

IBM WebSphere Commerce



Web 服務實作手冊

5.5 版

IBM WebSphere Commerce



Web 服務實作手冊

5.5 版

注意事項

在使用本資訊及其支援的產品之前，請先閱讀第 53 頁的『注意事項』中的資訊。

第一版，第一次修訂（2003 年 9 月）

本版適用於 IBM WebSphere Commerce 5.5 版，以及所有上述產品之後續版次與修訂，直到新的版本中另有指示為止。請確定您使用的是產品層次的正確版本。

本版適用於 IBM WebSphere Commerce Studio Business Developer Edition 5.5 版以及 IBM WebSphere Commerce Studio Professional Developer Edition 5.5 版以及所有上述產品之後續版次與修訂，直到新的版本中另有指示為止。請確定您使用的是產品層次的正確版本。

請透過您的 IBM 業務代表或您當地的 IBM 分公司來訂購出版品。

IBM 歡迎您提供意見。您可以利用線上 IBM WebSphere Commerce 文件讀者意見表來提出您的意見，URL 如下：

<http://www.ibm.com/software/commerce/pcf.html>

當您傳送資訊給 IBM 時，即授與 IBM 非獨占的資訊使用或公佈權利，IBM 不需對您負任何責任。

© Copyright International Business Machines Corporation 2002, 2003. All rights reserved.

目錄

開始之前	v
基本知識要求	v
本手冊中的所使用的慣例	vi
預設路徑	vi
變更摘要	vii
第 1 章 簡介	1
術語	1
概觀	2
使用 Web 服務的理由	2
Web 服務架構	3
必備需求	4
參照	4
第 2 章 WebSphere Commerce 的 Web 服務	5
WebSphere Commerce 中的傳訊基礎架構	5
認識 WebSphere Commerce 的 Web 服務實作	5
WebSphere Commerce 當作服務提供者	5
處理要求	6
回應	6
將 WebSphere Commerce 啓用作為服務提供者	6
WebSphere Commerce 當作服務要求者	7
第 3 章 將 WebSphere Commerce 啓用作為服務提供者	9
啓用 Web 服務的基礎架構	9
架構	9
安全注意事項	11
Web 服務和存取控制	11
傳輸層次安全	11
產生用戶端程式碼來存取 Web 服務	12
可供外部用戶端呼叫的 Web 服務範例	12
匯入範例程式碼	12
OrderCreate 服務 (Business Edition)	13
啓用 OrderCreate Web 服務範例	13
測試範例	16
OrderCreate Web 服務的安全	16
建立使用者和合約	16
將資料移入資料庫表格	17
新的訊息對映	19

配置訊息對映程式	21
OrderStatus 服務	21
啓用 OrderStatus Web 服務範例	21
OrderStatus 服務的存取控制	25
OrderStatus Web 服務的使用者職務	25
OrderStatus WSDL 定義	25
新的訊息對映	26
配置訊息對映程式	27
停用 OrderCreate 和 OrderStatus Web 服務	27

第 4 章 將 WebSphere Commerce 啓用作為服務要求者	31
架構	31
尋找服務實作	32
呼叫服務實作	33
WebSphere Commerce 可以在外部系統中呼叫的 Web 服務範例	33
OrderFulfillment 服務	33
介面 WSDL 定義	33
作業指令	34
商業物件	36
測試範例	38
停用 OrderFulfillment Web 服務	39

附錄 A. 範例 WSDL 定義檔	41
------------------------------------	-----------

附錄 B. 使用 IBM WebSphere Studio Application Developer	43
將 WSDL 匯入到 WebSphere Studio Application Developer 中	43
產生用戶端程式碼和資料類型	44
測試 Web 服務	44
使用 TCP 監視器來追蹤 SOAP 訊息	45
建立 TCP 監視器伺服器實例	45
配置 TCP 監視器伺服器實例	46
使用 TCP/IP 監視器來追蹤 SOAP 訊息	46
自訂用戶端程式碼	49

注意事項	53
商標	54

開始之前

*Web 服務實作手冊*的適用對象是要將 WebSphere® Commerce 5.5 版 Business Edition 以及 Professional Edition 中的商業程序外曝作為 Web 服務來執行的人員。本書同時也說明 WebSphere Commerce 如何存取由外部系統所代管的 Web 服務。本手冊可協助需要瞭解如何為 WebSphere Commerce 啟用及實作 Web 服務的程式開發人員。至於要配置 Web 服務以及使用 WebSphere Commerce 所提供的功能的其他使用者（例如系統管理者），都可以參閱本手冊。您必須先瞭解 WebSphere Commerce 程式設計模型。此外，您也需要瞭解如何呼叫和實作 Web 服務以及相關的通訊協定。

本手冊提供如何啟用 WebSphere Commerce 來使用 Web 服務和 Web 服務用戶端的相關資訊。本手冊也提供 Web 服務及其架構的概觀。本手冊簡要說明 WebSphere Commerce 中的傳訊基礎架構、為了啟用 Web 服務所提供的 WebSphere Commerce 執行時期支援，以及相關參照。本手冊亦說明所提供的 Web 服務範例，以及它們的相關商業物件與作業指令。

本手冊可分為下列章節：

第 1 章 簡介：Web 服務的簡要概觀、本手冊中所使用的術語的定義、必備需求和相關參照。

第 2 章 WebSphere Commerce 的 Web 服務： WebSphere Commerce 中的傳訊基礎架構以及 WebSphere Commerce 的 Web 服務架構的簡要概觀。同時說明將 WebSphere Commerce 商業程序外曝作為 Web 服務的方法，以及讓 WebSphere Commerce 存取由外部系統所代管的 Web 服務的方法。

第 3 章 啟用 WebSphere Commerce 作為 Web 服務提供者：說明如何透過範例定義，將 WebSphere Commerce 所定義的商業程序啟用作為 Web 服務。同時說明 WebSphere Commerce 新增的程式元素，以及 OrderCreate 和 OrderStatus Web 服務範例的實作明細。

第 4 章 啟用 WebSphere Commerce 作為 Web 服務要求者：說明如何透過範例定義，啟用 WebSphere Commerce 來呼叫外部系統所定義的服務。同時說明 WebSphere Commerce 新增的程式元素，以及 OrderFulfillment Web 服務範例的實作明細。

附錄 A. WSDL 定義檔範例：列出 Web 服務範例的 WSDL 定義檔。

附錄 B. 使用 IBM® WebSphere Application Developer 來建立 Web 服務用戶端程式碼：說明如何使用 IBM WebSphere Studio 來產生 Web 服務用戶端程式碼。

基本知識要求

為了使用本手冊中的資訊，您應該熟悉下列領域：

- Web 服務技術，例如 SOAP、WSDL 和 UDDI
- XML
- Java™
- JavaServer Pages 技術

- WebSphere Studio Application Developer (在本文件中亦稱為 WebSphere Studio)
- WebSphere Commerce 程式設計模型

本手冊中的所使用的慣例

本手冊使用下列慣例：


粗體字型 表示指令或圖形式使用者介面 (GUI) 控制項，如欄位名稱、按鈕或功能表選項。


固定寬度 表示您必須輸入與其完全相同的檔名及目錄路徑的文字範例。


斜體字型 用來強調及代表可換成您所要的值的變數。

i

 表示 WebSphere Commerce for AIX[®] 專用的資訊。

 表示 WebSphere Commerce for IBM @server iSeries[™] (舊稱為 AS/400[®]) 專用的資訊。

+
+
+
 表示 WebSphere Commerce for xSeries Linux 專用的資訊、WebSphere Commerce for @server zSeries[®] and S/390[®] Linux 專用的資訊，WebSphere Commerce for @server iSeries Linux 專用的資訊，以及 WebSphere Commerce for @server pSeries Linux 專用的資訊。


 表示 WebSphere Commerce for Solaris Operating Environment 軟體專用的資訊。

 表示 WebSphere Commerce for Windows[®] 專用的資訊。

 表示 DB2[®] Universal Database 專用的資訊。

 表示 Oracle[®] 資料庫專用的資訊。

 表示 WebSphere Commerce Business Edition 專用的資訊。

 指 WebSphere Commerce Studio 5.5 版 (Professional 或 Business Developer Edition)。

預設路徑

本手冊使用下列預設安裝路徑：

WC_installdir 是指 WebSphere Commerce 的安裝路徑。當您看到這個變數時，請將它替換成您安裝 WebSphere Commerce 的安裝路徑。例如，替換成

 C:\IBM\WebSphere\CommerceServer55

 /usr/WebSphere/CommerceServer55

 /opt/WebSphere/CommerceServer55

 /opt/WebSphere/CommerceServer55

 /QIBM/ProdData/CommerceServer55

WAS_installdir 是指 WebSphere Application Server 的安裝路徑。當您看到這個變數時，請將它替換成您安裝 WebSphere Application Server 的安裝路徑。例如，替換成

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

WCStudio_installdir 是指 WebSphere Commerce Studio 的安裝路徑。當您看到這個變數時，請將它替換成您安裝 WebSphere Commerce Studio 的安裝路徑。例如，替換成

▶ Studio C:\IBM\WebSphere\CommerceStudio55

WCStudio_workspacedir 是 WebSphere Commerce Studio 工作區的目錄。例如，替換成

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

WAS_userdir 是指 WebSphere Application Server 使用的所有資料中，使用者可以修改或需要使用者配置的目錄。當您看到這個變數時，請將它替換成

▶ 400 /QIBM/UserData/WebAS5/Base/*WAS_instance_name*

其中，*WAS_instance_name* 是代表與您的 WebSphere Commerce 實例相關的 WebSphere Application Server 名稱的變數。

變更摘要

- + 您可以從 WebSphere Commerce 網站的 Technical Library 網頁中，取得這份文件最新的版本（格式為 PDF），網址如下：
- + <http://www.ibm.com/software/commerce/library/>
- + 如果要瞭解產品的最新變更，請參閱現行的產品 README 檔，您也可以從 WebSphere Commerce 網站取得這個檔案。
- + 本書使用下列慣例作為修訂字元：
- + • 「+」（加號）字元表示此文件的現行修訂中所做的更新。
- + 本手冊的現行修訂中的更新是針對 Linux 所做的。

第 1 章 簡介

本章提供 Web 服務的概觀。本章定義本手冊中所使用的術語、列出必備需求，以及提供其他相關資訊來源的參照。

術語

本手冊使用下列術語：

服務提供者

服務提供者即為 Web 服務的擁有者。服務提供者會建立 Web 服務、將服務公佈到服務登錄或直接公佈到服務用戶端，以及代管提供服務存取權的平台。

服務要求者

服務要求者是指負責起始與 Web 服務之交談的應用程式。服務要求者的其中一個範例就是一般的網際網路瀏覽器。另一個範例則是使用 Web 服務，針對其處理基本需求來蒐集資訊的應用程式。服務要求者會使用已公佈的資訊來連結服務，並且會呼叫服務。

SOAP (簡易物件存取協定)

SOAP 會定義一個模型來使用以 XML 撰寫的簡易要求和回覆訊息，以作為電子通訊的基本通訊協定。這個傳輸機制是獨立於基礎平台和通訊協定之外。它可以促進公佈、尋找、連結和呼叫等作業。

UDDI (普及描述探索與整合)

UDDI 儲存庫儲存 Web 服務的說明。UDDI 是一種規格，其中定義可用 Web 服務的服務登錄，以作為廣域電子黃頁的集合。UDDI 可以讓公司將可用商品和服務的說明公佈到登錄中，將公司本身發表成為服務公司。

Web 服務

Web 服務是一種基於已公佈介面的軟體元件，以說明可利用標準化的 XML 傳訊來存取網路的作業集合。

WSDL (Web 服務說明語言)

WSDL 提供了一種方式來說明 Web 服務與應用程式設計介面 (API) 的特定介面。

XML 對映範本

XML 對映範本是用來處理入埠的 XML 訊息以及將它們轉譯成 WebSphere Commerce 指令參數。WebSphere Commerce 中的對映範本檔定義如何將送入的 XML 訊息的元素對映到目標指令參數。

有關對映範本的其他資訊，請參閱 WebSphere Commerce 正式作業與開發作業線上說明。

XML 訊息對映程式

WebSphere Commerce 中的 XML 訊息對映程式負責將送入的 XML 訊息轉換成目標指令所預期的 Java 物件。訊息對映程式可以使用 *instance_name.xml* 配置檔來登錄，並且專用訊息對映程式 ID 來識別。XML 訊息對映程式會使用對映範本檔中所定義的對映，將 XML 訊息轉換成對應的 Java 物件指令參數。

註: *instance_name* 是 WebSphere Commerce 實例的名稱。

有關訊息對映程式的其他資訊，請參閱 WebSphere Commerce 正式作業與開發作業線上說明中的 XML 訊息對映程式資訊。

概觀

Web 服務是一種新的 Web 應用程式。它們是自足式、自我說明的模組應用程式，可透過 Web 來公佈、尋找和呼叫。Web 服務會執行可呼叫的功能，包括簡單的要求，或者是複雜的商業程序。一旦部署和登錄 Web 服務，其他應用程式就可以探查及呼叫已部署的服務。Web 服務的基礎是許多標準，例如簡易物件存取協定 (SOAP)、Web 服務描述語言 (WSDL) 和普及描述探索與整合 (UDDI) 登錄。

WebSphere Commerce 5.5 版包含許多程式和文件，可讓您將 WebSphere Commerce 商業功能啓用作為 Web 服務。您可以啓用 WebSphere Commerce 的商業功能來作為可供外部系統存取的 Web 服務，讓它成為服務提供者。您也可以啓用 WebSphere Commerce 來呼叫外部系統所代管的 Web 服務，讓它成為服務要求者。

使用 Web 服務的理由

Web 服務可讓應用程式的整合更快速、更容易，成本也更低。整合是在通訊協定堆疊中的較高層次發生，它是根據輸入的訊息而定，比較著重於服務語意，而較少依賴網路通訊協定語意，以寬鬆地整合商業功能。這些特性最適合用來連接 Web 中的多個企業之間或單一企業內的商業功能。它們提供一個統一的程式設計模型，使企業內外的應用程式整合能夠以共同的方式來執行，充分利用共同的基礎架構。整合和運用 Web 服務，可以使用現有的語言和平台以及採用現有的應用程式，以遞增的方式來執行。

Web 服務是一種介面，用來說明可利用標準化的 XML 傳訊來存取網路的作業集合。使用標準、正式的 XML 表示法來說明它，稱為其服務說明。服務說明包括與服務互動所需要的全部明細，其中包括訊息格式（用來詳細說明作業）、傳輸通訊協定和位置。介面會隱藏服務的實作明細，讓服務可以在實作服務的硬體或軟體平台之外使用，同時也能在撰寫服務的程式設計語言之外使用。

WebSphere Commerce 的設計可讓 Web 服務同時作為 Web 服務的提供者以及作為其他 Web 服務的要求者（用戶端）。例如，外部 Web 服務用戶端可以連線到 WebSphere Commerce 並使用提供的 Web 服務（如圖 1 所示）。另外，WebSphere Commerce 可以連線到其他 Web 服務提供者並且要求資訊（如第 3 頁的圖 2 所示）。

