

IBM WebSphere Business Integration  
Adapters



# Web 服務配接器使用手冊

V 3.3.x



IBM WebSphere Business Integration  
Adapters



# Web 服務配接器使用手冊

V 3.3.x

**請注意！**

使用此資訊及其支援的產品之前，請參閱第 205 頁的『注意事項』中的資訊。

**2004 年 6 月 25 日**

除非新版中另有指示，本版文件適用於 IBM WebSphere Business Integration Adapter for Web Services (5724-H09) 3.3.x 版。

如果您有關於 IBM CrossWorlds 文件的寶貴意見，請以電子郵件將意見寄到 [doc-comments@us.ibm.com](mailto:doc-comments@us.ibm.com)。我們期待收到您的來信。

當您傳送資訊給 IBM 時，即表示您授與 IBM 非專屬權利，IBM 得以其認為適當的方式使用或散佈該資訊，而無需對您負責。

**© Copyright International Business Machines Corporation 2003. All rights reserved.**

---

## 關於本書

IBM(R) WebSphere(R) Business Integration Adapter 組合可提供領先群倫的電子商務技術及企業應用程式之間的整合網路功能。此文件說明有關 Web 服務配接器的安裝、配置及商業物件開發。

---

## 本書適用對象

本書適用於 IBM WebSphere 客戶、顧問、開發人員，以及負責 Web 服務之 WebSphere Business Integration Adapter 實作的任何人。

---

## 本書的必備條件

在本書中隨處可見各種必備條件的例證。有許多必備條件需參照一些包含 Web 服務相關資訊或資源的網站。您亦應熟悉 WebSphere 商業整合系統的實作方式。您可從 *IBM WebSphere InterChange Server 技術簡介* 這本書開始，該書包含更詳細的文件交互參照。

---

## 相關文件

本產品提供的完整文件集，說明適用於所有 WebSphere 配接器安裝作業的特性和元件，且包含特定元件的參考資料。

您可以從下列網站安裝相關文件：

- 若是一般的配接器資訊：
  - <http://www.ibm.com/websphere/integration/wbiadapters/infocenter>
- 若要將配接器與 InterChange Server 搭配使用：
  - <http://www.ibm.com/websphere/integration/wicserver/infocenter>
  - <http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>

這些網站包含下載、安裝及檢視文件的簡單指引。

**註：**本產品的相關重要資訊，可在發佈本書之後所發出的技術支援 Technotes 及 Flashes 中找到。您可以在 WebSphere Business Integration Support 網站上找到它們，網址是：<http://www.ibm.com/software/integration/websphere/support/>。請選取您有興趣的元件區域，並瀏覽 Technotes 及 Flashes 部份。位於 <http://www.redbooks.ibm.com/> 的 IBM Redbooks 也可能有其他資訊。

---

## 印刷使用慣例

本書採用下列使用慣例：

<code>courier</code> 字型	表示照字面上的值，例如指令名稱、檔案名稱、您鍵入的資訊，或系統列印於畫面上的資訊。
<b>粗體</b>	表示第一次出現的新術語。
<i>斜體</i> 、 <i>斜體</i>	表示變數名稱或交互參照。

---

藍色的輪廓	藍色的輪廓，只有當您在線上檢視手冊時才會出現，表示交互參照的超鏈結。按一下藍色輪廓的內部，即可跳至參照的物件。
{ }	在語法行中，大括弧括住的一組選項，表示您必須僅選擇其中一個選項。
[ ]	在語法行中，方括弧括住一個選用性參數。
...	在語法行中，省略符號表示重複先前的參數。例如， <code>option[,...]</code> 表示您可以輸入多個以逗點區隔的選項。
< >	於命名慣例中，角括弧括住一個名稱的個別元素來做區別，例如 <code>&lt;server_name&gt;&lt;connector_name&gt;tmp.log</code> 。
/, \	於本書中，反斜線 (\) 當做目錄路徑的使用慣例。在 UNIX 安裝方面，反斜線取代成斜線 (/)。所有 IBM 產品路徑名稱皆相對於您系統上安裝產品的目錄。
%text% 和 \$text	百分比 (%) 符號內的文字表示 Windows <code>text</code> 系統變數或使用者變數的值。UNIX 環境中同等的表示法為 <code>\$text</code> ，表示 <code>text</code> UNIX 環境變數的值。
<i>ProductDir</i>	代表 IBM WebSphere Business Integration Adapters 產品的安裝目錄。CROSSWORLDS 環境變數包含 <code>ProductDir</code> 目錄路徑，預設值為 <code>IBM\WebSphereAdapters</code> 。
”	指出功能表中的選項，例如：選擇檔案 ” 更新 ” SGML 參照

---

# 目錄

關於本書	iii
本書適用對象	iii
本書的必備條件	iii
相關文件	iii
印刷使用慣例	iii
本版的新功能	vii
3.3.x 版的新功能	vii
3.2.x 版的新功能	vii
3.1.x 版的新功能	viii
3.0.x 版的新功能	viii
第 1 章 連接器概觀	1
Web 服務配接器環境	1
專有名詞	4
Web 服務的連接器元件	6
Web 服務的連接器架構	10
安裝、配置及設計核對清單	11
限制	12
第 2 章 安裝及啟動	13
安裝作業概觀	13
安裝連接器及相關檔案	13
已安裝檔案的結構	13
配置作業概觀	15
執行配接器的多重實例	16
啟動及停止連接器	17
第 3 章 商業物件基本需求	19
商業物件 Meta 資料	19
連接器商業物件結構	19
開發商業物件	49
第 4 章 Web 服務連接器	51
連接器處理程序	51
SOAP/HTTP(S) Web 服務	53
SOAP/JMS Web 服務	54
事件處理程序	55
要求處理程序	65
連接器及 JMS	72
SSL	74
配置連接器	76
連接器啟動時	93
記載	94
追蹤	94
第 5 章 SOAP 資料處理常式	97
配置 SOAP 資料處理常式	97
SOAP 資料處理常式處理程序	103
使用應用程式特有資訊功能	108
指定可插入的名稱處理常式	125

限制	127
第 6 章 啓用要求處理程序的協同作業	129
要求處理程序協同作業核對清單	129
第 7 章 將協同作業顯現為 Web 服務	131
程序核對清單	131
識別或開發商業物件	132
選擇或開發協同作業範本	132
連結新協同作業物件的埠	132
WSDL 配置精靈	134
第 8 章 使用 WSDL ODA	141
啓動 WSDL ODA	141
執行 WSDL ODA	142
配置代理程式	142
指定 WSDL 文件	144
確認選項	146
產生物件	146
限制	147
第 9 章 疑難排解	149
啓動問題	149
執行時期錯誤	150
附錄 A. 連接器的標準配置內容	153
新的及已刪除的內容	153
配置標準連接器內容	153
標準內容的摘要	154
標準配置內容	159
附錄 B. 連接器配置程式	171
連接器配置程式概觀	171
啓動連接器配置程式	172
從系統管理程式執行配置程式	172
建立連接器特有的內容範本	173
建立新配置檔	175
使用現存檔	176
完成配置檔	177
設定配置檔內容	177
儲存配置檔	182
變更配置檔	183
完成配置	183
在全球化環境下使用連接器配置程式	183
附錄 C. Web 服務配接器指導教學	185
關於指導教學	185
開始之前	186
安裝及配置	186
執行非同步範例實務	192
執行同步範例實務	194

<b>附錄 D. 移轉至 3.0.x</b>	<b>199</b>
反向相容性	199
升級作業	199
<b>附錄 E. 配置 HTTPS/SSL</b>	<b>203</b>
Keystore 設定	203

TrustStore 設定	204
產生公開金鑰憑證的憑證簽章要求 (CSR)	204
<b>注意事項</b>	<b>205</b>
程式設計介面資訊	206
商標與服務標記	206



---

## 本版的新功能

---

### 3.3.x 版的新功能

本版包括下列加強功能：

- 與 2003 年 8 月發行的 WS-I Basic Profile 1.0 規格相符的 Web 服務配接器 (連接器、WSDL ODA 及 SOAP 資料處理常式)。
- 支援 SOAP 1.2 版；新的 SOAP 資料處理常式 Meta 物件內容 (請參閱第 98 頁的『MO\_DataHandler\_DefaultSOAPConfig』) 及新的 WSDL ODA 內容 (請參閱第 143 頁的表 47)，可讓您指定 SOAP 1.2 或 1.1。
- WSDL ODA 可讓您根據相同綱目重複使用商業物件名稱，以減少部署所需的成品總數。如需進一步資訊，請參閱第 143 頁的表 47。
- WSDL ODA 及 SOAP 資料處理常式可支援 sequence、choice 及 group 上的 maxOccurs 所有可能值。如需進一步資訊，請參閱第 121 頁的『sequence、choice、group 及 all 上的 maxOccurs 指示元』。
- 如果您尚未指定 java.protocol.handler.pkgs 的值，連接器就會在起始設定期間使用預設值。如需進一步資訊，請參閱第 74 頁的『JSSE』。
- HTTP 通訊協定接聽器可支援具有任何「接受」標頭值的要求；如有必要，可將標頭驗證委派到協同作業。
- 連接器特有內容 WorkerThreadCount 的最小值已經變更。如需進一步資訊，請參閱第 80 頁的『WorkerThreadCount』。
- 配接器可支援 SOAP/JMS 接聽器及處理常式中，入埠及離埠訊息的 TextMessage 及 BytesMessage 內容類型。
- 若由 SOAP/HTTP(S) 接聽器進行同步事件處理，則如果回應不是由協同作業移入，回應的 Content-Type HTTP 標頭的 ContentType 部份就會設定為要求的 ContentType。

Solaris 7 不支援配接器 3.3 版，所以本手冊已經刪除對於該平台版本的參照。

---

### 3.2.x 版的新功能

本版包括下列加強功能：

- 支援所有接聽器及通訊協定處理常式中的自訂內容
- 支援傳送及讀取 SOAP/HTTP(S) 接聽器和處理常式中的標準標頭
- 支援傳送及讀取 SOAP/JMS 接聽器和處理常式中的主要 JMS 標準標頭。
- 支援自訂 SOAP/HTTP 和 SOAP/HTTPS 接聽器中非同步 HTTP 處理程序的成功回覆碼
- 支援在 SOAP/HTTP-HTTPS 處理常式中執行非同步 HTTP 處理程序時，有多重可接受的成功回覆碼
- 支援 SOAP/JMS 接聽器中的「位元組訊息」。
- 支援 SOAP/HTTP 和 SOAP/HTTPS 接聽器中，MIME 類型及訊息 Charset 的自訂作業 (依據要求 ContentType 及要求 URL)

- 支援 SOAP/HTTP-HTTPS 處理常式中的執行時期 MIME 類型及訊息 CharSet 的自訂作業 (依據回應 ContentType)
- 可傳送及接收 SOAP/HTTP 和 SOAP/HTTPS 接聽器以及 SOAP/HTTP-HTTPS 處理常式中的任何 ContentType 訊息
- 可在 SOAP/HTTP 和 SOAP/HTTPS 接聽器及 SOAP/HTTP-HTTPS 處理常式中，於執行時期指定外送訊息的 MIME 類型和 CharSet

從 3.2.x 版開始，Web 服務配接器不會在 Microsoft Windows NT 上執行。

已經加強 WSDL ODA：它現在可以讀取介面檔以產生事件處理 TLO。

配接器安裝資訊已從本手冊中移去。如需進一步資訊，請參閱第 13 頁的『安裝連接器及相關檔案』。

---

### 3.1.x 版的新功能

已加強 Web 服務連接器的功能。新的加強功能及元件如下：

- 支援 HTTP 認證傳送。如需進一步資訊，請參閱第 29 頁的『事件處理程序的 HTTP 認證傳送』、第 44 頁的『要求處理程序的 HTTP 認證傳送』及第 86 頁的『ProxyServer』。
- WSDL ODA 及 SOAP 資料處理常式可支援下列 Style/Use 內容：rpc/literal、rpc/encoded 及 document/literal。如需進一步資訊，請參閱第 100 頁的『Style 及 Use 對 SOAP 訊息的影響』。
- 「WSDL 配置精靈」可支援 SOAP Config MO 中的 Use 屬性。如需 Use 屬性的相關資訊，請參閱第 99 頁的『SOAP 配置 meta 物件：每個 SOAP 商業物件的子項』。「WSDL 配置精靈」亦支援部分元素的國際化字元。如果需要詳細資訊，請參閱第 138 頁的『基本需求及例外的處理』。
- JMS 通訊協定接聽器有一個新的配置內容：SessionPoolSize。如需進一步資訊，請參閱第 83 頁的表 41。
- 連接器可在下列平台上執行：
  - Microsoft Windows NT 4.0 Service Pack 6A 或 Windows 2000
  - Solaris 7、8 或 AIX 5.1、5.2 或 HP UX 11.i

---

### 3.0.x 版的新功能

已重新設計 Web 服務連接器的功能。新的加強功能及元件如下：

- Web 服務連接器 (取代舊版的 SOAP 連接器及 Proxy 類別)，可雙向支援 SOAP/HTTP、SOAP/HTTPS 及 SOAP/JMS Web 服務
- 加強型 WebSphere Business Integration Toolset 功能，可將協同作業顯現成 Web 服務
- Web 服務說明語言 (WSDL) 物件探查代理程式 (ODA)
- 新的商業物件結構，會使用應用程式特有資訊 (ASI) 功能
- 新的 SOAP 資料處理常式

第 1 章包含新配接器的概觀。如需有關與舊版產品反向相容的資訊，請參閱第 199 頁的附錄 D，『移轉至 3.0.x』。

不再使用“CrossWorlds”名稱來說明整個系統，或修改元件或工具的名稱，除此之外大致和以前相同。例如，“CrossWorlds System Manager”現在變為“System Manager”（系統管理程式），而“CrossWorlds InterChange Server”現在變為“WebSphere InterChange Server”。

連接器已國際化。如需詳細資訊，請參閱第 3 頁的『語言環境相依資料』及第 153 頁的附錄 A，『連接器的標準配置內容』。

本手冊提供有關搭配 WebSphere InterChange Server (ICS) 來使用此配接器的資訊。



---

## 第 1 章 連接器概觀

- 『Web 服務配接器環境』
- 第 4 頁的『專有名詞』
- 第 6 頁的『Web 服務的連接器元件』
- 第 10 頁的『Web 服務的連接器架構』
- 第 11 頁的『安裝、配置及設計核對清單』
- 第 12 頁的『限制』

連接器為 WebSphere Business Integration Adapter for Web Services 的執行時期元件。連接器可使企業收集、發佈及運用 Web 服務，以供其組織內部或貿易夥伴使用。本書中所介紹的連接器和其他元件，提供在「簡式物件存取通訊協定 (SOAP)」訊息主體中，交換商業物件資訊時，所需的功能。

本章說明在實行 WebSphere Business Integration Adapter for Web 服務時，所使用的範圍、元件、設計工具及架構。它亦提供您在安裝及配置本書中所介紹的 Web 服務元件時，必須完成之作業的概觀。如需有關安裝及配置元件的資訊，請參閱第 11 頁的『安裝、配置及設計核對清單』。

**註：** Web 服務配接器會實行標準的「配接器組織架構 API」。因此，配接器可以搭配「組織架構」所支援的任何整合分配管理系統來操作。但是配接器所提供的功能是為支援 IBM WebSphere InterChange Server (ICS) 整合分配管理系統而特別設計的。所以當您選取「系統管理程式」中的「顯現成 Web 服務」選項時，這是指 ICS，而不是指任何其他的整合分配管理系統。

---

### Web 服務配接器環境

在安裝、配置及使用配接器之前，您必須先了解它的環境需求：

- 『分配管理系統相容性』
- 第 2 頁的『軟體需求』
- 第 2 頁的『配接器平台』
- 第 2 頁的『標準及 API』
- 第 3 頁的『語言環境相依資料』

#### 分配管理系統相容性

配接器使用的配接器組織架構必須與所通訊的整合分配管理系統 (或分配管理系統) 版本相容。下列配接器組織架構及整合分配管理系統可支援 Web 服務配接器 3.3 版：

- 配接器組織架構：WebSphere Business Integration Adapter Framework，版本：
  - 2.2.0
  - 2.3.0
  - 2.3.1
  - 2.4.0
- 僅限 WebSphere InterChange Server，版本：

- 4.2
- 4.2.1
- 4.2.2

請參閱「版本注意事項」，查看是否有任何例外狀況。

**註：**若要利用 3.2 版配接器所記載的所有功能，您必須安裝 WebSphere InterChange Server 4.2.2 版。

## 軟體需求

請先檢視下列假設及軟體需求，再安裝 Web 服務連接器：

- 連接器及其他元件的設計，是依據針對 SOAP 1.1 及 1.2 來公佈的規格。
- 若您正使用 SOAP/JMS Web 服務，則必須安裝您自己的 JMS 及 JNDI 實作方式。
- 若您正使用 HTTPS/SSL，則需要您自己的協力廠商軟體，來建立金鑰儲存庫及信任儲存庫。

## 配接器平台

配接器可在下列平台 (作業系統) 上執行：

- Microsoft Windows 2000
- Solaris 8
- AIX 5.1、5.2
- HP-UX 11i

## 標準及 API

與 2003 年 8 月發行的 WS-I Basic Profile 1.0 規格相符的 Web 服務配接器 (連接器、WSDL ODA 及 SOAP 資料處理常式)。

各種的標準及技術讓 Web 服務能夠透過網路來存取其功能。

配接器使用的標準如下：

- SOAP 1.2 及 1.1 版
- WSDL 1.1 SOAP 連結
- HTTP 1.0
- JMS 1.0.2

配接器使用的 API 如下：

- Apache SOAP 2.3.1 API：連接器從 Apache Foundation 納入了 SOAP API。Apache SOAP API 是 SOAP 1.1 版的開放原始碼實作方式。Apache SOAP API 的基本需求如下：
  - Java Activation Framework 1.0.1 (activation.jar)
  - JavaMail(TM) API 1.2 (mail.jar)
- Xerces Java 剖析器 1.4.3 及以上版本，Xerces2 是完全符合的 XML 綱目處理器
- JMS API 1.0.2 版
- WSDL4J 1.2.1 - Web Service Description Language for Java API (WSDL4J) 提供 WSDL 文件的物件模型

- UDDI4J-WSDL 2.1.0 - UDDI4J-WSDL API 封裝存在於 UDDI4J API 中的類別，以及一些由 WSDL4J API 所定義的類別
- JNDI 1.2.1
- WSDL4J 1.0
- IBM JSSE 1.0.2

根據您的配置，您可能需要安裝其他軟體。下面各節將會討論這些情況。

## JMS 通訊協定

若您正使用 JMS 通訊協定，則必須安裝 JMS 提供者，並建立佇列。至於要建立哪些佇列，就要視您的需求而定了。您可使用 JMS 通訊協定將協同作業顯現成 Web 服務，亦可使用它來呼叫外部 Web 服務。如需進一步資訊，請參閱第 72 頁的『連接器及 JMS』。

**JNDI:** 您必須配置 JNDI，然後在連接器的 JNDI 配置內容中輸入適當的參數。您也必須確定，在 JNDI 上，Connection Factory 及 JMS 目的地 (佇列) 物件都是可用的。若您要使用 JNDI，但沒有實行 JNDI，您可以從 Sun Microsystems 來下載 File System JNDI 的參照實作方式。如需進一步資訊，請參閱第 72 頁的『連接器及 JMS』。

## SSL

若您計劃要使用 SSL，則必須使用協力廠商的軟體來管理您的金鑰儲存庫、憑證及金鑰產生作業。我們並未提供任何工具來設定金鑰儲存庫、憑證或金鑰產生作業。您可以選擇使用 Keytool (隨附於 IBM JRE) 來建立自行簽章的憑證以及管理金鑰儲存庫。如需進一步資訊，請參閱第 74 頁的『SSL』。

## 語言環境相依資料

連接器已全球化，可以支援雙位元組字集。當連接器將資料從使用某種字元碼的地方，轉送到使用不同字碼集的地方時，它會執行字元轉換作業，保留該資料的意義。

「Java 虛擬機器 (JVM)」中的 Java 執行時期環境，代表 Unicode 字碼集的資料。Unicode 包含大部分已知字碼集的字元編碼 (包括單位元組及多位元組)。WebSphere Business Integration 系統中的大部分元件都是用 Java 來撰寫的。因此，當資料在大部分的整合元件之間轉送時，不需要轉換字元。

**註:** 連接器尚未國際化。這表示追蹤和日誌訊息都沒有翻譯。

## Web 服務連接器

本節討論區域化及連接器。

**事件通知:** 連接器使用可插入的通訊協定接聽器來執行事件通知。通訊協定接聽器會從傳輸埠擷取 SOAP 訊息，並呼叫 SOAP 資料處理常式。本節將說明各接聽器如何透過傳輸埠以將 SOAP 訊息編碼。

- **SOAP/HTTP 及 SOAP/HTTPS 接聽器** 這二種接聽器會以位元組為單位來讀取 HTTP 要求訊息的主體。主體的編碼是由 HTTP Content-Type 標頭的 charset 參數所指定。若 charset 參數遺失，就會採用 ISO-8859-1 (ISO Latin 1)。接聽器會使用此編碼將要求訊息的主體轉換成 Java 字串。此 Java 字串可用來呼叫 SOAP 資料處理常式。若為同步的 (要求與回應) Web 服務，會使用協同作業所傳回商業物件來呼叫 SOAP 資料處理常式。然後會使用 HTTP 要求訊息的編碼來將 SOAP 資料處理常式傳回的 Java 字串會轉換成位元組。

- **SOAP/JMS 接聽器**此接聽器可支援 JMS 文字訊息以及 JMS 位元組訊息。

**要求處理程序：** 連接器使用可插入的通訊協定處理常式來執行要求處理程序。通訊協定處理常式呼叫 SOAP 資料處理常式。本節將說明各處理常式如何透過傳輸埠來將 SOAP 訊息編碼。

- **SOAP/HTTP-HTTPS 處理常式**這些處理常式會呼叫 SOAP 資料處理常式。為了編製 Web 服務要求，會使用 UTF 8 編碼將資料處理常式傳回的字串轉換成位元組。若為同步的 (要求與回應) Web 服務，通訊協定處理常式會讀取 HTTP 回應訊息的主體。主體的編碼是由 HTTP Content-Type 標頭的 charset 參數所指定。若 charset 參數遺失，就會採用 ISO-8859-1。處理常式會使用此編碼將回應訊息的主體轉換成 Java 字串。將會使用此字串來呼叫 SOAP 資料處理常式。
- **SOAP/JMS 處理常式**此接聽器可支援 JMS 文字訊息。

## SOAP 資料處理常式

本節討論區域化及 SOAP 資料處理常式。

**SOAP 字元限制：** XML 元素名稱及屬性名稱必須是商業物件名稱、商業物件屬性名稱或商業物件應用程式特有資訊所允許的合法的 ascii 字元。

商業物件名稱或商業物件屬性名稱不支援國際化的字元。只有屬性值可以國際化。

**SOAP 資料處理常式處理程序：** 將 SOAP 訊息轉換成商業物件時，資料處理常式只能接收字串。資料處理常式只會將字串值移入商業物件，並傳回商業物件。Java 字串是 UCS2，所以可轉送雙位元組的字元，沒有問題。只有 XML 元素及屬性值可為非 ascii 字元 (請參閱字元限制)。將商業物件轉換成 SOAP 訊息時，資料處理常式會使用 Xerces 剖析器將商業物件轉換成字串。Java 字串是 UCS2，所以可轉送雙位元組的字元，沒有問題。只有 XML 元素及屬性值可為非 ascii 字元 (請參閱字元限制)。

## WSDL ODA

本節討論區域化及 WSDL ODA。

除了 WSDL 檔案中的合法 ASCII 字元之外，WSDL ODA 不支援其他字元。WSDL ODA 可以支援其他字集的字元名稱及 URL。但是這些檔案的內容必須是合法的 ASCII。

WSDL ODA 的「配置代理程式」表格中的內容已全球化，如下所示：

- **WSDL\_URL** URL 可為當地的語言
- **UDDI\_InquiryAPI\_URL** 檢查 UDDI 登錄支援
- **WebServiceProvider** 僅限合法的 ASCII 字元
- **WebService** 僅限合法的 ASCII 字元
- **MimeType** 僅限合法的 ASCII 字元
- **BOPrefix** 僅限合法的 ASCII 字元
- **BOVerb** 僅限合法的 ASCII 字元
- **Collaboration** 僅限合法的 ASCII 字元

---

## 專有名詞

本手冊使用的詞彙如下：



- **ASI (Application-Specific Information - 應用程式特有資訊)** 是為適應特定應用程式或技術而設計的程式碼。ASI 存在於商業物件定義的屬性層次及商業物件層次中。
- **ASBO (Application-Specific Business Object - 應用程式特有商業物件)** 可以包含 ASI 的商業物件。
- **BO (Business Object - 商業物件)** 一組屬性，代表商業實體 (例如：客戶) 以及對資料做的動作 (例如：建立或更新作業)。IBM WebSphere 系統的元件會使用商業物件來交換資訊及觸發動作。
- **Content-Type** HTTP 通訊協定標頭，它包含類型/子類型及選用性參數。例如，在 Content-Type 值 `text/xml; charset=ISO-8859-1` 中，`text/xml` 為類型/子類型，而 `charset=ISO-8859-1` 則為選用性 `Charset` 參數。
- **ContentType** 僅指 Content-Type 標頭值的類型/子類型部分。例如，在 Content-Type 值 `text/xml; charset=ISO-8859-1` 中，本書的 `ContentType` 就是 `text/xml`。
- **MO\_DataHandler\_DefaultSOAPConfig** SOAP 資料處理常式專屬的子項資料處理常式 Meta 物件。
- **GBO (Generic Business Object - 通用商業物件)** 不含 ASI，且不受任何應用程式限制的商業物件。
- **MO\_DataHandler\_Default** 資料處理常式 Meta 物件，可供連接器代理程式用來決定要將哪個資料處理常式實例化。此值指定於連接器的 `DataHandlerMetaObjectName` 配置內容中。
- **Non-Top Level Business Object (Non-TLO - 非最上層商業物件)** 非 TLO 是不屬於 Web 服務 TLO 結構的任何商業物件。
- **Protocol Config MO** 在要求處理程序期間，SOAP/JMS、SOAP/HTTP-HTTPS 通訊協定處理常式會使用 Protocol Config MO 來決定目標 Web 服務的目的地。若您於處理事件期間，將協同作業顯現成 SOAP/JMS Web 服務，連接器就會使用 Protocol Config MO，將 JMS 訊息標頭資訊從 SOAP/JMS 通訊協定接聽器傳送到協同作業。
- **SOAP (Simple Object Access Protocol - 簡式物件存取通訊協定)** 定義了一個模型，可將簡式要求及回應訊息 (以 XML 撰寫) 當作電子通訊的基本通訊協定來使用。SOAP 是傳訊平台中立的遠端程序呼叫 (RPC) 機制，但可用來交換任何種類的 XML 資訊 (文件交換)。
- **SOAP Business Object (SOAP 商業物件)** SOAP 商業物件是 TLO 的子項，可為「SOAP 要求」、「SOAP 回應」或「SOAP 錯誤」商業物件。SOAP 商業物件包含 SOAP 資料處理常式在執行處理時所需的資訊，包括 SOAP ConfigMO (SOAP 商業物件的子項)，亦包含 SOAP 標頭儲存區商業物件。
- **SOAP Config MO (Configuration Meta Object - 配置 Meta 物件)** 資料處理常式需要一個包含單一轉換相關配置資訊的物件，例如從 SOAP 訊息轉換成 SOAP 商業物件。此資訊是以 Meta 資料的形式，儲存在 SOAP 商業物件的子項中。這個子物件就是 SOAP Config MO。
- **SOAP Header Child Business Object (SOAP 標頭子項商業物件)** 一種商業物件，代表 SOAP 訊息中的單一標頭元素。標頭元素是 SOAP 訊息之 `SOAP-Env:Header` 元素的直接子項。標頭儲存區商業物件的所有屬性都必須是這個類型。這些商業物件可能有 `actor` 及 `mustUnderstand` 屬性。這兩個屬性對應於 SOAP 標頭元素的 `actor` 及 `mustUnderstand` 屬性。
- **SOAP Header Container Business Object (SOAP 標頭儲存區商業物件)** 一種商業物件，包含 SOAP 訊息中之標頭的相關資訊。此商業物件包含一或數個子項商業物件。每個子項商業物件各代表 SOAP 訊息中的一個標頭項目。SOAP 資料處理常

式商業物件可能會有一個類型為 SOAP 標頭儲存區商業物件的屬性。此屬性亦稱為 SOAP 標頭屬性。這種屬性具有特殊的應用程式特有資訊需求，說明請見第 97 頁的第 5 章, 『SOAP 資料處理常式』。此屬性必須是 SOAP 商業物件的直接子項。

- **Top-Level Business Object (最上層商業物件)** Web 服務最上層商業物件包含「SOAP 要求」、「SOAP 回應」(選用性) 及一或數個「SOAP 錯誤」(選用性) 商業物件。連接器會將 TLO 使用於事件處理程序及要求處理程序。
- **Web 服務)** 是獨立自足、模組化、分散式、動態的應用程式，可透過網路來描述、發佈、尋找或呼叫，以建立產品、處理程序及供應鏈。其可為本端、分散式或 Web 型。Web 服務建立於開放式標準的頂端，例如 TCP/IP、HTTP、Java、HTML 及 XML。Web 服務使用新的標準技術 (例如 SOAP - 簡式物件存取通訊協定) 來進行傳訊，並使用 UDDI (廣泛描述、探索與整合) 及 WSDL (Web 服務說明語言) 來進行發佈。
- **UDDI (Universal Description, Discovery and Integration - 廣泛描述、探索與整合)** 是一種規格，其定義一種方法，可用來發佈及探索 Web 服務的相關資訊。UDDI 規格提供 XML 型的介面 (API)，可以程式化的方式來存取 UDDI 登錄資訊。SOAP 是這些 API 的基礎 RPC 機制。
- **WSDL (Web Services Description Language - Web 服務說明語言)** 是一種 XML 詞彙，用以定義 Web 服務的軟體介面。它將於程式設計層次執行自動整合時所需的所有 Web 服務技術詳細資料組織化，並可用來將 IBM WebSphere 協同作業當成 Web 服務來發佈。WSDL 之於 Web 服務，如同 IDL 之 CORBA 物件。

如需 WSDL 的詳細資訊，請參訪下列網址：

<http://www.w3.org/TR/wsdl>

---

## Web 服務的連接器元件

圖 1 說明 Web 服務的連接器，包括其通訊協定處理常式和接聽器組織架構，以及 SOAP 資料處理常式。

註：「Web 服務配接器」的 XML 資料處理常式使用授權是有限制的。然而，配接器不需要 XML 資料處理常式即可運作。

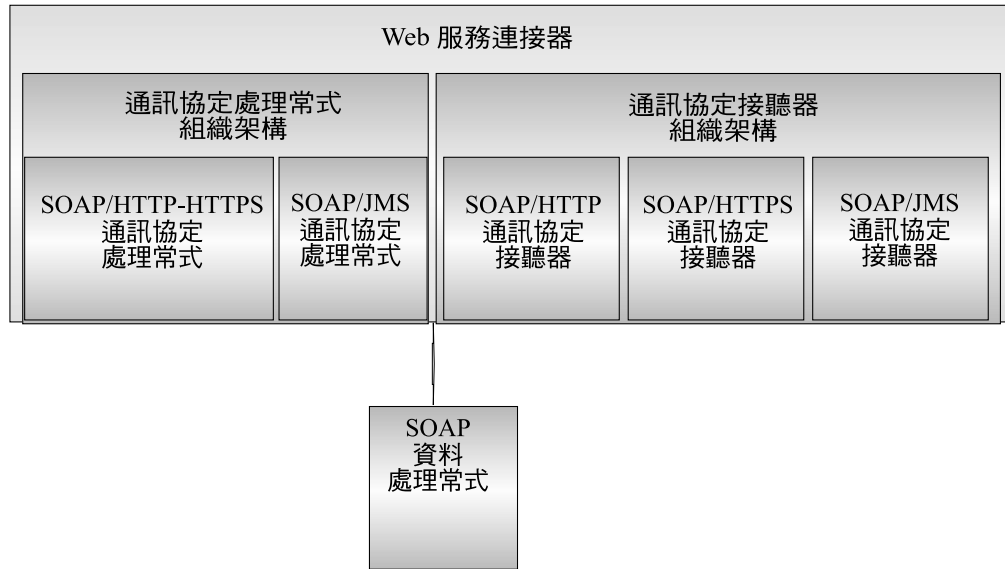


圖 1. Web 服務的連接器

下列元件會交互作用，以讓您透過網際網路來進行資料交換：

- Web 服務連接器，包括 SOAP 資料處理常式，以及通訊協定接聽器和處理常式
- 具 Web 服務功能的協同作業
- 商業物件及 SOAP 訊息
- WebSphere Business Integration InterChange Server

## Web 服務連接器

於要求處理程序期間，Web 服務連接器會將商業物件轉換成 SOAP 要求訊息，並將其傳送到目的地 Web 服務，以回應協同作業服務呼叫。然後 (選擇性作業，針對同步要求處理程序) 連接器將 SOAP 回應訊息轉換成「SOAP 回應」商業物件，並將這些物件傳回協同作業。

於事件處理程序期間，連接器會處理來自用戶端 Web 服務的 SOAP 要求訊息，它會將 SOAP 要求訊息轉換成「SOAP 要求」商業物件，並將其傳遞至協同作業 (已顯現成 Web 服務)，以進行處理。然後連接器會選擇性地接收來自協同作業的「SOAP 回應」商業物件 (轉換成 SOAP 回應訊息之後，再傳回用戶端 Web 服務)。

如需進一步資訊，請參閱第 51 頁的第 4 章，『Web 服務連接器』

註：在本書中，除非另有指示，否則所提到的任何連接器，都是指 Web 服務連接器。

### 通訊協定接聽器及處理常式

連接器包含下列通訊協定接聽器及處理常式：

- SOAP/HTTP 通訊協定接聽器
- SOAP/HTTPS 通訊協定接聽器
- SOAP/JMS 通訊協定接聽器
- SOAP/HTTP-HTTPS 通訊協定處理常式
- SOAP/JMS 通訊協定處理常式

通訊協定接聽器會以 SOAP/HTTP、SOAP/HTTPS 或 SOAP/JMS 格式，偵測內部或外部 Web 服務用戶端的事件。它會通知連接器，有事件需要由顯現成 Web 服務的協同作業來處理。然後通訊協定接聽器會讀取商業物件層次和屬性層次 ASI、連接器內容，以及內含在通訊協定配置物件中的轉換規則，以決定協同作業、資料處理常式、處理程序模式 (同步/非同步)，以及 Web 服務交易的傳輸埠特有層面。如需查詢通訊協定接聽器處理程序的詳細帳號，請參閱第 55 頁的『通訊協定接聽器』。

通訊協定處理常式會以 SOAP/HTTP、SOAP/HTTPS 或 SOAP/JMS 格式，代表協同作業來呼叫 Web 服務。通訊協定處理常式會讀取 TLO ASI 及通訊協定配置物件中所內含的轉換規則，來決定如何處理要求 (同步或非同步)、要用哪個資料處理常式來將 SOAP 訊息轉換成 SOAP 商業物件 (反之亦然)，並決定 Web 服務的目標位址 (從 SOAP 要求商業物件 Protocol Config MO 的 Destination 屬性來決定)。若為同步交易，通訊協定處理常式就會處理 SOAP 回應訊息、將其轉換成「SOAP 回應」商業物件，並將其傳回協同作業。

如需通訊協定處理常式的進一步資訊，請參閱第 65 頁的『通訊協定處理常式』。

## SOAP 資料處理常式

SOAP 資料處理常式會將 SOAP 商業物件轉換成 SOAP 訊息，反之亦然。如需 SOAP 資料處理常式的進一步資訊，請參閱第 97 頁的第 5 章，『SOAP 資料處理常式』。

如需進一步資訊，請參閱第 97 頁的第 5 章，『SOAP 資料處理常式』。

## Web 服務配置工具

您可使用呼叫 Web 服務或是顯現成 Web 服務的協同作業來部署 Web 服務解決方案。

當您啟用協同作業來進行要求處理程序時，請使用 WSDL Object Discovery Agent (ODA) 來產生 Web 服務 TLO。如需有關要求處理程序及 WSDL ODA 的進一步資訊，請參閱第 129 頁的第 6 章，『啟用要求處理程序的協同作業』。

當您將協同作業顯現成 Web 服務時，請使用「WSDL 配置精靈」，它可以協助您產生協同作業的 WSDL 文件，以供稍後透過 UDDI 登錄 (舉例說明) 來發佈。連接器沒有提供工具來發佈此資訊。如需有關將協同作業顯現成 Web 服務的資訊，請參閱第 131 頁的第 7 章，『將協同作業顯現為 Web 服務』。

## 部署連接器

部署 Web 服務連接器的方法有兩種：

- 企業的商業程序若是以 SOAP/HTTP、SOAP/HTTPS 或 SOAP/JMS Web 服務格式來進行通訊，則部署在防火牆後方，做為企業內部網路解決方案 (請參閱圖 2)。

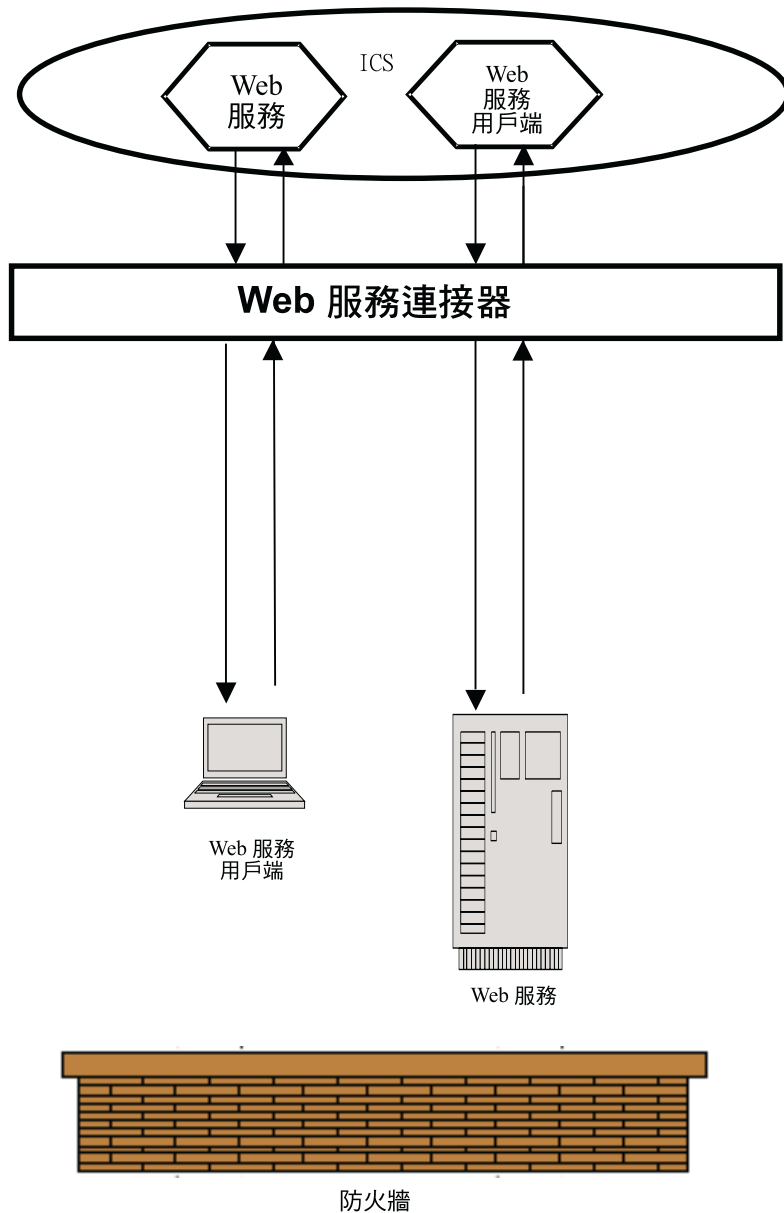


圖 2. Web 服務配接器作為企業內部網路解決方案

- 部署在防火牆後方，以前端系統或閘道伺服器來處理、過濾，要不然就是用企業外部的 Web 服務來管理通訊作業。

註: Web 服務連接器並沒有包含閘道或前端系統，無法管理與外部 Web 服務之間的內送及外送訊息。您必須配置及部署您自己的閘道。**連接器只能部署在企業內部，不能部署在 DMZ 中或是防火牆外面。**

## Web 服務的連接器架構

為了說明高層次元件的架構，本節介紹了兩個資料流程。圖 3 說明兩種範例實務。這兩種範例實務的說明如下。

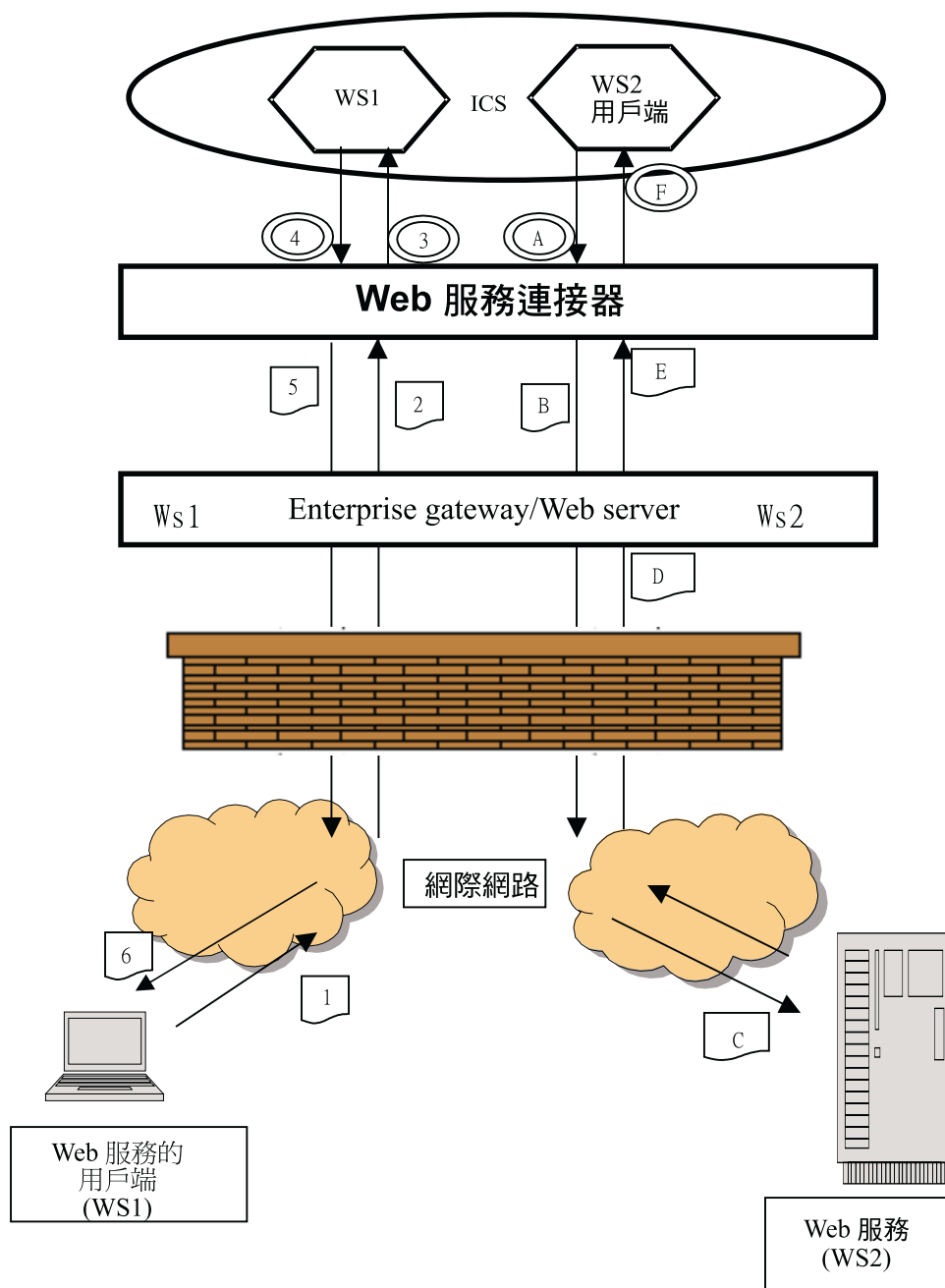


圖 3. Web 服務訊息的流程

**要求處理程序**說明當協同作業對連接器提出服務呼叫要求，以呼叫 Web 服務時，整個事件發展的過程。於此範例實務中，協同作業是扮演用戶端的角色，將要求傳送至伺服器。

- A 協同作業傳送服務呼叫要求給連接器，呼叫 SOAP 資料處理常式來將商業物件轉換成 SOAP 要求訊息。
- B 連接器傳送 SOAP 訊息，以呼叫 Web 服務 WS2。如果目的地是外部 Web 服務，連接器會將 SOAP 訊息傳送至閘道。閘道再將 SOAP 訊息傳送到對應於目的地 Web 服務的端點。如此便會呼叫 Web 服務。
- C 所呼叫的 Web 服務接收 SOAP 要求訊息，並執行所要求的處理程序。
- D 所呼叫的 Web 服務傳送一則 SOAP 回應 (或錯誤) 訊息。如果 Web 服務是在企業外部，則閘道會接收並遞送 SOAP 回應訊息。
- E SOAP 回應 (或錯誤) 訊息遞送回連接器，呼叫資料處理常式來將其轉換成回應或錯誤商業物件。
- F 連接器將 SOAP 回應或錯誤商業物件傳回協同作業。

**事件處理程序**說明將協同作業當作 Web 服務來呼叫時，整個事件發展的過程。在此範例實務中，協同作業 (顯現成 Web 服務) 是扮演伺服器的角色，它要接受來自用戶端 (外部或內部) 的要求，並依需要來回應。

- 1 用戶端 Web 服務 (WS1) 將 SOAP 要求訊息傳送至目的地 (針對協同作業來產生之 WSDL 文件中所指定的目的地)。
- 2 如果用戶端 Web 服務是外部的，閘道會接收訊息，並遞送至連接器。
- 3 連接器將 SOAP 訊息傳送到 SOAP 資料處理常式，以將 SOAP 訊息轉換成商業物件。連接器呼叫顯現成 Web 服務的協同作業。
- 4 協同作業傳回「SOAP 回應」(或「錯誤」) 商業物件。
- 5 連接器呼叫 SOAP 資料處理常式來將「SOAP 回應」(或「錯誤」) 商業物件轉換成 SOAP 回應訊息。連接器將回應傳回至閘道。
- 6 如果用戶端 Web 服務是外部的，閘道會就將 SOAP 回應訊息遞送至用戶端 Web 服務 (WS1)。

---

## 安裝、配置及設計核對清單

本節彙總了您在安裝、配置及設計 Web 服務解決方案時，必須執行的作業。每一節都會先簡單介紹該項作業，然後提供通往本書中其他小節的鏈結 (及相關文件的交互參照)，以說明如何執行作業，或是提供背景資訊。

### 安裝配接器

請參閱第 13 頁的第 2 章, 『安裝及啟動』, 以取得您必須安裝之項目和位置的說明。

### 配置連接器內容

連接器有兩種配置內容類型：標準配置內容和特定連接器配置內容。這些內容中有些具有預設值，您不需要變更。在執行連接器之前，您可能需要先設定其中一部份內容的值。如需詳細資訊，請參閱第 51 頁的第 4 章, 『Web 服務連接器』。

### 配置通訊協定處理常式及接聽器

當您指定連接器配置內容的值，來控管這些元件的行為時，就要配置通訊協定處理常式及接聽器。如需詳細資訊，請參閱第 51 頁的第 4 章, 『Web 服務連接器』。

## 啓用 Web 服務的協同作業

當您啓用 Web 服務的協同作業時，要建立可呼叫 (或顯現成) Web 服務的協同作業。您也要建立或改寫商業物件。如需相關作業的概觀，請參閱第 8 頁的『Web 服務配置工具』。

### 將協同作業顯現為 Web 服務

如需逐步說明，請參閱第 131 頁的第 7 章，『將協同作業顯現為 Web 服務』。

### 啓用協同作業來呼叫 Web 服務

如需逐步說明，請參閱第 129 頁的第 6 章，『啓用要求處理程序的協同作業』。

## 配置 SOAP 資料處理常式

您於安裝產品檔案之後，開機之前，配置資料處理常式 Meta 物件中的資訊。除非您是要新增自訂的名稱處理常式，否則您可使用預設的 SOAP 資料處理常式配置，以節省時間。但是您必須針對每一個資料處理常式轉換作業，配置特定的 meta 物件資訊。此資訊內含於 SOAP Config MO 中。當您建立商業物件時，您要指定 SOAP Config MO。當您開發協同作業來呼叫 Web 服務 (要求處理程序) 時，許多作業都是自動的：當您使用 WSDL ODA 來產生 SOAP 訊息的商業物件時，會自動為您產生 SOAP Config MO。

如需有關配置資料處理常式的進一步資訊，請參閱第 97 頁的第 5 章，『SOAP 資料處理常式』。

---

## 限制

- WSDL ODA 會自動產生商業物件。若結果不符合您的需求，您必須使用「商業物件設計程式」，以手動的方式來建立商業物件。

請參閱 WSDL ODA 支援的說明，使用類型和元素來查看各種組合的屬性樣式、用法及組件定義。

- 欲瞭解有關樣式 (rpc、文件)、用法 (文字、編碼) 的 XML 限制，以及如何定義組件，請參閱第 97 頁的第 5 章，『SOAP 資料處理常式』及第 129 頁的第 6 章，『啓用要求處理程序的協同作業』。
- 連接器僅支援 SOAP/HTTP 及 SOAP/JMS 連結。
- 連接器的 SOAP/JMS 通訊協定接聽器僅支援佇列目的地；不支援主題。可支援 JMS 文字及位元組訊息。
- 可支援 HTTP POST 要求及回應。但不支援其他 HTTP 方法，亦不支援 HTTP 1.1 持續性連線。



---

## 第 2 章 安裝及啟動

- 『安裝作業概觀』
- 『安裝連接器及相關檔案』
- 第 15 頁的『配置作業概觀』
- 第 16 頁的『執行配接器的多重實例』
- 第 17 頁的『啟動及停止連接器』

本章說明如何安裝元件，以為 Web 服務實作連接器。如需有關一般安裝 ICS 系統的資訊，請參閱適合您平台的系統安裝手冊。

---

### 安裝作業概觀

如需分配管理系統相容性、配接器組織架構、軟體需求、相依關係及標準與 API 的相關資訊，請參閱第 1 頁的『Web 服務配接器環境』。

若要安裝 Web 服務的連接器，您必須執行下列作業：

#### 安裝 ICS

這項作業包括安裝系統及啟動 ICS，「系統安裝手冊」中有詳細說明。您必須安裝 ICS 4.2 版。

若要將檔案載入儲存庫，請參閱 *WebSphere InterChange Server 實作手冊*。

#### 安裝連接器及相關檔案

這項作業包括從套裝軟體將連接器的檔案 (及相關元件) 安裝到您的系統上。請參閱『安裝連接器及相關檔案』。

---

### 安裝連接器及相關檔案

如需有關安裝 WebSphere Business Integration 配接器產品的資訊，請參閱 *WebSphere Business Integration Adapters 安裝手冊*，其位於 WebSphere Business Integration Adapters Infocenter，網址如下：

