** EDI メッセージをファイルに書き込んだり、ファイルからメッセージを読み込んだりするオプション。プロセスでのファイルの読み書き以後について別の転送メカニズムを提供する必要があります。
- **FTP:** トレーディング パートナとのメッセージのエクステンジに FTP サーバを使用するオプション
- **Get2Connect:** Peregrine Systems 社によって運営されている VAN。この VAN への接続は Power.Enterprise! ソフトウェアに含まれていますが、Get2Connect VAN を使用するには Peregrine Systems 社へ問い合わせでサインアップする必要があります。
- **Cleo A+:** 非同期通信が必要な VAN を通じて接続する場合はこのオプションを使用します。詳細については、第 2 章「アーキテクチャ」の「VAN および Network との接続」を参照してください。

他の接続オプションも指定できます。オプションの詳細については、『Power.Manager! User Reference Manual』の第 3 章を参照してください。

エクステンジ プロファイルのコンフィグレーション

エクステンジプロファイルは、トレーディング パートナが Power.Server! を通じてメッセージを交換する方法を指定します。Power.Server! は一種の EDI ゲートウェイなので、メッセージを受信する最終ポイントではありません。エクステンジプロファイルは、Power.Server! がメッセージを受信し、変換して、送り先へ送信する際の方法を示します。たとえば、エクステンジプロファイルでは、トレーディング パートナからの EDI メッセージを受信して、EDI メッセージの XML メッセージに変換し、その XML メッセージを BEA WebLogic Adapter for Power.Enterprise! へ送信するまでを定義することも可能です。

エクステンジプロファイルは、必要な情報があらかじめ定義されていないと作成できません。特に、自身のエクステンジ マップ、自身のトレーディング パートナ プロファイル、およびソース接続とターゲット接続の両方を作成して、サーバへプロモートしておく必要があります。ソース接続は Power.Server! へ

メッセージをポストし、ターゲット接続は **Power.Server!** からメッセージを受信します。たとえば、トレーディング パートナへメッセージを送信するためのエクステンジプロファイルを作成するには、以下の情報が必要です。

- 送信する EDI メッセージの XML-to-EDI トランザクション マップ
- 自身のトレーディング パートナ プロファイル
- 4-14 ページの「接続の設定および VAN との接続」で作成した BEA 発信 RMI 接続
- トレーディング パートナへの発信接続

この他にも、必要な情報があります。詳細については、『*Power.Manager! User Reference Manual*』の第 4 章を参照してください。着信および発信のエクステンジプロファイルの例については、このマニュアルの第 6 章「EDI のサンプル」を参照してください。

5 EDI Integration のコンフィグレーション

BEA WebLogic EDI Integration は、コンフィグレーションが完了しないと使用できません。EDI Integration の正常な動作は、BEA EDI Connect for WebLogic Integration に依存しているため、トレーディング パートナと通信を行うには Power.Enterprise! を適切にコンフィグレーションする必要があります。

共通のコンフィグレーション

この節では、トレーディング パートナとの特定のドキュメント交換をコンフィグレーションする前に済ませておく必要がある、共通のコンフィグレーション情報について説明します。

トレーディング パートナのコンフィグレーション

個々のトランザクションのコンフィグレーションを行う前に、自身の会社を含むすべてのトレーディング パートナのコンフィグレーションを Power.Manager! で行う必要があります。トレーディング パートナのコンフィグレーションの詳細については、以下のトピックを参照してください。

- 4-9 ページの「トレーディング パートナのコンフィグレーション」
- 『Power.Manager! User Reference Manual』の第4章

VAN/トレーディング パートナ接続のコンフィグレーション

トレーディング パートナとの接続のコンフィグレーションも、個々のドキュメント交換のコンフィグレーションよりも先に行う必要があります。以下のトピックを参照してください

- 4-14 ページの「接続の設定および VAN との接続」
- 『Power.Manager! User Reference Manual』の第3章

この時点でプロダクション トレーディング パートナとの接続を設定する必要はありませんが、開発およびテストの目的でファイル コネクタに関するプロダクション トレーディング パートナのコンフィグレーションが必要な場合もあります。

EDI ドキュメントを受信する EDI Integration のコンフィグレーション

この節では、BEA WebLogic Adapter for Power.Enterprise! 3.1 が EDI ドキュメントを受信して、それを XML ドキュメントとして WebLogic Integration プロセスへ送信できるように、EDI Integration をコンフィグレーションする方法を説明します。

準備

- 受信するドキュメントの EDI 標準、バージョン、および文書型を決めます。たとえば、X12 850 ドキュメント v4010 を選択した場合、標準は X12、文書型は 850、バージョンは 4010 となります。
- 内部で使用する XML フォーマットを定義します。この XML フォーマットは、使用する EDI ドキュメントにあるフィールドの詳細なデータ マップを含んでいる必要があります。

- XML ドキュメントで使用される DTD および XSD を作成します。DTD および XSD の両方を作成する必要があるのは、Power.Enterprise! および WebLogic Integration の使用するデータ ファイル フォーマットが、この段階では異なるためです。WebLogic Integration では XSD（提供できる場合）が必要ですが、Power.Enterprise! では DTD が必要です。Power.Enterprise! では、DTD または XSD をインポートした後で、マップ定義から DTD または XSD をエクスポートするためのユーティリティが一切ない点に注意してください。必要な DTD および XSD ファイルを別に作成し直してください。

Power.Enterprise! 内でのコンフィグレーション

1. XML ドキュメントと EDI ドキュメントの間のマップを作成します。
Power.Map! に DTD をインポートして、XML ドキュメントの標準をロードします。適切な EDI ドキュメントは、あらかじめ Power.Server! のリポジトリに用意されています。必要な EDI ドキュメントがない場合は、BEA カスタマサポートにお問い合わせください。この作業の詳細については、以下のトピックを参照してください。
 - 4-12 ページの「XML と EDI データのマッピング」
 - 『Power.Map! User Reference Manual』の第 3 章
2. 着信接続を定義します。この接続では、既に定義されている、トレーディング パートナの通信プロトコルを使用する必要があります。詳細については、4-14 ページの「接続の設定および VAN との接続」を参照してください。
3. RMI ターゲット接続を定義します。この接続は、WebLogic Integration のプロセス エンジンとの通信に使用されます。
4. 4-18 ページの「エクスチェンジプロファイルのコンフィグレーション」で説明した手順に従って、エクスチェンジプロファイルを定義します。このエクスチェンジプロファイルでは、トレーディング パートナからの着信接続を RMI ターゲット接続に接続する必要があります。

BEA WebLogic Integration 内でのコンフィグレーション

1. 「BEA WebLogic Adapter for Power.Enterprise! のコンフィグレーション」で説明した手順に従って、アプリケーションビューを作成します。
2. 3-7 ページの「イベントのコンフィグレーション」で説明した手順に従って、作成したアプリケーションビューのためのイベントを作成します。以下のパラメータを設定してください。
 - **Event Name:** イベントの内容を表す名前（たとえば、ReceivePurchaseOrder）
 - **Event Description:** イベントの詳細を示すテキスト
 - **RMI Service Name:** 「Power.Enterprise! 内でのコンフィグレーション」の節で説明した Power.Enterprise! RMI ターゲット接続の設定で定義された RMI サービス名と一致している必要があります。
 - **XSD:** 受信する XML メッセージの XML スキーマ（省略可能）
 - **XML Root Element:** XML スキーマの XML ルート要素（省略可能）
3. 作成および定義したアプリケーションビューをデプロイします。

Power.Server! のインスタンスが実行中であることを確認します。この時点でアプリケーションビューは、**Application Integration** プラグインを使用する **WebLogic Integration** プロセス エンジンで利用可能である必要があります。**Application Integration** プラグインの詳細については、『*Application Integration 入門*』の「BPM の Application Integration プラグインについて」を参照してください。