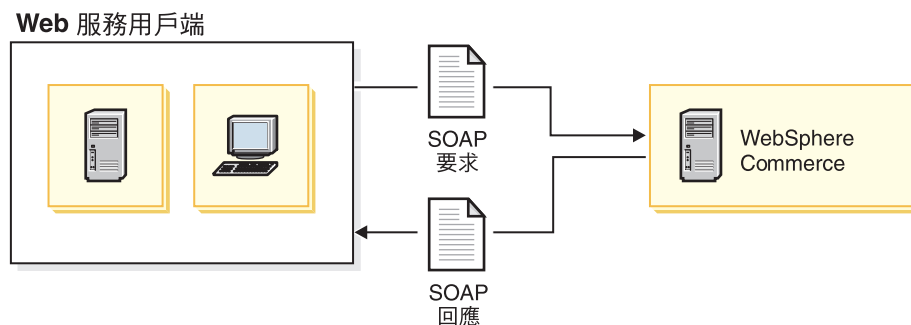


圖 1. WebSphere Commerce 作為服務提供者

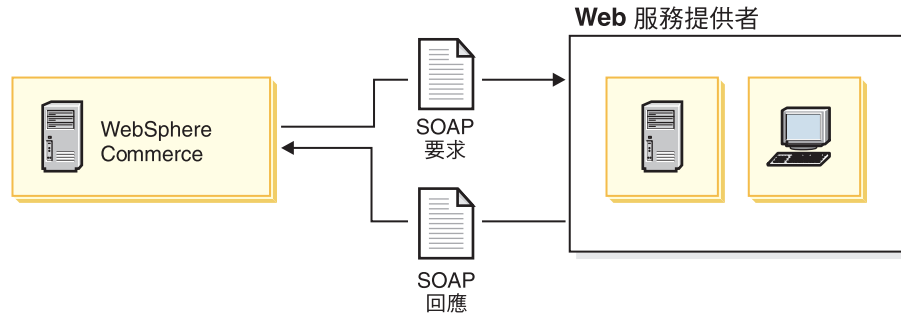


圖 2. WebSphere Commerce 作為服務要求者

Web 服務架構

Web 服務架構說明三種職務：服務提供者、服務要求者以及服務分配管理系統；以及三種基本作業：公佈、尋找和連結。網路元件可以扮演這三種職務中的任何一種或全部。服務提供者則會將 Web 服務公佈到服務分配管理系統。服務要求者會使用服務分配管理系統來尋找必要的 Web 服務，然後連結它們。

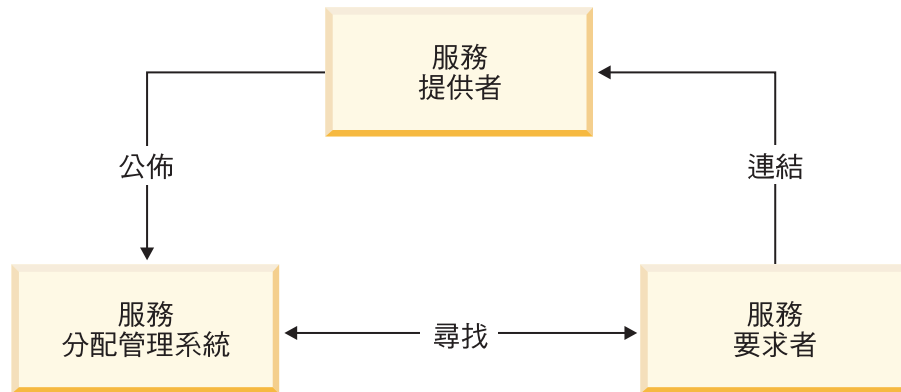


圖 3. Web 服務架構

在典型的實務內容中，服務提供者會代管可從網路存取的服務模組，也就是 Web 服務的實作。服務提供者則定義 Web 服務的服務說明，並將它公佈到服務要求者或服務登錄。服務要求者會使用尋找作業，從本端環境或服務分配管理系統所代管的服務登錄來擷取服務說明。服務要求者會使用服務說明來連結服務提供者，並且呼叫 Web 服務實作或與之互動。

這個解決方案提供執行時期支援，讓 WebSphere Commerce 所定義的商業程序可以當作 Web 服務。有關如何使用現有的 WebSphere Commerce 指令來作為 Web 服務的範例，請參閱第 9 頁的第 3 章，『將 WebSphere Commerce 啓用作為服務提供者』。有關 WebSphere Commerce 如何呼叫外部系統所定義的 Web 服務的範例，請參閱第 31 頁的第 4 章，『將 WebSphere Commerce 啓用作為服務要求者』。

必備需求

在使用本手冊所提供的資訊之前，請確定您已經安裝 IBM WebSphere Commerce 5.5 版。在安裝 WebSphere Commerce 5.5 版時，您必須選取自訂安裝選項，來安裝所提供的 Web 服務範例。系統會要求您選取想要安裝的元件。請選取 **WebSphere Commerce 範例檔案**。如需其他資訊，請參閱 *WebSphere Commerce 安裝手冊*。

另外，如果您使用 WebSphere Commerce Studio 來啓用 Web 服務範例，請確定您已經安裝好 WebSphere Commerce Studio 5.5 版。如需其他資訊，請參閱 *WebSphere Commerce Studio 安裝手冊*。

如果您想要修改 Web 服務範例，請確定您是以 WebSphere Commerce Studio 來作為開發環境。

參照

有關 Web 服務及相關技術的其他資訊，請造訪下列網站：

- 有關 Apache SOAP (SOAP 提交 W3C 的實作) 的資訊，請造訪 <http://ws.apache.org/soap/>
- 有關 SOAP 規格的深入資訊，請造訪 <http://www.w3.org/TR/SOAP/>
- 有關 UDDI 的資訊，請造訪 <http://www.uddi.org/>
- 有關 WSDL 的資訊，請造訪 <http://www.uddi.org/specification.html>
- 如果需要進一步瞭解 IBM Alphaworks 的 Web 服務，請造訪 <http://www.alphaworks.ibm.com/webservices>
- 如果要從程式開發人員的角度來進一步了解 Web 服務，請造訪 <http://www.ibm.com/developerworks/webservices>
- 如需 IBM WebSphere Commerce Business Edition 軟體的完整資訊，請造訪 http://www.ibm.com/software/webservers/commerce/wc_be/
- 如需 IBM WebSphere Commerce Professional Edition 軟體的完整資訊，請造訪 http://www.ibm.com/software/webservers/commerce/wc_pe/
- 有關支援事業夥伴來開發更優秀的 WebSphere Web 服務解決方案的資訊，請造訪 <http://www.ibm.com/websphere/wow>
- 有關 IBM WebSphere Studio 的資訊，請參閱 www.ibm.com/software/webservers/commerce/commercestudio/

註：IBM 可隨時變更前述的網址，而不另行通知。IBM 並不負責從非 IBM 網站所取得資訊的真實性與正確性。

第 2 章 WebSphere Commerce 的 Web 服務

本章簡要說明 WebSphere Commerce 中的傳訊基礎架構，並協助您瞭解 WebSphere Commerce 的 Web 服務架構。本章也說明將 WebSphere Commerce 商業程序外曝作為 Web 服務的方法，以及 WebSphere Commerce 存取外部系統所代管的 Web 服務的方法。

WebSphere Commerce 中的傳訊基礎架構

WebSphere Commerce 執行時期架構包括通訊協定接收程式、配接器、控制程式、指令和檢視畫面。WebSphere Commerce 中的傳訊基礎架構會使用這些元件。

通訊協定接收程式是一個 WebSphere Commerce 執行時期元件，可接收傳輸工具送來的入埠要求，再根據所用的通訊協定，將要求分派給適當的配接器。

在 WebSphere Commerce 5.5 版中，RequestServlet 是當作負責接收 SOAP 1.1 版訊息的通訊協定接收程式。當 RequestServlet 收到 XML 要求時，它會將要求傳送到配接器管理程式。然後配接器管理程式會查詢配接器類型，以判斷哪一個配接器可以處理要求。一旦決定特定的配接器之後，就會將要求傳送至該配接器。

透過 HTTP 通訊協定接收 XML 訊息時，會呼叫程式配接器。它能让外部系統透過 HTTP 通訊協定來傳送 XML 要求，以便與 WebSphere Commerce 通訊。程式配接器為外部系統（例如採購系統、供應商系統和其他系統）提供一種共通的方式，來透過 HTTP 通訊協定與 WebSphere Commerce 通訊。透過這種方式，WebSphere Commerce 就可以和外部系統緊密地整合。

有關要求流程的其他資訊，請參閱 *WebSphere Commerce 程式設計手冊與指導教學*。

認識 WebSphere Commerce 的 Web 服務實作

WebSphere Commerce 程式設計模型是由 URL 指令、控制程式指令、作業指令、檢視指令、程式配接器、Web 控制程式等組成。有關 WebSphere Commerce 程式設計模型的詳細資料，請參閱 *WebSphere Commerce 程式設計手冊與指導教學*。

下列各節說明 WebSphere Commerce 如何扮演服務提供者以及服務要求者的職務：

WebSphere Commerce 當作服務提供者

當您將 WebSphere Commerce 中的商業程序當作可供外部系統存取的 Web 服務時，WebSphere Commerce 會成為服務提供者。WebSphere Commerce 5.5 版包含傳訊系統的加強功能，以支援 SOAP 訊息和回應。此外，其中的範例將告訴您如何將現有的 WebSphere Commerce 商業邏輯外曝作為 Web 服務。

用戶端可以利用 HTTP 或安全 HTTP 通訊協定，透過網路來呼叫 URL 指令。WebSphere Commerce 也能透過 HTTP 和 HTTPS 通訊協定或傳訊中介軟體來接收 XML 訊息。這項功能已經進一步擴充成可容許 WebSphere Commerce 透過 HTTP 來接收 SOAP 訊息，而且這項功能可用來啟用 Web 服務。RequestServlet 是所有的 URL 要

求對 WebSphere Commerce 系統的進入點。當 RequestServlet 收到 XML 訊息時，它會將內容類型識別為 XML，並呼叫程式配接器。

處理要求

程式配接器可支援呼叫 WebSphere Commerce 指令的遠端系統。它會接收要求並使用訊息對映程式，將要求轉換成適當的指令輸入。在轉換之後，程式配接器會執行指令。

在 WebSphere Commerce 5.5 版中，WebSphere Commerce 執行時期架構新增了一個新的訊息對映程式和範本對映檔，以協助剖析和處理 SOAP XML 訊息。

這個新的範本對映檔包含送入的 SOAP XML 訊息以及指令參數之間的對映。新的訊息對映程式會檢視這個範本對映檔，並將 SOAP XML 訊息剖析成適當的指令輸入，然後將輸入傳回程式配接器。然後程式配接器會呼叫 Web 控制程式，而 Web 控制程式會執行訊息對映程式所指定的指令。

回應

在 WebSphere Commerce 中，當控制程式指令順利執行時，會傳回要執行的檢視畫面名稱。Web 控制程式會呼叫對應檢視畫面的檢視指令，並撰寫要傳回的訊息。組成回應檢視畫面的方法有許多種。其中包括重新導向至不同的 URL、轉遞至 JSP 範本或將 HTML 文件寫入回應中。

為了替不同類型的用戶端裝置產生訊息，WebSphere Commerce 支援許多不同的裝置格式。Web 控制程式會挑選與送入訊息的裝置格式相對應的檢視畫面名稱。在 WebSphere Commerce 5.5 版中，SOAP XML 訊息加入了一個新的裝置格式 SOAPHTTP。您可以為相同的檢視畫面名稱定義一組新的 JSP 範本，而且是使用新的裝置格式。這些 JSP 範本會撰寫要傳回 SOAP 用戶端的 SOAP 回應。回應會遵循「模型檢視畫面控制程式」型樣，而且會將檢視畫面與商業邏輯分開。

另外還有一個新的檢視畫面指令實作，可用來處理 SOAP 回應。這個檢視畫面指令實作的介面名稱為 `com.ibm.commerce.me.soap.commands.SOAPViewCommand`，而實作類別名稱是 `com.ibm.commerce.me.soap.commands.HttpSOAPViewCommandImpl`。因此，每次在 VIEWREG 表格中登錄新的 JSP 範本來撰寫 SOAP 回應訊息時，應該使用這個介面名稱和類別名稱。

新的 SOAP 檢視指令實作可以同時處理 HTTP 重新導向和 HTTP 轉遞要求。通常，HTTP 重新導向是由 URL 重新導向來處理。由於 SOAP 用戶端無法瞭解 URL 重新導向，這個 SOAP 檢視指令也會將 HTTP 重新導向當作 HTTP 轉遞來實作。

將 WebSphere Commerce 啓用作為服務提供者

以下是讓您將 WebSphere Commerce 商業程序當作 Web 服務的步驟概觀：

1. 指出您要外曝作為 Web 服務的商業邏輯。
2. 指出實作這個商業邏輯的 URL 指令。如果沒有指令，則建立一個新的 URL 指令。您可以撰寫新的指令來呼叫現有指令的集合。
3. 指出此 URL 指令所需要的必要及選用參數。
4. 定義 Web 服務的 WSDL 定義。
5. 使用對映範本檔將送入的 SOAP 要求對映到指令參數。確定 TemplateTag 包含 WSDL 定義中的所有元素的對映。在從用戶端傳送要求時，您需要移入 SOAP 要求中的所有元素的資料。對映範本檔定義如何利用 XPATH 對映將送入的 SOAP XML

訊息中的每一個元素對映到指令參數。您可以使用類似「TCP 監視器」這類的公用程式來截取 SOAP 要求，然後使用截取的訊息來定義對映。

註：「TCP 監視器」是隨附在 WebSphere Studio Application Developer 5.0 中，請參閱 WebSphere Studio 說明，以取得其他資訊。Apache AXIS 發行版也隨附一個獨立式的「TCP 監視器」版本。請造訪 <http://ws.apache.org/axis/> 以取得明細。

6. 撰寫一個 JSP 範本來撰寫回應。
7. 部署服務。這個步驟包括部署相關檔案及資源。如果是新的指令，您必須部署指令並且將它登錄到 WebSphere Commerce 指令登錄中。

WebSphere Commerce 中的商業邏輯是利用控制程式指令來實作。控制程式指令是指可以和 Web 控制程式直接互動的指令。URL 指令是指可利用 URL 來呼叫的控制程式指令。這樣您就可以在 Web 瀏覽器中輸入 URL 來呼叫 URL 指令。在實作 Web 服務之前，請指出用來實作商業邏輯（要外曝作為 Web 服務）的 URL 指令。例如，OrderStatus 指令可讓您更新 WebSphere Commerce 中的現有訂單的狀態。

有關 URL 指令的其他資訊，請參閱 WebSphere Commerce 正式作業與開發作業線上說明中的「參照」一節。

WebSphere Commerce 當作服務要求者

當您使用 WebSphere Commerce 來起始由外部系統所代管的 Web 服務時，WebSphere Commerce 就會成為服務要求者。WebSphere Commerce 5.5 版包含許多範例，以說明如何利用 Web 服務呼叫，將 WebSphere Commerce 商業邏輯和其他系統整合起來。WebSphere Commerce 會透過 Web 服務 Proxy 來傳送 Web 服務要求，而 Web 服務 Proxy 會傳送和接收 SOAP 型的要求和回應。商業邏輯會根據已識別的 Web 服務連結介面來運作。而此處提供的範例可支援基於這些介面的 Web 服務呼叫。