<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

---

### 已安裝檔案的結構

本節中的表格顯示已安裝之檔案的結構。

#### Windows 連接器檔案結構

「安裝程式」會將與連接器相關的標準檔案複製到您的系統中。

公用程式會安裝連接器，並將連接器代理程式的捷徑新增至「開始」功能表。

表 1 說明連接器使用的 Windows 檔案結構，並顯示當您選擇透過「安裝程式」來安裝連接器時會自動安裝的檔案。

表 1. 配接器的安裝 Windows 檔案結構

ProductDir 的子目錄	說明
\lib\WBIA.jar	WebSphere Business Integration Adapter jar 檔
\bin\CWConnEnv.bat	一般的連接器啟動檔案
\bin\ODAEnv.bat	一般的 ODA 啟動檔案
connectors\WebServices\CWebServices.jar	Web 服務連接器
connectors\WebServices\start_WebServices.bat	連接器的啟動檔案
DataHandlers\CwSOAPDataHandler.jar	SOAP 資料處理常式
DataHandlers\CwSOAPNameHandler.jar	SOAP 名稱處理常式
repository\DataHandlers\MO_DataHandler_SOAP.xsd	SOAP 資料處理常式相關檔案
bin\Data\App\WebServicesConnectorTemplate	Web 服務連接器範本
ODA\WSDL\WSDLODA.jar	WSDL ODA
ODA\WSDL\start_WSDLODA.bat	WSDL ODA 啟動檔案
connectors\WebServices\dependencies\soap.jar	SOAP 連接器、SOAP 資料處理常式、WSDL 配置精靈及 WSDL ODA 所需的 Apache SOAP API。
connectors\WebServices\dependencies\LICENSE	Apache 授權檔
connectors\WebServices\dependencies\mail.jar	JavaMail API
connectors\WebServices\dependencies\activation.jar	Java 啟動組織架構
connectors\WebServices\dependencies\ibmjse.jar	IBM 提供的 JSSE (Java Secure Socket Extension) API
connectors\WebServices\dependencies\jms.jar	Java 傳訊服務
connectors\WebServices\dependencies\uddi4j-wsd1.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\uddi4jv2.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\IPL10.txt	WSDL ODA 所需的授權檔
connectors\WebServices\dependencies\wsd14j.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\CPL10.txt	WSDL ODA 所需的授權檔
connectors\WebServices\dependencies\qname.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\j2ee.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\wsb2.1.1\common.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\wsb2.1.1\core.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\wsb2.1.1\xercesImpl.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\wsb2.1.1\xmlParserAPIs.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\wsb2.1.1\xsd.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\wsb2.1.1\xsd.resources.jar	為 WSDL ODA 所需
connectors\WebServices\dependencies\IBMReadme.txt	授權
connectors\WebServices\samples\WebSphereICS\WebServicesSample.jar	範例儲存庫檔案
connectors\WebServices\samples\WebSphereICS\CLIENT_SYNC_TLO_OrderStatus.bo	用於測試連接器的範例 (同步的) 商業物件
connectors\WebServices\samples\WebSphereICS\CLIENT_ASYNC_TLO_Order.bo	用於測試連接器的範例 (非同步的) 商業物件
connectors\messages\WebServicesConnector.txt	連接器訊息檔案
ODA\messages\WSDLODAAgent.txt	WSDL ODA 的訊息檔案

註：所有產品路徑名稱皆相對於您系統上安裝的產品目錄。

## UNIX 連接器檔案結構

「安裝程式」會將與連接器相關的標準檔案複製到您的系統中。

表 2 說明連接器使用的 UNIX 檔案結構，並顯示當您選擇透過「安裝程式」來安裝連接器時會自動安裝的檔案。

表 2. 配接器的安裝 UNIX 檔案結構

ProductDir 的子目錄	說明
/lib/WBIA.jar	WebSphere Business Integration Adapter jar 檔
/bin\CWConnEnv.bat	一般的連接器啟動檔案
/bin\ODAEnv.bat	一般的 ODA 啟動檔案
connectors/WebServices/CWebServices.jar	Web 服務連接器
connectors/WebServices/start_WebServices.sh	連接器的啟動檔案
DataHandlers/CwSOAPDataHandler.jar	SOAP 資料處理常式
DataHandlers/CwSOAPNameHandler.jar	SOAP 名稱處理常式
repository\DataHandlers\MO_DataHandler_SOAP.xsd	SOAP 資料處理常式相關檔案
bin/Data/App/WebServicesConnectorTemplate	Web 服務連接器範本

表 2. 配接器的安裝 UNIX 檔案結構 (繼續)

ProductDir 的子目錄	說明
ODA/WSDL/WSDL.ODA.jar	WSDL ODA
ODA/WSDL/start_WSDL.ODA.sh	WSDL ODA 啟動檔案
connectors/WebServices/dependencies/soap.jar	SOAP 連接器、SOAP 資料處理常式、WSDL 配置精靈及 WSDL ODA 所需的 Apache SOAP API。
connectors/WebServices/dependencies/LICENSE	Apache 授權檔
connectors/WebServices/dependencies/mail.jar	JavaMail API
connectors/WebServices/dependencies/activation.jar	Java 啟動組織架構
connectors/WebServices/dependencies/ibmjse.jar	IBM 提供的 JSSE (Java Secure Socket Extension) API
connectors/WebServices/dependencies/jms.jar	Java 傳訊服務
connectors/WebServices/dependencies/uddi4j-wsd1.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/uddi4jv2.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/IPL10.txt	WSDL ODA 所需的授權檔
connectors/WebServices/dependencies/wsd14j.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/CPL10.txt	WSDL ODA 所需的授權檔
connectors/WebServices/dependencies/qname.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/j2ee.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/wswb2.1.1/common.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/wswb2.1.1/ecore.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/wswb2.1.1/xercesImpl.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/wswb2.1.1/xmlParserAPIs.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/wswb2.1.1/xsd.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/wswb2.1.1/xsd.resources.jar	為 WSDL ODA 所需
connectors/WebServices/dependencies/IBMReadme.txt	授權
connectors/WebServices/samples/WebSphereICS/WebServicesSample.jar	範例儲存庫檔案
connectors/WebServices/samples/WebSphereICS/CLIENT_SYNC_TLO_OrderStatus.bo	用於測試連接器的範例 (同步的) 商業物件
connectors/WebServices/samples/WebSphereICS/CLIENT_ASYNC_TLO_Order.bo	用於測試連接器的範例 (非同步的) 商業物件
connectors/messages/WebServicesConnector.txt	連接器訊息檔案
ODA/messages/WSDL.ODAAgent.txt	WSDL ODA 的訊息檔案

註：所有產品路徑名稱皆相對於您系統上安裝的產品目錄。

## 配置作業概觀

您必須在安裝之後、啟動之前配置元件，如下所示：

### 配置連接器

這項作業包括設定及配置連接器。請參閱第 76 頁的『配置連接器』。

### 配置商業物件

配置商業物件的步驟，取決於您用來實作產品套件的方式：

- **要求處理程序**您必須建立對應於下列各項的商業物件：
  - 要傳送到每個 Web 服務的要求訊息
  - 每個可能的回應，包括錯誤

如需進一步資訊，請檢閱第 19 頁的第 3 章，『商業物件基本需求』，然後參閱第 129 頁的第 6 章，『啟用要求處理程序的協同作業』。

- **事件處理程序**您可使用 TLO 或非 TLO 商業物件。
  - 如需進一步資訊，請檢閱第 19 頁的第 3 章，『商業物件基本需求』，然後參閱第 131 頁的第 7 章，『將協同作業顯現為 Web 服務』。

## 配置資料處理常式

SOAP 資料處理常式 Meta 物件必須在安裝之後配置。此外，必須為每個 SOAP 商業物件配置 SOAP Config MO。若要配置資料處理常式，請參閱第 97 頁的第 5 章，『SOAP 資料處理常式』。

## 配置協同作業

- **要求處理程序**若協同作業呼叫 Web 服務來作為其處理程序的一部份，您要使用 WSDL ODA 來產生商業物件，然後將協同作業物件埠連結到連接器。如需進一步資訊，包括逐步式程序，請參閱第 129 頁的第 6 章，『啟用要求處理程序的協同作業』。
- **事件處理程序**若協同作業被顯現為 Web 服務，則您必須使用「WSDL 配置精靈」來產生 WSDL 文件，讓可能的用戶端皆可使用該文件，然後配置協同作業物件的埠，讓用戶端能夠呼叫協同作業。如需進一步資訊，包括逐步式程序，請參閱第 131 頁的第 7 章，『將協同作業顯現為 Web 服務』。

---

## 執行配接器的多重實例

建立多重連接器實例在許多方面都與建立自訂連接器相同。您可以遵循下列步驟，設定系統以建立及執行多重連接器實例。您必須：

- 為連接器實例建立新目錄
- 確定您有必備的商業物件定義
- 建立新的連接器定義檔案
- 建立新的啟動 Script

### 建立新目錄

您必須為每個連接器實例建立連接器目錄。這個連接器目錄應該命名為：

```
ProductDir\connectors\connectorInstance
```

其中 connectorInstance 可唯一地識別連接器實例。

如果連接器有任何連接器特有的 Meta 物件，則必須為連接器實例建立 Meta 物件。如果您要將 Meta 物件儲存為檔案，請建立此目錄 (如下所示)，並在此處儲存該檔案：

```
ProductDir\repository\connectorInstance
```

### 建立商業物件定義

如果專案中沒有每個連接器實例的商業物件定義，則必須建立它們。

1. 如果您需要修改與起始連接器相關的商業物件定義，請複製適當的檔案，並使用「商業物件設計程式」來匯入它們。您可以複製起始連接器的任何檔案。如果您變更這些檔案，只需要替它們重新命名即可。
2. 起始連接器的檔案應該位於下列目錄：

```
ProductDir\repository\initialConnectorInstance
```

您建立的任何其他檔案都應該位於 ProductDir\repository 的適當 connectorInstance 子目錄中。

## 建立連接器定義

您可以在「連接器配置程式」中，為連接器實例建立配置檔 (連接器定義)。做法是：

1. 複製起始連接器的配置檔 (連接器定義) 並將它重新命名。
2. 確定每個連接器實例都正確地列出它支援的商業物件 (及任何相關聯的 Meta 物件)。
3. 視需要自訂任何連接器內容。

## 建立啟動 Script

若要建立啟動 Script：

1. 複製起始連接器的啟動 Script，並替它命名以包括連接器目錄的名稱：

`dirname`

2. 把這個啟動 Script 放在您於第 16 頁的『建立新目錄』中建立的連接器目錄中。
3. 建立啟動 Script 捷徑 (僅限 Windows)。
4. 複製起始連接器的捷徑文字，並變更起始連接器的名稱 (在指令行中) 以符合新連接器實例的名稱。

您現在可以在整合伺服器上同時執行這兩個連接器實例。

如需建立自訂連接器的詳細資訊，請參閱 *Connector Development Guide for C++ or for Java*。

---

## 啟動及停止連接器

**重要事項：**如本章先前提到的，連接器、商業物件、SOAP 資料處理常式 Meta 物件及協同作業，都必須於安裝之後、啟動連接器之前配置完成，以確保運作正常。如需這些作業的摘要，請參閱第 15 頁的『配置作業概觀』。此外，請勿停用連接器輪詢功能 (依預設，會啟用連接器輪詢功能)。

連接器必須使用其**連接器啟動 Script** 明確地啟動。啟動 Script 應該位於連接器的執行時期目錄：

`ProductDir\connectors\connName`

其中 `connName` 可識別連接器。啟動 Script 的名稱視作業系統平台而定，如表 3 所示。

表 3. 連接器的啟動 Script

作業系統	啟動 Script
UNIX 型系統	<code>connector_manager_connName</code>
Windows	<code>start_connName.bat</code>

您可以利用下列任一方法來呼叫連接器啟動 Script：

- 在 Windows 系統上，從**開始功能表**

選取**程式集 > IBM WebSphere Business Integration Adapters > Adapters > 連接器**。依預設，程式名稱為“IBM WebSphere Business Integration Adapters”。然而，您可以自訂名稱。您也可以選擇建立連接器的桌面捷徑。

- 從指令行

– 在 Windows 系統上：

```
start_connName connName brokerName [-cconfigFile ]
```

– 在 UNIX 型系統上：

```
connector_manager_connName -start
```

其中 *connName* 是連接器名稱，而 *brokerName* 可識別您的整合分配管理系統，如下所示：

- 若為 WebSphere InterChange Server，請將 *brokerName* 指定為 ICS 實例名稱。
- 若為 WebSphere Message Brokers (WebSphere MQ Integrator、WebSphere MQ Integrator Broker 或 WebSphere Business Integration Message Broker) 或 WebSphere Application Server，請將 *brokerName* 指定為可識別分配管理系統的字串。

**註：**若為 Windows 系統上的 WebSphere Message Broker 或 WebSphere Application Server，則您必須包括 `-c` 選項，後面跟著連接器配置檔的名稱。若為 ICS，則 `-c` 是選用的。

- 從您啟動「系統管理程式」時啟動的「配接器監視器」(僅限 WebSphere Business Integration Adapters 產品)

您可以使用這個工具來載入、啟動、取消啟動、暫停、關閉或刪除連接器。

- 從「系統監視器」(僅限 WebSphere InterChange Server 產品)

您可以使用這個工具來載入、啟動、取消啟動、暫停、關閉或刪除連接器。

- 在 Windows 系統上，您可以配置連接器以啟動為 Windows 服務。在此案例中，當 Windows 系統開機時 (為「自動」服務)，或當您透過「Windows 服務」視窗啟動服務時 (為「手動」服務) 時，就會啟動連接器。

如需如何啟動連接器的詳細資訊 (包括指令行啟動選項)，請參閱下列文件之一：

- 若為 WebSphere InterChange Server，請參閱 *System Administration Guide*。
- 若為 WebSphere Message Brokers，請參閱 *Implementing Adapters with WebSphere Message Brokers*。
- 若為 WebSphere Application Server，請參閱 *Implementing Adapters with WebSphere Application Server*。

---

## 第 3 章 商業物件基本需求

- 『商業物件 Meta 資料』
- 『連接器商業物件結構』
- 第 20 頁的『處理 TLO 的同步事件』
- 第 31 頁的『處理 TLO 的非同步事件』
- 第 34 頁的『處理非 TLO 的事件』
- 第 35 頁的『處理 TLO 的同步要求』
- 第 35 頁的『處理 TLO 的同步要求』
- 第 45 頁的『處理 TLO 的非同步要求』
- 第 49 頁的『開發商業物件』

本章說明連接器商業物件的結構、基本需求及屬性。

---

### 商業物件 Meta 資料

Web 服務的連接器為以 Meta 資料導向的連接器。於商業物件中，Meta 資料是指應用程式的相關資料，儲存於商業物件定義中，可協助連接器與應用程式相互作用。Meta 資料導向的連接器依據商業物件定義中編碼的 Meta 資料，來處理其所支援的每個商業物件，而不是依據寫於連接器中的指示。

商業物件 Meta 資料包括商業物件的結構、其屬性內容的設定值，及其應用程式特有資訊的內容。因為連接器為 Meta 資料導向，所以可處理新的或修改過的商業物件，不需要修改連接器程式碼。然而，連接器之已配置資料處理常式會假設其商業物件的結構、物件基數、應用程式特定文字的格式，及商業物件的資料庫表示法。因此，當您建立或修改 Web 服務的商業物件時，您的修改必須符合連接器所遵循的規則，否則連接器無法正確地處理新的或修改過的商業物件。

如需有關 Meta 資料、Meta 物件、配置，及其與商業物件和 SOAP 訊息互動的詳細資訊，請參閱第 97 頁的第 5 章，『SOAP 資料處理常式』。

---

### 連接器商業物件結構

連接器處理兩種商業物件：

- **TLO** Web 服務最上層商業物件 (TLO) 包含「要求」商業物件，以及 (選擇性)「回應」和「錯誤」商業物件。這些子項物件包含內容資料和 SOAP Config MO，及 (選擇性) Protocol Config MO。下面各段將說明 TLO、「要求」、「回應」和「錯誤」物件，及有關要求對事件處理程序的應用程式特有資訊、屬性和基本需求，並附圖解。

**註：** TLO 適用於要求處理程序及事件處理程序。

- **非 TLO** 這些物件是非 TLO 的通用商業物件 (GBO) 及應用程式特有商業物件 (ASBO)，「WSDL 配置精靈」在產生 WSDL 時，會使用這些物件。連接器可於事件處理程序期間，處理非 TLO。將於第 34 頁的『處理非 TLO 的事件』中討論這些物件。如需進一步資訊，請參閱第 134 頁的『WSDL 配置精靈』。

註: 非 TLO 僅適用於事件處理程序。

註: SOAP 標頭儲存區及標頭商業物件內含於「要求」、「回應」和「錯誤」商業物件中，本章不加以討論。如需有關 SOAP 標頭儲存區及標頭商業物件的資訊，請參閱第 97 頁的第 5 章, 『SOAP 資料處理常式』。

## 處理 TLO 的同步事件

就事件處理程序而言，連接器可允許二種 TLO--同步及非同步。本節將討論處理 TLO 的同步事件。

圖 4顯示同步事件處理程序的商業物件階層。「要求」和「回應」物件是必要的，「錯誤」物件是選用性的。

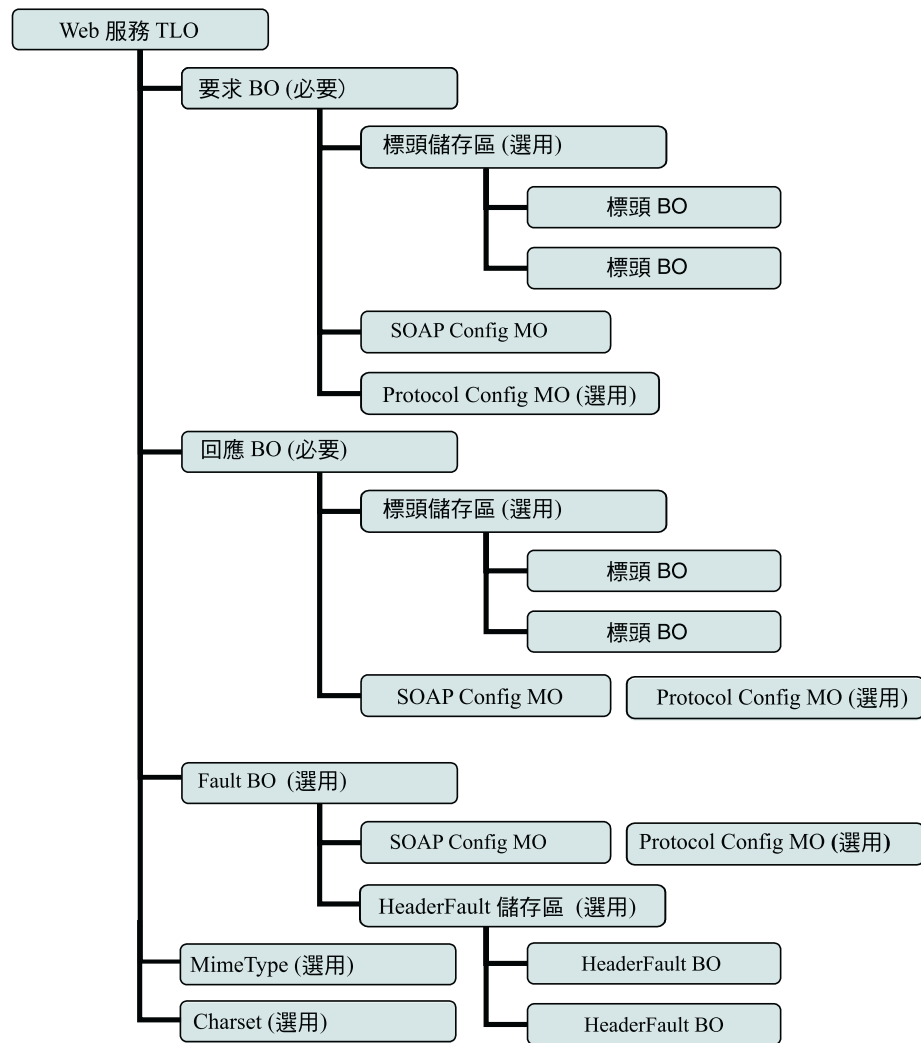


圖 4. 同步事件處理程序的商業物件階層

TLO 包含物件層次 ASI 以及具有屬性層次 ASI 的屬性。這二種 ASI 的說明如下。



## 同步事件處理程序 TLO 的物件層次 ASI

物件層次 ASI 提供有關 TLO 本質及其所包含之物件的基本資訊。圖 5 顯示 SERVICE\_SYNCH\_OrderStatus 的物件層次 ASI，這是同步事件處理程序的 TLO 範例。

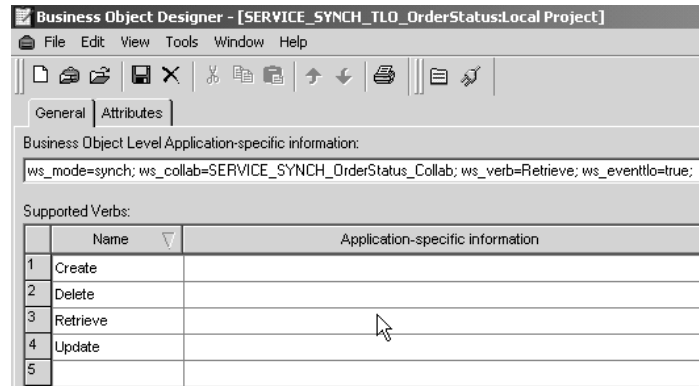


圖 5. 同步事件處理程序的最上層商業物件

以下的表 4 說明同步事件處理程序 TLO 的物件層次 ASI。

表 4. 處理 TLO 物件 ASI 的同步事件

物件層次 ASI	說明
ws_eventtlo=true	若此 ASI 內容的設定為 true，連接器就會將這個物件當成僅適用於事件處理程序的 TLO。  請注意，「WSDL 配置精靈」使用此 ASI 來判斷商業物件是否為 TLO。如需詳細資訊，請參閱第 134 頁的『WSDL 配置精靈』。
ws_collab=collabname	此 ASI 會指示連接器要呼叫哪一個協同作業。其值為協同作業的名稱。(於產生 WSDL 期間，亦會使用此 ASI 來判斷協同作業的 TLO。如需詳細資訊，請參閱第 134 頁的『WSDL 配置精靈』。) 於圖 5 所顯示的範例中，協同作業名稱為 SERVICE_SYNCH_OrderStatus_Collab)
ws_verb=verb	在將 TLO 傳遞至協同作業之前，連接器會使用此 ASI 來設定 TLO 上的動詞。在圖 5 所顯示的範例中，動詞為 Retrieve。
ws_mode=synch	於事件通知期間，連接器會使用這個 ASI 內容，來決定要以同步 (synch) 或非同步 (asynch) 的方式呼叫協同作業。若為同步的處理程序，此 ASI 必須設定為 synch。  預設值為 asynch。

## 同步事件處理程序 TLO 的屬性層次 ASI

每個處理 TLO 的同步事件皆有屬性及屬性層次 ASI。圖 6 顯示 SERVICE\_SYNCH\_OrderStatus 的屬性，這是 TLO 範例。它也在 App Spec Info 直欄中顯示了屬性層次 ASI。

The screenshot shows the 'Attributes' tab in Business Object Designer. It displays a table with 5 rows of attributes for a TLO. The columns are: Pos, Name, Type, Key, Foreign, Required, Card, and App Spec Info.

	Pos	Name	Type	Key	Foreign	Required	Card	App Spec Info
1	1	Request	SERVICE_SYNCH_OrderStatus_Request	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=request
2	2	Response	SERVICE_SYNCH_OrderStatus_Response	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=response
3	3	Fault	SERVICE_SYNCH_OrderStatus_Fault	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=fault
4	4	ObjectEventId	String					
5	5			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

圖 6. 同步事件處理程序的 TLO 屬性

表 5 彙總了同步事件處理程序 TLO 的屬性層次 ASI，包括 Request、Response、Fault、MimeType 及 Charset 屬性。

表 5. 處理 TLO 屬性 ASI 的同步事件

TLO 屬性	屬性層次 ASI	說明
MimeType		選用屬性：如果已指定，則它的值會作為對同步回應來呼叫的資料處理常式 MIME 類型。其類型為 String，預設值為 xml/soap。
Charset		此為 String 類型的選用性參數，其指定在將送出的商業物件轉換成訊息時，要設定於資料處理常式上的 charset。附註：在這個屬性中指定的 Charset 值將不會在回應訊息的 Content-Type 通訊協定標頭中傳送。
Request	ws_botype=request	此屬性對應於 Web 服務要求。連接器使用其 ASI 來判斷這個 TLO 屬性的類型是否為「SOAP 要求 BO」。這個 ASI (不是屬性名稱) 可判斷屬性類型。若有多個要求屬性，連接器就會使用第一個屬性的 ASI。  此為同步事件處理程序 TLO 的必要屬性。

表 5. 處理 TLO 屬性 ASI 的同步事件 (繼續)

TLO 屬性	屬性層次 ASI	說明
Response	ws_botype=response	此屬性對應於 Web 服務所傳回的回應。連接器會使用這個 ASI 來判斷此 TLO 屬性的類型是否為「SOAP 回應 BO」。這個 ASI (不是屬性名稱) 可判斷屬性類型。若有多個回應屬性，連接器就會使用第一個屬性的 ASI。  此為同步事件處理程序 TLO 的必要屬性。
Fault	ws_botype=fault ws_botype=defaultfault	此屬性為同步事件處理程序的選用性屬性，其對應於當處理程序無法順利移入回應時，協同作業所傳回的錯誤訊息。連接器會使用此 ASI (非屬性名稱) 來判斷屬性的類型是否為「SOAP 錯誤 BO」。若 ws_botype=defaultfault，「WSDL 配置精靈」便會使用此「錯誤」商業物件來處理標頭。如需進一步資訊，請參閱第 107 頁的『標頭錯誤處理程序』。

### 同步事件處理程序的「要求」商業物件

「要求」商業物件是 TLO 的子項，為同步事件處理程序的必要物件。「要求」商業物件具有物件層次 ASI。比方說，如果於「商業物件設計程式」中開啓 SERVICE\_SYNCH\_OrderStatus\_Request，並按一下「一般」標籤，就會顯示物件層次 ASI，如圖 7 所示。

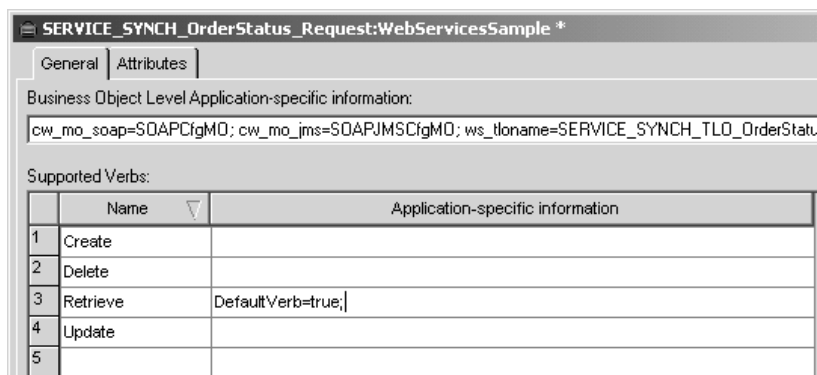


圖 7. 同步事件處理程序要求物件的物件層次 ASI

同步事件處理程序之「要求」商業物件的物件層次 ASI 說明位於表 6 中。如圖 7 所示，您可為「要求」商業物件指定預設動詞。若要這麼做，請將：

DefaultVerb=true;

指定於 ASI 欄位中，以作為「要求」商業物件最上層之「受支援動詞」清單中的動詞。若未指定 DefaultVerb ASI，且資料處理常式是於未設定動詞的情況下處理商業物件，則傳回商業物件時，就沒有動詞。

表 6. 同步的事件處理程序：「要求」商業物件的物件層次 ASI

物件層次 ASI	說明
<code>cw_mo_soap=SOAPCfgMO</code>	此 ASI 的值必須符合對應於 SOAP Config MO 之屬性的名稱。此為一個 Meta 物件，可針對「要求」商業物件來定義資料處理常式轉換作業。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。
<code>cw_mo_jms=SOAPJMSCfgMO</code> 或 <code>cw_mo_http=SOAPHTTPCfgMO</code>	此 ASI 的值必須符合對應於 Protocol Config MO 之屬性的名稱。第一個 ASI 指定 SOAP/JMS 通訊協定接聽器；第二個 ASI 指定 SOAP/HTTP 或 SOAP/HTTPS 通訊協定接聽器。ASI 和 Protocol Config MO 皆為選用性的。如需進一步資訊，請參閱第 25 頁的『Protocol Config MO』。
<code>ws_tlname=tlname</code>	此 ASI 指定此物件所屬之 Web 服務 TLO 的名稱。於事件處理程序期間，連接器會使用此 ASI 來判斷資料處理常式所遞送的「要求」商業物件是否為 TLO 的子項。如果是，連接器就會建立所指定的 TLO，將「要求」商業物件設定為其子項，並使用 TLO 物件層次 ASI，將之傳遞至訂閱協同作業。

## 同步事件處理程序的「回應」商業物件

「回應」商業物件是 TLO 的子項，為同步事件處理程序的必要物件。同步事件處理程序的「回應」商業物件的物件層次 ASI 說明位於表 7 中。

表 7. 同步的事件處理程序：「回應」商業物件的物件層次 ASI

物件層次 ASI	說明
<code>cw_mo_soap=SOAPCfgMO</code>	此 ASI 的值必須符合對應於 SOAP Config MO 之屬性的名稱。這是 SOAP Config MO，可針對「回應」商業物件來定義資料處理常式轉換作業。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。

註：您可以選擇為「回應 BO」併入 Protocol Config MO 物件層次 ASI。

## 同步事件處理程序的「錯誤」商業物件

「錯誤」商業物件是 TLO 的子項，為同步事件處理程序的選用性物件。同步事件處理程序的「錯誤」商業物件的物件層次 ASI 說明位於表 8 中。

表 8. 同步的事件處理程序：「錯誤」商業物件的物件層次 ASI

物件層次 ASI	說明
<code>cw_mo_soap=SOAPCfgMO</code>	此 ASI 的值必須符合對應於 SOAP Config MO 之屬性的名稱。此為 SOAP Config MO，可針對「錯誤」商業物件來定義資料處理常式轉換作業。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。

註: 您可以選擇為「錯誤 BO」併入 Protocol Config MO 物件層次 ASI。

## SOAP Config MO

圖 8 顯示於「商業物件設計程式」中展開的 SOAP Config MO 範例。

	Pos	Name	Type	Key	Foreign	Required	Card	Maximum Length	Default
1	1	OrderId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255	
2	4	SOAPJMSCfgMO	SERVICE_SYNCH_SOAP_JMS_OrderStatus_Req	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
3	5	ObjectEventId	String						
4	2	OrderHeader	SERVICE_SYNCH_OrderStatus_Request_Header	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
5	3	SOAPCfgMO	SERVICE_SYNCH_OrderStatus_Request_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
5.1	3.6	BodyName	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	getOrderStatus
5.2	3.3	Style	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	rpc
5.3	3.4	TypeInfo	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	true
5.4	3.5	TypeCheck	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	none
5.5	3.2	BOVerb	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	Retrieve
5.6	3.7	ObjectEventId	String						
5.7	3.1	BodyNS	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	http://www.mycompany.com/samples/orderstatus

圖 8. 同步事件處理程序的 SOAP Config MO 屬性

SOAP Config MO 可定義單一資料處理常式轉換的格式化行為 -- 由 SOAP 訊息轉換成商業物件，或是由商業物件轉換成 SOAP 訊息。每個「要求」、「回應」及「錯誤」屬性皆有 SOAP Config MO。其屬性

(BodyName、BodyNS、Style、Use、TypeInfo、TypeCheck 及 BOVerb) 類型一律為 String。這些屬性對應於 SOAP 訊息元素，其值可決定 SOAP 資料處理常式讀取及驗證訊息和物件的方式。如需有關 SOAP Config MO 及屬性的詳細資訊，請參閱第 99 頁的『SOAP 配置 meta 物件：每個 SOAP 商業物件的子項』。所有的 SOAP Config MO (無論是需要、回應或錯誤物件) 皆必須要有專屬的項目，以作為 BodyName 及 BodyNS 的預設值。

## Protocol Config MO

圖 9 顯示 JMS Protocol Config MO，其屬性對應於入埠 SOAP 訊息中的標頭。

	Pos	Name	Type	Key	Foreign	Required	Card
1	1	Request	SERVICE_SYNCH_OrderStatus_Request	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
1.1	1.1	OrderId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
1.2	1.5	ObjectEventId	String				
1.3	1.3	SOAPCfgMO	SERVICE_SYNCH_OrderStatus_Request_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
1.4	1.4	SOAPJMSCfgMO	SERVICE_SYNCH_SOAP_JMS_OrderStatus_Request_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
1.4.1	1.4.1	MessageId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.2	1.4.2	Priority	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.3	1.4.3	Expiration	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.4	1.4.4	DeliveryMode	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.5	1.4.5	ReplyTo	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
1.4.6	1.4.6	ObjectEventId	String				

圖 9. 同步事件處理程序的 JMS Protocol Config MO 屬性

此 MO 為選用項目，在事件處理程序中，用來作為要求、回應及錯誤商業物件的子項。您通常會在需要讀取（來自要求訊息）或傳送（到回應或錯誤訊息）通訊協定標頭及自訂內容時指定它。如上述，要求商業物件會選擇性地將 Protocol Config MO 的名稱宣告為商業物件層次 ASI：

- `cw_mo_jms=JMSProtocolListenerConfigMOAttribute`
- `cw_mo_http=HTTPProtocolListenerConfigMOAttribute`

於事件處理程序期間，連接器會使用通訊協定接聽器（SOAP/HTTP、SOAP/HTTPS 或 SOAP/JMS），從傳輸來擷取事件。這些事件是因為內部或外部 Web 服務用戶端向顯現為 Web 服務的協同作業要求服務，而傳回的訊息。每個傳輸皆有其自己的標頭基本需求。連接器會使用 Protocol Config MO，將通訊協定特有的標頭資訊從通訊協定接聽器傳送到協同作業。Protocol Config MO 屬性對應於入埠 SOAP/JMS 訊息中的標頭。連接器會使用入埠 SOAP 訊息內容，在商業物件中設定這些屬性的值。若為 SOAP/JMS 通訊協定，則事件及要求處理的 Protocol Config MO 屬性如下：

表 9. SOAP JMS Protocol Config MO 屬性：事件及要求處理

SOAP/JMS Protocol Config MO 屬性	JMSHeaderName	說明
CorrelationID	JMSCorrelationID	入埠訊息：會將來自 JMSCorrelationID 標頭的值移入這個屬性。 離埠訊息：這個屬性的值會設為外送訊息的 JMSCorrelationID 標頭。
MessageId	JMSMessageId	入埠訊息：會將來自 JMSMessageId 標頭的值移入這個屬性。 離埠訊息：這個屬性不會用於離埠訊息。

表 9. SOAP JMS Protocol Config MO 屬性：事件及要求處理 (繼續)

SOAP/JMS Protocol Config MO 屬性	JMSHeaderName	說明
優先順序	JMSPriority	入埠訊息：會將來自 JMSPriority 標頭的值移入這個屬性。離埠訊息：這個屬性的值會在外送訊息的 JMSPriority 標頭中設定。
Expiration	JMSExpiration	入埠訊息：會將來自 JMSExpiration 標頭的值移入這個屬性。離埠訊息：這個屬性的值會在外送訊息的 JMSExpiration 標頭中設定。
DeliveryMode	JMSDeliveryMode	入埠訊息：會將來自 JMSDeliveryMode 標頭的值移入這個屬性。離埠訊息：這個屬性的值會在外送訊息的 JMSDeliveryMode 標頭中設定。
Destination	JMSDestination	入埠訊息：會將來自 JMSDestination 標頭的值移入這個屬性。 離埠訊息： <b>要求處理程序</b> 這個屬性的值會被當作目標佇列名稱來使用，且會在外送訊息的 JMSDestination 標頭中間接地設定為衍生的目標路徑。 <b>同步回應</b> (在事件通知中)：不會使用這個屬性。
Redelivered	JMSRedelivered	入埠訊息：會將來自 JMSRedelivered 標頭的值移入這個屬性。離埠訊息：這個屬性的值會在外送訊息的 JMSRedelivered 標頭中設定。
ReplyTo	JMSReplyTo	入埠訊息：會將來自 JMSReplyTo 標頭的值移入這個屬性。離埠訊息：這個屬性的值會在外送訊息的 JMSReplyTo 標頭中設定。
TimeStamp	JMSTimeStamp	入埠訊息：會將來自 JMSTimeStamp 標頭的值移入這個屬性。離埠訊息：這個屬性的值會在外送訊息的 JMSTimeStamp 標頭中設定。

表 9. SOAP JMS Protocol Config MO 屬性：事件及要求處理 (繼續)

SOAP/JMS Protocol Config MO 屬性	JMSHeaderName	說明
類型	JMSType	入埠訊息：會將來自 JMSType 標頭的值移入這個屬性。 離埠訊息：這個屬性的值會在外送訊息的 JMSType 標頭中設定。
UserDefinedProperties	請參閱第 29 頁的『事件處理程序的使用者定義內容』。	此選用性讀寫屬性將保留使用者定義的通訊協定內容商業物件。如需進一步資訊，請參閱第 29 頁的『事件處理程序的使用者定義內容』。

註：協同作業要負責確定，在「要求-回應」事件的環境定義中，傳遞至 JMS Protocol Config MO 的標頭值在邏輯上是正確的。

若為 SOAP/HTTP(S) 通訊協定，則 Protocol Config MO 屬性如下：

表 10. 事件處理程序的 HTTP/HTTPS Protocol Config MO 屬性

屬性	必要的	類型	說明
Content-Type	否	字串	此屬性的值可定義外送訊息的 Content-Type 標頭 (包括外送訊息的訊息 ContentType 及 0 或其他參數 -- Charset)。語法與 HTTP 通訊協定中的 Content-Type 標頭語法相同，例如： <code>text/html; charset=ISO-8859-4</code> 。若未定義 Content-Type 屬性，連接器就會使用要求的 ContentType 作為回應/錯誤訊息的 ContentType。
UserDefinedProperties	否	商業物件	此屬性會保留使用者定義的通訊協定內容商業物件。
一或數個 HTTP 標頭	否	字串	此屬性可讓處理常式傳遞或擷取所指定之 HTTP 標頭的值。
Authorization_UserID	否	String	此屬性對應於 HTTP 基礎鑑別的 userID。
Authorization_Password	否	String	此屬性對應於 HTTP 基礎鑑別的密碼。

這些屬性的說明位於下列各節中：

- 第 29 頁的『事件處理程序的使用者定義內容』
- 第 29 頁的『事件處理程序的 HTTP 認證傳送』

如需有關通訊協定接聽器的進一步資訊，請參閱第 55 頁的『通訊協定接聽器』。(如需要處理程序的 Protocol Config MO 的說明，請參閱第 35 頁的『處理 TLO 的同步要求』。)



**事件處理程序的使用者定義內容:** 您可以選擇性地指定 HTTP(S) Protocol Config MO 中的自訂內容。若要這麼做，請併入 `UserDefinedProperties` 屬性。此屬性對應於商業物件，該商業物件中有一或數個子屬性具有內容值。此商業物件中的每個屬性都必須在訊息標頭的變數部分中，定義單一內容，以供讀取 (若為同步的回應，則是寫入)，如下所示：

- 無論通訊協定內容的類型為何，屬性的類型一律為 `String`。屬性的應用程式特有資訊可以包含二個名稱值組，定義屬性所對映之通訊協定訊息內容的名稱及格式。

表 11 彙總這些屬性的應用程式特有資訊。

表 11. 使用者定義通訊協定內容屬性的應用程式特有資訊：名稱=值配對內容

名稱	值	說明
<code>ws_prop_name</code> (不區分大小寫；如果沒有指定，就會使用屬性名稱來作為內容名稱)	任何有效的通訊協定內容名稱	此為通訊協定內容的名稱。有些供應商會保留部分內容，以提供更多功能。一般而言，除非您是要搜尋對這些供應商特定功能的存取權，否則不該定義以 <code>JMS</code> (若為 <code>JMS</code> 通訊協定) 開頭的自訂內容。
<code>ws_prop_type</code> (不區分大小寫，若為 <code>JMS</code> ，則為選用項目 - 若未指定，則會假設為 <code>String</code> ；因為只有 <code>String</code> 類型是有意義的，所以與 <code>HTTP(S)</code> 無關)	<code>String</code> 、 <code>Integer</code> 、 <code>Boolean</code> 、 <code>Float</code> 、 <code>Double</code> 、 <code>Long</code> 、 <code>Short</code>	通訊協定內容的類型。若為 <code>JMS</code> 通訊協定， <code>JMS API</code> 提供好幾種方法，可用來設定「 <code>JMS</code> 訊息」中的內容值： <code>setIntProperty</code> 、 <code>setLongProperty</code> 、 <code>setStringProperty</code> 等等。這裡所指定之 <code>JMS</code> 內容的類型，可指定要使用哪一種方法來設定訊息中的內容值。

若所給定的自訂內容 `ASI` (`ws_prop_name` 或 `ws_prop_type`) 無效，且沒有可處理此標頭的邏輯方法 (例如忽略 `HTTP` 處理程序的內容類型)，連接器就會記載警告，並忽略此內容。若對 `ws_prop_name` 或 `ws_prop_type` 執行必要的檢查之後，無法設定也無法擷取自訂內容的值，連接器就會記載此錯誤，且事件失敗。

若指定 `UserDefinedProperties` 屬性，連接器將建立 `UserDefinedProperties` 商業物件的實例。然後連接器會嘗試從訊息擷取內容值，並將其儲存於商業物件中。如果至少成功地擷取了一個內容值，連接器就會將修改過的 `UserDefinedProperties` 屬性設定於 `Protocol Config MO` 中。

就同步事件處理程序而言，若指定了 `UserDefinedProperties` 屬性，且已將其商業物件實例化，連接器會處理此子項商業物件的每個屬性，並依情況來設定訊息內容值。

**事件處理程序的 HTTP 認證傳送:** 為了傳送認證，連接器可支援 `HTTP Protocol Config MO` 中的 `Authorization_UserId` 及 `Authorization_Password` 屬性。該支援僅限於 `HTTP` 基礎鑑別方法中，用來傳送這些認證。

若 `SOAP/HTTP` 或 `SOAP/HTTPS` 通訊協定接聽器處理包含授權標頭的 `SOAP/HTTP` Web 服務要求，接聽器就會剖析該標頭，判斷它是否符合 `HTTP` 基礎鑑別。若符合，接聽器就會擷取使用者名稱及密碼，並使用 `Base64` 來解碼。解碼後的字串包含以冒號隔開的

使用者名稱及密碼。若通訊協定接聽器於 Protocol Config MO 中找到 Authorization\_UserID 及 Authorization\_Password 屬性，接聽器便會以事件授權標頭所擷取的那些值來設定這二個值。

## 標頭儲存區商業物件

圖 10 顯示展開後的標頭儲存區屬性 OrderHeader。

	Pos	Name	Type	Key	Required	Card	App Spec Info
	1	Request	SERVICE_SYNCH_OrderStatus_	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=request
1.1	1.1	OrderId	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
1.2	1.4	SOAPJMSCfgMO	SERVICE_SYNCH_SOAP_JMS_	<input type="checkbox"/>	<input type="checkbox"/>	1	
1.3	1.5	ObjectEventId	String				
1.4	1.2	OrderHeader	SERVICE_SYNCH_OrderStatus_	<input type="checkbox"/>	<input type="checkbox"/>	1	soap_location=SOAPHeader
1.4.1	1.2.1	transaction	SERVICE_SYNCH_OrderStatus_	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	headerFault=transactionFault;elem_ns=http://www.mycompany.com/samples/transaction;type_name=Transaction_HeaderChild
1.4.1.1	1.2.1.1	TransactionId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
1.4.1.2	1.2.1.1	ObjectEventId	String				
1.4.1.3	1.2.1.1	actor	String	<input type="checkbox"/>	<input type="checkbox"/>		attr_name=actor
1.4.1.4	1.2.1.1	mustUnderstand	String	<input type="checkbox"/>	<input type="checkbox"/>		attr_name=mustUnderstand
1.4.1.5	1.2.3	ObjectEventId	String				
1.4.3	1.2.2	affiliate	SERVICE_SYNCH_OrderStatus_	<input type="checkbox"/>	<input type="checkbox"/>	1	headerFault=affiliateFault;elem_ns=http://www.mycompany.com/samples/partner;type_name=TradingPartner_HeaderChild
1.4.3.1	1.2.2.1	partnerId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
1.4.3.2	1.2.2.1	ObjectEventId	String				
1.4.3.3	1.2.2.1	routingNumber	String	<input type="checkbox"/>	<input type="checkbox"/>		
1.5	1.3	SOAPCfgMO	SERVICE_SYNCH_OrderStatus_	<input type="checkbox"/>	<input type="checkbox"/>	1	
2	2	Response	SERVICE_SYNCH_OrderStatus_	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=response
3	3	Fault	SERVICE_SYNCH_OrderStatus_	<input type="checkbox"/>	<input type="checkbox"/>	1	ws_botype=fault

圖 10. 標頭儲存區及子項商業物件

標頭儲存區屬性亦稱為 SOAP 標頭屬性，其對應於只包含子項商業物件的商業物件。每個子項各代表 SOAP 訊息中的一個標頭項目。於圖 10 所顯示的範例中，要求標頭儲存區為 OrderHeader。SOAP 標頭屬性具有 SOAP 資料處理常式所需的應用程式特有資訊 (ASI)。例如，「標頭」儲存區商業物件就是以其 ASI 來識別的：soap\_location=SOAPHeader。如需有關標頭處理程序的相關資訊，請參閱第 103 頁的『SOAP 資料處理常式處理程序』。

所有 SOAP 商業物件 (無論是「要求」、「回應」或「錯誤」物件)，都有一個而且只有一個標頭儲存區。

## 標頭子項商業物件

於圖 10 所顯示的範例中，要求標頭儲存區 (OrderHeader) 的二個子屬性為 1) SERVICE\_SYNCH\_OrderStatus\_TransactionHeaderChild 類型的交易，以及 2) SERVICE\_SYNCH\_OrderStatus\_TradingPartnerHeaderChild 類型的成員。這二種屬性對應於標頭子項商業物件。各代表 SOAP 訊息中的單一標頭元素。標頭元素為 SOAP 訊息

之 SOAP-Env:Header 元素的直接子項。如圖 10 所示，標頭子項商業物件可能有 actor 及 mustUnderstand 屬性。這二個屬性對應於 SOAP 標頭元素的 actor 及 mustUnderstand 屬性。如需有關標頭處理程序的相關資訊，請參閱第 103 頁的『SOAP 資料處理常式處理程序』。

可依需要，使用多個標頭子項物件來代表 SOAP 標頭訊息元素。

## 處理 TLO 的非同步事件

圖 11 顯示非同步事件處理程序的商業物件階層。僅需要一個要求物件。

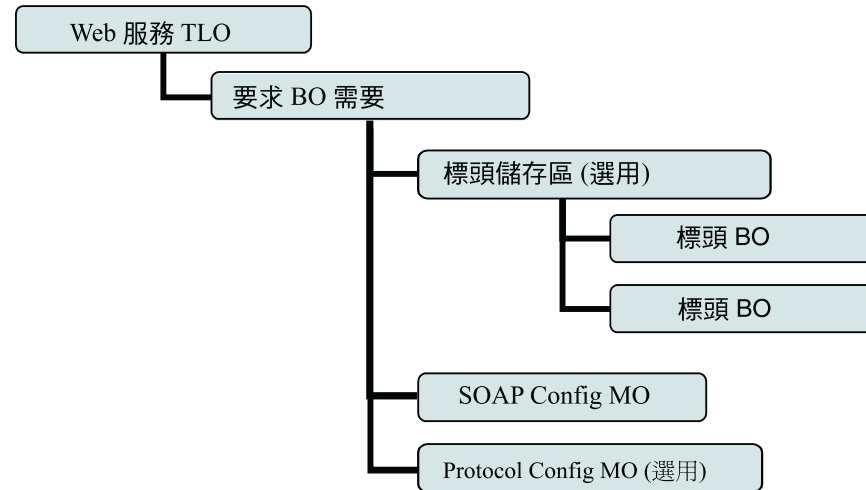


圖 11. 非同步事件處理程序的商業物件階層

TLO 包含物件層次 ASI 以及具有屬性層次 ASI 的屬性。這二種 ASI 的說明如下。如需有關標頭儲存區及標頭子項商業物件的資訊，請參閱第 30 頁的『標頭儲存區商業物件』。

### 非同步事件處理程序 TLO 的物件層次 ASI

物件層次 ASI 提供有關 TLO 本質及其所包含之物件的基本資訊。圖 12 顯示 SERVICE\_ASYNCH\_TLO\_Order 的物件層次 ASI，此為非同步事件處理程序的 TLO 範例。

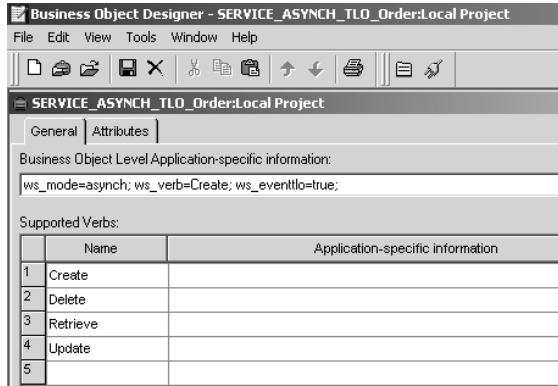


圖 12. 非同步事件處理程序的最上層商業物件

以下的表 4 說明處理 TLO 的非同步事件的物件層次 ASI。

表 12. 處理 TLO 物件 ASI 的非同步事件

物件層次 ASI	說明
ws_eventtlo=true	若此 ASI 內容的設定為 true，連接器就會將此物件當成事件處理程序的 TLO。  請注意，「WSDL 配置精靈」使用此 ASI 來判斷商業物件是否為 TLO。如需詳細資訊，請參閱第 134 頁的『WSDL 配置精靈』。
ws_verb=verb	在將 TLO 傳遞至協同作業之前，連接器會使用此 ASI 來設定 TLO 上的動詞。於圖 12 所顯示的範例中，動詞為 Create。
ws_mode=asynch	於事件通知期間，連接器會使用這個 ASI 內容，來決定要以同步 (synch) 或非同步 (asynch) 的方式呼叫協同作業。若為非同步的處理程序，此 ASI 必須設定為 asynch。  預設值為 asynch。

**註：** 與同步事件處理程序不同的是，非同步事件處理程序的 TLO 層次不需要協同作業名稱 ASI。且整合分配管理系統可確保應用程式事件可以到達所有訂閱的協同作業。

### 非同步事件處理程序 TLO 的屬性層次 ASI

每個處理 TLO 的非同步事件，皆有一個單一屬性對應於「要求」商業物件。圖 13 顯示 SERVICE\_ASYNC\_TLO\_Order 的要求屬性 (TLO 範例) 及該屬性的 ASI。

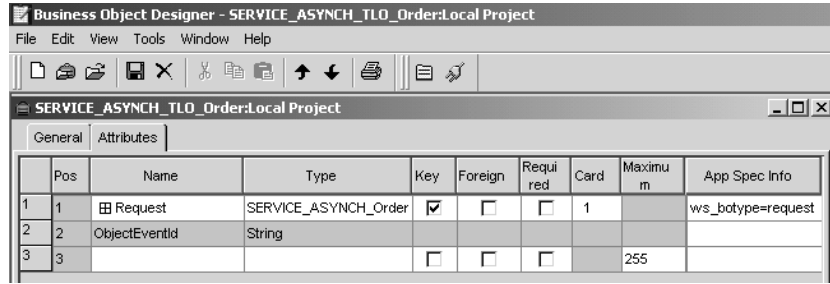


圖 13. 非同步事件處理程序的 TLO 屬性

表 13 為非同步事件處理程序 TLO 的要求屬性，彙總屬性層次 ASI。

表 13. 處理 TLO 屬性 ASI 的非同步事件

TLO 屬性	屬性層次 ASI	說明
Request	ws_botype=request	此屬性對應於 Web 服務要求。連接器使用其 ASI 來判斷這個 TLO 屬性的類型是否為「SOAP 要求 BO」。這個 ASI (不是屬性名稱) 可判斷屬性類型。若有多個要求屬性，連接器就會使用第一個屬性的 ASI。  此為同步事件處理程序 TLO 的必要屬性。