注意： 必ず、Power.Server! を起動してから **WebLogic Integration** または **WebLogic Server** を起動してください。WebLogic Integration のコネクタは初期化の時点で Power.Server! が実行中であることを前提にしているため、初期化の際に Power.Server! が実行中でないと例外を送出します。

EDI ドキュメントを送信する EDI Integration のコンフィグレーション

この節では、WebLogic Integration プロセス エンジンにある XML ドキュメントをソースを使用して EDI ドキュメントを送信するための EDI Integration をコンフィグレーションする方法について説明します。

準備

1. 受信するドキュメントの EDI 標準、バージョン、および文書型を決めます。たとえば、X12 850 ドキュメント v4010 を選択した場合、標準は X12、文書型は 850、バージョンは 4010 となります。
2. 内部で使用する XML フォーマットを定義します。この XML フォーマットは、使用する EDI ドキュメントにあるフィールドの詳細なデータ マップを含んでいる必要があります。
3. XML ドキュメントで使用される DTD および XSD を作成します。DTD および XSD の両方を作成する必要があるのは、Power.Enterprise! および WebLogic Integration の使用するデータ ファイル フォーマットが、この段階では異なるためです。WebLogic Integration では XSD（提供できる場合）が必要ですが、Power.Enterprise! では DTD が必要です。Power.Enterprise! では、DTD または XSD をインポートした後で、マップ定義から DTD または XSD をエクスポートするために使用するユーティリティが一切ない点に注意してください。必要な DTD および XSD ファイルを別に作成し直してください。

Power.Enterprise! 内でのコンフィグレーション

1. XML ドキュメントと EDI ドキュメントの間のマップを作成します。Power.Map! に DTD をインポートして、XML ドキュメントの標準をロードします。適切な EDI ドキュメントは、あらかじめ Power.Server! のリポジト

りに用意されています。必要な EDI ドキュメントがない場合は、BEA カスタマサポートにお問い合わせください。この作業の詳細については、以下のトピックを参照してください。

- 4-12 ページの「XML と EDI データのマッピング」
 - 『*Power.Map! User Reference Manual*』の第3章
2. 発信接続を定義します。この接続では、既に定義されている、トレーディング パートナの通信プロトコルを使用する必要があります。詳細については、4-14 ページの「接続の設定および VAN との接続」を参照してください。
 3. RMI ソース接続を定義します。この接続は、WebLogic Integration のプロセス エンジンとの通信に使用されます。
 4. 4-18 ページの「エクスチェンジプロファイルのコンフィグレーション」で説明した手順に従って、エクスチェンジプロファイルを定義します。このエクスチェンジプロファイルでは、RMI ソース接続をトレーディング パートナの発信接続に接続する必要があります。

BEA WebLogic Integration 内でのコンフィグレーション

1. このマニュアルの 3-3 ページの「BEA WebLogic Adapter for Power.Enterprise! のコンフィグレーション」で説明した手順に従って、アプリケーション ビューを作成します。
2. このマニュアルの 3-10 ページの「サービスのコンフィグレーション」で説明した手順に従って、作成したアプリケーション ビューのためのサービスを作成します。以下のパラメータを設定してください。
 - **Service Name:** イベントの内容を表す名前（たとえば、SendPurchaseOrder）
 - **Service Description:** サービスの詳細を示すテキスト
 - **Connection Name:** 「Power.Enterprise! 内でのコンフィグレーション」の節で説明した Power.Enterprise! RMI ターゲット接続の設定で定義された RMI サービス名と一致している必要があります。
 - **XSD:** 送信する XML メッセージの XML スキーマ（省略可能）

- XML Root Element: XML スキーマの XML ルート要素（省略可能）
- 作成および定義したアプリケーション ビューをデプロイします。

Power.Server! のインスタンスが実行中であることを確認します。この時点でアプリケーション ビューは、**Application Integration** プラグインを使用する **WebLogic Integration** プロセス エンジンで利用可能である必要があります。**Application Integration** プラグインの詳細については、『*Application Integration 入門*』の「BPM の **Application Integration** プラグインについて」を参照してください。

6 EDI のサンプル

この章では、EDI のサンプルのコンフィグレーションと操作について説明します。EDI サンプルは **Power.Server!** の適切なコンフィグレーションに依存しているので、**Power.Server!** のコンフィグレーションを済ませないとサンプルは実行できません。こうしたコンフィグレーションは、**BEA WebLogic Integration** で提供される他のサンプルを実行する際には必須ではありません。

この付録のトピックは以下のとおりです。

- サンプルの概要
- サンプルの設定と実行

サンプルの概要

この EDI サンプルは、単純な **Purchase Order** ビジネス プロセスを含んでいます。2 つのトレーディング パートナが関係します。

- バイヤー : **Advanced Networks Inc.** (略称 ANI)
- サプライヤー : **General Control Systems** (略称 GCS)

Advanced Networks は制御システム コンポーネントを **General Controls** から購入します。ANI は EDI を使用して **Purchase Orders** (発注書) を **GCS** へ送信し、サプライヤーである **GCS** から **Purchase Order Acknowledgement** (発注確認書) を受信します。

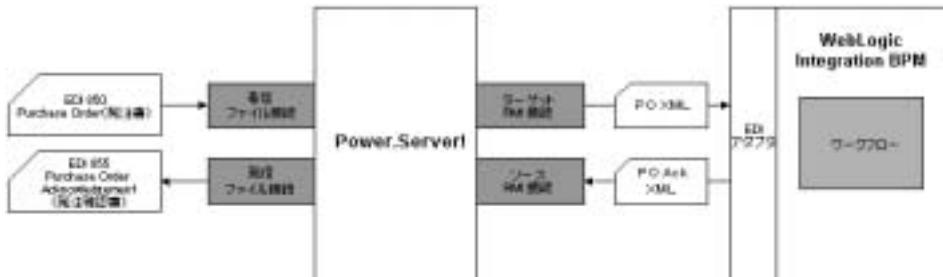
このサンプルは、サプライヤーである **GCS** が **BEA WebLogic Integration** を自身の企業間統合ソリューションとして使用すること、すなわち、トレーディング パートナと EDI ドキュメントを交換するためのフレームワークとして使用することを前提にしています。次の図に、**WebLogic EDI Integration** のアーキテクチャを示します。

図 6-1 EDI のサンプルのシナリオ



この EDI サンプルは、サプライヤである GCS が WebLogic Integration を EDI でトレーディング パートナと通信するための企業間統合ソリューションとして使用することを前提にしています。この EDI サンプルのアーキテクチャを図 6-2 に示します。

図 6-2 EDI サンプルのアーキテクチャ



このサンプルには、着信ファイル接続で Power.Server! によって受信される、着信の EDI 850 Purchase Order があります。イベント順序の詳細をまとめると次のようになります。

1. EDI 850 は XML フォーマットの Purchase Order に変換され、ターゲット RMI 接続で WebLogic Integration へ送信されます。
2. この XML ドキュメントは WebLogic Integration EDI アダプタによって受信されて、Application Integration プラグインを使用してアプリケーション統合イベントとしてサンプル ワークフローへ送信されます。

3. サンプル ワークフローは Purchase Order XML ドキュメントを Purchase Order Acknowledgement XML ドキュメントに変換してから、サービス呼び出しを使用して EDI アダプタ経由で Power.Server! へ送信します。
4. サービス呼び出しは RMI ソース接続を使用してドキュメントを Power.Server! へ送信します。
5. Purchase Order Acknowledgement XML ドキュメントは EDI 855 Purchase Order Acknowledgement へ変換されて、発信ファイル接続によってファイルシステムに書き込まれます。

