

IBM WebSphere Commerce



# Web サービス・インプリメンテーション・ガイド

バージョン 5.5



IBM WebSphere Commerce



# Web サービス・インプリメンテーション・ガイド

バージョン 5.5

**ご注意!**

本書および本書で紹介する製品をご使用になる前に、65 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 Commerce、バージョン 5.5、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。製品のレベルに合った適切な版を使用していることをご確認ください。

本書は、IBM WebSphere Commerce Studio、Business Developer Edition、バージョン 5.5、IBM WebSphere Commerce Studio、Professional Developer Edition、バージョン 5.5 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。製品のレベルに合った適切な版を使用していることをご確認ください。

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

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

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

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

(URL は、変更になる場合があります)

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

原 典： IBM WebSphere Commerce  
Web Services Implementation Guide  
Version 5.5

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 2003.8

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 2002, 2003. All rights reserved.

© Copyright IBM Japan 2003

# 目次

はじめに	v
知識要件	vi
本ガイドの表記規則	vi
デフォルト・パス	vii
変更の要約	viii
<b>第 1 章 概要</b>	<b>1</b>
用語	1
概説	2
Web サービスを使用する目的	2
Web サービスのアーキテクチャー	4
前提条件	5
参考資料	5
<b>第 2 章 WebSphere Commerce での Web サービス</b>	<b>7</b>
WebSphere Commerce のメッセージング・インフラストラクチャー	7
WebSphereCommerce での Web サービスのインプリメンテーション	7
サービス・プロバイダーとしての WebSphere Commerce	8
要求の処理	8
応答	8
WebSphere Commerce をサービス・プロバイダーとして使用可能にする	9
サービス・リクエスターとしての WebSphere Commerce	10
<b>第 3 章 WebSphere Commerce をサービス・プロバイダーとして使用可能にする</b>	<b>13</b>
Web サービスを使用可能にするためのインフラストラクチャー	13
アーキテクチャー	13
セキュリティ上の考慮事項	15
Web サービスとアクセス制御	15
トランスポート・レベルのセキュリティ	16
Web サービスにアクセスするためのクライアント・コードの生成	16
外部 Web クライアントから呼び出せるサンプル Web サービス	17
サンプル・コードのインポート	17
OrderCreate サービス (Business Edition)	18
OrderCreate サンプル Web サービスを使用可能にする	18
サンプルのテスト	21
OrderCreate Web サービスのセキュリティ	21

ユーザーおよび契約の作成	22
データベース・テーブルの移植	23
新規のメッセージ・マッピング	25
メッセージ・マップパーの構成	27
OrderStatus サービス	27
OrderStatus サンプル Web サービスを使用可能にする	27
OrderStatus サービスのアクセス制御	31
OrderStatus Web サービスでのユーザー役割	32
OrderStatus の WSDL 定義	32
新規のメッセージ・マッピング	33
メッセージ・マップパーの構成	34
OrderCreate および OrderStatus Web サービスを使用不可にする	35

<b>第 4 章 WebSphere Commerce をサービス・リクエスターとして使用可能にする</b>	<b>39</b>
アーキテクチャー	39
サービス・インプリメンテーションの検索	40
サービス・インプリメンテーションの呼び出し	41
WebSphere Commerce が呼び出せる外部システムのサンプル Web サービス	42
OrderFulfillment サービス	42
インターフェース WSDL 定義	42
タスク・コマンド	42
ビジネス・オブジェクト	45
サンプルのテスト	47
OrderFulfillment Web サービスを使用不可にする	48

<b>付録 A. サンプル WSDL 定義ファイル</b>	<b>51</b>
-------------------------------	-----------

<b>付録 B. IBM WebSphere Studio Application Developer の使用法</b>	<b>53</b>
WebSphere Studio Application Developer への WSDL のインポート	53
クライアント・コードとデータ・タイプの生成	54
Web サービスのテスト	55
TCP Monitor による SOAP メッセージのトレース	56
TCP Monitor サーバー・インスタンスの作成	56
TCP Monitor サーバー・インスタンスの構成	57
TCP/IP Monitor による SOAP メッセージのトレース	57
クライアント・コードのカスタマイズ	61

<b>特記事項</b>	<b>65</b>
商標	66



---

## はじめに

「Web サービス・インプリメンテーション・ガイド」は、WebSphere® Commerce バージョン 5.5 Business Edition および Professional Edition でのビジネス・プロセスを Web サービスとして公開するユーザーを対象にしています。また、WebSphere Commerce から外部システムにある Web サービスへアクセスする方法も説明しています。本ガイドは、WebSphere Commerce で Web サービスを使えるようにしてインプリメントする方法を理解する必要がある開発者に役立つように書かれています。Web サービスを構成して WebSphere Commerce が備えている機能を使用するシステム管理者など、その他のユーザーにとっても本ガイドが参考になります。本ガイドは WebSphere Commerce プログラミング・モデルの知識が前提条件となります。さらに、Web サービスおよび関連プロトコルの呼び出しとインプリメントに関する知識も必要です。

本ガイドには、WebSphere Commerce を Web サービスおよび Web サービス・クライアントと連動できるようにする方法を記載しています。本ガイドでは Web サービスの概要とそのアーキテクチャーを説明しています。WebSphere Commerce でのメッセージング・インフラストラクチャー、Web サービスを使用可能にするための WebSphere Commerce ランタイム・サポートを概説し、関連参考資料も記載しています。また、付属のサンプル Web サービスとそれに関連したビジネス・オブジェクトとタスク・コマンドも本ガイドで説明しています。

本ガイドは、以下のセクションに分かれています。

**第 1 章『概要』** Web サービスの概説、本ガイドで使用する用語の定義、前提条件、および関連参考資料。

**第 2 章『WebSphere Commerce での Web サービス』** WebSphere Commerce でのメッセージング・インフラストラクチャーと、WebSphere Commerce での Web サービスのアーキテクチャーの概説。WebSphere Commerce ビジネス・プロセスを Web サービスとして公開する方法と、WebSphere Commerce が外部システムにある Web サービスにアクセスできるようにする方法を説明します。

**第 3 章『WebSphere Commerce を Web サービス・プロバイダーとして使用可能にする』** WebSphere Commerce で定義されているビジネス・プロセスを Web サービスとして使用可能にする方法を、サンプル定義を用いて説明します。WebSphere Commerce に新たに導入されたプログラム・エレメントと、サンプル Web サービスの OrderCreate と OrderStatus のインプリメンテーションの詳細を説明します。

**第 4 章『WebSphere Commerce を Web サービス・リクエスターとして使用可能にする』** 外部システムで定義されているサービスを WebSphere Commerce が呼び出せるようにする方法を、サンプル定義を用いて説明します。WebSphere Commerce に新たに導入されたプログラム・エレメントと、サンプル Web サービスの OrderFulfillment のインプリメンテーションの詳細を説明します。

付録 A 『サンプル WSDL 定義ファイル』 サンプル Web サービスでの WSDL 定義ファイルの一覧です。

付録 B 『IBM® WebSphere Studio Application Developer の使用法』 IBM WebSphere Studio を使用して Webサービス・クライアント・コードを生成する方法を説明します。

## 知識要件

本ガイドに記載している情報を活用するためには、以下の分野の知識が必要です。

- SOAP、WSDL、および UDDI など Web サービスのテクノロジー
- XML
- Java™
- JavaServer Pages テクノロジー
- WebSphere Studio Application Developer (本ガイドでは WebSphere Studio と呼びびます)
- WebSphere Commerce プログラミング・モデル

## 本ガイドの表記規則

本ガイドでは、以下のような規則を使用しています。

太文字	コマンドまたは、フィールド名、ボタン名、メニュー選択などのグラフィカル・ユーザー・インターフェース (GUI) コントロールを示します。
モノスペース (monospaced)	ファイル名やディレクトリー・パスと同様に、示されたとおりに正確に入力する必要があるテキストの例を示します。
イタリック	強調のため、また独自の値に置き換えることができる変数を表すために使用されます。
	WebSphere Commerce for AIX® に固有の情報であることを示します。
	WebSphere Commerce for IBM @server iSeries™ (以前の AS/400®) に固有の情報であることを示します。
	WebSphere Commerce for xSeries Linux に固有の情報、WebSphere Commerce for @server zSeries® and S/390® Linux に固有の情報、WebSphere Commerce for @server iSeries Linux に固有の情報、および WebSphere Commerce for @server pSeries Linux に固有の情報であることを示します。
	WebSphere Commerce for Solaris オペレーティング環境ソフトウェアに固有の情報であることを示します。
	WebSphere Commerce for Windows® に固有の情報であることを示します。
	DB2® Universal Database に固有の情報であることを示します。
	Oracle® データベースに固有の情報であることを示します。
	WebSphere Commerce Business Edition に固有の情報であることを示します。
	WebSphere Commerce Studio バージョン 5.5 (Professional Edition または Business Developer Edition) の場合です。

+  
+  
+

## デフォルト・パス

本ガイドでは、デフォルトのインストール・パスとして以下を使用しています。

*WC\_installdir* は、WebSphere Commerce のインストール・パスを示します。この変数が示された場合は、WebSphere Commerce が実際にインストールされているインストール・パスに置き換えてください。たとえば、次のように置き換えます。

▶ Windows	C:¥IBM¥WebSphere¥CommerceServer55
▶ AIX	/usr/WebSphere/CommerceServer55
▶ Linux	/opt/WebSphere/CommerceServer55
▶ Solaris	/opt/WebSphere/CommerceServer55
▶ 400	/QIBM/ProdData/CommerceServer55

*WAS\_installdir* は、WebSphere Application Server のインストール・パスを示します。この変数が示された場合は、WebSphere Application Server が実際にインストールされているインストール・パスに置き換えてください。たとえば、次のように置き換えます。

▶ Windows	C:¥IBM¥WebSphere¥AppServer
▶ AIX	/usr/WebSphere/AppServer
▶ Linux	/opt/WebSphere/AppServer
▶ Solaris	/opt/WebSphere/AppServer
▶ 400	/QIBM/ProdData/WebAS5/Base

*WCStudio\_installdir* は、WebSphere Commerce Studio のインストール・パスを示します。この変数が示された場合は、WebSphere Commerce Studio が実際にインストールされているインストール・パスに置き換えてください。たとえば、次のように置き換えます。

▶ Studio	C:¥IBM¥WebSphere¥CommerceStudio5
----------	----------------------------------

*WCStudio\_workspacedir* は、WebSphere Commerce Studio ワークスペースのディレクトリです。たとえば、次のように置き換えます。

▶ DB2	C:¥WebSphere¥workspace_db2
▶ Oracle	C:¥WebSphere¥workspace_oracle

*WAS\_userdir* は、WebSphere Application Server で使用されるすべてのデータのディレクトリーを示し、ユーザーが変更可能または構成する必要があるものです。この変数が示された場合は置き換えてください。

400

`/QIBM/UserData/WebAS5/Base/WAS_instance_name`

ここで *WAS\_instance\_name* は、ご使用の WebSphere Commerce インスタンスが関連付けられる WebSphere Application Server の名前を表す変数です。

---

## 変更の要約

- + 本書の最新バージョンは、以下の WebSphere Commerce の Web サイトの
- + Technical Library ページで、PDF ファイルとして入手できます。
  
- + <http://www.ibm.com/software/commerce/library/>
  
- + プロダクトへの最新の変更点について学ぶには、現行のプロダクト README ファイルをご覧ください。これも WebSphere Commerce の Web サイトから入手できます。
  
- + 本書では、改訂の文字として以下のような規則を使用しています。
- + • "+" (プラス) 文字は、本書の現行の改訂でなされた更新箇所を示しています。
  
- + このガイドの現行の改訂における更新は、Linux に特有の点です。

---

## 第 1 章 概要

この章では Web サービスの概要を説明します。本ガイドで使用する用語の定義、前提条件、関連する他の情報源も示します。

---

### 用語

本ガイドでは以下の用語を使用します。

#### サービス・プロバイダー

サービス・プロバイダーとは Web サービスの所有者のことです。サービス・プロバイダーは Web サービスを作成し、サービスをサービス・レジストリーに対してまたは直接サービス・クライアントに対して発行し、サービスにアクセスするためのプラットフォームを提供します。

#### サービス・リクエスター

サービス・リクエスターとは、Web サービスとの対話を開始するアプリケーションのことです。たとえば、標準的なインターネット・ブラウザは、サービス・リクエスターの一例です。別の例として、処理要求に応じて Web サービスを用いて情報を収集するアプリケーションが挙げられます。サービス・リクエスターは発行済み情報を用いてサービスにバインドしてサービスを呼び出します。

#### SOAP (Simple Object Access Protocol)

SOAP は XML で記述された単純な要求と応答メッセージを使用するモデルを、電子通信の基本プロトコルとして定義したものです。これは、基礎部分のプラットフォームやプロトコルから独立しているトランスポート・メカニズムです。これによって発行、検索、バインド、および呼び出しの各操作が容易になります。

#### UDDI (Universal Description, Discovery, and Integration)

UDDI レジストリーは Web サービスの説明情報を保管します。UDDI は、利用可能な Web サービスのサービス・レジストリーを定義している仕様であり、世界的規模の電子イエローページのコレクションとしての機能を果たします。企業は UDDI を用いて、購入可能な商品やサービスの説明をレジストリーに発行して、自身がサービス・プロバイダーであることを告知することができます。

#### Web サービス

Web サービスは、標準 XML メッセージングを使用してネットワーク・アクセスが可能な操作のコレクションを記述した、発行済みインターフェースに基づいているソフトウェア・コンポーネントです。

#### WSDL (Web サービス記述言語)

WSDL は、Web サービスとアプリケーション・プログラミング・インターフェース (API) の特定のインターフェースの記述方法を規定したものです。

#### XML マッピング・テンプレート

XML マッピング・テンプレート (またはマッピング・テンプレート) は、

インバウンド XML メッセージを処理して、それを WebSphere Commerce のコマンド・パラメーターに変換するために使用します。 WebSphere Commerce のマッピング・テンプレート・ファイルで、着信 XML メッセージのエレメントをターゲットのコマンド・パラメーターにマップする方法を定義します。

マッピング・テンプレートについての詳細は、 WebSphere Commerce オンライン・ヘルプを参照してください。

### XML メッセージ・マッパー

WebSphere Commerce の XML メッセージ・マッパーは、着信 XML メッセージを、ターゲット・コマンドが要求している Java オブジェクトに変換するためのものです。メッセージ・マッパーは *instance\_name.xml* 構成ファイルに登録され、メッセージ・マッパー ID によって一意的に識別されます。 XML メッセージ・マッパーはマッピング・テンプレート・ファイルで定義されているマッピングに基づいて、 XML メッセージを対応する Java オブジェクトのコマンド・パラメーターに変換します。

**注:** *instance\_name* は、 WebSphere Commerce インスタンスの名前です。

メッセージ・マッパーについての詳細は、 WebSphere Commerce オンライン・ヘルプの、 XML メッセージ・マッパーに関する情報を参照してください。

---

## 概説

Web サービスは新しいタイプの Web アプリケーションです。 Web サービスは自己完結型で自己説明型のモジュラー・アプリケーションで、 Web 上で発行、配置、呼び出しを行うことができます。 Web サービスは、単純な要求から複雑なビジネス・プロセスまで呼び出すことができる機能を実行します。 Web サービスをデプロイして登録すると、他のアプリケーションはデプロイされたそのサービスを発見して呼び出すことができます。 Web サービスは、 Simple Object Access Protocol (SOAP)、 Web サービス記述言語 (WSDL)、 Universal Description Discovery and Integration (UDDI) レジストリーといった標準を基礎にしています。

WebSphere Commerce バージョン 5.5 にはコードとドキュメンテーションが含まれており、これを使って WebSphere Commerce ビジネス機能を Web サービスとして使用できるようになります。 WebSphere Commerce ビジネス機能を外部システムがアクセスできる Web サービスとして使用可能にすることにより、 WebSphere Commerce をサービス・プロバイダーにすることができます。外部システムによってホスティングされる Web サービスを呼び出せるようにして、 WebSphere Commerce をサービス・リクエスターにすることもできます。

## Web サービスを使用する目的

Web サービスは、アプリケーションを従来よりも迅速、簡単、かつ低コストで統合することができます。この統合はプロトコル・スタックの高位レベルで行い、多くのサービス・セマンティクスのメッセージ入力、および少ないネットワーク・プロトコル・セマンティクスのメッセージ入力を基本としており、ビジネス機能を自由に統合することができます。このような特性は、複数の企業間でも単一企業内でも、ビジネス機能を Web 上で接続するには最適です。 Web サービスには統一プロ

グラミング・モデルがあるため、企業内外でのアプリケーション統合を、共通のインフラストラクチャーを活用しながら共通のアプローチで行うことができます。Web サービスの統合およびアプリケーションは、既存の言語とプラットフォームを用いて既存のアプリケーションを取り入れながら、段階的に行うことができます。

Web サービスは、標準 XML メッセージを用い、ネットワークを介してアクセスできる操作のコレクションを記述した、インターフェースです。Web サービスは、そのサービス説明を標準の正式な XML 表記で記述したものです。サービス説明には、メッセージ形式 (操作の詳細) やトランスポート・プロトコルなど、サービスとの対話に必要なすべての詳細情報が記述されます。このインターフェースではサービスのインプリメンテーションの詳細が表面に現れないため、このインターフェースはインプリメントされるハードウェアやソフトウェアのプラットフォームに関係なく、また記述に使用されるプログラム言語にも関係なく使用することができます。

WebSphere Commerce は、Web サービスのプロバイダーとしても、また他の Web サービスのリクエスター (クライアント) としても Web サービスをサポートできるように設計されています。たとえば、外部の Web サービス・クライアントが WebSphere Commerce に接続して、そこで提供している Web サービスを利用することができます (図 1 を参照)。一方 WebSphere Commerce も他の Web サービス・プロバイダーに接続して、情報を要求することができます (図 2 を参照)。