### 非同步事件處理程序的要求商業物件

「要求」商業物件是 TLO 的子項，為非同步事件處理程序的必要物件。您可以為「要求」商業物件指定預設動詞。若要這麼做，請將：

DefaultVerb=true;

指定於 ASI 欄位中，以作為「要求」商業物件最上層之「受支援動詞」清單中的動詞。若未指定 DefaultVerb ASI，且資料處理常式是於未設定動詞的情況下處理商業物件，則傳回商業物件時，就沒有動詞。非同步事件處理程序的「要求」商業物件的物件層次 ASI 說明位於表 14 中。

表 14. 非同步的事件處理程序：「要求」商業物件的物件層次 ASI

物件層次 ASI	說明
cw_mo_soap=SOAPCfgMO	此 ASI 的值必須符合對應於 SOAP Config MO 之屬性的名稱。這是 SOAP Config MO，可針對「要求」商業物件來定義資料處理常式轉換作業。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。
cw_mo_jms=SOAJMSCfgMO 或 cw_mo_http=SOAPHTTPCfgMO	此 ASI 的值必須符合對應於 Protocol Config MO 之屬性的名稱。第一個 ASI 指定 SOAP/JMS 通訊協定接聽器；第二個 ASI 指定 SOAP/HTTP 或 SOAP/HTTPS 通訊協定接聽器。ASI 和 Protocol Config MO 皆為選用性的。如需進一步資訊，請參閱第 25 頁的『Protocol Config MO』。

表 14. 非同步的事件處理程序：「要求」商業物件的物件層次 ASI (繼續)

物件層次 ASI	說明
ws_tloname=tloname	此 ASI 指定此物件所屬之 Web 服務 TLO 的名稱。於事件處理程序期間，連接器會使用此 ASI 來判斷資料處理常式所遞送的「要求」商業物件是否為 TLO 的子項。如果是，連接器就會建立所指定的 TLO，將「要求」商業物件設定為其子項，並使用 TLO 物件層次 ASI，將它傳遞至訂閱協同作業。

於圖 14 所顯示的範例中，Request 屬性包含 SOAP Config MO 和標頭儲存區 (OrderHeader)，以及與內容相關的屬性 (OrderLineItems)。SOAP Config MO、Protocol Config MO、SOAP 標頭儲存區及標頭子項商業物件的基本需求及性質，在非同步事件處理程序及同步事件處理程序中，都是一樣的。如需進一步資訊，請參閱第 20 頁的『處理 TLO 的同步事件』中的上述這些主題。

Pos	Name	Type	Key	Required	Card	Maximum	App Spec Info
1	Request	SERVICE_ASYNC_Order	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1		ws_botype=request
1.1	OrderId	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		255	
1.2	OrderDate	Date	<input type="checkbox"/>	<input type="checkbox"/>			
1.3	CustomerId	String	<input type="checkbox"/>	<input type="checkbox"/>		255	
1.4	OrderLineItems	SERVICE_ASYNC_Order_LineItem	<input type="checkbox"/>	<input type="checkbox"/>	N		type_name=Order_LineItem
1.5	OrderHeader	SERVICE_ASYNC_Order_Header	<input type="checkbox"/>	<input type="checkbox"/>	1		soap_location=SOAPHeader
1.6	SOAPConfigMO	SERVICE_ASYNC_Order_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	1		
1.7	SOAPJMSConfigMO	SERVICE_ASYNC_SOAP_JMS_Order_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	1		
1.8	ObjectEventId	String					
2	ObjectEventId	String					
3			<input type="checkbox"/>	<input type="checkbox"/>		255	

圖 14. 非同步事件處理程序的 Request 屬性

## 處理非 TLO 的事件

若商業物件中沒有物件層次 ASI ws\_eventtlo=true，連接器就會認定該商業物件不是 TLO。於事件處理程序期間，連接器可以處理非 TLO--通用商業物件和應用程式特定的商業物件。若包含非 TLO，則相同的商業物件代表「要求」和「回應」商業物件。

非 TLO 並無 SOAP Config MO。當您將協同作業顯現為 Web 服務時，「WSDL 配置精靈」將配置連接器的 WSCollaborations 內容。連接器會使用 WSCollaborations 內容來判斷要求訊息的 BodyName 和 BodyNS。請注意，就非 TLO 而言，WSCollaborations 內容是用來進行商業物件解析的。

使用非 TLO 的好處，就是您不需要開發新的 TLO 結構商業物件來配合使用您的 Web 服務解決方案。但是 TLO 可使用較準確且經濟的方式來顯現資料--客戶、公司及其他。TLO 商業物件本身也比非 TLO 提供更大的自訂空間。

欲進一步瞭解使用非 TLO 來作為「WSDL 配置精靈」輸入值的基本需求，請參閱第 132 頁的『識別或開發商業物件』。

## 處理 TLO 的同步要求

就要求處理程序而言，連接器可允許兩種 TLO--同步及非同步。本節將討論處理 TLO 的同步要求。

圖 15 顯示同步要求處理程序的 TLO 商業物件階層。「要求」和「回應」物件是必要的，「錯誤」物件是選用性的。和事件處理程序不同的是，「要求」物件需要 Protocol Config MO，而對「回應」和「錯誤」物件而言，Protocol Config MO 是選用性的。如需有關標頭儲存區及標頭子項商業物件的資訊，請參閱第 30 頁的『標頭儲存區商業物件』。

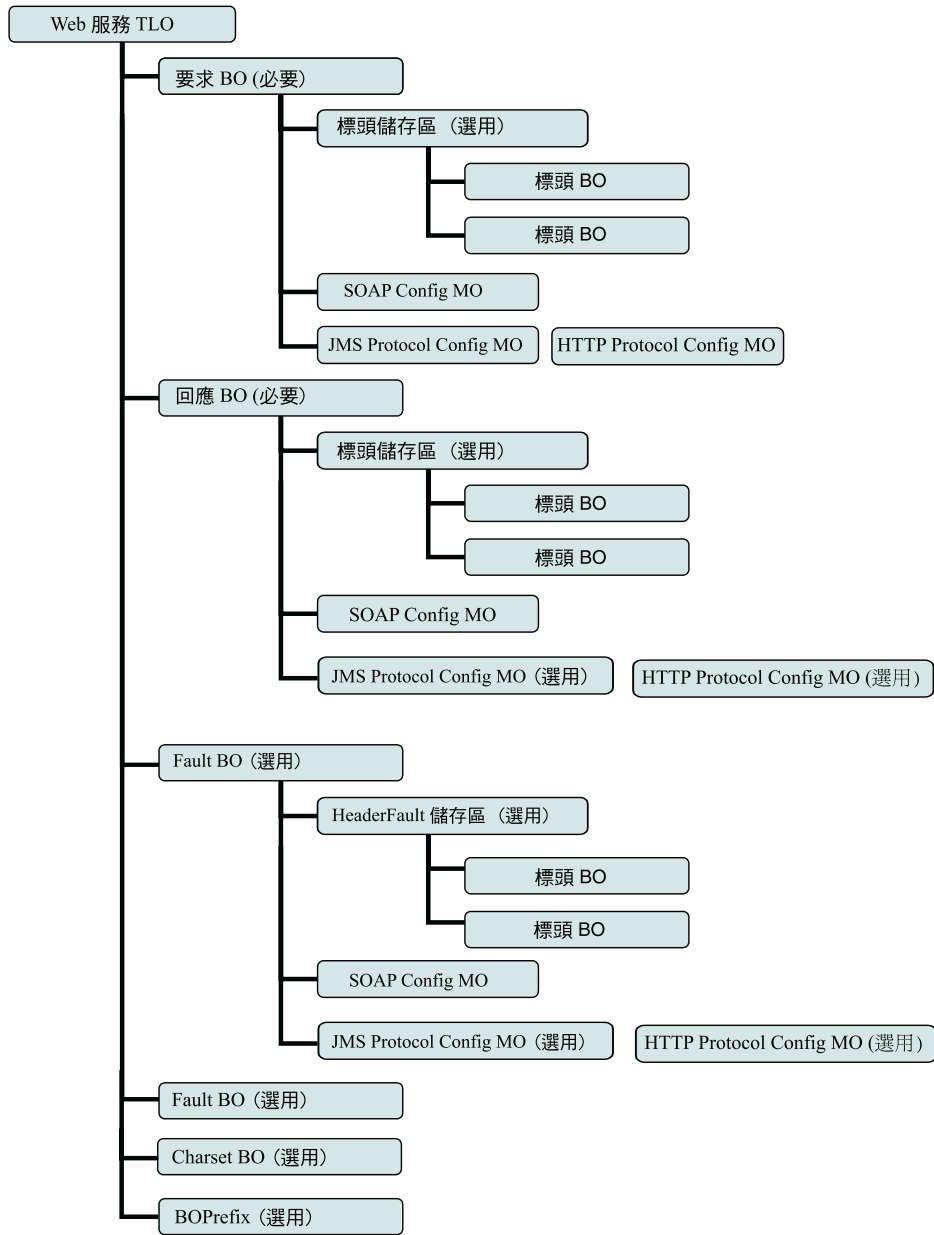


圖 15. 同步要求處理程序的商業物件階層

### 同步要求處理程序 TLO 的物件層次 ASI

物件層次 ASI 提供有關 TLO 本質及其所包含之物件的重要資訊。圖 16 顯示 CLIENT\_SYNCH\_TLO\_OrderStatus (同步要求處理程序的 TLO 範例)。



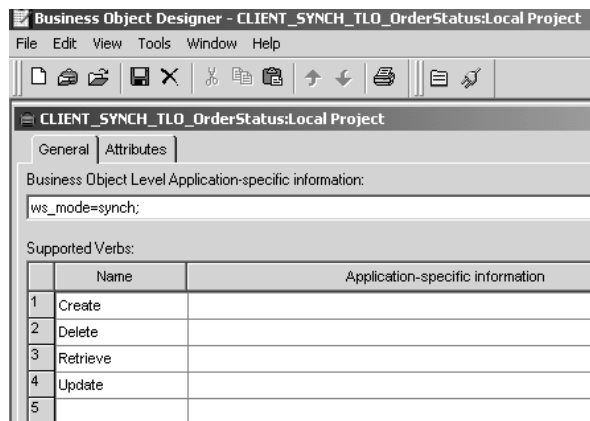


圖 16. 同步要求處理程序的最上層商業物件

表 15 說明同步要求處理程序的物件層次 ASI。其與同步事件處理程序 TLO 的 ASI 不同，這個層次的要求處理程序並不需要 ws\_collab、ws\_verb 或 ws\_eventtlo ASI。

表 15. 處理 TLO 物件 ASI 的同步要求

物件層次 ASI	說明
ws_mode=synch	<p>於要求處理程序期間，連接器會使用這個 ASI 內容，來決定要以同步 (synch) 或非同步 (asynch) 的方式呼叫 Web 服務。若指定 synch，連接器就會預期有回應，且 TLO 必須包含要求和回應商業物件，及選用性的一或數個錯誤物件。</p> <p>預設值為 asynch。</p>

## 同步要求處理程序 TLO 的屬性層次 ASI

圖 17 顯示 CLIENT\_SYNCH\_TLO\_OrderStatus TLO 的屬性，以及屬性層次 ASI。

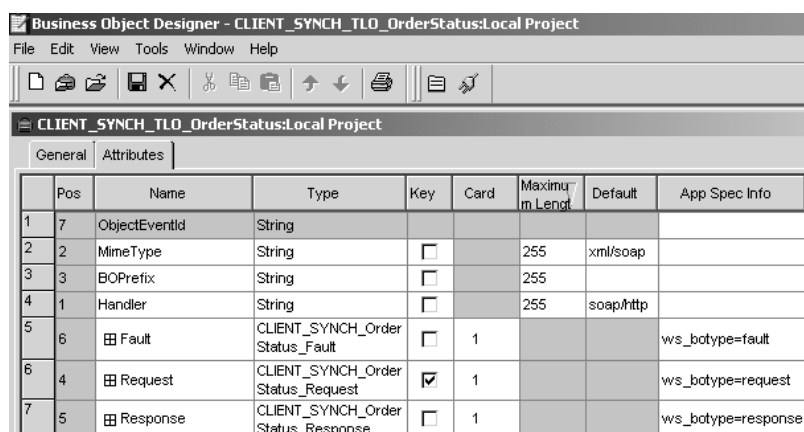


圖 17. 同步要求處理程序的 TLO 屬性

表 16 說明圖 17 中所顯示的屬性及 ASI。

表 16. 要求處理程序 TLO 屬性

TLO 屬性	屬性層次 ASI	說明
MimeType	無	此屬性指定連接器所呼叫之資料處理常式的 MIME 類型，以將「要求」商業物件轉換為要求訊息。這個值可用來將同步回應/錯誤訊息轉換為商業物件，視「訊息轉換規則」配置而定。
BOPrefix	無	類型 String 的這個屬性會傳遞至資料處理常式。
處理常式	無	<p>此屬性指定要用來處理 Web 服務要求的通訊協定處理常式，且僅限用於要求處理程序。其使用下列其中一個值：</p> <ul style="list-style-type: none"> <li>• soap/jms 連接器使用 SOAP/JMS 通訊協定處理常式來處理要求。</li> <li>• soap/http 連接器使用 SOAP/HTTP、SOAP/HTTPS 通訊協定處理常式來處理此 Web 服務要求。</li> </ul> <p>預設值為 soap/http</p>
Charset		<p>類型 String 的這個選用性參數可指定將「要求」商業物件轉換為訊息時，要在資料處理常式上設定的 Charset。</p> <p>附註：在這個屬性中指定的 Charset 值將不會在要求訊息的 Content-Type 通訊協定標頭中傳送。</p>
Request	ws_botype=request	此屬性對應於 Web 服務要求商業物件。連接器會使用此屬性 ASI 來判斷這個 TLO 屬性的類型是否為「SOAP 要求 BO」。這個 ASI (不是屬性名稱) 可判斷屬性類型。若有多個要求屬性，連接器就會使用第一個移入屬性的 ASI。
Response	ws_botype=response	此屬性對應於協同作業所傳回的回應，且為同步要求處理程序的必要的屬性。連接器會使用這個屬性 ASI 來判斷此 TLO 屬性的類型是否為「SOAP 回應 BO」。這個 ASI (不是屬性名稱) 可判斷屬性類型。

表 16. 要求處理程序 TLO 屬性 (繼續)

TLO 屬性	屬性層次 ASI	說明
Fault	ws_botype=fault 或 ws_botype=defaultfault	此屬性為同步要求處理程序的選用性屬性，其對應於當處理程序無法順利移入回應時，Web 服務所傳回的錯誤訊息。  連接器會使用這個 ASI 來判斷 TLO 的屬性類型是否為「SOAP 錯誤 BO」。這個 ASI (不是屬性名稱) 可判斷屬性類型。若錯誤訊息為 detail 元素，就會傳回 defaultfault 商業物件。預設的商業物件解析會用到 defaultfault。如需進一步資訊，請參閱第 97 頁的第 5 章, 『SOAP 資料處理常式』。

## 同步要求處理程序的「要求」商業物件

「要求」商業物件是 TLO 的子項，為同步要求處理程序的必要物件。「要求」商業物件具有物件層次 ASI。

假設您開啓 CLIENT\_SYNCH\_OrderStatus\_Request，並按一下「一般」標籤，就會顯示像圖 18 中的物件層次 ASI。

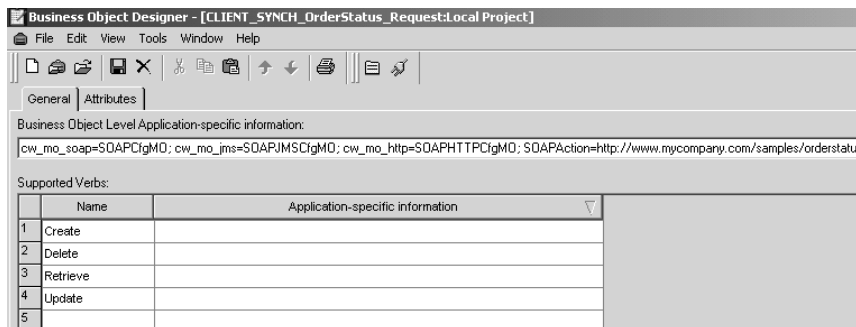


圖 18. 同步要求處理程序的要求物件 ASI

表 17 說明同步要求處理程序之「要求」商業物件的物件層次 ASI。

表 17. 同步的要求處理程序：「要求」商業物件的物件層次 ASI

物件層次 ASI	說明
cw_mo_soap=SOAPCfgMO	此 ASI 的值必須符合對應於 SOAP Config MO 之屬性的名稱。這是 SOAP Config MO，可針對「要求」商業物件來定義資料處理常式轉換作業。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。

表 17. 同步的要求處理程序：「要求」商業物件的物件層次 ASI (繼續)

物件層次 ASI	說明
<code>cw_mo_jms=SOAPJMSCfgMO</code>	此 ASI 的值必須符合對應於 Protocol Config MO 之屬性的名稱。此 Protocol Config MO 可用來指定 JMS 通訊協定處理常式的目的地 Web 服務。如需進一步資訊，請參閱第 41 頁的『要求處理程序之「要求」商業物件的 JMS Protocol Config MO』。
<code>cw_mo_http=SOAPHTTPCfgMO</code>	此選用性 ASI 的值必須符合對應於 Protocol Config MO 之屬性的名稱。此為另一個 Protocol Config MO，可用來指定 SOAP/HTTP-HTTPS 通訊協定處理常式的目的地。SOAP/HTTP 和 SOAP/HTTPS 通訊協定處理常式將會使用此 ASI。請注意，TLO 要求屬性必須要有 JMS 或 HTTP Protocol Config MO，才能進行要求處理程序，至於要有 JMS 還是 HTTP Protocol Config MO，需視您所使用的 Web 服務通訊協定類型而定。如需進一步資訊，請參閱第 41 頁的『要求處理程序的 HTTP Protocol Config MO』。
<code>SOAPAction=SOAPActionURI</code>	連接器使用這個 ASI 來判斷是否要在要求訊息上設定 SOAPAction 標頭。只有當目標 Web 服務需要 SOAPAction 標頭時，才可以指定此 ASI。請注意，此 ASI 是要用在要求處理程序，而非事件通知。

## 同步要求處理程序的「回應」商業物件

「回應」商業物件是 TLO 的子項，為同步要求處理程序的必要物件。同步要求處理程序的「回應」商業物件的物件層次 ASI 說明位於表 18 中。

表 18. 同步的要求處理程序：回應商業物件的物件層次 ASI

物件層次 ASI	說明
<code>cw_mo_soap=SOAPCfgMO</code>	此 ASI 的值必須符合對應於 SOAP Protocol Config MO 之屬性的名稱。這是 SOAP Config MO，可針對「回應」商業物件來定義資料處理常式轉換作業。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。
<code>cw_mo_jms=SOAPJMSCfg MO</code> 或 <code>cw_mo_http=SOAPHTTPCfgMO</code>	此 ASI 的值必須符合對應於 Protocol Config MO 之屬性的名稱。此 Protocol Config MO (「回應」商業物件的選用項目) 可為 JMS 或 HTTP(s) 通訊協定處理常式指定回應 SOAP 訊息中的標頭。如需進一步資訊，請參閱第 25 頁的『Protocol Config MO』。

您可為「回應」商業物件指定預設動詞。若要這麼做，請將：

```
DefaultVerb=true;
```

指定於 ASI 欄位中，以作為「回應」商業物件最上層之「受支援動詞」清單中的動詞。若未指定 DefaultVerb ASI，且資料處理常式是在沒有設定動詞的情況下處理商業物件，則傳回「回應」商業物件時，就沒有動詞。

## 同步要求處理程序的「錯誤」商業物件

「錯誤」商業物件是 TLO 的子項，為同步要求處理程序的選用性物件。同步要求處理程序的「錯誤」商業物件的物件層次 ASI 說明位於表 8 中。

表 19. 同步的要求處理程序：「錯誤」商業物件的物件層次 ASI

物件層次 ASI	說明
cw_mo_soap=SOAPCfgMO	此 ASI 的值必須符合對應於 SOAP Protocol Config MO 之屬性的名稱。此為 SOAP Config MO，可針對「錯誤」商業物件來定義資料處理常式轉換作業。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。
cw_mo_jms=SOAJMSCfg MO 或 cw_mo_http=SOAPHHTPCfgMO	此 ASI 的值必須符合對應於 Protocol Config MO 之屬性的名稱。這個 Protocol Config MO (「錯誤」商業物件的選用項目) 可為 JMS 通訊協定處理常式指定回應 SOAP 訊息中的標頭。如需進一步資訊，請參閱第 25 頁的『Protocol Config MO』。

## SOAP Config MO

SOAP Config MO (SOAPCfgMO) 的屬性和事件處理程序 SOAP Config MO 的屬性相同。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』，以及第 99 頁的『SOAP 配置 meta 物件：每個 SOAP 商業物件的子項』。

## 要求處理程序之「要求」商業物件的 JMS Protocol Config MO

當您使用 JMS Web 服務時，「要求」商業物件必須要有 JMS Protocol Config MO，對「回應」和「錯誤」物件而言，Protocol Config MO 是選用性的。表 20 說明要求處理程序 JMS Protocol Config MO--Destination 是最重要也是唯一的必要屬性。JMS 通訊協定處理常式會使用此屬性尋找所要求的 Web 服務。此外，第 25 頁的『Protocol Config MO』中有關 JMS Config MO 的所有屬性，都是選用性的。

表 20. 要求處理程序的 JMS Protocol Config MO 屬性

屬性	必要的	類型	說明
Destination	是	String	目標 Web 服務的目的地佇列名稱。「JMS 通訊協定處理常式」會使用此屬性來判斷 Web 服務的目的地。若連接器特有的 JNDI 內容 LookupQueuesUsingJNDI 設定為 true，「JMS 通訊協定處理常式」就會使用 JNDI 來查閱此佇列。請確定此屬性有提供目的地佇列的 JNDI 名稱。

## 要求處理程序的 HTTP Protocol Config MO

於要求處理程序期間，SOAP/HTTP-HTTPS 通訊協定處理常式會使用 HTTP Protocol Config MO 來決定目標 Web 服務的目的地。「要求」商業物件必須要有這個 Protocol Config MO。SOAP/HTTP-HTTPS 通訊協定處理常式僅支援 HTTP 1.0 POST 要求。如第 42 頁的表 21 所示，唯一需要的屬性 (Destination) 是目標 Web 服務的完整 URL。下面的部份則說明選用性的授權屬性。

表 21. 要求處理程序的 HTTP Protocol Config MO 屬性

屬性	必要的	類型	說明
Destination	是	String	目標 Web 服務的目的地 URL。SOAP/HTTP-HTTPS 通訊協定處理常式會使用此屬性來判斷 Web 服務的目的地。
Content-Type	「要求」商業物件的必要屬性，對其他物件則是選用性的。	String	此屬性的值可定義外送訊息的 Content-Type 標頭 (包括外送訊息的訊息 ContentType，及選用性的 charset)。語法與 HTTP 通訊協定中的 Content-Type 標頭語法相同，例如：text/html; charset=ISO-8859-4。若未定義 Content-Type 屬性，連接器就會使用 text/xml 作為訊息的 ContentType。
Authorization_UserID	否	String	此屬性對應於 HTTP 基礎鑑別的 userID。如需進一步資訊，請參閱第 44 頁的『要求處理程序的 HTTP 認證傳送』。
Authorization_Password	否	String	此屬性對應於 HTTP 基礎鑑別的密碼。如需進一步資訊，請參閱第 44 頁的『要求處理程序的 HTTP 認證傳送』。
一或數個 HTTP 標頭	否	String	此屬性可讓處理常式傳遞或擷取所指定之 HTTP 標頭的值。
UserDefinedProperties	否	商業物件	此屬性會保留使用者定義的通訊協定內容商業物件。如需進一步資訊，請參閱第 43 頁的『要求處理程序的使用者定義內容』。
MessageTransformationMap	否	單一基數商業物件	此屬性指向含有 0 個以上訊息轉換規則的商業物件。這些規則包含有關 MIME 類型及 charset 的資訊，以套用於規則中所指定的送入訊息。如需進一步資訊，請參閱第 43 頁的『訊息轉換對映』。

圖 19 顯示「商業物件設計程式」中的部分 HTTP Protocol Config MO 屬性。

General		Attributes								
Pos	Name	Type	Key	Foreign	Required	Card	Maximum	Default	App Spec Info	
1	1	OrderId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255		
2	2	<input checked="" type="checkbox"/> OrderHeader	CLIENT_SYNCH_OrderStatus_Request_Header	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1			
3	3	<input checked="" type="checkbox"/> HTTPCfgMO	CLIENT_SYNCH_OrderStatus_HTTPCfgMO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1			
3.1	3.1	Date	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.2	3.2	Content-Type	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.3	3.3	<input checked="" type="checkbox"/> MessageTransformationMap	HTTP_CfgMo_MsgTrnsfMap	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1			
3.3	3.3.1	<input checked="" type="checkbox"/> TransformationRule	HTTP_CfgMo_MsgTrnsfRule	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	N			
3.3	3.3.1.1	Content-Type	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255	/*	
3.3	3.3.1.2	MimeType	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.3	3.3.1.3	Charset	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.3	3.3.1.4	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
3.3	3.3.2	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
3.4	3.4	<input checked="" type="checkbox"/> UserDefinedProperties	HTTP_CfgMo_CustomProperties	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1			
3.4	3.4.1	CustomProperty1	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		
3.4	3.4.2	CustomProperty2	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	ws_prop_type=Integer	
3.4	3.4.3	CustomPropertyN	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	ws_prop_type=Boolean	
3.4	3.4.4	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
3.5	3.5	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
4	4	ObjectEventId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
5	5			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255		

圖 19. 要求處理程序的 HTTP Protocol Config MO 屬性

HTTP Protocol Config MO 屬性的說明位於下列各節中：

- 『要求處理程序的使用者定義內容』
- 『訊息轉換對映』
- 第 44 頁的『要求處理程序的 HTTP 認證傳送』

**要求處理程序的使用者定義內容:** 您可以自由指定 HTTP Protocol Config MO 中的自訂內容。若要這麼做，請併入 `UserDefinedProperties` 屬性。此屬性對應於商業物件，該商業物件中有一或數個子屬性具有內容值。此商業物件中的每個屬性都必須在訊息標頭的變數部分中，定義單一內容，以供讀取 (若為同步的回應，則是寫入)，如下所示：

- 無論通訊協定內容的類型為何，屬性的類型一律為 `String`。屬性的應用程式特有資訊可以包含二個名稱值組，定義屬性所對映之通訊協定訊息內容的名稱及格式。

表 22 彙總這些屬性的應用程式特有資訊。

表 22. 使用者定義通訊協定內容屬性的應用程式特有資訊：`name=value` 配對內容

名稱	值	說明
<code>ws_prop_name</code> (不區分大小寫；若未指定，就會使用屬性名稱來作為內容名稱)	任何有效的通訊協定內容名稱	此為通訊協定內容的名稱。有些供應商會保留部分內容，以提供更多功能。一般而言，除非您是要搜尋對這些供應商特定功能的存取權，否則不該定義以 <code>JMS</code> (若為 <code>JMS</code> 通訊協定) 開頭的自訂內容。
<code>ws_prop_type</code> (不區分大小寫，若為 <code>JMS</code> ，則為選用項目 - 若未指定，則會假設為 <code>String</code> ；因為只有 <code>String</code> 類型是有意義的，所以與 <code>HTTP(S)</code> 無關)	<code>String</code> 、 <code>Integer</code> 、 <code>Boolean</code> 、 <code>Float</code> 、 <code>Double</code> 、 <code>Long</code> 、 <code>Short</code>	通訊協定內容的類型。若為 <code>JMS</code> 通訊協定， <code>JMS API</code> 提供好幾種方法，可用來設定「 <code>JMS</code> 訊息」中的內容值： <code>setIntProperty</code> 、 <code>setLongProperty</code> 、 <code>setStringProperty</code> 等等。這裡所指定之 <code>JMS</code> 內容的類型，可指定要使用哪一種方法來設定訊息中的內容值。

若所給定的自訂內容 `ASI` (`ws_prop_name` 或 `ws_prop_type`) 無效，且沒有可處理此標頭的邏輯方法 (例如忽略 `HTTP` 處理程序的內容類型)，連接器就會記載警告，並忽略此內容。若對 `ws_prop_name` 或 `ws_prop_type` 執行必要的檢查之後，無法設定也無法擷取自訂內容的值，連接器就會記載錯誤，且事件失敗。

若指定了 `UserDefinedProperties` 屬性，且已將其商業物件實例化，連接器會處理這個子項商業物件的每個屬性，並依情況來設定訊息內容值。

若為同步的要求處理程序，在從 `Web` 服務/`URL` 接收回應訊息時，如果指定了 `UserDefinedProperties` 屬性，連接器就會建立 `UserDefinedProperties` 商業物件的實例，並嘗試從訊息擷取內容值，然後儲存於新的商業物件中。若至少成功擷取了一個內容值，連接器就會將修改過的 `UserDefinedProperties` 商業物件設定於 `Protocol Config MO` 中。

**訊息轉換對映:** 「訊息轉換對映 (`MTM`)」功能僅適用於要求處理程序 `HTTP(S)` 通訊協定處理常式。`MessageTransformationMap` 是 `Protocol Config MO` 中的選用屬性，指向一個商業物件。該商業物件包含以規則中所指定之 `MIME` 類型和的 `charset` 來轉換

訊息的規則。若它發現 (區分大小寫) 屬性名稱 MessageTransformationMap，且此屬性的類型是商業物件 (請參閱圖 19)，連接器就會使用該物件中的規則來轉換訊息。

如圖 19 所示，MTM 屬性必須要有一個名為 TransformationRule 的基數 N 子項商業物件屬性。在嘗試尋找訊息的 TransformationRule 時，「SOAP/HTTP(s) 通訊協定處理常式」會先嘗試以所有 TransformationRule 中所指定的 ContentType 來詳細比對訊息。若不成功，連接器會嘗試尋找套用於多種訊息類型的規則。如需通訊協定處理常式處理程序的進一步資訊，請參閱第 66 頁的『SOAP/HTTP-HTTPS 通訊協定處理常式處理程序』。

每個 TransformationRule 商業物件的實例都必須指定表 23 中所顯示的屬性。

表 23. HTTP Protocol Config MO 中之 MessageTransformationMap 的 TransformationRule 屬性

屬性名稱	必要的	類型	預設值	說明
TransformationRule	否	商業物件、基數 N		此為包含 1 個訊息轉換規則的屬性。MessageTransformationMap 屬性之下可有 0 個以上的這個屬性實例。
+ContentType	是	String	*/*	這個內容的值可指定要套用此轉換規則之訊息的 HTTP ContentType。此屬性的預設值 */* 可讓連接器將此規則套用於任何 ContentType。如需通訊協定處理常式處理程序的進一步資訊，請參閱第 66 頁的『SOAP/HTTP-HTTPS 通訊協定處理常式處理程序』。請注意，若「通訊協定處理常式」發現有多個規則的 ContentType 與其他規則相同，「通訊協定處理常式」就會記載警告，並忽略所有重複的規則，但會使用唯一的規則。
+MimeType	否			在處理此商業物件中所指定之 ContentType 的訊息時，要用來呼叫資料處理常式的 MIME 類型。
+Charset	否			於轉換此商業物件中所指定之 ContentType 的要求時，所要使用的 charset。

**要求處理程序的 HTTP 認證傳送:** 為了傳送認證，連接器可支援 HTTP Protocol Config MO 中的 Authorization\_UserID 及 Authorization\_Password 屬性。該支援僅限在 HTTP 基礎鑑別方法中，用來傳送這些認證。

若於要求處理程序期間需要傳送認證，您必須手動將 Authorization\_UserID 及 Authorization\_Password 屬性新增至 WSDL ODA 所產生的 Protocol Config MO。您可於產生商業物件和 Meta 物件定義之後，在「商業物件設計程式」中執行此作業。(如需 WSDL ODA 的進一步資訊，請參閱第 129 頁的第 6 章，『啟用要求處理程序的協同作業』。)

協同作業會設定 Protocol Config MO 中的 Authorization\_UserID 和 Authorization\_Password 屬性值。若這些屬性不是空值也不是空白的，連接器就會在其傳送到目標 Web 服務的



要求上，建立授權標頭。於建立授權標頭時，SOAP HTTP/HTTPS 通訊協定處理常式會遵循 *HTTP Authentication: Basic and Digest Access Authentication (RFC 2617)*。

**註：**不支援摘要鑑別方法，亦不支援 Rfc2617 中所定義之 HTTP 鑑別的選用性盤查-回應機制。若 HTTP(s) 通訊協定處理常式呼叫的伺服器需要認證，連接器並不會等待伺服器的盤查回應，而會繼續傳送認證。

## 處理 TLO 的非同步要求

圖 20 顯示非同步要求處理程序的商業物件階層。只需要一個要求物件，且這個物件包含一個 SOAP 資料處理常式的 SOAP Config MO，以及二個 Protocol Config MO，SOAP/JMS 和 SOAP/HTTP/HTTPS 通訊協定處理常式各一個。下列各節將說明這些事項。

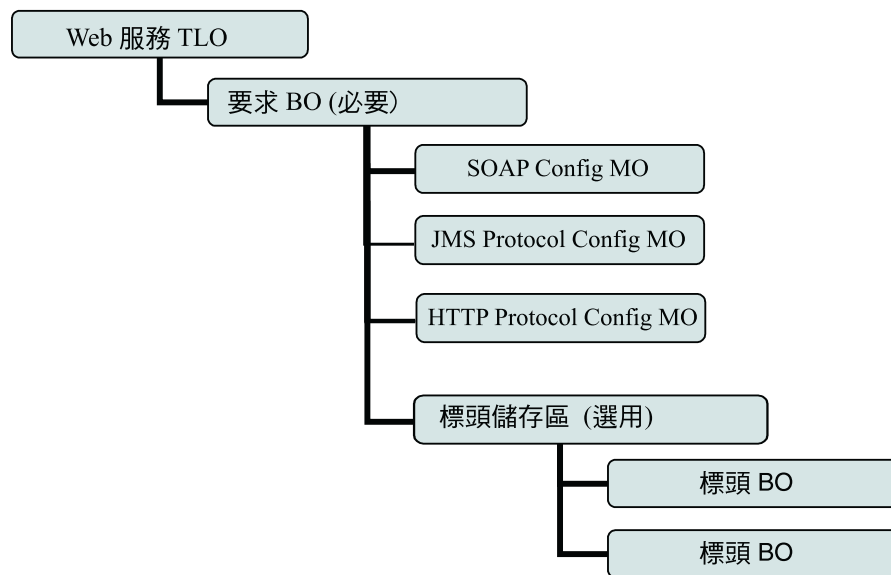


圖 20. 非同步要求處理程序的商業物件階層

TLO 包含物件層次 ASI 以及具有屬性層次 ASI 的屬性。這二種 ASI 的說明如下。如需有關標頭儲存區及標頭子項商業物件的資訊，請參閱第 30 頁的『標頭儲存區商業物件』。

## 非同步事件處理程序 TLO 的物件層次 ASI

圖 21 顯示 CLIENT\_ASYNC\_Order\_TLO (非同步要求處理程序的 TLO 範例)。

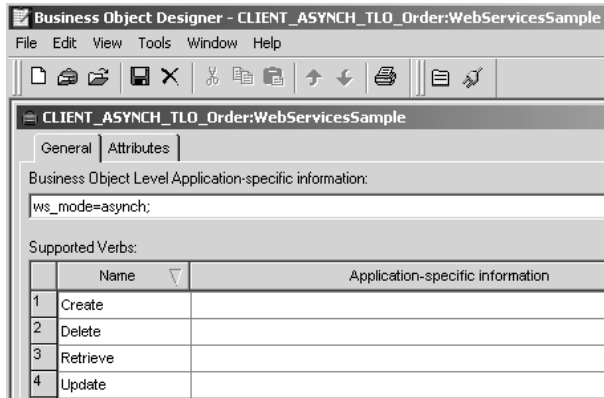


圖 21. 非同步要求處理程序的最上層商業物件

下面的表 24 說明非同步要求處理程序 TLO 的物件層次 ASI。

表 24. 處理 TLO 物件 ASI 的非同步要求

物件層次 ASI	說明
ws_mode=asynch	於要求處理程序期間，連接器會使用這個 ASI 內容，來決定要以同步 (synch) 或非同步 (asynch) 的方式呼叫協同作業。若為非同步的要求處理程序，此 ASI 必須設定為 asynch。  預設值為 asynch。

## 非同步要求處理程序 TLO 的屬性層次 ASI

圖 22 顯示 CLIENT\_ASYNC\_TLO\_Order 的屬性 (要求處理程序 TLO 範例)。

	Pos	Name	Type	Key	Required	Card	Maximum Length	Default	App Spec Info
1	1	Handler	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		255		soap/http
2	2	MimeType	String	<input type="checkbox"/>	<input type="checkbox"/>		255		xml/soap
3	3	BOPrefix	String	<input type="checkbox"/>	<input type="checkbox"/>		255		
4	4	Request	CLIENT_ASYNC_Order	<input type="checkbox"/>	<input type="checkbox"/>	1			ws_botype=request
5	5	ObjectEventId	String						

圖 22. 非同步要求處理程序的 TLO 屬性

表 25 為非同步要求處理程序 TLO 的要求屬性，彙總屬性層次 ASI。

表 25. 處理 TLO 屬性的非同步要求

TLO 屬性	屬性層次 ASI	說明
MimeType	無	此屬性指定連接器所呼叫之資料處理常式的 MIME 類型。請注意，此屬性僅適用「要求處理程序」。(若為事件處理程序，通訊協定接聽器會使用 SOAPDHMimeType 連接器特有的配置內容。) 預設值為 xml/soap。
BOPrefix	無	類型 String 的這個屬性要保留給將來開發之用，而且非必要屬性。
處理常式	無	此屬性指定要用來處理 Web 服務要求的通訊協定處理常式，且僅限用於要求處理程序。其使用下列其中一個值： <ul style="list-style-type: none"> <li>• soap/jms 連接器使用 SOAP/JMS 通訊協定處理常式來處理要求。</li> <li>• soap/http 連接器使用 SOAP/HTTP-HTTPS 通訊協定處理常式來處理此 Web 服務要求。</li> </ul> 預設值為 soap/http
Request	ws_botype=request	此屬性對應於 Web 服務要求商業物件。連接器會使用此屬性 ASI 來判斷這個 TLO 屬性的類型是否為「SOAP 要求 BO」。這個 ASI (不是屬性名稱) 可判斷屬性類型。若有多個要求屬性，連接器就會使用第一個屬性的 ASI。

### 非同步要求處理程序的要求商業物件

「要求」商業物件是 TLO 的子項，為非同步要求處理程序的必要物件。非同步要求處理程序的「要求」商業物件的物件層次 ASI 說明位於表 26 中。

表 26. 非同步的要求處理程序：「要求」商業物件的物件層次 ASI

物件層次 ASI	說明
cw_mo_soap=SOAPCfgMO	此 ASI 的值必須符合對應於 SOAP Config MO 之屬性的名稱。這是 SOAP Config MO，可針對「要求」商業物件來定義資料處理常式轉換作業。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。
cw_mo_jms=SOAPJMSCfgMO	此 ASI 的值必須符合對應於 Protocol Config MO 之屬性的名稱。此 Protocol Config MO 可用來指定 JMS 通訊協定處理常式的目的地 Web 服務。如需進一步資訊，請參閱第 41 頁的『要求處理程序之「要求」商業物件的 JMS Protocol Config MO』。

表 26. 非同步的要求處理程序：「要求」商業物件的物件層次 ASI (繼續)

物件層次 ASI	說明
cw_mo_http=SOAPHTTPCfgMO	此 ASI 的值必須符合對應於 Protocol Config MO 之屬性的名稱。此為另一個 Protocol Config MO，可用來指定 SOAP/HTTP-HTTPS 通訊協定處理常式的目的地。SOAP/HTTP-HTTPS 通訊協定處理常式將會使用此 ASI。請注意，TLO 要求屬性必須要有 JMS 和 HTTP Protocol Config MO，才可進行要求處理程序。如需進一步資訊，請參閱第 41 頁的『要求處理程序的 HTTP Protocol Config MO』。
SOAPAction=SOAPActionURI	連接器使用這個 ASI 來判斷是否要在要求訊息上設定 SOAPAction 標頭。只有當目標 Web 服務需要 SOAPAction 標頭時，才可以指定此 ASI。請注意，此 ASI 是要用在要求處理程序，而非事件通知。

於圖 14 所顯示的範例中，Request 屬性包含 SOAP Config MO 和標頭儲存區 (OrderHeader)，以及與內容相關的屬性 (OrderLineItems)。SOAP Config MO、Protocol Config MO、SOAP 標頭儲存區及標頭子項商業物件的基本需求及性質，在非同步要求處理程序及同步要求處理程序中，都是一樣的。如需進一步資訊，請參閱第 35 頁的『處理 TLO 的同步要求』中的上述這些主題。

Pos	Name	Type	Key	Required	Card	Maximum	App Spec Info
1	Request	SERVICE_ASYNC_Order	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1		ws_botype=request
1.1	OrderId	String	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		255	
1.2	OrderDate	Date	<input type="checkbox"/>	<input type="checkbox"/>			
1.3	CustomerId	String	<input type="checkbox"/>	<input type="checkbox"/>		255	
1.4	OrderLineItems	SERVICE_ASYNC_Order_LineItem	<input type="checkbox"/>	<input type="checkbox"/>	N		type_name=Order_LineItem
1.5	OrderHeader	SERVICE_ASYNC_Order_Header	<input type="checkbox"/>	<input type="checkbox"/>	1		soap_location=SOAPHeader
1.6	SOAPCfgMO	SERVICE_ASYNC_Order_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	1		
1.7	SOAPJMSCfgMO	SERVICE_ASYNC_SOAP_JMS_Order_CfgMO	<input type="checkbox"/>	<input type="checkbox"/>	1		
1.8	ObjectEventId	String					
2	ObjectEventId	String					
3			<input type="checkbox"/>	<input type="checkbox"/>		255	

圖 23. 非同步事件處理程序的 Request 屬性

## 非同步要求處理程序的 Config MO

SOAP Config MO (SOAPCfgMO) 的屬性和事件處理程序 SOAP Config MO 的屬性相同。如需進一步資訊，請參閱第 25 頁的『SOAP Config MO』。以及第 99 頁的『SOAP 配置 meta 物件：每個 SOAP 商業物件的子項』。

當您使用 JMS Web 服務時，「要求」商業物件必須要有 JMS Protocol Config MO。如需進一步資訊，請參閱第 41 頁的『要求處理程序之「要求」商業物件的 JMS Protocol Config MO』。

於要求處理程序期間，SOAP/HTTP-HTTPS 通訊協定處理常式會使用 HTTP Protocol Config MO 來決定目標 Web 服務的目的地。「要求」商業物件必須要有這個 Protocol Config MO。如需進一步資訊，請參閱第 41 頁的『要求處理程序的 HTTP Protocol Config MO』。

---

## 開發商業物件

您可使用「商業物件設計程式」來建立商業物件，並使用「連接器配置程式」來配置支援那些商業物件的連接器。如需有關「商業物件設計程式」工具的詳細資訊，請參閱 *Business Object Development Guide* 及第 131 頁的第 7 章，『將協同作業顯現為 Web 服務』。如需「連接器配置程式」的進一步資訊，請參閱第 171 頁的附錄 B，『連接器配置程式』。



---

## 第 4 章 Web 服務連接器

- 『連接器處理程序』
- 第 53 頁的『SOAP/HTTP(S) Web 服務』
- 第 54 頁的『SOAP/JMS Web 服務』
- 第 55 頁的『事件處理程序』
- 第 65 頁的『要求處理程序』
- 第 74 頁的『SSL』
- 第 72 頁的『連接器及 JMS』
- 第 76 頁的『配置連接器』
- 第 93 頁的『連接器啟動時』
- 第 94 頁的『記載』
- 第 94 頁的『追蹤』

本章說明 Web 服務連接器，及其配置方式。

所有 WebSphere 商業整合連接器都要搭配整合分配管理系統一起運作。Web 服務連接器需與 IBM WebSphere InterChange Server 整合分配管理系統搭配操作，相關資訊，請參閱 *IBM WebSphere InterChange Server 技術簡介*。

連接器是配接器的執行時期元件。連接器由一個特定應用程式元件和連接器組織架構組成。特定應用程式元件包含針對特定應用程式所量身訂做的程式碼。連接器組織架構的程式碼為所有連接器所通用，扮演整合分配管理系統與特定應用程式元件之間的中介角色。連接器組織架構在整合分配管理系統與特定應用程式之間提供下列服務：

- 接收和傳送商業物件
- 管理啟動訊息與管理訊息的交換

本書包含有關應用程式特有元件及連接器組織架構的資訊。這些元件統稱為連接器。

如需有關整合分配管理系統與連接器之間關係的詳細資訊，請參閱 *系統管理手冊*。

---

### 連接器處理程序

連接器包括一個用來處理事件的通訊協定接聽器組織架構，以及一個用來處理要求的通訊協定處理常式組織架構。這種雙向功能可讓連接器組織架構從事下列活動：

- 將協同作業當作 Web 服務來顯現，然後從 Web 服務用戶端來處理呼叫
- 處理呼叫 Web 服務之協同作業的要求

如需 SOAP 資料處理常式的進一步資訊，請參閱第 97 頁的第 5 章，『SOAP 資料處理常式』。

**註：** 連接器僅支援 SOAP/HTTP 及 SOAP/JMS 連結。

## 事件處理程序概觀

連接器事件處理程序 (或事件通知) 可用來處理 Web 服務用戶端的要求。此事件處理程序功能含有通訊協定接聽器組織架構，包括下列元件，本章稍後將詳細討論這些元件：

- SOAP/HTTP 通訊協定接聽器
- SOAP/HTTPS 通訊協定接聽器
- SOAP/JMS 通訊協定接聽器

連接器使用接聽器將協同作業作為 Web 服務來顯現，並且在傳輸器上接聽來自 Web 服務用戶端的呼叫，再傳送到顯現的協同作業。

SOAP/HTTP 及 SOAP/HTTPS 通訊協定接聽器將協同作業當作 SOAP/HTTP Web 服務來顯現。SOAP/JMS 通訊協定接聽器則將協同作業當作 SOAP/JMS Web 服務來顯現。

當 Web 服務用戶端的要求到達時，接聽器會將 SOAP 要求訊息轉換成商業物件，並呼叫協同作業。若為同步要求，連接器會收到與「要求」商業物件相同類型的「回應」商業物件。接聽器會將「回應」商業物件轉換成 SOAP 回應訊息，然後將 SOAP 回應訊息傳輸到 Web 服務用戶端。請注意，對此連接器而言，事件排序並非必要事項；連接器可使用任何順序來傳遞事件。

Web 服務連接器會利用 SOAP 資料處理常式，將送入的 SOAP 要求訊息轉換成商業物件。為了輔助資料處理常式來判斷要為送入的 SOAP 要求訊息解析哪個商業物件，連接器會提供有關其支援之商業物件的 meta 資訊給資料處理常式。連接器會先從其支援的商業物件來建立一個清單，其中包含可用於轉換作業之所有可能的商業物件。此清單可能會同時包含 TLO 及非 TLO。受支援的 TLO 商業物件，就是物件層次為 `ASIS_ws_eventtlo=true` 的那些商業物件。

若使用 TLO，通訊協定接聽器就會讀取 TLO 的 ASI 物件層次，如下所示：

- `ws_collab=` 此決定所要呼叫的協同作業
- `ws_mode=` 此決定呼叫協同作業的方式：同步 (synch) 或非同步 (asynch)

若使用非 TLO，則通訊協定接聽器會從「WSDL 配置精靈」所產生的 WSCollaborations 配置內容值，來讀取協同作業及處理模式。

連接器將 SOAP 要求中的 `BodyName` 和 `BodyNamespace` 與可能的商業物件名稱做比較，並嘗試比對。若為 TLO，則會使用「SOAP 要求」商業物件的 SOAP Config MO 內容來尋找這個 `BodyName/BodyNamespace` 配對。若為非 TLO，則會使用 WSCollaborations 連接器配置內容來尋找 `BodyName/BodyNamespace` 配對。(請注意，連接器只會考慮那些在 WSCollaborations 內容中擁有輸入項目的非 TLO。) 資料處理常式會使用 `BodyName/BodyNamespace` 配對來決定商業物件，以處理對商業物件轉換作業提出的 SOAP 要求。

連接器視察由 SOAP 資料處理常式傳回的「要求」商業物件。若此商業物件有 `ws_tlname` ASI，連接器就會在此 TLO 中設定「要求」商業物件。使用此 TLO 來呼叫協同作業。但是若未設定此 ASI，連接器就會使用 SOAP 資料處理常式所傳回的「要求」商業物件來呼叫協同作業。



若是執行同步協同作業，連接器會利用 SOAP 資料處理常式來建立 SOAP 回應或錯誤訊息，以傳回至用戶端。於此情況下，連接器只是將 SOAP 商業物件 (TLO 的子項) 或非 TLO 傳遞至資料處理常式。SOAP 資料處理常式會根據傳遞給它的商業物件，傳回 SOAP 訊息。

## 要求處理程序概觀

連接器可代表協同作業，透過 SOAP/HTTP(S) 及 SOAP/JMS 來呼叫 Web 服務。WSDL Object Discovery Agent (ODA) 及通訊協定處理常式組織架構皆可支援此要求處理功能。WSDL ODA 是設計期間所使用的工具，可用來產生包含目標 Web 服務相關資訊的 SOAP 商業物件。如需進一步資訊，請參閱第 129 頁的第 6 章，『啓用要求處理程序的協同作業』。通訊協定處理常式組織架構是一種可配置的執行時期模組，其包括下列元件，本章稍後將詳細討論這些元件：

- SOAP/HTTP-HTTPS 通訊協定處理常式
- SOAP/JMS 通訊協定處理常式

在接收協同作業「要求」商業物件時 (一律透過 WSDL ODA 設定於 TLO 中)，通訊協定處理常式組織架構會載入適當的通訊協定處理常式。通訊協定處理常式負責管理傳輸層次的詳細資料，在呼叫 Web 服務及 (選擇性地) 保護回應時，將會需要這些資料，以執行下列三項主要作業：將協同作業「要求」商業物件轉換成 SOAP 要求訊息，以該要求訊息來呼叫端點 Web 服務，若為「要求/回應 (同步)」模式，則會將 SOAP 回應訊息轉換商業物件，並將該物件傳回協同作業。連接器會使用 SOAP/HTTP-HTTPS 通訊協定處理常式來呼叫 SOAP/HTTP(S) Web 服務，並使用 SOAP/JMS 通訊協定處理常式來呼叫 SOAP/JMS Web 服務。

Web 服務連接器一律使用 TLO 從協同作業來呼叫的。連接器會從 TLO 來判斷「SOAP 要求」商業物件，並以此商業物件來呼叫 SOAP 資料處理常式。資料處理常式會將連接器所傳送的要求訊息傳回給 Web 服務。