サンプルの設定と実行

この節では、BEA WebLogic Integration EDI サンプルを実行するための要件および手順をリストします。

要件

EDI サンプルを実行する前に、BEA WebLogic Integration のすべての製品コンポーネントをインストールしておく必要があります。以下のコンポーネントも含まれます。

- BEA WebLogic Server 6.1 SP1 以降のバージョン
- BEA WebLogic Integration 7.0
- Power.Enterprise! 3.1 (Power.Manager! および Power.Map! で構成される Power.Server!、Power.Client! を含む)

注意： Power.Enterprise! ソフトウェアをインストールする前に、システムが要件を満たしており、必要なデータベースへのアクセス権を持っているか確認してください。これらの前提条件の詳細については、4-1 ページの「BEA EDI Connect for WebLogic Integration のインストール」を参照してください。

ハードウェアおよび OS の要件

EDI サンプルは、1 台のマシンにあるすべてのソフトウェア（Power.Server!、Power.Manager!、および Power.Map! を含む）へのアクセス権があることを前提にしているために、プロダクション環境の場合よりも OS のインストール要件が厳しくなっています。

- 最小メモリ容量：512MB
- OS：Windows NT 4.0（Service Pack 6）以降、または Windows 2000（Service Pack 1）

EDI サンプルのコンフィグレーション

EDI サンプルのコンフィグレーションおよび実行では、以下の手順を行う必要があります。

- 手順 1：Power.Server! の起動
- 手順 2：Power.Manager! の起動およびパートナーのコンフィグレーション
- 手順 3：Power.Map! の起動およびマップとアダプタのロード
- 手順 4：接続の設定
- 手順 5：エクスチェンジ プロファイルの設定
- 手順 6：ワークフローの設定
- 手順 7：アプリケーション ビューのデプロイ
- 手順 8：サンプルの実行

以下のファイルが、`WLI_HOME/samples/EDISample` にあります。

- `Readme.txt`
- `WLI_EDI_SAMPLE.pdf`
- `XMLEnvelope.jar`
- `EDI850toPOXML.jar`
- `POAckXMLtoEDI855.jar`

- EDISampleWorkflow.jar
- Edi850.edi

注意：必ず、Power.Server! を起動してから WebLogic Integration または WebLogic Server を起動してください。WebLogic Integration のコネクタは初期化の時点で Power.Server! が実行中であることを前提にしているため、初期化の際に Power.Server! が実行中でないと例外を送出します。

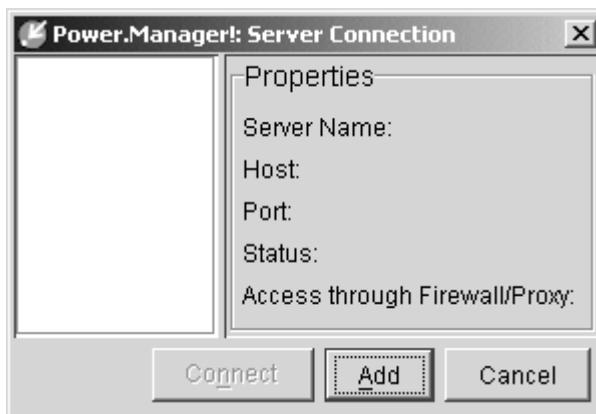
手順 1 : Power.Server! の起動

Power.Server! を起動するには、[スタート | プログラム | Power.It | Power.Server!] を選択します。サーバの起動には 10 秒ほど時間がかかります。

手順 2 : Power.Manager! の起動およびパートナのコンフィグレーション

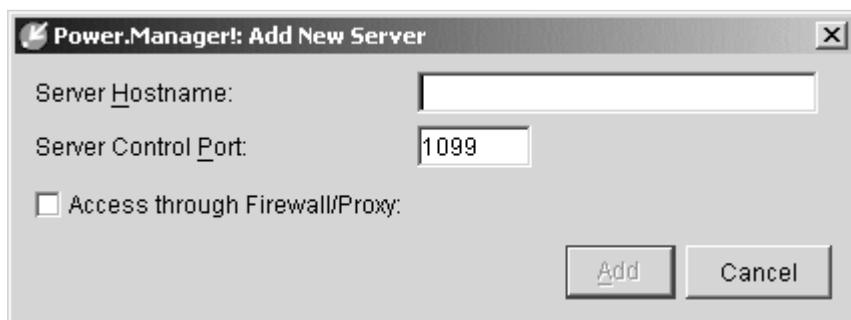
1. Power.Manager! を起動するには、[スタート | プログラム | Power.It | Power.Manager!] を選択します。
[Server Connection] ダイアログ ボックスが表示されます。

図 6-3 [Power.Manager! Server Connection] ダイアログ ボックス



2. 左側のペインにサーバがリストされていない場合は、[Add] をクリックします。[Add New Server] ダイアログ ボックスが表示されます。

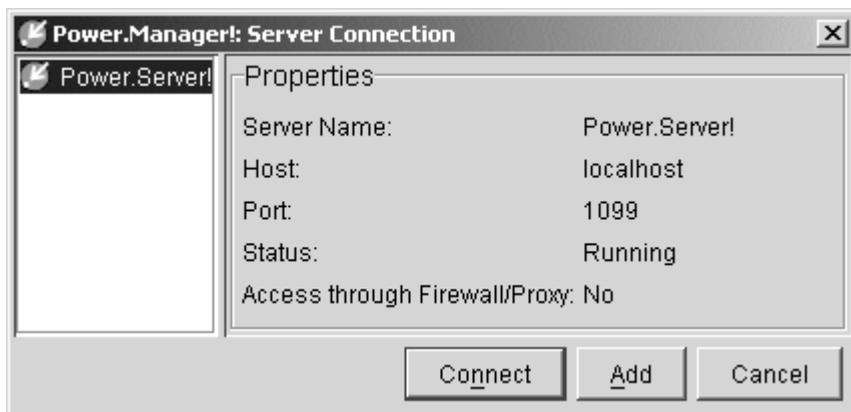
図 6-4 [Power.Manager! Add New Server] ダイアログ ボックス



EDI Server の [Server Hostname] および [Server Control Port] を入力します。
[Add] をクリックします。

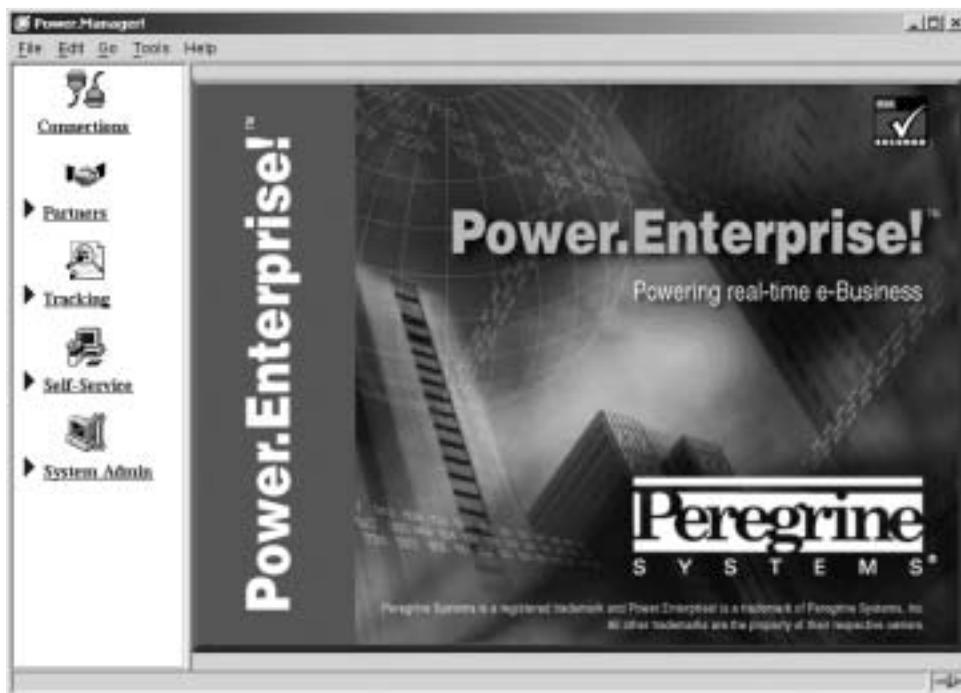
3. 接続する Power.Server! のインスタンスを選択してから、[Connect] をクリックします。[Server Connection] ダイアログ ボックスが表示されます。

図 6-5 [Power.Manager! Server Connection] ダイアログ ボックス



Power.Manager! のメイン ウィンドウが表示されます。

図 6-6 Power.Manager!