您可以將實作公佈到服務登錄中，或儲存在本端檔案系統中。從 WebSphere Commerce 呼叫 Web 服務的步驟包括下列活動：

1. 識別已公佈的 Web 服務介面定義，它符合 WebSphere Commerce 所支援的現有商業實務內容。
2. 將服務實作的 WSDL 定義下載到您的本端 WebSphere Commerce 機器中。
3. 將 Web 服務定義檔登錄到 *instance_name.xml* 配置檔（其中 *instance_name* 是您的 WebSphere Commerce Server 實例的名稱）。
4. 建立 WebSphere Commerce 可以使用的用戶端 Proxy。您可以利用 IBM WebSphere Studio 這類工具來產生它們。當您產生 Web 服務介面定義的用戶端 Proxy 時，也會同時產生輸入訊息和輸出訊息所需要的資料類型。請參閱本文件第 12 頁的『產生用戶端程式碼來存取 Web 服務』一節，以取得其他資訊。

您也可以使用其他工具來產生用戶端 Proxy，例如，「Web 服務工具集」。請造訪 <http://www.alphaworks.ibm.com/webservices>。

5. 撰寫一個作業指令，實作邏輯以執行下列步驟：
 - a. 擷取存取點。
 - b. 呼叫 Web 服務的 Proxy。
 - c. 處理從 Web 服務傳回的結果。
6. 更新 WebSphere Commerce 指令登錄來登錄作業指令。

第 3 章 將 WebSphere Commerce 啓用作為服務提供者

本章說明如何將 WebSphere Commerce 所定義的商業程序啓用作為 Web 服務。在這個實務內容中，WebSphere Commerce 是外部系統的服務提供者。同時也提供範例，來示範如何啓用 OrderCreate 和 OrderStatus 商業程序作為 Web 服務。

啓用 Web 服務的基礎架構

以下是支援送入的 Web 服務要求的 WebSphere Commerce 執行時期元素：

- SOAP 訊息對映程式
- Web 服務對映範本
- SOAP 回應檢視指令

外部使用者可以使用用戶端程式來呼叫 WebSphere Commerce 服務。這些使用者可以從任何能夠傳送及接收 SOAP 訊息的系統來進行連線。WebSphere Commerce 5.5 版可透過 HTTP 傳輸通訊協定來支援 SOAP 訊息。

WebSphere Commerce 服務應該當作 WSDL 定義來公佈。Web 服務用戶端可以透過 HTTP，利用 SOAP 來下載 WSDL 定義、產生用戶端，以及呼叫 WebSphere Commerce Web 服務。

架構

這個架構中的訊息流程是同步的。這表示外部系統會先等待 WebSphere Commerce 系統回應，然後才繼續執行進一步的處理。入埠傳訊系統則是用來讓外部用戶端呼叫已公佈作為 Web 服務的 WebSphere Commerce 商業程序。

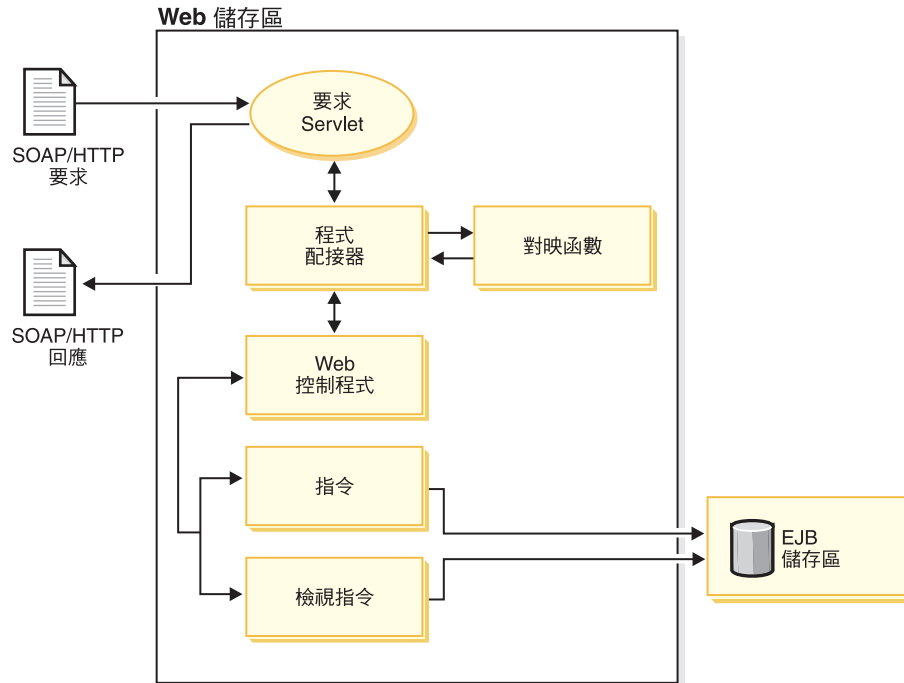


圖 4. 將 WebSphere Commerce 啓用作為服務提供者的架構

下列清單說明當外部系統透過 Web 服務來呼叫 WebSphere Commerce 商業程序時的互動：

1. 外部系統傳送格式正確的 SOAP 要求給 WebSphere Commerce。
2. RequestServlet 收到要求，並將要求委派給程式配接器。
3. 程式配接器呼叫已登錄的 SOAP 訊息對映程式。
4. SOAP 訊息對映程式剖析 SOAP 訊息，並使用範本檔將名稱/值配對傳回給程式配接器，這個範本檔包含 Web 服務要求以及指令參數中的參數之間的對映資訊。
5. 程式配接器呼叫 Web 控制程式，並將指令名稱和名稱/值配對傳送給 Web 控制程式。
6. Web 控制程式執行指令。在提供的範例中，會呼叫 BatchOrderRequest 和 OrderStatusCmd 指令。
7. 執行指令後，會將檢視畫面名稱傳回給 Web 控制程式。
8. Web 控制程式執行已登錄的檢視指令，以組成 SOAP 回應，並將它傳回給程式配接器。
9. HTTP 程式配接器將訊息傳回給 RequestServlet。
10. RequestServlet 將回應傳送給等待中的外部用戶端。

以下是為了支援服務提供者所提供的 Web 服務所新增的範例：

- 提供 webservice_template.xml 檔，以將 XPath 對映到指令內容。
- 提供下列範例 JSP 範本以撰寫 SOAP 回應和錯誤訊息：
 - SOAP_OrderStatusResponse.jsp
 - SOAP_PurchaseOrderResponse.jsp
 - SOAP_GenericApplicationError.jsp
 - SOAP_GenericSystemError.jsp

安全注意事項

下列幾節說明在啓用 Web 服務時所涉及的安全注意事項。

Web 服務和存取控制

存取控制是指將 WebSphere Commerce 系統中的受保護資源限定為僅有已授權的個人和組織可存取的程序。爲了協助進行資料庫管理以及確保安全，WebSphere Commerce 包含一個存取控制機制，可將資源的存取權限定在特定的使用者。存取控制可以定義成許多安全準則，包括：

1. 容許或拒絕系統使用者存取系統所管理的資源。
2. 指定使用者對每一個資源可執行的動作。

WebSphere Commerce 的存取控制模型是根據執行的存取控制原則而定。如果要進一步了解存取控制和原則，請參閱 *WebSphere Commerce Security Guide*。

將 WebSphere Commerce 當作服務提供者時，它可以容許外部 Web 服務用戶端呼叫由控制程式指令所定義的商業程序。結果，Web 服務會繼承 Web 服務用戶端呼叫的指令所適用的存取控制原則。因此，您需要針對與 WebSphere Commerce 一起使用的每一個 WSDL 定義，分別定義認證資訊。除了傳入傳送鑑別使用者所需要的認證外，Web 服務用戶端必須確定使用者有執行指令所需要的權限。

認證的 WSDL 定義中的 XML 綱目格式必須與下面的格式類似：

```
<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>
```

認證資訊必須對映到對映範本檔中可用的範本定義的下列欄位：

- 使用者登入的欄位名稱是 'logonId'，而 FieldInfo 是 'CONTROL'。
- 密碼的欄位名稱是 'logonPassword'，而 FieldInfo 是 'CONTROL'。

傳輸層次安全

如果要啓用傳輸層次安全，SOAP 訊息可以透過 HTTPS 來傳送。強烈建議服務提供者使用 HTTPS 來傳送 SOAP 訊息；否則任何想截取訊息的人都可以看到傳送的認證。

註：如果要啓用 HTTPS，SOAP 用戶端必須使用 URL 以及 HTTPS 作為通訊協定名稱。在從用戶端發出呼叫之前，請設定內容來啓用 Secure Socket Layer (SSL)。如果您在 Web 服務用戶端中使用 IBM Developer Kit，Java Technology Edition 1.2 版或之後的版本，則啓用 HTTPS 的程式碼與下面類似：

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

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

如果您使用 WebSphere Commerce Studio 來開發 Web 服務用戶端，就會使用前述的程式碼。

如果您使用 Sun Microsystems 的 J2SE SDK (Java 2 Standard Edition 軟體開發工具集)，則用來透過 HTTPS 呼叫 Web 服務的用戶端程式碼與下面類似：

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







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

如果您在嘗試呼叫 WebSphere Commerce 所定義的入埠服務時，在用戶端收到「訊息交換失敗」、「不明的 CA」或「伺服器拒絕連線異常狀況」，就表示用戶端無法辨識伺服器所提出的認證。如果是這種情況，而您信任伺服器，就可以將伺服器憑證新增至您的授信憑證清單中。

產生用戶端程式碼來存取 Web 服務

本節說明如何產生用戶端程式碼。

Web 服務範例是使用 WSDL 定義檔來公佈，並且儲存在下列目錄中：

-  `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`

您可以使用 WSDL 定義來開發如 Web 服務要求者之類的用戶端應用程式。您可以使用適當的工具，從 Web 服務的 WSDL 定義來產生為了存取 WebSphere Commerce 實作的 Web 服務時，所需要的用戶端程式碼和資料類型。

IBM WebSphere Studio Application Developer (在本文件中又稱為 WebSphere Studio) 是其中一個可讓您用來從 WSDL 定義產生用戶端和 Proxy 程式碼的工具。如需其他資訊，請參閱第 43 頁的『附錄 B. 使用 IBM WebSphere Studio Application Developer』。

您可以用來產生用戶端 Proxy 的另一個工具是 Web 服務工具集。請造訪 <http://www.alphaworks.ibm.com/webservices>。

可供外部用戶端呼叫的 Web 服務範例

本節說明如何啓用以下的 Web 服務範例：

1.  OrderCreate 服務
2. OrderStatus 服務

您可以在 WebSphere Commerce 機器或已安裝 WebSphere Commerce Studio 的開發機器上啓用 Web 服務範例。如果您要啓用 WebSphere Commerce Studio 機器上的 Web 服務範例，請遵循本節中的指示，並參閱標示  的目錄路徑。

匯入範例程式碼

如果您要編輯或修改 Web 服務範例，就必須將範例原始碼匯入到 WebSphere Commerce Studio 中。如果要匯入這個程式碼，請執行下列步驟：

1. 開啓 WebSphere Commerce 開發環境，方法是選取開始 > 程式集 > **IBM WebSphere Commerce Studio** > **WebSphere Commerce 開發環境**。
2. 切換到 J2EE 視景並選取「J2EE 導覽器」檢視畫面。
3. 展開 **WebSphereCommerceServerExtensionsLogic** 專案。用滑鼠右鍵按一下 **src** 目錄，然後選取匯入。這時會開啓「匯入精靈」。
4. 選取**檔案系統**，然後按一下**下一步**。
5. 按一下**瀏覽**並尋找下列目錄：
`WCStudio_install_dir\Commerce\samples\webservices\com`
6. 按一下**全選**，然後按一下**完成**。
7. 用滑鼠右鍵按一下 **WebSphereCommerceServerExtensionsLogic** 專案，然後選取**內容**。
8. 按一下 **Java 建置路徑**，然後移至**程式庫標籤**。
9. 按一下**新增變數**。選取 SOAPJAR 和 XERCES 變數，然後按一下**確定**。
10. 按一下**確定**。
11. 用滑鼠右鍵按一下 **WebSphereCommerceServerExtensionsLogic** 專案，然後選取**重新建置專案**。

您現在已經將範例原始程式碼匯入 WebSphere Commerce Studio 中。有關如何在 WebSphere 測試環境中啓動 WebSphereCommerceServer 伺服器的其他資訊，請參閱 *WebSphere Commerce Studio 安裝手冊*。

OrderCreate 服務 (Business Edition)


OrderCreate Web 服務適用於 WebSphere Commerce Business Edition 以及 WebSphere Commerce Business Developer Edition。

本節示範如何在 WebSphere Commerce 中將商業邏輯啓用作爲 Web 服務，以接收來自外部應用程式（例如採購應用程式）的訂單。採購系統可以使用 WebSphere Commerce BatchOrderRequest 指令，在 WebSphere Commerce 中建立訂單。這個範例 OrderCreate Web 服務實作說明如何建立一個 Web 服務來接收自買方組織送出的訂單。如果要使用 OrderCreate Web 服務範例，您需要在 WebSphere Commerce 系統中擁有一家已經公佈的商店。





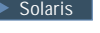

通常，賣方必須先知道買方組織，才能在 WebSphere Commerce 中建立訂單。因此，買方組織必須先登錄到 WebSphere Commerce 中，買方才能呼叫 OrderCreate Web 服務。

啓用 OrderCreate Web 服務範例

如果要啓用 OrderCreate Web 服務範例，請執行下列步驟：

1. 指出您要外曝作爲 Web 服務的商業邏輯。在此情況下，讓買方組織使用與採購系統類似的應用程式來建立一份訂購單。
2. 指出實作這個商業邏輯的指令。在此情況下，指令是 BatchOrderRequest。
3. 指出 BatchOrderRequest 指令所需要的必要及選用參數。您可以在 WebSphere Commerce 正式作業與開發作業線上說明中取得 BatchOrderRequest 指令的說明。有關參數的完整清單資訊，請參閱下列目錄中所提供的 `webservice_template.xml` 檔案。
 -  `WC_install_dir\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. 建立 OrderCreate Web 服務的 WSDL 定義。在下列目錄中有一個範例 OrderCreate.wsd1 定義檔：

-  `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`

您可以修改這個 WSDL 定義，以符合您要提供的服務的定義需求。







為 OrderCreate Web 服務所公佈的 WSDL 定義包含了呼叫此 Web 服務所需的存取點、連結範本以及資料類型綱目。

5. 使用對映範本檔將送入的 SOAP 要求對映到指令參數。在下面的目錄中有一個範例對映範本：







-  `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`

您可以自訂這個檔案。如果您已經修改範例 WSDL 定義，則您需要修改這個範例對映範本，以符合 WSDL 定義中的變更。

6. 如果要啟用這個範本（其中包括 OrderCreate Web 服務範例），請執行下列步驟：
- a. 將 webservice_template.xml 檔複製到下列目錄中：