若是執行同步 Web 服務，連接器會利用 SOAP 資料處理常式來將 SOAP 回應及錯誤訊息轉換為「SOAP 回應」及「錯誤」商業物件。為了輔助資料處理常式來判斷要為這些 SOAP 回應/錯誤解析哪個商業物件，以進行商業物件轉換，連接器會提供特定的 meta 資訊給資料處理常式。尤其是，連接器會建立一個清單，其中包含所有作為呼叫 TLO 之子項的「回應」及「錯誤」商業物件。其中應僅有一個回應商業物件，但可能會有許多「錯誤」商業物件。此外，可能會有一個且僅有一個 defaultfault 商業物件。連接器會嘗試比對，然後將 SOAP BodyName 和 BodyNamespace 對映至出現於所有「回應」商業物件之清單中的商業物件名稱。若為「SOAP 回應」商業物件，則會使用「SOAP 回應」商業物件的 SOAP Config MO 內容來尋找此配對。若為「SOAP 錯誤」商業物件，則會使用詳細元素第一個子項的 elem\_name 和 elem\_ns 屬性層次 ASI 內容來尋找此配對。若為 defaultfault 商業物件，連接器只會將 defaultfault 商業物件的名稱通知給資料處理常式。若沒有為此轉換作業解析其他的錯誤商業物件，則資料處理常式應將 defaultfault 商業物件作為最後的手段。

---

## SOAP/HTTP(S) Web 服務

Web 服務支援 HTTP 傳輸通訊協定。HTTP 內含主從架構模型，HTTP 用戶端以此模型來開啓連線，並將要求訊息傳送至 HTTP 伺服器。用戶端要求訊息的目的是要呼叫 Web 服務。HTTP 伺服器會分派含有呼叫的訊息，並關閉連線。

當協同作業是以 Web 服務來顯現，而要處理對其提出的用戶端要求時，連接器的 SOAP/HTTP 及 SOAP/HTTPS 通訊協定接聽器會使用 HTTP 主從架構及「要求/回應」模型。但是 SOAP/HTTP 接聽器的主要功能並非作為 HTTP 伺服器 -- proxy、媒介或其他。SOAP/HTTP 接聽器的主要功能，是使用於企業中作為端點，並置於防火牆之後。因此，必須在防火牆中部署個別的 Web 伺服器或閘道，以將用戶端要求遞送至接聽器。如需進一步資訊，請參閱第 1 頁的第 1 章，『連接器概觀』。

SOAP/HTTP 及 SOAP/HTTPS 通訊協定接聽器將協同作業作為 SOAP/HTTP(S) Web 服務來顯現。連接器會使用 SOAP/HTTP-HTTPS 通訊協定處理常式來呼叫 SOAP/HTTP(S) Web 服務。

## 同步的 SOAP/HTTP(S) Web 服務

從連接器處理程序的角度來看，同步的 HTTP Web 服務是依「要求/回應」路徑來執行的程序。若 SOAP/HTTP 或 SOAP/HTTPS 通訊協定接聽器順利處理了 HTTP 要求訊息，其主體將會包含 Web 服務回應及 200 OK 的 HTTP 狀態碼。若傳回錯誤，主體會包含錯誤訊息及 500 的狀態碼。

## 非同步的 SOAP/HTTP(S) Web 服務

從連接器處理程序的角度來看，非同步的 HTTP Web 服務是依「僅限要求」路徑來執行的程序。若 SOAP/HTTP 或 SOAP/HTTPS 通訊協定接聽器順利接收並處理了「僅限要求」的 Web 服務作業，就會產生 202 Accepted 的 HTTP 狀態碼。您亦可配置連接器來產生 200 OK 的 HTTP 狀態碼 -- 相關資訊，請參閱表 40 中的 HTTPAsyncResponseCode 內容。若發生錯誤，就會產生 500 的 HTTP 狀態碼。雖可能會傳回錯誤主體，但沒有回應。

---

## SOAP/JMS Web 服務

JMS 為傳輸層次的 API，企業可將其與 Web 服務解決方案合併，以進行傳訊、維護資料持續性，及存取 Java 型的應用程式。SOAP/JMS Web 服務是實行 JMS 佇列型傳輸的一種 Web 服務。

Web 服務解決方案可針對佇列或主題來實作 JMS 目標。連接器的 SOAP/JMS 通訊協定接聽器僅支援佇列目的地；不支援主題。僅支援 JMS 文字訊息。

在事件處理程序期間，SOAP/JMS Web 服務用戶端會將要求訊息與 JMS 訊息包裝在一起，並將其發佈至 JMS 目的地為連接器的佇列。JMS 目的地會擷取包含 Web 服務要求的 JMS 訊息，並從 JMS 訊息擷取 SOAP 要求訊息。然後它會處理 SOAP 要求訊息。

## 同步的 SOAP/JMS Web 服務

若為同步的連接器處理程序 (要求/回應)，回應訊息會和 JMS 訊息包裝在一起 (就像要求訊息一樣)。然後，會將包含 Web 服務回應的 JMS 訊息從送入的要求傳送到 JMSReplyTo 佇列。回應訊息中的 JMS 標頭設定為 JMS 要求訊息中的標頭值，如下所示：

- 回應訊息的 JMSCorrelationID 必須設定為 JMS 要求訊息的 JMSMessageID 值。
- 回應訊息的 JMS DeliveryMode 設定為要求的 JMSDeliveryMode。
- 回應訊息的 JMSPriority 設定為要求的 JMSPriority。
- 要求訊息的 JMSExpiration 設定為要求的 JMSExpiration。

第 59 頁的『SOAP/JMS 通訊協定接聽器處理程序』中將詳細討論此處理程序。

## 非同步的 SOAP/JMS Web 服務

從連接器處理程序的角度來看，非同步的 SOAP/JMS Web 服務是依「僅限要求」路徑來執行的程序。若 SOAP/JMS 通訊協定接聽器順利接收並處理了「僅限要求」的 Web 服務訊息，則不會將任何包含回應的 JMS 訊息傳回用戶端。若已配置 ReplyToQueue，並於接收 JMS 訊息時，發生錯誤，則會將錯誤訊息傳回 Web 服務用戶端。此外，若 SOAP/JMS 接聽器中指定了 ErrorQueue，則會將錯誤訊息保存於該處。

---

## 事件處理程序

實作事件處理程序功能的首要步驟，就是將商業程序 -- 協同作業 -- 顯現為 Web 服務。然後您要將此 Web 服務發佈於 UDDI 登錄中 (舉例說明)，並配置連接器來回應呼叫協同作業的 Web 服務用戶端。

於事件處理程序期間，連接器會使用通訊協定接聽器及 SOAP 資料處理常式，將 Web 服務用戶端的 SOAP 要求訊息轉換成商業物件，以供顯現為 Web 服務的協同作業操作。於事件處理程序中，通訊協定接聽器扮演著很重要的角色。

## 通訊協定接聽器

Web 服務要求可能來自各種傳輸埠，包括 HTTP、HTTPS 及 JMS。Web 服務通訊協定接聽器會監視這些要求到達其傳輸通道的狀況。通訊協定接聽器及對應的通道有三種：

- SOAP/HTTP 通訊協定接聽器
- SOAP/HTTPS 通訊協定接聽器
- SOAP/JMS 通訊協定接聽器

這些中每一種都由其傳輸通道上接聽的執行緒所構成。當它接收來自用戶端的 SOAP 要求訊息時，接聽器會向通訊協定接聽器組織架構登錄該事件。

通訊協定接聽器組織架構負責管理通訊協定接聽器，在有資源可用時，將要求排程。當您設定連接器特有的內容值時，您要配置接聽器及通訊協定接聽器組織架構的各個層面。於通訊協定接聽器組織架構內容中，您可以配置的項目如下：

- **WorkerThreadCount** 可用於通訊協定接聽器組織架構的執行緒總數，亦即其可並行處理的要求數。
- **RequestPoolSize** 可登錄於通訊協定接聽器組織架構的要求數上限。若它收到的要求超過此限，就不會再登錄新的要求。

這二個連接器特有的內容，可控制記憶體配置，防止通訊協定接聽器以無止盡的 Web 服務事件來癱瘓連接器。配置演算法如下：在任何時候，連接器可接收的事件總數等於  $WorkerThreadCount + RequestPoolSize$ 。它可以並行處理  $WorkerThreadCount$  的要求數。

您可於通訊協定接聽器組織架構中插入額外的通訊協定接聽器。如需進一步資訊，請參閱第 93 頁的『建立多重通訊協定接聽器』及第 76 頁的『連接器特有的配置內容』。

## SOAP/HTTP 及 SOAP/HTTPS 通訊協定接聽器處理程序

SOAP/HTTP(S) 通訊協定接聽器包含一個執行緒，可持續接聽來自 Web 服務用戶端的 HTTP(S) 要求。接聽器執行緒連結了指定於「主機與埠」連接器特有配置 (接聽器) 內容中的主機和埠。另一個配置內容 -- RequestWaitTimeout-- 則定義了於檢查連接器是否關閉之前，接聽器要等待要求的時間間隔。

圖 24 說明同步作業的 SOAP/HTTP 通訊協定接聽器處理程序。

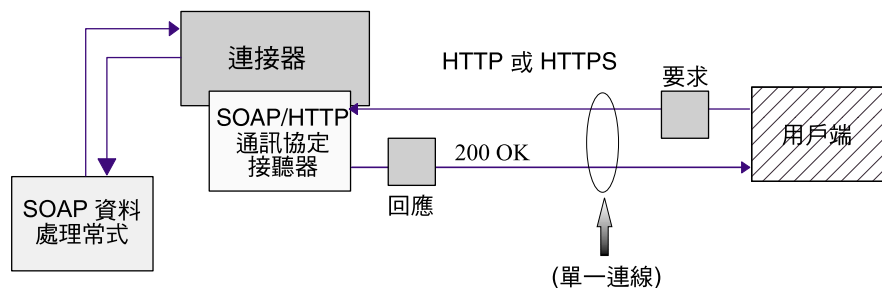


圖 24. SOAP/HTTP 通訊協定接聽器：同步的事件處理程序

圖 25 顯示非同步作業的 SOAP/HTTP 通訊協定接聽器處理程序。

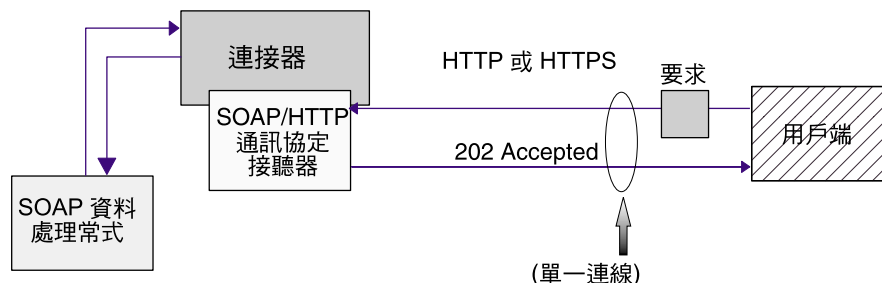


圖 25. SOAP/HTTP 通訊協定接聽器：非同步的事件處理程序

當 Web 服務用戶端起始 SOAP/HTTP 或 SOAP/HTTPS 要求時，它會將 SOAP 要求訊息登記在 SOAP/HTTP 或 SOAP/HTTPS 接聽器的 URL。用戶端應使用 HTTP POST 方法來呼叫通訊協定接聽器 URL。

當 HTTP(S) 要求到達時，接聽器會向通訊協定接聽器組織架構登錄該要求，排定該事件於有資源可用時，加以處理。然後接聽器會從要求擷取通訊協定標頭及內容。

表 27 彙總接聽器所使用的規則優先順序，以決定入埠訊息的 Charset、MimeType、ContentType 及 Content-Type 標頭。

表 27. 入埠訊息的 SOAP/HTTP(s) 通訊協定接聽器處理程序規則

優先順序	Charset	MimeType	ContentType	Content-Type 標頭
------	---------	----------	-------------	-----------------

表 27. 入埠訊息的 SOAP/HTTP(s) 通訊協定接聽器處理程序規則 (繼續)

1	來自送入的 HTTP 訊息 Content-Type 標頭值的 Charset 參數值	此接聽器的 URLsConfiguration 連接器內容值	來自 Content-Type 標頭值的送入的 HTTP 訊息類型/子類型值	送入的 HTTP 訊息 Content-Type 標頭
2	此接聽器的 URLsConfiguration 內容值	SOAPDHMimeType 連接器內容值		
3	如果要求訊息 ContentType 的類型為 text 且具有任何子類型 (例如 text/xml、text/plain 等), 則預設值為 ISO-8859-1。否則將不會使用 charset。	預設為 ContentType		

如第 56 頁的表 27 所示：

- 通訊協定接聽器會根據下列規則來決定入埠訊息的 Charset：
  1. 接聽器試圖擷取 HTTP 訊息 Content-Type 標頭值的 Charset 參數中的 Charset。
  2. 如果未從 Content-Type 標頭取得 Charset 值，那麼通訊協定接聽器就會試圖為這個接聽器讀取 URLsConfiguration 內容值。
  3. 如果無法用前幾個步驟中所說明的方法來取得 Charset 值，且訊息 ContentType 的類型為 text 且具有任何子類型 (例如 text/xml、text/plain 等)，則接聽器會使用預設 Charset 值 ISO-8859-1。否則不會使用 Charset 值。
- 接聽器會根據下列規則來決定回應訊息的 MimeType：
  1. 如果您已針對送入的要求訊息所使用的 URL 配置 TransformationRules，且要求 ContentType 符合 TransformationRule 的 ContentType，那麼接聽器就會使用 TransformationRule 來擷取 MimeType，以將要求訊息轉換為「SOAP 要求」商業物件。接聽器會在所要求之 URL 的 URLsConfiguration 內容中，根據 ContentType 值 (例如 text/xml)，試圖尋找正確的 TransformationRule 相符項目。
  2. 若失敗了，接聽器便會嘗試在所要求的 URL 下，尋找適用於多個 ContentType 的 TransformationRule (例如 \*/\*)。
  3. 如果 MimeType 沒有 TransformationRule 相符項目，那麼接聽器就會使用 SOAPDHMimeType 連接器配置內容作為 MimeType 值。
  4. 如果前幾個步驟都無法判斷 MimeType，將會使用 ContentType 的值作為 MimeType 來呼叫 SOAP 資料處理常式，並將要求訊息轉換為「SOAP 要求」商業物件。
- 接聽器可擷取送入的 HTTP 訊息 Content-Type 標頭的類型/子類型，以判斷 ContentType。
- 接聽器可從送入的 HTTP 訊息 Content-Type 標頭的 Content-Type 標頭，來判斷 Content-Type 標頭。

若以非同步的方式來呼叫協同作業，接聽器會傳遞要求商業物件至整合分配管理系統，並以 HTTP 狀態碼 202 Accepted 來回應 Web 服務用戶端。接聽器處理程序到此結束。

若為同步的呼叫，接聽器就會同步呼叫協同作業。而協同作業則會以「SOAP 回應」商業物件來回應。

在決定回應訊息的 Charset、MimeType、ContentType 及 Content-Type 標頭時，表 28 會彙總接聽器所使用的規則優先順序。

表 28. 離埠同步回應訊息的 SOAP/HTTP(s) 通訊協定接聽器處理規則

優先順序	Charset	MimeType	ContentType	Content-Type 標頭
1	Protocol Config MO Content-Type 標頭	TLO 中的 MimeType 內容	Protocol Config MO Content-Type 標頭	Protocol Config MO Content-Type 標頭
2	TLO 中的 Charset 內容值	要求訊息 MimeType，但僅限要求與回應 ContentType 相符時。	要求訊息 ContentType	使用 ContentType 及 Charset 來建構 Content-Type 標頭
3	要求訊息 Charset，但僅限要求與回應 ContentType 相符時。	SOAPDHMimeType 連接器內容值		
4	如果 ContentType 為 text/*，則預設值為 ISO-8859-1。否則將不會使用 charset。	使用 ContentType 值作為 MimeType		

如表 28 所示：

- 接聽器會根據下列規則來決定回應訊息的 Charset：
  1. 若已於「回應」商業物件 Protocol Config MO 中指定 Charset，就會使用該值。
  2. 若未於「回應」商業物件 Protocol Config MO 標頭中指定 Charset 值，且若「要求」與「回應」商業物件為 TLO 的子項，接聽器就會檢查 TLO 中是否指定了 Charset。
  3. 若未在 TLO 中指定 Charset，或者「回應」商業物件不是 TLO，在此情形下，若回應的 ContentType 與要求相同，就會以要求的 Charset 來用於回應。
  4. 如果前幾個步驟都無法判斷回應 Charset 值，且訊息 ContentType 的類型部份為 text 且子類型為任何項目 (例如 text/xml、text/plain 等)，則接聽器會使用預設 Charset 值 ISO-8859-1。否則不會使用 Charset 值。
- 接聽器會根據下列規則來決定回應訊息的 MimeType：
  1. TLO 的 MimeType 屬性
  2. 若遺漏了 TLO MimeType 屬性，且要求與回應 ContentType 相符，接聽器就會使用回應訊息的要求 MimeType。
  3. 如果前幾個步驟都失敗，那麼接聽器會使用 SOAPDHMimeType 連接器內容的值。
  4. 否則接聽器會使用 ContentType 值作為 MimeType。
- 接聽器會根據下列規則來決定回應訊息的 ContentType：
  1. 若在「回應」商業物件 Protocol Config MO 中指定了 Content-Type 標頭，則會使用 Content-Type 標頭的類型/子類型部份作為 ContentType。
  2. 若未在「回應」商業物件 Protocol Config MO 中指定 Content-Type 標頭，接聽器便會使用已判斷的 ContentType 及 Charset 來建構 Content-Type 標頭 (若已決定將該 Charset 用於回應訊息)。

接聽器會處理 HTTP Protocol Config MO。協同作業要負責確定，在「要求-回應」事件的環境定義中，傳遞至 HTTP Protocol Config MO 中的標頭值是正確的。接聽器會根據下列規則，載入標準標頭和自訂內容：

1. 接聽器會調查 HTTP Protocol Config MO 的每個項目，以忽略特殊屬性 (例如 ObjectEventId)。

2. 每個非空值的標頭都會放在外送訊息，且有可能發生其他處理程序（例如 Content-Type 標頭）。
3. 請記得，利用上述方法時，接聽器可能會在訊息中設定非標準的標頭，但不會檢查該訊息的邏輯及語法是否正確。
4. 若 HTTP Protocol Config MO UserDefinedProperties 中有一或數項自訂內容，接聽器便會在「實體標頭區段」(最後一個標頭區段) 中新增它們。如需自訂內容的相關資訊，請參閱第 29 頁的『事件處理程序的使用者定義內容』。

**註：**在 HTTP Protocol Config MO 中指定下列任一標頭，很有可能會導致不正確的 HTTP 訊息：Connection、Trailer、Transfer-Encoding、Content-Encoding、Content-Length、Content-MD5、Content-Range。

然後接聽器會呼叫 SOAP 資料處理常式，將「協同作業」所傳回的「回應」商業物件轉換為 SOAP 回應訊息。

接聽器將回應訊息傳遞至 Web 服務用戶端，並包含 200 OK HTTP 狀態碼。若協同作業傳回「SOAP 錯誤」商業物件，就會轉換成「錯誤」訊息。此錯誤訊息會與 500 Internal Server Error HTTP 程式碼一起傳遞至 Web 服務用戶端。

然後接聽器會關閉連線，而處理事件的執行緒會變成可用的狀態。

## 不受支援的 SOAP/HTTP 通訊協定接聽器處理程序功能

SOAP/HTTP 通訊協定接聽器不支援下列各項：

- 快取：通訊協定接聽器不執行定義於 HTTP 指定作業 (RFC2616) 中的任何快取功能。
- Proxy：通訊協定接聽器不執行定義於 HTTP 指定作業 (RFC2616) 中的任何 Proxy 功能。
- 持續性連線：通訊協定接聽器不支援定義於 HTTP 指定作業 (RFC2616) 中的持續性連線。反之，通訊協定接聽器會假定每個 HTTP 連線的範圍都是單一用戶端要求，且於服務要求完成時，關閉連線。通訊協定接聽器不會嘗試透過服務呼叫來重複使用連線。
- 重新導向：通訊協定接聽器不支援重新導向。
- 大型檔案轉送：通訊協定接聽器不可用於大型檔案轉送。您可考慮以參照的方式來傳遞大型檔案。
- 狀態管理：通訊協定接聽器不支援 RFC2965 所描述的 HTTP 狀態管理機制。
- Cookie：通訊協定接聽器不支援 Cookie。

## 使用安全 Socket 的 SOAP/HTTPS 接聽器處理程序

SOAP/HTTPS 通訊協定接聽器處理程序與 SOAP/HTTP 通訊協定接聽器處理程序區段中所描述的相同，但 HTTPS 使用安全的 Socket。如需進一步資訊，請參閱第 74 頁的『SSL』。

## SOAP/JMS 通訊協定接聽器處理程序

SOAP/JMS 通訊協定接聽器包含一個執行緒，可持續接聽 InputQueue (來自 Web 服務用戶端之要求的 JMS 目的地)。RequestWaitTimeout 連接器配置內容可定義，在檢查連接器是否關閉之前，接聽器等待要求的時間有多長。

圖 26 顯示同步作業的 SOAP/JMS 通訊協定接聽器處理程序。該圖例並未顯示 JMS 提供者資訊。

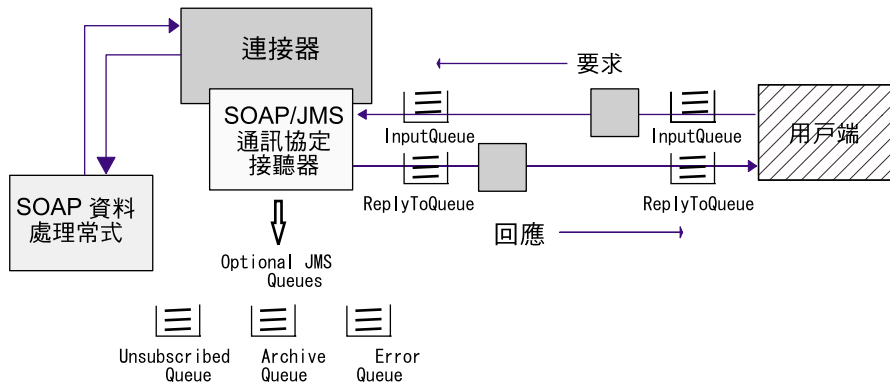


圖 26. SOAP/JMS 通訊協定接聽器：同步的事件處理程序

圖 27 顯示非同步作業的 SOAP/JMS 通訊協定接聽器處理程序。

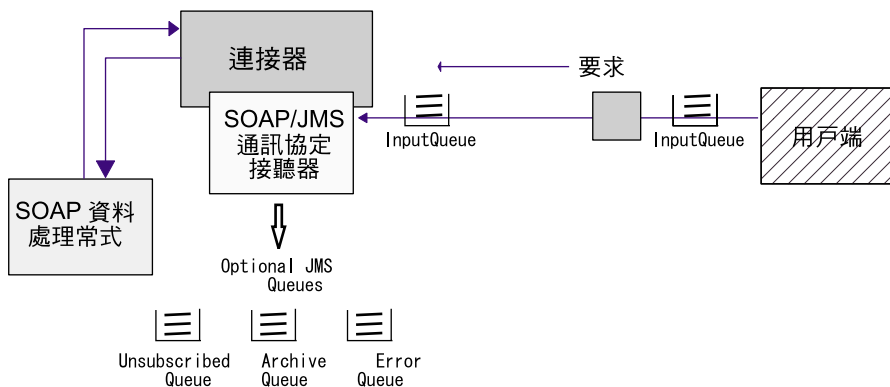


圖 27. SOAP/JMS 通訊協定接聽器：非同步的事件處理程序

**註：**若 `LookupQueueUsingJNDI` 配置內容設定為 `true`，則 SOAP/JMS 通訊協定接聽器使用 JNDI 來查閱佇列。JNDI 內容指定於連接器內容中。如需進一步資訊，請參閱第 72 頁的『連接器及 JMS』，以及第 76 頁的『連接器特有的配置內容』中的 JNDI 相關內容。

當 Web 服務用戶端起始 SOAP/JMS 要求時，其會傳送 SOAP 要求訊息至 SOAP/JMS 接聽器所接聽的 `InputQueue`。當它接收來自 `InputQueue` 的 SOAP 要求訊息時，SOAP/JMS 通訊協定接聽器會向通訊協定接聽器組織架構登錄該要求。通訊協定接聽器組織架構會隨著資源的可用狀態要求排程。

**註：**若連接器配置內容 `InDoubtEvents` 的設定為 `Reprocess`，則通訊協定接聽器組織架構會先排定來自 `InProgressQueue` 的 JMS 訊息，再排定來自 `InputQueue` 的訊息。

然後接聽器會將此訊息 --主體及必要的 JMS 標頭 (`JMSCorrelationID`、`JMSMessageID`、`JMSPriority`、`JMSExpiration`、`JMSDeliveryMode`、`JMSReplyTo`、`JMSTimeStamp`、`JMSType`)-- 分派至 `InProgressQueue`。接著，通訊協定接聽器組織架構會登錄事件。



接聽器讀取來自 InProgressQueue 的 JMS 訊息，擷取訊息的主體及下列標頭：

- JMSTDestination
- JMSRedelivered
- JMSCorrelationID
- JMSMessageID
- JMSPriority
- JMSExpiration
- JMSDeliveryMode
- JMSReplyTo
- JMSTimeStamp
- JMSType
- JMS 訊息內容類型 (不是標頭，而是來自訊息的資訊)

**JMS 訊息內容類型** 接聽器會決定送入的訊息的內容類型，並將資訊儲存在 JMS Protocol Config MO 的 MessageType 屬性中。內容可為 TextMessage 或 BytesMessage。若為 TextMessage 格式，則接聽器會透過「字串 API」，以擷取為「字串」的 Web 服務要求訊息來呼叫資料處理常式。若為 BytesMessage，則接聽器會透過「位元組資料處理常式 API」，以擷取為位元組陣列的 Web 服務要求訊息來呼叫資料處理常式。

接聽器使用 SOAPDHMimeType 連接器配置內容來呼叫 SOAP 資料處理常式，將要求訊息轉換成「SOAP 要求」商業物件。若於轉換期間發生錯誤，且已指定 JMSReplyTo JMS 標頭，接聽器便會以 SOAP 錯誤訊息來回應，並將 faultcode 設定為 Client，將 faultstring 設定為 Cannot Parse。錯誤訊息不提供其他細節。

接聽器使用資料處理常式所傳回之「SOAP 要求」商業物件的物件層次 cw\_mo\_jms ASI，來決定 Protocol Config MO。請注意，就事件處理程序而言，ASI 和 Protocol Config MO 都是選用性的。若它找到 Protocol Config MO，接聽器就會用之前所擷取的 JMS 訊息標頭來載入資料。表 29 顯示 Protocol Config MO 與 JMS 訊息標頭之間屬性的對映關係。

表 29. JMS 標頭與 Protocol Config MO 屬性的對映表

Protocol Config MO 屬性	JMS 標頭名稱	說明
CorrelationID	JMSCorrelationID	來自要求訊息的 JMSCorrelationID 標頭
MessageId	JMSMessageId	來自要求訊息的 JMSMessageID 標頭
優先順序	JMSPriority	來自要求訊息的 JMSPriority 標頭
Expiration	JMSExpiration	來自要求訊息的 JMSExpiration 標頭
DeliveryMode	JMSDeliveryMode	來自要求訊息的 JMSDeliveryMode 標頭
ReplyTo	JMSReplyTo	來自要求訊息的 JMSReplyTo 標頭。JMS API 以 JMSTDestination 的形式傳回此標頭，但 SOAP/JMS 通訊協定接聽器則是傳回佇列名稱。
Timestamp	JMSTimestamp	來自要求訊息的 JMSTimestamp 標頭
Redelivered	JMSRedelivered	來自要求訊息的 JMSRedelivered 標頭
類型	JMSType	來自回應訊息的 JMSType 標頭

表 29. JMS 標頭與 Protocol Config MO 屬性的對映表 (繼續)

Destination	JMSDestination	來自要求訊息的 JMSDestination 標頭
MessageType	不適用	要求訊息內容的類型。這個屬性的值對於 TextMessage 內容來說是「文字」，對於 BytesMessage 內容來說是「位元組」。

如果 SOAP/JMS Protocol Config MO 的 UserDefinedProperties 屬性中有一或多個自訂內容，則接聽器就會嘗試從訊息中擷取其值，並移入 UserDefinedProperties 商業物件。如需自訂內容的相關資訊，請參閱第 29 頁的『事件處理程序的使用者定義內容』。

若整合分配管理系統沒有訂閱 TLO (若為「非 TLO SOAP 要求」商業物件)，接聽器就會記載錯誤。若要求訊息中指定了 JMSReplyTo 標頭，接聽器就會建立 SOAP 錯誤訊息，並將其置於 JMSReplyTo 佇列中。faultcode 的設定為 Client，而 faultString 的設定為 Not subscribed to this message。錯誤訊息中沒有提供其他細節。若配置指定這麼做，接聽器亦會將原始的 JMS 要求訊息 (包括其 JMS 標頭) 保存於 UnsubscribedQueue 中。

若以非同步的方式呼叫協同作業，接聽器會傳遞「要求」商業物件至整合分配管理系統。然後接聽器會將訊息從 InProgressQueue 移除。若配置指定這麼做，接聽器亦會將原始的 JMS 要求訊息 (包括其 JMS 標頭) 保存於 ArchiveQueue 中。

若於非同步處理程序期間發生錯誤，且已指定 JMSReplyTo，接聽器就會以錯誤訊息來回應。其 faultcode 的設定為 Server，且其 faultstring 的設定為 Internal Error。若配置指定這麼做，接聽器亦會將原始的 JMS 要求訊息 (包括其 JMS 標頭) 保存於 ErrorQueue 中。

若為同步的呼叫，接聽器就會同步呼叫協同作業。而協同作業則會以「SOAP 回應」商業物件來回應。接聽器會呼叫 SOAP 資料處理常式，將「協同作業」所傳回的「回應」商業物件轉換為 SOAP/JMS 回應訊息。回應內容的類型是根據「SOAP 回應」商業物件的 JMS Protocol Config MO 中的 MessageType 屬性值而定。如果 MessageType 是「文字」，接聽器就會透過「字串」資料處理常式 API，將「SOAP 回應」商業物件轉換成「字串」。如果 MessageType 是「位元組」，接聽器就會透過「位元組」資料處理常式 API，將「SOAP 回應」商業物件轉換成位元組陣列。(預設訊息內容類型是相對應的同步要求的訊息內容類型)。接聽器會將回應訊息傳遞至 ReplyTo 佇列 (此為原始要求訊息上的 JMSReplyTo 標頭所提供)。接聽器然後將資料處理常式所傳回的回應訊息設定為 TextMessage 或 BytesMessage (視之前決定的 MessageType 而定)，並設定表 30 中所顯示的標頭。

表 30. SOAP/JMS 通訊協定接聽器於回應訊息中所設定的標頭值

JMS 標頭名稱	值
JMSCorrelationId	要求訊息的 JMSMessageId
JMSDeliveryMode	要求訊息的 JMSDeliveryMode
JMSPriority	要求訊息的 JMSPriority
JMSExpiration	要求訊息的 JMSExpiration
JMSRedelivered	要求訊息的 JMSRedelivered
JMSReplyTo	要求訊息的 JMSReplyTo
JMSTimestamp	要求訊息的 JMSTimestamp

表 30. SOAP/JMS 通訊協定接聽器於回應訊息中所設定的標頭值 (繼續)

JMSType	要求訊息的 JMSType
---------	---------------

如果在「回應」或「錯誤」商業物件的 JMS Protocol Config MO UserDefinedProperties 屬性中出現「JMS 自訂內容」，接聽器將會在回應訊中設定它們。

若配置指定這麼做，接聽器便會將原始的 JMS 訊息 (來自 Web 服務用戶端的要求) 及其標頭從 InProgressQueue 移至 ArchiveQueue。

若發生錯誤，且已指定 JMSReplyTo，接聽器便會以錯誤訊息來回應，且若配置指定這麼做，接聽器亦會將原始的 JMS 要求訊息保存於 ErrorQueue。

## 事件持續性及遞送

事件持續性取決於通訊協定：

- **SOAP/HTTP 通訊協定接聽器** 沒有持續性，因此也沒有保證的遞送
- **SOAP/HTTPS 通訊協定接聽器** 沒有持續性，因此也沒有保證的遞送
- **SOAP/JMS 通訊協定接聽器** JMS 佇列事件持續性，至少有一次保證遞送。如需 JMS 佇列的相關資訊，請參閱第 76 頁的『連接器特有的配置內容』。

## 事件排序

連接器可使用任何順序來傳遞事件。

## 事件觸發

事件觸發機制取決於通訊協定接聽器的配置內容。

- **SOAP/HTTP 通訊協定接聽器** 當有 HTTP 連線要求時，會透過 ServerSocket 來執行接聽作業
- **SOAP/HTTPS 通訊協定接聽器** 當有 HTTPS 連線要求時，會透過安全的 ServerSocket 層來執行接聽作業
- **SOAP/JMS 通訊協定接聽器** 當有送入的 JMS 訊息附帶 Web 服務要求時，會透過輸入佇列來執行接聽作業。如需 JMS 佇列的相關資訊，請參閱第 76 頁的『連接器特有的配置內容』。

**註：** 連接器不會分辨「建立」或「更新」或「擷取」或「刪除」。這些事件皆遵循相同的做法。

## 事件偵測

事件偵測由各個通訊協定接聽器所執行。事件偵測機制完全取決於傳輸，及您為各個接聽器配置連接器特有內容的方式。如需這些內容的相關資訊，請參閱第 76 頁的『連接器特有的配置內容』。

## 事件狀態

事件狀態是由通訊協定接聽器所管理，並取決於傳輸埠，及您配置接聽器的方式。

- **SOAP/HTTP 通訊協定接聽器** HTTP 承襲其本質非持續及同步化的特性。因此，不維持事件狀態。

- **SOAP/HTTPS 通訊協定接聽器** HTTP 承襲其本質非持續及同步化的特性。因此，不維持事件狀態。
- **SOAP/JMS 通訊協定接聽器** JMS 為持續性的傳輸埠。以佇列來維護事件狀態。如需 JMS 佇列的相關資訊，請參閱第 76 頁的『連接器特有的配置內容』。

## 事件擷取

事件擷取是由通訊協定接聽器所管理，並取決於傳輸埠，及您配置接聽器的方式。

- **SOAP/HTTP 通訊協定接聽器**藉由從 Socket 擷取 HTTP 要求來擷取事件。
- **SOAP/HTTPS 通訊協定接聽器**藉由從 Socket 擷取 HTTP 要求來擷取事件。
- **SOAP/JMS 通訊協定接聽器**使用 JMS API 來擷取事件。JMS 通訊協定接聽器從 JMS 輸入佇列擷取事件，並將其移至進行中的佇列。如需 JMS 佇列的相關資訊，請參閱第 76 頁的『連接器特有的配置內容』。

## 事件保存

事件保存是由通訊協定接聽器所管理，並取決於傳輸埠，及您配置接聽器的方式。

- **SOAP/HTTP 通訊協定接聽器**因為傳輸之非持續性及同步化本質的關係，不執行保存作業。
- **SOAP/HTTPS 通訊協定接聽器**因為傳輸之非持續性及同步化本質的關係，不執行保存作業。
- **SOAP/JMS 通訊協定接聽器**您可配置連接器將事件保存於 JMS 佇列中，包括那些未訂閱的事件、成功的事件，以及失敗的事件。如需 JMS 佇列的相關資訊，請參閱第 76 頁的『連接器特有的配置內容』。

## 事件回復

事件回復是由通訊協定接聽器所管理，並取決於傳輸埠，及您配置接聽器的方式。

- **SOAP/HTTP 通訊協定接聽器**因為傳輸之非持續性本質的關係，不執行事件回復作業。
- **SOAP/HTTPS 通訊協定接聽器**因為傳輸之非持續性本質的關係，不執行事件回復作業。
- **SOAP/JMS 通訊協定接聽器** JMS 為持續性的傳輸埠。若於處理事件時，關閉了連接器，事件仍會留在 InProgressQueue 中。您可配置連接器，於啟動時處理這些事件，進而啟用事件回復功能。InDoubtEvents 連接器配置內容可決定事件回復機制。

**註:** SOAP/JMS 接聽器保證至少會遞送一次至整合分配管理系統。接聽器無法保證會有一次且僅有一次的遞送。且接聽器所接收的事件可使用任何順序來遞送至到整合分配管理系統。

啟動時，JMS 通訊協定接聽器會先嘗試從 InProgressQueue 擷取事件。接下來會做什麼，則要看指定給 InDoubtEvents 配置內容的值而定。表格中舉例說明了回復範例。如需 JMS 佇列的相關資訊，請參閱第 76 頁的『連接器特有的配置內容』。

表 31. SOAP/JMS 通訊協定接聽器於回應訊息中所設定的標頭值

InDoubtEvents 值	事件回復處理程序
FailOnStartup	若它在 InProgressQueue 中發現事件，接聽器會記載嚴重錯誤，並立即關閉。

表 31. SOAP/JMS 通訊協定接聽器於回應訊息中所設定的標頭值 (繼續)

Reprocess	若它在 InProgressQueue 中發現事件，接聽器會先處理那些事件。接聽器可追蹤於 InProgressQueue 中所發現的訊息個數。
Ignore	會忽略 InProgressQueue 中的事件。接聽器可追蹤於 InProgressQueue 中所發現的事件，及接聽器忽略那些事件的狀況。
LogError	若它在 InProgressQueue 中發現事件，接聽器會記載錯誤，並繼續處理。

## 要求處理程序

您可使用連接器的要求處理功能，讓協同作業呼叫 Web 服務。開發工作包括使用 WSDL ODA 來產生 Web 服務最上層物件 (TLO)，並配置協同作業來部署該物件。如需進一步資訊，請參閱第 129 頁的第 6 章，『啓用要求處理程序的協同作業』。您亦必須配置連接器及其要求處理程序元件：通訊協定處理常式組織架構及通訊協定處理常式。

在執行時期，連接器會接收來自協同作業的商業物件格式要求。商業物件--「SOAP 要求」、選擇性的「SOAP 回應」及「SOAP 錯誤」商業物件-- 內含於 WSDL ODA 所產生的 TLO 中，並由已配置來使用 Web 服務的協同作業發出。TLO 及其子項商業物件所包含的屬性和 ASI 可指定處理程序模式 (同步或非同步)、通訊協定處理常式使用的資料處理常式 MIME 類型，以及目標 Web 服務的位址。通訊協定處理常式會使用此資訊來呼叫 SOAP 資料處理常式的實例、將「要求」商業物件轉換成 SOAP 要求訊息，並呼叫目標 Web 服務。若為同步模式，通訊協定處理常式會再次呼叫資料處理常式，將回應訊息轉換成「SOAP 回應」商業物件，並將其傳回協同作業。

在回應 SOAP 要求訊息時，連接器可從遠端交易友機接收下列各項：

- 包含資料的 SOAP 回應訊息
- 包含錯誤資訊的 SOAP 回應訊息

通訊協定處理常式於要求處理程序中，扮演著非常重要的角色。

### 通訊協定處理常式

協同作業可透過 HTTP、HTTPS 或 JMS 傳輸來呼叫 Web 服務。連接器有二個通訊協定處理常式，及對應的通道：

- SOAP/HTTP-HTTPS 通訊協定處理常式，用以呼叫 SOAP/HTTP 及 SOAP/HTTPS Web 服務
- SOAP/JMS 通訊協定處理常式，用以呼叫 SOAP/JMS Web 服務

通訊協定處理常式組織架構負責管理通訊協定處理常式，在啓動時將其載入。當連接器接收「要求」商業物件時，要求執行緒 (請注意，每個協同作業要求都是隨著自己的執行緒傳進來的) 會呼叫通訊協定處理常式組織架構來處理要求。

通訊協定處理常式組織架構會讀取 TLO 處理常式屬性 ASI，以決定要使用哪一個通訊協定處理常式。藉由套用一系列的規則 (請參閱第 66 頁的『SOAP/HTTP-HTTPS 通訊協定處理常式處理程序』及第 69 頁的『SOAP/JMS 通訊協定處理常式處理程序』)，通訊協定處理常式會呼叫資料處理常式，將「要求」商業物件轉換成 SOAP 要求訊息。通訊協定處理常式會將要求訊息包裝於傳輸埠--HTTP(S) 或 JMS-- 訊息中。若其於「要求」商業物件中發現 SOAPAction ASI，通訊協定處理常式便會將其新增至要求訊息標頭中。

通訊協定處理常式接著會讀取「要求」商業物件 Protocol Config MO 的 Destination 屬性，以決定目標位址。然後通訊協定處理常式會以要求訊息來呼叫目標 Web 服務。

通訊協定處理常式會讀取 ws\_mode TLO ASI，判斷處理程序模式是同步還是非同步的。若此 ASI 的設定為 asynch，則通訊協定處理常式處理程序已完成。否則通訊協定處理常式會等待回應訊息。若回應訊息到達了，通訊協定處理常式便會擷取通訊協定標頭及內容，然後呼叫資料處理常式 (由 MIMEType TLO 屬性指定)，將訊息轉換為「回應」或「錯誤」商業物件。通訊協定處理常式會再度使用 Protocol Config MO，以設定商業物件中的通訊協定標頭。接著，通訊協定處理常式會將「回應」或「錯誤」商業物件傳回協同作業。

根據連接器的配置，可能會有一或數個通訊協定處理常式插入連接器。連接器特有的內容可讓您配置通訊協定處理常式。

### SOAP/HTTP-HTTPS 通訊協定處理常式處理程序

SOAP/HTTP(S) 通訊協定處理常式執行方式與第 65 頁的『通訊協定處理常式』中所描述的相同，但有一些例外，本節將加以說明。圖 28 顯示同步作業的 SOAP/HTTP-HTTPS 通訊協定處理常式。

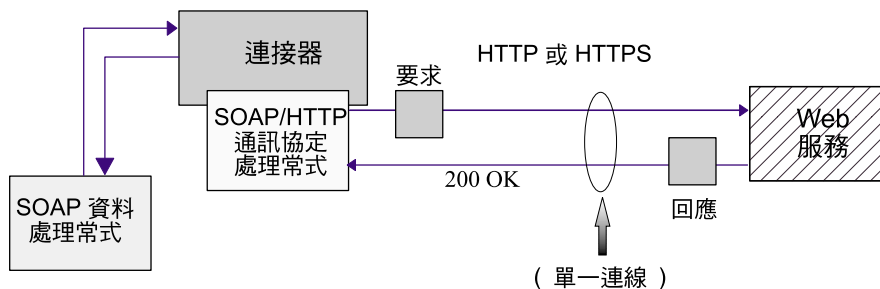


圖 28. SOAP/HTTP-HTTPS 通訊協定處理常式：同步的要求處理程序

圖 29 顯示非同步要求處理程序的 SOAP/HTTP-HTTPS 通訊協定處理常式

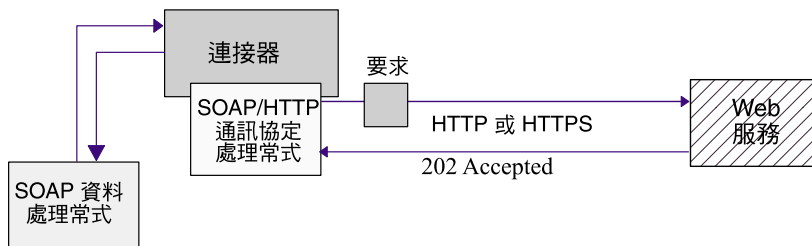


圖 29. SOAP/HTTP-HTTPS 通訊協定處理常式：非同步的要求處理程序

註：本節僅說明 SOAP/HTTP 通訊協定的處理。

SOAP/HTTP-HTTPS 通訊協定處理常式使用「SOAP 要求」商業物件的物件層次 ASI (cw\_mo\_http) 來決定 Protocol Config MO。SOAP/HTTP-HTTPS 通訊協定處理常式是藉由讀取 HTTP Protocol Config MO 中的 Destination 屬性，來決定目標 Web 服務的 URL。若該 URL 遺失或不完整，通訊協定處理常式就會使該服務呼叫失效。如需 HTTP Protocol Config MO 及其屬性的進一步資訊，請參閱第 41 頁的『要求處理程序的 HTTP Protocol Config MO』。

SOAP/HTTP-HTTPS 通訊協定處理常式會使用 SOAP 資料處理常式所傳回的 SOAP 要求訊息來呼叫 Web 服務。若已指定 HTTP Proxy 連接器配置內容，SOAP/HTTP(S) 通訊協定處理常式就會依照該配置來執行。若已傳回回應，SOAP/HTTP(S) 通訊協定處理常式就會加以讀取。

表 32 彙總 SOAP/HTTP-HTTPS 通訊協定處理常式所使用的規則優先順序，以決定送出之要求訊息的 Charset、MimeType、ContentType 及 ContentType 標頭。

表 32. 離埠訊息的 SOAP/HTTP-HTTPS 通訊協定處理常式處理程序規則

優先順序	Charset	MimeType	ContentType	ContentType 標頭
1	Protocol Config MO 的 Content-Type 標頭	TLO 屬性中的 MimeType 內容	Protocol Config MO 的 Content-Type 標頭	Protocol Config MO 的 Content-Type 標頭
2	TLO 屬性中的 Charset 內容	預設為 ContentType	預設為 text/xml	使用 ContentType 及 Charset 來建構 Content-Type 標頭
3	如果 ContentType 為 text/*，則預設值為 ISO-8859-1。否則將不會使用 charset。			

如表 32 所示：

- SOAP/HTTP-HTTPS 通訊協定處理常式會根據下列規則來決定回應訊息的 Charset：
  1. 若已於「要求」商業物件 Protocol Config MO 標頭中指定 Charset 值，便會使用該值。
  2. 若上一個步驟並未決定 Charset，通訊協定處理常式就會嘗試從 TLO 屬性擷取 Charset。
  3. 若上一個步驟中所描述的作業未順利完成，就會使用此表格來決定 Charset：

表 33. 預設的要求處理程序 Charset

ContentType	預設的 Charset
text/*	ISO-8859-1 如需進一步資訊，請參閱 RFC2616，
application/*	無預設值
所有其他值	無預設值

4. 若上一個步驟決定了 Charset，就會將該 Charset 設定於資料處理常式上。
  5. 資料處理常式以「串流」或「位元組」陣列 API 來呼叫，依寫出要求時所需的資料結構而定。
- SOAP/HTTP-HTTPS 通訊協定處理常式依據下列規則來決定要求的 MimeType：
    1. TLO MimeType 屬性。

- 2. 若遺漏了 TLO MimeType 屬性，通訊協定處理常式便會使用 ContentType 來決定 MimeType。
- SOAP/HTTP-HTTPS 通訊協定處理常式會根據下列規則來決定要求訊息的 ContentType：
  1. 如果在「回應」商業物件 Protocol Config MO 中指定了 Content-Type 標頭，則會使用標頭的類型/子類型作為 ContentType。
  2. 否則，處理常式使用預設的 ContentType：text/xml。
- SOAP/HTTP-HTTPS 通訊協定處理常式會根據下列規則來決定要求訊息的 Content-Type 標頭：
  1. 若於「要求」商業物件 Protocol Config MO 中指定 Content-Type 標頭，便會將其值設定於外送訊息中。
  2. 若未在「要求」商業物件 Protocol Config MO 中指定 Content-Type 標頭，接聽器便會使用 ContentType 及 Charset 參數來建構 Content-Type 標頭 (若已決定將該 Charset 用於要求訊息)。

當為回應訊息決定 Charset、MimeType、ContentType 及 Content-Type 標頭時，表 34 會彙總處理常式所使用的規則優先順序。

表 34. 入埠同步回應訊息的 SOAP/HTTP(s) 通訊協定處理常式處理規則

優先順序	Charset	MimeType	ContentType	ContentType 標頭
1	來自送入的 HTTP 訊息 Content-Type 標頭值的 Charset 參數值	「要求」商業物件之 Protocol Config MO 中的訊息 TransformationMap 子項商業物件	來自 Content-Type 標頭值的送入的 HTTP 訊息類型/子類型值	送入的 HTTP 訊息 Content-Type 標頭
2	「要求」商業物件之 Protocol Config MO 中的訊息 TransformationMap 子項商業物件	要求訊息 MimeType，但僅限要求與回應 ContentType 相符時。		
3	要求訊息 Charset，但僅限要求與回應 ContentType 相符時。	TLO 中的 MimeType 內容		
4	TLO 中的 Charset 內容。	預設為 ContentType		
5	若 Content-Type 為 text/*，則預設值為 ISO-8859-1。否則不會使用 Charset。			

如表 34 所示：

- 通訊協定處理常式會根據下列規則來決定同步回應訊息的 Charset：
  1. 若已在送入的回應訊息之 Content-Type 標頭中設定 Charset 參數，通訊協定處理常式便會使用該 Charset 值在資料處理常式上設定。
  2. 若回應訊息標頭中無 Charset 值，通訊協定處理常式便會嘗試從 TLO 要求的 Protocol Config MO MessageTransformationMap 來讀取協同作業定義的 Charset。
  3. 若未在 TLO 中指定 Charset 值，或者根本沒有 TLO，在此情形下，若回應的 ContentType 與要求的相同，就會以要求的 Charset 來用於回應。



4. 若上個步驟無法產生 Charset 值，通訊協定處理常式便會嘗試讀取 TLO Charset 屬性。
  5. 如果無法用前幾個步驟中所說明的方法來取得 Charset 值，且訊息 ContentType 的類型為 text 且具有任何子類型 (例如 text/xml、text/plain 等)，則預設值為 ISO-8859-1。否則不會使用 Charset 值。
- 通訊協定處理常式會根據下列規則來決定同步回應訊息的 MIMEType：
    1. 首先，通訊協定處理常式會嘗試從 TLO 要求 Protocol Config MO 的 MessageTransformationMap 來擷取 MIMEType。尤其是，通訊協定處理常式會嘗試於 MTM 中尋找正確的 ContentType 相符項目來擷取 MessageTransformationRule，然後使用其中的 MIMEType 內容值。否則，通訊協定處理常式便會尋找適用於多個 ContentType (ContentType 為 \*/\*) 的 MessageTransformationRule。
    2. 若無法藉由使用 MessageTransformationMap 來決定 MIMEType，則只有當要求和回應的 ContentType 相符時，通訊協定處理常式才會將要求的 MIMEType 用於回應的 MIMEType。
    3. 若無法使用前面的步驟來擷取 MIMEType，通訊協定處理常式便會使用 TLO 的 MIMEType 屬性或預設的 MIMEType (若適用於該通訊協定處理常式)。
    4. 若前面所有的步驟皆失敗，通訊協定處理常式會使用 ContentType 來設定 MIMEType。
  - 處理常式可擷取送入的 HTTP 訊息 Content-Type 標頭的類型/子類型，以判斷 ContentType。

處理常式會處理 HTTP Protocol Config MO。協同作業要負責確定，在「要求-回應」事件的環境定義中，傳遞至 HTTP Protocol Config MO 中的標頭值是正確的。處理常式會根據下列規則，載入標準標頭和自訂內容：

1. 處理常式會調查 HTTP Protocol Config MO 的每個項目，以忽略特殊屬性 (例如 ObjectEventId)。
2. 每個非空值的標頭都會放在外送訊息，且有可能發生其他處理程序 (例如 Content-Type 標頭)。
3. 請記得，利用上述方法時，處理常式可能會在訊息中設定非標準的標頭，但不保證該訊息的邏輯及語法是正確的。
4. 若 HTTP Protocol Config MO UserDefinedProperties 中有一或數項自訂內容，處理常式便會在「實體標頭區段」(最後一個標頭區段) 中新增它們。如需自訂內容的相關資訊，請參閱第 43 頁的『要求處理程序的使用者定義內容』。

**註：**在 HTTP Protocol Config MO 中指定下列任一標頭，很有可能會導致不正確的 HTTP 訊息：Connection、Trailer、Transfer-Encoding、Content-Encoding、Content-Length、Content-MD5、Content-Range。