4. 企業プロフィールを設定します。[Partners] をクリックし、次に [Company Profiles] をクリックします。[Company Profiles] ウィンドウが表示されます。

図 6-7 企業プロフィールのリスト



5. GCS の企業プロフィールを作成します。[New Company] アイコン  をクリックします。

[Company Properties] ダイアログ ボックスが表示されます。

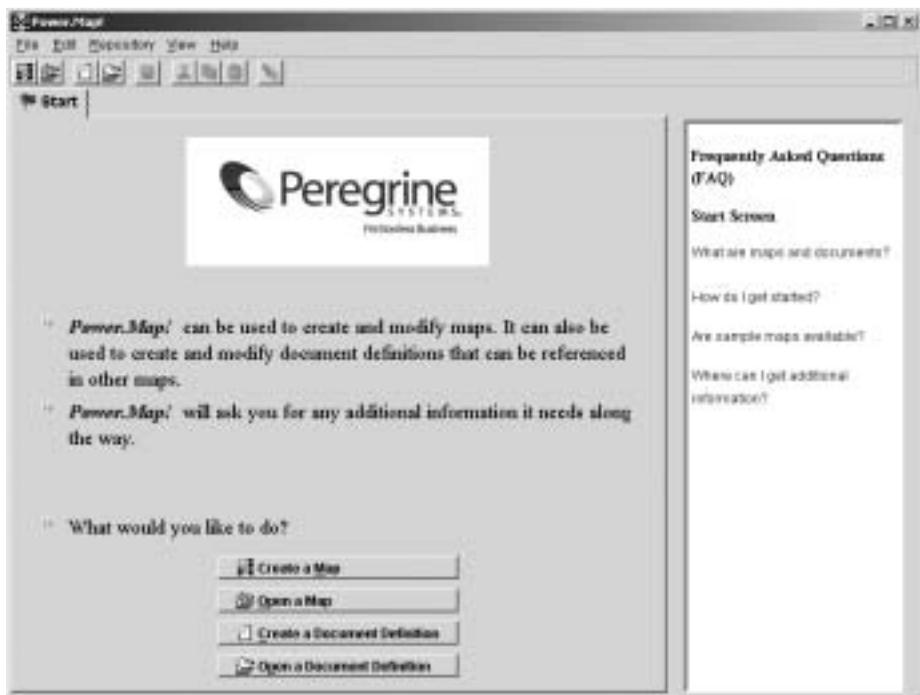
図 6-8 新しい企業プロフィールのための [Company Properties] ダイアログボックス

6. GCS に関する必要な情報を追加します。
7. ANI の企業プロフィールを作成します。[New Company] アイコン  をクリックします。
8. [Company Properties] ダイアログ ボックスが表示されたら、ANI に関する必要な情報を入力します。

手順 3 : Power.Map! の起動およびマップとアダプタのロード

1. Power.Map! を起動するには、[スタート | プログラム | Power.It | Power.Map!] を選択します。Power.Map! のメイン ウィンドウが表示されます。

図 6-9 Power.Map! メイン ウィンドウ

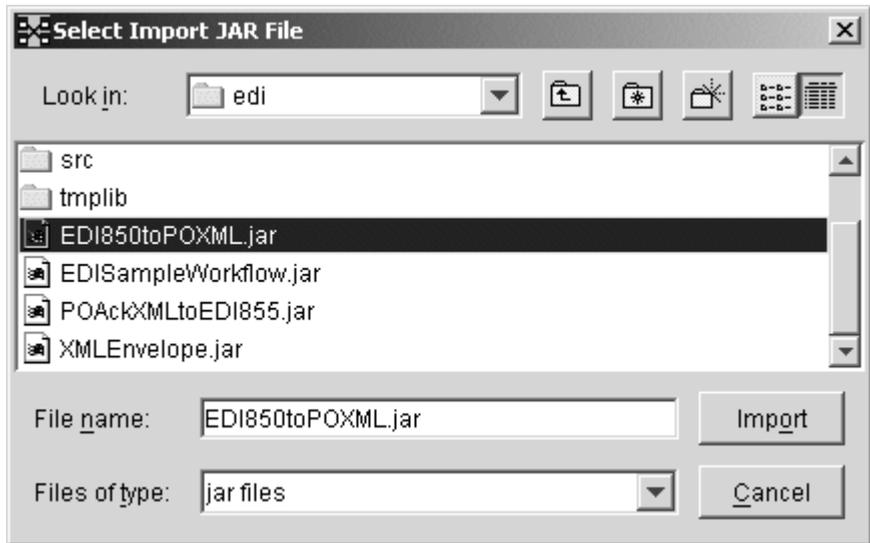


2. EDI 850 トランザクションから Purchase Order XML へ変換するためのマップをインポートします。[File | Import] を選択します。[Select Import JAR] ウィンドウで、次のファイルを選択します。

`WLI_HOME\samples\edi\EDI850toPOXML.jar`

[Select Import JAR File] ダイアログ ボックスが表示されます。

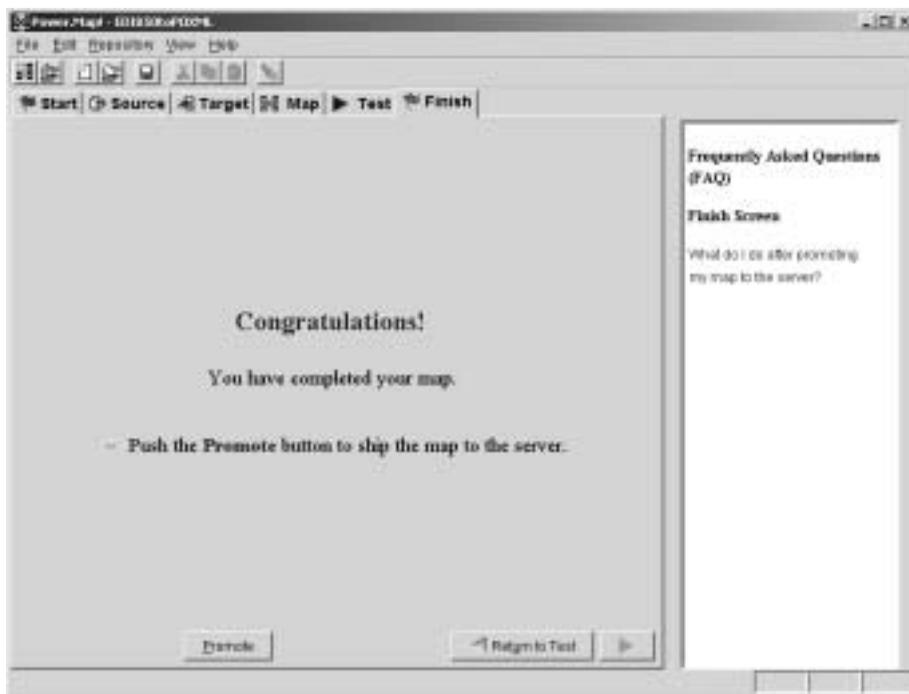
図 6-10 [Select Import JAR File] ダイアログ ボックス