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







- b. 如果 `webservice_template.xml` 檔已經存在，而且不含任何資訊，就會改寫它。如果檔案包含資訊，則將與 `OrderCreate` 服務對應的 `TemplateDocument` 資訊複製到現有的檔案中。
7. 啓用 "SOAPINTEGRATION" 的訊息對映程式。如果要這麼做，請開啓位於下列目錄中的 `instance_name.xml` 配置檔：

-  `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`



找到名稱爲 "WCS.SOAPINTEGRATION" 的訊息對映程式。如果 "enable" 屬性的值是 "false"，就將它設定爲 "true"。

8. 撰寫 JSP 範本來撰寫回應。在這個實務內容中，請複製下列的範例 JSP 範本：
- `SOAP_PurchaseOrderResponse.jsp` - 這個範本會撰寫 `OrderCreate` 要求的 SOAP 回應訊息。
 - `SOAP_GenericApplicationError.jsp` - 這個範本會在發生應用程式錯誤時撰寫 SOAP 錯誤回應。
 - `SOAP_GenericSystemError.jsp` - 這個範本會在發生系統錯誤時撰寫 SOAP 錯誤回應。

請從下面的目錄來進行複製：

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

複製到下面的目錄中：

-  `WAS_installdir\installedApps\cell_name\
WC_instance_name.ear\Stores.war`
其中，`cell_name` 是您已經安裝 WebSphere Application Server 的機器名稱。
-  `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. 建立一個買方組織，並且在買方和供應商組織之間建立一份合約。如需其他資訊，請參閱本文件中的第 16 頁的『建立使用者和合約』。
10. 按照本文件中的第 17 頁的『將資料移入資料庫表格』的說明，將資料移入資料庫表格中。

11. 如果您使用 WebSphere Commerce，請重新啟動 Commerce Server 實例，使變更生效。如果您使用 WebSphere Commerce Studio 機器，則重新啟動 WebSphere 測試環境中的 WebSphereCommerceServer 伺服器。

測試範例

您可以從 WSDL 定義來產生 Web 服務用戶端。用戶端可以藉由傳送 PurchaseOrderRequest 訊息來呼叫這個 Web 服務。如果順利建立訂單，則回應訊息會包含一個狀態碼 (200)、一個「確定」狀態訊息、狀態文字，以及在 WebSphere Commerce 中建立的 OrderID。如果發生錯誤，您會收到適當的錯誤訊息。

OrderCreate Web 服務的安全

在本範例中外曝為 Web 服務的商業邏輯需要鑑別發出要求的買方組織。透過 WebSphere Commerce 存取控制模型，指令及其相關 Web 服務的鑑別可藉由提供要求者的 WebSphere Commerce 使用者 ID 和密碼來完成。這時必須將使用者登錄為 WebSphere Commerce 使用者。

對想要使用範例 OrderCreate Web 服務的使用者進行鑑別時，必須將使用者所屬的買方組織登錄到 WebSphere Commerce 中。必須先行驗證傳送的認證，然後才處理 OrderCreate 服務。

OrderCreate Web 服務的認證包括 WebSphere Commerce 系統中的供應商和買方的組織代碼與組織代碼類型。用戶端必須傳送與 ORGCODE 表格中所登錄的組織代碼和組織代碼類型相同的值。

以下是認證的 WSDL 定義中的綱目，其中 identity 是組織代碼，domain 是組織代碼類型，而共用秘密是指買方和供應商組織所共用的密碼：

```
<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>
```

註： 共用秘密僅適用於傳送者認證。

存取控制

當 WebSphere Commerce 收到 OrderCreate 訊息時，會呼叫 BatchOrderRequest 指令來處理送入的要求。如果要使用 OrderCreate Web 服務，使用 WebSphere Commerce 的供應商與擔任 Web 服務用戶端的買方之間必須有合約存在。

建立使用者和合約

如果買方組織不存在，請先完成下列步驟，然後再呼叫 OrderCreate Web 服務。請使用下列步驟來建立新的買方組織、指派探購買方的職務給買方組織，以及在賣方和買方組織之間建立一份合約：

1. 使用「管理主控台」來建立買方商業組織。相關明細請參閱 WebSphere Commerce 正式作業與開發作業線上說明。
2. 指定下列職務給這個買方組織：
 - a. 探購買方
 - b. 探購買方管理者

- c. 採購經理
3. 從「管理主控台」為這個買方組織建立一個使用者。這個使用者的使用者 ID 必須是買方組織的群組使用者 ID，而密碼必須為買方組織的共用秘密。群組使用者 ID 和共用秘密是第 11 頁的『安全注意事項』所提及的買方組織認證的一部分。
4. 指定下列職務給步驟 3 中所建立的使用者：
 - a. 採購買方
 - b. 採購買方管理者
5. 使用 WebSphere Commerce Accelerator 為這個買方組織建立一個商業帳戶。有關如何建立商業帳戶的指示，請參閱 WebSphere Commerce 正式作業與開發作業線上說明。在建立商業帳戶時，請注意下列事項：
 - 請勿選取「客戶」頁面中的容許客戶在遵循商店預設合約的條款時進行購買勾選框。
 - 請勿選取「訂購單」頁面中的可以在訂購時指定訂購單號碼勾選框。
 - 啟用信用額度付款方式並且在「信用額度」頁面中指定信用額度帳戶號碼。
6. 為建立的商業帳戶建立一個合約。有關如何為商業帳戶建立合約的指示，請參閱 WebSphere Commerce 正式作業與開發作業線上說明。在建立合約時，請注意下列事項：
 - 按一下**一般**標籤。輸入「合約名稱」以及「說明」，然後選取您要合約作用的期間。
 - 按一下**客戶**標籤。選取您之前建立的買方組織。
 - 按一下**付款**標籤。確定**信用額度**用法已經勾選。
7. 按一下**提交**來啟用合約。







前述的步驟會為 WebSphere Commerce 中的買方組織建立一個帳戶，並且在供應商 (WebSphere Commerce) 組織與買方組織之間建立一份合約。然後將必要的存取許可權指定給買方組織。

將資料移入資料庫表格

如果要執行 BatchOrderRequest 指令，您必須執行資料庫 Script 來將資料移入特定的表格。

如果要更新您的資料庫，請執行下列步驟：

1. 導覽至下列目錄：

-  `WC_installdir\samples\webservices\bin`
-  `WCStudio_installdir\Commerce\samples\webservices\bin`
-     `WC_installdir/samples/webservices/bin`

2. 按照下列表格中的說明，找到您的配置所屬的正確 Script 檔案集：

作業系統	資料庫類型	Script 檔案名稱
------	-------	-------------

Windows Studio	DB2	Webservices_DBUpdate.db2.bat Webservices_UndoDBUpdate.db2.bat
	Oracle	Webservices_DBUpdate.oracle.bat Webservices_UndoDBUpdate.oracle.bat
AIX Solaris	DB2	Webservices_DBUpdate.db2.sh Webservices_UndoDBUpdate.db2.sh
	Oracle	Webservices_DBUpdate.oracle.sh Webservices_UndoDBUpdate.oracle.sh
400 Linux	DB2	Webservices_DBUpdate.db2.sh Webservices_UndoDBUpdate.db2.sh

+
+
+
+
+

3. 將適當的 Script 檔複製到下列目錄中：

- Windows `WC_installdir\samples\webservices\bin`
- Studio `WCStudio_installdir\Commerce\samples\webservices\bin`
- AIX Linux Solaris `WC_installdir/samples/webservices/bin`
- 400 `/QIBM/ProdData/CommerceServer55/samples/webservices/bin`

4. DB2 從下列目錄中找到 `wcs.webservices.enable.sql` 和 `wcs.webservices.disable.sql` Script：

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

複製到下面的目錄中：

- Windows `WC_installdir\schema\db2`
- Studio `WCStudio_installdir\Commerce\schema\db2`
- AIX Linux Solaris `WC_installdir/schema/db2`
- 400 `/QIBM/UserData/CommerceServer55/WebSphere_Commerce_instance_name/schema`

Oracle 從下列目錄中找到 `wcs.webservices.enable.sql` 和 `wcs.webservices.disable.sql` Script：

- Windows `WC_installdir\samples\webservices\schema\oracle`
- Studio `WCStudio_installdir\Commerce\samples\webservices\ schema\oracle`

- ▶ AIX ▶ Solaris `WC_installdir/samples/webservices/schema.oracle`

然後複製到下面的目錄中：

- ▶ Windows `WC_installdir\schema\oracle`
- ▶ Studio `WCStudio_installdir\Commerce\schema\oracle`
- ▶ AIX ▶ Solaris `WC_installdir/schema/oracle`

- 修改您在步驟 4 中複製的 `wcs.webservices.enable.sql` 和 `wcs.webservices.disable.sql` SQL Script 中所使用的下列預設值，以符合呼叫 OrderCreate Web 服務的買方組織的明細：
 - `WebServicesUserId` - 這是買方組織的組織代碼。這個代碼和組織代碼類型專用來識別買方組織。
 - `Webservices@ibm.com` - 這是買方組織的組織代碼類型。
 - `aurora` - 這是供應商組織的組織代碼。通常，這是組織的 D-U-N-S 號碼，或是組織任何其他的標準唯一識別碼。（D-U-N-S 號碼是一個包含九個位數的數字，這是業界的一種廣域標準，用以識別和追蹤企業）。
 - `DUNS` - 這是供應商組織的組織代碼類型。如果您使用組織代碼的 DUNS 號碼，請使用 "DUNS"。
- 在升級 SQL Script 之後，請執行適當的指令 Script 檔：

▶ DB2 如果您使用 DB2 資料庫：

- ▶ Windows ▶ Studio 從 DB2 指令視窗中，執行 `Webservices_DBUpdate.db2.bat`
- ▶ AIX ▶ Linux ▶ Solaris 從指令視窗中，執行 `Webservices_DBUpdate.db2.sh`
- ▶ 400 啟動一個 QSH 階段作業。執行指令 `/QIBM/ProdData/CommerceServer55/samples/webservices/bin/ Webservices_DBUpdate.db2.sh`

▶ Oracle 如果您使用 Oracle 資料庫：

- ▶ Windows ▶ Studio 從指令視窗中，執行 `Webservices_DBUpdate.oracle.bat`
- ▶ AIX ▶ Solaris 從指令視窗中，執行 `Webservices_DBUpdate.oracle.sh`

驗證 `WC_installdir\logs\Webservices_DBUpdate.log` 日誌檔，並確定已順利執行完成。

註：如果您只要啓用 OrderCreate Web 服務，則您必須註銷 SQL Script 中為 OrderFulfillment Web 服務登錄 `ExtOrderProcessServiceCmdImpl` 和 `WSDLFinderCmdImpl` 作業指令的程式行。

新的訊息對映

爲了啓用 OrderCreate Web 服務，`webservice_template.xml` 檔案提供了一組新的範例對映。以下是訊息對映的範本文件：







```
<!-- 入埠訂單建立訊息的對映 -->
<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>

```

這個服務的對應訊息對映稱為 BatchOrderCreate10Map，您可以在下列目錄中的 webservice_template.xml 範本檔裡取得它：

-  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

所提供的訊息對映會與下列目錄中所提供的 WSDL 定義相對應：

-  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

它可以使用 Web 服務用戶端所傳送的任何 SOAP 要求訊息（這個訊息是從所提供的 WSDL 定義產生的）。

定義訊息對映時，您需要瞭解 SOAP 要求中的 XPATH 資訊。您可以從 WSDL 定義中取得 XPATH 資訊。另一種更方便的方法是察看 SOAP 訊息。下一節將說明如何存取 SOAP 訊息以及定義對映範本。

如果要從對應的 SOAP 要求來定義一個與 OrderCreate 訊息 (BatchOrderCreate10Map) 相對應的訊息對映類似的訊息對映，請執行下列步驟：

1. 使用 WSDL 定義來產生用戶端 Proxy。請參閱第 43 頁的『附錄 B. 使用 IBM WebSphere Studio Application Developer』中所說明的步驟。
2. 使用傳輸層次截取公用程式來察看 SOAP 要求以及 SOAP 回應文件。例如，WebSphere Studio Application Developer 5.0 中的「TCP 監視器」，或任何可以記載訊息的其他 HTTP 通道公用程式。設定存取點，讓用戶端 Proxy 指向截取器。檢視 SOAP 要求，以協助您撰寫要求的對映，以及協助您撰寫 JSP 檔來建置回應。
3. 定義範本定義檔的 TemplateDocument 元素。其中包含在收到 SOAP XML 訊息時要執行的指令明細。其他資訊請參閱 WebSphere Commerce 正式作業與開發作業線上說明中的「範本定義檔的 TemplateDocument 元素」一節。
4. 使用您在步驟 2 中取得的 SOAP 要求來定義內含標籤定義清單的 TemplateTag。確定 TemplateTag 包含 WSDL 定義中的所有元素的對映。在從用戶端傳送要求時，您需要移入 SOAP 要求中的所有元素的資料。每一個標籤定義都會將送入的 SOAP XML 訊息中的某個元素或屬性對映到一個指令參數。請參閱 WebSphere Commerce 正式作業與開發作業線上說明中的「範本定義檔的 TemplateTag 元素」一節。

配置訊息對映程式

SOAP 訊息對映程式使用 `webservice_template.xml` Web 服務範本定義檔來將送入的 SOAP 要求對映到指令內容。因此，您可以藉由變更範本來配置 SOAP 訊息對映程式。

如果您有不同的 WSDL 定義或者要變更服務定義，就必須修改範本中的對映。對映是由對映至指令參數的 XML 訊息的 XPATH 定義所組成。在此情況下，您需要變更 XPATH 結構，以根據新的服務定義來符合資料類型的綱目。

OrderStatus 服務






本節說明如何在 WebSphere Commerce 中啓用商業邏輯作為 Web 服務，來更新在 WebSphere Commerce 中的訂單狀態。利用 OrderStatus Web 服務，您可以透過 Web 服務呼叫來呼叫 WebSphere Commerce OrderStatus 指令。OrderStatus 控制程式指令可讓外部系統更新與 WebSphere Commerce 中的現有訂單狀態相關的資訊。如果要使用 OrderStatus Web 服務範例，您必須有一份現成的訂單，也就是需要在 WebSphere Commerce 系統中擁有一家已經公佈的商店。

此處所提供的範例說明如何為 Web 服務要求啓用 WebSphere Commerce OrderStatus 指令。






啓用 OrderStatus Web 服務範例


以下是實作範例 OrderStatus Web 服務的步驟概觀：

1. 指出您要外曝作為 Web 服務的商業邏輯。在此情況下，請容許應用程式對訂單進行供貨，以便更新 WebSphere Commerce 中的訂單狀態。
2. 指出實作這個商業邏輯的指令。這裡指的是範例 OrderStatusService 指令。它可以加強現有的 OrderStatus 指令。
3. 指出這個指令所需要的必要及選用參數。這項資訊可以在 WebSphere Commerce 正式作業與開發作業線上說明中取得。
4. 建立 OrderStatus Web 服務的 WSDL 定義。範例 OrderStatus.wsdl 定義檔是位於下列目錄中：