## SOAP/JMS 通訊協定處理常式處理程序

SOAP/JMS 通訊協定處理常式執行方式與第 65 頁的『通訊協定處理常式』中所描述的可相同，但有一些例外，本節將加以說明。

**註：**若 LookupQueueUsingJNDI 配置內容設定為 true，則 SOAP/JMS 通訊協定處理常式會使用 JNDI 來查閱目的地佇列。JNDI 內容指定於連接器內容中。如需進一步資訊，請參閱第 72 頁的『連接器及 JMS』，以及第 76 頁的『連接器特有的配置內容』中的 JNDI 相關內容。

SOAP/JMS 通訊協定處理常式使用 SOAP 資料處理常式所傳回的 Web 服務要求訊息主體來建立 JMS 傳輸訊息，JMS 標頭的設定如表 35 所示。回應內容的類型是根據「SOAP 要求」商業物件的 JMS Protocol Config MO 中的 MessageType 屬性值而定。如果 MessageType 是「文字」，處理常式就會透過「字串」資料處理常式 API，將「SOAP 要求」商業物件轉換成「字串」。如果 MessageType 是「位元組」，處理常式就會透過「位元組」資料處理常式 API，將「SOAP 要求」商業物件轉換成位元組陣列。(預設訊息內容類型是 TextMessage)。

表 35. SOAP/JMS 通訊協定處理常式設定於要求訊息中的標頭值

JMS 標頭名稱	如果未在 SOAP/JMS Protocol Config MO 中設定則為預設值
JMSPriority	4
JMSExpiration	0
JMSDeliveryMode	PERSISTENT
JMSReply	
JMSCorrelationId	
JMSRedelivered	
JMSTimestamp	
JMSType	

若是非同步呼叫目標 Web 服務，則不會設定 JMSReplyTo 標頭。否則 (就同步處理程序而言)，SOAP/JMS 通訊協定處理常式會設定 JMSReplyTo 標頭。SOAP/JMS 通訊協定處理常式使用 ReplyToQueue 配置內容來取得 JMSDestination (從目標 Web 服務傳回回應或錯誤的目的地)，並且將其設定於 JMS 傳輸訊息的 JMSReplyTo 標頭上。

圖 30 顯示同步要求作業的 SOAP/JMS 通訊協定處理常式處理程序。

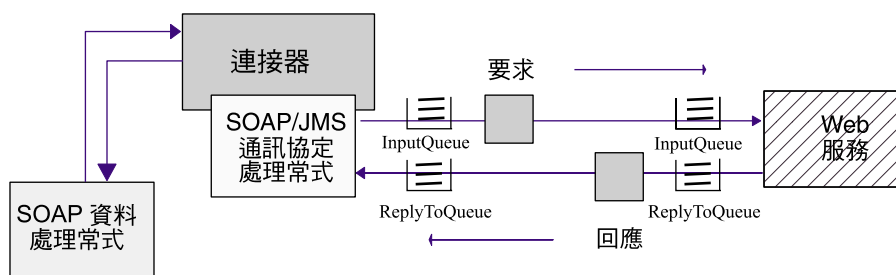


圖 30. SOAP/JMS 通訊協定處理常式：同步的要求處理程序

圖 31 顯示非同步要求作業的 SOAP/JMS 通訊協定處理常式處理程序。

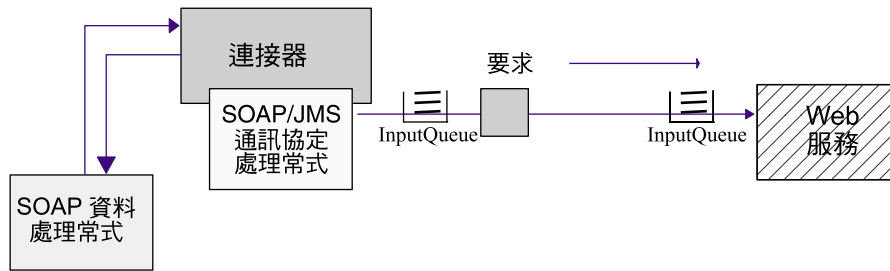


圖 31. SOAP/JMS 通訊協定處理常式：非同步的要求處理程序

SOAP/JMS 通訊協定處理常式使用「SOAP 要求」商業物件的物件層次 ASI (cw\_mo\_jms) 來決定 Protocol Config MO。Protocol Config MO 的 Destination 屬性提供目標 Web 服務的佇列名稱。若已啟用 JNDI，SOAP/JMS 通訊協定處理常式就會查閱 JNDI 物件，以取得 SOAP 要求訊息的 JMSDestination。否則就會使用 SOAP Protocol Config MO 中的 Destination 屬性。

若回應未於 ResponseWaitTimeout 內容中所指定的時間內到達，SOAP/JMS 通訊協定處理常式就會使協同作業要求失效。當 SOAP 回應 (或錯誤) 訊息到達時，SOAP/JMS 通訊協定處理常式會擷取 JMS 標頭及內容，以供 SOAP 資料處理常式進行轉換。處理常式會決定送入的訊息的內容類型，並將資訊儲存在 JMS Protocol Config MO 的 MessageType 屬性中。內容可為 TextMessage 或 BytesMessage。若為 TextMessage 格式，則處理常式會透過「字串 API」，以擷取為「字串」的 Web 服務回應訊息來呼叫資料處理常式。若為 BytesMessage，則處理常式會透過「位元組」資料處理常式 API，以擷取為位元組陣列的 Web 服務回應訊息來呼叫資料處理常式。然後 SOAP/JMS 通訊協定處理常式會使用「回應」(或「錯誤」) 商業物件中的 Protocol Config MO 來對映 JMS 標頭，以於 TLO 中設定 SOAP 回應 (或錯誤) 商業物件。表 36 顯示此對映表。

表 36. Protocol Config MO--同步要求處理程序期間，回應的 JMS 標頭屬性對映

Protocol Config MO 屬性	JMS 標頭名稱	說明
Destination	JMSDestination	來自回應訊息的 JMSDestination 標頭。
MessageId	JMSMessageId	來自回應訊息的 JMSMessageId 標頭。
優先順序	JMSPriority	來自回應訊息的 JMSPriority 標頭
Expiration	JMSExpiration	來自回應訊息的 JMSExpiration 標頭
DeliveryMode	JMSDeliveryMode	來自回應訊息的 JMSDeliveryMode 標頭。
ReplyTo	JMSReplyTo	來自回應訊息的 JMSReplyTo 標頭。 JMS API 以 JMSDestination 的形式傳回此標頭，但 SOAP/JMS 通訊協定接聽器則是傳回佇列名稱。
CorrelationId	JMSCorrelationId	來自回應訊息的 JMSCorrelationId 標頭
Redelivered	JMSRedelivered	來自回應訊息的 JMSRedelivered 標頭。
TimeStamp	JMSTimeStamp	來自回應訊息的 JMSTimeStamp 標頭
類型	JMSType	來自回應訊息的 JMSType 標頭

表 36. Protocol Config MO--同步要求處理程序期間，回應的 JMS 標頭屬性對映 (繼續)

MessageType	不適用	回應訊息內容的類型。這個屬性的值對於 TextMessage 內容來說是「文字」；對於 BytesMessage 內容來說是「位元組」。
-------------	-----	--

然後 SOAP/JMS 通訊協定處理常式會將 TLO 傳回協同作業。

## 連接器及 JMS

**註:** 本節假設您已熟悉 JMS 及 JNDI，尤其是 JMS 的運作方式。若需進一步的資訊，請參閱您的 JMS 及 JNDI 原始檔文件。

連接器可將協同作業顯現為 SOAP/JMS Web 服務，並可讓協同作業來呼叫 SOAP/JMS Web 服務。將 SOAP/JMS 與 Web 服務連接器搭配使用的基本需求如下：

1. 您已安裝並配置您的 JMS 服務提供者。
2. 您已安裝並配置您的 JNDI。
3. 您的 JMS 提供者可支援 JMS API 1.0.2 版。
4. 所有必要的 Jar 檔皆於連接器的類別路徑中。(請參閱您的 JMS 提供者文件，以決定所有必要的 Jar 檔。)
5. 所有必要的程式庫皆於連接器的路徑中。(請參閱您的 JMS 提供者文件，以決定所有必要的程式庫。)

## JNDI

若為 SOAP/JMS，連接器使用 JNDI 來查閱使用 JNDI 環境定義的 Connection Factory。於起始設定期間，連接器會讀取 JNDI 連接器特有的內容，以連接 JNDI。若您未配置此內容，您將無法使用 SOAP/JMS。您可指定下列 JNDI 連接器特定的內容：

- JNDIProviderURL
- InitialContextFactory
- JNDIConnectionFactoryName
- CTX\_ObjectFactories
- CTX\_ObjectFactories
- CTX\_StateFactories
- CTX\_URLPackagePrefixes
- CTX\_DNS\_URL
- CTX\_Authoritative
- CTX\_Batchsize
- CTX\_Referral
- CTX\_SecurityProtocol
- CTX\_SecurityAuthentication
- CTX\_SecurityPrincipal
- CTX\_SecurityCredentials
- CTX\_Language
- LookupQueuesUsingJNDI

如需指定這些內容的指引，請參閱您的 JNDI 文件。若要將 SOAP/JMS 與連接器搭配使用，需要下列 JNDI 連接器特有的內容：

- **JNDIProviderURL** 將此內容設定為「JNDI 服務」提供者的 URL。如需此內容的值，請參閱您的 JNDI 提供者文件。
- **InitialContextFactory** 將此內容設定為要建立 JNDI 起始環境定義之 Factory 類別的完整類別名稱。如需此內容的值，請參閱您的 JNDI 提供者文件。請確定此類別及其相依關係皆於連接器的類別路徑中。
- **JNDIConnectionFactoryName** 將此內容設定為所要查閱之 Connection Factory 的 JNDI 名稱 (使用 JNDI 環境定義)。請確定可使用 JNDI 來查閱此名稱。

若您將 `LookupQueuesUsingJNDI` 設為 `true`，請確定連接器使用的所有佇列皆可使用 JNDI 來加以查閱。

## 將協同作業顯現為 SOAP/JMS Web 服務

若要將協同作業顯現為 SOAP/JMS Web 服務，您必須使用 SOAP/JMS 通訊協定接聽器。使用 SOAP/JMS 通訊協定接聽器需要您指定 JNDI 連接器內容。

您的 JMS 提供者配置應反映 SOAP/JMS 通訊協定接聽器的基本需求。請確定您的 JMS 服務提供者已定義了 SOAP/JMS 通訊協定接聽器需要的所有佇列。請務必檢查您的 JMS 提供者文件 -- 定義佇列的作業隨提供者而有所不同。您必須為 SOAP/JMS 通訊協定接聽器定義六個佇列。您必須於 SOAP/JMS 接聽器配置內容中設定佇列名稱。如果您已將 `JNDI` 的 `LookupQueuesUsingJNDI` 設定為 `true`，亦必須於 SOAP/JMS 接聽器配置內容中指定佇列的 JNDI 名稱。

您應指定下列 SOAP/JMS 接聽器配置內容中的佇列名稱值：

- `InputQueue`
- `InProgressQueue`
- `ArchiveQueue`
- `UnsubscribedQueue`
- `ErrorQueue`
- `ReplyToQueue`

`InputQueue` 及 `InProgressQueue` 為必要內容。請確定您已正確地配置這些佇列。

`ArchiveQueue`、`UnsubscribedQueue` 及 `ErrorQueue` 是選用性內容。這些佇列可用來保存 Web 服務要求。若您計劃使用這些內容的任何一項，請確定您已正確地配置對應的 JMS 佇列。當與您的 JMS 提供者定義這些佇列時，應謹慎地指定這些佇列的容量。

## 呼叫 SOAP/JMS Web 服務的協同作業

若要啟用協同作業來呼叫 SOAP/JMS Web 服務，請使用 SOAP/JMS 通訊協定處理常式。SOAP/JMS 通訊協定處理常式需要您指定 JNDI 連接器內容。與您的 Web 服務提供者一起決定 JMS 及 JNDI 基本需求。

若要呼叫 SOAP/JMS Web 服務，連接器需要將 SOAP/JMS Protocol Config MO 中的 `Destination` 屬性值設定為目標 Web 服務的輸入佇列。若您已將 `JNDI` 的 `LookupQueuesUsingJNDI` 設定為 `true`，則必須指定輸入佇列的 JNDI 名稱。

若您要呼叫要求-回覆 Web 服務，必須與您的 Web 服務提供者一起決定 ReplyTo 佇列的基本需求。請確定已定義 ReplyTo 佇列。亦請確定您已指定於 SOAP/JMS 通訊協定處理常式的 ReplyToQueue 配置內容中，指定 ReplyTo 佇列的名稱。若 JNDI ”LookupQueuesUsingJNDI” 的設定為 true，則 ReplyToQueue 配置內容值中應提供此佇列的 JNDI 名稱。

請特別注意，通訊協定處理常式和通訊協定接聽器不同，通訊協定處理常式無法插入 Web 服務連接器。因此，對於連接器所呼叫的所有要求-回覆 Web 服務，連接器都會使用相同的 ReplyTo 佇列。

---

## SSL

本節討論連接器如何實行 SSL 功能。如需背景資訊，請參閱您的 SSL 文件。本節假設您已熟悉 SSL 技術。

### JSSE

連接器可將協同作業顯現為 SOAP/HTTPS Web 服務，並可讓協同作業呼叫 SOAP/HTTPS Web 服務。連接器使用 JSSE 提供 HTTPS 及 SSL 的支援。IBM JSSE 隨連接器一起出貨。若要啟用此功能，請確定 java.security 檔案 (隨著連接器一同安裝的檔案) 中已含有下列項目：

```
security.provider.5=com.ibm.jsse.IBMJSSEProvider
```

請注意，java.security 位於安裝連接器的 \$ProductDir\lib\security 目錄中。連接器使用 JavaProtocolHandlerPackages 連接器內容的值來設定系統內容 java.protocol.handler.pkgs。請注意，就與連接器一起出貨的 IBM JSSE 而言，此內容的值應設定為 com.ibm.net.ssl.internal.www.protocol。JavaProtocolHandlerPackages 配置內容預設為此值。不過，如果您系統中的 java.protocol.handler.pkgs 系統內容具有非空值，則只有在同時設定 JavaProtocolHandlerPackages 連接器內容時，連接器才會改寫它。

於起始設定期間，連接器停用 JSSE 支援的所有匿名密碼組合。

### KeyStore 及 TrustStore

若要搭配連接器來使用 SSL，您必須設定金鑰儲存庫及信任儲存庫。但是並未提供任何工具來設定金鑰儲存庫、憑證及金鑰產生作業。您必須使用協力廠商軟體工具來完成這些作業。

### SSL 內容

您可以指定下列 SSL 連接器特有內容：

- SSLVersion
- SSLDebug
- KeyStore
- KeyStoreAlias
- KeyStorePassword
- TrustStore
- TrustStorePassword

請注意，這些內容套用在連接器實例上。插入該連接器的所有 SOAP/HTTPS 通訊協定接聽器，及每個連接器實例的 SOAP/HTTP-HTTPS 通訊協定處理常式，皆使用同一組 SSL 內容值。如需 HTTPS/SSL 設定的進一步資訊，請參閱第 203 頁的附錄 E，『配置 HTTPS/SSL』。

## 將協同作業顯現為 SOAP/HTTPS Web 服務

當您將協同作業顯現為 SOAP/HTTPS Web 服務時，您要使用 SOAP/HTTPS 通訊協定接聽器。若要使用 SOAP/HTTPS 通訊協定接聽器，您必須指定 SSL 連接器特有內容。您指定給這些內容的值，應反映您的 SSL 基本需求。

- **SSLVersion** 請確定 JSSE 可支援您所要使用的 SSLVersion。
- **KeyStore** 因為 SOAP/HTTPS 通訊協定接聽器於 SSL 通訊中的作用如同伺服器，所以您必須指定金鑰儲存庫。接聽器使用 SSL ” KeyStore 配置內容中所指定的金鑰儲存庫。此內容的值必須是通往金鑰儲存庫檔的完整路徑。請確定金鑰儲存庫具有連接器的金鑰組 (私密金鑰及公開金鑰)。私密金鑰的別名應指定為 SSL ” KeyStoreAlias 內容的值。用來存取金鑰儲存庫的密碼必須指定為 SSL ” KeyStorePassword 內容的值。亦請確定存取金鑰儲存庫所需的密碼及私密金鑰 (在金鑰儲存庫中) 是相同的。最後，您必須將連接器的數位憑證分送至您的 Web 服務用戶端，使它們可以鑑別該連接器。
- **TrustStore** 若您要 SOAP/HTTPS 通訊協定接聽器來鑑別 Web 服務用戶端，則必須啟動用戶端鑑別功能。將 SSL ” UseClientAuth 內容設定為 true 即可。您亦必須指定：
  - 信任儲存庫的位置，與 SSL ” TrustStore 配置內容的值相同
  - 存取信任儲存庫所需的密碼，與 SSL ” TrustStorePassword 內容的值相同

請確定您的信任儲存庫包含了 Web 服務用戶端的數位憑證。「Web 服務」用戶端所使用的數位憑證可為自行簽章的，亦可由 CA 簽發。請注意，若您的信任儲存庫信任 CA 的主要憑證，JSSE 就會鑑別 CA 簽發的所有數位憑證。

如需 HTTPS/SSL 設定的進一步資訊，請參閱第 203 頁的附錄 E，『配置 HTTPS/SSL』。

## 呼叫 SOAP/HTTPS Web 服務的協同作業

若要啟用協同作業來呼叫 SOAP/HTTPS Web 服務，請使用 SOAP/HTTP-HTTPS 通訊協定處理常式。如果您將 SSL 搭配 SOAP/HTTP-HTTPS 通訊協定處理常式來使用，則必須指定 SSL 連接器特有內容。您指定給這些內容的值，應反映 Web 服務提供者的 HTTPS/SSL 基本需求：

- **SSLVersion** 請確定您的 Web 服務提供者及 JSSE 可支援您要使用的 SSLVersion。
- **TrustStore** 由於 SOAP/HTTP-HTTPS 通訊協定處理常式的作用如同 SSL 通訊作業中的用戶端，所以您必須設定信任儲存庫。處理常式使用 SSL -> Truststore 配置內容中所指定的信任儲存庫。此內容的值必須是通往信任儲存庫檔的完整路徑。您必須於 SSL -> TrustStorePassword 內容中指定存取信任儲存庫時所需的密碼。請確定您的信任儲存庫包含了 Web 服務提供者的數位憑證。Web 服務提供者所使用的數位憑證可為自行簽章的，亦可由 CA 簽發。請注意，若您的信任儲存庫信任 CA 的主要憑證，JSSE 就會鑑別 CA 簽發的所有數位憑證。
- **KeyStore** 若您的 Web 服務提供者要求用戶端鑑別，則必須設定金鑰儲存庫。SOAP/HTTP-HTTPS 通訊協定處理常式使用 SSL ” KeyStore 配置內容中所指定的金鑰儲存庫。此值必須是通往金鑰儲存庫檔的完整路徑。請確定金鑰儲存庫有為連接器所配置的金鑰組 (私密金鑰及公開金鑰)。必須於 SSL ” KeyStoreAlias 內容中指定

私密金鑰的別名。必須於 `SSL` 的 `KeyStorePassword` 內容中指定存取金鑰儲存庫所需的密碼。最後，請確定存取金鑰儲存庫所需的密碼及私密金鑰 (在金鑰儲存庫中) 是相同的。您必須將連接器的數位憑證分送給您的 Web 服務提供者，以供鑑別。

如需 `HTTPS/SSL` 設定的進一步資訊，請參閱第 203 頁的附錄 E, 『配置 `HTTPS/SSL`』。

## 配置連接器

使用「安裝程式」將連接器檔案安裝至您的系統後，您必須設定標準的及應用程式特有的連接器配置內容。

### 設定配置內容

連接器有兩種配置內容類型：標準配置內容和連接器特有配置內容。在執行連接器之前，您必須使用「系統管理程式 (SM)」來設定這些內容的值。

#### 標準配置內容

標準配置內容提供所有連接器使用的資訊。請參閱第 153 頁的附錄 A, 『連接器的標準配置內容』，以取得這些內容的相關文件。下表提供附錄中有關此連接器配置內容的特有資訊。

內容	說明
<code>CharacterEncoding</code>	此連接器不使用此內容。
<code>Locale</code>	由於此連接器尚未國際化，所以您無法變更此內容的值。請參照連接器的版本注意事項，判斷目前所支援的語言環境。

因為此連接器僅支援 `InterChange Server (ICS)` 作為整合分配管理系統，所以與其相關的配置內容僅有 `ICS` 的配置內容。

您至少必須設定下列標準連接器配置內容：

- `AgentTraceLevel`
- `ApplicationName`
- `ControllerTraceLevel`
- `DeliveryTransport`

#### 連接器特有的配置內容

連接器特有的配置內容提供連接器代理程式於執行時期所需的資訊。連接器特有內容亦提供一種方法，可直接變更連接器代理程式中的靜態資訊或邏輯，不需重新編碼和重新建置代理程式。

表 37 列出連接器特有的配置內容。請參閱後續幾節的內容說明。請注意，有些內容包含了其他內容。+ 字元指出輸入項目在內容階層中的位置。

**註：**若您不打算將 `SOAP/JMS` 通訊協定接聽器或 `SOAP/JMS` 通訊協定處理常式搭配連接器來使用，請務必刪除與 `SOAP/JMS` 相關的連接器特有內容，或是將其留白。

表 37. 特有連接器配置內容

名稱	可能值	預設值	必要的
<code>ConnectorType</code>	任何有效的連接器類型	<code>WebService</code>	是
<code>DataHandlerMetaObjectName</code>	資料處理常式 <i>Meta</i> 物件名稱	<code>MO_DataHandler_Default</code>	是



表 37. 特有連接器配置內容 (繼續)

名稱	可能值	預設值	必要的
JavaProtocolHandlerPackages	有效的 <i>Java</i> 通訊協定處理常式套件	com.ibm.net.ssl. internal.www.protocol	否
ProtocolHandlerFramework	此為階層式內容，沒有值	無	否
+ProtocolHandlers	此為階層式內容，沒有值		否
++SOAPHTTPHTTPSHandler	此為階層式內容。如需其子內容的相關資訊，請參閱第 78 頁的『SOAPHTTPHTTPSHandler』。		是
++SOAPJMSHandler	此為階層式內容。如需其子內容的相關資訊，請參閱第 79 頁的『SOAPJMSHandler』。		
ProtocolListenerFramework	此為階層式內容，沒有值。		否
+WorkerThreadCount	不小於 1 的整數，指示可用的接聽器執行緒數量。	10	否
+RequestPoolSize	大於 <i>WorkerThreadCount</i> 的整數，指示資源儲存池的大小。	20	否
+ProtocolListeners	此為階層式內容，沒有值		
++Listener1	具有唯一名稱的通訊協定接聽器		是
+++Protocol	soap/http, soap/https, soap/jms		是
+++SOAPDHMimeType	SOAP 資料處理常式的任何有效 <i>MIME</i> 類型	xml/soap	
+++ListenerSpecific	接聽器專屬或必要的內容請參閱第 81 頁的『ListenerSpecific』。		
ProxyServer	此為階層式內容，沒有值		否
+HttpProxyHost	<i>HTTP Proxy</i> 伺服器的主機名稱		否
+HttpProxyPort	<i>HTTP Proxy</i> 伺服器的埠號	80	否
+HttpNonProxyHosts	需要直接連線的 <i>HTTP</i> 主機		否
+HttpsProxyHost	<i>HTTPS Proxy</i> 伺服器的主機名稱		否
+HttpsProxyPort	<i>HTTPS Proxy</i> 伺服器的埠號	443	否
+HttpsNonProxyHosts	需要直接連線的 <i>HTTPS</i> 主機		否
+SocksProxyHost	<i>Socks Proxy</i> 伺服器名稱		否
+SocksProxyPort	<i>Socks Proxy</i> 伺服器埠		否
+HttpProxyUsername	<i>Http Proxy</i> 伺服器使用者名稱		否
+HttpProxyPassword	<i>Http Proxy</i> 伺服器密碼		否
+HttpsProxyUsername	<i>Https Proxy</i> 伺服器使用者名稱		否
+HttpsProxyPassword	<i>Https Proxy</i> 伺服器密碼		否
SSL	此為階層式內容，沒有值		否
+SSLVersion	SSL, SSLv2, SSLv3, TLS, TLSv1	SSL	否
+SSLDebug	true, false	false	否
+KeyStoreType	任何有效的金鑰儲存庫類型	JKS	否
+KeyStore	<i>KeyStore</i> 檔案的路徑。		否
+KeyStorePassword	<i>KeyStore</i> 中的私密金鑰密碼		否
+KeyStoreAlias	<i>KeyStore</i> 中的金鑰組別名		否
+TrustStore	<i>TrustStore</i> 檔案的路徑		否
+TrustStorePassword	<i>TrustStore</i> 的密碼		否
+UseClientAuth	true false	false	否
WSCollaborations	此為「WSDL 配置精靈」建立的階層式內容，沒有值。請參閱第 90 頁的『WSCollaborations』。		
+Collaboration1	此為階層式內容，沒有值		
++CollaborationPort1	協同作業埠的名稱		是
+++WebServiceOperation1	此為階層式內容，沒有值		是

表 37. 特有連接器配置內容 (繼續)

名稱	可能值	預設值	必要的
++++BodyName	Web 服務方法的名稱；必須是有效的 XML 元素名稱		是
++++BodyNS	Web 服務方法的名稱空間；必須是有效的 XML 名稱空間		是
++++BOName	用於作業的「要求」商業物件名稱		是
++++Mode	synch asynch	asynch	否
JNDI	此為與 JMS 相關的階層式內容，沒有值		否
+LookupQueuesUsingJNDI	true false	false	否
+JNDIProviderURL	有效的 JNDI URL		否
+InitialContextFactory	起始環境定義的 Factory 類別名稱		否
+JNDIConnectionFactoryName	使用 JNDI 環境定義來查閱的 Connection Factory 名稱。		否
+CTX_ObjectFactories	內容指定有關 JNDI 環境定義中安全性及物件查閱的其他資訊		否
+CTX_properties			

**ConnectorType:** 若此內容設定為 `WebService`，則於連結協同作業埠時，「系統管理程式」會將連接器顯示為 Web 服務連接器。否則會顯示為正常的連接器。

預設值 = `WebService`。

**DataHandlerMetaObjectName:** 此為 Meta 物件的名稱，資料處理常式使用此名稱來設定配置內容。

預設值 = `MO_DataHandler_Default`。

**JavaProtocolHandlerPackages:** 這個內容的值提供「Java 通訊協定處理常式」套件。連接器使用此內容的值來設定系統內容 `java.protocol.handler.pkgs`。

預設值 = `com.ibm.net.ssl.internal.www.protocol`。

**ProtocolHandlerFramework:** 「通訊協定處理常式組織架構」使用此內容來載入及配置其通訊協定處理常式。此為階層式內容，沒有值。

預設值 = 無。

**ProtocolHandlers:** 此階層式內容沒有值。其第一層的子項代表分離的通訊協定處理常式。

預設值 = 無。

**SOAPHTTPHTTPSHandler:** SOAP/HTTP-HTTPS 通訊協定處理常式的名稱。請注意，此為階層式內容。通訊協定處理常式與接聽器不同，通訊協定處理常式不可複製，且每個通訊協定只能有一個處理常式。下面的表 38 顯示 SOAP/HTTP-HTTPS 通訊協定處理常式的子內容。+ 字元指出輸入項目在內容階層中的位置。

表 38. SOAP/HTTP-HTTPS 通訊協定處理常式配置內容

名稱	可能值	預設值	必要的
++SOAPHTTPHTTPSHandler	此為階層式內容，沒有值。		是

表 38. SOAP/HTTP-HTTPS 通訊協定處理常式配置內容 (繼續)

名稱	可能值	預設值	必要的
+++Protocol	處理常式所實行的通訊協定類型。若為 SOAP/HTTP 及 SOAP/HTTPS，此值為 soap/http。 註: 若您未指定此內容的值，連接器將不會起始設定此通訊協定處理常式。		是
+++HTTPReadTimeout	SOAP/HTTP 特有的內容，它可指定從遠端主機 (Web 服務) 讀取時的逾時間隔 (毫秒)。若未指定這個內容，或是將它設定為 0，則從遠端主機讀取時，通訊協定處理常式將會無限期地暫停執行。	0	否

圖 32 顯示如「連接器配置程式」中所顯示的內容。

Standard Properties		Application Config Properties		Supported Business Objects		Trace	
	Property	Value	Update Method	Encrypt			
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>			
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>			
3	<input type="checkbox"/> ProxyServer		agent restart	<input type="checkbox"/>			
4	<input type="checkbox"/> SSL		agent restart	<input type="checkbox"/>			
5	<input type="checkbox"/> ProtocolListenerFramework		agent restart	<input type="checkbox"/>			
6	<input type="checkbox"/> ProtocolHandlerFramework		agent restart	<input type="checkbox"/>			
7	<input type="checkbox"/> ProtocolHandlers		agent restart	<input type="checkbox"/>			
8	<input type="checkbox"/> SOAPHTTPHTTTPHandler		agent restart	<input type="checkbox"/>			
9	Protocol	soap/http	agent restart	<input type="checkbox"/>			
10	HTTPReadTimeout	0	agent restart	<input type="checkbox"/>			
11	<input type="checkbox"/> SOAPJMSHandler		agent restart	<input type="checkbox"/>			
12	<input type="checkbox"/> JNDI		agent restart	<input type="checkbox"/>			
13	<input type="checkbox"/> WSCollaborations		agent restart	<input type="checkbox"/>			

圖 32. SOAP/HTTP-HTTPS 通訊協定處理常式內容

**SOAPJMSHandler:** SOAP/JMS 通訊協定處理常式的名稱。請注意，此為階層式內容。通訊協定處理常式與接聽器不同，通訊協定處理常式不可複製，且每個通訊協定只能有一個處理常式。下面的表 39 顯示 SOAP/JMS 通訊協定處理常式的子內容。+ 字元指出輸入項目在內容階層中的位置。

表 39. SOAP/JMS 通訊協定處理常式配置內容

名稱	可能值	預設值	必要的
++SOAPJMSHandler	此為階層式內容，沒有值。		是
+++Protocol	處理常式所實行的通訊協定類型。若為 SOAP/JMS，則該值為 soap/jms。 註: 若您未指定此內容的值，連接器將不會起始設定此通訊協定處理常式。		是
+++ResponseWaitTimeout	這是 JMS 通訊協定特定處理常式的內容，其指定通訊協定處理常式在 ReplyToQueue 上等待同步要求處理程序的逾時值 (毫秒)。若回應未於此期限內到達，處理常式將使協同作業要求失效。若未指定這個內容，或其設定為 0，通訊協定處理常式將在 ReplyToQueue 上一直等待。	0	否

表 39. SOAP/JMS 通訊協定處理常式配置內容 (繼續)

名稱	可能值	預設值	必要的
+++ReplyToQueue	此為必要的 JMS 通訊協定特定處理常式內容，用以為 ReplyTo 佇列命名。若為同步要求處理程序，處理常式會將 JMSReplyTo 欄位設為這個 JMS 目的地。如果 LookupQueuesUsingJNDI = true，SOAP/JMS 通訊協定處理常式就會使用 JNDI 來查閱此佇列。	無	是

圖 33 顯示如「連接器配置程式」中所顯示的內容。

Standard Properties		Application Config Properties		Supported Business Objects		Trace/Log Files		De	
	Property	Value	Update	Encrypt	Description				
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>					
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>					
3	<input type="checkbox"/> ProxyServer		agent restart	<input type="checkbox"/>					
4	<input type="checkbox"/> SSL		agent restart	<input type="checkbox"/>					
5	<input type="checkbox"/> ProtocolListenerFramework		agent restart	<input type="checkbox"/>					
6	<input type="checkbox"/> ProtocolHandlerFramework		agent restart	<input type="checkbox"/>					
7	<input type="checkbox"/> ProtocolHandlers		agent restart	<input type="checkbox"/>					
8	<input type="checkbox"/> SOAPHTTPHandler		agent restart	<input type="checkbox"/>					
9	<input type="checkbox"/> SOAPJMSHandler		agent restart	<input type="checkbox"/>					
10	Protocol	soap/jms	agent restart	<input type="checkbox"/>					
11	ResponseWaitTimeout	0	agent restart	<input type="checkbox"/>					
12	ReplyToQueue		agent restart	<input type="checkbox"/>					
13	<input type="checkbox"/> JNDI		agent restart	<input type="checkbox"/>					
14	<input type="checkbox"/> WSCollaborations		agent restart	<input type="checkbox"/>					

圖 33. SOAP/JMS 通訊協定處理常式內容

**ProtocolListenerFramework:** 通訊協定接聽器組織架構使用這個內容來載入通訊協定接聽器。此為階層式內容，沒有值。

**WorkerThreadCount:** 這個內容必須是不小於 1 的整數，建立可用於通訊協定接聽器組織架構的通訊協定接聽器工作程式執行緒數量。如需進一步資訊，請參閱第 55 頁的『通訊協定接聽器』。預設值 = 10。

**RequestPoolSize:** 這個內容必須是大於 WorkerThreadCount 的整數，設定通訊協定接聽器組織架構的資源儲存池大小。組織架構可同時處理的要求上限為 WorkerThreadCount + RequestPoolSize。

預設值 = 20。

**ProtocolListeners:** 此為階層式內容，沒有值。這個內容的每個第一層子項代表分離的通訊協定接聽器。

**Listener1:** 通訊協定接聽器的名稱。可能會有多重通訊協定接聽器。請注意，此為階層式內容。您可建立這個內容的多重實例，並建立額外的、具有唯一名稱的接聽器。執行時，您可變更特定接聽器的內容，但不能變更通訊協定內容。多重接聽器的名稱必須是唯一的。可能的名稱 (非值)：SOAPHTTPListener1、SOAPHTTPSListener1、SOAPJMSListener1

**Protocol:** 這個內容指定此接聽器所要實行的通訊協定。可能的值：soap/http、soap/https、soap/jms。

**註:** 若您未指定這個內容的值，連接器將不會起始設定此通訊協定接聽器。

**SOAPDHMimeType:** SOAP 資料處理常式 MIME 類型，用於此接聽器所接收的要求。

預設值 = xml/soap

**ListenerSpecific:** 特定的接聽器內容，對所指定的通訊協定接聽器來說是唯一的，或是必要的。例如，HTTP 接聽器具有特定接聽器的內容--「埠」，其代表「接聽器」用來監視要求的「埠號」。表 40 彙總 HTTP-HTTPS 接聽器特定的內容。+ 字元指出輸入項目在內容階層中的位置。

表 40. SOAP/HTTP 及 SOAP/HTTPS 通訊協定特定接聽器的配置內容

名稱	可能值	預設值	必要的
+++SOAPHTTPListener1	HTTP 通訊協定接聽器的專屬名稱。這是 ProtocolListenerFramework -> ProtocolListeners 階層式內容的子項。可以有重接聽器：您可以另外建立此內容的實例及其階層，以外掛其他 HTTP 接聽器。		是
++++Protocol	soap/http (若為 SOAP/HTTP 通訊協定接聽器) soap/https (若為 SOAP/HTTPS 通訊協定接聽器) <b>註:</b> 若您未指定這個內容的值，連接器將不會起始設定此通訊協定接聽器。		是
++++SOAPDHMimeType	xml/soap 這個內容的值會傳遞至資料處理常式。	xml/soap	否
++++BOPrefix			否
++++Host	接聽器將於這個內容值所指定的 IP 位址上接聽。若未指定 Host，其預設為 localhost。請注意，您可以指定主機名稱 (DNS 名稱)，亦可指定接聽器執行所在之機器的 IP 位址。一台機器可以有重 IP 位址或多重名稱。	localhost	否
++++Port	接聽器用來接聽要求的埠。如果未指定，若為 SOAP/HTTP，埠預設值為 80，若為 SOAP/HTTPS，則為 443。如果您在連接器中複製接聽器，則 Host 和 Port 內容的組合必須是唯一的，否則接聽器就無法連結至該埠來接受要求。	80 (若為 SOAP/HTTP 接聽器) 443 (若為 SOAP/HTTPS 接聽器)	否
++++SocketQueueLength	送入之連線要求的佇列 (socket 佇列) 長度。指定主機在一次儲存多少送入的連線之後，即拒絕連線。佇列長度的上限需視作業系統而定。	5	否
++++RequestWaitTimeout	接聽器執行緒在等待 Web 服務要求到達時，在主機或埠上暫停執行的時間 (毫秒)。如果它在此期限內收到 Web 服務要求，接聽器將處理之。否則接聽器執行緒會檢查是否設定了連接器關閉旗標。如果已設定，連接器將終止。否則將依照 RequestWaitTimeout 的期限，繼續暫停執行。如果這個內容設定為 0，其將永遠暫停執行。若未指定，其預設為 60000ms。	60000 (ms)	否
++++HTTPReadTimeout	在從用戶端讀取 Web 服務要求時，將暫停執行接聽器的時間 (毫秒)。若此參數設定為 0，接聽器就會無限地暫停執行，直到它收到整個要求訊息。	0	否

表 40. SOAP/HTTP 及 SOAP/HTTPS 通訊協定特定接聽器的配置內容 (繼續)

名稱	可能值	預設值	必要的
++++HttpAsyncResponseCode	對接聽器提出非同步要求時的 HTTP 回應程式碼： 200 (OK) 202 (ACCEPTED)	2 0 2 (ACCEPTED)	否
++++URLsConfiguration	此為階層式內容，沒有值。其包含一或數個此接聽器所支援的 URL 配置，亦可能會有 MIME 類型及 charset 值 (選用性)。請注意，此為 ProtocolListenerFramework->ProtocolListeners->SOAPHTTPListener1 階層式內容的子內容。若未指定這個內容，接聽器會採用預設值。	ContextPath : / 已啟用 : true 資料處理常式 MimeType : 等於要求的 ContentType Charset : NONE。 如需進一步資訊，請參閱第 56 頁的『SOAP/HTTP 及 SOAP/HTTPS 通訊協定接聽器處理程序』。	否
+++++URL1	此為階層式內容，沒有值。其子項提供此接聽器所支援的 URL 名稱。可有多重受支援的 URL。請注意，您可以複製這個內容及其階層，以插入額外的 URL。		否
+++++ContextPath	接聽器所接收之 HTTP 要求的 URI。在 URLsConfiguration 內容的 ContextPath 值中，此值必須是唯一的。否則連接器將記載錯誤，且無法啟動。ContextPath 需區分大小寫；然而其可包含通訊協定、主機名稱及埠，不區分大小寫。若於 ContextPath 中指定通訊協定，則應為 http。若指定了主機，則應與 Host 接聽器內容的值相同。若已指定埠，則應與 Port 接聽器內容的值相同。		否
+++++Enabled	這個內容的值可決定是否要為連接器啟用母項 URL 階層式內容。	True	否
+++++TransformationRules	此為階層式內容，沒有值。它可保留一或多個轉換規則。		
+++++++TransformationRule1	此為階層式內容，沒有值。它可保留轉換規則。		否
+++++++ContentType	這個內容的值對應套用特殊處理方式 (資料處理常式 MIME 類型或 charset) 的送入要求，指定 ContentType。若 TransformationRuleN 階層式內容未指定 ContentType，連接器會記載警告訊息，並忽略 TransformationRuleN 內容。 為這個內容指定特殊的值 */*，可讓通訊協定接聽器將此規則套用於任何 ContentType。請注意，如果接聽器發現針對共用 ContentType 之同一個環境定義路徑有多個規則，接聽器會記載錯誤，且無法起始設定。		否
+++++++MimeType	當呼叫資料處理常式以處理所指定之 ContentType 的要求時，所使用的 MIME 類型。		否
+++++++Charset	當所指定之 ContentType 的要求轉換成商業物件時，所要使用的 Charset。		否

圖 34 顯示如「連接器配置程式」中所顯示的內容。

Standard Properties		Connector-Specific Properties	Supported Business Objects	
	Property	Value	Encrypt	Update Method
1	ConnectorType	WebService	<input type="checkbox"/>	agent restart
2	DataHandlerMetaObjectName	MO_DataHandler_Default	<input type="checkbox"/>	agent restart
3	⊞ JNDI		<input type="checkbox"/>	agent restart
4	⊞ ProtocolHandlerFramework		<input type="checkbox"/>	agent restart
5	⊞ ProtocolListenerFramework		<input type="checkbox"/>	agent restart
6	WorkerThreadCount	10	<input type="checkbox"/>	agent restart
7	RequestPoolSize	20	<input type="checkbox"/>	agent restart
8	⊞ ProtocolListeners		<input type="checkbox"/>	agent restart
9	⊞ SOAPHTTPListener1		<input type="checkbox"/>	agent restart
10	Protocol	soap/http	<input type="checkbox"/>	agent restart
11	SOAPDHMimeType	xml/soap	<input type="checkbox"/>	agent restart
12	Host	localhost	<input type="checkbox"/>	agent restart
13	Port	8080	<input type="checkbox"/>	agent restart
14	SocketQueueLength	5	<input type="checkbox"/>	agent restart
15	HTTPReadTimeout	0	<input type="checkbox"/>	agent restart
16	RequestWaitTimeout	60000	<input type="checkbox"/>	agent restart
17	BOPrefix		<input type="checkbox"/>	agent restart
18	⊞ URLsConfiguration		<input type="checkbox"/>	agent restart
19	⊞ URL1		<input type="checkbox"/>	agent restart
20	ContextPath	/	<input type="checkbox"/>	agent restart
21	Enabled	True	<input type="checkbox"/>	agent restart
22	⊞ TransformationRules		<input type="checkbox"/>	agent restart
23	⊞ TransformationRule1		<input type="checkbox"/>	agent restart
24	ContentType	*/*	<input type="checkbox"/>	agent restart
25	MimeType	xml/soap	<input type="checkbox"/>	agent restart
26	Charset	UTF8	<input type="checkbox"/>	agent restart
27	⊞ SOAPHTTPListener1		<input type="checkbox"/>	agent restart
28	⊞ SOAPJMSListener1		<input type="checkbox"/>	agent restart
29	⊞ ProxyServer		<input type="checkbox"/>	agent restart
30	⊞ SSL		<input type="checkbox"/>	agent restart
31	UseDefaults	true	<input type="checkbox"/>	agent restart

圖 34. SOAP/HTTP 通訊協定接聽器內容

表 41 彙總 SOAP/JMS 通訊協定特定接聽器的內容。+ 字元指出輸入項目在內容階層中的位置。

表 41. SOAP/JMS 通訊協定特定接聽器的配置內容

名稱	可能值	預設值	必要的
+++SOAPJMSListener1	JMS 通訊協定接聽器的專屬名稱。這是 <i>ProtocolListenerFramework</i> -> <i>ProtocolListeners</i> 階層式內容的子項。可以有重接聽器：您可以另外建立此內容的實例及其階層，以外掛其他 JMS 接聽器。		是
++++Protocol	soap/jms		是
++++SOAPDHMimeType	xml/soap	xml/soap	否
++++BOPrefix	這個內容的值會傳遞至 <i>SOAPDHMimeType</i> 內容所指定的資料處理常式。		否

表 41. SOAP/JMS 通訊協定特定接聽器的配置內容 (繼續)

名稱	可能值	預設值	必要的
++++RequestWaitTimeout	這個內容設定 SOAP/JMS 接聽器執行緒於等待 Web 服務要求時，暫停執行 InputQueue 的時間長度。如果它於此期限內收到 Web 服務要求，接聽器將處理之。若於此期間內未收到要求，接聽器執行緒會先檢查是否設定了連接器關閉旗標。若設定了連接器關閉旗標，連接器將終止。否則將依照 RequestWaitTimeout 的期限，繼續暫停執行。如果這個內容設定為 0，將無限期地暫停執行。	60000 毫秒	否
++++SessionPoolSize	可配置給指定接聽器及其工作程式執行緒的階段作業數上限。階段作業數下限 (及預設值) 為 2。對於較大的階段作業儲存池，連接器會需要較多的記憶體。	2	否
++++InputQueue	這個內容可指定接聽器在輪詢來自 Web 服務的入埠訊息時，所要登記的輸入佇列名稱。如果 LookupQueuesUsingJNDI = true，接聽器會使用 JNDI 來查閱此佇列，並將 InputQueue 內容的值設定為 SOAP/JMS 連結之 jms:address 元素的 jndiDestinationName 屬性。jms:address 元素指定於 WSDL 文件的 wsdl:port 區段中。如果您於產生 WSDL 期間選取 SOAP/JMS 接聽器，「系統管理程式」會自動使用這個內容的值來建立 jndiDestinationName 屬性。如果 LookupQueueUsingJNDI = false，則「系統管理程式」會建立 jmsProviderDestinationName 屬性。		是
++++InProgressQueue	這個內容提供進行中佇列的名稱。接聽器會將入埠訊息的複本由 InputQueue 傳送到 InProgressQueue。如果 LookupQueuesUsingJNDI = true，接聽器使用 JNDI 來查閱此佇列。		是
++++ArchiveQueue	這個內容提供保存佇列的名稱。接聽器將處理成功的訊息複本由 InProgressQueue 傳送到 ArchiveQueue。如果 LookupQueuesUsingJNDI = true，接聽器使用 JNDI 來查閱此佇列。		否
++++UnsubscribedQueue	這個內容提供未訂閱之佇列的名稱。接聽器將未訂閱之訊息的複本由 InProgressQueue 傳送到 UnsubscribedQueue。如果 LookupQueueUsingJNDI = true，接聽器使用 JNDI 來查閱此佇列。		否
++++ErrorQueue	這個內容提供錯誤佇列的名稱。接聽器將失敗訊息的複本傳送到 ErrorQueue。如果 LookupQueueUsingJNDI = true，接聽器使用 JNDI 來查閱此佇列。		否
++++InDoubtEvents	這個內容指定如何處理於 InProgressQueue 中，因連接器意外終止而尚未完全處理好的訊息。其可採用下列其中一個值： <ul style="list-style-type: none"> <li>• FailOnStartup 記載錯誤，並立即關閉</li> <li>• Reprocess 處理 InProgressQueue 中剩餘的訊息</li> <li>• Ignore 不處理進行中佇列中的任何訊息</li> <li>• LogError 記載錯誤，但不關閉</li> </ul>	Ignore	否



表 41. SOAP/JMS 通訊協定特定接聽器的配置內容 (繼續)

名稱	可能值	預設值	必要的
++++ReplyToQueue	<p>這個內容提供 <i>ReplyTo</i> 佇列的名稱。「WSDL 配置精靈」讀取這個內容，並將它寫入 WSDL 文件。若未指定這個內容，公用程式便不會於 WSDL 文件的 SOAP/JMS 連結中建立 <i>ReplyTo</i> JMS 標頭。(接聽器不使用這個內容。) 若指定了 JNDI 內容，且 <i>LookupQueueUsingJNDI = false</i>，「WSDL 產生公用程式」仍會於 WSDL 文件中建立 JNDI 特定屬性。請注意，這些 JNDI 特定屬性是必要的，因為 SOAP/JMS 連結並未提供任何方法，可在沒有 JNDI 的情況下，指定 <i>ReplyTo</i> 屬性。雖然不需要 <i>InputQueue</i> 的 JNDI 查閱作業，但 <i>ReplyTo</i> 佇列必須要有特定 JNDI 內容。若 WSDL 公用程式未找到特定 JNDI 內容，公用程式便無法於 SOAP/JMS 連結中建立 <i>ReplyTo</i> 屬性。</p>		
++++ JMSVendorURI	<p>特別用來識別 JMS 實作方式的字串，其對應於 SOAP/JMS 連結之 <i>jms:address</i> 元素的 <i>jmsVendorURI</i> 屬性。<i>jms:address</i> 元素指定於 WSDL 文件的 <i>wsdl:port</i> 區段中。接聽器不使用這個內容。</p>		否