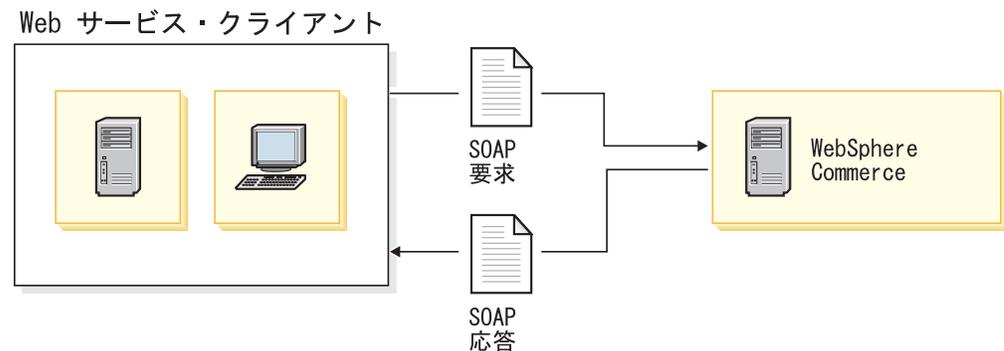


図 1. サービス・プロバイダーとしての WebSphere Commerce

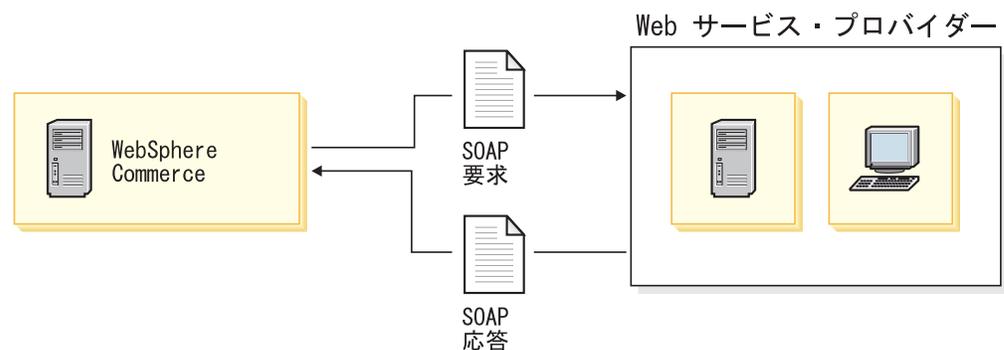


図 2. サービス・リクエスターとしての WebSphere Commerce

## Web サービスのアーキテクチャー

Web サービスのアーキテクチャーには、サービス・プロバイダー、サービス・リクエスター、サービス・ブローカーという 3 つの役割と、発行、検索、バインドという 3 つの基本操作が含まれます。ネットワーク・コンポーネントはこれらの役割のいずれか、またはすべてを実行することができます。サービス・プロバイダーはサービス・ブローカーに対して Web サービスを発行します。サービス・リクエスターは、要求された Web サービスをサービス・ブローカーを使用して検索し、その Web サービスにバインドします。

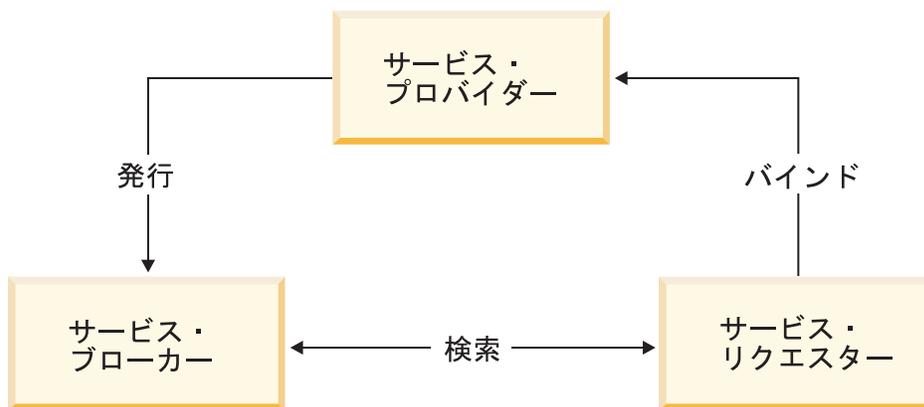


図3. Web サービスのアーキテクチャー

一般的なシナリオでは、サービス・プロバイダーにはネットワーク・アクセスが可能なサービス・モジュールが置かれ、これが Web サービスのインプリメンテーションになります。サービス・プロバイダーで定義された Web サービスのサービス説明は、サービス・リクエスターまたはサービス・レジストリーに対して発行されます。サービス・リクエスターは検索操作を行います。この場合サービス説明をローカルに検索するか、サービス・ブローカーが扱うサービス・レジストリーから検索します。サービス・リクエスターはサービス説明を用いてサービス・プロバイダーにバインドし、Web サービスのインプリメンテーションを呼び出したり対話を行ったりします。

このソリューションでは、WebSphere Commerce で定義されているビジネス・プロセスを Web サービスとして使用できるようにするためのランタイム・サポートを示します。既存の WebSphere Commerce コマンドを Web サービスとして使用する方法のサンプルについては、13 ページの『第 3 章 WebSphere Commerce をサービス・プロバイダーとして使用可能にする』を参照してください。外部システムで定義されている Web サービスを WebSphere Commerce が呼び出す方法のサンプルについては、39 ページの『第 4 章 WebSphere Commerce をサービス・リクエスターとして使用可能にする』を参照してください。

---

## 前提条件

本ガイドに記載している情報を使用する前に、IBM WebSphere Commerce Business Edition バージョン 5.5 がインストールされていることを確認してください。

WebSphere Commerce バージョン 5.5 をインストールするときに付属のサンプル Web サービスもインストールするには、「**カスタム (Custom)**」インストール・オプションを選択する必要があります。どのコンポーネントを選択してインストールするか尋ねられます。「**WebSphere Commerce サンプル・ファイル (WebSphere Commerce example files)**」を選択してください。詳しくは、「*WebSphere Commerce インストール・ガイド*」を参照してください。

あるいは WebSphere Commerce Studio を用いてサンプル Web サービスを使用可能にする場合は、WebSphere Commerce Studio バージョン 5.5 のインストールが完了していることを確認してください。詳しくは、「*WebSphere Commerce Studio インストール・ガイド*」を参照してください。

サンプル Web サービスを変更する場合は、開発環境として必ず WebSphere Commerce Studio を使用してください。

---

## 参考資料

Web サービスおよび関連テクノロジーについての詳細は、以下の Web サイトを参照してください。

- W3C に準拠した SOAP 送信のインプリメンテーションである Apache SOAP については、<http://ws.apache.org/soap/> を参照してください。
- SOAP 仕様の詳細は、<http://www.w3.org/TR/SOAP/> を参照してください。
- UDDI については、<http://www.uddi.org/> を参照してください。
- WSDL については、<http://www.uddi.org/specification.html> を参照してください。
- IBM Alphaworks の Web サービスの詳細は、<http://www.alphaworks.ibm.com/webservices> を参照してください。
- 開発者の立場での Web サービスについて理解を深めるには、<http://www.ibm.com/developerworks/webservices> を参照してください。
- IBM WebSphere Commerce Business Edition ソフトウェアについての詳細は、[http://www.ibm.com/software/webservers/commerce/wc\\_be/](http://www.ibm.com/software/webservers/commerce/wc_be/) を参照してください。
- IBM WebSphere Commerce Professional Edition ソフトウェアについての詳細は、[http://www.ibm.com/software/webservers/commerce/wc\\_pe/](http://www.ibm.com/software/webservers/commerce/wc_pe/) を参照してください。
- 優れた WebSphere Web サービス・ソリューションを開発するビジネス・パートナーのサポートについては、<http://www.ibm.com/websphere/wow> を参照してください。
- IBM WebSphere Studio については、[www.ibm.com/software/webservers/commerce/commercestudio/](http://www.ibm.com/software/webservers/commerce/commercestudio/) を参照してください。

**注:** 上記の Web アドレスは、予告なしに変更されることがあります。IBM 以外の Web サイトの情報の信頼性や正確性については、IBM は責任を負いません。



---

## 第 2 章 WebSphere Commerce での Web サービス

この章では、WebSphere Commerce での Web サービスのアーキテクチャーの理解に役立つように、WebSphere Commerce のメッセージング・インフラストラクチャーを簡単に説明します。ここでは WebSphere Commerce ビジネス・プロセスを Web サービスとして公開する方法と、WebSphere Commerce が外部システムにある Web サービスにアクセスする方法も説明します。

---

### WebSphere Commerce のメッセージング・インフラストラクチャー

WebSphere Commerce ランタイム・アーキテクチャーには、プロトコル・リスナー、アダプター、コントローラー、コマンド、およびビューが含まれます。WebSphere Commerce のメッセージング・インフラストラクチャーでは、これらのコンポーネントが使用されます。

プロトコル・リスナーは、トランスポートからインバウンド要求を受信し、使用するプロトコルに基づいてその要求を適切なアダプターにディスパッチする WebSphere Commerce ランタイム・コンポーネントです。

WebSphere Commerce バージョン 5.5 では、RequestServlet がプロトコル・リスナーとして機能して SOAP バージョン 1.1 のメッセージを受け取ります。RequestServlet は XML 要求を受け取ると、その要求をアダプター・マネージャーに渡します。次にアダプター・マネージャーはアダプター・タイプを参照して、その要求を処理できるアダプターを決定します。特定のアダプターが決定したら、その要求はそのアダプターに渡されます。

プログラム・アダプターは、XML メッセージを HTTP プロトコルで受け取ったときに呼び出されます。したがって外部システムは、XML 要求を HTTP プロトコルで渡せば、WebSphere Commerce と通信を行うことができます。プログラム・アダプターは、調達システムやサプライヤー・システムなどの外部システムに、HTTP プロトコルで WebSphere Commerce と通信を行える共通の方法を提供するものです。このような方法で、WebSphere Commerce は外部システムとシームレスに統合されます。

要求のフローについての詳細は、「*WebSphere Commerce プログラミング・ガイドとチュートリアル*」を参照してください。

---

### WebSphereCommerce での Web サービスのインプリメンテーション

WebSphere Commerce プログラミング・モデルは、URL コマンド、コントローラー・コマンド、タスク・コマンド、ビュー・コマンド、プログラム・アダプター、Web コントローラーなどで構成されます。WebSphere Commerce プログラミング・モデルについての詳細は、「*WebSphere Commerce プログラミング・ガイドとチュートリアル*」を参照してください。

以下のセクションでは、サービス・プロバイダーおよびサービス・リクエスターの役割としての WebSphere Commerce について説明します。

## サービス・プロバイダーとしての WebSphere Commerce

WebSphere Commerce のビジネス・プロセスを外部システムがアクセスできる Web サービスとして使用可能にすると、WebSphere Commerce はサービス・プロバイダーになります。WebSphere Commerce バージョン 5.5 では、SOAP のメッセージと応答をサポートできるように、メッセージング・システムが拡張されています。また、提供されているサンプルは、既存の WebSphere Commerce ビジネス・ロジックを Web サービスとして公開する方法を示しています。

クライアントは HTTP プロトコルまたはセキュア HTTP プロトコルを用いて、ネットワーク上の URL コマンドを呼び出すことができます。WebSphere Commerce は、HTTP および HTTPS プロトコルまたはメッセージング・ミドルウェアを介して、XML メッセージを受け取ることもできます。この機能がさらに拡張されて、WebSphere Commerce は HTTP で SOAP メッセージを受け取ることができるようになっていました。Web サービスを使用可能にするには、この機能を使用します。RequestServlet は、WebSphere Commerce システムへのすべての URL 要求のエントリー・ポイントです。RequestServlet は XML メッセージを受け取ると、XML としての内容タイプを識別してプログラム・アダプターを呼び出します。

### 要求の処理

プログラム・アダプターは、WebSphere Commerce コマンドを呼び出すリモート・プログラムをサポートします。プログラム・アダプターは要求を受け取ると、メッセージ・マッパーを用いてその要求をコマンドの適切な入力に変換します。この変換後、プログラム・アダプターはそのコマンドを実行します。

WebSphere Commerce バージョン 5.5 では、WebSphere Commerce ランタイム・アーキテクチャーに新規のメッセージ・マッパーとテンプレート・マッピング・ファイルを追加して、SOAP XML メッセージの構文解析と処理が容易に行えるようにしています。

この新規のテンプレート・マッピング・ファイルには、着信 SOAP XML メッセージとコマンド・パラメーター間のマッピングが含まれます。新規のメッセージ・マッパーはこのテンプレート・マッピング・ファイルを調べて SOAP XML メッセージを構文解析し、コマンドの適切な入力に変換した後にその入力をプログラム・アダプターに戻します。次にプログラム・アダプターが Web コントローラーを呼び出し、今度は Web コントローラーが、メッセージ・マッパーの指定したコマンドを実行します。

### 応答

WebSphere Commerce では、コントローラー・コマンドの実行が正常に終了すると、実行対象のビューの名前を戻します。Web コントローラーは対応するビューのビュー・コマンドを呼び出し、メッセージを組み立てて送り返します。応答ビューを形成する方法には、いくつかあります。たとえば、別の URL へのリダイレクト、JSP テンプレートへの転送、応答への HTML 文書の書き込みなどがあります。

各種クライアント・デバイスに対してメッセージを生成できるように、WebSphere Commerce はさまざまなデバイス形式をサポートしています。Web コントローラーは着信メッセージのデバイス形式に対応するビュー名を選択します。WebSphere Commerce バージョン 5.5 の新規デバイス形式として、SOAP XML メッセージ用

に SOAPHTTP が導入されています。同じビュー名に対して、新規デバイス形式の JSP テンプレートのセットを新たに定義することができます。SOAP クライアントに戻される SOAP 応答は、これらの JSP テンプレートによって構成されます。この場合モデル・ビュー・コントローラー・パターンに従って、ビジネス・ロジックからビューが分離されます。

SOAP 応答を処理するための新しいビュー・コマンドがインプリメントされています。このビュー・コマンド・インプリメンテーションのインターフェースの名前は `com.ibm.commerce.me.soap.commands.SOAPViewCommand` で、このインプリメンテーションのクラス名は `com.ibm.commerce.me.soap.commands.HttpSOAPViewCommandImpl` です。したがって、新規 JSP を VIEWREG テーブルに登録して SOAP 応答メッセージを構成するときは、必ずこのインターフェース名とクラス名を使用する必要があります。

新規 SOAP ビュー・コマンド・インプリメンテーションは、HTTP リダイレクトと HTTP 転送の両方の要求を処理できます。通常 HTTP リダイレクトは URL リダイレクトで処理されます。SOAP クライアントは URL リダイレクトを認識できないため、この SOAP ビュー・コマンドには、HTTP 転送としても HTTP リダイレクトがインプリメントされています。

## WebSphere Commerce をサービス・プロバイダーとして使用可能にする

WebSphere Commerce ビジネス・プロセスを Web サービスとして使用可能にするステップの概要は、以下のとおりです。

1. Web サービスとして公開するビジネス・ロジックを特定します。
2. このビジネス・ロジックをインプリメントする URL コマンドを特定します。使用できる URL コマンドがない場合は、新規に URL コマンドを作成します。既存のコマンドのコレクションを呼び出すコマンドを新規に作成することができます。
3. この URL コマンドに必要な必須パラメーターとオプション・パラメーターを特定します。
4. Web サービスの WSDL 定義を定義します。
5. マッピング・テンプレート・ファイルを用いて、着信 SOAP 要求をコマンド・パラメーターにマップします。TemplateTag には、WSDL 定義にあるすべてのエレメントについてのマッピングを必ず含めてください。これは、クライアントから要求が送られてきたときに、SOAP 要求内のすべてのエレメントのデータを取り込めるようにするためです。マッピング・テンプレート・ファイルには、着信 SOAP XML メッセージ内の各エレメントがコマンド・パラメーターにどのようにマップされているかを、XPath マッピングを用いて定義します。TCP Monitor のようなユーティリティーを用いて SOAP 要求をインターセプトし、インターセプトしたメッセージを基にマップを定義します。

**注:** TCP Monitor は WebSphere Studio Application Developer 5.0 で使用できません。詳しくは WebSphere Studio のヘルプを参照してください。TCP Monitor のスタンドアロン・バージョンは Apache AXIS 配布で出荷されています。詳しくは、<http://ws.apache.org/axis/> を参照してください。

6. JSP テンプレートを作成して応答を構成します。

7. サービスをデプロイします。これには、関連したファイルおよびリソースのデプロイメントも含まれます。新規コマンドの場合は、新規コマンドを WebSphere Commerce コマンド・レジストリーにデプロイして登録する必要があります。

WebSphere Commerce に含めるビジネス・ロジックは、コントローラー・コマンドを用いてインプリメントします。コントローラー・コマンドは、Web コントローラーと直接対話するコマンドです。URL コマンドは URL を使用して呼び出すことができるコントローラー・コマンドです。したがって、Web ブラウザーに URL を入力することで、URL コマンドを呼び出すことができます。Web サービスをインプリメントする前に、Web サービスとして公開するビジネス・ロジックをインプリメントする URL コマンドを特定してください。たとえば OrderStatus コマンドで、WebSphere Commerce 内の既存のオーダーの状況を更新することができます。

URL コマンドについての詳細は、WebSphere Commerce オンライン・ヘルプの『Reference』のセクションを参照してください。

## サービス・リクエスターとしての WebSphere Commerce

WebSphere Commerce を用いて外部システムにある Web サービスを開始する場合、WebSphere Commerce はサービス・リクエスターになります。WebSphere Commerce バージョン 5.5 には、Web サービス呼び出しを行って WebSphere Commerce ビジネス・ロジックを他のシステムと統合する方法を示すサンプルが含まれています。WebSphere Commerce は、SOAP ベースの要求と応答の送受信を行う Web サービス・プロキシーを経由して Web サービス要求を送信します。ビジネス・ロジックは、特定の Web サービス・バインディング・インターフェースで機能します。付属のサンプルは、それらのインターフェースでの Web サービスの呼び出しをサポートしています。

インプリメンテーションはサービス・レジストリーに発行するか、ローカル・ファイル・システムに保管します。WebSphere Commerce から Web サービスを呼び出す場合は、以下のアクティビティが必要でです。