-  `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`

5. 如果您已經啓用 OrderCreate 服務，請勿執行這個步驟。否則，請使用對映範本檔將送入的 SOAP 要求對映到指令參數。範例對映範本是在下面的目錄中：

-  `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/websevice_template.xml

a. 將 websevice_template.xml 檔複製到下列目錄中：

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

b. 如果 websevice_template.xml 檔已經存在，而且不含任何資訊，就會改寫它。如果檔案包含資訊，則將與 OrderStatus 服務對應的 TemplateDocument 複製到現有的檔案中。







6. 如果您已經啓用 OrderCreate 服務，請勿執行這個步驟。否則，請啓用 "SOAPINTEGRATION" 的訊息對映程式。如果要這麼做，請開啓位於下列目錄中的 *instance_name.xml* 配置檔：

-  *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







找到名稱爲 "WCS.SOAPINTEGRATION" 的訊息對映程式。如果 "enable" 屬性的值是 "false"，就將它設定爲 "true"。

7. 撰寫 JSP 範本來撰寫回應。您可以使用範例 JSP 檔 SOAP_OrderStatusResponse.jsp 來撰寫回應。

a. 請從下列目錄中找到 SOAP_OrderStatusResponse.jsp：







-  *WC_installdir*\samples\webservices\wcstores
-  *WCStudio_installdir*\Commerce\samples\webservices\wcstores
-    *WC_installdir*/samples/webservices/wcstores
-  *WC_installdir*/samples/webservices/wcstores

複製到下面的目錄中：





-  *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

當 WebSphere Commerce 中的訂單狀態順利更新時，這個 JSP 範本會撰寫 SOAP 回應訊息。

- b. 如果您已經啓用 OrderCreate 服務，就不需要執行這個步驟。否則，請從下面的目錄中找到 SOAP_GenericApplicationError.jsp 和 SOAP_GenericSystemError.jsp 檔案：












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

複製到下面的目錄中：

-  `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`


發生應用程式或系統錯誤時，這些 JSP 範本會撰寫 SOAP 錯誤訊息。

8. 如果您已經啓用 OrderCreate 服務，就不需要執行這個步驟。否則請執行下列步驟：
- a. 按照下列表格中的說明，找到您的配置所屬的正確 Script 檔案集：

作業系統	資料庫類型	Script 檔案名稱
 		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

+
+
+
+
+

- b. 將適當的 Script 檔複製到下列目錄中：

-  `WC_installdir\samples\webservices\bin`

- Studio WCStudio_installdir\Commerce\samples\webservices\bin
 - AIX Linux Solaris WC_installdir/samples/webservices/bin
 - 400 /QIBM/ProdData/CommerceServer55/samples/webservices/bin
- c. DB2 從下列目錄中找到 wcs.webservices.enable.sql 和 wcs.webservices.disable.sql Script :

- Windows WC_installdir\samples\webservices\schema\db2
- Studio WCStudio_installdir\Commerce\samples\webservices\ schema\db2
- AIX Linux Solaris WC_installdir/samples/webservices/ schema/db2
- 400 /QIBM/ProdData/CommerceServer55/samples/ webservices/schema/db2

複製到下面的目錄中：

- Windows WC_installdir\schema\db2
- Studio WCStudio_installdir\Commerce\schema\db2
- AIX Linux Solaris WC_installdir/schema/db2
- 400 /QIBM/ProdData/CommerceServer55/schema/db2

Oracle 從下列目錄中找到 wcs.webservices.enable.sql 和 wcs.webservices.disable.sql Script :

- Windows WC_installdir\samples\webservices\schema\oracle
- Studio WCStudio_installdir\Commerce\samples\ webservices\schema\oracle
- AIX Solaris WC_installdir/samples/webservices/schema/oracle

複製到下面的目錄中：

- Windows WC_installdir\schema\oracle
- Studio WCStudio_installdir\Commerce\schema\oracle
- AIX Solaris WC_installdir/schema/oracle

- d. 更新您在步驟 7c 中複製的 SQL Script 檔，方法是註銷與 OrderStatus Web 服務無關的 Script 陳述式。
 - e. 執行您在步驟 7b 中所複製的 Script 檔。
9. 如果您使用 WebSphere Commerce 機器，請重新啓動 Commerce Server 實例，使變更生效。如果您使用 WebSphere Commerce Studio 機器，則重新啓動 WebSphere 測試環境中的 WebSphereCommerceServer 伺服器。

OrderStatus 服務的存取控制

在 OrderStatus Web 服務中，會使用 WebSphere Commerce 存取控制模型。您需要在 Web 服務要求中傳送使用者 ID 和密碼，WebSphere Commerce 才能鑑別要求者，並驗證要求者有權執行必要的動作。在 WSDL 定義中定義的使用者 ID 和密碼如下：

認證的 XML 綱目格式必須與下面的格式類似：

```
<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>
```

認證資訊必須對映到範本定義檔中的下列欄位：

- 使用者登入的欄位名稱是 'logonId'，而 FieldInfo 是 'CONTROL'。
- 密碼的欄位名稱是 'logonPassword'，而 FieldInfo 是 'CONTROL'。

例如，在 OrderStatus 服務中，LogonID 和密碼是利用對映範本中的下列區段來對映到 OrderStatus 指令：

```
<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' />
```




這個範例提供了一種典型的方法，來符合為 WebSphere Commerce 所定義的 Web 服務的存取控制基本需求。

OrderStatus Web 服務的使用者職務

OrderStatus 服務使用 OrderStatusServiceCmd 和 OrderStatusCmd 控制程式指令。在預設的情況下，這些控制程式指令會當作資源來加入到 SellersCmdResourceGroup 資源群組中；並且當作動作來加入到 BackendOrderStatusCreate 動作群組中。因此，只有被指定為「賣方」職務的使用者才能執行 OrderStatusCmd 和 OrderStatusServiceCmd 指令。這表示將認證當作 OrderStatus 訊息的一部分來傳入的使用者必須具備「賣方」職務。在預設的情況下，任何具有 WebSphere Commerce 管理者權限的使用者都可以執行這些指令。所以，只有當您要為不具備管理者權限的使用者啟用 OrderStatus 服務時，才適用這一點。

OrderStatus WSDL 定義

OrderStatus Web 服務是根據下列目錄中可用的 OrderStatus WSDL 定義來定義的：

-  *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

新的訊息對映

爲了啓用 OrderStatus Web 服務，對映範本 webservice_template.xml 檔案中提供了一組新的對映。以下是訊息對映的範本文件：

```
<!-- 入埠訂單狀態更新訊息的對映 -->
<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'>
        <Constant Field='redirecturl'>
          SOAP_OrderStatusResponse
        </Constant>
        <Constant Field='SOAP_OrderStatusResponse'>
          SOAP_OrderStatusResponse.jsp
        </Constant>
      </Command>
    </CommandMapping>
  </TemplateDocument>
```

這個服務的對應訊息對映稱爲 OrderStatus20Map，您可以在下列目錄中的 webservice_template.xml 範本檔裡取得它：

- Windows WC_installdir\samples\webservices\xml\messaging
- Studio WCStudio_installdir\Commerce\samples\webservices\xml\messaging
- AIX Linux Solaris WC_installdir/samples/webservices/xml/messaging
- 400 /QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging

所提供的訊息對映會與下列目錄中所說明的 WSDL 定義相對應：

- Windows WC_installdir\samples\webservices\xml\ messaging\Orderstatus.wsdl
- Studio WCStudio_installdir\Commerce\samples\webservices\xml\ messaging\Orderstatus.wsdl
- AIX Linux Solaris WC_installdir/samples/webservices/xml/ messaging/Orderstatus.wsdl
- 400 /QIBM/ProdData/CommerceServer55/samples/webservices/ xml/messaging/OrderStatus.wsdl

訊息對映可以使用由 Web 服務用戶端所傳送的任何訊息（訊息是從所提供的 WSDL 定義來產生的）。

定義訊息對映時，您需要瞭解 SOAP 要求中的 XPATH 資訊。您可以從 WSDL 定義中取得 XPATH 資訊。另一種更方便的方法是察看 SOAP 訊息。下一節將說明如何存取 SOAP 訊息以及定義對映範本。

如果要從對應的 SOAP 要求來定義一個與 OrderStatus 訊息 (OrderStatus20Map) 相對應的訊息對映類似的訊息對映，請執行下列步驟：

1. 使用 WSDL 定義檔來產生用戶端 Proxy。請參閱第 43 頁的『附錄 B. 使用 IBM WebSphere Studio Application Developer』中所說明的步驟，以取得其他資訊。

2. 使用傳輸層次截取公用程式來察看 SOAP 要求以及 SOAP 回應文件。例如，Apache AXIS 發行版中的「TCP 監視器」，或任何可以記載訊息的其他 HTTP 通道公用程式。設定存取點，讓用戶端 Proxy 指向截取器。這樣可以協助您撰寫要求的對映，以及協助您撰寫 JSP 檔來建置回應。
3. 定義範本定義檔的 TemplateDocument 元素。其中包含在收到 XML 訊息時要執行的指令明細。其他資訊請參閱 WebSphere Commerce 正式作業與開發作業線上說明中的「範本定義檔的 TemplateDocument 元素」一節。
4. 定義內含標籤定義清單的 TemplateTag。確定 TemplateTag 包含 WSDL 定義中的所有元素的對映。在從用戶端傳送要求時，您需要移入 SOAP 要求中的所有元素的資料。每一個標籤定義都會將送入的 XML 訊息中的某個元素或屬性對映到一個指令參數。請參閱 WebSphere Commerce 正式作業與開發作業線上說明中的「範本定義檔的 TemplateTag 元素」一節。

配置訊息對映程式

SOAP 訊息對映程式使用 Web 服務範本定義檔 (webservice_template.xml) 來將送入的 SOAP 要求對映到指令內容。因此，您可以藉由修改範本來配置 SOAP 訊息對映程式。有關入埠訊息範本定義檔的其他資訊，請參閱 WebSphere Commerce 正式作業與開發作業線上說明。







如果您有不同的 WSDL 定義或者要變更服務定義，就必須修改範本中的對映。對映是由對映至指令參數的 XML 訊息的 XPATH 定義所組成。在此情況下，您需要變更 XPATH 結構，以根據新的服務定義來符合資料類型的綱目。

停用 OrderCreate 和 OrderStatus Web 服務

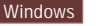

本節說明如何停用 WebSphere Commerce 所提供的 OrderCreate 和 OrderStatus Web 服務。





1. 刪除下列的 JSP 檔案：
 - SOAP_PurchaseOrderResponse.jsp
 - SOAP_GenericApplicationError.jsp
 - SOAP_GenericSystemError.jsp
 - SOAP_OrderStatusResponse.jsp

這些檔案位於下列目錄中：

-  `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. 將下列目錄中的 webservice_template.xml 檔案：

-  `WC_installdir\xml\messaging`
-  `WCStudio_installdir\Commerce\xml\messaging`







-    `WC_installdir/xml/messaging`
-  `/QIBM/UserData/CommerceServer55/instances/
instance_name/xml/messaging`

取代為內含空白 ECTemplate 節點的空白範本，如下列範例所示：





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

3. 在下列目錄中找到 `wcs.webservices.disable.sql` Script：

 如果您使用 DB2 資料庫：

-  `WC_installdir\schemadb2`
-  `WCStudio_installdir\Commerce\schemadb2`
-    `WC_installdir/schema/db2`
-  `QIBM/UserData/CommerceServer55/instances/instance_name/schema`

 如果您使用 Oracle 資料庫：

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

4. 開啓您在步驟 3 中找到的 `wcs.webservices.disable.sql` Script，在文字編輯器中，確定下面的值與 `wcs.webservices.enable.sql` Script 中的值相同：





- `storeent_id` - 是您商店的唯一識別碼。
- `OrgCode1` - 這是 ORGCODE 表格中與供應商組織相對應的項目的主要鍵。
- `OrgCode2` - 這是 ORGCODE 表格中與買方組織相對應的項目的主要鍵。
- `MemberId` - 這是買方組織所屬的成員群組的主要鍵。

5. 在完成步驟 4 中所說明的變更之後，請執行適當的指令 Script 檔：

 如果您使用 DB2 資料庫：

-   從 DB2 指令視窗中，執行 `Webservices_UndoDBUpdate.db2.bat`
-    從指令視窗中，執行 `Webservices_UndoDBUpdate.db2.sh`
-  啓動一個 QSH 階段作業。執行指令 `/QIBM/ProdData/CommerceServer55/samples/webservices/bin/
Webservices_UndoDBUpdate.db2.sh`

 如果您使用 Oracle 資料庫：


-   從 DB2 指令視窗中，執行 `Webservices_UndoDBUpdate.oracle.bat`
-   從指令視窗中，執行 `Webservices_UndoDBUpdate.oracle.sh`

驗證日誌檔，確定已順利執行完成。在完成 Script 檔的執行之後，會顯示日誌檔的名稱和位置。

註：如果您只要顯示 OrderCreate 和 OrderStatus Web 服務，或者您尚未啓用 OrderFulfillment Web 服務，請註銷步驟 4 中列出的 Script 檔中的下列 SQL 陳述式：

- `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';`

其中，`storeent_id` 是您商店的唯一識別碼。

6.  刪除您在第 12 頁的『匯入範例程式碼』中所匯入的範例程式碼，然後重新建置 WebSphereCommerceServerExtensionsLogic 專案。
7. 如果您使用 WebSphere Commerce 機器，請重新啓動 Commerce Server 實例，使變更生效。如果您使用 WebSphere Commerce Studio 機器，則重新啓動 WebSphere 測試環境中的 WebSphereCommerceServer 伺服器。

第 4 章 將 WebSphere Commerce 啓用作為服務要求者

本章說明如何啓用 WebSphere Commerce 來呼叫外部系統所定義的 Web 服務。本章也說明所提供的範例，這些範例可以讓 WebSphere Commerce 尋找及呼叫 Web 服務。此外，也說明 OrderFulfillment Web 服務範例的實作明細。

架構

這個架構中的訊息流程是同步的。這表示 WebSphere Commerce 會先等待來自外部系統的回應，然後才繼續進一步的處理。同時藉由呼叫 Web 服務來實作作業指令，以便與外部系統相互作用。