[Import] ボタンをクリックします。

3. EDI850toPOXML マップ ファイルを開き、[File | Open Map] を選択します。続いて EDI850toPOXML マップを選択します。
4. 生成されたマップを Power.Server! へエクスポートします。[Finish] タブを選択し、次に [Promote] を選択します。

図 6-11 完成したマップのプロモート



5. Purchase Order Acknowledgement XML から EDI 855 トランザクションへ変換するためのマップをインポートします。[File | Import] を選択します。[Select Import JAR] ウィンドウで、次のファイルを選択します。

`WLI_HOME\samples\edi\POAckXMLtoEDI855.jar`

[Import] ボタンをクリックします。

6. [File | Open Map] を選択し、次に POAckXMLtoEDI855 マップを選択します。
7. 生成されたマップを Power.Server! へエクスポートします。[Finish] タブを選択し、[Promote] を選択します。[Move Map to Production] を選択してから [OK] をクリックします。
8. XML エンベロープをインポートします。[File | Import] を選択し、次のファイルをインポートします。

`WLI_HOME\samples\edi\XMLEnvelope.jar`

9. [File | Open Document Definition] を選択します。次に XMLEnvelope を選択し、[OK] をクリックします。
10. 結果を Power.Server! へエクスポートします。[Finish] タブを選択し、次に [Promote] を選択します。[Move Doc to Production] を選択してから [OK] をクリックします。
11. これで、Power.Map! での作業は終わりました。Power.Map! を終了してかまいません。

手順 4：接続の設定

1. Power.Manager! で、[Connections] をクリックします。[Connections] ウィンドウが表示されます。

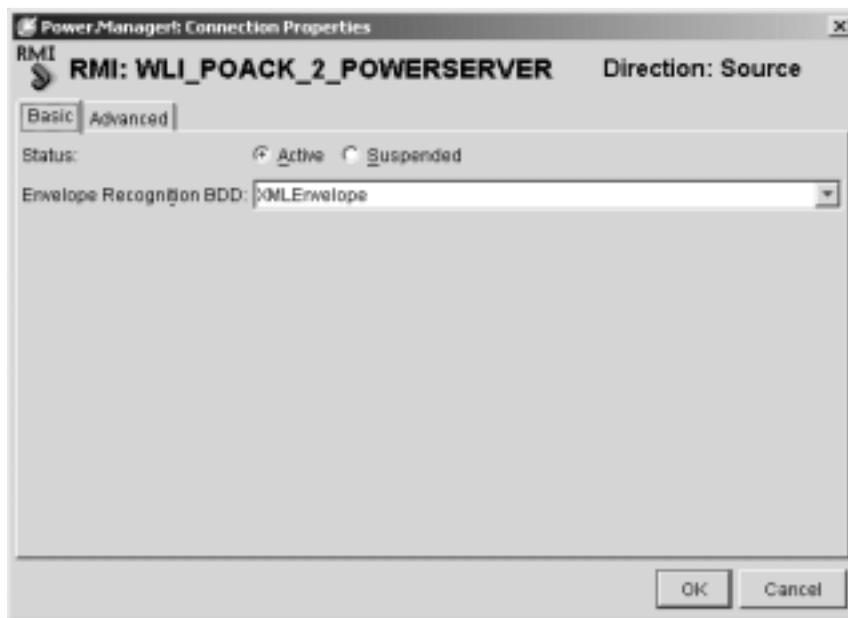
図 6-12 Power.Manager! の接続リスト



2. WebLogic Integraton から Power.Server! への RMI 接続を作成します。[Add New] アイコン  をクリックします。

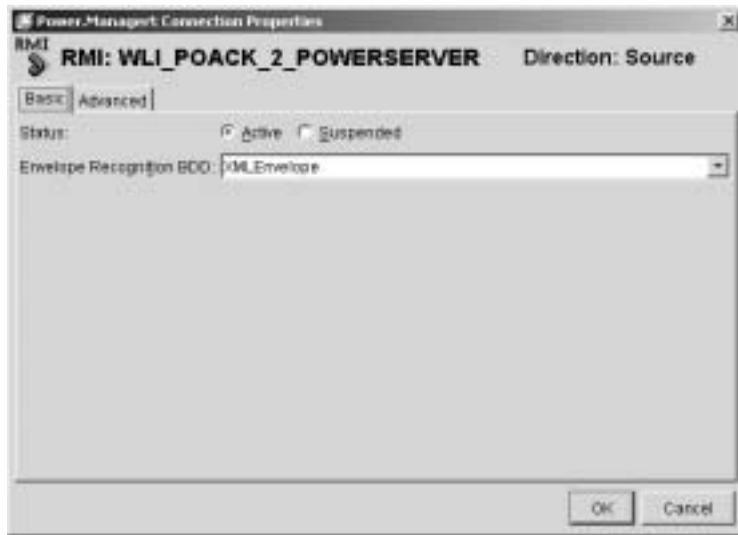
新しい接続を作成するためのダイアログ ボックスが表示されます。

図 6-13 [Create New Connection] ダイアログ ボックス



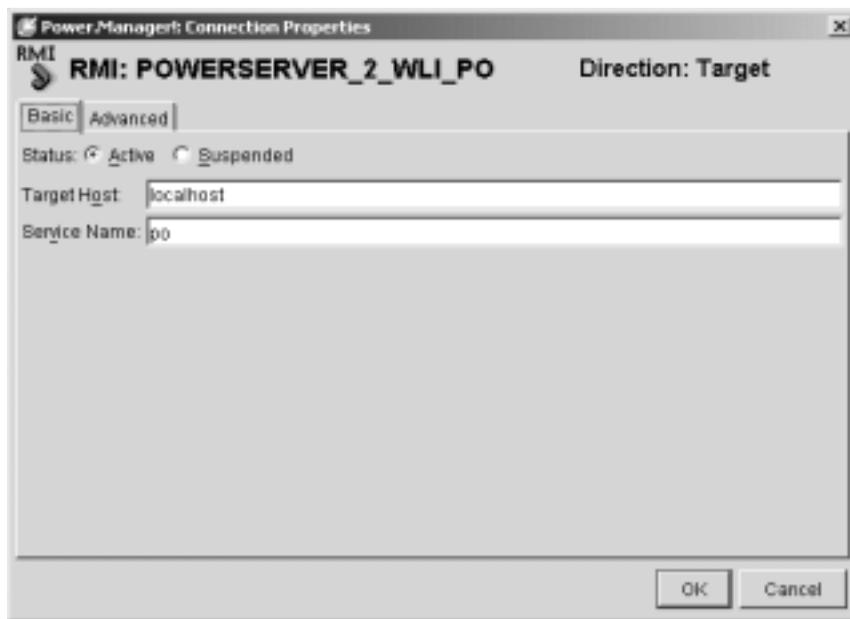
3. 名前を WLI_POACK_2_POWERSERVER とし、[Transport] は [RMI] を選択してから、[Direction] では [Source] をクリックします。[OK] をクリックします。
着信 RMI のプロパティを設定するためのダイアログ ボックスが表示されます。