1. 発行済み Web サービスのインターフェース定義を特定します。これは、WebSphere Commerce がサポートしている既存のビジネス・シナリオに適合するものにします。
2. そのサービスのインプリメンテーションの WSDL 定義を、ローカル WebSphere Commerce マシンにダウンロードします。
3. Web サービス定義ファイルを *instance\_name.xml* 構成ファイルに登録します (*instance\_name* は WebSphere Commerce Server インスタンスの名前です)。
4. WebSphere Commerce が使用できるクライアント・プロキシーを作成します。この作成には IBM WebSphere Studio などのツールを使用することができます。Web サービスのインターフェース定義用のクライアント・プロキシーを生成すると、入力メッセージと出力メッセージに必要なデータ・タイプも生成されます。詳しくは、本ガイドの 16 ページの『Web サービスにアクセスするためのクライアント・コードの生成』のセクションを参照してください。

たとえば Web Services Toolkit など、他のツールを用いてクライアント・プロキシーを生成することもできます。 <http://www.alphaworks.ibm.com/webservices> を参照してください。

5. ロジックをインプリメントするタスク・コマンドを作成して、以下のようにします。
  - a. アクセス・ポイントを検索します。
  - b. Web サービス用のプロキシーを呼び出します。
  - c. Web サービスから戻される結果を処理します。
6. WebSphere Commerce コマンド・レジストリーを更新して、タスク・コマンドを登録します。



---

## 第 3 章 WebSphere Commerce をサービス・プロバイダーとして使用可能にする

この章では、WebSphere Commerce で定義されているビジネス・プロセスを Web サービスとして使用可能にする方法を説明します。このシナリオでは、WebSphere Commerce は外部システムに対するサービス・プロバイダーとして機能します。サンプルを用いて、ビジネス・プロセスの OrderCreate と OrderStatus を、Web サービスとして使用可能にする方法を示します。

---

### Web サービスを使用可能にするためのインフラストラクチャー

着信 Web サービス要求をサポートするために必要な WebSphere Commerce ランタイム・エレメントは、以下のとおりです。

- SOAP メッセージ・マッパー
- Web サービス・マッピング・テンプレート
- SOAP 応答ビュー・コマンド

外部ユーザーはクライアント・プログラムを用いて、WebSphere Commerce のサービスを呼び出すことができます。外部ユーザーは、SOAP メッセージの送信と受信を行えるシステムであれば、どのシステムからでも接続することができます。

WebSphere Commerce バージョン 5.5 は、HTTP トランスポート・プロトコルによる SOAP メッセージをサポートしています。

WebSphere Commerce のサービスを WSDL 定義として発行する必要があります。Web サービス・クライアントは WSDL 定義をダウンロードし、クライアントを生成し、SOAP over HTTP を用いて WebSphere Commerce Web サービスを呼び出すことができます。

### アーキテクチャー

このアーキテクチャーのメッセージ・フローは同期しています。つまり外部システムは、WebSphere Commerce システムからの応答を待ってから、次の処理に進みます。Web サービスとして発行されている WebSphere Commerce ビジネス・プロセスを外部クライアントが呼び出せるようにするために、インバウンド・メッセージング・システムを使用します。

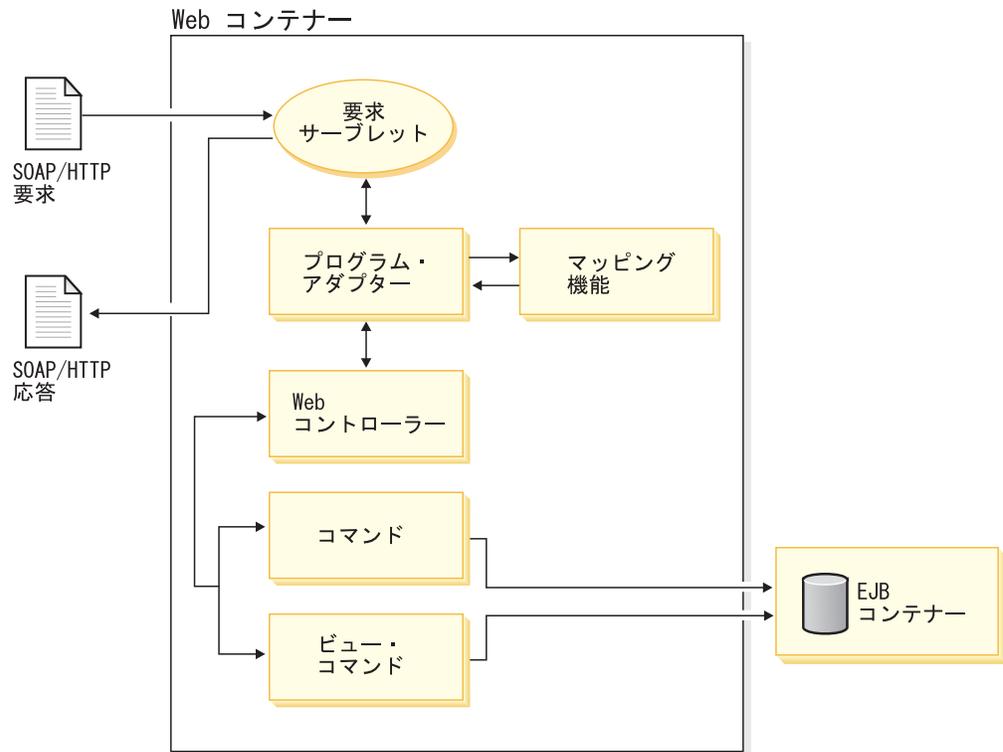


図4. WebSphere Commerce をサービス・プロバイダーとして使用可能にするためのアーキテクチャー

外部システムが Web サービスを使用して WebSphere Commerce ビジネス・プロセスを呼び出すときの対話は、以下のように行われます。

1. 外部システムが整形形式 SOAP 要求を WebSphere Commerce に送信します。
2. その要求を RequestServlet が受け取り、その処理をプログラム・アダプターに依頼します。
3. プログラム・アダプターは登録済み SOAP メッセージ・マッパーを呼び出します。
4. SOAP メッセージ・マッパーは SOAP メッセージを構文解析して、名前と値の対をプログラム・アダプターに戻します。このときテンプレート・ファイルが使われ、このテンプレート・ファイルには Web サービス要求の中のパラメーターとコマンド・パラメーター間のマッピング情報が含まれています。
5. プログラム・アダプターは Web コントローラーを呼び出し、コマンド名および名前と値の対を Web コントローラーに渡します。
6. Web コントローラーはそのコマンドを実行します。この例では、BatchOrderRequest および OrderStatusCmd コマンドが呼び出されます。
7. コマンドの実行後、ビュー名が Web コントローラーに戻されます。
8. Web コントローラーは登録済みビュー・コマンドを実行して SOAP 応答を形成し、それをプログラム・アダプターに戻します。
9. HTTP プログラム・アダプターはメッセージを RequestServlet に戻します。
10. RequestServlet は、待機している外部クライアントに応答を送信します。

サービス・プロバイダーが提供する Web サービスをサポートするために、以下のサンプルを追加しています。

- `webservice_template.xml` ファイルは、XPath をコマンド・プロパティにマップするためのものです。
- 以下のサンプル JSP テンプレートは、SOAP 応答とエラー・メッセージを構成するためのものです。
  - `SOAP_OrderStatusResponse.jsp`
  - `SOAP_PurchaseOrderResponse.jsp`
  - `SOAP_GenericApplicationError.jsp`
  - `SOAP_GenericSystemError.jsp`

---

## セキュリティ上の考慮事項

以下のセクションでは、Web サービスを使用可能にする場合に必要なセキュリティ上の考慮事項を取り上げます。

### Web サービスとアクセス制御

アクセス制御は、WebSphere Commerce システム内の保護リソースに対するアクセスを、許可した個人と組織だけに制限する方法です。データベース管理を容易にしてセキュリティを確保するために、WebSphere Commerce はリソースへのアクセスを特定のユーザーに制限するアクセス制御のメカニズムを備えています。アクセス制御は、以下のセキュリティ・ガイドラインで行います。

1. システムによって管理されているリソースへのシステム・アクセスを行うユーザーを許可または拒否する。
2. リソースごとにユーザーが実行できるアクションを指定する。

WebSphere Commerce のアクセス制御モデルは、アクセス制御ポリシーの実施が基本になります。アクセス制御とポリシーについての詳細は、「*WebSphere Commerce* セキュリティ・ガイド」を参照してください。

WebSphere Commerce がサービス・プロバイダーとして機能しているときは、外部の Web サービス・クライアントは、コントローラー・コマンドで定義されているビジネス・プロセスを呼び出すことができます。Web サービスは、Web サービス・クライアントによって呼び出されるコマンドに適用されるアクセス制御ポリシーを必然的に継承します。したがって、WebSphere Commerce で使用する WSDL 定義ごとに認証情報を定義する必要があります。Web サービス・クライアントは、ユーザーの認証に必要な認証情報を渡すことの他に、ユーザーがそのコマンドを実行するのに必要な権限を持っていることを確認しなければなりません。

認証情報のための WSDL 定義の XML スキーマの形式は、以下のようなものになります。

```
<complexType
name="com.ibm.commerce.webservice.datatype.Credentials">
<all>
<element name="password" nillable="true" type="string"/>
<element name="loginId" nillable="true" type="string"/>
</all>
</complexType>
```

認証情報は、マッピング・テンプレート・ファイルにあるテンプレート定義の中の以下のフィールドにマップする必要があります。

- ユーザー・ログインのフィールド名は 'logonId' で FieldInfo は 'CONTROL'。
- パスワードのフィールド名は 'logonPassword' で FieldInfo は 'CONTROL'。

## トランスポート・レベルのセキュリティー

トランスポート・レベルのセキュリティーを有効にするには、 SOAP メッセージを HTTPS で送信します。サービス・プロバイダーは HTTPS を用いて SOAP メッセージを送信することを強くお勧めします。こうしない場合、メッセージがインターセプトされて、渡された認証情報が見られる可能性があります。

**注:** HTTPS を使用可能にするには、 SOAP クライアントはプロトコル名に HTTPS を指定した URL を使用する必要があります。クライアントからの呼び出しを受け入れる前に、プロパティーを設定して SSL (Secure Sockets Layer) を使用可能にしてください。 Web サービス・クライアントに IBM Developer Kit, Java Technology Edition Version 1.2 以降を使用する場合は、 HTTPS を使用可能にするためのコードは以下のようなものになります。

```
System.setProperty("java.protocol.handler.pkgs", "com.ibm.net.ssl.internal.www.protocol");
```

```
java.security.Security.addProvider  
(new com.ibm.jsse.JSSEProvider());
```

このコードは、WebSphere Commerce Studio を使用して Web サービス・クライアントを開発する場合に使用します。

Sun Microsystems の J2SE SDK (Java 2 Standard Edition Software Development Kit) を使用する場合は、 HTTPS で Web サービスを呼び出すためのクライアント・コードは、以下のようなものになります。

```
System.setProperty("java.protocol.handler.pkgs", "com.sun.net.ssl.internal.www.protocol");
```

```
java.security.Security.addProvider  
(new com.sun.net.ssl.internal.ssl.Provider());
```

WebSphere Commerce で定義されているインバウンド・サービスを呼び出そうとして、クライアント・サイドで 'handshake failed'、'unknown CA'、または 'connection refused by server exception' を受け取った場合は、サーバーから提示された証明書をクライアントが認識できないことを意味します。これが起きてもそのサーバーを信頼できる場合は、そのサーバー証明書をトラステッド証明書のリストに追加することができます。

---

## Web サービスにアクセスするためのクライアント・コードの生成

このセクションでは、クライアント・コードの生成方法を説明します。

サンプル Web サービスは WSDL 定義ファイルを用いて発行され、以下のディレクトリにあります。

-  `WC_install\dir\%samples%\webservices\xml\messaging`
-  `WCStudio_install\dir\%Commerce%\samples%\webservices\xml\messaging`

-    `WC_installdir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging`

WSDL 定義を使って、Web サービス・リクエスターとしてのクライアント・アプリケーションを開発することができます。WebSphere Commerce によってインプリメントされている Web サービスにアクセスするのに必要なクライアント・コードとデータ・タイプを、適切なツールを用いて Web サービスの WSDL 定義から生成することができます。

IBM WebSphere Studio Application Developer (本ガイドでは WebSphere Studio とも呼びます) も、WSDL 定義からクライアントおよびプロキシー・コードを生成するのに使用できるツールの 1 つです。詳しくは、53 ページの『付録 B. IBM WebSphere Studio Application Developer の使用法』を参照してください。

クライアント・プロキシーの生成に使用できる他のツールとして、Web Services Toolkit があります。 <http://www.alphaworks.ibm.com/webservices> を参照してください。

---

## 外部 Web クライアントから呼び出せるサンプル Web サービス

このセクションでは、以下のサンプル Web サービスを使用可能にする方法を説明します。

1.  OrderCreate サービス
2. OrderStatus サービス

サンプル Web サービスは、WebSphere Commerce マシン上でも、WebSphere Commerce Studio がインストールされている開発マシン上でも、どちらでも使用可能にすることができます。WebSphere Commerce Studio マシン上でサンプル Web サービスを使用可能にする場合は、このセクションの説明に従い  のマークが付いているディレクトリー・パスを適用してください。

## サンプル・コードのインポート

サンプル Web サービスを編集または変更する場合は、サンプル・ソース・コードを WebSphere Commerce Studio にインポートする必要があります。このコードをインポートするには、以下のようにします。

1. 「スタート」 > 「プログラム」 > 「IBM WebSphere Commerce Studio」 > 「WebSphere Commerce 開発環境 (WebSphere Commerce development environment)」を選択して、WebSphere Commerce 開発環境を開きます。
2. J2EE パースペクティブに切り替えて、「J2EE ナビゲーター (J2EE Navigator)」ビューを選択します。
3. **WebSphereCommerceServerExtensionsLogic** プロジェクトを拡張表示します。**src** ディレクトリーを右マウス・ボタンでクリックして「インポート」を選択します。インポート・ウィザードが開きます。
4. 「ファイル・システム」を選択して、「次へ」をクリックします。

5. 「ブラウズ」をクリックして次のディレクトリーを見つけます。  
WCStudio\_install\dir¥Commerce¥samples¥webservices¥com
6. 「全選択」をクリックし、次に「終了」をクリックします。
7. **WebSphereCommerceServerExtensionsLogic** プロジェクトを右マウス・ボタンでクリックして「プロパティ」を選択します。
8. 「Java ビルド・パス (Java Build Path)」をクリックして、次に「ライブラリー (Libraries)」タブを選択します。
9. 「変数の追加 (Add variable)」をクリックします。変数として SOAPJAR と XERCES を選択して「OK」をクリックします。
10. 「OK」をクリックします。
11. **WebSphereCommerceServerExtensionsLogic** プロジェクトを右マウス・ボタンでクリックして「プロジェクトの再ビルド (Rebuild project)」を選択します。

これで WebSphere Studio へのサンプル・ソース・コードのインポートが完了しました。 WebSphereCommerceServer サーバーを WebSphere Test Environment 内で開始する方法の詳細は、「WebSphere Commerce Studio インストール・ガイド」を参照してください。

## OrderCreate サービス (Business Edition)

OrderCreate Web サービスは、 WebSphere Commerce Business Edition および WebSphere Commerce Business Developer Edition に適用されます。

このセクションでは、 WebSphere Commerce 内のビジネス・ロジックを、調達アプリケーションなど外部アプリケーションからのオーダー受信をサポートする、 Web サービスとして使用可能にする方法を、例を用いて説明します。調達システムは、 WebSphere Commerce BatchOrderRequest コマンドを用いて、 WebSphere Commerce 内にオーダーを作成することができます。このサンプル OrderCreate Web サービスのインプリメンテーションで、バイヤー組織からオーダーを受け取る Web サービスの作成方法を説明します。 OrderCreate サンプル Web サービスを使用するには、 WebSphere Commerce システム内に発行済みストアが必要です。

通常、セラーは WebSphere Commerce にオーダーを作成する前に、バイヤー組織を知っておかなければなりません。結果的には、バイヤーが OrderCreate Web サービスを呼び出す前に、そのバイヤー組織は WebSphere Commerce に登録されていなければなりません。

### OrderCreate サンプル Web サービスを使用可能にする

サンプル OrderCreate Web サービスを使用可能にするには、以下のようになります。

1. Web サービスとして公開するビジネス・ロジックを特定します。ここでは、バイヤー組織が調達システムに似たアプリケーションを用いて購入オーダーを作成できるようにします。
2. このビジネス・ロジックをインプリメントするコマンドを特定します。ここでは BatchOrderRequest コマンドです。
3. BatchOrderRequest コマンドに必要な必須パラメーターとオプション・パラメーターを特定します。 BatchOrderRequest コマンドについての説明は、

WebSphere Commerce オンライン・ヘルプを参照してください。パラメーターの詳細なリストについては、以下のディレクトリーにある付属の `webservice_template.xml` ファイルを参照してください。

-  `WC_installdir\%samples%\webservices\%xml%\messaging`
  -  `WCStudio_installdir\%Commerce%\samples%\webservices\%xml%\messaging`
-    `WC_installdir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging`

4. OrderCreate Web サービスの WSDL 定義を作成します。サンプル `OrderCreate.wsdl` 定義ファイルは、以下のディレクトリーにあります。

-  `WC_installdir\%samples%\webservices\%xml%\messaging`
  -  `WCStudio_installdir\%Commerce%\samples%\webservices\%xml%\messaging`
-    `WC_installdir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging`

この WSDL 定義を変更して、提供するサービスの定義要件に合わせることができます。

OrderCreate Web サービス用に発行された WSDL 定義には、この Web サービスを呼び出すのに必要なアクセス・ポイント、バインディング・テンプレート、データベース・スキーマが含まれています。

5. マッピング・テンプレート・ファイルを用いて、着信 SOAP 要求をコマンド・パラメーターにマップします。サンプル・マッピング・テンプレートは、以下のディレクトリーにあります。

-  `WC_installdir\%samples%\webservices\%xml%\messaging\webservice_template.xml`
  -  `WCStudio_installdir\%Commerce%\samples%\webservices\%xml%\messaging\webservice_template.xml`
-    `WC_installdir/samples/webservices/xml/messaging/webservice_template.xml`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging/webservice_template.xml`