圖 35 顯示如「連接器配置程式」中所顯示的內容。

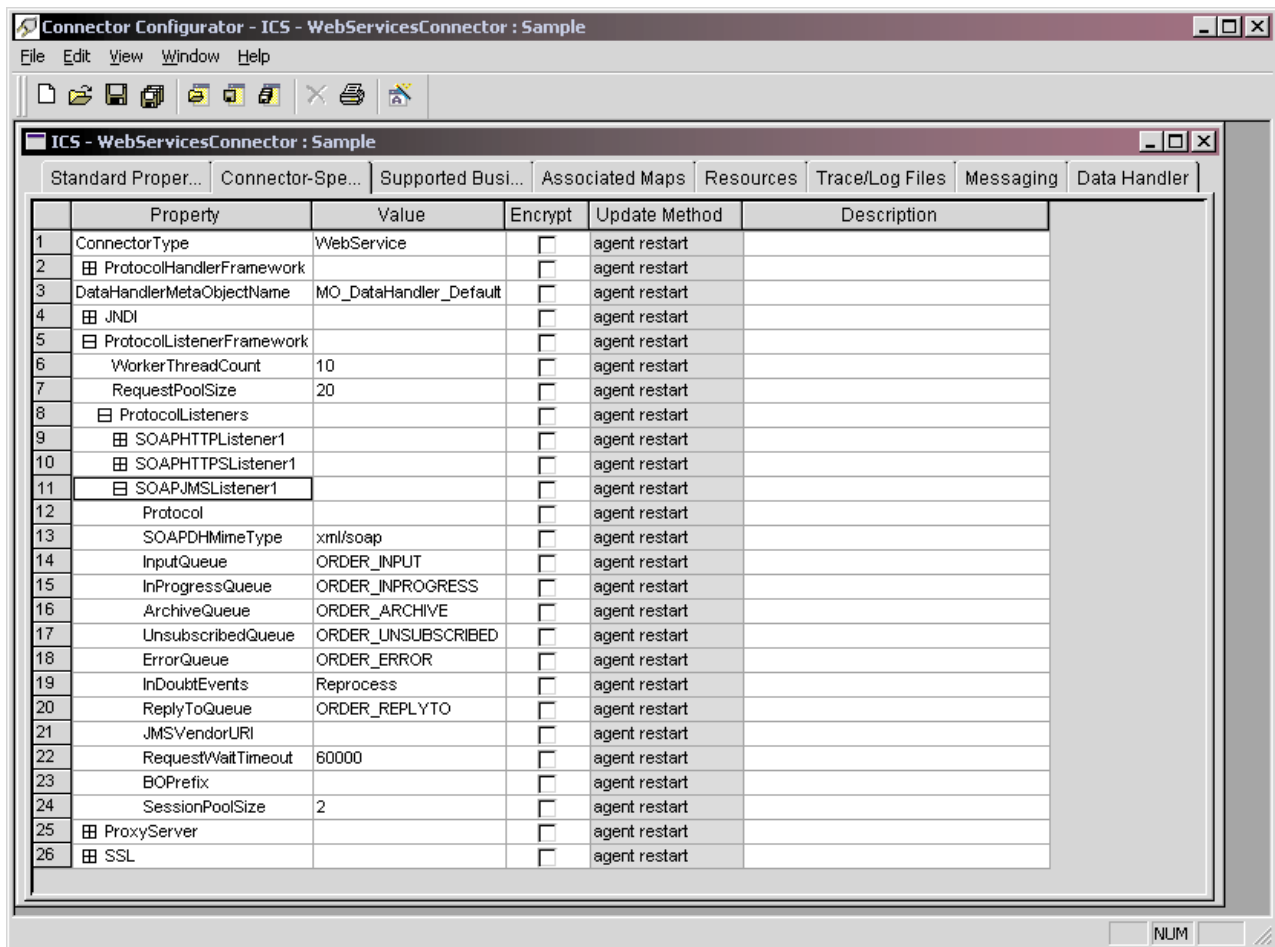


圖 35. SOAP/JMS 通訊協定接聽器內容

註: 確定指定於下列內容中的佇列名稱都是唯一的：

- InputQueue
- InProgressQueue
- ArchiveQueue
- UnsubscribedQueue
- ErrorQueue

**ProxyServer:** 當網路使用 Proxy 伺服器時，於這個內容之下配置這些值。此為階層式內容，沒有值。SOAP/HTTP/HTTPS 通訊協定處理常式將使用指定於這個內容下的值。

圖 36 顯示如「連接器配置程式」所顯示及下面所討論的 ProxyServer 內容。

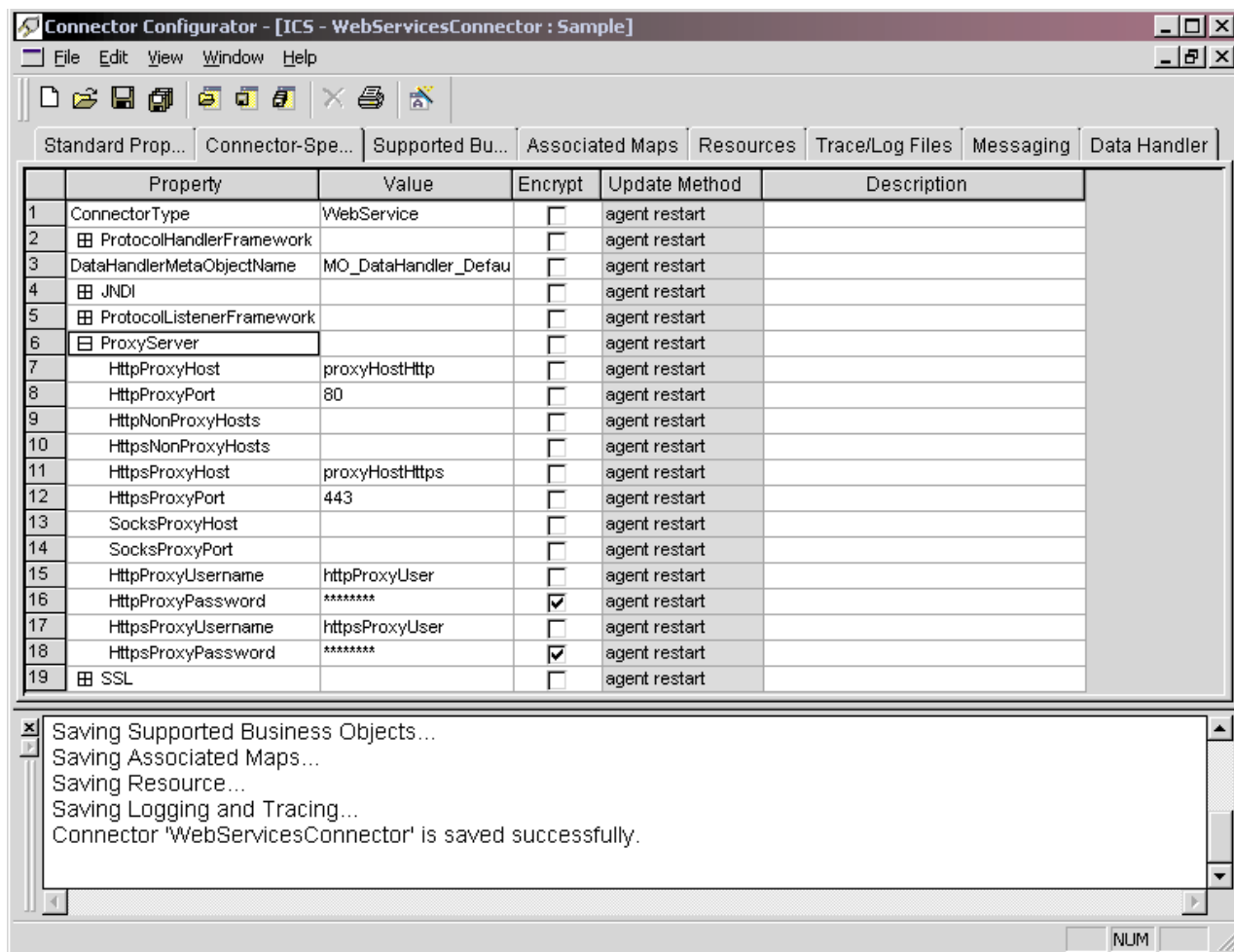


圖 36. ProxyServer 內容

**HttpProxyHost:** HTTP Proxy 伺服器的主機名稱。若網路為 HTTP 通訊協定使用 Proxy 伺服器，請指定這個內容。

預設值 = 無

**HttpProxyPort:** 連接器用來連接 HTTP Proxy 伺服器的埠號。

預設值 = 80

**HttpNonProxyHosts:** 這個內容的值提供一或數個主機 (適用於 HTTP)，這些主機必須是直接連線，而非透過 Proxy 伺服器。該值可為一個主機清單，用 “|” 將每個主機隔開。

預設值 = 無

**HttpsProxyHost:** HTTPS Proxy 伺服器的主機名稱。

預設值 = 無

**HttpsProxyPort:** 連接器用來連接 HTTPS Proxy 伺服器的埠號。

預設值 = 443

**HttpsNonProxyHosts:** 這個內容的值提供一或數個主機 (適用於 HTTPS)，這些主機必須是直接連線，而非透過 Proxy 伺服器。該值可為一個主機清單，用 "|" 將每個主機隔開。

預設值 = 無

**SocksProxyHost:** Socks Proxy 伺服器的主機名稱。當網路使用 Socks Proxy 時，請指定這個內容。

註: 基礎 JDK 必須支援 socks。

預設值 = 無

**SocksProxyPort:** 用來連接 Socks Proxy 伺服器的埠號。當網路使用 Socks Proxy 時，請指定這個內容。

預設值 = 無

**HttpProxyUsername:** HTTP Proxy 伺服器的使用者名稱。若 Web 服務要求的目的地為 HTTP URL，且您指定了 ProxyServer ->HttpProxyUsername，當使用 Proxy 鑑別時，SOAP HTTP/HTTPS 通訊協定處理常式會建立 Proxy-Authorization 標頭。處理常式使用 CONNECT 方法來進行鑑別。

proxy-authentication 標頭是以 base64 來編碼的，其結構如下：

Proxy-Authorization: Basic

Base64EncodedString

處理常式連結使用者名稱及密碼內容值 (以冒號 (;) 隔開)，以建立 base64 編碼的字串。

預設值 = 無

**HttpProxyPassword:** HTTP Proxy 伺服器的密碼。如需此值用法的相關資訊，請參閱『HttpProxyUsername』。

預設值 = 無

**HttpsProxyUsername:** HTTPS Proxy 伺服器的使用者名稱。如果 Web 服務要求的目的地為 HTTPS URL，且您指定了 ProxyServer ->HttpsProxyUsername，SOAP HTTP/HTTPS 通訊協定處理常式會建立 Proxy-Authorization 標頭，以 Proxy 鑑別。處理常式連結 HttpsProxyUsername 及 HttpsProxyPassword 配置內容值 (以冒號 (;) 隔開)，建立 base64 編碼的字串。

預設值 = 無

**HttpsProxyPassword:** HTTPS Proxy 伺服器的密碼。如需此值用法的相關資訊，請參閱『HttpsProxyUsername』。

預設值 = 無

**SSL:** 指定這個內容下的值，為連接器配置 SSL。此為階層式內容，沒有值。

圖 37 顯示如「連接器配置程式」所顯示及下面所討論的 SSL 內容。

Standard Properties		Application Config Properties		Supported Business Objects		Trace/Log Files		D:	
	Property	Value	Update Method	Encrypt	Description				
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>					
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>					
3	<input type="checkbox"/> ProxyServer		agent restart	<input type="checkbox"/>					
4	<input type="checkbox"/> SSL		agent restart	<input type="checkbox"/>					
5	SSLVersion	SSL	agent restart	<input type="checkbox"/>					
6	SSLDebug	False	agent restart	<input type="checkbox"/>					
7	KeyStoreType	JKS	agent restart	<input type="checkbox"/>					
8	KeyStore		agent restart	<input type="checkbox"/>					
9	KeyStorePassword		agent restart	<input type="checkbox"/>					
10	KeyStoreAlias		agent restart	<input type="checkbox"/>					
11	TrustStore		agent restart	<input type="checkbox"/>					
12	TrustStorePassword		agent restart	<input type="checkbox"/>					
13	UseClientAuth	False	agent restart	<input type="checkbox"/>					
14	<input type="checkbox"/> ProtocolListenerFramework		agent restart	<input type="checkbox"/>					
15	<input type="checkbox"/> ProtocolHandlerFramework		agent restart	<input type="checkbox"/>					
16	<input type="checkbox"/> JNDI		agent restart	<input type="checkbox"/>					
17	<input type="checkbox"/> WSCollaborations		agent restart	<input type="checkbox"/>					

圖 37. SSL 內容

**SSLVersion:** 連接器所使用的 SSL 版本。如需進一步資訊，請參閱 IBM JSSE 文件，查詢受支援的 SSL 版本。

預設值 = SSL

**SSLDebug:** 若這個內容的值設定為 true，連接器就會將 javax.net.debug 系統內容的值設定為 true。IBM JSSE 使用這個內容來開啓追蹤機能。如需進一步資訊，請參閱 IBM JSSE 文件。

預設值 = false

**KeyStoreType:** 這個內容的值提供 KeyStore 及 TrustStore 的類型。如需進一步資訊，請參閱 IBM JSSE 文件，查詢有效的金鑰儲存庫類型。

預設值 = JKS

**KeyStore:** 這個內容提供金鑰儲存庫的完整路徑。若未指定 KeyStore 及/或 KeyStoreAlias 內容，就會忽略 KeyStorePassword、KeyStoreAlias、TrustStore、TrustStorePassword 內容。若連接器無法使用這個內容中所指定的路徑來載入金鑰儲存庫，即無法啓動。該路徑必須是金鑰儲存庫檔的完整路徑。

預設值 = 無

**KeyStorePassword:** 這個內容提供 Keystore 中之私密金鑰的密碼。

預設值 = 無

**KeyStoreAlias:** 這個內容提供 KeyStore 中之金鑰組的別名。SOAP/HTTPS 接聽器使用 KeyStore 中的這個私密金鑰。此外，SOAP/HTTP-HTTPS 通訊協定處理常式於呼叫需要用戶端鑑別的 Web 服務時，會使用 KeyStore 中的這個別名。該內容必須設為有效的 JSSE 別名。

預設值 = 無

**TrustStore:** 這個內容提供 TrustStore 的完整路徑。TrustStore 可用來儲存連接器所信任的憑證。TrustStore 的類型必須與 KeyStore 相同。您必須指定 TrustStore 檔案的完整路徑。

預設值 = 無

**TrustStorePassword:** 這個內容提供 Truststore 的密碼。

預設值 = 無

**UseClientAuth:** 這個內容指定是否要使用 SSL 用戶端鑑別功能。若設定為 true，SOAP/HTTPS 接聽器使用用戶端鑑別功能。

預設值 = false

**WSCollaborations:** 當您將協同作業物件顯現為 Web 服務，並用於非 TLO 時，便會自動建立這個內容。此為階層式內容，沒有值。這個內容的每個第一層子項代表顯現為 Web 服務的協同作業。如需用來自動建立這些內容之工具的資訊，請參閱第 131 頁的第 7 章，『將協同作業顯現為 Web 服務』。

**註:** 若您於「系統管理程式」中刪除協同作業或其埠，連接器並不會自動刪除代表該協同作業的內容。您必須使用「連接器配置程式」來刪除這些內容。

圖 38 顯示如「連接器配置程式」所顯示及下面所討論的 WSCollaborations 內容。

	Standard Properties	Application Config Properties	Supported Business Objects	Trace/Log Files	D
	Property	Value	Update	Encrypt	Description
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>	
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>	
3	▣ ProxyServer		agent restart	<input type="checkbox"/>	
4	▣ SSL		agent restart	<input type="checkbox"/>	
5	▣ ProtocolListenerFramework		agent restart	<input type="checkbox"/>	
6	▣ ProtocolHandlerFramework		agent restart	<input type="checkbox"/>	
7	▣ JNDI		agent restart	<input type="checkbox"/>	
8	▣ WSCollaborations		agent restart	<input type="checkbox"/>	
9	▣ Collaboration1		agent restart	<input type="checkbox"/>	
10	▣ CollaborationPort1		agent restart	<input type="checkbox"/>	
11	▣ WebServiceOperation1		agent restart	<input type="checkbox"/>	
12	BodyName		agent restart	<input type="checkbox"/>	
13	BodyNS		agent restart	<input type="checkbox"/>	
14	BOName		agent restart	<input type="checkbox"/>	
15	BOVerb		agent restart	<input type="checkbox"/>	
16	Synchronous		agent restart	<input type="checkbox"/>	

圖 38. WSCollaborations 內容

**Collaboration1:** 這個內容會透過此連接器為顯現成 Web 服務的協同作業物件命名。此為階層式內容，沒有值。這樣的內容可以有好幾個，每個顯現為 Web 服務的協同作業物件各一個內容。這個內容的每個第一層子項代表此協同作業物件的埠。

**CollaborationPort1:** 這個內容是要為協同作業埠命名。此為階層式內容，沒有值。這樣的內容可以有好幾個，在此協同作業中，每個連結到連接器的埠各一個內容。這個內容的每個第一層子項代表 Web 服務作業。

**WebServiceOperation1:** 這個內容代表協同作業物件的 Web 服務作業。此為階層式內容，沒有值。可有一或數個這樣的內容，針對在產生 WSDL 文件時，使用者所定義的每個 Web 服務作業，各設定一個內容。

**BodyName:** 這個內容提供 Web 服務方法的名稱，且必須是有效的 XML 元素名稱。

預設值 = 無

**BodyNS:** 這個內容提供 Web 服務方法的名稱空間，且必須是有效的 XML 名稱空間。

預設值 = 無

**BOName:** 這個內容提供此作業的「要求」商業物件名稱。

預設值 = 無

**Mode:** 這個內容指定作業的處理模式。若設定為 `synch`，連接器便會同步呼叫協同作業。否則，依預設，連接器會以非同步的方式，將協同作業當作只有要求的作業來呼叫。

預設值 = `asynch`

**JNDI:** 連接器維護了一組 JNDI (Java 命名和目錄介面) 提供者內容，當 SOAP/JMS 通訊協定處理常式及 JMS 通訊協定接聽器要連接至 JNDI 時，會使用這些內容。此為階層式內容，沒有值。連接器使用 JNDI 來查閱 JMS Connection Factory 物件。請注意，「WSDL 配置精靈」在產生 SOAP/JMS 連結時，會使用這個內容。

圖 39 顯示如「連接器配置程式」所顯示及下面所討論的 JNDI 內容。

Standard Properties		Application Config Properties	Supported Business Objects	Trace/Log Files	D:
	Property	Value	Update	Encrypt	Description
1	DataHandlerMetaObjectName	MO_DataHandler_Default	agent restart	<input type="checkbox"/>	
2	ConnectorType	WebService	agent restart	<input type="checkbox"/>	
3	<input type="checkbox"/> ProxyServer		agent restart	<input type="checkbox"/>	
4	<input type="checkbox"/> SSL		agent restart	<input type="checkbox"/>	
5	<input type="checkbox"/> ProtocolListenerFramework		agent restart	<input type="checkbox"/>	
6	<input type="checkbox"/> ProtocolHandlerFramework		agent restart	<input type="checkbox"/>	
7	<input type="checkbox"/> JNDI		agent restart	<input type="checkbox"/>	
8	LookupQueuesUsingJNDI	False	agent restart	<input type="checkbox"/>	
9	InitialContextFactory		agent restart	<input type="checkbox"/>	
10	JNDIConnectionFactoryName		agent restart	<input type="checkbox"/>	
11	CTX_ObjectFactories		agent restart	<input type="checkbox"/>	
12	CTX_StateFactories		agent restart	<input type="checkbox"/>	
13	CTX_URLPackagePrefixes		agent restart	<input type="checkbox"/>	
14	CTX_DNS_URL		agent restart	<input type="checkbox"/>	
15	CTX_Authoritative		agent restart	<input type="checkbox"/>	
16	CTX_Batchsize		agent restart	<input type="checkbox"/>	
17	CTX_Referral		agent restart	<input type="checkbox"/>	
18	CTX_SecurityProtocol		agent restart	<input type="checkbox"/>	
19	CTX_SecurityAuthentication		agent restart	<input type="checkbox"/>	
20	CTX_SecurityPrincipal		agent restart	<input type="checkbox"/>	
21	CTX_SecurityCredentials		agent restart	<input type="checkbox"/>	
22	CTX_Language		agent restart	<input type="checkbox"/>	
23	<input type="checkbox"/> WSCollaborations		agent restart	<input type="checkbox"/>	

圖 39. JNDI 內容

**LookupQueuesUsingJNDI:** 若這個內容的值設定為 true，連接器的 SOAP/JMS 接聽器及 SOAP/JMS 通訊協定處理常式將會使用 JNDI 來查閱佇列。

預設值 = false

**JNDIProviderURL:** 這個內容提供 JNDI 服務提供者的 URL，其對應於 SOAP/JMS 連結之 jms:address 元素的 jndiProviderURL 屬性。 jms:address 元素指定於 wsdl:port 區段中。其用來作為預設的 JNDI 提供者，且必須是有效的 JNDI URL。如需進一步資訊，請參閱 JNDI 規格。

預設值 = 無

**InitialContextFactory:** 這個內容提供 Factory 類別的完整類別名稱（例如，com.ibm.NamingFactory），以建立起始環境定義。請注意，此對應於 SOAP/JMS 連結之 jms:address 元素的 initialContextFactory 屬性。 jms:address 元素指定於 wsdl:port 區段中。

預設值 = 無

**JNDIConnectionFactoryName:** 這個內容提供 Connection Factory 的名稱，使用 JNDI 環境定義來查閱。請注意，此對應於 SOAP/JMS 連結之 jms:address 元素的 jndiConnectionFactoryName 屬性。 jms:address 元素指定於 wsdl:port 區段中。

預設值 = 無



**CTX\_ObjectFactories:** 內容指定有關 JNDI 環境定義中安全性及物件查閱的其他資訊。表 42 彙總這些內容。+ 字元指出輸入項目在內容階層中的位置。

表 42. 「Java 命名和目錄介面 (JNDI)」提供者內容

內容名稱	說明
+CTX_StateFactories	內容指定有關 JNDI 環境定義中安全性及物件查閱的其他資訊。如需詳細資訊，請參閱 J2EE 文件。這些內容反映了「配接器」用於 JMS 的那些內容。
+CTX_URLPackagePrefixes	
+CTX_DNS_URL	
+CTX_Authoritative	
+CTX_Batchsize	
+CTX_Referral	
+CTX_SecurityProtocol	
+CTX_SecurityAuthentication	
+CTX_SecurityPrincipal	
+CTX_SecurityCredentials	
+CTX_Language	

## 建立多重通訊協定接聽器

您可建立多重通訊協定接聽器的實例。通訊協定接聽器被配置為 ProtocolListenerFramework -> ProtocolListeners 連接器內容的子內容。(ProtocolListenerFramework -> ProtocolListeners 的) 每項子內容皆可為連接器識別不同的通訊協定接聽器。因此，您可於 ProtocolListeners 內容之下配置新的子內容，以建立其他通訊協定接聽器。請確定您為新建立的接聽器內容，指定所有的子內容。每個接聽器的名稱皆必須是唯一的。然而，請勿變更接聽器通訊協定內容 (soap/http、soap/https 或 soap/jms)，對於接聽器的多重實例，這些內容仍維持原狀。

**註：**「通訊協定」內容是非常重要的，因為其作為切換選項來使用。若您不想使用接聽器或處理常式，則將這個內容留白。

如果您要建立 SOAP/HTTP 或 SOAP/HTTPS 接聽器的多重實例，請務必要為每個實例指定不同的「埠」及「主機」內容。若您正指定多個 SOAP/JMS 接聽器，請務必為每個實例使用不同的佇列組合。

您無法建立處理常式的多重實例。每個通訊協定只能有一個處理常式。

---

## 連接器啟動時

當您啟動連接器時，init() 方法會讀取以「系統管理程式」之「連接器配置程式」所設定的配置內容。為了能適當地發揮作用，請勿停用連接器輪詢功能 (依預設，會啟用連接器輪詢功能)。下列各節將說明其影響。

## Proxy 設定

如果您指定 ProxyServer 連接器特有的內容，連接器會設定 Proxy 系統內容。Proxy 伺服器與 SOAP/HTTP-HTTPS 通訊協定處理常式搭配使用，只能處理要求。連接器亦會追蹤其所設定的每項系統內容。如需 ProxyServer 內容的相關資訊，請參閱第 76 頁的『連接器特有的配置內容』。

## JNDI 起始設定

連接器特有的內容 JNDI 指定連接器所要使用的 JNDI。連接器使用 JNDI 來查閱 JMS Connection Factory 物件。如果 JNDI ” LookupQueuesUsingJNDI 的設定為 true，連接器使用 JNDI 來查閱 JMS 佇列物件。

如果您不想使用 SOAP/JMS (SOAP/JMS 通訊協定接聽器及 SOAP/JMS 通訊協定處理常式)，則不需指定 JNDI 內容。如果您指定 JNDI 內容，但連接器無法起始設定 JNDI，連接器就會終止。請注意，必須指定下列所有的連接器特有 JNDI 內容，連接器才可起始設定 JNDI：

- JNDIProviderURL
- InitialContextFactory
- JNDIConnectionFactoryName

註：連接器並未提供 JNDI 實作方式

## 通訊協定接聽器組織架構起始設定

於啟動期間，連接器將通訊協定接聽器組織架構實例化，並加以起始設定。此組織架構讀取連接器特有的內容 ProtocolListenerFramework，然後連接器會讀取 WorkerThreads 及 RequestPoolSize 連接器內容的值。若未指定 ProtocolListenerFramework 內容，或是內容遺失，連接器將無法從 Web 服務用戶端接收要求，並記載警告。

連接器接著讀取 ProtocolListenerFramework -> ProtocolListeners 內容。ProtocolListeners 內容的所有第一層內容代表通訊協定接聽器。通訊協定接聽器組織架構嘗試載入並起始設定每個接聽器，並加以追蹤。如果接聽器具有持續性事件功能，就會嘗試進行事件回復作業。

## 通訊協定處理常式組織架構起始設定

連接器會讀取連接器特有的內容 ProtocolHandlerFramework，並實例化及起始設定通訊協定處理常式組織架構。如果這個內容遺失或設定不正確，連接器即無法執行要求處理程序，而且會記載警告。接著連接器會讀取所有的 ProtocolHandlerFramework ” ProtocolHandlers 內容 (對應於通訊協定處理常式)，並嘗試載入、起始設定及追蹤。請注意，通訊協定處理常式是在連接器起始設定期間載入的，並不會在協同作業提出服務要求時起始設定。通訊協定處理常式在多緒的情況下是安全的。

---

## 記載

連接器將於下列情形記載警告：

- 未指定 ProtocolListenerFramework 內容。連接器會警告您，它無法執行事件通知。(連接器無法呼叫顯現為 Web 服務的協同作業。)
- 未指定 ProtocolHandlerFramework 內容。連接器會警告您，它無法執行 (協同作業) 要求處理程序。

---

## 追蹤

追蹤是一種選用性的除錯功能，您可啓用此功能以密切注意連接器的行為。依預設，追蹤訊息會寫至 STDOUT。如需配置追蹤訊息的相關資訊，請參閱連接器配置內容。如需追蹤的詳細資訊 (包括如何啓用和設定它)，請參閱 *Connector Development Guide for Java*。

連接器追蹤層次如下：

- 層次 0 此層次可用來追蹤識別連接器版本的訊息。
- 層次 1 每次追蹤 `pollForEvents` 方法是否被呼叫。追蹤接聽器建立並遞送至 ICS 的 TLO 名稱。追蹤「要求」商業物件名稱，及於 TLO 中的對應屬性名稱。
- 層次 2 對於每一次商業物件不論來自 `gotApplEvent()` 或 `executeCollaboration()` 公佈至 InterChange Server 時就記載的追蹤訊息，請使用這個層次。此外，亦可追蹤是哪個通訊協定處理常式在處理要求。
- 層次 3 追蹤處理中之商業物件的 ASI。追蹤處理中之商業物件的屬性。追蹤事件通知期間，「SOAP 要求」商業物件的 TLO。追蹤資料處理常式傳回的商業物件。
- 層次 4 與下列相關的追蹤及傳輸標頭：
- 通訊協定接聽器從傳輸埠擷取的 SOAP 要求訊息
  - 由通訊協定接聽器傳送給用戶端的回應訊息。
- 追蹤大量產生執行緒的過程、已處理的所有 ASI，及重要功能的所有輸入項目和結束。
- 層次 5 追蹤下列各項：
- 每個重要方法的輸入項目及結束
  - 所有的特定配置內容
  - 每個通訊協定接聽器的負荷量
  - 通訊協定接聽器從傳輸埠擷取的要求訊息
  - 通訊協定接聽器在傳輸埠上傳送到用戶端的回應訊息
  - 每個通訊協定處理常式的負荷量
  - SOAP 資料處理常式所傳回的訊息
  - 傳送至協同作業的 TLO 商業物件傾出
  - 資料處理常式所傳回的商業物件傾出。



---

## 第 5 章 SOAP 資料處理常式

- 『配置 SOAP 資料處理常式』
- 第 103 頁的『SOAP 資料處理常式處理程序』
- 第 127 頁的『SOAP style 及 use 準則』
- 第 128 頁的『XML 限制』

SOAP 資料處理常式為一種資料轉換模組，其主要角色是將商業物件轉換為 SOAP 訊息，及將 SOAP 訊息轉換成商業物件。 SOAP 資料處理常式可執行下列功能：

- 要求處理程序
  - 將 SOAP 要求商業物件轉換成 SOAP 要求訊息
  - 將 SOAP 回應訊息轉換成 SOAP 回應商業物件
  - 將 SOAP 錯誤訊息轉換成 SOAP 錯誤商業物件
- 事件處理程序
  - 將 SOAP 要求訊息轉換成 SOAP 要求商業物件
  - 將 SOAP 回應商業物件轉換成 SOAP 回應訊息
  - 將 SOAP 錯誤商業物件轉換成 SOAP 錯誤訊息

本章說明如何配置 SOAP 資料處理常式、SOAP 資料處理常式如何處理訊息和物件，及如何自訂資料處理常式。

---

### 配置 SOAP 資料處理常式

SOAP 資料處理常式是 Web 服務連接器中的中樞元件。連接器會呼叫 SOAP 資料處理常式將商業物件轉換成與 Web 服務相容的 SOAP 訊息。

當協同作業是以 Web 服務來顯現時，連接器亦會呼叫 SOAP 資料處理常式。然後資料處理常式將從遠端交易友機 (或內部用戶端) 傳出的 SOAP 訊息，轉換成商業物件。連接器會將商業物件傳遞至已為 Web 服務配置的協同作業。

在這些轉換作業中，資料處理常式 meta 物件中的資訊扮演著非常重要的角色。於安裝產品檔案之後，開機之前，配置此資訊。除非您要新增自訂的名稱處理常式，否則您可使用預設的 SOAP 資料處理常式配置，以節省時間。但是您必須對每一個資料處理常式轉換配置特定的 meta 物件資訊。將於下面一節討論資料處理常式 meta 物件。

#### Meta 物件基本需求

Meta 物件為包含配置資訊的商業物件。連接器會在執行時期使用 Meta 物件，以配置資料處理常式，並建立其實例。 SOAP 資料處理常式亦使用 Meta 物件來尋找 SOAP 訊息的主體、判斷該主體所對應的商業物件及動詞、將 SOAP 訊息中的商業物件編碼，及執行本章中所討論的其他數項作業。本節說明這些 Meta 物件的基本需求。

#### Meta 物件階層及專有名詞

圖 28 顯示 Web 服務產品之配接器的 Meta 物件結構。在下面的圖例及討論中，將以粗體字來表示 Meta 物件名稱。

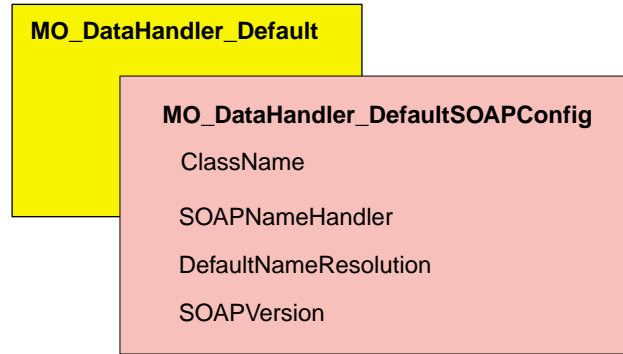


圖 40. Meta 物件結構

在這整份文件中，討論到 Meta 物件時，將會使用下列專有名詞：

- `MO_DataHandler_Default` 資料處理常式 Meta 物件，可供連接器代理程式用來決定要將哪個資料處理常式實例化。此值指定於連接器的 `DataHandlerMetaObjectName` 內容中。
- `MO_DataHandler_DefaultSOAPConfig` SOAP 資料處理常式專屬的子項資料處理常式 Meta 物件。
- SOAP Configuration Meta-Object (SOAP Config MO - SOAP 配置 Meta 物件) 被指定為 SOAP 商業物件子項的 Meta 物件，其包含單次由商業物件轉換成 SOAP 訊息的配置資訊，反之亦然。

### **MO\_DataHandler\_Default**

`MO_DataHandler_Default` 為連接器所呼叫之所有資料處理常式的最上層 Meta 物件。這些 Meta 物件中所包含的 MIME 類型，可決定要使用哪個資料處理常式。連接器代理程式會使用此 Meta 物件來建立 SOAP 資料處理常式的實例。因此，`MO_DataHandler_Default` 物件必須包含一個名為 `xml_soap` 的屬性，且其類型為 `MO_DataHandler_DefaultSOAPConfig`。

您可於安裝 `MO_DataHandler_Default` 物件之後，再加以配置。您必須新增 `MO_DataHandler_DefaultSOAPConfig` 類型的 `xml_soap`。

### **MO\_DataHandler\_DefaultSOAPConfig**

連接器代理程式會在執行時期，使用此 Meta 物件來建立及配置 SOAP 資料處理常式。`MO_DataHandler_DefaultSOAPConfig` 具有兩個 `string` 類型的屬性，其指定：

- SOAP 資料處理常式的類別名稱
- SOAP 名稱處理常式
- 自訂名稱處理常式失敗時的預設名稱解析
- SOAP 版本 (1.1 或 1.2)

這些屬性顯示於表 43 中。

除非您想要實行自訂的名稱處理常式 (本章稍後將會討論)，否則您可於遞送及安裝時，使用 `MO_DataHandler_DefaultSOAPConfig`。不需要配置。

表 43. *MO\_DataHandler\_DefaultSOAPConfig* 的 *Meta* 物件屬性

名稱	類型	預設值	說明
ClassName	字串	com.ibm.adapters .dataHandlers.xml.soap	資料處理常式基礎類別用來尋找類別名稱的標準屬性，以傳遞至 createHandler 方法的 MIME 類型為基礎。
SOAPName 處理常式	字串		所使用之 SOAP 名稱處理常式的名稱。
DefaultName 解析	字串	false	決定當自訂名稱處理常式失敗時，是否使用預設名稱解析。
SOAPVersion	字串	1.1	決定資料處理常式用來讀取及寫入 SOAP 訊息的 SOAP 標準 (1.1 或 1.2)。

### SOAP 配置 meta 物件：每個 SOAP 商業物件的子項

SOAP Config MO 可定義單一資料處理常式轉換的資料格式化行爲 -- 由 SOAP 訊息轉換成商業物件，或是由商業物件轉換成 SOAP 訊息。SOAP Config MO 是 SOAP 商業物件的子項。這些子項 SOAP Config MO 對預設商業物件解析來說是非常重要的。當使用預設商業物件解析時，所有的子項 SOAP Config MO (無論是要求、回應或錯誤物件) 都必須要有專屬的輸入項目，以作為 BodyName 及 BodyNS 的預設值。表 44 顯示這些屬性以及 SOAP Config MO 的其他屬性。

表 44. *SOAP Config MO* 的屬性

名稱	必要的	說明
BodyNS	是	要用於 SOAP 主體的名稱空間。
BodyName	是	SOAP 訊息主體的名稱。針對 SOAP 錯誤 (fault)，請將預設值設定為 soap: fault。
BOVerb	是	包含 SOAP Config MO 之商業物件的動詞。
TypeInfo	否	True 或 false 屬性，可指定要將哪種類型的資訊 (xsi:type) 寫入 SOAP 元素，及從該元素讀取哪種類型的資訊。預設值 = false
TypeCheck	否	如果 TypeInfo 的設定為 true，則此為唯讀內容。可能的值：none 及 strict。若為 none，則在將 SOAP 訊息讀取至此商業物件時，會略過類型驗證。若為 strict，資料處理常式就會比對商業物件的應用程式特有資訊，嚴格地驗證所有的 SOAP 類型名稱及名稱空間。預設值 = none
Style	否	這個內容可指定 SOAP 訊息樣式，並具有其他屬性的含意，例如：BodyName 及 BodyNS。此屬性的可能值為 rpc 及 document。預設值 = rpc
Use	否	這個內容指定 SOAP 訊息的用法，並會影響從商業物件建構 SOAP 主體的方式。可能的值為 literal 及 encoded。預設值是 literal。

圖 41 顯示 SOAP 商業物件與 SOAP Config MO 之間的關係。

## SOAP 商業物件

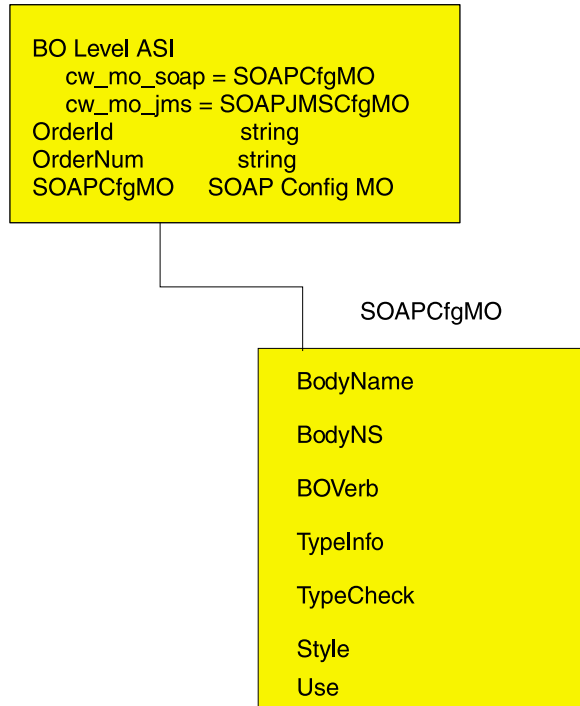


圖 41. SOAP 配置 Meta 物件

圖 41 顯示 SOAP 回應商業物件及其子項商業物件。子項商業物件 (SOAPCfgMO) 為 SOAP Config MO，可指定 SOAP 資料處理常式將商業物件回應轉換成 SOAP 回應訊息的方式。用來指示子項 SOAP Config MO 的屬性，必須使用以 `cw_mo_soap` 開頭的名稱值組。

依照慣例，當讀取以 `cw_mo_` 開頭的商業物件層次應用程式特有資訊時，資料處理常式會辨識出名稱值組所指定的子項物件中包含了轉換 Meta 物件資訊，因此並未包含此子項作為其所轉換之訊息的主體內容。於該範例中，名稱值組 `cw_mo_jms` 及 `cw_mo_soap` 所指定的子項物件被當作是 Meta 物件，不會寫入 SOAP 回應訊息。此外，只要是以 `cw_mo_` 開頭的所有的商業物件層次應用程式特有資訊，SOAP 資料處理常式都會忽略 (`cw_mo_soap` 除外)。因此，SOAP 資料處理常式會忽略應用程式特有資訊，例如：`cw_mo_tpi`。但是 SOAP 資料處理常式會讀取並使用 `cw_mo_soap` 中所指定的 SOAP Config MO 來執行 SOAP 回應轉換作業，由商業物件轉換成 SOAP 訊息。

所有的 SOAP 商業物件皆必須有子項 SOAP Config MO，且必須在商業物件層次，將這些子項 SOAP Config MO 指定為應用程式特有資訊。這項作業中有許多都是自動完成的：當您使用 WSDL ODA 來產生 SOAP 訊息的商業物件時，會自動為您產生 SOAP Config MO。

### Style 及 Use 對 SOAP 訊息的影響

SOAP Config MO 的選用性內容 `Style` 及 `Use` 會影響 SOAP 訊息的建立方式。`Style` 的可能值為：`rpc` 及 `document`，`Use` 的可能值為：`literal` 及 `encoded`。下面各節將討論 `Style` 及 `Use` 組合對 SOAP 訊息建立的影響。



**rpc/literal:** 若 Style 內容的設定為 rpc，Use 內容的設定為 literal，就會分別從 SOAP ConfigMO 的 BodyName 及 BodyNS 內容來讀取「SOAP 訊息」的「主體名稱」及「主體名稱空間」。

下列為 rpc/literal 樣式訊息的範例，其中「主體名稱」及「主體名稱空間」已分別解析為 getOrderStatus 及 OrderStatusNS：

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getOrderStatus xmlns:ns1="http://www.ibm.com/">
      <Part1>
        <Part1>
          <ns2:Elem1 xmlns:ns2="http://www.ibm.com/elem1">
            <Child1>1</Child1>
            <Child2>2</Child2>
          </ns2:Elem1>
          <ns3:Elem1 xmlns:ns3="http://www.ibm.com/elem1">
            <Child1>3</Child1>
            <Child2>4</Child2>
          </ns3:Elem1>
          <Elem2>10</Elem2>
        </Part1>
      </ns1:getOrderStatus>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

圖 42 顯示此 rpc/literal 訊息的對應商業物件。

Name	Type	Key	Card	Default	App Spec Info
Part1	SOAP_Part1Type	<input checked="" type="checkbox"/>	1		
Elem1	SOAP_MaxType	<input checked="" type="checkbox"/>	N		maxoccurs=5;elem_ns=http://www.ibm.com/elem1
Child1	String	<input checked="" type="checkbox"/>			
Child2	String	<input type="checkbox"/>			
ObjectEventId	String	<input type="checkbox"/>			
Elem2	String	<input type="checkbox"/>			
ObjectEventId	String	<input type="checkbox"/>			
SOAPConfigMO	SOAP_Req_Cfg_MO	<input type="checkbox"/>	1		
BodyName	String	<input checked="" type="checkbox"/>		getOrderStatus	
BodyNS	String	<input type="checkbox"/>		http://www.ibm.com	
BOVerb	String	<input type="checkbox"/>		Retrieve	
TypeInfo	String	<input type="checkbox"/>		false	
TypeCheck	String	<input type="checkbox"/>		none	
Style	String	<input type="checkbox"/>		rpc	
Use	String	<input type="checkbox"/>		literal	

圖 42. rpc/literal SOAP Config MO

**註:** 您必須正確地配置這些內容及商業物件屬性，如此才能建立對應的 SOAP 訊息。

**rpc/encoded:** 若 Style 內容的設定為 rpc，Use 內容的設定為 encoded，就會分別從 Child ConfigMO 的 BodyName 和 BodyNS 內容來讀取「SOAP 訊息」的「主體名稱」及「主體名稱空間」。此外，亦會將 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" 屬性新增至「主體」標籤。

下列為 rpc/encoded 訊息的範例，其中「主體名稱」及「主體名稱空間」已分別解析為 getOrderStatus 及 OrderStatusNS。

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body SOAP-ENV:encodingStyle=
    "http://schemas.xmlsoap.org/soap/encoding/">
    <ns1:getOrderStatus xmlns:ns1="http://www.ibm.com/">
      <Part1 xsi:type="ns1:SOAP_Part1Type">
        <ns2:Elem1 SOAP-ENC:arrayType="ns2:SOAP_MaxType[2]"
          xsi:type="SOAP-ENC:Array" xmlns:ns2="http://www.ibm.com/Elem1">
          <item>
            <Child1 xsi:type="xsd:string">1</Child1>
            <Child2 xsi:type="xsd:string">2</Child2>
          </item>
          <item>
            <Child1 xsi:type="xsd:string">3</Child1>
            <Child2 xsi:type="xsd:string">4</Child2>
          </item>
        </ns2:Elem1>
        <Elem2 xsi:type="xsd:string">10</Elem2>
      </Part1>
    </ns1:getOrderStatus>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

圖 43 顯示此 rpc/encoded 訊息的對應商業物件。

Name	Type	Key	Card	Default	App Spec Info
☐ Part1	SOAP_Part1Type	<input checked="" type="checkbox"/>	1		
☐ Elem1	SOAP_MaxType	<input checked="" type="checkbox"/>	N		elem_ns=http://www.ibm.com/Elem1
Child1	String	<input checked="" type="checkbox"/>			
Child2	String	<input type="checkbox"/>			
ObjectEventId	String				
Elem2	String	<input type="checkbox"/>			
ObjectEventId	String				
☐ SOAPConfigMO	SOAP_Req_Cfg_MO	<input type="checkbox"/>	1		
BodyName	String	<input checked="" type="checkbox"/>		getOrderStatus	
BodyNS	String	<input type="checkbox"/>		http://www.ibm.com	
BOVerb	String	<input type="checkbox"/>		Retrieve	
TypeInfo	String	<input type="checkbox"/>		true	
TypeCheck	String	<input type="checkbox"/>		none	
Style	String	<input type="checkbox"/>		rpc	
Use	String	<input type="checkbox"/>		encoded	

圖 43. rpc/encoded SOAP Config MO

**document/literal:** 若 Style 內容設定為 document，Use 內容設定為 literal 時，所包含的整個「主體名稱」標籤將不存在。下面是以上述 BO 為基礎的文件樣式 SOAP 訊息範例：

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:Elem1 xmlns:ns1="http://www.ibm.com/Elem1">
      <Child1>1</Child1>
```

```

        <Child2>2</Child2>
    </ns1:Elem1>
    <ns2:Elem1 xmlns:ns2="http://www.ibm.com/elem1">
        <Child1>3</Child1>
        <Child2>4</Child2>
    </ns2:Elem1>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

圖 44 顯示此 document/literal 訊息的對應商業物件。

Name	Type	Key	Card	Default	App Spec Info
Elem1	SOAP_Elem1	<input checked="" type="checkbox"/>	1		maxoccurs=3;elem_ns=http://www.ibm.com/elem1
Child1	String	<input checked="" type="checkbox"/>			
Child2	String	<input type="checkbox"/>			
ObjectEventId	String				
SOAPConfigMO	SOAP_Req_Cfg_MO	<input type="checkbox"/>	1		
BodyName	String	<input checked="" type="checkbox"/>		getOrderStatus	
BodyNS	String	<input type="checkbox"/>		http://www.ibm.com	
BOVerb	String	<input type="checkbox"/>		Retrieve	
TypeInfo	String	<input type="checkbox"/>		false	
TypeCheck	String	<input type="checkbox"/>		none	
Style	String	<input type="checkbox"/>		document	
Use	String	<input type="checkbox"/>		literal	

圖 44. document/literal SOAP Config MO

請注意，尚未設定 XML 程式碼片段中的 encodingStyle 屬性。

**document/encoded:** 此 Style/Use 組合不受支援。若資料處理常式遇到 SOAP ConfigMO 的 Style 設定為 document，且 Use 設定為 encoded，將會失敗。

## SOAP 資料處理常式處理程序

SOAP 資料處理常式會以下列方式執行 SOAP 訊息與商業物件之間的轉換作業：

- 從 SOAP 訊息轉換成商業物件的處理程序

- 當 Web 服務用戶端對顯現為 Web 服務的協同作業發出呼叫時，將於該事件處理程序的階段中，進行從要求訊息轉換成 SOAP 要求商業物件的資料處理程序。
- 當 Web 服務將 SOAP 回應訊息傳回發出呼叫的協同作業時，將於該要求處理程序期間，進行從回應訊息轉換成 SOAP 回應商業物件的資料處理程序。另一種會在此階段中發生的作業，便是從錯誤訊息轉換成 SOAP 商業物件的資料處理程序。

如需此處理程序的詳細說明，請參閱本節稍後的第 104 頁的『從 SOAP 主體訊息轉換成商業物件的處理程序』。

- 從商業物件轉換成 SOAP 訊息的處理程序

- 當顯現為 Web 服務的協同作業傳回回應商業物件時，將於該事件處理程序期間，進行從商業物件轉換成 SOAP 回應訊息的資料處理程序。另一種會在此階段中發生的作業，便是從錯誤商業物件轉換成 SOAP 錯誤訊息的資料處理程序。
- 當協同作業對連接器發出服務呼叫，將商業物件轉換成 SOAP 要求訊息時，將於該要求處理程序的階段中，進行從商業物件轉換成 SOAP 要求訊息的資料處理程序。

如需此處理程序的詳細說明，請參閱本節稍後的第 106 頁的『從商業物件轉換成 SOAP 訊息主體的處理程序』。

## 從 SOAP 主體訊息轉換成商業物件的處理程序

本節提供從 SOAP 主體訊息轉換成商業物件的逐步式說明。

1. SOAP 資料處理常式接收 SOAP 訊息。
2. 資料處理常式使用 Apache SOAP API 來剖析 SOAP 訊息。
3. 資料處理常式擷取 SOAP 訊息的元件：封套、標頭和主體。
4. **標頭處理程序**如需詳細資訊，請參閱第 105 頁的『從 SOAP 標頭訊息轉換成商業物件的處理程序』。
5. **主體處理程序**資料處理常式讀取 SOAP 主體的第一個元素，判斷其是否帶有錯誤或資料。若主體的內容不是個錯誤，資料處理常式將進行下列作業：
  - a. 執行商業物件解析，以決定要在轉換作業中使用哪個商業物件。若您已配置自訂名稱處理常式，則不適用下面所討論的預設商業物件解析。如需有關指定可插入的名稱處理常式的資訊，請參閱第 125 頁的『指定可插入的名稱處理常式』。
  - b. 資料處理常式亦會解析將要用來從事轉換作業的 SOAP Config MO (資料處理常式所建立之 SOAP 商業物件的子項)。若 SOAP Config MO 的實例不存在，資料處理常式便會建立實例，並讀取其預設值。資料處理常式會從 ConfigMO 屬性值來讀取商業物件動詞。資料處理常式會將 SOAP 商業物件實例化，並依情況來設定動詞。資料處理常式將會嘗試把 SOAP 訊息寫入這個商業物件。
  - c. 資料處理常式繼續剖析 SOAP 訊息，一次一個元素。若為 rpc，資料處理常式會預期第一個元素為母項。
  - d. 資料處理常式預期商業物件的屬性 (或其應用程式特有資訊：如需進一步資訊，請參閱第 109 頁的『從商業物件轉換成 SOAP 訊息之作業中的 ASI』) 的名稱應與子元素的名稱相同。若於商業物件中找不到屬性，資料處理常式便會擲出異常狀況。子元素可為簡式類型，亦可為複式類型。複式元素就是具有子元素的那些元素。
  - e. **簡式元素**若子元素為簡式元素，依預設，資料處理常式會預期商業物件屬性的名稱 (或 ASI) 與簡式元素的名稱相同。資料處理常式會讀取簡式元素的值，並將它設定於商業物件中。
  - f. **複式元素**若子元素為複式類型，資料處理常式會預期商業物件的屬性名稱 (或 ASI) 和類型與子項商業物件相同。此屬性可為單一基數或是多重基數，需視有無複式 SOAP 元素或 SOAP 陣列而定。接著，資料處理常式會將子項商業物件實例化 (依預設，屬性的類型會提供子項商業物件的名稱)，並讀取此複式元素的所有子元素，於子項商業物件中設定其值。資料處理常式在驗證此屬性的基數後，會將此子項商業物件設定於母項商業物件屬性中。若屬性為基數 n，資料處理常式便會將此商業物件附加至儲存區。複式元素可以有簡式或複式子元素。這些元素也是以同樣的方式來處理：若是簡式元素，資料處理常式就會在子項 BO 中設定值；若是複式元素，資料處理常式就會將子項商業物件實例化。
6. **錯誤處理程序**資料處理常式會讀取 SOAP 主體第一個元素的名稱，判斷它是否為錯誤。若第一個元素的名稱為 Fault，資料處理常式就會認定這是錯誤訊息。然後進行錯誤商業物件解析，判斷要將此錯誤訊息轉換成哪一種商業物件。接著資料處理常式會遵循與主體處理程序相同的處理程序來進行。資料處理常式預期子項商業物件中所指定的商業物件具有下列屬性：

- a. faultcode：必要的。String 屬性
  - b. faultstring：必要的。String 屬性
  - c. faultactor：非必要的 String 屬性
  - d. detail：非必要的。子項 BO
7. 若錯誤處理程序因任何原因而導致失敗，所擲出的異常狀況將會包含來自 SOAP 錯誤訊息中之 faultcode、faultstring 及 faultactor 元素的文字。

**註：**根據錯誤訊息的 SOAP 規格，faultcode、faultstring 及 faultactor 為簡式元素，而 detail 為複式元素 (含有子元素的元素)。此外，faultcode、faultstring、faultactor 及 detail 是屬於 SOAP 封套名稱空間，而 detail 子元素可能是屬於使用者定義的名稱空間。

## 從 SOAP 標頭訊息轉換成商業物件的處理程序

本節說明資料處理常式如何將 SOAP 訊息的標頭轉換成商業物件。

1. SOAP 資料處理常式處理 SOAP 訊息的主體。主體處理程序會建立 SOAP 商業物件。
2. 若 SOAP 訊息具有 SOAP 標頭元素，SOAP 資料處理常式預期由主體處理程序所取得的商業物件中，會有 SOAP 標頭屬性。SOAPHeader 屬性為商業物件的子屬性，並以 soap\_location=SOAPHeader 作為其應用程式特有資訊。若無這樣的屬性，SOAP 資料處理常式就會擲出錯誤。SOAPHeader 屬性的類型必須是「SOAP 標頭儲存區」商業物件。SOAP 資料處理常式會於步驟 1 中所取得的 SOAP 商業物件中，建立此屬性的實例。
3. 對於 SOAP-Env:Header 元素的每個直接子項：
  - a. 資料處理常式預期「SOAP 標頭儲存區商業物件」中有子屬性。此屬性的名稱必須與標頭元素的名稱相同，且必須符合「SOAP 標頭子項」商業物件。若資料處理常式找不到這樣的屬性，就會擲出錯誤。此外，此元素的名稱空間應與此屬性之 elem\_ns 應用程式特有資訊中所指定的相同。若不同，資料處理常式就會擲出錯誤。
  - b. 資料處理常式建立「SOAP 標頭子項」商業物件的實例，並將其置於步驟 2 所建立的「SOAP 標頭儲存區」商業物件中。
  - c. 若此標頭元素具有 actor 屬性，資料處理常式便會預期上面所建立的子項商業物件中，有 actor 屬性的存在。若找不到 actor 屬性，資料處理常式就會擲出錯誤。

**註：**若您要新增 actor 屬性，請參閱第 113 頁的『指定 SOAP 屬性』。

- d. 若此標頭元素具有 mustUnderstand 屬性，資料處理常式就會預期上面所建立的子項商業物件中，有 mustUnderstand 屬性的存在。若找不到 mustUnderstand 屬性，資料處理常式就會擲出錯誤。

**註：**若您要新增 mustUnderstand 屬性，請參閱第 113 頁的『指定 SOAP 屬性』。

- e. 對於此標頭元素的每個子元素，資料處理常式都會預期子項商業物件中有一個具有相同名稱的屬性。將會以處理 SOAP-Env:Body 元素之子元素的相同方式，來處理這些元素。

## 從商業物件轉換成 SOAP 訊息主體的處理程序

以下是從商業物件轉換成 SOAP 主體訊息的逐步式說明。欲參考有關應用程式特有資訊的特殊案例，請參閱第 109 頁的『從商業物件轉換成 SOAP 訊息之作業中的 ASI』

1. SOAP 資料處理常式尋找對應其所轉換之 SOAP 商業物件的 SOAP ConfigMO。
2. 資料處理常式編製 SOAP 訊息的封套及標頭。
3. 資料處理常式解析 SOAP ConfigMO。若 SOAP ConfigMO 的實例不存在，資料處理常式便會建立實例，並讀取其預設值。依預設，資料處理常式讀取 SOAP ConfigMO 中的 BodyName 屬性值，以判斷其是否正在處理錯誤商業物件。若其設定為 soap: fault，則該商業物件會被當作 SOAP 錯誤商業物件。若不是錯誤商業物件，資料處理常式就會執行下面編製主體所描述的處理程序，或是編製錯誤所描述的處理程序。
4. **編製主體**下列步驟詳細說明資料處理常式從商業物件編製 SOAP 訊息主體的處理程序：

- 資料處理常式從 SOAP ConfigMO 屬性取得 BodyName 及 BodyNS，然後編製 SOAP 訊息的第一個 (母項) 元素。依預設，第一個元素的名稱就是 BodyName 的值。於本書中，亦指主體元素。依預設，主體元素的名稱空間為 BodyNS 所決定的值。若 SOAP ConfigMO 的 Style 屬性設定為 document，則會跳過這個步驟 (建立第一個主體元素)。
- 然後資料處理常式會讀取商業物件的屬性，並依類型加以處理。各類型之屬性的處理程序說明如下。
  - **簡式屬性**若屬性是簡式類型，資料處理常式會從主體元素來建立子元素，並使用與屬性相同的名稱 (除非特殊的應用程式特有資訊另有指定)。資料處理常式會將此元素的值設定為商業物件中的屬性值。
  - **基數 1 子項商業物件屬性**

若屬性為單一基數子項商業物件，資料處理常式會建立主體元素的子元素。這就是所謂的子項商業物件元素。所建立之子元素的名稱與屬性的名稱相同 (除非特殊 ASI 內容另有指定)。然後資料處理常式會遍訪子項商業物件的屬性，以其用來處理送入之商業物件屬性的相同方法，來建立屬性的子元素。但是這些子元素不是建成立主體元素的子項，而是建成立子項商業物件元素的子項。

- **基數 n 子項商業物件屬性**如果屬性是基數 n 子項商業物件，資料處理常式就會建立 SOAP 陣列。每個屬性的處理方式，都與處理單一基數子項商業物件的方式一樣。
5. **編製錯誤**下列各節將說明資料處理常式編製錯誤訊息的整個過程。
    - 資料處理常式預期商業物件中有下列屬性：
      - faultcode：必要的，String 屬性
      - faultstring：必要的，String 屬性
      - faultactor：非必要的。String 屬性
      - detail：非必要的。子項 BO 屬性。若有遺漏任何必要的屬性，資料處理常式就會產生錯誤。
    - 資料處理常式建立 faultcode 的元素。其會設定商業物件之 faultcode 屬性所給定的值。

- 資料處理常式建立 faultstring 的元素。其會設定商業物件之 faultstring 屬性所給定的值。
  - 資料處理常式建立 faultactor。其會設定商業物件之 faultactor 屬性所給定的值。
  - 若 detail 屬性存在於商業物件中，則該屬性應屬於子項商業物件類型。否則資料處理常式會產生錯誤。它會處理上述編製主體一節中強調顯示之每個 detail 商業物件的屬性。
6. **CxIgnore 處理程序**若資料處理常式發現屬性值的設定為 CxIgnore，資料處理常式就不會為此屬性建立元素。
  7. **CxBlank 處理程序**若資料處理常式判斷出屬性值的設定為 CxBlank，資料處理常式就會為此屬性建立元素，但不會設定其值。

## 從商業物件轉換成 SOAP 訊息標頭的處理程序

本節僅說明 SOAP 標頭屬性的處理程序。所有其他屬性的處理程序說明，請見第 106 頁的『從商業物件轉換成 SOAP 訊息主體的處理程序』。

1. SOAP 資料處理常式從商業物件取得 SOAPHeader 屬性。此屬性具有 soap\_location=SOAPHeader，為其應用程式特有資訊。僅當此屬性的值為非空值時，SOAP 資料處理常式才會建立 SOAP-Env:Header 元素。若商業物件包含多個 SOAPHeader 屬性，則會處理第一個屬性，其他屬性則會被當作主體的一部分。
2. SOAP 資料處理常式預期 SOAPHeader 屬性為單一基數子項，代表「SOAP 標頭儲存區」商業物件。資料處理常式會處理類型為「SOAP 標頭子項」商業物件的「SOAP 標頭儲存區」商業物件子屬性。
3. 針對「SOAP 標頭儲存區」商業物件的每個屬性，資料處理常式將進行下列作業：
  - a. 檢查基數：若此屬性「不是」1 或 n 基數子項物件，則將之忽略。
  - b. 檢查值：若此屬性的值為「空值」，則將之忽略。
  - c. 若屬性為 1 或 n 基數子項物件，SOAP 資料處理常式就會建立一個標頭元素，作為步驟 1 所建立之 SOAP-Env:Header 元素的直接子項。此標頭元素的名稱與屬性的名稱相同。此元素的名稱空間是由此屬性的 elem\_ns 應用程式特有資訊所給定的。
  - d. 若屬性為「SOAP 標頭子項」商業物件，則會處理此商業物件的所有屬性。此屬性可能具有 actor 及 mustUnderstand 屬性。
 

**註：**若您要新增 mustUnderstand 或 actor 屬性，請參閱第 113 頁的『指定 SOAP 屬性』。
  - e. 若「SOAP 標頭子項」商業物件具有非空值的 actor 屬性，資料處理常式便會於步驟 c 中所建立的標頭元素中，建立 actor 屬性。
  - f. 若「SOAP 標頭子項」商業物件具有非空值的 mustUnderstand 屬性，資料處理常式便會於步驟 c 中所建立的標頭元素中，建立 mustUnderstand 屬性。
  - g. 「SOAP 標頭子項」商業物件的所有其他非空值屬性，皆會變成此標頭元素的子元素。其編製的方式和 SOAP-Env:Body 元素的子元素相同。

## 標頭錯誤處理程序

SOAP 規格指定專屬於標頭的錯誤，皆必須傳回至標頭中。將這些標頭傳回 SOAP 錯誤訊息中。就像訊息標頭是指定於要求及回應商業物件的 SOAPHeader 屬性中，錯誤標頭是指定於錯誤商業物件的 SOAPHeader 屬性中。

要求或回應商業物件的每個可能標頭，都可能會導致錯誤。這種錯誤皆會記錄於錯誤訊息的標頭中。

WSDL 文件有一個 SOAP 連結標頭錯誤元素，可讓您指定錯誤標頭。如需詳細資訊，請參閱「第 1 章」所列的 SOAP 及 WSDL 規格。

headerfault 的應用程式特有資訊，可讓您為每個標頭指定標頭錯誤。您可為「SOAP 標頭儲存區」商業物件的每個屬性，指定 headerfault 應用程式特有資訊。錯誤商業物件之「SOAP 標頭儲存區」商業物件中的屬性清單如下：

headerfault=attr1, attr2, attr3...