図 6-14 着信 RMI のプロパティ



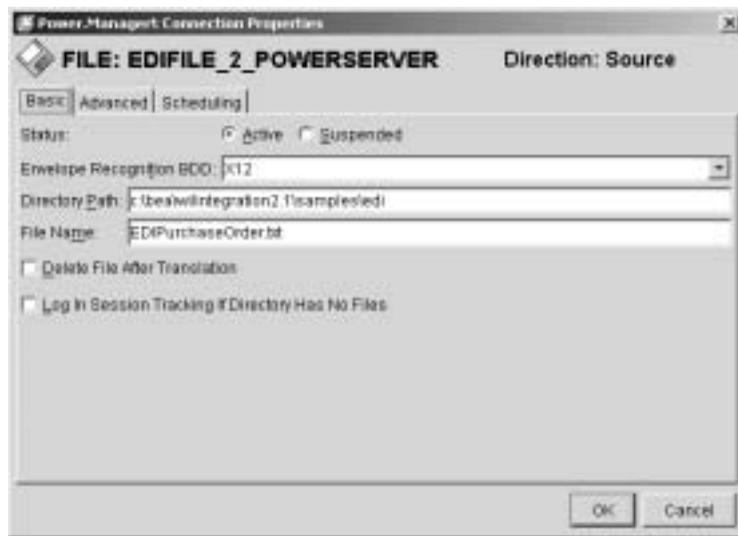
4. [Status] を [Active] に設定し、[Envelope Recognition BDD] には XMLEnvelope と入力してから、[OK] をクリックします。
5. Power.Server! から WebLogic Integraton への RMI 接続を作成します。[Add New] アイコン  をクリックします。
6. 名前を POWERSERVER_2_WLI_PO とし、[Transport] は [RMI] を選択してから、[Direction] では [Target] をクリックします。[OK] をクリックします。発信 RMI のプロパティを設定するためのダイアログ ボックスが表示されま

図 6-15 発信 RMI のプロパティ



7. [Status] を [Active] に設定し、[Target] には localhost と入力してから、[Service Name] には PO と入力して、[OK] をクリックします。
 8. 着信ファイル接続を作成します。[Add New] アイコン  をクリックします。
 9. 名前を EDIFILE_2_POWERSERVER とし、[Transport] は [File] を選択してから、[Direction] では [ソース] をクリックします。[OK] をクリックします。
- 着信ファイルのプロパティを設定するためのダイアログ ボックスが表示されます。

図 6-16 着信ファイルのプロパティ



以下のプロパティを設定します。

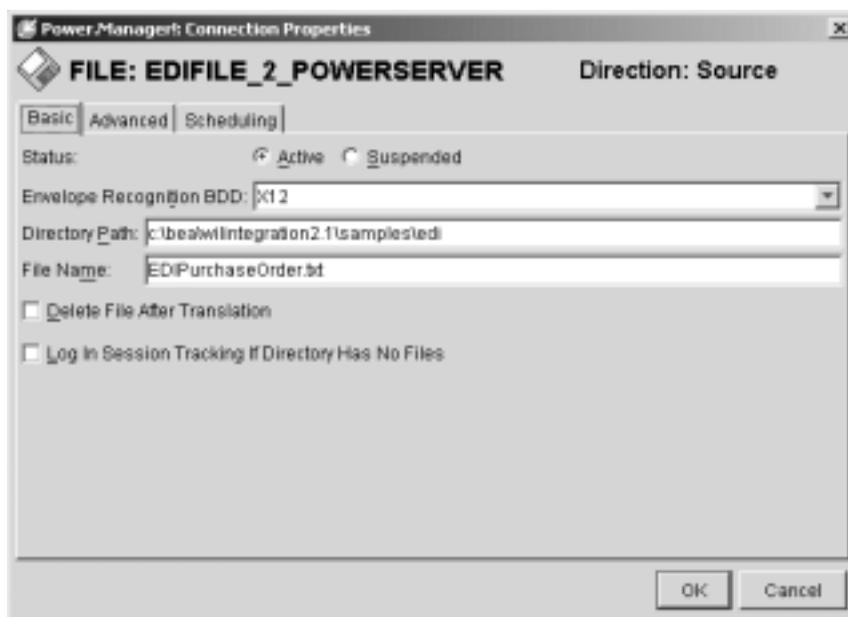
- Status: Active
- Envelope Recognition BDD: X12
- Directory Path: *WLI_HOME*\samples\edi
- Filename: EDIPurchaseOrder.txt
- 他のオプションはすべてオフにします。

設定を済ませたら [OK] をクリックします。

10. 発信ファイル接続を作成します。[Add New] アイコン  をクリックします。

11. 名前を **POWERSERVER_2_EDIFILE** とし、[Transport] は [File] を選択してから、[Direction] では [Target] をクリックします。[OK] をクリックします。
発信ファイルのプロパティを設定するためのダイアログ ボックスが表示されます。

図 6-17 発信ファイルのプロパティ



以下のプロパティを設定します。

- Status: Active
- Directory Path: *WLI_HOME*\samples\edi
- Auto-generate File Name: このオプションはオンにします。
- 他のオプションはすべてオフにします。

設定を済ませたら [OK] をクリックします。

手順 5 : エクスチェンジ プロファイルの設定

1. Power.Manager! で、[Exchange Profiles] を選択します。

エクスチェンジプロファイルをリストしたウィンドウが表示されます。

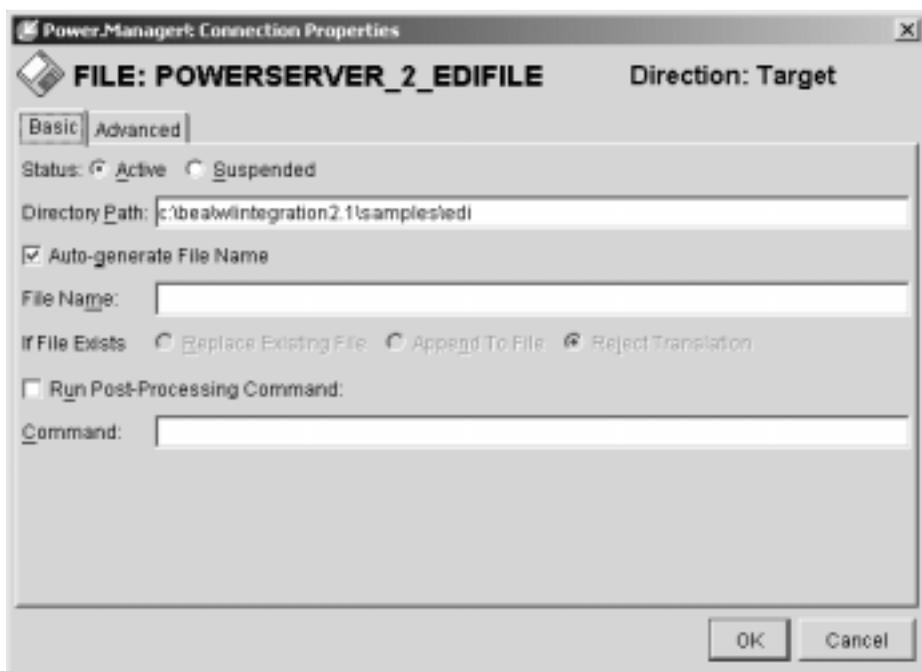
図 6-18 Power.Manager! のエクスチェンジ プロファイル リスト



2. 着信 EDI エクスチェンジ プロファイルを設定します。[Add New] アイコンをクリックします。

[New Exchange Profile] ダイアログ ボックスが表示されます。

図 6-19 新しいエクスチェンジ プロファイルのダイアログ ボックス



3. 以下のプロパティを設定します。
- Map Name: EDI850toPOXML
 - Source Connection: EDIFILE_2_POWERSERVER
 - Target Connection: POWERSERVER_2_WLI_PO
 - Profile Status: Active
 - Partner Company: AES
 - Role: Sender
 - Identifier: ISASNDR99
 - On Compliance Error: Reject Document
 - 他のオプションはすべてオフにします。
- 設定を済ませたら [OK] をクリックします。

4. 発信 EDI エクスチェンジプロファイルを設定します。[Add New] アイコン



をクリックします。

5. 以下のプロパティを設定します。

- Map Name: POAckXMLtoEDI855
- Source Connection: WLI_POACK_2_POWERSERVER
- Target Connection: POWERSERVER_2_EDIFILE
- Profile Status: Active
- Partner Company: AES
- Role: Receiver
- Identifier: ISASNDR99
- On Compliance Error: Reject Document
- 他のオプションはすべてオフにします。