このファイルをカスタマイズすることができます。サンプル WSDL 定義を変更した場合は、WSDL 定義での変更に合わせて、このサンプル・マッピング・テンプレートを変更する必要があるかもしれません。

6. OrderCreate サンプル Web サービスを含むこのテンプレートを使用可能にするには、以下のようにします。

a. webservice\_template.xml ファイルを、以下のディレクトリーにコピーします。

-  `WC_installdir%\xml%\messaging`
-  `WCStudio_installdir%\Commerce%\xml%\messaging`
-    `WC_installdir/xml/messaging`
-  `/QIBM/UserData/CommerceServer55/  
instances/instance_name/xml/messaging`

b. webservice\_template.xml ファイルがすでに存在していてそのファイルに情報が何も含まれていない場合は、上書きします。そのファイルに情報が含まれている場合は、OrderCreate サービスに対応する TemplateDocument 情報を既存ファイルにコピーします。

7. SOAPINTEGRATION 用のメッセージ・マッパーを使用可能にします。これを行うには、以下のディレクトリーにある `instance_name.xml` 構成ファイルを開きます。

-  `WC_installdir%\instances%instance_name%.xml`
-  `WCStudio_installdir%\Commerce%\instances%instance_name%.xml`
-    `WC_installdir/instances/instance_name/xml`
-  `/QIBM/UserData/CommerceServer55/instances/instance_name/xml`

「WCS.SOAPINTEGRATION」という名前のメッセージ・マッパーを見つけます。「enable」属性の値が「false」になっていたら「true」に設定します。

8. JSP テンプレートを作成して応答を構成します。ここでは、以下のサンプル JSP テンプレートをコピーします。

- SOAP\_PurchaseOrderResponse.jsp - OrderCreate 要求に対する SOAP 応答メッセージを構成します。
- SOAP\_GenericApplicationError.jsp - アプリケーション・エラーが発生した場合に SOAP エラー応答を構成します。
- SOAP\_GenericSystemError.jsp - システム・エラーが発生した場合に SOAP エラー応答を構成します。

以下のディレクトリーから、

-  `WC_installdir%\samples%\webservices%\wcstores`
-  `WCStudio_installdir%\Commerce%\samples%\webservices%\wcstores`
-    `WC_installdir/samples/webservices/wcstores`
-  `WC_installdir/samples/webservices/wcstores`

以下のディレクトリーへコピーします。

-  `WAS_installdir%\installedApps% cell_name%\  
WC_instance_name.ear%\Stores.war`

ここで、*cell\_name* は WebSphere Application Server がインストールされているマシンの名前です。

-  *WCStudio\_workspacedir*¥Stores¥Web Content
-    *WAS\_installdir/installedApps/cell\_name*¥  
*WC\_instance\_name.ear/Stores.war*
-  *WAS\_userdir/installedApps/cell\_name*¥  
*WC\_instance\_name.ear/Stores.war*

9. バイヤー組織を作成し、バイヤー組織とサプライヤー組織の間で契約を行います。詳しくは、本ガイドの 22 ページの『ユーザーおよび契約の作成』を参照してください。
10. 本ガイドの 23 ページの『データベース・テーブルの移植』の説明に従って、データベース・テーブルを移植します。
11. WebSphere Commerce を使用している場合は、Commerce Server インスタンスを再始動して変更を有効にします。WebSphere Commerce Studio マシンを使用している場合は、WebSphere Test Environment 内で WebSphereCommerceServer サーバーを再始動します。

## サンプルのテスト

WSDL 定義から Web サービス・クライアントを生成することができます。クライアントは PurchaseOrderRequest メッセージを送信することにより、この Web サービスを呼び出すことができます。オーダーが正常に発行された場合の応答メッセージには、状況コード (200)、"OK" という状況メッセージ、状況テキスト、WebSphere Commerce で作成された OrderID が含まれます。エラーの場合は、該当するエラー・メッセージを受け取ることになります。

## OrderCreate Web サービスのセキュリティ

このサンプルで Web サービスとして公開するビジネス・ロジックでは、要求側のバイヤー組織が認証されなければなりません。WebSphere Commerce のアクセス制御モデルでは、コマンドおよびそれに関連する Web サービスの認証は、要求者の WebSphere Commerce ユーザー ID とパスワードを指定することにより行います。したがって、ユーザーは WebSphere Commerce ユーザーとして登録されていなければなりません。

サンプル OrderCreate Web サービスを使用するユーザーを認証するには、そのユーザーが属するバイヤー組織が WebSphere Commerce に登録されている必要があります。渡された認証情報は、OrderCreate サービスを処理する前に検査する必要があります。

OrderCreate Web サービスの認証情報には、WebSphere Commerce システムでのサプライヤーとバイヤーの、組織コードおよび組織コード・タイプが含まれます。クライアントは、ORGCODE テーブルに登録されているものと同じ組織コードおよび組織コード・タイプの値を送信しなければなりません。

WSDL 定義内の認証情報のスキーマを以下に示します。ここで *identity* は組織コード、*domain* は組織コード・タイプ、*sharedSecret* はバイヤー組織とサプライヤー組織で共有しているパスワードです。

```
<complexType
name="com.ibm.commerce.webservice.datatype.ordercreate.
Credential">
<all>
<element name="identity" nillable="true" type="string"/>
<element name="domain" nillable="true" type="string"/>
<element name="sharedSecret" nillable="true" type="string"/>
</all>
```

注: sharedSecret が適用されるのは、送信側の認証情報に対してのみです。

### アクセス制御

WebSphere Commerce は OrderCreate メッセージを受け取ると、BatchOrderRequest コマンドを呼び出して着信要求を処理します。OrderCreate Web サービスを使用するためには、WebSphere Commerce を使用するサプライヤーと Web サービス・クライアントになるバイヤーの間に、契約が存在しなければなりません。

### ユーザーおよび契約の作成

バイヤー組織が存在しない場合は、以下を完了してから OrderCreate Web サービスを呼び出します。以下のステップで、新しいバイヤー組織の作成、それへの調達バイヤーの役割の割り当て、セラー組織とバイヤー組織の間の契約の作成を行います。

1. 管理コンソールを用いてバイヤー企業組織を作成します。詳しくは、WebSphere Commerce オンライン・ヘルプを参照してください。
2. バイヤー組織に以下の役割を割り当てます。
  - a. 調達バイヤー
  - b. 調達バイヤー管理者
  - c. 調達マネージャー
3. 管理コンソールで、このバイヤー組織のユーザーを作成します。このユーザーのユーザー ID はバイヤー組織のグループ・ユーザー ID、パスワードはバイヤー組織の共有秘密でなければなりません。グループ・ユーザー ID と共有秘密は、15 ページの『セキュリティ上の考慮事項』で説明したバイヤー組織の認証情報の一部になります。
4. ステップ 3 で作成したユーザーに、以下の役割を割り当てます。
  - a. 調達バイヤー
  - b. 調達バイヤー管理者
5. WebSphere Commerce アクセラレーターを用いて、このバイヤー組織のビジネス・アカウントを作成します。ビジネス・アカウントの作成方法は、WebSphere Commerce オンライン・ヘルプを参照してください。ビジネス・アカウントを作成するときは、以下のことに注意してください。
  - 「顧客」ページの「ストアのデフォルトの契約条件で顧客が購入するのを許可する (Allow customers to purchase under the terms and conditions of store's default contract)」チェック・ボックスを選択しないでください。
  - 「購入オーダー」ページの「購入オーダー番号はオーダー時に指定可能」チェック・ボックスを選択しないでください。

- 「貸付限度額」ページの「貸付限度支払いメソッド (credit line payment method)」を使用可能にして、「貸付限度アカウント番号」を指定してください。
- 作成したビジネス・アカウントの契約を作成します。ビジネス・アカウントの契約の作成方法は、WebSphere Commerce オンライン・ヘルプを参照してください。契約を作成するときは、以下のことに注意してください。
    - 「一般」タブをクリックします。契約名と説明を入力し、その契約をアクティブにしておく期間を選択します。
    - 「顧客」タブをクリックします。作成済みのバイヤー組織を選択します。
    - 「支払い」タブをクリックします。「貸付限度額」の使用にチェックマークが付いていることを確認してください。
  - 「送信」をクリックして、その契約をアクティブにします。

以上のステップで WebSphere Commerce にバイヤー組織のアカウントを作成し、サプライヤー (WebSphere Commerce) 組織とバイヤー組織の間の契約を作成しました。バイヤー組織に必要なアクセス許可が割り当てられます。

### データベース・テーブルの移植

BatchOrderRequest コマンドを実行するには、データベース・スクリプトを実行して特定のテーブルを移植しなければなりません。

データベースを更新するには、以下のようにします。

- 以下のディレクトリーに移動します。
  -  `WC_installdir\samples\webservices\bin`
  -  `WCStudio_installdir\Commerce\samples\webservices\bin`
  -     `WC_installdir/samples/webservices/bin`
- 以下の表に従って、使用している構成に合ったスクリプト・ファイルのセットを見つけます。

オペレーティング・システム	データベース・タイプ	スクリプト・ファイル名
 		Webservices_DBUpdate.db2.bat Webservices_UndoDBupdate.db2.bat
		Webservices_DBUpdate.oracle.bat Webservices_UndoDBupdate.oracle.bat
 		Webservices_DBUpdate.db2.sh Webservices_UndoDBupdate.db2.sh
		Webservices_DBUpdate.oracle.sh Webservices_UndoDBupdate.oracle.sh

+  
+  
+  
+  
+

		Webservices_DBUpdate.db2.sh  Webservices_UndoDBUpdate.db2.sh
		

3. 該当するスクリプト・ファイルを以下のディレクトリーへコピーします。

-  `WC_installdir¥samples¥webservices¥bin`
  -  `WCStudio_installdir¥Commerce¥samples¥webservices¥bin`
-    `WC_installdir/samples/webservices/bin`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/bin`

4.  以下のディレクトリーから、 `wcs.webservices.enable.sql` スクリプトと `wcs.webservices.disable.sql` スクリプトをコピーします。

-  `WC_installdir¥samples¥webservices¥schema¥db2`
  -  `WCStudio_installdir¥Commerce¥samples¥webservices¥schema¥db2`
-    `WC_installdir/samples/webservices/schema/db2`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/schema/db2`

以下のディレクトリーへコピーします。

-  `WC_installdir¥schema¥db2`
  -  `WCStudio_installdir¥Commerce¥schema¥db2`
-    `WC_installdir/schema/db2`
-  `/QIBM/UserData/CommerceServer55/WebSphere_Commerce_instance_name/schema`

 以下のディレクトリーから、 `wcs.webservices.enable.sql` スクリプトと `wcs.webservices.disable.sql` スクリプトをコピーします。

-  `WC_installdir¥samples¥webservices¥schema¥oracle`
  -  `WCStudio_installdir¥Commerce¥samples¥webservices¥schema¥oracle`
-   `WC_installdir/samples/webservices/schema/oracle`

以下のディレクトリーへコピーします。

-  `WC_installdir¥schema¥oracle`
  -  `WCStudio_installdir¥Commerce¥schema¥oracle`
-   `WC_installdir/schema/oracle`

- ステップ 4 でコピーした `wcs.webservices.enable.sql` および `wcs.webservices.disable.sql` SQL スクリプトで使用されている以下のデフォルト値を変更して、`OrderCreate Web` サービスを呼び出すバイヤー組織の詳細に合わせます。
  - `WebServicesUserId` - バイヤー組織の組織コード。このコードと組織コード・タイプでバイヤー組織が一意的に識別されます。
  - `Webservices@ibm.com` - バイヤー組織の組織コードのタイプ。
  - `aurora` - サプライヤー組織の組織コード。これは通常、組織の D-U-N-S 番号、または組織の他の標準固有 ID になります。(D-U-N-S 番号は 9 桁の番号で、ビジネスの識別およびトラッキングの業界のグローバル・スタンダードとして使用されています。)
  - `DUNS` - サプライヤー組織の組織コードのタイプ。組織コードに DUNS 番号を使用する場合は、“DUNS” を使用してください。
- SQL スクリプトを更新後、該当するコマンド・スクリプト・ファイルを実行します。

 DB2 データベースを使用している場合

-   DB2 のコマンド・ウィンドウから、`Webservices_DBUpdate.db2.bat` を実行します。
-    コマンド・ウィンドウから、`Webservices_DBUpdate.db2.sh` を実行します。
-  QSH セッションを開始します。コマンド `/QIBM/ProdData/CommerceServer55/samples/webservices/bin/Webservices_DBUpdate.db2.sh` を実行します。

 Oracle データベースを使用している場合

-   コマンド・ウィンドウから、`Webservices_DBUpdate.oracle.bat` を実行します。
-   コマンド・ウィンドウから、`Webservices_DBUpdate.oracle.sh` を実行します。

`WC_installdir¥logs¥Webservices_DBUpdate.log` ログ・ファイルを検査して、実行が正常に完了していることを確認してください。

**注:** `OrderCreate Web` サービスのみを使用可能にする場合は、`OrderFulfillment Web` サービス用の `ExtOrderProcessServiceCmdImpl` および `WSDLFinderCmdImpl` タスク・コマンドが登録されている SQL スクリプト内の行をコメント化する必要があります。

## 新規のメッセージ・マッピング

`OrderCreate Web` サービスを使用可能にするために、サンプル・マッピングの新規セットが `webservice_template.xml` ファイルに用意されています。メッセージ・マップのテンプレート文書を以下に示します。

```
<!-- mapping for inbound order create message -->
<TemplateDocument>
```

```

<DocumentType version='1.0'>ns1:orderCreate</DocumentType>
<StartElement>ns1:orderCreate</StartElement>
<TemplateTagName>BatchOrderCreate10Map</TemplateTagName>
<CommandMapping>
  <Command CommandName='BatchOrderRequest'>
    <Constant Field='protocolName'>SOAP</Constant>
    <Constant Field='protocolVersion'>1.1</Constant>
  </Command>
</CommandMapping>
</TemplateDocument>

```

このサービスに対応するメッセージ・マッピングの名前は BatchOrderCreate10Map で、以下のディレクトリーにある webservice\_template.xml テンプレート・ファイルに入っています。

-  `WC_installdir\%samples%\webservices\xml\messaging`
-  `WCStudio_installdir\%Commerce%\samples%\webservices\xml\messaging`
-    `WC_installdir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging`

用意されているメッセージ・マップは、以下のディレクトリーにある WSDL 定義に対応しています。

-  `WC_installdir\%samples%\webservices\xml\messaging\%OrderCreate.wsdl`
-  `WCStudio_installdir\%Commerce%\samples%\webservices\xml\messaging\%OrderCreate.wsdl`
-    `WC_installdir/samples/webservices/xml/messaging/OrderCreate.wsdl`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging/OrderCreate.wsdl`

このメッセージ・マップは、与えられている WSDL 定義から生成された Web サービス・クライアントが送信する SOAP 要求メッセージを扱います。

メッセージ・マップを定義するには、SOAP 要求に含まれる XPATH 情報が分かっている必要があります。XPATH 情報は WSDL 定義から得ることができます。しかし、SOAP メッセージを調べる方が便利でしょう。次のセクションでは、SOAP メッセージにアクセスしてマッピング・テンプレートを定義する方法を説明します。

OrderCreate メッセージ (BatchOrderCreate10Map) に対応するメッセージ・マップに類似したメッセージ・マップを、対応 SOAP 要求から定義するには、以下のようになります。

1. WSDL 定義を用いてクライアント・プロキシを生成します。ステップの説明は、53 ページの『付録 B. IBM WebSphere Studio Application Developer の使用法』を参照してください。
2. トランスポート・レベルのインターセプト・ユーティリティーを用いて、SOAP 要求と SOAP 応答文書がどのようなものかを調べます。たとえば WebSphere Studio Application Developer 5.0 で使用できる TCP Monitor など、メッセージ

をログに記録できる HTTP トンネル・ユーティリティーを使用します。アクセス・ポイントを設定して、クライアント・プロキシが、インターセプトするものを指すようにします。SOAP 要求を見てみると、要求に対応したマッピングの作成や、応答の構成に使用する JSP ファイルの作成に役立ちます。

3. テンプレート定義ファイルの `TemplateDocument` エlementを定義します。これには、SOAP XML メッセージを受け取ったときに実行するコマンドの詳細を含めます。詳しくは、WebSphere Commerce オンライン・ヘルプの『テンプレート定義ファイルの `TemplateDocument` Element』のセクションを参照してください。
4. ステップ 2 で調べた SOAP 要求を用いて、タグ定義のリストを含む `TemplateTag` を定義します。 `TemplateTag` には、WSDL 定義にあるすべての Element についてのマッピングを必ず含めてください。これは、クライアントから要求が送られてきたときに、SOAP 要求内のすべての Element のデータを取り込めるようにするためです。各タグ定義で、着信 SOAP XML メッセージにある Element または属性を、コマンド・パラメーターにマップします。WebSphere Commerce オンライン・ヘルプの『テンプレート定義ファイルの `TemplateTag` Element』のセクションを参照してください。

### メッセージ・マッパーの構成

SOAP メッセージ・マッパーは、 `webservice_template.xml` Web サービス・テンプレート定義ファイルを用いて、着信 SOAP 要求をコマンドのプロパティにマップします。したがって、そのテンプレートを変更することにより、SOAP メッセージ・マッパーを構成することができます。

別の WSDL 定義がある場合やサービス定義を変更する場合は、テンプレートのマップを変更する必要があります。マップは、コマンド・パラメーターにマップされた XML メッセージの XPATH 定義で構成されます。この場合は、新しいサービス定義のデータ・タイプスキーマに合わせて XPATH 構造を変更する必要があります。

## OrderStatus サービス

このセクションでは、WebSphere Commerce に置かれているオーダーの状況を更新するビジネス・ロジックを、Web サービスとして使用可能にする方法について例を用いて説明します。OrderStatus Web サービスを使用すると、Web サービス呼び出しで WebSphere Commerce OrderStatus コマンドを呼び出すことができます。OrderStatus コントローラー・コマンドを使用することにより、外部システムは WebSphere Commerce 内の既存のオーダーの状況に関連した情報を更新することができます。OrderStatus サンプル Web サービスを使用するには、WebSphere Commerce システム内の発行済みストアを必要とする既存のオーダーがなければなりません。