若「WSDL 配置精靈」於要求或回應物件的「SOAP 標頭子項」商業物件中，找到 headerfault 應用程式特有資訊，該公用程式就會在為這些標頭建立的 WSDL 中，建立 headerfault 元素。請注意，WSDL 可讓您為您的每個要求（輸入）及回應（輸出）標頭，指定多重標頭錯誤。因此，此應用程式特有資訊的值是一個以逗號隔開的屬性清單。

## 使用應用程式特有資訊功能

您可指定物件層次及屬性層次的應用程式特有資訊 (ASI)，以擴充及加強 SOAP 資料處理常式功能。表 45 顯示這些屬性，將於下面的一節中加以討論。表格中的所有輸入項目都是屬性層次的 ASI，除非另有指定。

表 45. SOAP 物件 ASI 摘要

ASI	可能值	說明
soap_location	SOAPHeader	將此商業物件屬性指定為標頭屬性
headerfault	String	指出錯誤 BO 中對應 SOAP 標頭的 BO 屬性名稱
elem_name	String	指定對應此 BO 屬性之 SOAP 元素的名稱
elem_ns	String	指定對應此 BO 屬性之 SOAP 元素的名稱空間
type_name	String	指定對應此 BO 屬性之 SOAP 元素的類型
type_ns	String	指定對應此 BO 屬性之元素的類型名稱空間
xsdtype	true	指定 xsd 作為對應此 BO 屬性之元素的名稱空間，並以最新版的 xsd (例如：2001) 置換舊版的 xsd (例如：1999、2000 等等)。
attr_name	String	指定對應此 BO 屬性之 SOAP 屬性的名稱
attr_ns	String	指定對應此 BO 屬性之 SOAP 屬性的名稱空間
arrayof	String	指定必須作為簡式類型陣列項目的位置保留區之 n 基數子項商業物件屬性的名稱



表 45. SOAP 物件 ASI 摘要 (繼續)

ASI	可能值	說明
dh_mimetype	String	指定將用來轉換此複式類型屬性之資料處理常式的 mimeType
cw_mo_*	String	此商業物件層次 ASI 指定 Child Config MO 的名稱，SOAP 資料處理常式會將該名稱解譯為 Meta 資料，而不是內容。只有 cw_mo_soap 可指定被處理為 Meta 資料的 Child Config MO；所有其他的 cw_mo_* 皆指定不同的元件，所以不包含於 SOAP 資料處理常式的處理程序內。所有其他 cw_mo* 都會被忽略。
cw_mo_soap	String	此商業物件層次 ASI 指定於轉換此商業物件時，所應使用之 Child Config MO 屬性的名稱。
cw_mo_jms	String	此商業物件層次 ASI 指定所要使用之 JMS Protocol Config MO 的名稱
cw_mo_http	String	此商業物件層次 ASI 指定所要使用之 HTTP Protocol Config MO 的名稱
wrapper	true	指定此商業物件中之封套物件的屬性名稱。封套物件用於特定的綱目指示元，絕不可序列化
maxoccurs	整數	指定此商業物件屬性可能出現的次數上限。依據 maxoccurs 的值，商業物件可能有，也可能沒有封套。
minoccurs	整數	指定此商業物件屬性可能出現的次數下限。依據 minoccurs 的值，物件可能有，也可能沒有封套。
全部	String	指定代表綱目中之 all 指示元的子屬性。
choice	String	指定代表綱目中之 choice 指示元的子屬性。

## 從商業物件轉換成 SOAP 訊息之作業中的 ASI

SOAP 資料處理常式使用商業物件的 ASI 來決定建構 SOAP 訊息的方法。除非另有說明，否則下面一節中討論的所有 ASI，都是指屬性層次 ASI，且無論情況如何，皆會執行所有字串型的比較作業。

## elem\_name 及 elem\_ns 處理程序

本節中所討論的範例假設屬性名稱爲 OrderId，SOAP 元素名稱空間字首爲 ns0。

1. 若 elem\_name 和 elem\_ns 皆未指定，則將 elem\_name 預設爲屬性名稱，將 elem\_ns 預設爲元素母項的名稱空間。不指定 ASI。

```
<OrderId>1</OrderId>
```

2. 若指定了 elem\_name，但未指定 elem\_ns，則會將 elem\_name 設定爲 ASI elem\_name 值，並將 elem\_ns 預設爲「SOAP 主體」的名稱空間。ASI 如下：

```
elem_name=CustOrderId  
<CustOrderId>2</CustOrderId>
```

3. 若指定了 elem\_ns，但未指定 elem\_name，則會將 elem\_name 預設爲屬性名稱，並將 elem\_ns 設定爲 ASI elem\_ns 值。只有在此元素範圍的其他地方都找不到元素名稱空間時，才會明確地寫入 xmlns 屬性。若找到了元素名稱空間，就會使用已定義的名稱空間字首。否則 (若未找到元素名稱空間)，就會爲 elem\_ns 產生專屬的字首。請參酌下列範例，其假設字首已定義於範圍中 (ns1 代表對應已定義於此元素範圍內之名稱空間的字首)。ASI 如下：

```
elem_ns= http://www.w3.org/2001/XMLSchema  
<ns1:OrderId>3</ns1:OrderId>
```

下列範例假設並未找到字首 (ns2 代表專屬的字首)。ASI 如下：

```
elem_ns=CustOrderIdNamespace  
<ns2:OrderId xmlns:ns2="CustOrderIdNamespace">3</ns2:OrderId>
```

4. 當 elem\_name 和 elem\_ns 皆已指定，elem\_name 和 elem\_ns 都會設定爲 ASI 值。在上述案例 3 中執行的相同檢查 (有關已定義的名稱空間)，亦適用於此。就如於案例 3 一樣，若尚未定義名稱空間，將爲 elem\_ns 產生專屬的字首。ASI 如下：

```
elem_name=CustOrderId;elem_ns=CustOrderIdNamespace  
<ns2:CustOrderId xmlns:ns2="CustOrderIdNamespace">1</ns2:OrderId>
```

## 簡式屬性的 type\_name 及 type\_ns 處理程序

例如本節中的範例，屬性名稱爲 OrderId，SOAP 元素名稱空間字首爲 ns0，屬性類型爲 String。

**註：** 僅當 Config MO 屬性 TypeInfo 爲 true 時，才會進行 type\_name 及 type\_ns 處理程序。

1. 當 type\_name 和 type\_ns 皆未指定，則會將 type\_name 預設爲簡式類型，將 type\_ns 預設爲 xml 綱目定義的名稱空間 (xsd)。不指定 ASI。

```
<OrderId xsi:type="xsd:string">1</OrderId>
```

2. 當指定了 type\_name，而未指定 type\_ns，則會將 type\_name 設定爲 ASI type\_name 值，將 type\_ns 預設爲元素的名稱空間。ASI 如下：

```
type_name=CustString  
<OrderId xsi:type="ns0:CustString">2</OrderId>
```

3. 當指定了 type\_ns，而未指定 type\_name，則會將 type\_ns 預設爲簡式類型名稱，並將 type\_name 設定爲 ASI type\_ns 值。用來處理字首的方式，可與建立 elem\_ns 的方式相比擬。除非元素範圍內已有名稱空間存在，否則就會產生類型名稱空間的專屬字首。ASI 如下：

```
type_ns=CustStringNamespace  
<OrderId xmlns:ns2="CustStringNamespace" xsi:type="ns2:String">3</OrderId>
```

4. 當 `type_name` 和 `type_ns` 皆已指定，便會將其設定為所指定的 ASI 值，並產生類型名稱空間的專屬字首。ASI 如下：

```
type_name=CustString;type_ns=CustStringNamespace
<OrderId xmlns:ns2="CustStringNamespace" xsi:type="
"ns2:CustString">1</OrderId>
```

### 單一基數屬性的 `type_name` 及 `type_ns` 處理程序

於本節中的範例，屬性名稱為 `OrderStaus`，SOAP 元素名稱空間字首為 `ns0`，屬性類型為 `OrderStatus`。

**註：** 僅當 Config MO 屬性 `TypeInfo` 為 `true` 時，才會進行 `type_name` 及 `type_ns` 處理程序。

1. 當 `type_name` 和 `type_ns` 皆未指定，則會將 `type_name` 預設為商業物件名稱，並將類型名稱空間預設為元素的名稱空間。不指定 ASI：

```
<OrderStatus xsi:type="ns0:OrderStatus">1</OrderStatus>
```

2. 當指定了 `type_name`，但未指定 `type_ns`，則會將 `type_name` 設定為所指定的 ASI 值，將 `type_ns` 預設為元素的名稱空間。ASI 如下：

```
type_name=CustOrderStatus
<OrderStatus xsi:type="ns0:CustOrderStatus">1</OrderStatus>
```

3. 當指定了 `type_ns`，但未指定 `type_name`，則會將 `type_name` 預設為商業物件名稱，將 `type_ns` 設定為所指定的 `type_ns` 值。並產生類型名稱空間的專屬字首。ASI 如下：

```
type_ns=CustTypeNS
<OrderStatus xsi:type="ns2:SOAP_OrderStatusLine
" xmlns:ns2="CustTypeNS">1</OrderStatus>
```

4. 當 `type_name` 和 `type_ns` 皆已指定，便會將其設定為所指定的 ASI 值，並產生類型名稱空間的專屬字首。ASI 如下：

```
type_name=CustOrderStatus;type_ns=CustTypeNS
<OrderStatus
xsi:type="ns2:CustOrderStatus" xmlns:ns2="CustTypeNS">1</OrderStatus>
```

### 多重基數屬性的 `type_name` 及 `type_ns` 處理程序

本節中的所有範例皆假設屬性名稱為 `MultiLines`，SOAP 元素名稱空間字首為 `ns0`。並假設屬性類型為 `OrderStatus`。

**註：** 僅當 Config MO 屬性 `TypeInfo` 為 `true` 時，才會進行 `type_name` 及 `type_ns` 處理程序。

1. 若 `type_name` 和 `type_ns` 皆未指定，則會將 `type_name` 預設為商業物件名稱，並將 `type_ns` 預設為元素的名稱空間。ASI 如下：

```
<MultiLines SOAP-ENC:arrayType="ns0:OrderStatus[2]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="SOAP-ENC:Array">
```

2. 若指定了 `type_name`，而未指定 `type_ns`，則會將 `type_name` 設定為所指定的 ASI `type_name` 值，將 `type_ns` 預設為元素的名稱空間。ASI 如下：

```
type_name=CustOrderStatus
<MultiLines SOAP-ENC:arrayType="ns0:CustOrderStatus[2]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="SOAP-ENC:Array">
```

3. 當指定了 `type_ns`，但未指定 `type_name`，則會將 `type_name` 預設為商業物件名稱，將 `type_ns` 設定為所指定的 ASI `type_ns` 值，並產生類型名稱空間的專屬字首。ASI 如下：

```
type_ns=CustTypeNS
<MultiLines SOAP-ENC:arrayType="ns2:OrderStatus[2]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns2="CustTypeNS" xsi:type="SOAP-ENC:Array">
```

4. 當 `type_name` 和 `type_ns` 皆已指定，便會將其設定為所指定的 ASI 值，並產生類型名稱空間的專屬字首。ASI 如下：

```
type_name=CustOrderStatus;type_ns=CustTypeNS
<MultiLines SOAP-ENC:arrayType="ns2:CustOrderStatus[2]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns2="CustTypeNS" xsi:type="SOAP-ENC:Array">
```

註：代表各「陣列」元素母項的項目元素，皆具有與 `arrayType` 相同的類型及名稱空間。

### 簡式、單一及多重基數類型的 `xsdtype`

若為簡式、單一及多重基數類型，請將類型名稱的 `xsdtype` ASI 屬性設定為 `true`，以符合 SOAP 訊息的現行 XSD。當 `type_name` 和 `type_ns` 內容皆已設定，則 `xsdtype` 為唯讀內容。在已指定 `type_name` 及 `type_ns` 的情況下，SOAP 資料處理常式會先嘗試使用 SOAP API，將該配對對映至 Java 類型。然後資料處理常式會嘗試使用「SOAP 訊息」的現行 XSD，將 Java 類型轉換回 SOAP 元素類型。比方說，若現行 XSD 為 `http://www.w3.org/2001/XMLSchema`

且具有下列 ASI：

```
type_name=timeInstant;type_ns=http://www.w3.org/1999/XMLSchema;xsdtype=true
```

則會將 SOAP 訊息類型寫成：

```
<OrderDate xsi:type="xsd:dateTime">
```

因為 2001 XSD 的 `dateTime` 同等於 1999 XSD 的 `timeInstant`。

### `xsdtype` 及簡式類型陣列

若為多重基數物件，您可建立簡式類型陣列，如下所示：

```
<MultiLines SOAP-ENC:arrayType="xsd:string[4]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="SOAP-ENC:Array">
```

欲達成此作業，請將 `type_name` 內容設定為所需的簡式類型（例如：`string`），並將 `type_ns` 內容設定為適當的 XSD 規格。然後，將 `xsdtype` 內容設定為 `true`，將類型轉換為現行的 XSD 類型。最後，`arrayOf` 內容應設定為要保存該簡式類型值之儲存區中屬性的名稱。下面為字串陣列 ASI 的樣式範例：

```
arrayof=size;type_name=string;type_ns=http://www.w3.org/2001/XMLSchema;xsdtype=true
```

## ASI 對錯誤處理程序的影響

`faultcode`、`faultactor`、`faultstring` 及 `detail` 元素應遵循下列規則：

1. 這些屬性中的任何 `elem_name`、`elem_ns`、`type_name` 及 `type_ns` ASI 皆會被忽略。
2. `detail` 元素的所有子項，都會完全按照主體處理程序中所描述的方法來寫入。

## ASI 對標頭處理程序的影響

您可以使用標頭子項物件層次以下的所有 ASI 內容 (請參閱表 45)。

### 指定 SOAP 屬性

#### 簡式類型的 attr\_name 處理程序

這個 XML 綱目案例之 complexTypes 的 simpleContent 延伸或限制有值和屬性。例如，請考量下列 SOAP 標籤：

```
<size system="us">10</size>
```

它是根據下列綱目：

```
<complexType name="SizeType">
  <simpleContent>
    <extension base="int">
      <attribute name="system" type="string"/>
    </extension>
  </simpleContent>
</complexType>
<element name="size" type="ns:SizeType"/>
```

對應於複式類型 (具有簡式內容延伸或限制) 的商業物件，除了對應於複式類型屬性的屬性之外，還必須包含一個其他屬性。這個其他屬性必須包含簡式內容值 (在上述範例中為 10，也就是元素大小的值)。將商業物件對應於複式類型作為其類型的商業物件屬性，會將 elem\_value=simpleContentValue 作為它的屬性層次 ASI。

圖 45 顯示對應的商業物件。

圖 45. 簡式類型的 attr\_name 商業物件

#### 單一及多重基數類型的 attr\_name 處理程序

您可以指定 ASI，將商業物件屬性轉換為 SOAP 屬性，而非轉換為 SOAP 元素。資料處理常式僅支援將 SOAP 屬性新增至複式單一及 n-card 類型。請參酌下列範例：

```
<CustInfo City="4" State="5" Street="2" Zip="6">
  <Name xsi:type="xsd:string">1</Name>
  <Street2 xsi:type="xsd:string">3</Street2>
</CustInfo>
```

假設商業物件定義結構如下 (屬性層次 ASI 指定於圖 46 中，每個屬性的右邊)，資料處理常式會遵循下列處理步驟：

Name	Type	App Spec Info
☐ CustInfo	CustomerInfo	
Name	String	
Street1	String	attr_name=Street
Street2	String	
City	String	attr_name=City
State	String	attr_name=State
Zip	String	attr_name=Zip

圖 46. attr\_name 商業物件

1. 當遍訪複式屬性時，資料處理常式會先為此複式商業物件屬性產生對應的標籤。在此範例中，CustInfo 代表複式商業物件屬性。
2. 資料處理常式會在複式商業物件的子項之間往返。僅考慮以簡式類型屬性來建立屬性。若簡式類型具有名為 attr\_name 的 ASI 內容，資料處理常式便會將此簡式類型當成屬性寫入 SOAP 元素。於此範例中，元素 (CustInfo) 將會有四種屬性：Street、City、State 及 Zip。
3. 至於商業物件的其餘屬性，則會使用標準的「主體」處理程序來將其寫入。這表示，即使是沒有 attr\_name ASI 的商業物件屬性，還是會為其評估所有相關的 ASI。

處理多重基數類型的邏輯，與處理單一基數類型的邏輯相同。尤其，每個 <item> 標籤都會對應於多重基數物件中的每個商業物件實例，且皆使用 ASI 加以處理。例如，假設此多重基數商業物件定義結構具有對應的 ASI：

Name	Type	Card	App Spec Info
☐ CustInfo	CustomerInfo	N	
Name	String		
Street1	String		attr_name=Street
Street2	String		
City	String		attr_name=City
State	String		attr_name=State
Zip	String		attr_name=Zip

圖 47. attr\_name 多重基數商業物件

若傳送至資料處理常式的事件具有此多重基數物件的二個實例，則所建立的 SOAP 訊息外觀如下：

```
<CustInfo>
  <item City="Armonk" Street="Main Street">
    <Name>IBM</Name>
    <Street2>None</Street2>
  </item>
  <item City="Burlingame" State="Ca" Street="577 Airport Blvd" Zip="94010">
    <Name>Burlingame Labs</Name>
    <Street2>Suite 600</Street2>
  </item>
</CustInfo>
```

請注意，item 標籤會被視為複式元素類型。BO 定義中的任何屬性將成為對應之 item 標籤的 SOAP 屬性。

## 簡式類型陣列的 `arrayof` 處理

只有在 SOAP 編碼簡式類型陣列的狀況下，才應該使用 `arrayof` ASI 內容。例如，類似下列序列化：

```
<CustomerNames SOAP-ENC:arrayType="xsd:string[4]" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="SOAP-ENC:Array">
  <item xsi:type="xsd:string">value1</item>
  <item xsi:type="xsd:string">value2</item>
  <item xsi:type="xsd:string">value3</item>
  <item xsi:type="xsd:string">value4</item>
</CustomerNames>
```

就需要商業物件定義，如圖 48 中的商業物件定義：

Request	SOAP_echoStringArray_N0562488530_Request	1	ws_botype=request
CustomerNames	SOAP_echoStringArray_N0562488530_N1185926546_ArrayOfstring	n	arrayof=Item,type_name=string,type_ns=http://www.w3.org/2001/XMLSchema
Item	String		type_name=string,type_ns=http://www.w3.org/2001/XMLSchema

圖 48. `arrayof` 商業物件

(為了明確起見，商業物件會從「要求」層次顯示)。

**註：**雖然並未顯示 SOAP Config MO 的 `TypeInfo` 內容，但是在這個範例中，它必須設定為 `true`，才能從商業物件結構衍生上述 SOAP 序列化。

同時，`arrayof` 內容也可以用來建立名稱不是 `item` 的陣列項目。在上述範例中，如果 `BO` 屬性名稱及 `"arrayof"` `asi` 內容值都是 `name`，則可以用 `<name>` 標籤取代 `<item>` 標籤。這會是序列化：

```
<CustomerNames SOAP-ENC:arrayType="xsd:string[4]"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="SOAP-ENC:Array">
  <name xsi:type="xsd:string">value1</name>
  <name xsi:type="xsd:string">value2</name>
  <name xsi:type="xsd:string">value3</name>
  <name xsi:type="xsd:string">value4</name>
</CustomerNames>
```

## `attr_name` 及 `attr_ns` 處理程序

您可能需要提供一個對應於已建立之 SOAP 屬性的名稱空間。若要這麼做，您可以為簡式類型指定 `attr_ns` ASI 內容。只有當 `attr_name` 存在於相同屬性的 ASI 時，資料處理常式才會處理 `attr_ns` 內容。`attr_name` 及 `attr_ns` 遵循下列規則：

1. 若 `attr_name` 和 `attr_ns exist` 都不存在，就會將商業物件屬性轉換成 SOAP 元素。
2. 僅當設定 `attr_name`，SOAP 屬性的名稱空間會預設為元素的名稱空間：

```
<CustInfo Street="577 Airport"></CustomerInfo>
```

3. 當僅設定 `attr_ns`，就會忽略該內容，並將商業物件屬性轉換成 SOAP 元素。
4. 當 `attr_name` 和 `attr_ns` 皆存在，就會建立 SOAP 屬性，如下所示：

```
<CustInfo ns2:Street="577 Airport" xmlns:ns2="AttrNS"></CustomerInfo>
```

## dh\_mimetype：呼叫資料處理常式

SOAP 資料處理常式可呼叫另一個資料處理常式，將商業物件撰寫為目前存在之資料處理常式的任何格式。若要這麼做，請在將 SOAP 子項商業物件轉送至「SOAP 字串」時，將已編碼的文字新增至 SOAP 訊息。

RNIF 文件為用來將 SOAP 元素值編碼的其中一種格式。若要使用這項功能，請在任何層次的 SOAP 子項商業物件中，新增 RNIF BO。若要通知 SOAP 資料處理常式，於 RNIF 商業物件轉換成字串時，呼叫另一個資料處理常式，請在屬性的 ASI 中，新增 dh\_mimetype 內容。dh\_mimetype ASI 內容的值必須是指定於 MO\_DataHandler\_Default Meta 物件中的合法的 mimeType。mimeType 可用來決定要呼叫哪一個資料處理常式來處理商業物件。

圖 49 顯示 SOAP 子項商業物件，其中 CustomerInfo 為複式子項，RNET\_Pip3A2PriceAndAvailabilityQuery 為 RNIF 商業物件：

Name	Type	App Spec Info
CustomerInfo	CustomerInfo	
Name	String	
CustID	String	
RNIFMsg	RNET_Pip3A2PriceAndAvailabilityQuery	elem_name=RNIFexample;dh_mimetype=application/x_rossettanet_agent;type_name=base64Binary;type_ns=http://www.w3.org/2001/XMLSchema;xsdtype=true

圖 49. 具有 dh\_mimetype 的 RNIF 商業物件

從此商業物件所建立的 SOAP 訊息，看起來就像下面這個樣子：

```
<CustomerInfo>
<Name>IBM Corporation</Name>
<CustID>95626</CustID>
<RNIFexample
xsi:type="xsd:base64Binary">1AWERYER238W98EYR9238728374871892787ASRJK23423
JKAWERJ234AWERIJHI423488R4HASF1AWERYER238W98EYR9238728374871892787ASRJK234
34JKAWERJ234AWERIJHI423488R4HASF1AWERYER238W98EYR9238728374871892787ASRJK2
4234JKAWERJ234AWERIJHI423488R4HASF1AWERYER238W98EYR9238728374871892787ASRJK
234234JKAWERJ234AWERIJHI423488R4HASF1AWERYER238W98EYR9238728374871892787ASRJK
234234JKAWERJ234AWERIJHI423488R4HASF1AWERYER238W98EYR9238728374871892787ASRJK
</RNIFexample>
</CustomerInfo>
```

請注意，RNIF 範例元素包含 RNIF 編碼的字串，且該字串已用 base64 二進位碼來編碼為其元素值。另請注意，elem\_name、elem\_ns、type\_name、type\_ns 及 xsdtype ASI 內容仍會為此商業物件屬性保持其關聯性。於此範例中，所指定的 elem\_name 會在建立訊息時，指定 SOAP 元素的名稱。

**註：**若所呼叫之資料處理常式傳回的元素值是已編碼的文字，則 type\_name 內容必須設定為 base64Binary，type\_ns 必須對應 xsd 名稱空間，而 xsdtype 必須設定為 true。

**xsd:base64Binary：**當您設定 type\_name 及 type\_ns 來解析成 xsd:base64Binary 時，SOAP 資料處理常式會先從商業物件來將該值編碼，再設定對應元素的值。資料處理常式使用 Apache API 來查詢 base64Binary 序列化常式的登錄，將所呼叫之資料處理常式傳回的字串序列化，並設定元素的值。



## 綱目 complexType 指示元

下面的一節將討論綱目 complexType 指示元對商業物件的影響。指示元包含：

- maxOccurs
- minOccurs
- all
- sequence
- choice

**簡式類型的 maxOccurs 及 minOccurs 指示元：** maxOccurs 指示元指定元素可出現在複式類型中的次數上限。 minOccurs 指示元指定元素應出現在複式類型中的次數下限。

請參酌下面這個綱目：

```
<xs:element name="Address" type="Address">
<xs:complexType name="Address">
  <xs:sequence>
    <xs:element name="AddressLine" type="xsd:string" maxOccurs="10"/>
    <xs:element name="SuiteNumber" type="xsd:string" minOccurs="3"
      maxOccurs="unbounded"/>
    <xs:element name="City" type="xsd:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

上述範例指出 AddressLine 元素最多可出現在 Address 元素中 10 次，而 SuiteNumber 元素則必須至少出現 3 次。對應此綱目的商業物件必須針對具有下列 ASI 的每個 maxOccurs/minOccurs 指示元，各有一個 N 基數封套物件：

```
maxOccurs=N;wrapper=true
```

或

```
minOccurs=3;wrapper=true;
```

wrapper=true ASI 指示此物件是一個封套，因此不會明確地寫入 SOAP 訊息中。但是此封套物件中必須有一個簡式類型的子項。在執行時期，為了進行從 SOAP 轉換成商業物件的作業，資料處理常式會讀取封套的 N 子項物件，並為每個子項物件建立對應的元素。當執行從商業物件轉換成 SOAP 訊息的作業時，資料處理常式會針對它所遇到的每個元素，在 N 基數封套中建立子項物件。

對應的 SOAP 商業物件類似圖 50 所顯示的物件。

Pos	Name	Type	Key	Card	App Spec Info
1	☐ Address	Address	<input checked="" type="checkbox"/>	1	
1.1	☐ AddressLine	AddressLine_wrap	<input checked="" type="checkbox"/>	N	maxoccurs=10;wrapper=true
1.1.1	AddressLine	String	<input checked="" type="checkbox"/>		
1.1.2	ObjectEventId	String			
1.2	☐ SuiteNumber	SuiteNumber_wrap	<input checked="" type="checkbox"/>	N	minoccurs=3;wrapper=true
1.2.1	SuiteNumber	String	<input checked="" type="checkbox"/>		
1.2.2	ObjectEventId	String			
1.3	City	String	<input type="checkbox"/>		
1.4	ObjectEventId	String			
2	ObjectEventId	String			

圖 50. SOAP 商業物件中之簡式類型 ASI 的 *minOccurs* 及 *maxOccurs*

對應於圖 50 中所顯示之商業物件的 SOAP 訊息如下：

```
<Address xsi:type="ns0:Address">
  <AddressLine xsi:type="xsd:string">Line1</AddressLine>
  <AddressLine xsi:type="xsd:string">Line2</AddressLine>
  <SuiteNumber xsi:type="xsd:string">600</SuiteNumber>
  <SuiteNumber xsi:type="xsd:string">650</SuiteNumber>
  <SuiteNumber xsi:type="xsd:string">700</SuiteNumber>
  <City xsi:type="xsd:string">San Francisco</City>
</Address>
```

**註：** SOAP 資料處理常式處理 *maxOccurs* 及 *minOccurs* 指示元的方式是相同的，不會驗證元素出現次數的上限或下限。資料處理常式只會提供儲存區結構，以存放具有 *maxOccurs* 及 *minOccurs* 指示元之特定元素的多重實例。此作業適用於簡式及複式類型。

**複式類型的 *maxOccurs* 及 *minOccurs* 指示元：** *<maxOccurs>* 指示元指定元素可出現在複式類型中的次數上限。*<minOccurs>* 指示元指定元素應出現在複式類型中的次數下限。請參酌下列綱目中的 *maxOccurs* 指示元：

```
<xs:element name="Address" type="Address">
<xs:complexType name="Address">
  <xs:sequence>
    <xs:element name="AddressInfo" type="AddressInfo" maxOccurs="3"/>
    <xs:element name="City" type="xsd:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:complexType name="AddressInfo">
  <xs:sequence>
    <xs:element name="StreetLine" type="xsd:string"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

上述範例指出 *AddressInfo* 複式類型元素最多可出現在 *Address* 元素中 3 次。此綱目的對應商業物件將不會有封套物件，因為 *complexType AddressInfo* 本身即屬於 N 基數。下列 ASI 將會置於 N 基數屬性中：*maxOccurs=3*

圖 51 顯示對應的 SOAP 商業物件。

Pos	Name	Type	Key	Card	App Spec Info
1	☐ Address	Address	<input checked="" type="checkbox"/>	1	
1.1	☐ AddressInfo	AddressInfo	<input checked="" type="checkbox"/>	N	maxoccurs=3
1.1.1	StreetLine	String	<input checked="" type="checkbox"/>		
1.1.2	ObjectEventId	String			
1.2	City	String	<input type="checkbox"/>		
1.3	ObjectEventId	String			
2	ObjectEventId	String			

圖 51. SOAP 商業物件中之複式類型 ASI 的 *minOccurs* 及 *maxOccurs*

對應於圖 51 中所顯示之商業物件的 SOAP 訊息如下：

```
<Address xsi:type="ns0:Address">
  <AddressInfo xsi:type="ns0:AddressInfo">
    <StreetLine xsi:type="xsd:string">100 Market St.</ StreetLine>
    <StreetLine xsi:type="xsd:string">Apt 15</ StreetLine>
  </AddressInfo>
  <City xsi:type="xsd:string">San Francisco</City>
</Address>
```

**all 指示元:** 依預設，all 指示元指定此 complexType 的子元素可以任何順序出現，而每個子元素必須出現零或一次。請參酌下列綱目：

```
<complexType name="Item">
<all>
  <element name="quantity" type="xsd:int"/>
  <element name="product" type="xsd:string"/>
</all>
</complexType>
```

上述範例指示元素 quantity 及 product 可於 SOAP 訊息中，以任何順序出現。quantity 元素可先出現，然後出現 product 元素，反之亦然。

圖 52 顯示對應於此綱目片段的商業物件。

Pos	Name	Type	Card	App Spec Info
1	☐ Item	Item	1	all=Item_wrapper
1.1	☐ Item_wrapper	Item_wrapper	N	wrapper=true
1.1.1	quantity	String		
1.1.2	product	String		
1.1.3	ObjectEventId	String		
1.2	ObjectEventId	String		
2	ObjectEventId	String		

圖 52. SOAP 商業物件中 ALL 指示元 ASI

對應的 SOAP 訊息片段如下：

```
<Item xsi:type="ns0:Item">
  <quantity xsi:type="xsd:string">12</quantity>
  <product xsi:type="xsd:string">2</product>
</Item>
```

**處理具有 'all' 內容模型的陣列內容:** SOAP 資料處理常式可處理具有 'all' 內容模型的複式類型陣列內容，如本節所說明。在範例中，ArrayOfSOAPStruct 包含具有 'all' 內容模型的 SOAPStruct。

```
<complexType name="SOAPStruct">
  <all>
    <element name="varString" type="string" />
    <element name="varInt" type="int" />
    <element name="varFloat" type="float" />
  </all>
</complexType>
<complexType name="ArrayOfSOAPStruct">
  <complexContent>
    <restriction base="SOAP-ENC:Array">
      <attribute ref="SOAP-ENC:arrayType"
        wsdl:arrayType="typens:SOAPStruct[]" />
    </restriction>
  </complexContent>
</complexType>
```

SOAP 資料處理常式必須在序列化上產生下列 SOAP 資料：

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns0:echoStructArray xmlns:ns0="http://soapinterop.org/">
      <inputStructArray SOAP-ENC:arrayType="ns1:SOAPStruct[2]"
        xmlns:ns1="http://soapinterop.org/xsd" xsi:type="SOAP-ENC:Array">
        <item>
          <ns1:varFloat xsi:type="xsd:string">1.1</ns1:varFloat>
          <ns1:varInt xsi:type="xsd:string">1</ns1:varInt>
          <ns1:varString xsi:type="xsd:string">hi</ns1:varString>
        </item>
        <item>
          <ns1:varString xsi:type="xsd:string">hello</ns1:varString>
          <ns1:varInt xsi:type="xsd:string">1</ns1:varInt>
          <ns1:varFloat xsi:type="xsd:string">1.1</ns1:varFloat>
        </item>
      </inputStructArray>
    </ns0:echoStructArray>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

在此範例中，echoStructArray 是作業的名稱，而 inputStructArray 是參數名稱，類型為 ArrayOfSOAPStruct。

**sequence 指示元:** sequence 指示元指定子元素必須以 complexType 中所指定的順序出現。

```
<complexType name="Item">
  <sequence>
    <element name="quantity" type="int"/>
    <element name="product" type="string"/>
  </sequence>
</complexType>
```

SOAP 資料處理常式不需要特殊的 ASI，或是此指示元的封套物件。依預設，資料處理常式會以商業物件中所指定的順序來讀取及寫入 SOAP 元素。

**choice 指示元:** choice 指示元指定僅有其中一個 complexType 中的元素可出現於 SOAP 訊息中。請參酌下列綱目：

```

<complexType name="Item">
  <choice>
    <element name="quantity" type="int"/>
    <element name="product" type="string"/>
  </choice>
</complexType>

```

SOAP 資料處理常式不需要特殊的 ASI，或是此指示元的封套物件。將商業物件轉換為 SOAP 訊息時，資料處理常式會延遲，讓您選擇哪些元素應出現在 SOAP 訊息中。將 SOAP 訊息轉換成商業物件時，資料處理常式會讀取現有的元素，並移入其所對應的屬性。

**sequence、choice、group 及 all 上的 maxOccurs 指示元：** 模型群組 (sequence、choice、group 及 all) 具有 minOccurs 及 maxOccurs 屬性。minOccurs 及 maxOccurs 的預設值是 one。對於 all 群組來說，maxOccurs 只能採用 one 這個值。WSDL ODA 及 SOAP 資料處理常式可支援 sequence、choice 及 group 上的 maxOccurs 所有可能值。

## 從 SOAP 轉換成商業物件之作業中的 ASI

SOAP 資料處理常式使用商業物件的 ASI 來讀取及驗證送入的 SOAP 訊息。以下為 SOAP 資料處理常式用來進行 ASI 驗證的規則：

- 標頭與主體處理程序相同。
- SOAP ConfigMO 內容 TypeCheck 必須設定為 strict，而 TypeInfo 必須設定為 true，使資料處理常式執行下面一節所說明的驗證工作。
- type\_name 及 type\_ns 驗證工作同時進行，因為類型驗證對這二種內容的相依性都愈來愈重。

**註：** 除非另有說明，否則下面一節中討論的所有 ASI，皆為屬性層次 ASI。

### elem\_name 驗證

下列規則適用於簡式、基數 1 及基數 n 屬性的驗證工作：

1. 若於剖析 SOAP 訊息時遇到元素，資料處理常式將先搜尋商業物件層次的所有 ASI，並嘗試將元素的名稱與 elem\_name 值做比對。
2. 若找不到相符項，資料處理常式會嘗試將元素的名稱與該商業物件層次的每個屬性名稱做比對。
3. 若所有搜尋皆不成功，資料處理常式即失敗。

### elem\_ns 驗證

下列案例適用於簡式、基數 1 及基數 n 屬性的驗證工作：

1. 若此元素的 SOAP 訊息中沒有 elem\_ns ASI，xmlns 亦不存在，則此元素已驗證完成。
2. 若 elem\_ns ASI 不存在，且 SOAP 訊息的對應元素指定了 xmlns，則資料處理常式會將 elem\_ns 預設成從範圍中之商業物件讀取的最後一個 elem\_ns。資料處理常式會將此值與 SOAP 訊息的 xmlns 值比較。若無相符的項目，驗證作業就會失敗。
3. 若 elem\_ns ASI 確實存在，且 SOAP 訊息的對應元素未指定 xmlns，資料處理常式會驗證 ASI 中指定的 elem\_ns 與 SOAP 訊息現行範圍中的其中一個名稱空間相符。若無相符的項目，驗證作業就會失敗。

## type\_name 及 type\_ns 驗證

下面的一節將會討論 type\_name 及 type\_ns 驗證作業。

**簡式屬性:** 若 xsdType 為 true，則下列規則適用於 type\_name 及 type\_ns 驗證作業：

- **type\_name 和 type\_ns 皆已指定** 資料處理常式會使用 type\_name 與 type\_ns 的配對來建立對應的 java 類別物件。若使用送入的 SOAP 訊息 typename 及 typenamespace，則會查詢另一個 java「類別」物件。若這二個 java 類別物件相符，則驗證成功。否則驗證失敗。
- **type\_name 和 type\_ns 皆未指定** 資料處理常式會將簡式商業物件屬性對映至 java「類別」物件。若使用送入的 SOAP 訊息 typename 及 typenamespace，則會查詢另一個 java「類別」物件。若這二個 java「類別」物件相符，則驗證成功。否則驗證失敗。
- **僅指定 type\_name** 「簡式類型驗證」失敗。若 xsdType 為 true，則 type\_name 和 type\_ns 二個都要指定，或是都不指定。
- **僅指定 type\_ns** 「簡式類型驗證」失敗。若 xsdType 為 true，則 type\_name 和 type\_ns 二個都要指定，或是都不指定。

若 xsdType 為 false，則下列規則適用於 type\_name 及 type\_ns 驗證作業：

- **type\_name 和 type\_ns 皆已指定** 資料處理常式會於 SOAP 訊息 typename 和 typenamespace 配對與 ASI 中所指定的 type\_name 和 type\_ns 值之間，執行直接比較。若這二對完全相符，則驗證成功。否則驗證失敗。
- **type\_name 和 type\_ns 皆未指定** 資料處理常式會將簡式商業物件屬性對映至 java「類別」物件。若使用送入的 SOAP 訊息 typename 及 typenamespace，則會查詢另一個 java「類別」物件。若這二個 java「類別」物件相符，則驗證成功。否則驗證失敗。
- **僅指定 type\_name** type\_ns 值預設為在商業物件 ASI 中找到的元素名稱空間。資料處理常式會使用這個預設的 type\_ns 和 ASI 中所指定 type\_name，於這些值與 SOAP 訊息 typename 和 typenamespace 之間，執行直接比較。若這二對完全相符，則驗證成功。否則驗證失敗。
- **僅指定 type\_ns** type\_name 值預設為商業物件屬性類型。資料處理常式會使用此預設的 type\_name 和 ASI 中所指定 type\_ns，於這些值與 SOAP 訊息 typename 和 typenamespace 之間，執行直接比較。若這二對完全相符，則驗證成功。否則驗證失敗。

**複式屬性 (基數 1 及 n):** 若 xsdType 為 true，則下列規則適用於 type\_name 及 type\_ns 驗證作業：

- **type\_name 及 type\_ns 皆已指定** 將忽略 xsdType。資料處理常式處理的方式就如 xsdType 為 false 一樣。
- **type\_name 及 type\_ns 皆未指定** 將忽略 xsdType。資料處理常式處理的方式就如 xsdType 為 false 一樣。
- **僅指定 type\_name** 將忽略 xsdType。資料處理常式處理的方式就如 xsdType 為 false 一樣。
- **僅指定 type\_ns** 將忽略 xsdType。資料處理常式處理的方式就如 xsdType 為 false 一樣。

若 xsdType 為 false，則下列規則適用於 type\_name 及 type\_ns 驗證作業：

- **type\_name** 和 **type\_ns** 皆已指定資料處理常式會於 SOAP 訊息 typename 和 typenamespace 配對與 ASI 中所指定的 type\_name 和 type\_ns 值之間，執行直接比較。若這二對完全相符，則驗證成功。否則驗證失敗。
- **type\_name** 和 **type\_ns** 皆未指定 type\_name 值預設為商業屬性類型。type\_ns 值預設為在商業物件 ASI 中找到的元素名稱空間。資料處理常式會使用此預設行為，在這些值與 SOAP 訊息 typename 和 typenamespace 的配對之間，執行直接比較。若這二對完全相符，則驗證成功。否則驗證失敗。
- **僅指定 type\_name** type\_ns 值預設為在商業物件 ASI 中找到的元素名稱空間。資料處理常式會使用這個預設的 type\_ns 和 ASI 中所指定 type\_name，於這些值與 SOAP 訊息 typename 和 typenamespace 之間，執行直接比較。若這二對完全相符，則驗證成功。否則驗證失敗。
- **僅指定 type\_ns** type\_name 值預設為商業物件屬性類型。資料處理常式會使用此預設的 type\_name 和 ASI 中所指定 type\_ns，於這些值與 SOAP 訊息 typename 和 typenamespace 之間，執行直接比較。若這二對完全相符，則驗證成功。否則驗證失敗。

### attr\_name 及 attr\_ns 驗證

將 SOAP 訊息讀入商業物件時，會在每個 SOAP 元素中搜尋 SOAP 屬性。若找到了，則將這些屬性與對應 BO 的 attr\_name 內容值做比較。例如，您可參酌下面這個 SOAP 訊息：

```
<CustInfo City="4" State="5" Street="2" Zip="6">
  <Name xsi:type="xsd:string">1</Name>
  <Street2 xsi:type="xsd:string">3</Street2>
</CustInfo>
```

現在請參照圖 53 中所顯示的商業物件定義結構 (屬性層次 ASI 指定在每個屬性的右邊)。

Name	Type	App Spec Info
☐ CustInfo	CustomerInfo	
Name	String	
Street1	String	attr_name=Street
Street2	String	
City	String	attr_name=City
State	String	attr_name=State
Zip	String	attr_name=Zip

圖 53. attr\_name 及 attr\_ns 驗證

資料處理常式將遵循下列處理步驟：

1. 讀取元素名稱 CustInfo。
2. 解析對應於此元素名稱的商業物件屬性。
3. 讀取 SOAP 元素的屬性，並嘗試將其與子屬性的 ASI 做比對。於此案例中，SOAP 訊息 Street 要比對商業物件屬性 Street1，City 要比對商業物件屬性 City，以此類推。
4. 讀取和處理 CustInfo 之子元素的方式與主體的其餘部份一樣。

註: 未驗證 attr\_ns。

資料處理常式會在所有 SOAP 屬性中巡迴，尋找給定的元素。每遇到一個屬性，資料處理常式就會在商業物件中搜尋對應的屬性。若找到了，則將該 SOAP 屬性的值移入商業物件屬性。若找不到對應的商業物件屬性，資料處理常式就會繼續尋找下一個 SOAP 屬性。

## 從 SOAP 資料處理常式中呼叫資料處理常式

SOAP 資料處理常式可以使用另一個資料處理常式，將已編碼的元素值從 SOAP 訊息讀取至商業物件中。例如，RNIF 文件可能是將 SOAP 元素值編碼的其中一種格式。若要使用此功能，可於任何層次的 SOAP 子項商業物件中，新增 RNIF 商業物件。若要對 SOAP 資料處理常式表示，在將此 RNIF 編碼的「字串」轉換成 RNIF 商業物件時，必須使用另一個資料處理常式，則您必須將 dh\_mimetype 內容新增至屬性的 ASI。dh\_mimetype ASI 的值應是指定於 MO\_DataHandler\_Default 商業物件中的合法 mimeType。mimeType 可用來決定要用哪個資料處理常式來處理「字串」。例如，下列的 SOAP 訊息，其中 RNIFExample 是包含 RNIF 編碼「字串」的 SOAP 元素：

```
<CustInfo>
<Name>IBM Corporation</Name>
<CustID>95626</CustID>
<RNIFexample xsi:type="xsd:base64Binary">
1AWERYER238W98EYR9238728374871892787ASRJK234234JKAWER
J234AWERIJHI423488R4HASF1AWERYER238W98EYR923872837487
1892787ASRJK234234JKAWERJ234AWERIJHI423488R4HASF1AWER
YER238W98EYR9238728374871892787ASRJK234234JKAWERJ234A
WERIJHI423488R4HASF1AWERYER238W98EYR92387283748718927
87ASRJK234234JKAWERJ234AWERIJHI423488R4HASFWR234
</RNIFexample>
</CustomerInfo>
```

SOAP 商業物件看起來如圖 54 所顯示。

Name	Type	App Spec Info
☐ CustInfo	CustomerInfo	
Name	String	
Street1	String	attr_name=Street
Street2	String	
City	String	attr_name=City
State	String	attr_name=State
Zip	String	attr_name=Zip

圖 54. RNIFExample 商業物件

請注意，RNIFExample 元素包含 RNIF 編碼的「字串」，以作為其元素值。另請注意，elem\_name、elem\_ns、type\_name、type\_ns 及 xsdtype ASI 內容仍會為此商業物件屬性保持其關聯性。

註: 若所呼叫之資料處理常式傳回的元素值為已編碼的文字，則 type\_name 內容必須設定為 base64Binary，type\_ns 必須對應 xsd 名稱空間，而 xsdtype 必須設定為 true。

## 預設商業物件解析

為了進行由 SOAP 轉換為商業物件的作業，SOAP 資料處理常式及 Web 服務連接器會遵循一個特殊的交換資訊合約，解析商業物件名稱。連接器會為 SOAP 資料處理常式，



提供一個對映至 `BodyName` 和 `BodyNamespace` 配對的商業物件名稱清單。此外，若 TLO 中設定了 `defaultfault` 商業物件，則將此資訊傳遞給資料處理常式。在已提供此資訊的情況下，SOAP 資料處理常式會使用下列步驟來進行處理程序：