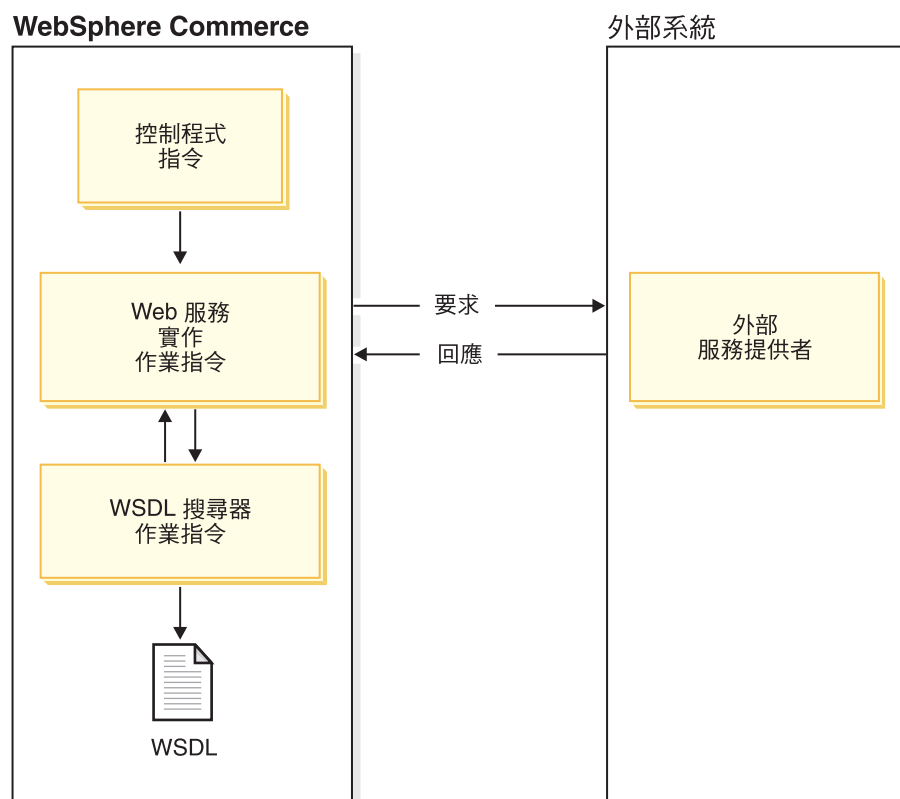


圖 5. 將 WebSphere Commerce 啓用作為服務要求者的架構

WebSphere Commerce 程式設計模型可讓您藉由改寫指令來自訂現有商業程序的規則。控制程式指令會呼叫許多作業指令來完成商業程序的執行。修改控制程式指令規則的其中一種方法，是為控制程式指令所呼叫的作業指令撰寫新的實作。







如果要尋找當作控制程式指令的一部分來呼叫的作業指令，請參閱 WebSphere Commerce 正式作業與開發作業線上說明中的「指令、作業和表格的交互參照」一節。下列清單說明 WebSphere Commerce 商業程序在呼叫外部系統所代管的 Web 服務時的互動：

1. 控制程式指令呼叫已登錄的作業指令。在此情況下，會藉由呼叫 Web 服務來實作作業，以便與外部系統相互作用。這個作業指令包含用來整合外部系統的程式碼。







2. 作業指令擷取已指定的 Web 服務的存取點。
3. 作業指令呼叫 Web 服務 Proxy，而這個 Proxy 指定服務存取點及其輸入參數。
4. Proxy 呼叫 Web 服務，接收回應，然後將它傳回給作業指令。

尋找服務實作

為了啟用 WebSphere Commerce 來呼叫外部系統所定義的 Web 服務，您必須先找到為所需要的服務類型代管實作的服務提供者。如果要呼叫 Web 服務，您可以將作業指令指向已識別的服務提供者。為了提供更多的彈性，這裡也提供一個作業指令來協助您找到服務提供者，您可以在 WSDLFinderCmd 作業指令中找到它的定義。下列目錄中有這個作業指令的範例實作：

-  `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`

尋找服務實作的其中一個方法是察看服務登錄。然而，WebSphere Commerce 5.5 版所提供的範例使用本端檔案系統來擷取 Web 服務的 WSDL 定義。WSDLFinderCmdImpl 會在 `instance_name.xml` 配置檔中尋找可從本端環境中取得的已登錄 WSDL 定義檔。如果要登錄 WSDL 服務定義，請編輯下列目錄中的 `instance_name.xml` 配置檔：

-  `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`

然後執行下列步驟：

1. 找到 SOAPWSDLFinderInfo 元素。這個元素包含許多屬性，以用來登錄 WSDL 定義路徑以及 WSDL 檔案。以下摘錄自 `instance_name.xml` 檔案，顯示這些參數：

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

2. 設定 LocalWSDLFiles 的值，以併入在您的本端檔案系統中以逗號隔開的 WSDL 定義清單，並且將 LocalWSDLPath 設定成這些檔案所在的目錄。
3. 將 UDDIQueryURL 和 UDDIPublishURL 的值設定成空白字串，因為這個範例使用本端檔案系統來尋找服務定義。
4. 將 WSDLFinderImplClassName 的值設定成空白字串。

如果要以動態方式尋找夥伴以及呼叫夥伴在服務登錄中已經實作並公佈的服務，就必須對 WebSphere Commerce 5.5 版所提供的 WSDLFinderCmdImpl 作業指令範例建立及登錄一個新的實作。在您的自訂實作中，併入用來查閱 WSDL 定義的程式碼。在這裡，您可以使用 UDDIQueryURL 和 UDDIPublishURL 欄位來登錄 UDDI 登錄。有關如何改寫作業指令的其他資訊，請參閱 *WebSphere Commerce 程式設計手冊與指導教學*。

呼叫服務實作

如果要尋找當作控制程式指令的一部分來呼叫的作業指令，請參閱 WebSphere Commerce 正式作業與開發作業線上說明中的「指令、作業和表格的交互參照」一節。


下表顯示在 OrderProcessCmd 中一併執行的作業：

作業指令名稱	說明
CheckOrderCmd	CheckOrderCmd 作業指令會檢查處於「已提交」狀態的訂單是否已經處理。
DoPaymentCmd	OrderProcessCmd 控制程式指令會呼叫 DoPaymentCmd 作業指令來處理訂單付款。
ExtOrderProcessCmd	在完成 OrderProcess 指令之前，執行任何必要的其他程序。
UpdateInventoryCmd	UpdateInventory 作業指令可讓您更新項目的庫存。

從這個表格中可以看到，如果您要自訂 OrderProcess 指令的規則，將供貨要求傳送到外部供貨中心，您可以改寫 ExtOrderProcessCmd 作業指令。新的 ExtOrderProcessCmd 實作可以查閱已經實作 OrderFulfillment 服務的夥伴並且呼叫服務。

WebSphere Commerce 可以在外部系統中呼叫的 Web 服務範例

本節說明如何啓用 OrderFulfillment Web 服務範例。

您可以在 WebSphere Commerce 機器或已安裝 WebSphere Commerce Studio 的開發機器上啓用 OrderFulfillment 範例。如果您要在 WebSphere Commerce Studio 機器上啓用這個範例，請遵循本節中的指示，並參閱標示為「 Studio」的目錄路徑。

OrderFulfillment 服務

本節說明如何啓用 WebSphere Commerce 來呼叫外部系統所實作的 Web 服務。OrderFulfillment 服務是用來示範如何將 WebSphere Commerce 當作服務要求者。如果要使用 OrderFulfillment Web 服務範例，您需要在 WebSphere Commerce 系統中擁有一家已經公佈的商店。

如果您要在 WebSphere Commerce 環境中啓用這個 Web 服務範例，則您還需要一個 WebSphere Commerce Studio 環境。

介面 WSDL 定義

OrderFulfillment 服務的商業邏輯的設計是爲了與預先定義的服務介面一起使用，也就是 OrderFulfillment 服務。OrderFulfillment 服務的介面定義 WSDL 檔是在本端檔案系統中，以供參照以及對已知的服務進行預設靜態連結。

作業指令





當作 `OrderProcessCmd` 的一部分來呼叫的 `ExtOrderProcessServiceCmdImpl` 指令會實作 `ExtOrderProcessServiceCmd` 介面，來與外部服務相互作用。它會透過 `Proxy` 來起始對 `Web` 服務的要求。這個作業指令是藉由傳送 `OrderFulfillment` 要求到相關的供貨中心，來處理已提交的訂單。

以下是啓用 `OrderFulfillment` 服務的步驟：





1. 如果您已經按照第 3 章，第 17 頁的『將資料移入資料庫表格』中的說明，將 `WebSphere Commerce` 啓用作為服務提供者，請不要執行這個步驟。然而，如果您先前已經註銷 `OrderFulfillment` 指令的區段，您就必須執行這個步驟。

如果要將 `ExtOrderProcessServiceCmdImpl` 和 `WSDLFinderCmdImpl` 登錄到 `WebSphere Commerce Server` 指令登錄中，請執行與這些指令相關的 `SQL` 陳述式。您可以在下面的目錄中找到 `SQL Script` 檔：

 如果您使用 `DB2` 資料庫：

-  `WC_installdir\samples\webservices\schema\db2\wcs.webservices.enable.sql`
-  `WCStudio_installdir\Commerce\samples\webservices\schema\db2\wcs.webservices.enable.sql`
-    `WC_installdir/samples/webservices/schema/db2/wcs.webservices.enable.sql`
-  `/QIBM/ProdData/CommerceServer55/samples/webservices/schema/db2/wcs.webservices.enable.sql`

 如果您使用 `Oracle` 資料庫：

-  `WC_installdir\samples\webservices\schema\oracle\wcs.webservices.enable.sql`
-  `WCStudio_installdir\Commerce\samples\webservices\schema\oracle\wcs.webservices.enable.sql`
-   `WC_installdir/samples/webservices/schema/oracle/wcs.webservices.enable.sql`

以下是您必須執行的前述 `Script` 檔的 `SQL` 陳述式：

```
insert into cmdreg (storeent_id, interfacename, description,
  classname, properties, lastupdate, target) values(0, 'com.ibm.commerce.
webservice.utils.WSDLFinderCmd', 'WsdI 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');
```

其中，`storeent_id` 是您商店的唯一識別碼。







2. 如果要啓用 OrderFulfillment 服務的查閱，請執行下列步驟：
 - a. 在 *instance_name.xml* 配置檔中，找到 SOAPWSDLFinderInfo 元素。
 - b. 確定 LocalWSDLFiles 和 LocalWSDLPath 的值是正確的，並且反映出 WSDL 定義檔所在的目錄。SOAPWSDLFinderInfo 元素必須與下面類似：

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

- c. 將下列目錄中的範例 OrderFulfillment.wsdl WSDL 檔：

-  *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*

複製到下面的目錄中：

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

3. 爲 OrderFulfillment.wsdl 檔案中定義的服務指出服務提供者。使用服務提供者的資訊來更新 OrderFulfillment.wsdl。
 - a. 開啓您在步驟 2c 中所複製的 OrderFulfillment.wsdl 檔案。
 - b. 找到服務定義節點（以 service name 作爲開頭的區段）。它看起來與下面類似：


```
<service name=" POCreateService ">
<port binding="tns: POCreateBinding " name=" POCreatePort">
<soap:address location="http://localhost:8080/WebServicesForWCBEWeb/
servlet/rpcrouter"/>
</port>
</service>
```
 - c. 將現有且已經標示的 URL 變更為目標 Web 服務的 URL。
 - d. 儲存並關閉檔案。
4. 如果您已經啓用 OrderCreate 或 OrderStatus Web 服務範例，請略過步驟 4 和 5，如果沒有的話，請繼續執行這個步驟。如果要部署爲 OrderFulfillment Web 服務所提供的範例程式碼，就必須具備已經安裝和配置 WebSphere Commerce Studio 的開發機器。下列清單中的步驟（WebSphere Studio Application Developer）必須在這部開發機器上執行。
5. 如果您要編輯或修改 Web 服務範例，就必須將範例原始碼匯入到 WebSphere Commerce Studio 中。如果要匯入這個程式碼，請執行下列步驟：

- a. 開啓 WebSphere Commerce 開發環境，方法是選取開始 > 程式集 > **IBM WebSphere Commerce Studio > WebSphere Commerce** 開發環境。
 - b. 切換到 J2EE 視景並選取「J2EE 導覽器」檢視畫面。
 - c. 展開 **WebSphereCommerceServerExtensionsLogic** 專案。用滑鼠右鍵按一下 **src** 目錄，然後選取匯入。這時會開啓「匯入精靈」。
 - d. 選取**檔案系統**，然後按一下下一步。
 - e. 按一下**瀏覽**並尋找下列目錄：
`WCStudio_install_dir\Commerce\samples\webservices\com`
 - f. 按一下**全選**，然後按一下**完成**。
 - g. 用滑鼠右鍵按一下 **WebSphereCommerceServerExtensionsLogic** 專案，然後選取**內容**。
 - h. 按一下 **Java 建置路徑**，然後移至**程式庫**標籤。
 - i. 按一下**新增變數**。選取 SOAPJAR 和 XERCES 變數，然後按一下**確定**。
 - j. 按一下**確定**。
 - k. 用滑鼠右鍵按一下 **WebSphereCommerceServerExtensionsLogic** 專案，然後選取**重新建置專案**。
6. 如果您要在 WebSphere Commerce 伺服器上啓用 Web 服務範例，就必須將這個程式碼部署到目標 WebSphere Commerce 伺服器上。請參閱 *WebSphere Commerce 程式設計手冊與指導教學*，以取得有關如何部署新指令的資訊。
 7. 如果您使用 WebSphere Commerce 機器，請重新啓動 Commerce Server 實例，使變更生效。如果您使用 WebSphere Commerce Studio 機器，則重新啓動 WebSphere 測試環境中的 WebSphereCommerceServer 伺服器。

商業物件

ExtOrderProcessServiceCmdImpl 作業指令會將資料移入一組商業物件中，讓 Web 服務 Proxy 用來建構 SOAP XML 要求訊息。這個要求訊息反映出 OrderFulfillment WSDL 定義的元素。OrderFulfillment 服務包含一組從 OrderFulfillment WSDL 定義產生的範例商業物件。當 WebSphere Commerce 呼叫外部 Web 服務時，您可以按照第 43 頁的『附錄 B. 使用 IBM WebSphere Studio Application Developer』中的說明，使用 IBM WebSphere Studio 或其他工具為它們產生用戶端 Proxy 和商業物件。

下表列出 OrderFulfillment 服務所使用的套件以及它們的商業物件：

套件名稱	物件名稱	說明
com.ibm.commerce. webservice.datatype	Address ContactInfo ContactPersonName ControlArea Credentials Email InvoiceInfo ProductDimension ProductMeasurement ProductWeight RequisitionerID Telephone UserData UserDataField	這個套件是由不同的 Web 服務 訊息中使用的所有物件組成。

套件名稱	物件名稱	說明
com.ibm.commerce. webservice.datatype. orderfulfillment	BillToInfo BuyOrgAccountingDetail BuyOrgInfo DataArea DateTimeReference ItemShippingSchedule ItemUnitPrice MerchantInfo MonetaryAmount OrderFulfillmentStatus PCardInfo Report_NC_PurchaseOrder ReportPO ReportPOHeader ReportPOItem RequisitionerInfo ServiceAllowanceCharge ShipDateReference ShippingCarrierInfo ShipToInfo TaxInfo TotalPriceInfo	這個套件包含了其中有 OrderFulfillment XML 訊息特定資料的所有物件。 Report_NC_PurchaseOrder 是母項外層物件，它被傳送至 Proxy 以及當作回應來接收的 OrderFulfillmentStatus 物件。




測試範例

當您在 WebSphere Commerce 中已公佈的商店中下訂單時，會呼叫 OrderFulfillment Web 服務。Web 服務會連線到外部服務公司來對供貨中心下訂單。如果服務提供者傳回訂單 ID，就會在 WebSphere Commerce 資料庫中的 ORDERS 表格中，更新 ORMORDER 直欄中的 OrderID。