提供サンプルを用いて、WebSphere Commerce OrderStatus コマンドを Web サービス要求で使用できるようにする方法を説明します。

### OrderStatus サンプル Web サービスを使用可能にする

サンプル OrderStatus Web サービスをインプリメントするためのステップの概要を以下に示します。

1. Web サービスとして公開するビジネス・ロジックを特定します。ここでは、オーダーを履行するアプリケーションが WebSphere Commerce 内のオーダーの状況を更新できるようにします。
2. このビジネス・ロジックをインプリメントするコマンドを特定します。ここではサンプルの OrderStatusService コマンドです。これは既存の OrderStatus コマンドを拡張したものです。
3. このコマンドに必要な必須パラメーターとオプション・パラメーターを特定します。これについての情報は、WebSphere Commerce オンライン・ヘルプを参照してください。
4. OrderStatus Web サービスの WSDL 定義を作成します。サンプル OrderStatus.wsdl 定義ファイルは、以下のディレクトリーにあります。

-  `WC_installldir¥samples¥webservices¥xml¥messaging`  
 -  `WCStudio_installldir¥Commerce¥samples¥webservices¥xml¥messaging`
-    `WC_installldir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging`

5. すでに OrderCreate サービスを使用可能にしている場合は、このステップは実行しないでください。そうでない場合は、マッピング・テンプレート・ファイルを用いて、着信 SOAP 要求をコマンド・パラメーターにマップします。サンプル・マッピング・テンプレートは、以下のディレクトリーにあります。

-  `WC_installldir¥samples¥webservices¥xml¥messaging¥webservice_template.xml`  
 -  `WCStudio_installldir¥Commerce¥samples¥webservices¥xml¥messaging¥webservice_template.xml`
-    `WC_installldir/samples/webservices/xml/messaging/webservice_template.xml`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging/webservice_template.xml`

- a. webservice\_template.xml ファイルを、以下のディレクトリーにコピーします。

-  `WC_installldir¥xml¥messaging`
-  `WCStudio_installldir¥Commerce¥xml¥messaging`
-    `WC_installldir/xml/messaging`
-  `/QIBM/UserData/CommerceServer55/instances/instance_name/xml/messaging`

- b. webservice\_template.xml ファイルがすでに存在していてそのファイルに情報が何も含まれていない場合は、上書きします。そのファイルに情報が入っている場合は、OrderStatus サービスに対応する TemplateDocument を既存ファイルにコピーします。

6. すでに OrderCreate サービスを使用可能にしている場合は、このステップは実行しないでください。そうでない場合は SOAPINTEGRATION 用のメッセージ・マップパーを使用可能にします。これを行うには、以下のディレクトリーにある `instance_name.xml` 構成ファイルを開きます。

-  `WC_installdir¥instances¥instance_name.xml`
  -  `WCStudio_installdir¥Commerce¥instances¥instance_name.xml`
-    `WC_installdir/instances/instance_name.xml`
-  `/QIBM/UserData/CommerceServer55/instances/instance_name.xml`

「WCS.SOAPINTEGRATION」という名前のメッセージ・マップパーを見つけます。「enable」属性の値が「false」になっていたら「true」に設定します。

7. JSP テンプレートを作成して応答を構成します。応答を構成するには、サンプル JSP ファイル `SOAP_OrderStatusResponse.jsp` が使用できます。
- a. `SOAP_OrderStatusResponse.jsp` を、以下のディレクトリーからコピーします。

-  `WC_installdir¥samples¥webservices¥wcstores`
  -  `WCStudio_installdir¥Commerce¥samples¥webservices¥wcstores`
-    `WC_installdir/samples/webservices/wcstores`
-  `WC_installdir/samples/webservices/wcstores`

以下のディレクトリーへコピーします。

-  `WAS_installdir¥installedApps¥ cell_name¥WC_instance_name.ear¥Stores.war`
  -  `WCStudio_installdir¥Commerce¥IBM WebSphere Test Environment¥hosts¥default_host¥default_app¥web`
-    `WAS_installdir/installedApps/cell_name/WC_instance_name.ear/Stores.war`
-  `WAS_userdir/installedApps/cell_name/WC_instance_name.ear/Stores.war`

WebSphere Commerce 内のオーダーの状況が正常に更新されたときに、この JSP テンプレートで SOAP 応答メッセージを構成します。

- b. すでに OrderCreate サービスを使用可能にしている場合は、このステップを実行する必要はありません。そうでない場合は、`SOAP_GenericApplicationError.jsp` および `SOAP_GenericSystemError.jsp` ファイルを、以下のディレクトリーからコピーします。

-  `WC_installdir¥samples¥webservices¥wcstores`
  -  `WCStudio_installdir¥Commerce¥samples¥webservices¥wcstores`
-    `WC_installdir/samples/webservices/wcstores`

-  /QIBM/ProdData/CommerceServer55/samples  
/webservices/wcstores

以下のディレクトリーへコピーします。

-  *WAS\_installdir*¥installedApps¥ *cell\_name*¥  
*WC\_instance\_name*.ear¥Stores.war
-  *WCStudio\_workspacedir*¥Stores¥Web Content
-    *WAS\_installdir*/installedApps/*cell\_name*/  
*WC\_instance\_name*.ear/Stores.war
-  *WAS\_userdir*/installedApps/*cell\_name*/  
*WC\_instance\_name*.ear/Stores.war

アプリケーション・エラーまたはシステム・エラーが発生したときに、これらの JSP テンプレートで SOAP エラー・メッセージを構成します。

- すでに OrderCreate サービスを使用可能にしている場合は、このステップを実行する必要はありません。そうでない場合は、以下のようにします。
  - 以下の表に従って、使用している構成に合ったスクリプト・ファイルのセットを見つけます。

オペレーティング・システム	データベース・タイプ	スクリプト・ファイル名
 		Webservices_DBUpdate.db2.bat Webservices_UndoDBupdate.db2.bat
		Webservices_DBUpdate.oracle.bat Webservices_UndoDBupdate.oracle.bat
 		Webservices_DBUpdate.db2.sh Webservices_UndoDBupdate.db2.sh
		Webservices_DBUpdate.oracle.sh Webservices_UndoDBupdate.oracle.sh
 		Webservices_DBUpdate.db2.sh Webservices_UndoDBupdate.db2.sh

- 該当するスクリプト・ファイルを以下のディレクトリーへコピーします。

-  *WC\_installdir*¥samples¥webservices¥bin
-  *WCStudio\_installdir*¥Commerce¥samples¥webservices¥bin
-    *WC\_installdir*/samples/webservices/bin
-  /QIBM/ProdData/CommerceServer55/samples/webservices/bin

+  
+  
+  
+  
+

- c.  以下のディレクトリーから、 `wcs.webservices.enable.sql` スクリプトと `wcs.webservices.disable.sql` スクリプトをコピーします。

-  `WC_installdir\samples\webservices\schema\db2`
-  `WCStudio_installdir\Commerce\samples\webservices\schema\db2`
-    `WC_installdir/samples/webservices/schema/db2`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/schema/db2`

以下のディレクトリーへコピーします。

-  `WC_installdir\schema\db2`
-  `WCStudio_installdir\Commerce\schema\db2`
-    `WC_installdir/schema/db2`
-  `/QIBM/ProdData/CommerceServer55/schema/db2`

-  以下のディレクトリーから、 `wcs.webservices.enable.sql` スクリプトと `wcs.webservices.disable.sql` スクリプトをコピーします。

-  `WC_installdir\samples\webservices\schema\oracle`
-  `WCStudio_installdir\Commerce\samples\webservices\schema\oracle`
-   `WC_installdir/samples/webservices/schema/oracle`

以下のディレクトリーへコピーします。

-  `WC_installdir\schema\oracle`
-  `WCStudio_installdir\Commerce\schema\oracle`
-   `WC_installdir/schema/oracle`

- d. `OrderStatus Web` サービスに関連しないスクリプト・ステートメントをコメント化して、ステップ 7c でコピーした SQL スクリプト・ファイルを更新します。
- e. ステップ 7b でコピーしたスクリプト・ファイルを実行します。
9. WebSphere Commerce マシンを使用している場合は、Commerce Server インスタンスを再始動して変更を有効にします。WebSphere Commerce Studio マシンを使用している場合は、WebSphere Test Environment 内で `WebSphereCommerceServer` サーバーを再始動します。

## OrderStatus サービスのアクセス制御

`OrderStatus Web` サービスに対して、WebSphere Commerce のアクセス制御モデルを使用します。このためには、WebSphere Commerce が要求者を認証して、さらに要求アクションを実行できる権限を要求者が持っているかどうかを検査できるよ

うに、 Web サービス要求時にユーザー ID とパスワードが渡される必要があります。ユーザー ID とパスワードの定義は、 WSDL 定義の中で以下のように行います。

認証情報のための XML スキーマの形式は、以下のようなものになります。

```
<complexType
name="com.ibm.commerce.webservice.datatype.Credentials">
<all>
<element name="password" nillable="true" type="string"/>
<element name="logonId" nillable="true" type="string"/>
</all>
</complexType>
```

認証情報は、テンプレート定義の中の以下のフィールドにマップする必要があります。

- ユーザー・ログインのフィールド名は 'logonId' で FieldInfo は 'CONTROL'。
- パスワードのフィールド名は 'logonPassword' で FieldInfo は 'CONTROL'。

たとえば OrderStatus サービスの場合、ログオン ID とパスワードは、マッピング・テンプレートの中の以下のセクションを用いて OrderStatus コマンドにマップします。

```
<TemplateTag name='OrderStatus20Map'>
<!--Execution Environment -->
<Tag XPath='updateOrderStatusMessage/controlArea/credentials/logonId'
Field='logonId' FieldInfo='CONTROL' />
<Tag XPath='updateOrderStatusMessage/controlArea/credentials/password'
Field='logonPassword' FieldInfo='CONTROL' />
```

このサンプルは、WebSphere Commerce で定義された Web サービスのアクセス制御要件を満たすための典型的なメソッドを示したものです。

## OrderStatus Web サービスでのユーザー役割

OrderStatus サービスでは、コントローラー・コマンドの OrderStatusServiceCmd と OrderStatusCmd を使用します。デフォルトでは、これらのコントローラー・コマンドはリソースとして SellersCmdResourceGroup リソース・グループへ、アクションとして BackendOrderStatusCreate アクション・グループへ追加されます。そのため OrderStatusCmd コマンドと OrderStatusServiceCmd コマンドを実行できるのは、「セラー」の役割が割り当てられているユーザーのみとなります。このことは、認証情報が OrderStatus メッセージの一部として渡されるユーザーは、「セラー」の役割を持っていないかならないことを意味します。デフォルトでは WebSphere Commerce 管理者権限を持っているユーザーは、これらのコマンドのいずれも実行することができます。したがって、このことが当てはまるのは、管理者権限を持っていないユーザーが OrderStatus サービスを使用可能にする場合のみです。

## OrderStatus の WSDL 定義

OrderStatus Web サービスは、以下のディレクトリーにある OrderStatus WSDL 定義に従って定義されます。

-  `WC_installdir%samples%webservices$xml%messaging`
-  `WCStudio_installdir%Commerce%samples%webservices$xml%messaging`

-    `WC_installdir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging`

## 新規のメッセージ・マッピング

OrderStatus Web サービスを使用可能にするために、マッピングの新規セットがマッピング・テンプレートの `webservice_template.xml` ファイルに用意されています。メッセージ・マップのテンプレート文書を以下に示します。

```
<!-- mapping for inbound order status update message -->
<TemplateDocument>
  <DocumentType version='1.0'>ns1:update_WCS_OrderStatus
  </DocumentType>
  <StartElement>ns1:update_WCS_OrderStatus</StartElement>
  <TemplateTagName>OrderStatus20Map</TemplateTagName>
  <CommandMapping>
    <Command CommandName='OrderStatusService'>
      Condition='OrderStatusCommand'
    </Command>
    <Constant Field='redirecturl'>
      SOAP_OrderStatusResponse
    </Constant>
    <Constant Field='SOAP_OrderStatusResponse'>
      SOAP_OrderStatusResponse.jsp
    </Constant>
  </CommandMapping>
</TemplateDocument>
```

このサービスに対応するメッセージ・マッピングの名前は `OrderStatus20Map` で、以下のディレクトリーにある `webservice_template.xml` テンプレート・ファイルに入っています。

-  `WC_installdir\samples\webservices\xml\messaging`
-  `WCStudio_installdir\Commerce\samples\webservices\xml\messaging`
-    `WC_installdir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging`

用意されているメッセージ・マップは、以下のディレクトリーにある WSDL 定義に対応しています。

-  `WC_installdir\samples\webservices\xml\messaging\Orderstatus.wsdl`
-  `WCStudio_installdir\Commerce\samples\webservices\xml\messaging\Orderstatus.wsdl`
-    `WC_installdir/samples/webservices/xml/messaging/Orderstatus.wsdl`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging/OrderStatus.wsdl`

このメッセージ・マップは、与えられている WSDL 定義で生成された Web サービス・クライアントが送信するメッセージを扱います。

メッセージ・マップを定義するには、SOAP 要求に含まれる XPATH 情報が分かっているなければなりません。XPath 情報は WSDL 定義から得ることができます。しかし、SOAP メッセージを調べる方が便利でしょう。次のセクションでは、SOAP メッセージにアクセスしてマッピング・テンプレートを定義する方法を説明します。

OrderStatus メッセージ (OrderStatus20Map) に対応するメッセージ・マップに類似したメッセージ・マップを、対応 SOAP 要求から定義するには、以下のようになります。

1. WSDL 定義ファイルを用いてクライアント・プロキシを生成します。ステップの説明は、53 ページの『付録 B. IBM WebSphere Studio Application Developer の使用法』を参照してください。
2. トランスポート・レベルのインターセプト・ユーティリティを用いて、SOAP 要求と SOAP 応答文書がどのようなものかを調べます。たとえば Apache AXIS 配布で入手できる TCP Monitor など、メッセージをログに記録できる HTTP トンネル・ユーティリティを使用します。アクセス・ポイントを設定して、クライアント・プロキシが、インターセプトするものを指すようにします。こうすると、要求に対応したマッピングの作成や、応答の構成に使用する JSP ファイルの作成をするのに役立ちます。
3. テンプレート定義ファイルの `TemplateDocument` エlementを定義します。これには、XML メッセージを受け取ったときに実行するコマンドの詳細を含めません。詳しくは、WebSphere Commerce オンライン・ヘルプの『テンプレート定義ファイルの `TemplateDocument` Element』のセクションを参照してください。
4. タグ定義のリストを含む `TemplateTag` を定義します。TemplateTag には、WSDL 定義にあるすべての Element についてのマッピングを必ず含めてください。これは、クライアントから要求が送られてきたときに、SOAP 要求内のすべての Element のデータを取り込めるようにするためです。各タグ定義で、着信 XML メッセージにある Element または属性を、コマンド・パラメーターにマップします。WebSphere Commerce オンライン・ヘルプの『テンプレート定義ファイルの `TemplateTag` Element』のセクションを参照してください。

## メッセージ・マップの構成

SOAP メッセージ・マップは、Web サービス・テンプレート定義ファイル (`webservice_template.xml`) を用いて、着信 SOAP 要求をコマンドのプロパティにマップします。したがって、そのテンプレートを変更することにより、SOAP メッセージ・マップを構成することができます。インバウンド・メッセージのテンプレート定義ファイルについての詳細は、WebSphere Commerce オンライン・ヘルプを参照してください。

別の WSDL 定義がある場合やサービス定義を変更する場合は、テンプレートのマップを変更する必要があります。マップは、コマンド・パラメーターにマップされた XML メッセージの XPath 定義で構成されます。この場合は、新しいサービス定義のデータ・タイプスキーマに合わせて XPath 構造を変更する必要があります。

## OrderCreate および OrderStatus Web サービスを使用不可にする

このセクションでは、WebSphere Commerce が提供する OrderCreate および OrderStatus Web サービスを使用不可にする方法を説明します。

1. 以下の JSP ファイルを削除します。

- SOAP\_PurchaseOrderResponse.jsp
- SOAP\_GenericApplicationError.jsp
- SOAP\_GenericSystemError.jsp
- SOAP\_OrderStatusResponse.jsp

これらは以下のディレクトリーにあります。

-  *WAS\_installdir*¥installedApps¥ *cell\_name*¥  
*WC\_instance\_name*.ear¥Stores.war
-  *WCStudio\_workspacedir*¥Stores¥Web Content
-    *WAS\_installdir/installedApps/cell\_name/*  
*WC\_instance\_name*.ear/Stores.war
-  *WAS\_userdir/installedApps/cell\_name/*  
*WC\_instance\_name*.ear/Stores.war

2. 以下のディレクトリーにある webservice\_template.xml ファイルを、

-  *WC\_installdir*¥xml¥messaging
-  *WCStudio\_installdir*¥Commerce¥xml¥messaging
-    *WC\_installdir/xml/messaging*
-  /QIBM/UserData/CommerceServer55/instances/  
*instance\_name/xml/messaging*