設定を済ませたら [OK] をクリックします。

手順 6 : ワークフローの設定

注意 : Run Sample が『*B2B Integration サンプルの使い方*』の「はじめに」の「サンプルの実行前の作業」に従ってセットアップ済みであれば、この手順を省略して「手順 7 : アプリケーション ビューのデプロイ」に進むことができます。

1. WebLogic Server を起動します。[スタート | プログラム | BEA WebLogic E-Business Platform | WebLogic Integration 7.0 | Samples | Start Server] を選択します。
2. WebLogic Integration Studio を起動します。[スタート | プログラム | BEA WebLogic E-Business Platform | WebLogic Integration 7.0] を選択します。
3. WebLogic Integration Studio にログオンします。サンプルのデフォルトのログイン設定は、次のとおりです。
 - ユーザ ID : joe
 - パスワード : password

4. [Tools | Import Package] を選択します。
5. `WLI_HOME\samples\edi` にあるファイル `EDISampleWorkflow.jar` を選択します。

図 6-20 EDI ワークフローのインポート



6. [Active Workflows After Import] を選択します。指定を求められたら、`Org1` を使用します。

手順 7: アプリケーション ビューのデプロイ

1. Web ブラウザを起動します。
2. `http://localhost:7001/wlai` へ接続します。
3. システムへのログイン ID を使用してログインします。
 - ID: システム
 - パスワード: セキュリティ (Security)

あるいは、**WebLogic Integration** で提供されている **Application Integration** 機能を使用して、新しいユーザを作成します。その後で、新しいユーザとしてログインします。

4. アプリケーション ビュー **EDIAppView** を選択します。
5. [Edit] をクリックします。
6. **EDIAppView** のデプロイに進みます。

これで、EDI サンプルのコンフィグレーションが完了しました。

手順 8 : サンプルの実行

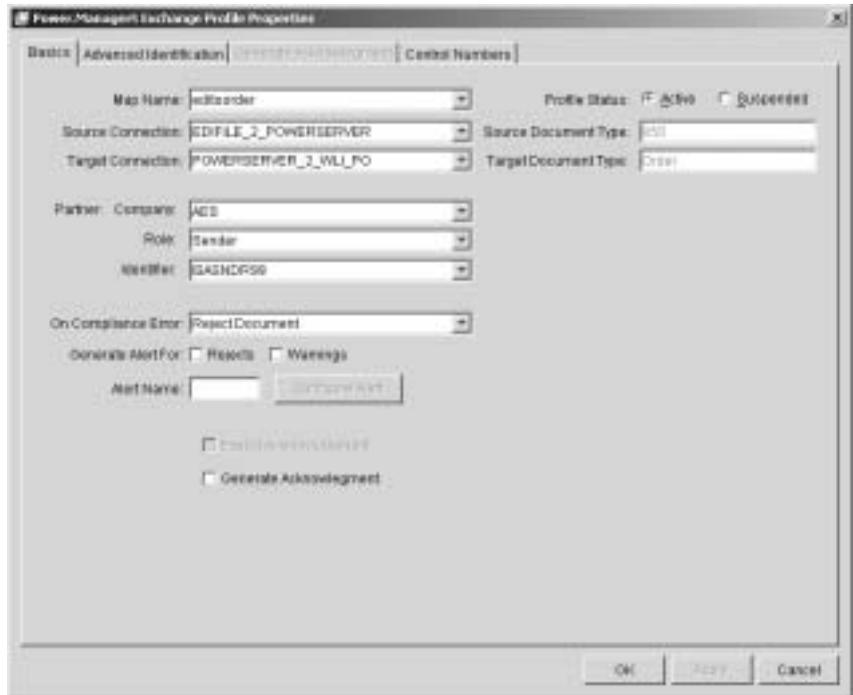
サンプルを起動する方法は2つあります。

- **Power.Commerce!** のインストール先ディレクトリにあるスクリプトを実行することができます。コマンドプロンプトのウィンドウで、次のコマンド行を実行します。

```
power_enterprise_dir\powerit\powerapi\file_request -c
EDIFILE_2_POWERSERVER
```

- 『*B2B Integration* サンプルの使い方』で説明されているように、サンプル起動ページでサンプルを起動することができます。

図 6-21 サンプル起動ページ



実行中のサンプルは、Power.Manager! 内からモニタできます。

- 進行中のセッションを追跡するには、[Tracking | Session | Search] を選択します。次に、[Any Status]、[Any Connection] で検索して、目的のサンプルセッションを探します。
- システム中のドキュメントを追跡するには、[Tracking | Documents | Search] を選択します。次に、すべての検索パラメータを [Any] に設定して、最も広い範囲のドキュメントを検索します。

また、WebLogic Integration Studio の内部からサンプルをモニタすることもできます。

- WebLogic Integration Studio の内部から、EDI Sample Instance を選択し、実行しているインスタンスをダブルクリックしてから、変数を調べます。

サンプルが正しく実行されていることを確認するには、`WLI_HOME\samples\edi`を確認します。このディレクトリに新しいファイルがあれば、それは EDI メッセージがサイクルを完了するときに **Power.Server!** によって作成されたものです。

7 Power.Enterprise! チューニングガイド

この章では、Power.Enterprise! および Power.Server! のチューニングについて説明します。説明するトピックは以下のとおりです。

- Java 仮想マシン (JVM) のチューニング
- データベースのチューニング
- Power.Server! のチューニング
- HTTP および HTTPS サーバのコンフィグレーション

Java 仮想マシン (JVM) のチューニング

この節では、Java 仮想マシンで使用されるチューニング オプションについて説明します。Java 仮想マシンは、すべての Power.Enterprise! ソフトウェアの実行で使用されます。Windows および UNIX システムでの JVM の引数のチューニングについて説明するほか、メモリの使用、および Power.Server! アクセスでのポート番号割り当てについても説明します。

Windows での JVM 引数の指定

`power_enterprise_install\bin` にあるファイル `applaunch.properties` に、Power.Map!、Power.Manager!、Power.Server!、および DBWizard といった各アプリケーション用のセクションが個別に用意されています。

例：

```
[PowerMap]
```

```
install_dir="D:\Program Files\PowerE-3.1GA"  
jvm_args="-Duser.dir=.-Dsun.java2d.noddraw  
-Dvalidationlist.dir=.\data\codes"
```

```
jre_dir=.\jre

classpath=.\lib\regional.jar;.\lib\activation.jar;
.\lib\mail.jar;.\lib\PowerMapper\ecm.jar;
.\lib\PowerMapper\hlvm.jar;
.\lib\hlmetadata.jar;.\lib\xml4j.jar;.\lib\standards.jar;
.\lib\mek.jar;.\jre\lib\ext\jhall.jar;.\lib\fesi.jar;
.\lib\fldapi.jar;.\lib\fldshared.jar;.\lib\hlcommon.jar;
.\lib\fldcommon.jar;.\lib\mekshared.jar;.\lib\fldrmi.jar;
.\lib\fldserver.jar;.\lib\PowerManager\ecm.jar;.\bin\resources;
.\bin\resources\msgd;.\lib\pipeline.zip;.\lib\edi.jar

connectors=.\connectors

database=.\database

standards=.\standards

imports=.\standards\import
```