停用 OrderFulfillment Web 服務

本節說明如何停用 WebSphere Commerce 所提供的範例 OrderFulfillment Web 服務。

1. 刪除您爲了啓用 OrderFulfillment Web 服務所部署的 webservices.jar JAR 檔。
2. 從下列目錄中刪除 OrderFulfillment.wsd1 檔：

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

3. 從 `instance_name.xml` 配置檔中移除對 OrderFulfillment.wsd1 檔案的參照。
4. 如果您尚未執行 SQL Script 來停用 OrderCreate 和 OrderStatus Web 服務，或者在停用 OrderCreate 和 OrderStatus Web 服務時尚未註銷下列 SQL 陳述式，請執行下列 SQL 陳述式：







```
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';
```





其中，`storeent_id` 是您商店的唯一識別碼。

前面的 SQL 陳述式可以從下面的 Script 檔中取得：


 如果您使用 DB2 資料庫：

-   從 DB2 指令視窗中，執行 `Webservices_UndoDBUpdate.db2.bat`
-    從指令視窗中，執行 `Webservices_UndoDBUpdate.db2.sh`
-  啓動一個 QSH 階段作業。執行 `Webservices_UndoDBUpdate.db2.sh` 指令。

 如果您使用 Oracle 資料庫：







-   從 DB2 指令視窗中，執行 `Webservices_UndoDBUpdate.oracle.bat`
-   從指令視窗中，執行 `Webservices_UndoDBUpdate.oracle.sh`

驗證日誌檔，確定已順利執行完成。在完成 Script 檔的執行之後，會顯示日誌檔的名稱和位置。

5.  刪除您在第 12 頁的『匯入範例程式碼』中所匯入的範例程式碼，然後重新建置 `WebSphereCommerceServerExtensionsLogic` 專案。
6. 如果您使用 WebSphere Commerce 機器，請重新啓動 Commerce Server 實例，使變更生效。如果您使用 WebSphere Commerce Studio 機器，則重新啓動 WebSphere 測試環境中的 `WebSphereCommerceServer` 伺服器。

附錄 A. 範例 WSDL 定義檔

您可以從下列目錄中取得 WebSphere Commerce 5.5 版所提供的 Web 服務範例的 WSDL 定義：

-  `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 服務名稱	檔名
OrderFulfillment	OrderFulfillment.wsdl
OrderCreate	OrderCreate.wsdl
OrderStatus	OrderStatus.wsdl

附錄 B. 使用 IBM WebSphere Studio Application Developer

您可以使用許多不同的工具來產生 Web 服務程式碼。這個 Web 服務實作所提供的範例程式碼是利用 IBM WebSphere Studio Application Developer 來產生的。本節中的資訊已經在 IBM WebSphere Studio Application Developer 5.0 版中測試過。如果您沒有 WebSphere Commerce，或者您沒有使用 WebSphere Commerce Studio，但是想要開發 Web 服務用戶端來存取 WebSphere Commerce 所提供的 Web 服務，您可以使用任何可以產生 Web 服務用戶端程式碼的工具。

本附錄說明如何使用 WebSphere Studio 來為 WebSphere Commerce 所提供的服務產生測試用戶端。這個步驟包括將 WSDL 定義檔匯入到 WebSphere Studio 中、產生測試用戶端、執行測試用戶端、建立及配置「TCP 監視器」伺服器實例，以及使用「TCP 監視器」來追蹤 SOAP 訊息。

您也可以使用「Web 服務工具集 (WSTK)」來產生用戶端程式碼和資料類型。有關「Web 服務工具集」的其他資訊，請造訪

<http://www.alphaworks.ibm.com/tech/webservicestoolkit>。WSTK 展現了 Web 服務的新技術，並協助您瞭解最新的 Web 服務規格。然而，對於 Web 服務的產品層次開發環境而言，建議您使用 IBM WebSphere Studio Application Developer。

另外，您可以用 Apache SOAP 實作來作為用戶端，用手動方式撰寫用戶端程式碼和資料類型。相關明細請參閱 Apache SOAP 文件。

本節假設您已經具備 WebSphere Studio Application Developer 的知識。如果您對於使用 WebSphere Studio 來開發 Web 服務及企業應用程式不熟悉，建議您先閱讀 WebSphere Studio 文件或 IBM developerworks 網站中的指導教學 (<http://www.ibm.com/developerworks/webservices>)。

將 WSDL 匯入到 WebSphere Studio Application Developer 中

在將 WSDL 匯入到 WebSphere Studio 之前，請確定您已經建立 Web 應用程式，同時也已經建立這個 Web 應用程式所屬的企業應用程式。如果 Web 應用程式不存在，在繼續下面的指示之前，請先建立它。如果您已經設定好 WebSphere Commerce 開發環境，請不要建立新的企業應用程式和 Web 應用程式。反之，請使用 WebSphereCommerceServerExtensionsLogic 專案。

以下是使用 IBM WebSphere Studio 來產生用戶端程式碼和資料類型的程序概觀：

1. 在您要匯入 WSDL 定義的 Web 專案中，建立一個新資料夾，例如 `wsdl`。如果您沒有 Web 專案，請建立 Web 專案。
2. 用滑鼠右鍵按一下 `wsdl` 資料夾。
 - a. 選取**匯入** > **檔案系統**，然後按一下**下一步**。
 - b. 按一下**瀏覽**並導覽至儲存 WSDL 定義檔的資料夾。
 - c. 按一下**確定**。您會在下面的目錄中找到 WSDL 定義：

-  `WC_installdir\samples\webservices\xml\messaging`

- Studio WCStudio_installdir\Commerce\samples\webservices\xml\messaging
- AIX Linux Solaris WC_installdir/samples/webservices/xml/messaging
- 400 /QIBM/ProdData/CommerceServer55/samples/webservices/xml/messaging

3. 從右邊的畫面中，選取您要匯入到 WebSphere Studio 中的 WSDL 檔案，然後按一下完成。

產生用戶端程式碼和資料類型

將 WSDL 定義匯入 WebSphere Studio 之後，您可以產生用戶端 Proxy 程式碼和一個測試範例用戶端來測試 Web 服務。用戶端 Proxy 是一種 Java 類別，您可以利用它來從 Java 用戶端存取 Web 服務。測試範例用戶端提供了一個 JSP 範例，它讓您從 WebSphere Studio 來測試 Web 服務。請執行下列步驟來產生用戶端 Proxy 和測試範例用戶端：

註： 您可以使用下面的指令集來產生用戶端程式碼，以便從 WebSphere Commerce 來存取外部用戶端所代管的 Web 服務。唯一的差別是您必須從服務提供者取得外部服務實作定義，然後將該 WSDL 定義匯入 WebSphere Studio 中，才能夠開始。

1. 移至 Web 視景，然後選取您要儲存所產生的用戶端程式碼的 Web 專案。如果您沒有任何 Web 專案，請建立 Web 專案。
2. 選取 WSDL 定義檔並按一下滑鼠右鍵，來產生用戶端程式碼。從功能表中，選取**新建 > 其他**。這時會顯示「新建」精靈。
3. 從左邊的畫面中選取 **Web 服務**，然後從右邊的畫面中選取 **Web 服務用戶端**。按一下**下一步**。
4. 從 Web 服務用戶端對話框中，選取 **Java Proxy**，然後按一下**下一步**。
5. 在「Web 服務 WSDL 檔案選擇」對話框中，確定所選的 WSDL 檔案已經選取，然後按一下**下一步**。
6. 請記下「Web 服務連結 Proxy 產生」對話框中的類別名稱，然後按一下**下一步**。
7. 選取**測試產生的 Proxy**。確定測試設備的值是設定成 Web 服務範例 JSP 檔案。按一下**完成**。
8. 切換到 Web 視景，然後展開「範例」資料夾。您將會看到測試 JSP 用戶端。

測試 Web 服務

本節說明如何使用您在前一節中產生的測試範例用戶端來測試 WebSphere Commerce 所提供的 Web 服務。請執行下列步驟來測試 Web 服務：

1. 確定至少有一個 WebSphere 測試環境伺服器存在。如果沒有伺服器，請執行下列步驟來建立伺服器：
 - a. 移至伺服器視景。
 - b. 選取**檔案 > 新建 > 伺服器與伺服器配置**。
 - c. 輸入伺服器的名稱，然後選取 **WebSphere 5.0 版 > 測試環境**作為類型。

2. 用滑鼠右鍵按一下 TestClient.jsp，然後選取在伺服器上執行。測試用戶端會出現在瀏覽器視窗中。
3. 從左頁框中選取 **getEndPoint()** 方法，然後按一下**呼叫**。在結果頁框中，確定傳回的端點是您要連線的伺服器的 URL。
4. 如果傳回的端點與執行 WebSphere Commerce 伺服器的伺服器不符，請將 URL 設定成指向 WebSphere Commerce Server：
 - a. 從左頁框中選取 **setEndPoint()** 方法。
 - b. 更新右頁框中的 WebSphere Commerce 伺服器 URL，然後按一下**呼叫**。
5. 選取用戶端 Proxy 的商業方法。如果是 OrderCreate Proxy，則商業方法是 orderCreate。
 - a. 在套表中輸入要傳送的參數的明細，套件是顯示在右頁框中。
 - b. 輸入所有必要欄位的值。
 - c. 按一下**呼叫**。這時會撰寫一則 SOAP 訊息，並傳送到端點所指出的伺服器。

您可以在「結果」頁框中看到呼叫 Web 服務的結果。

使用 TCP 監視器來追蹤 SOAP 訊息

本節說明如何建立及配置 TCP/IP 監視器伺服器實例，以及如何使用「TCP/IP 監視器」伺服器來追蹤 SOAP 訊息。它能協助您對 Web 服務進行除錯以及為您開發的 Web 服務定義對映範本。

「TCP/IP 監視器」是一個簡易的伺服器，可監視 Web 瀏覽器以及應用程式伺服器之間的所有要求和回應。它可以擷取並顯示用戶端與應用程式伺服器之間的所有流量。

在這裡，您需要建立一個新的「TCP/IP 監視器」伺服器實例以及執行下列步驟：

- 配置「TCP/IP 監視器」伺服器，將要求轉遞到 WebSphere Commerce 伺服器。
- 配置您的測試用戶端來指向「TCP/IP 監視器」，讓您可以截取 SOAP 要求和回應。

建立 TCP 監視器伺服器實例

如果要建立一個叫做 TCPMon 的「TCP/IP 監視器」伺服器實例，請執行下列步驟：

1. 從主功能表中，選取**檔案 > 新建 > 其他**。
2. 從左畫面中選取**伺服器**，然後從右畫面中選取**伺服器與伺服器配置**，然後按一下**下一步**。
3. 在「建立新的伺服器與伺服器配置」視窗中，執行下列步驟：
 - a. 輸入**伺服器名稱**，例如，TCPMon。
 - b. 選取**伺服器類型**，亦即 TCP/IP 監視伺服器。
 - c. 按一下**下一步**。
4. 在「建立新的伺服器與伺服器配置」視窗中，執行下列步驟：
 - a. 在**遠端主機欄位**中，輸入 WebSphere Commerce 伺服器主機名稱。
 - b. 在**遠端埠欄位**中，輸入 WebSphere Commerce 伺服器接聽的埠號。如果執行 WebSphere Commerce 的機器與用戶端相同，您可以使用 localhost 來作為主機名稱。
 - c. 按一下**完成**。

配置 TCP 監視器伺服器實例

配置「TCP 監視器」，使 WebSphere Commerce Server 的接聽埠是「TCP 監視器」的遠端埠。請執行以下步驟：

1. 移至伺服器視景。
2. 按兩下伺服器配置檢視畫面中的「TCP 監視器」實例。「TCP 監視器」的伺服器配置會開啓在編輯器檢視畫面中，如下圖所示：

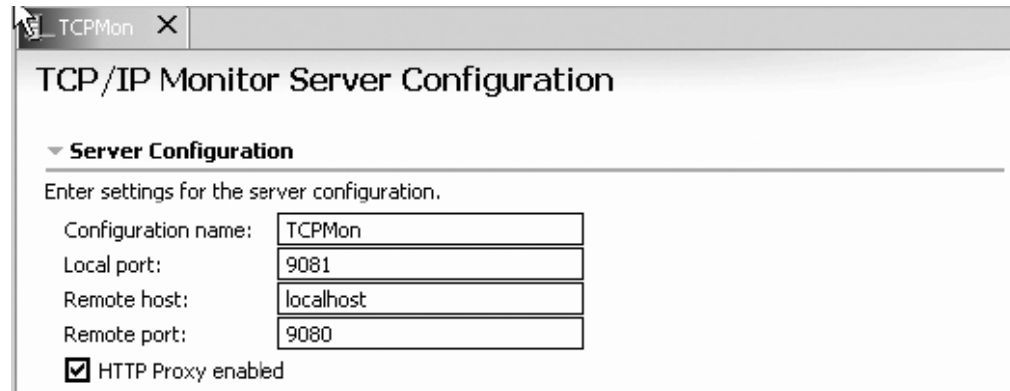


圖 6. 配置 TCP/IP 監視器伺服器配置

在這裡，本端埠號是指 TCP 監視器接聽的埠號，而遠端埠號是 WebSphere Commerce 伺服器接聽的埠號。

使用 TCP/IP 監視器來追蹤 SOAP 訊息

在配置「TCP/IP 監視器」後，請使用它來追蹤 SOAP 訊息。如果要這麼做，請啓動在伺服器上所產生的測試範例用戶端，然後配置它來將要求傳送至「TCP/IP 監視器」。請執行以下步驟：

1. 確定「TCP/IP 監視器」和 WebSphere Commerce 伺服器正在執行中。
 - a. 如果 WebSphere Commerce 伺服器是在 WebSphere Studio 外部執行，則使用 WebSphere Application Server 管理主控台來啓動伺服器。
 - b. 如果 WebSphere Commerce 伺服器是在 WebSphere Commerce Studio 內部執行，則移至**伺服器**視景。開啓**伺服器**檢視畫面。如果 WebSphereCommerceServer 伺服器的狀態是「已停止」，請用滑鼠右鍵按一下伺服器，然後選取**啓動**。請等待「伺服器」檢視畫面中的「狀態」變成「已啓動」。
2. 移至 Web 視景。瀏覽 sample\TestClient.jsp。用滑鼠右鍵按一下 sample\TestClient.jsp，然後選取在**伺服器上執行**。
3. 在伺服器選項視窗中，確定必須執行 JSP 檔的伺服器已經選取。按一下**完成**。
4. 如果系統提示您選取伺服器用戶端，請選取 **Web 瀏覽器**，然後按一下**完成**。這時會開啓一個包含三個頁框的新瀏覽器視窗，如下圖所示：