以下の例のような空の ECTemplate ノードを含む空のテンプレートに置き換えます。

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE ECTemplate SYSTEM 'ec_template.dtd'>
<ECTemplate>
</ECTemplate>
```

3. 以下のディレクトリーにある wcs.webservices.disable.sql スクリプトを見つけます。

 DB2 データベースを使用している場合

-  *WC\_installdir*¥schema¥db2
-  *WCStudio\_installdir*¥Commerce¥schema¥db2
-    *WC\_installdir/schema/db2*
-  /QIBM/UserData/CommerceServer55/instances/*instance\_name*/schema

 Oracle データベースを使用している場合

-  `WC_installdir¥schema¥oracle`
  -  `WCStudio_installdir¥Commerce¥schema¥oracle`
  -   `WC_installdir/schema/oracle`
4. ステップ 3 で見つけた `wcs.webservices.disable.sql` スクリプトをテキスト・エディターで開いて、以下の値を `wcs.webservices.enable.sql` スクリプト内のものと同じにします。
- `storeent_id` - ストアの固有 ID。
  - `OrgCode1` - サプライヤー組織に対応する `ORGCODE` テーブル内のエントリーの基本キー。
  - `OrgCode2` - バイヤー組織に対応する `ORGCODE` テーブル内のエントリーの基本キー。
  - `MemberId` - バイヤー組織が属するメンバー・グループの基本キー。
5. ステップ 4 で説明した変更を完了後、該当するコマンド・スクリプト・ファイルを実行します。

 **DB2** DB2 データベースを使用している場合

-   DB2 のコマンド・ウィンドウから、`Webservices_UndoDBUpdate.db2.bat` を実行します。
-    コマンド・ウィンドウから、`Webservices_UndoDBUpdate.db2.sh` を実行します。
-  `QSH` セッションを開始します。コマンド `/QIBM/ProdData/CommerceServer55/samples/webservices/bin/Webservices_UndoDBUpdate.db2.sh` を実行します。

 **Oracle** データベースを使用している場合

-   DB2 のコマンド・ウィンドウから、`Webservices_UndoDBUpdate.oracle.bat` を実行します。
-   コマンド・ウィンドウから、`Webservices_UndoDBUpdate.oracle.sh` を実行します。

ログ・ファイルを検査して、実行が正常に完了していることを確認してください。ログ・ファイルの名前と場所は、スクリプト・ファイルの実行完了後に表示されます。

**注:** `OrderCreate` および `OrderStatus Web` サービスのみを使用不可にする場合、または `OrderFulfillment Web` サービスを使用可能にしていない場合は、ステップ 4 に示したスクリプト・ファイルの中の以下の SQL ステートメントをコメント化します。

- `delete from cmdreg where interfacename='com.ibm.commerce.webservice.utils.WSDLFinderCmd' and classname='com.ibm.commerce.webservice.utils.WSDLFinderCmdImpl';`
- `delete from cmdreg WHERE storeent_id=storeent_id AND interfacename='com.ibm.commerce.order.commands.ExtOrderProcessCmd';`

ここで *storeent\_id* は、ストアの固有 ID です。

6. **Studio** 17 ページの『サンプル・コードのインポート』でインポートしたサンプル・コードを削除した後に `WebSphereCommerceServerExtensionsLogic` プロジェクトを再ビルドします。
7. WebSphere Commerce マシンを使用している場合は、Commerce Server インスタンスを再始動して変更を有効にします。WebSphere Commerce Studio マシンを使用している場合は、WebSphere Test Environment 内で `WebSphereCommerceServer` サーバーを再始動します。



## 第 4 章 WebSphere Commerce をサービス・リクエスターとして使用可能にする

このセクションでは、外部システムで定義されている Web サービスを WebSphere Commerce が呼び出せるようにする方法を説明します。ここでは WebSphere Commerce が Web サービスを検索して呼び出せるようにする付属のサンプルを解説します。さらに、OrderFulfillment サンプル Web サービスのインプリメンテーションの詳細を説明します。

### アーキテクチャー

このアーキテクチャーのメッセージ・フローは同期しています。つまり WebSphere Commerce は、外部システムからの応答を待ってから次の処理に進みます。Web サービスを呼び出して外部システムと対話するタスク・コマンドがインプリメントされています。

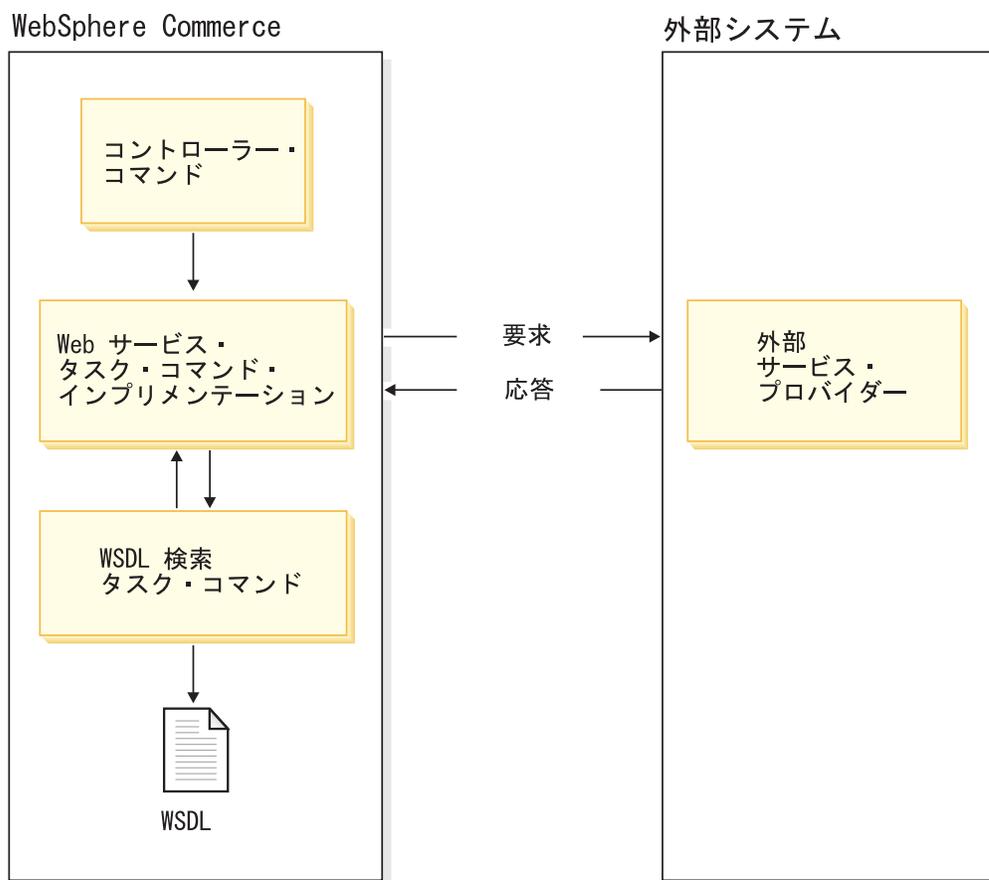


図 5. WebSphere Commerce をサービス・リクエスターとして使用可能にするアーキテクチャー

WebSphere Commerce プログラミング・モデルでは、コマンドをオーバーライドすることによって、既存のビジネス・プロセスの振る舞いをカスタマイズすることが

できます。コントローラー・コマンドは、ビジネス・プロセスの実行を完了するまで、いくつものタスク・コマンドを呼び出します。コントローラー・コマンドの振る舞いを変更する唯一の方法は、コントローラー・コマンドが呼び出すタスク・コマンドのインプリメンテーションを新規に作成することです。

コントローラー・コマンドの一部として呼び出されるタスク・コマンドを調べるには、WebSphere Commerce オンライン・ヘルプの『コマンド、タスク、およびテーブルの相互参照』のセクションを参照してください。外部システムにある Web サービスを WebSphere Commerce ビジネス・プロセスが呼び出すときの対話は、以下のように行われます。

1. コントローラー・コマンドが登録済みタスク・コマンドを呼び出します。ここでは Web サービスを呼び出して外部システムと対話するタスクがインプリメントされています。このタスク・コマンドには、外部システムと統合するためのコードが含まれます。
2. タスク・コマンドは指定された Web サービスのアクセス・ポイントを検索します。
3. タスク・コマンドはサービス・アクセス・ポイントとその入力パラメーターを指定して、Web サービス・プロキシーを呼び出します。
4. プロキシーは Web サービスを呼び出し、その応答を受け取ってタスク・コマンドに戻します。

## サービス・インプリメンテーションの検索

外部システムで定義されている Web サービスを WebSphere Commerce が呼び出せるようにするには、まず目的のサービス・タイプをホスティングしているサービス・プロバイダーを見つけなければなりません。Web サービスを呼び出すには、タスク・コマンドに対して特定のサービス・プロバイダーを指定します。これを柔軟に行えるように、サービス・プロバイダーを見つけるための WSDLFinderCmd タスク・コマンドを組み込みます。このタスク・コマンドのサンプル・インプリメンテーションが提供されており、以下のディレクトリーにあります。

-  `WC_installdir¥samples¥webservices¥com¥ibm¥commerce¥webservice¥utils`
-  `WCStudio_installdir¥Commerce¥samples¥webservices¥com¥ibm¥commerce¥webservice¥utils`
-    `WC_installdir/samples/webservices/com/ibm/commerce/webservice/utils`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/com/ibm/commerce/webservice/utils`

サービス・インプリメンテーションを探す唯一の方法は、サービス・レジストリーを調べることです。ただし WebSphere Commerce バージョン 5.5 に付属のサンプルでは、ローカル・ファイル・システムを用いて Web サービスの WSDL 定義を検索します。WSDLFinderCmdImpl は、`instance_name.xml` 構成ファイルにローカルに置かれている登録済み WSDL 定義ファイルを探します。WSDL サービス定義を登録するには、以下のディレクトリーにある `instance_name.xml` 構成ファイルを編集します。

-  `WC_installdir¥instances¥instance_name¥xml`
-  `WCStudio_installdir¥Commerce¥instances¥instance_name¥xml`
-    `WC_installdir/instances/instance_name/xml`
-  `/QIBM/UserData/CommerceServer55/instances/instance_name/xml`

そこで以下を行います。

1. SOAPWSDLFinderInfo エレメントを見つけます。このエレメントには、WSDL 定義パスと WSDL ファイルを登録するための属性があります。以下は、`instance_name.xml` ファイルからのそれらのパラメーターの抜粋です。

```
<SOAPWSDLFinderInfo
LocalWSDLFiles="OrderFulfillment.wsdl"
LocalWSDLPath="WC_installdir/xml/messaging"
UDDIQueryURL=""
UDDIPublishURL=""
WSDLFinderImplClassName=" "
display="false" />
```

2. LocalWSDLFiles の値はローカル・ファイル・システムにある WSDL 定義をコマンドで区切ったリストに設定し、LocalWSDLPath の値はこれらのファイルがあるディレクトリーに設定します。
3. このサンプルではサービス定義をローカル・ファイル・システムで探すため、UDDIQueryURL と UDDIPublishURL の値は空ストリングに設定します。
4. WSDLFinderImplClassName の値を空ストリングに設定します。

パートナーを動的に探して、そのパートナーがインプリメントしてサービス・レジストリーに発行済みのサービスを呼び出すには、WebSphere Commerce バージョン 5.5 に付属の WSDLFinderCmdImpl サンプル・タスク・コマンドのインプリメンテーションを新規作成して登録する必要があります。カスタム・インプリメンテーションでは WSDL 定義を調べるためのコードを組み込みます。この場合、UDDIQueryURL および UDDIPublishURL フィールドを用いて UDDI レジストリーを登録します。タスク・コマンドをオーバーライドする方法についての詳細は、「WebSphere Commerce プログラミング・ガイドとチュートリアル」を参照してください。

## サービス・インプリメンテーションの呼び出し

コントローラー・コマンドの一部として呼び出されるタスク・コマンドを調べるには、WebSphere Commerce オンライン・ヘルプの『コマンド、タスク、およびテーブルの相互参照』のセクションを参照してください。

次の表は、OrderProcessCmd の一部として実行されるタスクを示したものです。

タスク・コマンド名	説明
CheckOrderCmd	CheckOrderCmd タスク・コマンドは、「送信済み」状態のオーダーが処理されたかどうかを検査します。
DoPaymentCmd	OrderProcessCmd コントローラー・コマンドは DoPaymentCmd タスク・コマンドを呼び出してオーダーの支払いを処理します。
ExtOrderProcessCmd	OrderProcess コマンドの完了直前に必要な追加処理を実行します。

UpdateInventoryCmd	UpdateInventory タスク・コマンドはアイテムの在庫を更新します。
--------------------	---

この表から、OrderProcess コマンドの振る舞いを、外部の配送センターに対して配送要求を送信するようにカスタマイズするには、ExtOrderProcessCmd タスク・コマンドをオーバーライドすればよいことが分かります。ExtOrderProcessCmd の新規インプリメンテーションでは、OrderFulfillment サービスをインプリメントしたパートナーを調べて、そのサービスを呼び出します。

## WebSphere Commerce が呼び出せる外部システムのサンプル Web サービス

このセクションでは、OrderFulfillment サンプル Web サービスを使用可能にする方法を説明します。

OrderFulfillment サンプルは、WebSphere Commerce マシン上でも、WebSphere Commerce Studio がインストールされている開発マシン上でも、どちらでも使用可能にすることができます。WebSphere Commerce Studio マシン上でこのサンプルを使用可能にする場合は、このセクションの説明に従い " Studio "

### OrderFulfillment サービス

このセクションでは、外部システムにインプリメントされている Web サービスを WebSphere Commerce が呼び出せるようにする方法を説明します。

OrderFulfillment サービスを例に、サービス・リクエスターとしての WebSphere Commerce の動作を説明します。OrderFulfillment サンプル Web サービスを使用するには、WebSphere Commerce システム内に発行済みストアが必要です。

このサンプル Web サービスを WebSphere Commerce 環境で使用可能にする場合は、WebSphere Commerce Studio 環境も必要です。

### インターフェース WSDL 定義

OrderFulfillment サービスのビジネス・ロジックは、OrderFulfillment サービスの定義済みサービス・インターフェースを対象として設計されています。

OrderFulfillment サービスのインターフェース定義 WSDL ファイルは、参考のため、および既存のサービスとのデフォルトの静的バインディングのために、ローカル・ファイル・システムに置かれています。

### タスク・コマンド

OrderProcessCmd の一部として呼び出される ExtOrderProcessServiceCmdImpl コマンドには、外部サービスと対話するための ExtOrderProcessServiceCmd インターフェースがインプリメントされています。これにより Web サービスの要求がプロキシを介して開始されます。このタスク・コマンドは送信済みオーダーの処理を、OrderFulfillment 要求をそれぞれの配送センターに送信することにより行います。

OrderFulfillment サービスを使用可能にするステップは、以下のとおりです。

1. WebSphere Commerce を、第 3 章の 23 ページの『データベース・テーブルの移植』の説明に従って、すでにサービス・プロバイダーとして使用可能にしている場合は、このステップは実行しないでください。ただし、これまでに OrderFulfillment コマンドのセクションをコメント化している場合は、このステップを実行する必要があります。

ExtOrderProcessServiceCmdImpl と WSDLFinderCmdImpl を WebSphere Commerce Server コマンド・レジストリーに登録するには、これらのコマンドに関連した SQL ステートメントを実行します。この SQL スクリプト・ファイルは、以下のディレクトリーにあります。

▶ DB2 DB2 データベースを使用している場合

- ▶ Windows `WC_installdir\samples\webservices\schema\db2\wcs.webservices.enable.sql`
- ▶ Studio `WCStudio_installdir\Commerce\samples\webservices\schema\db2\wcs.webservices.enable.sql`
- ▶ AIX ▶ Linux ▶ Solaris `WC_installdir/samples/webservices/schema/db2/wcs.webservices.enable.sql`
- ▶ 400 `/QIBM/ProdData/CommerceServer55/samples/webservices/schema/db2/wcs.webservices.enable.sql`

▶ Oracle Oracle データベースを使用している場合

- ▶ Windows `WC_installdir\samples\webservices\schema\oracle\wcs.webservices.enable.sql`
- ▶ Studio `WCStudio_installdir\Commerce\samples\webservices\schema\oracle\wcs.webservices.enable.sql`
- ▶ AIX ▶ Solaris `WC_installdir/samples/webservices/schema/oracle/wcs.webservices.enable.sql`

以下は、実行する必要があるこのスクリプト・ファイルの SQL ステートメントです。

```
insert into cmdreg (storeent_id, interfacename, description,
  classname, properties, lastupdate, target) values(0, 'com.ibm.commerce.
webservice.utils.WSDLFinderCmd', 'Wsd1 finder class for webservices',
com.ibm.commerce.webservice.utils.WSDLFinderCmdImpl',
null, null, 'Local');
```

```
insert into cmdreg (storeent_id, interfacename, description,
  classname, properties, lastupdate, target) values(storeent_id,
'com.ibm.commerce.order.commands.ExtOrderProcessCmd', '
webservices implementation of ExtOrderProcessCmd',
'com.ibm.commerce.webservice.order.commands.ExtOrderProcessServiceCmdImpl',
null,null,'Local');
```

ここで `storeent_id` は、ストアの固有 ID です。

2. OrderFulfillment サービスの検索を使用可能にするには、以下のようにします。

- a. `instance_name.xml` 構成ファイルの中から SOAPWSDLFinderInfo エレメントを見つけます。
- b. LocalWSDLFiles と LocalWSDLPath の値が正しく、WSDL 定義ファイルがあるディレクトリーを示していることを確認します。SOAPWSDLFinderInfo エレメントは、以下のようになっているはずです。

```
<SOAPWSDLFinderInfo
  LocalWSDLFiles="OrderFulfillment.wsdl"
  LocalWSDLPath="WC_installdir/xml/messaging"
  WSDLFinderImplClassName=""
  display="false" />
```

- c. 以下のディレクトリーにあるサンプル OrderFulfillment.wsdl WSDL ファイルをコピーします。

-  `WC_installdir\samples\webservicess\xml\messaging`
-  `WCStudio_installdir\Commerce\samples\webservicess\xml\messaging`
-    `WC_installdir/samples/webservicess/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservicess/xml/messaging`

以下のディレクトリーへコピーします。

-  `WC_installdir\xml\messaging`
-  `WCStudio_Installdir\Commerce\xml\messaging`
-    `WC_installdir/xml/messaging`
-  `/QIBM/UserData/CommerceServer55/instances/instance_name/xml/messaging`

3. OrderFulfillment.wsdl ファイルで定義されているサービスのサービス・プロバイダーを特定します。そのサービス・プロバイダーの情報を用いて OrderFulfillment.wsdl を更新します。

- a. ステップ 2c でコピーした OrderFulfillment.wsdl ファイルを開きます。
- b. サービス定義ノード ("service name" で始まるセクション) を見つけます。このノードは以下のようになっています。