JVM の引数を変更または追加するには、`jvm_args` 行を変更してください。

注意： 他の値は変更しないでください。アプリケーションの動作に悪影響を与える可能性があります。

UNIX での JVM 引数の指定

UNIX プラットフォームでは、JVM の引数は Java アプリケーションを呼び出すスクリプトの中で直接編集できます。スクリプトは、ディレクトリ

`power_enterprise_install/bin` および

`power_enterprise_install/powerapi` に格納されています。

たとえば、`power_enterprise_install/bin` にある `server_start.sh` の内容は次のとおりです。

```
. /home/fld/PE3.1VERYFINALGA/bin/power_env.sh

java -Xmx512m -Xms256m -Dfld.rmiregistry=true
-Djava.security.policy=$INSTALLDIR/bin \
/resources/fld.policy
-Dvalidationlist.dir=./data/codes -Dfld.trace=false
-user.dir=$INSTALLDIR com.harbinger.fld.jtd.container.HLFLDServer
```

JVM のメモリ消費

JVM で利用可能なメモリ量の最小値および最大値は、標準的な JVM 引数である `-Xms` と `-Xmx` でそれぞれ指定されます。大きなドキュメントの変換でメモリ不足エラーが発生した場合は、メモリ量の最大値を増やしてからサーバを再起動してください。デフォルトでは、サーバは最少で **256 MB**、最大で **512 MB** のメモリを使用するようにコンフィグレーションされています。

サーバのポート番号

サーバでは、クライアントからの通信および変換の要求のために 4 つのポート番号が使用されます。

これらの番号のうち 2 つは **Power.Manager!** の **System Administration** セクションでコンフィグレーションされ、**RMI** レジストリを実行するポートに関係します。必要に応じて、クライアント通信および変換の要求のために 2 つの **RMI** レジストリが使用可能になります。しかし、デフォルトでは、ポート 1099 番にある 1 つのレジストリのみが使用されます。

残りの 2 つの番号は、サーバがこれらの要求をリスンする実際のポートを表します。デフォルトでは、これらのポート番号は **2500** および **2501** です。これらを変更する方法は、スクリプト `powere.rmi.dcpport` および `powere.rmi.apiport` に 2 つの JVM 引数を追加することのみです。これらのスクリプトは、それぞれデーモンの制御および変換 API の要求を行います。次のように引数を追加します。

```
. /home/fld/PE3.1VERYFINALGA/bin/power_env.sh

java -Xmx512m -Xms256m -Dpowere.rmi.dcpport=2597
-Dpowere.rmi.apiport=2598 -Dfld.rmiregistry=true
-Djava.security.policy=$INSTALLDIR/bin/resources/fld.policy
-Dvalidationlist.dir=./data/codes -Dfld.trace=false
-user.dir=$INSTALLDIR com.harbinger.fld.jtd.container.HLFLDServer
```

または

```
[PowerServer]
install_dir="D:\Program Files\PowerE-3.1GA-Server"
jvm_args="-Xmx512m -Xms256m -Dpowere.rmi.dcpport=2597
-Dpowere.rmi.apiport=2598
-Djava.security.policy=.\bin\resources\fld.policy
-Dfld.trace=false -Duser.dir=.-Dpowere.rmi.internalregistry=true
-Dvalidationlist.dir=.\data\codes"
jre_dir=.\jre
classpath=;.\lib\regional.jar;.\lib\PowerServer\j2ee.jar;
.\lib\PowerServer\jcert.jar;.\lib\PowerServer\jnet.jar;
.\lib\PowerServer\jsse.jar;.\lib\PowerServer\sslj.jar;
```

```

.\lib\PowerServer\jsafe.jar;.\lib\hlmetadata.jar;
.\lib\PowerServer\hlvm.jar;.\lib\serverrmi.jar;.\lib\xml4j.jar;
.\lib\fesi.jar;.\lib\mek.jar;.\lib\activation.jar;.\lib\mail.jar;
\jre\lib\ext\jhall.jar;.\lib\fldapi.jar;.\lib\fldshared.jar;
.\lib\fldserver.jar;.\lib\hlcommon.jar;.\lib\fldcommon.jar;
.\lib\standards.jar;.\lib\mekshared.jar;.\lib\pipeline.zip;
.\lib\ejalbert.jar;.\lib\fldrmi.jar;.\bin\resources;
.\bin\resources\msgd;.\lib\PowerServer\CertReqTool.jar;
.\lib\edi.jar
connectors=connectors
database=database
standards=standardsimports=standards\import

```

データベースのチューニング

データベース アクセスの情報は、サーバ上（特に、`power_enterprise_install/bin/resources`）にある `power_server.properties` というファイルで保守されます。クライアントがサーバに接続すると、この情報がクライアントに提供されて、同じデータベースへの接続ができるようになります。ファイルに格納される情報は、使用されるデータベースの種類によって異なり、以下のような項目から成ります。

<code>db_userid</code>	Power.Enterprise! のテーブルを保持しているアカウントのユーザ ID。
<code>db_passwd</code>	Power.Enterprise! のテーブルを保持しているアカウントのパスワード。
<code>db_type</code>	使用されているデータベースの種類。sqlserver、Oracle、または db2 から選択。
<code>no_dbconnections</code>	Power.Enterprise! からデータベースに対して実行されているデータベース接続の数。
<code>db_host</code>	データベースが実行されているホストの名前。このホストにサーバおよびクライアント マシンの両方からアクセスできるようにするには、ホスト名（クライアントに提供される）を省略なしで指定。

db_server	SQL Server および DB2 の場合は、Power.Enterprise! のテーブルを含んでいるデータベースの名前。Oracle の場合は、Power.Enterprise! のテーブルを含んでいる Oracle のインスタンスの名前。
db_dbname	SQL Server および DB2 の場合は、Power.Enterprise! のテーブルを含んでいるデータベースの名前。Oracle の場合は、Power.Enterprise! のテーブルを含んでいる Oracle のインスタンスの名前。
db_port	Power.Enterprise! のテーブルを含んでいるデータベースが使用する TCP/IP ポート。

以下の 3 つの節に、Microsoft SQL Server、Oracle、および IBM DB2 のデータベース用のファイルの例を示します。

SQL Server の場合

```
#FLD Server Properties
#Thu Aug 30 12:03:49 GMT+01:00 2001
db_userid=fld2
db_passwd=fld2
db_type=sqlserver
no_dbconnections=5
db_host=greedo.man.harbinger.co.uk
db_server=fld2
db_dbname=fld2
db_port=1433
```

Oracle の場合

```
#FLD Server Properties
#Tue Sep 18 16:00:18 BST 2001
db_userid=fld8
db_passwd=fld8
db_type=Oracle
no_dbconnections=5
db_host=mace.man.harbinger.co.uk
db_server=fldWE8ISO
db_port=1521
```

DB2 の場合

```
#FLD Server Properties
#Tue Sep 18 14:00:45 GMT+01:00 2001
db_userid=fld
db_passwd=fld
db_type=db2
no_dbconnections=5db_host=manuapartington
db_server=FLD
db_dbname=FLD
db_port=6789
```

Power.Server! のチューニング

この節では、サーバが停止して、サーバへの接続を確立できない状態でも **Power.Manager!** を使用してコンフィグレーションの変更を続行できる方法を説明します。

1. **[Power.Manager! Server Connection]** ダイアログ ボックスが表示されているときに、左側のペイン（以前に接続したサーバの名前がリストされている場所）を右クリックします。ポップアップ メニューが表示されます。
2. **[Connect to database]** を選択します。サーバのデータベース コンフィグレーションが表示されます。
3. データベース接続を確立するために必要な指定をして、サーバを介さずにデータベースに直接変更を加えます。
4. 設定の変更を済ませたら、更新されたデータベースを使用してサーバを起動することができます。この手順の詳細については、『*Power.Manager! User Reference Guide*』を参照してください。

HTTP および HTTPS サーバのコンフィグレーション

`socket.properties` ファイルを使用すると、HTTP および HTTPS サーバを低レベルでコンフィグレーションすることができます。このファイルは、サーバのインストール先ディレクトリの `power_enterprise_install/bin/resources` にあります。設定の各パラメータの説明も、このファイルに記述されています。