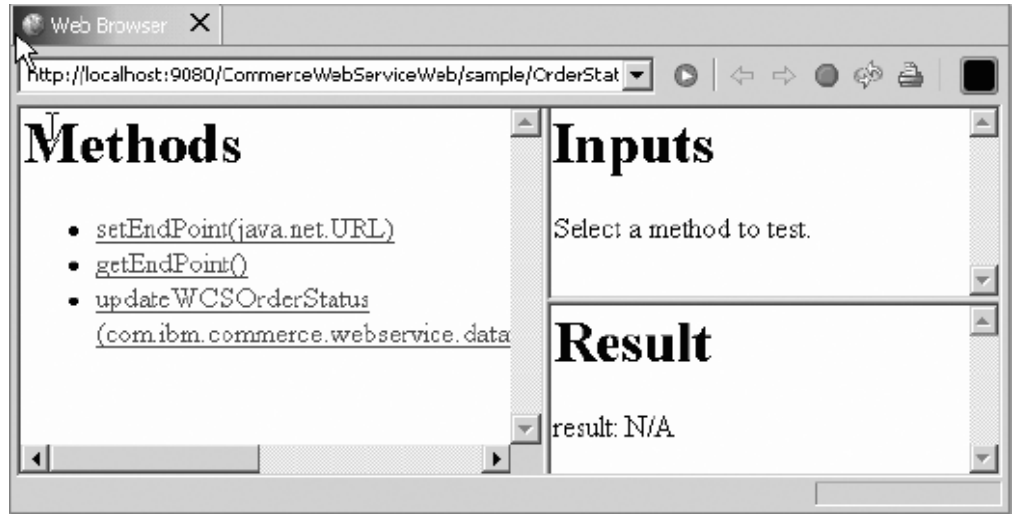


圖 7. 選取一個 Web 瀏覽器

- a. 從「方法」頁框中選取 **getEndPoint()** 方法，如下圖所示：

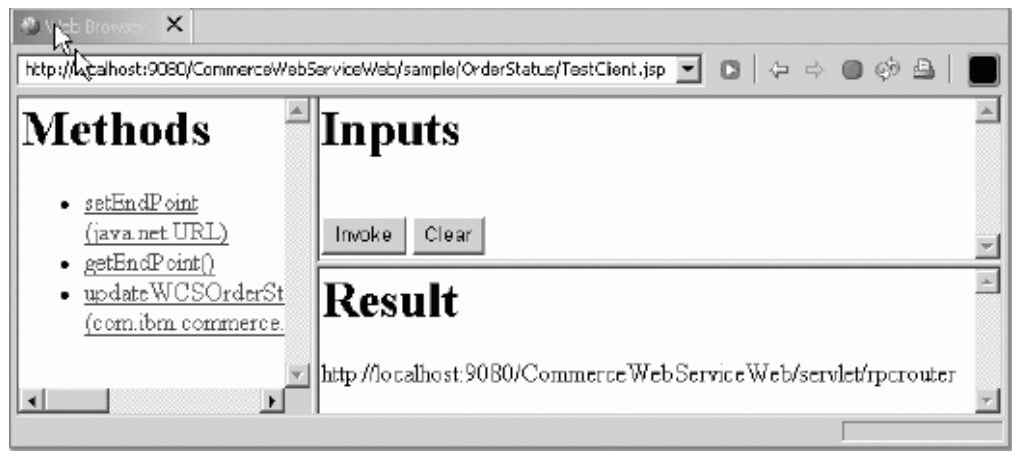


圖 8. 選取 *getEndPoint()* 方法

- b. 按一下「輸入」頁框中的**呼叫**。結果會顯示在「結果」頁框中
5. 在配置測試用戶端以將要求傳送到「TCP 監視器」時，必須執行下列步驟：
 - a. 從「方法」頁框中，選取 **setEndPoint()** 方法。
 - b. 在「輸入」頁框中，輸入「TCP 監視器」的 **url**。例如，
 http://localhost:9080/webapp/wcs/stores/servlet/，其中 9080 是執行「TCP 監視器」的埠號，而 webapp/wcs/stores/servlet/ 是 WebSphere Commerce 要求 Servlet 的 Web 路徑。

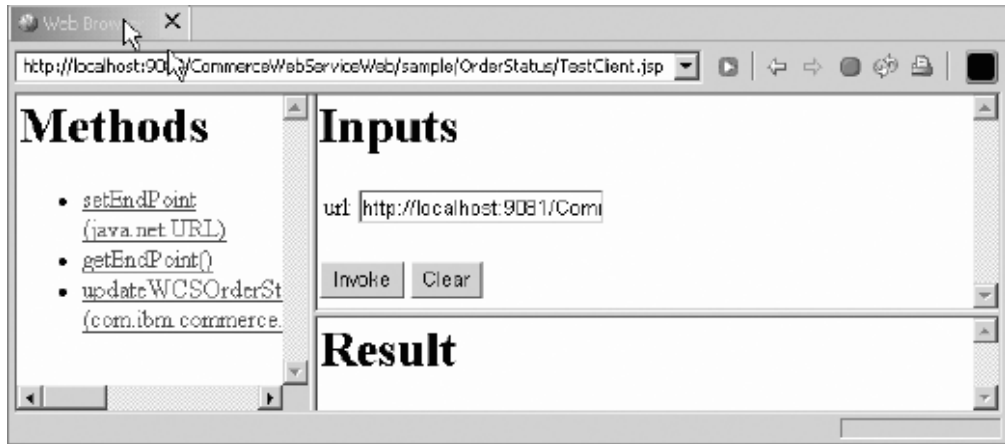


圖 9. 選取 `setEndPoint()` 方法

- c. 按一下呼叫。
6. 重新呼叫 `getEndPoint()` 方法來驗證端點。
7. 如果要呼叫商業邏輯，請執行下列步驟：
 - a. 按一下「方法」頁框中的商業邏輯方法，例如，`updateWCSOrderStatus()`。
 - b. 在「輸入」頁框中，輸入所有必要參數的值，如下圖所示：

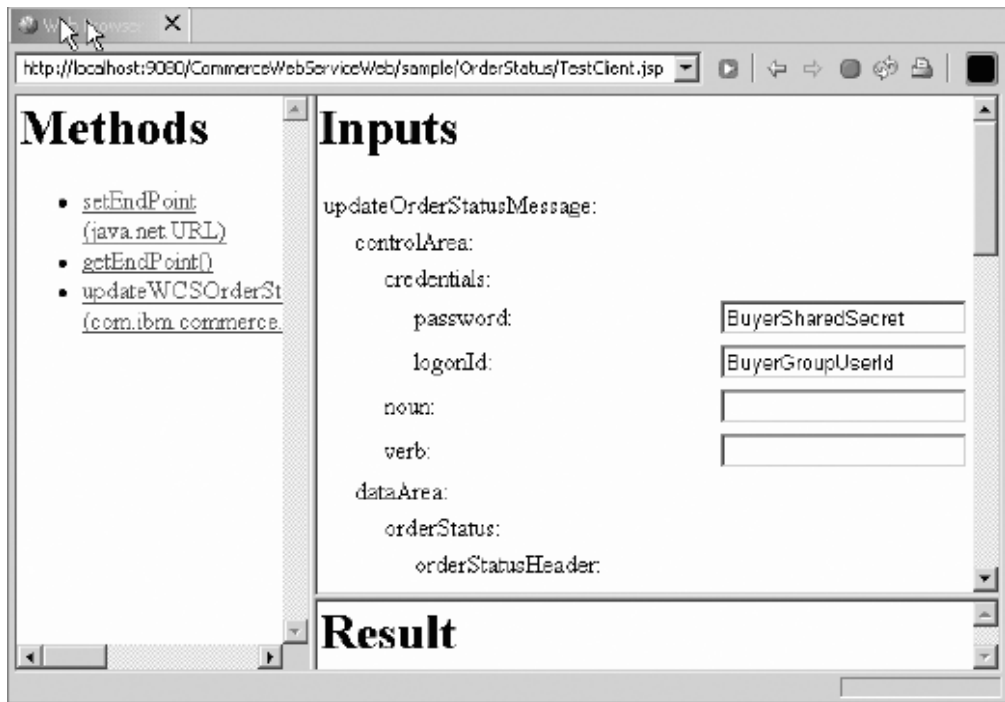


圖 10. 選取商業邏輯

- c. 按一下呼叫。
- d. 結果會顯示在結果頁框中。
- e. 移至「TCP/IP 監視器」檢視畫面，然後選取最新的訊息。現在，「SOAP 要求與回應」會顯示出來，如下圖所示：

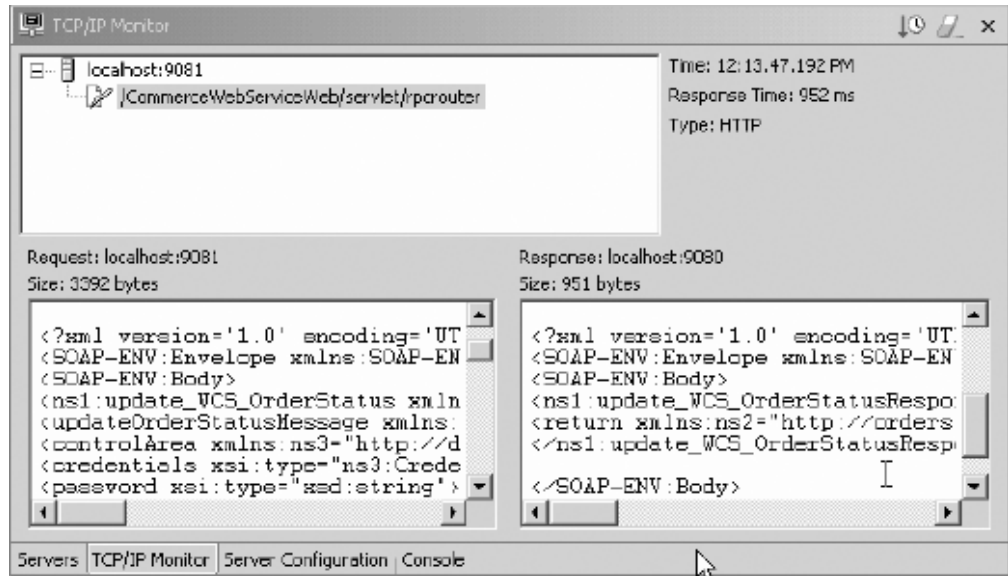


圖 11. 顯示 SOAP 要求與回應

自訂用戶端程式碼

在產生用戶端程式碼時，您必須考量產生的預設用戶端程式碼並不一定能用於 WebSphere Commerce。比方說，如果您使用預設的 BeanSerializer 類別將資料類型序列化，則某些資料類型會轉換成伺服器端的 WebSphere Commerce 可能無法瞭解的 XML 字串。例如，OrderCreate Web 服務範例就包含這類的資料結構。這種資料結構是用來作為 WebSphere Commerce 指令的介面，稱為 UserData 資料結構。UserData 資料結構可容許透過訊息導向介面將自訂資料傳送給指令。

下一節說明如何將 UserData 資料結構序列化，讓 WebSphere Commerce 可以使用它。

註： UserData 資料結構是作為範例使用，這是 OrderCreate Web 服務的選用項目。

WebSphere Commerce 可容許外部系統將自訂的名稱/值配對併入到傳送的 XML 訊息中。為了支援這個功能，所以使用了一個叫做 USERDATA 的 XPATH 類型，會根據以下包含 UserData 元素的範例 XML 訊息中的說明，來處理這個類型：

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

這個 UserData 元素的 XPATH 定義如下：

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

XPATH 處理器會將 UserData 元素轉譯成名稱/值配對 "abc=xyz"。

如果您的資料結構與下面的結構類似，則 Apache SOAP 實作所附的預設 Bean 序列化程式可能會有問題。Apache SOAP 實作會將資料結構中的所有資料類型轉換成 XML 文件中的個別元素。如果您使用 Apache SOAP 用戶端將 Java 物件轉換成 SOAP XML 訊息，則 WebSphere Commerce 將無法剖析這個 XML，因此，就無法組成 UserData 物件。

```

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

```

這個類別會序列化成為 XML 結構。

```

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

```

因此，WebSphere Commerce 訊息對映程式就無法瞭解這是 USERDATA 結構。

為了支援將 UserData 資料類型序列化成為必要的格式，您必須撰寫一個自訂的序列化程式，並將它登錄到 SOAP 對映登錄中。自訂的序列化程式必須具備下列 marshal 方法：

```

/*
 * 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 marshal(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();
}
```

註: WebSphere Commerce 5.5 版提供的服務範例可以和前述的程式碼一起使用。

注意事項

本資訊是針對 IBM 在美國所提供的產品與服務所開發的。而在其他國家中，IBM 並不一定會提供本文件中所提及的各項產品、服務或功能。有關您所在的區域目前可以取得的產品和服務的相關資訊，請向您當地的 IBM 業務代表查詢。凡提及 IBM 產品、程式或服務時，並不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯任何 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以替代使用。不過，其他非 IBM 產品、程式、或服務在運作上的評價與驗證，其責任屬於使用者。

本文件中可能包含 IBM 所擁有之專利或專利申請案。本書使用者並不享有前述專利之任何授權。您可以用書面方式來查詢授權，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

如果要查詢有關雙位元組 (DBCS) 資訊的授權事宜，請聯絡您國家的 IBM 智慧財產部門，或者用書面方式寄到：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

下列段落若與該國之法律條款抵觸，即視為不適用：

IBM 僅以「現狀」提供本書，而不提供任何明示或默示之保證（包括但不限於可售性或符合特定效用的保證）。有些地區在特定的交易中並不容許放棄明示或默示的保證，因此此項聲明不見得適用於您。

本資訊中可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 得隨時改進及（或）變動本書中所說明的產品及（或）程式，而不另行通知。

IBM 得以適當的方式使用或散布由 貴客戶提供的任何資訊，而無需對您負責。

本程式的獲授權者若希望取得程式相關資料，以便使用下列功能者，請洽詢 IBM。其下列資訊是指：（1）在個別建立的程式與其他程式（包括此程式）之間交換資訊（2）相互使用已交換之資訊。若有任何問題請聯絡：

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

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方同意的「IBM 客戶合約」、「IBM 國際程式授權合約」或任何同等合約的條款，提供本文件中所說明的授權程式及其所有適用的授權資料。

本文件可能包含其他公司的產品，包括參考這些公司的網際網路網址。IBM 對這類資訊的正確性、完整性或使用將不負任何責任。

著作權授權：

本書包含範例應用程式的來源語言，用來說明在各種作業平台上的程式設計技術。您不需付費給 IBM，即可使用任何格式來複製、修改或散布這些範例程式，以開發、使用、行銷或散布與用來撰寫範例程式的作業平台的應用程式設計介面相符的應用程式。這些範例並未在所有條件下進行完整測試。因此，IBM 並不保證或暗示這些程式的可靠性、有用性或功能。您不需付費給 IBM，即可使用任何格式來複製、修改及散布這些範例程式，以開發、使用、行銷或散布與 IBM 的應用程式設計介面相符的應用程式。

商標

IBM 與 IBM 標誌是 International Business Machines Corporation 在美國及（或）其他國家或地區的商標或註冊商標。

下列詞彙為 International Business Machines Corporation 在美國及（或）其他國家或地區的商標或註冊商標：

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

Microsoft®、Windows 和 Windows NT® 是 Microsoft Corporation 的商標或註冊商標。

Java 與所有 Java 相關的商標是 Sun Microsystems, Inc. 在美國及（或）其他國家或地區的商標。

其他公司、產品或服務名稱可能是其他者的商標或服務標記。

IBM