```
<service name=" POCreateService ">
<port binding="tns: POCreateBinding " name=" POCreatePort>
<soap:address location="http://localhost:8080/WebservicessForWCBWeb/
  servlet/rpcrouter"/>
</port>
</service>
```

- c. 強調表示されている既存の URL を、ターゲット Web サービスの URL に変更します。
  - d. このファイルを保管して閉じます。
4. OrderCreate または OrderStatus サンプル Web サービスをすでに使用可能にしている場合は、ステップ 4 および 5 は無視してください。そうでない場合はこのステップを進めてください。OrderFulfillment Web サービス用のサンプル

ル・コードをデプロイするには、WebSphere Commerce Studio がインストールされ、構成されている開発マシンが必要です。以下に示すステップでは WebSphere Studio Application Developer が必要であり、これらのステップはその開発マシン上で実行することを対象としています。

5. サンプル Web サービスを編集または変更する場合は、サンプル・ソース・コードを WebSphere Commerce Studio にインポートする必要があります。このコードをインポートするには、以下のようにします。
  - a. 「スタート」 > 「プログラム」 > 「IBM WebSphere Commerce Studio」 > 「WebSphere Commerce 開発環境 (WebSphere Commerce development environment)」を選択して、WebSphere Commerce 開発環境を開きます。
  - b. J2EE パースペクティブに切り替えて、「J2EE ナビゲーター (J2EE Navigator)」ビューを選択します。
  - c. **WebSphereCommerceServerExtensionsLogic** プロジェクトを拡張表示します。 **src** ディレクトリーを右マウス・ボタンでクリックして「インポート」を選択します。インポート・ウィザードが開きます。
  - d. 「ファイル・システム」を選択して、「次へ」をクリックします。
  - e. 「ブラウズ」をクリックして次のディレクトリーを見つけます。  
`WCStudio_installdir\Commerce\samples\webservices\com`
  - f. 「全選択」をクリックし、次に「終了」をクリックします。
  - g. **WebSphereCommerceServerExtensionsLogic** プロジェクトを右マウス・ボタンでクリックして「プロパティー」を選択します。
  - h. 「Java ビルド・パス (Java Build Path)」をクリックして、次に「ライブラリー (Libraries)」タブを選択します。
  - i. 「変数の追加 (Add variable)」をクリックします。変数として SOAPJAR と XERCES を選択して「OK」をクリックします。
  - j. 「OK」をクリックします。
  - k. **WebSphereCommerceServerExtensionsLogic** プロジェクトを右マウス・ボタンでクリックして「プロジェクトの再ビルド (Rebuild project)」を選択します。
6. サンプル Web サービスを WebSphere Commerce Server 上で使用可能にするには、このコードをターゲットの WebSphere Commerce Server にデプロイする必要があります。新規コマンドをデプロイする方法についての詳細は、「WebSphere Commerce プログラミング・ガイドとチュートリアル」を参照してください。
7. WebSphere Commerce マシンを使用している場合は、Commerce Server インスタンスを再始動して変更を有効にします。WebSphere Commerce Studio マシンを使用している場合は、WebSphere Test Environment 内で WebSphereCommerceServer サーバーを再始動します。

## ビジネス・オブジェクト

ExtOrderProcessServiceCmdImpl タスク・コマンドは、Web サービス・プロキシヤーが SOAP XML 要求メッセージを構成するのに使用する一連のビジネス・オブジェクトを取り込みます。この要求メッセージには OrderFulfillment WSDL 定義の要素が反映されます。OrderFulfillment サービスは OrderFulfillment

WSDL 定義から生成される一連のサンプル・ビジネス・オブジェクトを備えています。WebSphere Commerce が外部 Web サービスを呼び出す場合は、53 ページの『付録 B. IBM WebSphere Studio Application Developer の使用法』の説明に従って IBM WebSphere Studio を使用して、または他のツールを使用して、外部 Web サービスのためのクライアント・プロキシとビジネス・オブジェクトを生成します。

次の表は、OrderFulfillment サービスで使用するパッケージとそのビジネス・オブジェクトを示したものです。

パッケージ名	オブジェクト名	説明
com.ibm.commerce.webservice.datatype	Address	このパッケージは、さまざまな Web サービス・メッセージで使用されるすべてのオブジェクトで構成されています。
	ContactInfo	
	ContactPersonName	
	ControlArea	
	Credentials	
	Email	
	InvoiceInfo	
	ProductDimension	
	ProductMeasurement	
	ProductWeight	
	RequisitionerID	
	Telephone	
	UserData	
UserDataField		

パッケージ名	オブジェクト名	説明
com.ibm.commerce. webservice.datatype. orderfulfillment	BillToInfo	このパッケージは、 OrderFulfillment XML メッセージに固有のデータを含むすべてのオブジェクトを含んでいます。  Report_NC_PurchaseOrder はプロキシに送信される親ラッパー・オブジェクト、 OrderFulfillmentStatus は応答として受信するオブジェクトです。
	BuyOrgAccountingDetail	
	BuyOrgInfo	
	DataArea	
	DateTimeReference	
	ItemShippingSchedule	
	ItemUnitPrice	
	MerchantInfo	
	MonetaryAmount	
	OrderFulfillmentStatus	
	PCardInfo	
	Report_NC_PurchaseOrder	
	ReportPO	
	ReportPOHeader	
	ReportPOItem	
	RequisitionerInfo	
	ServiceAllowanceCharge	
	ShipDateReference	
	ShippingCarrierInfo	
	ShipToInfo	
TaxInfo		
TotalPriceInfo		

## サンプルのテスト

WebSphere Commerce の発行済みストアにオーダーを発行すると、OrderFulfillment Web サービスが呼び出されます。この Web サービスは、配送センターにオーダーを発行する外部サービス・プロバイダーに接続します。このサービス・プロバイダーから発行済みオーダーの ID が戻されると、WebSphere Commerce データベース内の ORDERS テーブルの、ORMORDER 列の OrderID が更新されます。

## OrderFulfillment Web サービスを使用不可にする

このセクションでは、WebSphere Commerce が提供するサンプル OrderFulfillment Web サービスを使用不可にする方法を説明します。

1. デプロイした `webservices.jar` JAR ファイルを削除して OrderFulfillment Web サービスを使用可能にします。
2. 以下のディレクトリーにある `OrderFulfillment.wsdl` ファイルを削除します。
  -  `WC_installdir\xml\messaging`
  -  `WCStudio_installdir\Commerce\xml\messaging`
  -    `WC_installdir/xml/messaging`
  -  `/QIBM/UserData/CommerceServer55/  
instances/instance_name/xml/messaging`
3. `instance_name.xml` 構成ファイルから `OrderFulfillment.wsdl` ファイルの参照を除去します。
4. まだ SQL スクリプトを実行して `OrderCreate` および `OrderStatus` Web サービスを使用不可にしていない場合、または `OrderCreate` および `OrderStatus` Web サービスを使用不可にするときに以下の SQL ステートメントをコメント化した場合は、以下の SQL ステートメントを実行します。

```
delete from cmdreg where interfacename='com.ibm.commerce.webservice.utils.  
WSDLFinderCmd' and classname='com.ibm.commerce.webservice.utils.  
WSDLFinderCmdImpl';  
delete from cmdreg where storeent_id= storeent_id  
and interfacename='com.ibm.commerce.order.commands.  
ExtOrderProcessCmd';
```

ここで `storeent_id` は、ストアの固有 ID です。

この SQL ステートメントは、以下のスクリプト・ファイルに入っています。

### DB2 データベースを使用している場合

-   DB2 のコマンド・ウィンドウから、`Webservices_UndoDBUpdate.db2.bat` を実行します。
-    コマンド・ウィンドウから、`Webservices_UndoDBUpdate.db2.sh` を実行します。
-  QSH セッションを開始します。 `Webservices_UndoDBUpdate.db2.sh` コマンドを実行します。

### Oracle データベースを使用している場合

-   DB2 のコマンド・ウィンドウから、`Webservices_UndoDBUpdate.oracle.bat` を実行します。
-   コマンド・ウィンドウから、`Webservices_UndoDBUpdate.oracle.sh` を実行します。

ログ・ファイルを検査して、実行が正常に完了していることを確認してください。ログ・ファイルの名前と場所は、スクリプト・ファイルの実行完了後に表示されます。

5.  17 ページの『サンプル・コードのインポート』でインポートしたサンプル・コードを削除した後に `WebSphereCommerceServerExtensionsLogic` プロジェクトを再ビルドします。
6. WebSphere Commerce マシンを使用している場合は、Commerce Server インスタンスを再始動して変更を有効にします。WebSphere Commerce Studio マシンを使用している場合は、WebSphere Test Environment 内で `WebSphereCommerceServer` サーバーを再始動します。



---

## 付録 A. サンプル WSDL 定義ファイル

WebSphere Commerce バージョン 5.5 に付属のサンプル Web サービスの WSDL 定義は、以下のディレクトリーにあります。

-  `WC_installdir\samples\webservices\xml\messaging`
-  `WCStudio_installdir\Commerce\samples\webservices\xml\messaging`
-    `WC_installdir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/  
samples/webservices/xml/messaging`

Web サービス名	ファイル名
OrderFulfillment	OrderFulfillment.wsdl
OrderCreate	OrderCreate.wsdl
OrderStatus	OrderStatus.wsdl



---

## 付録 B. IBM WebSphere Studio Application Developer の使用法

Web サービスのコードは、さまざまなツールを用いて生成することができます。Web サービスのこのインプリメンテーションで提供されているサンプル・コードは、IBM WebSphere Studio Application Developer を用いて生成されています。このセクションに記載している情報は、IBM WebSphere Studio Application Developer バージョン 5.0 上で検証されたものです。WebSphere Commerce がない場合、または WebSphere Commerce Studio を使用していない場合に、WebSphere Commerce が提供する Web サービスにアクセスする Web サービス・クライアントを開発するときは、Web サービス・クライアントのコードを生成できるツールであればどのツールでも使用できます。

この付録では、WebSphere Commerce で提供しているサービスのテスト・クライアントを、WebSphere Studio を用いて生成する方法を説明します。内容としては、WebSphere Studio への WSDL 定義ファイルのインポート、テスト・クライアントの生成、テスト・クライアントの実行、TCP Monitor サーバー・インスタンスの作成と構成、TCP Monitor による SOAP メッセージのトレースがあります。

Web Services Toolkit (WSTK) を用いてクライアント・コードとデータ・タイプを生成することもできます。Web Services Toolkit についての詳細は、<http://www.alphaworks.ibm.com/tech/webservicestoolkit> を参照してください。WSTK は Web サービスにおけるこれからのテクノロジーを示しており、また最新の Web サービスの仕様を理解するのに役立ちます。ただ、Web サービスの製品レベルの開発環境としては、IBM WebSphere Studio Application Developer をお勧めします。

もう一つの方法として、Apache SOAP インプリメンテーションをクライアントとして使用して、クライアント・コードとデータ・タイプを手動で作成するという方法があります。詳しくは、Apache SOAP の資料を参照してください。

このセクションでは、WebSphere Studio Application Developer についての知識があることを前提としています。WebSphere Studio を使用した Web サービスやエンタープライズ・アプリケーションの開発に詳しくない方は、まず WebSphere Studio の資料または IBM developerworks サイト (<http://www.ibm.com/developerworks/webservices>) にあるチュートリアルをお読みください。

---

### WebSphere Studio Application Developer への WSDL のインポート

WSDL を WebSphere Studio にインポートする前に、Web アプリケーションが作成されていて、その Web アプリケーションが属するエンタープライズ・アプリケーションも作成済みであることを必ず確認してください。Web アプリケーションがない場合は、それを作成してからこの説明に進んでください。WebSphere Commerce 開発環境がセットアップされている場合は、エンタープライズ・アプリケーションおよび Web アプリケーションを新規に作成しないでください。その代わりに WebSphereCommerceServerExtensionsLogic プロジェクトをご使用ください。

IBM WebSphere Studio を用いてクライアント・コードとデータ・タイプを生成する手順の概要は、以下のとおりです。

1. WSDL 定義をインポートする Web プロジェクト内に、 wsdl というフォルダーを新規作成します。 Web プロジェクトがない場合は Web プロジェクトを作成します。
2. wsdl フォルダーを右マウス・ボタンでクリックします。
  - a. 「インポート」 > 「ファイル・システム」 を選択して、「次へ」をクリックします。
  - b. 「ブラウズ」をクリックして、 WSDL 定義ファイルを保管するフォルダーを指定します。
  - c. 「OK」をクリックします。 WSDL 定義は、以下のディレクトリーにあります。

-  `WC_installdir\samples\webservices\xml\messaging`
-  `WCStudio_installdir\Commerce\samples\webservices\xml\messaging`
-    `WC_installdir/samples/webservices/xml/messaging`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging`

3. 右側のパネルで WebSphere Studio にインポートする WSDL ファイルを選択して、「終了」をクリックします。

---

## クライアント・コードとデータ・タイプの生成

WSDL 定義を WebSphere Studio にインポート後、 Web サービスのテストに使用するクライアント・プロキシー・コードとサンプル・テスト・クライアントを生成することができます。クライアント・プロキシーは Java クラスで、これを用いて Java クライアントから Web サービスにアクセスすることができます。サンプル・テスト・クライアントにはサンプル JSP があり、これを用いて WebSphere Studio から Web サービスをテストすることができます。クライアント・プロキシーとサンプル・テスト・クライアントを生成するには、以下のようになります。

**注:** WebSphere Commerce から見て外部のクライアントによる Web サービスにアクセスするクライアント・コードを生成する場合も、以下の一連の手順で行うことができます。唯一の相違点は、この場合は最初に外部サービスのインプリメンテーションの定義をサービス・プロバイダーから取得して、その WSDL 定義を WebSphere Studio にインポートしなければならないということです。

1. Web パースペクティブに移動して、生成するクライアント・コードを保管する Web プロジェクトを選択します。 Web プロジェクトがない場合は Web プロジェクトを作成します。
2. WSDL 定義ファイルを選択して右マウス・ボタンでクリックし、クライアント・コードを生成します。メニューから「新規」 > 「その他」を選択します。「新規」ウィザードが表示されます。

3. 左側のパネルの「**Web サービス (Web service)**」を選択し、右側のパネルの「**Web サービス・クライアント (Web service client)**」を選択します。「次へ」をクリックします。
4. Web サービス・クライアントのダイアログで、「**Java プロキシ (Java proxy)**」を選択して「次へ」をクリックします。
5. 「Web サービスの WSDL ファイルの選択 (Web Service WSDL File Selection)」ダイアログで、選択済みの WSDL ファイルが選択されていることを確認して、「次へ」をクリックします。
6. 「Web サービスのバインディング・プロキシの生成 (Web Service Binding Proxy Generation)」ダイアログでクラス名を指定し、「次へ」をクリックします。
7. 「生成したプロキシをテストする (**Test the generated proxy**)」を選択します。テスト機能の値を Web サービスのサンプル JSP ファイルに必ず設定してください。「終了」をクリックします。
8. Web パースペクティブに切り替えて「Samples」フォルダーを拡張表示します。テスト JSP クライアントが表示されます。

---

## Web サービスのテスト

このセクションでは、前のセクションで生成したサンプル・テスト・クライアントを用いて、WebSphere Commerce が提供する Web サービスをテストする方法を説明します。Web サービスをテストするには、以下のようになります。

1. WebSphere Test Environment サーバーが少なくとも 1 つあることを確認してください。このサーバーがない場合は、以下のようにして作成します。
  - a. サーバー・パースペクティブに移動します。
  - b. 「ファイル」 > 「新規」 > 「サーバーとサーバー構成 (**Server and Server Configuration**)」を選択します。
  - c. サーバーの名前を入力し、タイプは「**WebSphere バージョン 5.0 (WebSphere version 5.0)**」 > 「**Test Environment**」を選択します。
2. TestClient.jsp を右マウス・ボタンでクリックして「**サーバー上で実行 (Run on Server)**」を選択します。ブラウザ・ウィンドウにテスト・クライアントが表示されます。
3. 左側のフレームで「**getEndPoint()**」メソッドを選択して「**呼び出し (Invoke)**」をクリックします。結果フレームで、戻されたエンドポイントが、接続するサーバーの URL であることを確認します。
4. 戻されたエンドポイントが WebSphere Commerce Server の実行されているサーバーと一致しない場合は、次のようにして URL が WebSphere Commerce Server を指すように設定します。
  - a. 左側のフレームで「**setEndPoint()**」メソッドを選択します。
  - b. 右側のフレームの WebSphere Commerce Server の URL を更新して「**呼び出し (Invoke)**」をクリックします。
5. クライアント・プロキシのビジネス・メソッドを選択します。OrderCreate プロキシの場合のビジネス・メソッドは orderCreate です。
  - a. 右側のフレームに表示されるフォームに、渡すパラメーターの詳細を入力します。

- b. すべての必須フィールドに値を入力します。
- c. 「呼び出し (Invoke)」をクリックします。 SOAP メッセージが構成されて、エンドポイントで識別されるサーバーに送信されます。

Web サービス呼び出しの結果が「結果」フレームに表示されます。

---

## TCP Monitor による SOAP メッセージのトレース

このセクションでは、TCP/IP Monitor サーバー・インスタンスを作成して構成する方法、および TCP/IP Monitor サーバーを用いて SOAP メッセージをトレースする方法を説明します。この内容は、Web サービスをデバッグするときや、開発中の Web サービス用のマッピング・テンプレートを定義するときに役に立ちます。

TCP/IP Monitor は、Web ブラウザーとアプリケーション・サーバー間のすべての要求と応答をモニターする単純なサーバーです。TCP/IP Monitor はクライアントとアプリケーション・サーバー間のすべてのトラフィックを取り込んで表示することができます。

ここでは TCP/IP Monitor サーバー・インスタンスを新規に作成した後に、以下のことを行います。

- TCP/IP Monitor サーバーを、WebSphere Commerce Server に要求を転送するように構成します。
- テスト・クライアントを、TCP/IP Monitor を指すように構成して、SOAP 要求および応答をインターセプトできるようにします。

## TCP Monitor サーバー・インスタンスの作成

TCPMon という名前の TCP/IP Monitor サーバー・インスタンスを作成するには、以下のようにします。

1. メインメニューから、「ファイル」 > 「新規」 > 「その他」を選択します。
2. 左側のパネルで「サーバー (Server)」を、右側のパネルで「サーバーとサーバー構成 (Server and Server Configuration)」を選択して、「次へ」をクリックします。
3. 「サーバーの新規作成とサーバー構成 (Create a New Server and Server Configuration)」ウィンドウで、以下のようにします。
  - a. 「サーバー名 (Server name)」に、たとえば TCPMon を入力します。
  - b. 「サーバー・タイプ (Server type)」に「TCP/IP モニター・サーバー (TCP/IP monitoring server)」を選択します。
  - c. 「次へ」をクリックします。
4. 「サーバーの新規作成とサーバー構成 (Create a New Server and Server Configuration)」ウィンドウで、以下のようにします。
  - a. 「リモート・ホスト」フィールドに WebSphere Commerce Server のホスト名を入力します。
  - b. 「リモート・ポート (Remote port)」フィールドに、WebSphere Commerce Server が listen するポート番号を入力します。WebSphere Commerce をクライアントと同じマシン上で実行している場合は、ホスト名として localhost を使用できます。

- c. 「終了」をクリックします。

## TCP Monitor サーバー・インスタンスの構成

WebSphere Commerce Server のリスナー・ポートが TCP Monitor のリモート・ポートになるように TCP Monitor を構成します。以下のようにします。

1. サーバー・パースペクティブに移動します。
2. サーバー構成ビューにある TCP Monitor インスタンスをダブルクリックします。エディター・ビューの中に次の図のような TCP Monitor のサーバー構成ウィンドウが開きます。

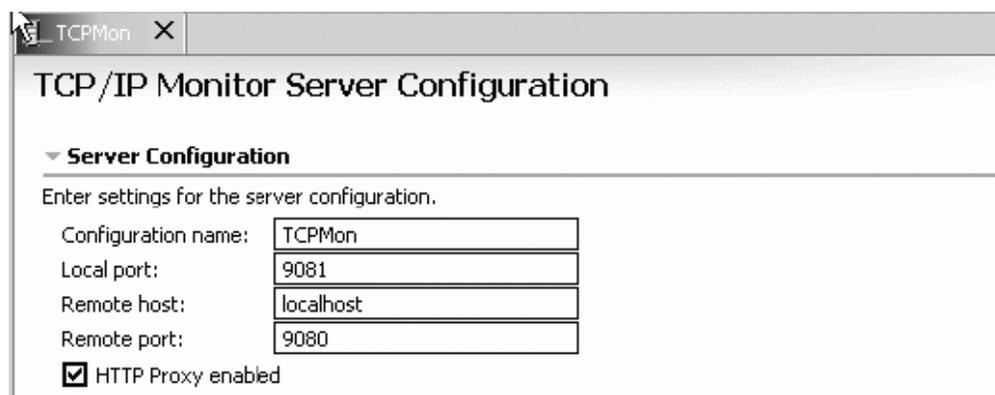


図6. 「TCP/IP Monitor サーバー構成 (TCP/IP Monitor Server Configuration)」の構成

ローカル・ポートは TCP Monitor が listen するポート、リモート・ポートは WebSphere Commerce Server が listen するポートです。

## TCP/IP Monitor による SOAP メッセージのトレース

TCP/IP Monitor を構成後、TCP/IP Monitor を用いて SOAP メッセージをトレースします。これを行うには、生成したサンプル・テスト・クライアントをサーバー上で開始して、要求を TCP/IP Monitor に送信するように構成します。以下のようにします。

1. TCP/IP Monitor と WebSphere Commerce Server が実行中であることを確認してください。
  - a. WebSphere Commerce Server が WebSphere Studio 外で実行されている場合は、WebSphere Application Server 管理コンソールを用いてサーバーを開始します。
  - b. WebSphere Commerce Server が WebSphere Commerce Studio 内で実行されている場合は、「サーバー」パースペクティブに移動します。「サーバー」ビューを開きます。WebSphereCommerceServer サーバーの「状況」が「停止済み」になっている場合は、サーバーを右マウス・ボタンでクリックして「開始」を選択します。「サーバー」ビューに表示される「状況」が「開始済み」になるまで待ちます。
2. Web パースペクティブに移動します。「ブラウズ」で sample¥TestClient.jsp を見つけます。sample¥TestClient.jsp を右マウス・ボタンでクリックして「サーバー上で実行 (Run on server)」を選択します。

3. サーバー選択のウィンドウで、JSP ファイルを実行するサーバーを選択します。「終了」をクリックします。
4. サーバー・クライアントを選択するプロンプトが出されたら、「Web ブラウザー (Web Browser)」を選択して「終了」をクリックします。次の図のような、フレームが 3 つあるブラウザ・ウィンドウが新たに開きます。

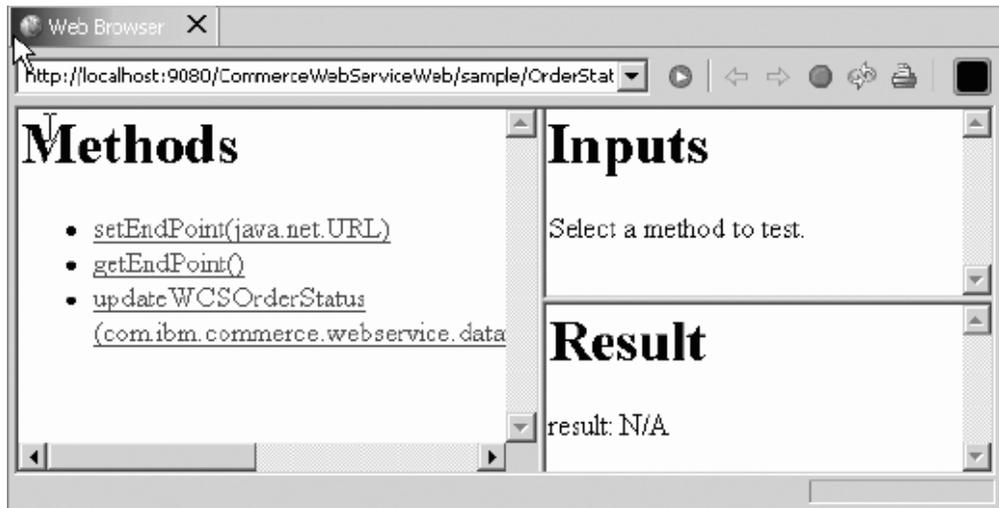


図 7. Web ブラウザーの選択

- a. 次の図のように、「メソッド (Methods)」フレームにある「**getEndPoint()**」メソッドを選択します。



図 8. getEndPoint() メソッドの選択

- b. 「入力 (Inputs)」フレームにある「呼び出し (Invoke)」をクリックします。結果が「結果」フレームに表示されます。
5. 要求を TCP Monitor に送信するようにテスト・クライアントを構成するには、以下のようにします。
  - a. 「メソッド (Methods)」フレームで「**setEndPoint()**」メソッドを選択します。

- b. 「入力 (Inputs)」フレームで TCP Monitor の **url** を入力します。たとえば `http://localhost:9080/webapp/wcs/stores/servlet/` のようにします。ここで 9080 は TCP Monitor が実行しているポート、 `webapp/wcs/stores/servlet/` は WebSphere Commerce 要求サーブレットの Web パスです。

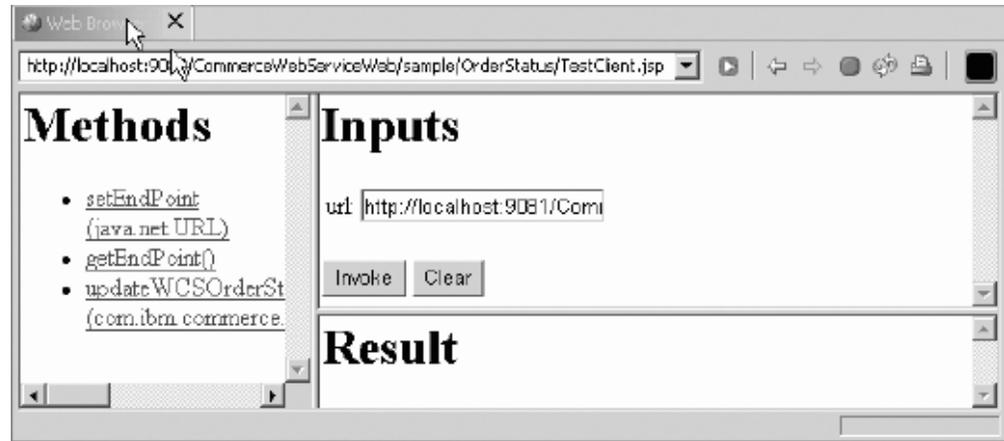


図9. `setEndPoint()` メソッドの選択

- c. 「呼び出し (**Invoke**)」をクリックします。
6. 再び **getEndPoint()** メソッドを呼び出して、エンドポイントを確認します。
7. ビジネス・ロジックを呼び出すには、以下のようにします。
- a. 「メソッド (Methods)」フレームでビジネス・ロジックのメソッド (例: **updateWCSOrderStatus()**) をクリックします。
- b. 次の図のように、「入力 (Inputs)」フレームで必須パラメーターの値をすべて入力します。

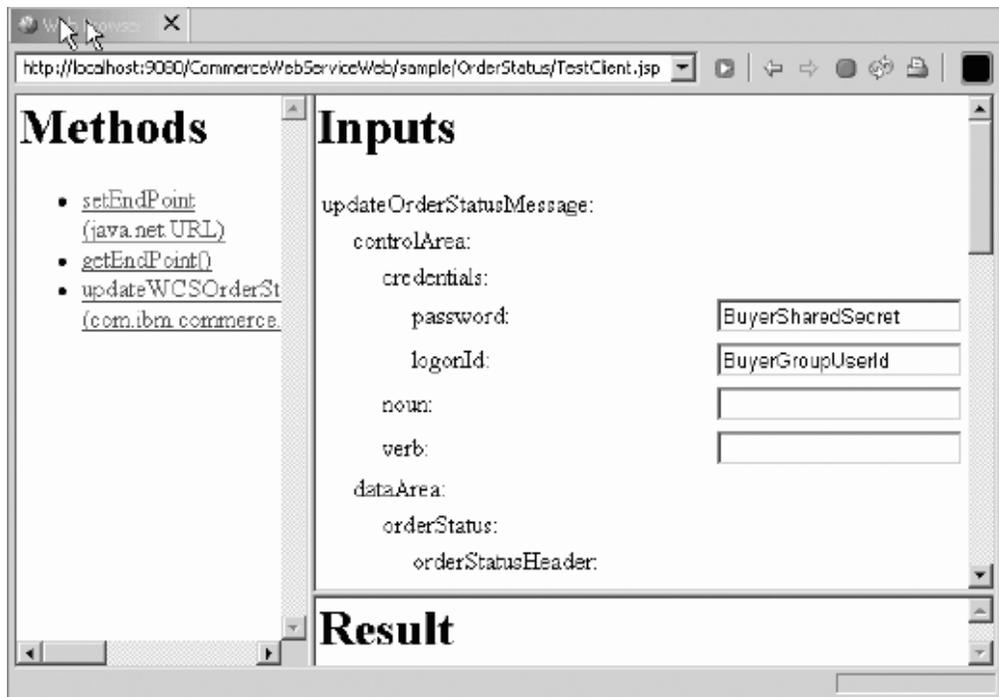


図 10. ビジネス・ロジックの選択

- c. 「呼び出し (Invoke)」をクリックします。
- d. 結果が「結果」フレームに表示されます。
- e. TCP/IP Monitor ビューに移動して、最新のメッセージを選択します。次の図のように SOAP 要求および応答が表示されます。

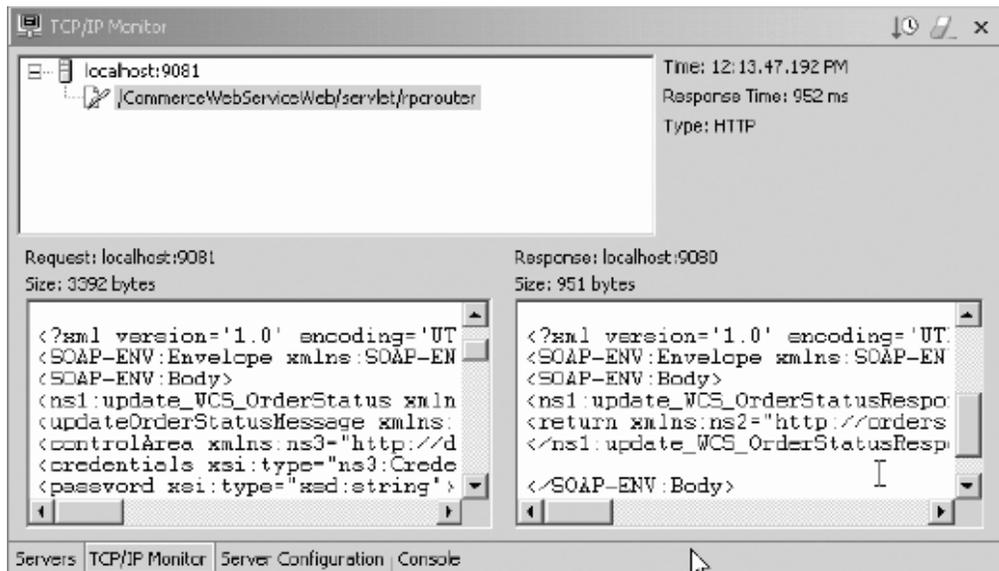


図 11. SOAP 要求および応答の表示

## クライアント・コードのカスタマイズ

クライアント・コードを生成するときは、生成されたデフォルトのクライアント・コードが必ずしも WebSphere Commerce と連動するわけではないということに注意してください。たとえばデータ・タイプの直列化をデフォルトの BeanSerializer クラスを用いて行っている場合、一部のデータ・タイプは WebSphere Commerce がサーバー・エンドで認識できない XML ストリングに変換されます。たとえば OrderCreate サンプル Web サービスは、その種のデータ構造を含んでいます。このデータ構造は WebSphere Commerce コマンドとのインターフェースのためのものであり、UserData データ構造と呼ばれます。UserData データ構造を使用すると、カスタマイズ・データをメッセージ・ドリブン・インターフェースを使ってコマンドに渡すことができます。

以下のセクションでは、UserData データ構造を WebSphere Commerce が使用できるように直列化する方法を説明します。

**注:** 提供されている UserData データ構造はサンプルであり、OrderCreate Web サービスでのオプションです。

WebSphere Commerce では、外部システムが送信 XML メッセージの一部としてカスタムの名前と値の対を組み込むことが可能です。これが行えるのは、USERDATA という XPATH タイプを使用しているためです。これは次に示すサンプル XML メッセージのように、UserData エlementを組み込んで処理します。

```
<UserData>
  <UserDataField name="abc">xyz</UserDataField>
</UserData>
```

UserData Elementに対する XPATH 定義は、次のようになります。

```
<Tag XPath='.../UserData/UserDataField'
  XPathType='USERDATA' />
```

XPATH プロセッサは UserData Elementを名前と値の対「abc=xyz」として解釈します。

次のようなデータ構造がある場合は、Apache SOAP インプリメンテーションに組み込まれている bean シリアライザーが問題になることがあります。Apache SOAP インプリメンテーションでは、データ構造の中のすべてのデータ・タイプが変換されて、XML 文書の中では別々のElementになります。Apache SOAP クライアントを用いて Java オブジェクトを SOAP XML メッセージに変換すると、WebSphere Commerce はその XML を構文解析できなくなるために UserData オブジェクトを形成することができません。

```
class UserData
(
  private String userDataField = null;
  private String name = null;
  // getter and setter methods here ....
)
```

このクラスは XML 構造に直列化されます。

```
<UserData>
  <UserDataField>xyz</UserDataField>
  <name>abc</name>
</UserData>
```

その結果 WebSphere Commerce メッセージ・マッパーは、これを USERDATA 構造として認識することができません。

UserData データ・タイプを必要な形式に直列化できるようにするには、カスタム・シリアライザーを作成して、それを SOAP マッピング・レジストリーに登録する必要があります。カスタム・シリアライザーには、以下に示す marshall メソッドを使用する必要があります。

```
/*
 * serialize the com.ibm.commerce.webservices.datatype.UserData
 * to the corresponding XML schema as expected by WC.
 *
 * @param String inScopeEncStyle
 * @param Class javaType
 * @param Object src
 * @param Object context
 * @param Writer sink
 * @param NSStack nsStack
 * @param XMLJavaMappingRegistry xjmr
 * @param SOAPContext ctx
 * @exception IllegalArgumentException
 * @exception IOException
 */
public void marshall(String inScopeEncStyle,
                    Class javaType,
                    Object src,
                    Object context,
                    Writer sink,
                    NSStack nsStack,
                    XMLJavaMappingRegistry xjmr,
                    SOAPContext ctx)
    throws IllegalArgumentException, IOException
{
    if (!javaType.equals(UserData.class))
    {
        throw new IllegalArgumentException(
            "Can only serialize UserData instances");
    }
    nsStack.pushScope();
    if (src != null)
    {
        SoapEncUtils.generateStructureHeader(inScopeEncStyle,
                                           javaType,
                                           context,
                                           sink,
                                           nsStack,
                                           xjmr);

        UserData data = (UserData)src;
        sink.write("<UserDataField name=" + data.getName() + ">");
        sink.write(data.getValue() +
            "</UserDataField>");
        sink.write("</" + context + ">");
    }
    else
    {
        SoapEncUtils.generateNullStructure(inScopeEncStyle,
                                           javaType,
                                           context,
                                           sink,
                                           nsStack, xjmr);
    }
    nsStack.popScope();
}
```

**注:** WebSphere Commerce バージョン 5.5 で提供されているサンプル・サービスは、このコードと連動するように設計されています。



---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、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。

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 Canada Ltd.  
Office of the Lab Director  
8200 Warden Avenue,  
Markham, Ontario L6G 1C7  
Canada

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。

この文書には、他社のインターネット・サイトへの参照を含む、他社製品の情報が含まれている場合があります。IBM は、そのような情報の正確性、完全性、または使用については何ら責任を負いません。

著作権使用許諾:

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

---

## 商標

IBM、および IBM ロゴは、IBM Corporation の商標です。

以下は、IBM Corporation の商標です。

<b>AS/400</b>	<b>iSeries</b>
<b>AIX</b>	<b>MQSeries®</b>
<b>DB2</b>	<b>S/390</b>
<b>DB2 Universal Database™</b>	<b>zSeries</b>
<b>@server</b>	<b>WebSphere</b>
<b>IBM</b>	

Microsoft、Windows および Windows NT は、Microsoft 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。







Printed in Japan