1. 資料處理常式接收 SOAP 訊息
2. 資料處理常式要判斷此為 SOAP 要求、回應或錯誤訊息。
  - a. 若為 SOAP 要求或回應訊息，資料處理常式便會從 `SOAP-ENV:Body` 元素的第一個子元素來讀取 `BodyName` 及 `BodyNamespace`。
  - b. 若為 SOAP 錯誤訊息，資料處理常式便會從錯誤訊息中之 `detail` 元素的第一個子元素來讀取 `BodyName` 及 `BodyNamespace`。若錯誤訊息中沒有 `detail` 元素，資料處理常式就會使用 `defaultfault` 商業物件來進行此轉換作業。
3. 若尚未選取 `defaultfault` 商業物件，資料處理常式便會嘗試將在步驟 2 中所找到的 `BodyName` 和 `BodyNamespace`，與在連接器所提供之清單中找到的配對做比對。若找到了相符的配對，則商業物件解析作業成功。若未找到相符的配對，則資料處理常式失敗，並產生一則有意義的錯誤訊息。

---

## 指定可插入的名稱處理常式

若使用預設商業物件解析作業，您可以指定可插入的名稱處理常式，以決定要在從 SOAP 訊息轉換成商業物件的作業中，使用哪個商業物件。您可變更 `MO_DataHandler_DefaultSOAPConfig` 屬性來執行這個動作。

`MO_DataHandler_DefaultSOAPConfig` 具有兩個類型為 `string` 的屬性，而它們可指定：

- **ClassName** SOAP 資料處理常式基礎類別的類別名稱。當您指定可插入的名稱處理常式時，請勿變更此屬性值。
- **SOAPNameHandler** `SOAPNameHandler` 屬性可指定要呼叫哪一個名稱處理常式。您可以為可插入的名稱處理常式指定值。這個內容的值應為類別名稱。`SOAPNameHandler` 類別是具有下列簽章的抽象類別：

```
public abstract String getBOName(Envelope msgEnv, SOAPProperty prop)
```

若 `SOAPNameHandler` 屬性有一個值，SOAP 資料處理常式就會呼叫所指定的名稱處理常式。若該值不存在，或所指定的名稱處理常式無法取得商業物件名稱，就會依預設來呼叫 SOAP 資料處理常式，以執行預設的商業物件解析作業。

SOAP `DataHandler` 會使用在 MO 中指定的 `SOAPNameHandler` 內容，來建立自訂名稱處理常式類別的實例。然後它會呼叫 `getBOName` 來解析商業物件名稱。SOAP `DataHandler` 會將它接收的 `SOAPProperty` 物件，從連接器傳遞到自訂名稱處理常式實作類別。

這個 `SOAPProperty` 物件包含潛在候選 BO 的結構式清單以供解析。清單中包含 `BodyName`、`BodyNamespace` 及 `BOName` 一組三個項目。這組三個項目是根據 SOAP `Config MO` 配置資訊而來。「預設名稱處理常式」會使用這個物件來解析 BO。自訂名稱處理常式開發人員可自行判斷是否使用這個物件。

## 使用 SOAPProperty 物件

您可使用 `SOAPPropertyUtils` 類別，從 `SOAPProperty` 擷取商業物件名稱。若要這樣做，請使用下列方法：

```

/**
 * Retrieve the business object name based on the body name and the body
 * namespace
 * .
 * @param soapProp top level SOAPProperty object that is passed by the
 * connector
 * @param name body name from the SOAP message
 * @param uri body namespace from the SOAP message
 * @return business object name from the SOAPProperty object with the body
 * name and body namespace.
 */
java.lang.String findBOName(SOAPProperty soapProp, String name, String uri);

```

## 範例 NameHandler

以下是範例 NameHandler:package

```

com.ibm.adapters.datahandlers.soap.namehandlers;
// DOM and Parsers
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.xml.sax.InputSource;
// Apache Xerces and SOAP
import org.apache.soap.Envelope;
import org.apache.soap.Header;
import org.apache.soap.Body;
import org.apache.soap.Constants;
import org.apache.soap.util.xml.DOMUtils;
import org.apache.soap.util.xml.XMLParserUtils;
import org.apache.soap.util.xml.QName;
import org.apache.soap.encoding.soapenc.SoapEncUtils;
import org.apache.soap.encoding.soapenc.Base64;
// java
import java.util.Vector;
// SOAP data handler
import com.ibm.adapters.datahandlers.soap.*;
import com.ibm.adapters.datahandlers.soap.exceptions.*;
public class MyCustomNameHandler extends SOAPNameHandler {
    private static final String BOPREFIX = "MyCustomBOPrefix";
    private static final char UNDERSCORE = '_';
    private static final char EMPTY_STRING = "";

    public String getBOName(Envelope msgEnv, SOAPProperty prop)
        throws SOAPNameHandlerException
    {
        // Initialize a String Buffer
        StringBuffer boName = new StringBuffer();
        // Determine the "MyCustomBOPrefix" SOAP data handler
        // MO property. If it exists, and is populated append
        // this prefix to the front of the BOName.
        String pref = dh.getOption(BOPREFIX);
        if (pref != null) {
            boName.append(pref.equals(EMPTY_STRING)
                ? EMPTY_STRING : pref + UNDERSCORE);
        }
        // Begin parsing the SOAP msg envelope.
        Element bodyEl, requestEl;
        Body msgBody = msgEnv.getBody();
        Vector bodyEntries = msgBody.getBodyEntries();
        if((bodyEntries == null) || (bodyEntries.size() <= 0))
            throw new SOAPNameHandlerException("No Body Entries exist
                for this SOAP message. Cannot determine BOName to use.");
        // Grab the first <SOAP-ENV:Body> Element
        bodyEl = (Element) bodyEntries.elementAt(0);
        // Grab the first Child Element of the <SOAP-ENV:Body>
        // Element

```

```

requestEl = (Element) DOMUtils.getFirstChildElement(bodyEl);
// Read the name and namespace of this first child
String name = bodyEl.getLocalName();
String uri = bodyEl.getNamespaceURI();
if (uri == null)
    uri = Constants.NS_URI_SOAP_ENV;
// Use the SOAPPropertyUtils findBOName() method to search
// the SOAPProperty object for this messages first element
// name and namespace. If no match is found, a
// SOAPDataHandlerException will be thrown. If a match is
// found, and it's not an empty string, append to the boname.
String returnedBOName = SOAPPropertyUtils.findBOName(prop, name, uri);
if (returnedBOName != null &&
    !returnedBOName.equals(EMPTY_STRING))
    boName.append(returnedBOName);
return boName.toString()
}
}

```

## 限制

下面一節將討論資料處理常式的限制。

### SOAP style 及 use 準則

SOAP 訊息使用 Web 服務所定義的 style 和 use 來建立的。SOAP 資料處理常式提供表 46 中所顯示的支援層次。

表 46. Style 及 use 準則

Style	Use	定義的組件使用	資料處理常式支援
document	literal	element	全部
document	literal	type	有限制 (請參閱下面說明)
document	encoded	element	無
document	encoded	type	有限制 (請參閱下面說明)
rpc	literal	element	無
rpc	literal	type	全部
rpc	encoded	element	無
rpc	encoded	type	全部

### 組件及組件元素順序

當 SOAP 資料處理常式將 SOAP 訊息轉換成商業物件，且 SOAP 訊息遵循 document/literal/type 或 document/encoded/type 格式時，訊息組件的順序必須依照 WSDL 中所描述的順序。例如您可以參酌下列 WSDL：

```

<operation name="GetQuote"
    style="document" ...>
<input>
    <soap:body parts="Part1 Part2 Part3 Part4" use="literal">
</input>
</operation>

<definitions
    xmlns:stns="(SchemaTNS)"
    xmlns:wtns="(Wsd1TNS)"

```

```

targetNamespace="(Wsd1TNS)">
<schema targetNamespace="(SchemaTNS)"
        elementFormDefault="qualified">
<element name="SimpleElement" type="xsd:int"/>
<element name="CompositElement" type="stns:ComPOSITEType"/>
<complexType name="ComPOSITEType">
<all>
    <element name='elem_a' type="xsd:int"/>
    <element name='elem_b' type="xsd:string"/>
</all>
</complexType>
</schema>

<message...>
<part name='Part1' type="stns:ComPOSITEType"/>
<part name='Part2' type="xsd:int"/>
<part name='Part3' element="stns:SimpleElement"/>
<part name='Part4' element="stns:ComPOSITEElement"/>
</message>
0
</definitions>

```

SOAP 訊息必須遵循組件所定義的順序。於以下的 SOAP 範例中，請注意 Part1 元素要在 Part2、Part3 及 Part4 元素之前。爲了可正確地進行 BO 解析，必須維持此順序。

```

<soapenv:body... xmlns:mns="(MessageNS)"
        xmlns:stns="(SchemaTNS)">
    <stns:elem_a>123</stns:elem_a>
    <stns:elem_b>hello</stns:elem_b>
<soapenc:int>123</soapenc:int>123</soapenc:int>
<stns:SimpleElement>123</stns:SimpleElement>
<stns:ComPOSITEElement>
    <stns:elem_a>123</stns:elem_a>
    <stns:elem_b>hello</stns:elem_b>
</stns:ComPOSITEElement>
</soapenv:body>

```

當 SOAP 訊息遵循 document/literal/type 或 document/encoded/type 格式時，組件元素亦必須遵循該順序。於上例的 Part1 中，elem\_a 標籤必須在 elem\_b 標籤之前。此限制是由資料處理常式的商業物件解析程序來指定的。由於文件樣式的預設商業物件解析會使用第一個元素的主體名稱及名稱空間，所以在此特定要求、回應或錯誤的所有 SOAP 訊息中，這些都必須是相同元素，如此才能在每個案例中解析相同的商業物件。

**註：** 當 SOAP 訊息遵循 document/literal/type 或 document/encoded/type 格式時，元素絕不可爲選用性的。

## XML 限制

下列 XML 結構、特性及表示法不受支援：

- 多維度陣列
- 部分傳輸的陣列
- 內容貧乏的陣列
- 混合的內容
- maxOccurs 大於 1 的序列、群組及選項模型群組元件

---

## 第 6 章 啓用要求處理程序的協同作業

- 『要求處理程序協同作業核對清單』

本章說明啓用協同作業以供要求處理時必須遵循的步驟。協同作業會使用連接器來呼叫 Web 服務。

---

### 要求處理程序協同作業核對清單

使用「商業物件設計程式」來產生商業物件為開發協同作業程序的一部分。您必須執行下列作業 (說明於下面各節中)，以產生協同作業可用來呼叫 Web 服務的商業物件：

1. 從 URL、UDDI 或檔案系統來識別 WSDL 文件。您使用協力廠商工具來執行這項作業--Web 服務連接器沒有提供這項作業的工具。
2. 開啓「商業物件設計程式」，並啓動 WSDL ODA。如需進一步資訊，請參閱第 141 頁的『啓動 WSDL ODA』。
3. 配置 ODA。
4. 確認您的選項。
5. 產生最上層的商業物件，其中包含「要求」以及 (用於同步要求作業)「回應」和「錯誤」商業物件，還有 SOAP Config MO、Protocol Config MO、標頭儲存區和子項物件，及適用於各物件和屬性的應用程式特有資訊。WSDL ODA 會自動執行此處理程序。

在您產生商業物件之後，必須執行作業，讓協同作業能夠使用連接器和 SOAP 資料處理常式來呼叫 Web 服務。如需有關開發協同作業的步驟，包括建立協同作業範本和物件，以及連結其埠，請參閱 *IBM WebSphere InterChange Server 協同作業開發手冊*。欲進一步瞭解如何建立通用商業物件與 WSDL ODA 所產生的應用程式特有商業物件之間的對映，請參閱 *IBM WebSphere InterChange Server 對映開發手冊*。



---

## 第 7 章 將協同作業顯現為 Web 服務

- 『程序核對清單』
- 第 132 頁的『識別或開發商業物件』
- 第 132 頁的『選擇或開發協同作業範本』
- 第 132 頁的『連結新協同作業物件的埠』
- 第 134 頁的『WSDL 配置精靈』
- 第 136 頁的『TLO 格式之商業物件的 WSDL 配置精靈處理程序』
- 第 138 頁的『基本需求及例外的處理』

本章說明將協同作業顯現成 Web 服務的設計時期程序。這樣可於 Web 服務用戶端呼叫協同作業時，讓連接器處理事件。

整合性設計工具簡化了將協同作業顯現成 Web 服務的作業。在為 Web 服務配置協同作業及商業物件之後，您要使用「WSDL 配置精靈」。該精靈會建立 WSDL 文件及 XML 綱目，將協同作業呈現為 Web 服務。WSDL 的輸出內容不僅說明了協同作業，且會構成使 Web 服務用戶端來進行呼叫的基礎。

---

### 程序核對清單

您必須執行下列作業（下面各節中將有詳細說明），以將協同作業顯現成 Web 服務：

1. 識別或（依需要）開發商業物件，以用來作為要求及選擇性的（針對同步的事件處理程序）回應和錯誤 SOAP 訊息。有兩種方法可用來產生這些物件：1) 使用「商業物件設計程式」手動進行，或是 2) 如果您的 Web 服務中有 WSDL 介面檔，您可以使用 WSDL ODA 來產生「要求」及其他（「回應」或「錯誤」）商業物件。如果您遵循第二種方法：
  - a. 在「協同作業 WSDL ODA」配置內容中指定協同作業名稱。此值可指定 TLO 中的 ws\_collab ASI。
  - b. 為 WSDL 介面檔指定 WSDL\_URL 或 UDDI\_InquiryAPI\_URL WSDL ODA 配置內容（如果目錄路徑位於您的網路或本端環境，您也可以將它指定到這個檔案）。

如需進一步資訊，請參閱第 141 頁的『啟動 WSDL ODA』。

2. 開發協同作業範本，或是選擇現有的範本，使用商業物件。
3. 建立協同作業物件及其用於 Web 服務的埠。

您必須先確定協同作業物件已正確地移入商業物件。如需有關建立協同作業物件的詳細資訊及逐步程序，請參閱 *WebSphere InterChange Server 實作手冊*。

**註：**必須針對適當的轉換作業來配置協同作業物件的對映。對映會將在 SOAP 要求訊息中接收的商業物件，轉換成協同作業所使用的商業物件。對映也會將協同作業傳回的商業物件，轉換成內嵌在 SOAP 回應訊息中的商業物件。如需有關對映及對映程序的詳細資訊，請參閱 *對映開發手冊*。

4. 使用「WSDL 配置精靈」來建立 WSDL 文件。公用程式也會配置 Web 服務連接器。

註: 「WSDL 配置精靈」會建立實作方式、介面及一或數個綱目檔。本書將這些輸出值統稱為 WSDL 文件。

5. 依需要來發佈 WSDL 文件。

註: 連接器沒有提供用來發佈 WSDL 文件的工具或支援。

---

## 識別或開發商業物件

您可使用「商業物件設計程式」來建立商業物件，並使用「連接器配置程式」來配置支援那些商業物件的連接器。

如需有關「商業物件設計程式」的詳細資訊，請參閱商業物件設計程式。如需有關 Web 服務商業物件的詳細資訊，請參閱第 19 頁的第 3 章, 『商業物件基本需求』。

---

## 選擇或開發協同作業範本

您所選擇或開發的協同作業範本，必須要有一或數個範例實務來顯現成 Web 服務。如需有關協同作業範本的進一步資訊，*協同作業開發手冊*。

---

## 連結新協同作業物件的埠

在您為商業物件類型配置了協同作業範本的埠之後，必須建立協同作業物件，並將其埠連結到 Web 服務連接器的實例。

若要建立新的協同作業物件，並將其埠連結到 Web 服務連接器的實例，請：

1. 用滑鼠右鍵按一下「協同作業物件」資料夾，並選取「建立新協同作業物件」。這樣會顯示「建立新協同作業」視窗，列出範本清單 (如圖 55 所示)。



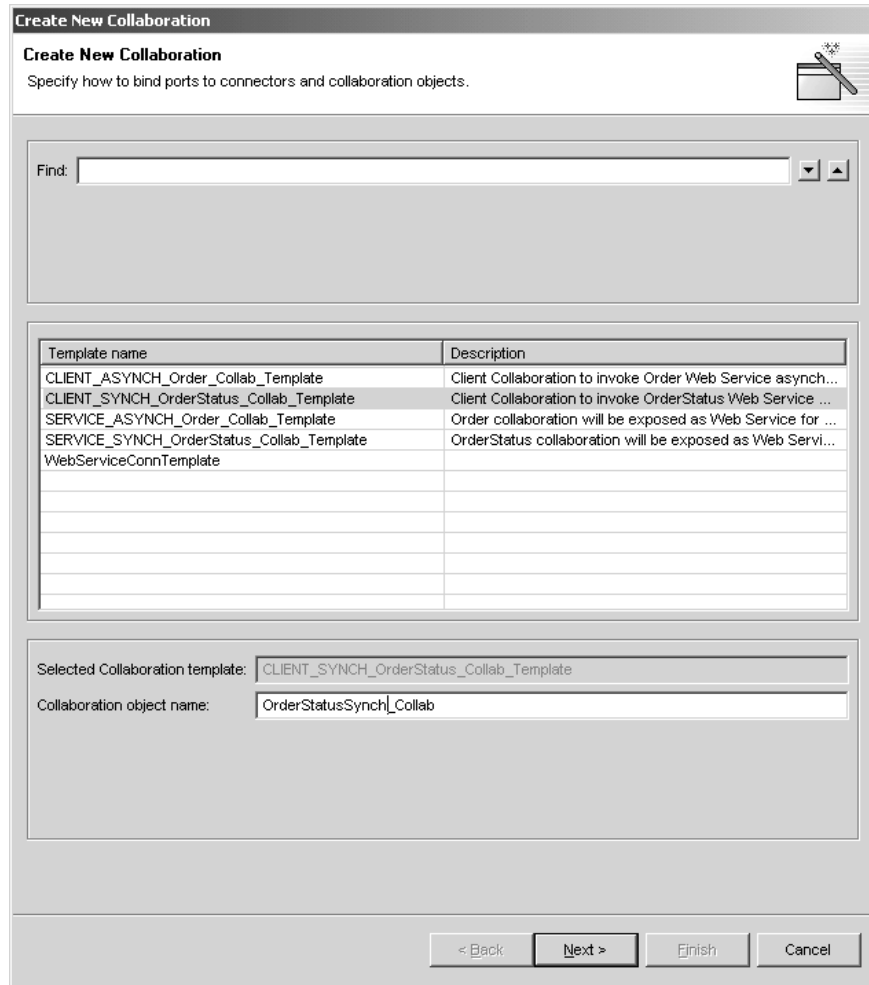


圖 55. 「建立新協同作業」視窗

2. 從「範本名稱」選取協同作業範本，並於「協同作業」物件名稱欄位中輸入協同作業物件的名稱。這樣會顯示「連結埠」視窗，如圖 56 所示。

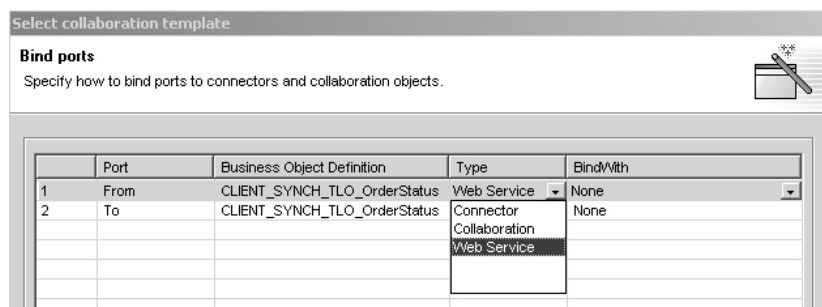


圖 56. 「連結埠」視窗

3. 選取一個埠，按一下「類型」的箭頭來顯示該埠的下拉功能表，並選擇 WebService (如圖 56 所示)。

Web 服務連接器的所有實例皆具有 ConnectorType 應用程式特有內容。依預設，這個內容的設定為 WebService。「系統管理程式」中的「連結協同作業埠」視窗會使用 ConnectorType 內容的值來判斷哪些連接器是 Web 服務連接器。

4. 按一下 BindWith 的箭頭，以顯示連接器實例的清單。「系統管理程式」會顯示 ConnectorType 內容值設定為 WebService 的連接器實例。請選擇一個 Web 服務連接器的實例。(範例顯示於圖 57 中)。

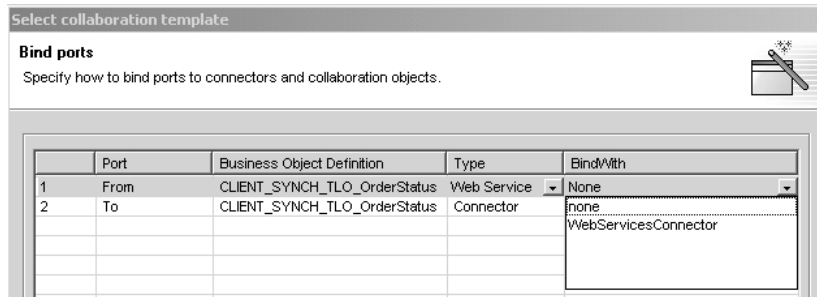


圖 57. 選取 Web 服務連接器的實例

5. 按一下「完成」。

您現在已準備好，可以執行「WSDL 配置精靈」了。

## WSDL 配置精靈

在您建立協同作業物件，並將其觸發埠連結到 Web 服務連接器的實例之後，即已準備好，可以使用「WSDL 配置精靈」了。公用程式會使用您為協同作業、商業物件定義和連接器指定的連結、埠名、作業及其他資料，來產生 WSDL 實作檔案 (\*.impl.wsdl)、WSDL 介面檔 (\*.wsdl) 及 xml 綱目檔 (\*.xsd)。這些檔案是一組顯現成 Web 服務的協同作業。且公用程式可讓您指定是要將這些檔案產生成個別的檔案，還是一個檔案。該公用程式可支援 HTTP、HTTPS 及 JMS 通訊協定上的 SOAP。可從連接器特定的內容 ProtocolListenerFramework 來擷取通訊協定接聽器組織架構的配置資訊。這個內容也使得接聽器清單變成有效的。

### 執行精靈

若要執行「WSDL 配置精靈」，請：

1. 用滑鼠右鍵按一下您為 Web 服務所配置的協同作業物件，並於顯現功能表中選擇「顯現為 Web 服務」。「WSDL 配置精靈」的顯示如圖 58 所示。

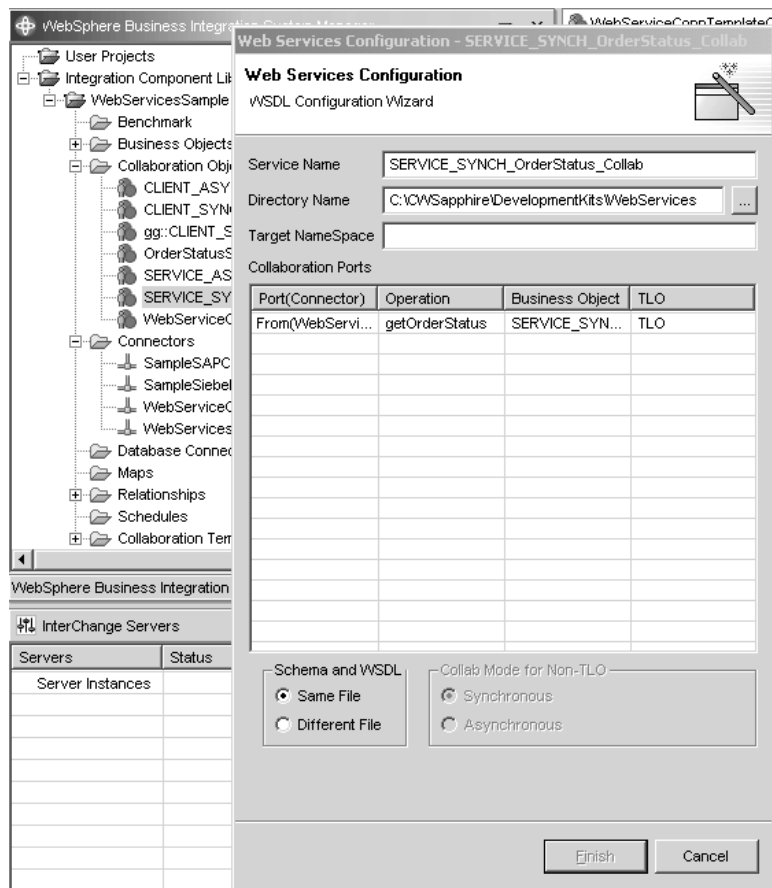


圖 58. WSDL 配置精靈

如圖 58，直欄如下所示：

- **埠 (連接器)**連結至 Web 服務連接器之協同作業物件上的觸發埠。精靈是從協同作業物件來取得此資訊的。
  - **作業**如果商業物件為 TLO，精靈就會從「要求」商業物件的 SOAP Config Mo BodyName 屬性來取得此資訊。如果商業物件是非 TLO，精靈就會將商業物件名稱和埠名合併。
  - **商業物件**用來建立綱目。精靈取得此資訊的來源為這個觸發埠的連接器支援商業物件。
2. 請依需要輸入下列資訊：
    - **服務名稱**依預設，就是您用來說明協同作業物件的名稱。
    - **目錄名稱** Web 服務之配接器，以及協同作業範本和物件所在的地方。
    - **目標名稱空間**被顯現為 Web 服務之協同作業的 URL。
    - **協同作業埠**這些欄位中的資訊，和協同作業物件配置程序的「連結埠」視窗中所指定的一樣。
    - **非 TLO 的協同作業模式**如果您是使用 TLO，則不適用這一項。如果您是使用非 TLO 物件來作為輸入值，則必須指定同步或非同步。
    - **綱目及 WSDL**指定您想要將這些輸出值放在單一檔案中，還是分開的檔案中。
  3. 按一下「完成」。公用程式會根據您所輸入的輸入值及規格來產生輸出項目，下節中會有這些事項的摘要。

## TLO 格式之商業物件的 WSDL 配置精靈處理程序

配置精靈會針對連結至 Web 服務連接器之協同作業物件的每個觸發埠，來建立 WSDL 作業。作業的建立是以與此協同作業呼叫作業相關的商業物件為基礎。

配置精靈會讀取物件層次 ASI `ws_eventtlo`，以判斷商業物件是否為 TLO 格式。如果 ASI 內容的設定為 `true`，則商業物件為 TLO。使用該 TLO，可找到下列 WSDL 內容：

- **作業名稱及 BodyNS**當精靈找到 TLO 格式的商業物件時，就會使用該 TLO 之「SOAP 要求」商業物件中，SOAP Config MO 的 `BodyName` 內容，來建立作業名稱。同樣地，精靈也會判斷訊息名稱空間，作為相同 SOAP Config MO 中的 `BodyNS` 內容。
- **執行模式**精靈會檢查 TLO 之商業物件層次 ASI 中的 `ws_mode` 內容，以判斷模式是同步的還是非同步的，並分別建立 `REQUEST_RESPONSE` 或 `ONE_WAY` WSDL。

若要以 TLO 來建立 WSDL 作業，可以用兩種方式來配置協同作業，就是有對映，和沒有對映。

**有對映的 TLO:** 通常協同作業的配置是可以接受「通用商業物件 (GBO)」要求。意即，協同作業範本觸發埠會訂閱 GBO。若要用此案例的 TLO，則協同作業必須連結到 Web 服務連接器，且連接器必須支援透過對映來將 GBO 轉換成 TLO 的作業。圖 59 顯示此範例實務。

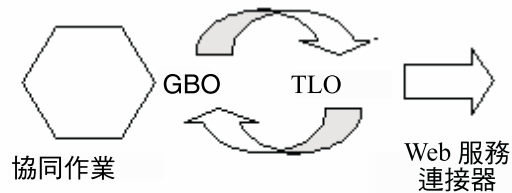


圖 59. 含對映的 TLO

若以這個方式來配置協同作業及連接器，精靈會決定使用 TLO 商業物件，建立 WSDL 文件中所說明的作業。這是在視察過連接器支援的商業物件及相關對映後，所做出的決定。對 Web 服務連接器的執行時期處理程序而言，所配置的對映必須將協同作業的 GBO 轉換成一個且只能有一個 TLO，這是很重要的。此外，入埠對映的來源和目的地商業物件，要分別轉換成離埠對映的目的地和來源商業物件，這也是很重要的。

**沒有對映的 TLO:** 精靈亦支援處理沒有對映的 TLO。在此情況下，協同作業範本的觸發埠會直接訂閱 TLO。因為 Web 服務連接器支援 TLO，所以不需要對映。圖 60 說明此範例實務。



圖 60. 沒有對映的 TLO

若是以這個方式來配置協同作業及連接器，精靈就會使用在協同作業中找到的 TLO 商業物件，來建立 WSDL 文件中所說明的作業。精靈會決定不為此埠配置對映。

## 非 TLO 格式之商業物件的 WSDL 配置精靈處理程序

對非 TLO 商業物件的支援，可讓您使用預先存在的協同作業和對映，來顯現成 Web 服務。因此，精靈亦可支援用非 TLO 格式的商業物件來建立 WSDL 作業。

和 TLO 處理程序一樣，精靈會讀取物件層次 ASI ws\_eventtlo，以判斷商業物件是否為非 TLO 格式。若 ASI 內容不存在，或是雖然存在，但不是設定為 true，則此商業物件為非 TLO。非 TLO 是不屬於 Web 服務 TLO 結構的任何商業物件。若使用非 TLO，精靈會探索下列內容：

- **作業名稱及 BodyNS** 當精靈找到非 TLO 格式的商業物件時，它會使用協同作業名稱、商業物件名稱與埠名的組合，來建立作業名稱。WSDL 作業的「主體名稱空間」是用「WSDL 配置精靈」中的「目標名稱空間」項目來配置的。
- **WSCollaborations** 精靈會在 Web 服務連接器中建立內容階層，包括顯現成 Web 服務的協同作業埠中，每個 WSDL 作業的「BO 名稱」、「SOAP 主體名稱」、「SOAP 主體名稱空間」及「模式」。圖 61 顯示 WSCollaborations 內容的範例：

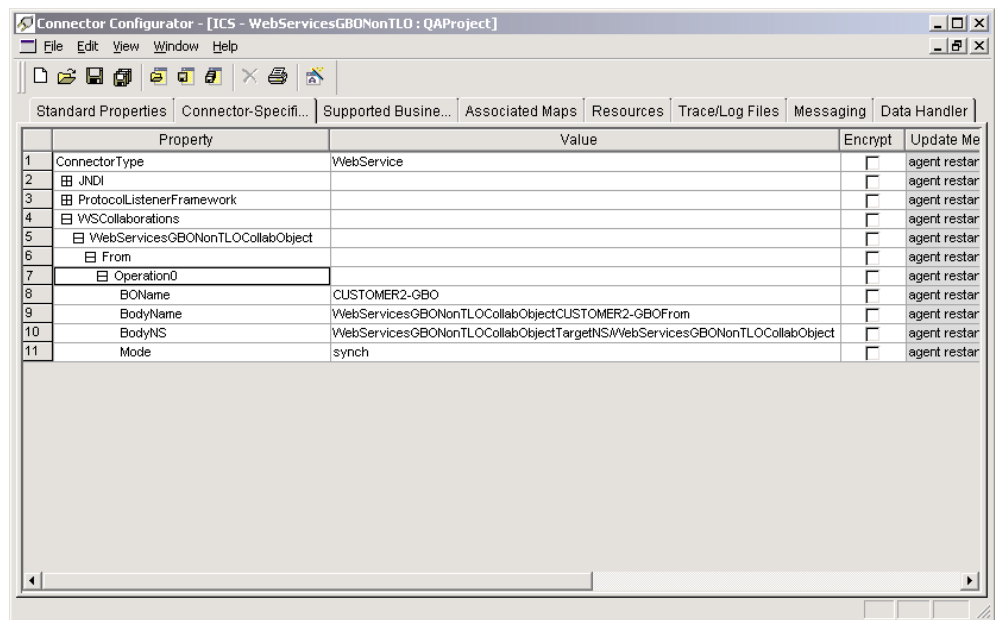


圖 61. WSCollaborations

- **執行模式** WSDL 作業的「執行模式」是用「WSDL 配置精靈」中，「非 TLO」選項按鈕的「協同作業模式」來配置的。

若要以非 TLO 來建立 WSDL 作業，可以用兩種方式來配置協同作業，就是有對映，和沒有對映。

**有對映的非 TLO:** 通常協同作業的配置是可以接受「通用商業物件 (GBO)」要求。同時，其中可能會有預先存在的對映，可將 GBO 從協同作業轉換成非 TLO 商業物件。圖 62 顯示此範例實務。



圖 62. 有對映的非 TLO

於此情況下，精靈會使用非 TLO 商業物件來建立 WSDL 文件中所描述的 WSDL 作業。對 Web 服務連接器的執行時期處理程序而言，所配置的對映必須將協同作業的 GBO 轉換成一個且只能有一個非 TLO，這是很重要的。此外，入埠對映的來源和目的地商業物件，要正確地分別轉換成離埠對映的目的地和來源商業物件，這也是很重要的。

**沒有對映的非 TLO:** 於高度特殊化的案例中，可將協同作業配置成可接受不是來自 GBO 的商業物件之要求。在此情況下，非 TLO 是協同作業的直接商業物件，而且沒有對映存在。圖 63 顯示此範例實務。

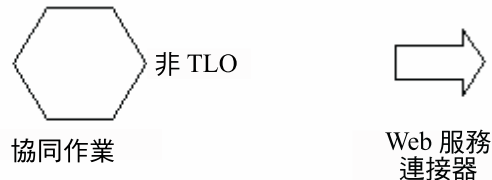


圖 63. 沒有對映的非 TLO

於此情況下，精靈會決定不為此埠配置對映，所以它會使用非 TLO 商業物件來建立 WSDL 文件中所描述的 WSDL 作業。

### 基本需求及例外的處理

除非另有明確指示，否則下面一節所討論的「WSDL 配置精靈」基本需求，適用於所有類型的物件 (TLO 及非 TLO)。如需有關 Web 服務 TLO 之商業物件基本需求的進一步資訊，請參閱第 19 頁的第 3 章，『商業物件基本需求』。

**註:** 在 WSDL 工具所讀取的商業物件 ASI 當中，只有下列各項可以含有國際化字元：

- elem\_name
- elem\_ns
- attr\_name
- attr\_ns
- BodyName
- BodyNS
- type\_name
- type\_ns

**支援 SOAP Config MO 中的 Use 內容:** 「WSDL 配置精靈」可支援 SOAP Config MO 中的 Use 內容，但是如果「SOAP 要求 BO」與對應之「SOAP 回應 BO」中的 Use 值不同，就會擲出錯誤。您可以將 Use 值設定成文字或編碼，以產生 WSDL 文件。如需 Use 內容及其值的相關資訊，請參閱第 100 頁的『Style 及 Use 對 SOAP 訊息的影響』。

**支援 SOAP Config MO 中的 Style:** 就顯現為 Web 服務的協同作業而言，僅支援 rpc 樣式。若於 SOAP Config MO 中將 Style 指定為 document，精靈就會擲出錯誤。

**錯誤處理程序:** 「SOAP 錯誤」商業物件中的 details 屬性只能有一個子屬性。否則公用程式會產生錯誤。

公用程式可接受「錯誤」商業物件。若發現多重「錯誤」商業物件，公用程式會處理第一個標頭儲存區，或是預設的錯誤商業物件。處理程序如下：

- 於 binding 區段中，沒有為 soap:fault 元素指定「名稱空間」。
- 「錯誤」一律以 document style 和 use literal 來指定。
- 訊息組件是以元素屬性來指定。

**標頭錯誤處理程序:** 標頭錯誤會被處理成 soap:headerfault (WSDL 文件 binding 區段中之 soap:header 的子元素)。標頭錯誤是以標頭子項商業物件中的 headerfault ASI 來處理的，如下所示：

- 沒有為 soap:headerfault 元素指定「名稱空間」。
- 標頭錯誤一律以 document style 和 use literal 來指定。
- 訊息組件是以元素屬性來指定，而不是以類型屬性來指定。

**標頭處理程序:** 多重標頭屬性會被指定為 SOAP 標頭儲存區商業物件中的 SOAP 標頭子項商業物件。「標頭」儲存區商業物件是以其 ASI 來識別的：

soap\_location=SOAPHeader。於公用程式處理期間，會針對標頭儲存區商業物件中的每項屬性，在 binding 區段中各建立一個 soap:header 元素，並套用下列規則：

- 標頭一律以 document style 和 use literal 來指定。
- 訊息組件是以元素屬性來指定，而不是以類型屬性來指定。
- 如果沒有指定 elem\_ns，就會將標頭寫入「主體名稱空間」。

**註:** 標頭儲存區商業物件可為 SOAP 要求、回應或錯誤商業物件的子項。並沒有為 soap:header 元素指定名稱空間屬性。

**elem\_ns ASI 處理程序:** 公用程式會忽略訊息組件層次的 elem\_ns ASI。但是會將 elem\_ns 用在第二層次以下的屬性。如果有指定 elem\_ns，則第二層次的商業物件屬性可定義在分開的名稱空間中。

**JMS 通訊協定處理程序:** WSDL 文件之 port 區段中的 SOAP/JMS 連結，包含 jms:address 元素。以下為 jms:address 元素的範例：(字尾有 "?" 的屬性是選用屬性。)

```
<jms:address
    destinationStyle = "queue"
    jmsVendorURI = "http://ibm.com/ns/mqseries"?
    initialContextFactory = "com.ibm.NamingFactory"?
    jndiProviderURL = "iiop://something:900/wherever"?
    jndiConnectionFactoryName = "orange"
    jndiDestinationName = "fred"
    jmsProviderDestinationName="trash" />
```

若 `LookupQueuesUsingJNDI` 連接器內容的設定為 `true`，則 `InputQueue` 內容的值對應於 SOAP/JMS 連結之 `jms:address` 元素的 `jndiDestinationName` 屬性。 `jms:address` 元素指定於 `wsdl:port` 區段中。若 `LookupQueueUsingJNDI` 的設定為 `false`，則 `jmsProviderDestinationName` 屬性的設定為 `InputQueue`。 `InputQueue` 是 `Listener_JMS` 階層式內容下的可用連接器內容。 `initialContextFactory`、`jndiProviderURL` 及 `jndiConnectionFactoryName` 內容僅限指定進行同步處理程序。

**HTTP 通訊協定處理程序：** WSDL 文件中的範例 `port` 區段，如下所示：

```
<service name="StockQuoteWebService">
<port name="StockQuoteWebServicePort" binding="intf:StockQuoteBinding">
<soap:address location="http://localhost:8080/wbia/webservices/stockquoteservice"/>
</port>
</service>
```

「WSDL 配置精靈」會使用主機名稱的值，以及環境定義路徑的埠。若環境定義路徑僅包含相對路徑，但沒有主機名稱及埠，就會使用主機名稱的值，及位於 `Listener_HTTP` 配置內容之下的埠內容，來指定 `soap:address` xml 元素中的位置屬性。



---

## 第 8 章 使用 WSDL ODA

- 『啓動 WSDL ODA』
- 第 142 頁的『執行 WSDL ODA』
- 第 142 頁的『配置代理程式』
- 第 144 頁的『指定 WSDL 文件』
- 第 146 頁的『確認選項』
- 第 146 頁的『產生物件』
- 第 147 頁的『限制』

**註：**「Web 服務說明語言 (WSDL) 物件探查代理程式 (ODA)」可用來產生要求處理程序的商業物件，以及在可以使用「WSDL 介面」檔案時用於事件處理。協同作業會使用連接器來呼叫 Web 服務。或是您也可以將協同作業顯現為 Web 服務。而 Web 服務則是使用 WSDL (Web 服務說明語言) 來加以說明。本章說明如何使用「Web 服務說明語言 (WSDL) 物件探查代理程式 (ODA)」來產生商業物件。當協同作業呼叫 Web 服務，以及將協同作業顯現為 Web 服務時，連接器及 SOAP 資料處理常式就會使用這些商業物件。

您可以針對下列兩個目的，使用 WSDL ODA 來產生商業物件：

1. WSDL ODA 可以採用 WSDL 實作檔案，並為協同作業產生商業物件，以呼叫外部 Web 服務。
2. WSDL ODA 可以採用 WSDL 介面檔，並為協同作業產生顯現為 Web 服務的商業物件。

當您使用「商業物件設計程式」時，可以啓動 WSDL ODA。WSDL ODA 會讀取 WSDL 文件，並建立連接器及 SOAP 資料處理常式所需的商業物件。WSDL ODA 簡化了開發商業物件的工作。

**註：**WSDL ODA 會處理 WSDL 中的 SOAP/HTTP 及 SOAP/JMS 連結。

---

### 啓動 WSDL ODA

您可使用下列其中一個 Script 來啓動 WSDL ODA：

- Windows
  - start\_WSDLODA.bat

**註：**您亦可使用「安裝程式」自動為 Windows 環境建立的捷徑，來啓動 WSDL ODA。

- UNIX
  - start\_WSDLODA.sh

請使用「商業物件設計程式」來選取、配置及執行 WSDL ODA。「商業物件設計程式」會依照各 Script 或批次檔之 AGENTNAME 變數中所指定的名稱，來尋找每個 ODA。

## 執行 WSDL ODA

「物件探查代理程式 (ODA)」簡化了為要求處理程序建立商業物件的工作。「商業物件設計程式」提供所有可用 ODA 的圖形式介面，並可協助您尋找您所需要的代理程式。依預設，WSDL ODA 的名稱為 WSDL0DA。該名稱出現於「WSDL 精靈」中的樣子，需視 start\_WSDL0DA.bat 或 start\_WSDL0DA.sh 檔案中的 AGENTNAME 變數值而定。欲進一步瞭解 ODA 和商業物件定義，及如何配置、啟動和使用 ODA，請參閱 *IBM WebSphere Business Object 開發手冊*。當您進行下列程序時，建議您依需要來參考文件。

啟動「物件探查代理程式」之後，請遵循下列步驟來啟動 WSDL ODA：

1. 開啟「商業物件設計程式」。
2. 從「檔案」功能表選取「使用 ODA 開啓新檔案...」子功能表。「商業物件設計程式」會在「商業物件精靈」中顯示「選取代理程式」對話框。圖 64 說明此視窗。
3. 按一下「尋找代理程式」按鈕，以顯示所有執行中的代理程式，並選取 WSDL ODA。

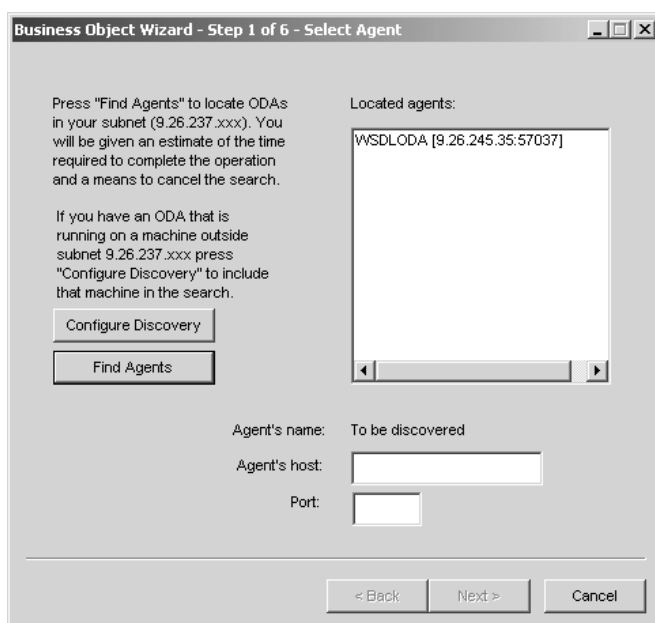


圖 64. 「選取代理程式」視窗

若「商業物件設計程式」找不到您的 WSDL ODA，請檢查 ODA 的設定。

4. 於「找到的代理程式」窗格清單中選取 WSDL ODA，並按「下一步」。

如此會顯示「配置代理程式」精靈視窗，其中顯示您需要指定的配置內容。

## 配置代理程式

圖 65 顯示「WSDL ODA 商業物件精靈」的「配置代理程式」視窗。

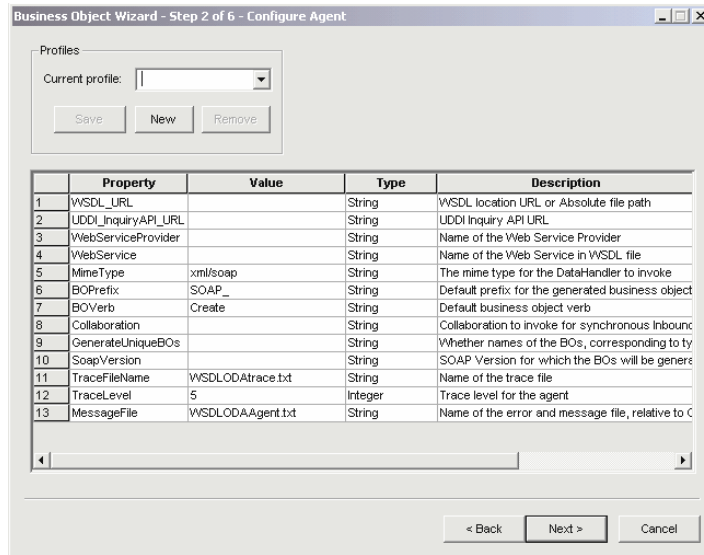


圖 65. 「配置代理程式」視窗

表 47 列出您必須為 WSDL ODA 配置的內容。

**註:** 當您第一次使用 WSDL ODA 時，必須指定每項配置內容的值。指定好之後，可以按一下「儲存」按鈕，將內容值儲存於設定檔中。當您下次使用 WSDL ODA 時，就可從「選取設定檔」方塊中選取所儲存的設定檔。

表 47. WSDL ODA 配置內容

內容	類型	必要的	預設值	說明
WSDL_URL	String	是的，若不是指定 UDDI 的話	無	WSDL 文件的 URL。此值亦可設定為本端 WSDL 檔案的絕對路徑。您可使用當地的語言來指定 URL。
UDDI_InquiryAPI_URL	String	是的，適用於 UDDI	無	UDDI 查詢 API 的 URL。
WebServiceProvider	String	是的，適用於 UDDI	無	目標 Web 服務提供者的名稱。這通常和發佈在 UDDI 登錄上的「企業」名稱相同。此項目需區分大小寫，而且只能用英文字元。
WebService	String	是的，適用於 UDDI		Web 服務的名稱。此項目需區分大小寫，而且只能用英文字元。
MimeType	String	否	xml/soap	連接器所呼叫之資料處理常式的 MIME 類型。此值設定於商業物件 TLO 中，為預設值，且只能用英文字元。

表 47. WSDL ODA 配置內容 (繼續)

內容	類型	必要的	預設值	說明
BOPrefix	String	否	SOAP_	這要附加於所建立之每個商業物件的前面。可由使用者配置 (僅限英文字元)，最多 8 個字元。
BOVerb	String	是	建立	在「要求」及選用性的「回應」和「錯誤」商業物件的 SOAP Config MO 上設定的動詞。
協同作業	String	否	無	此值可指定 TLO 中的 ws_collab ASI，且在產生物件以用於事件處理時，此值也是必要的。
GenerateUniqueBOs	String	否	true	如果這個內容為 true，則商業物件名稱在所有 Web 服務之間就會是唯一的。如果這個內容為 false，您就可以在具有相同組件類型的作業之間重複使用商業物件。
SOAPVersion	String	否	1.1	決定用來產生 BO 的 SOAP 標準。可能的值是 1.1 及 1.2。

下一節將說明如何在「配置代理程式」中指定 WSDL 文件。

## 指定 WSDL 文件

Web 服務商業物件是從 WSDL 文件來產生的。本節將告訴您如何在 ODA 的「配置代理程式」中，選取及指定 WSDL 文件的來源。

WSDL 文件可能位於本端檔案系統上，或是在網路上的 URL 位置，或是在 UDDI 登錄中 --您要指定 WSDL 文件的所在位置，以及擷取該文件的 WSDL ODA。(完整的 WSDL 服務說明可能包含多份文件。)

### 從 URL 位置取得 WSDL 文件

如上圖 65 所示：

1. 在配置內容 WSDL\_URL 中指定 WSDL 文件的 URL

然後 ODA 會解析所匯入之文件的 URL，從 WSDL 文件擷取 Web 服務的清單。WSDL\_URL 內容亦可讓您使用 URL 語法，指定本端檔案系統上的 WSDL 檔案位置 (例如：file://C:/test/wsdl)，或絕對路徑 (例如：C:\test\wsdl)。您必須確定 ODA 可以存取此文件及其相依項 (所有匯入的文件)。

2. 按「下一步」。

ODA 會查詢 Web 服務提供者的 URL，並擷取此 URL 位置之 WSDL 中定義的服務清單，然後顯示該清單，如圖 66 所示。

註: WSDL ODA 顯示只包含 SOAP/JMS 或 SOAP/HTTP 連結的埠，並排除其他類型的連結。

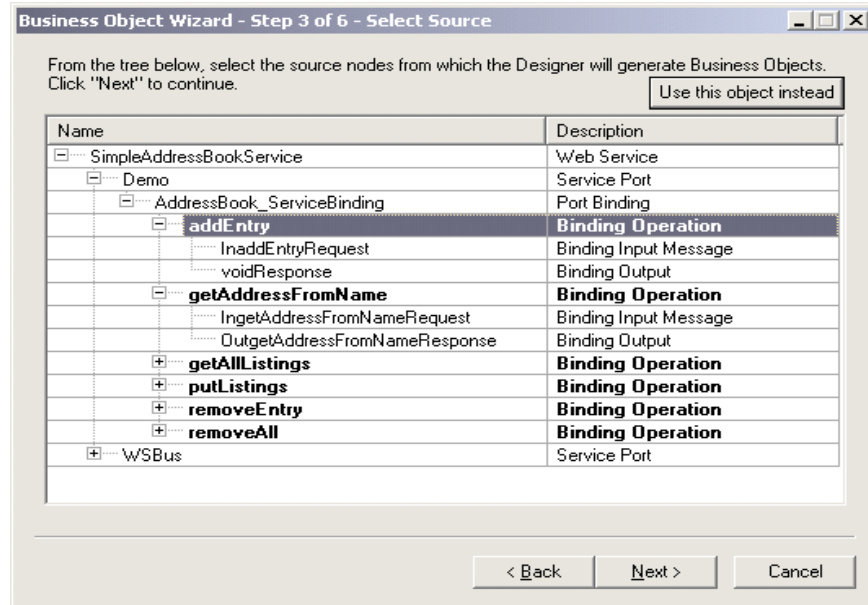


圖 66. 「選取來源」視窗

3. 從埠清單選取一個 (且只能一個) 作業 (可以選取的作業會強調顯示)。您不能選取服務或埠節點，這些節點僅供顯示。請注意，WSDL 作業的類型如下：  
ONE\_WAY、REQUEST\_RESPONSE、  
SOLICIT\_RESPONSE 及 NOTIFICATION。WSDL ODA 只能支援及顯示  
REQUEST\_RESPONSE 和 ONE\_WAY 作業。
4. 按「下一步」，並跳至第 146 頁的『確認選項』。

## 從 UDDI 登錄取得 WSDL 文件

ODA 亦可從 UDDI 登錄來擷取 WSDL 文件，不需從 URL 位置。若要這麼做，請：

1. 於「配置代理程式」視窗中，針對您的“search key”來指定下列內容：
  - UDDI\_InquiryAPI\_URL (例如：<https://uddi.ibm.com/ubr/inquiryapi>)
  - WebServiceProvider (例如：IBM Corporation)
  - WebService (例如：StockQuoteService)
  - WSDL ODA 會使用完全相符的名稱比對方法 (findQualifier)，在 UDDI 登錄中查詢。請確定您輸入的參數值是正確的。您可使用一般的 UDDI 瀏覽器來尋找服務提供者所提供的服務。

WSDL ODA 會使用表 47 中所描述的這些內容來連接 UDDI 登錄。

2. 按「下一步」。

ODA 會查詢 Web 服務提供者的 UDDI 登錄，並擷取符合您所指定之 Web 服務參數的服務清單。WSDL ODA 會在像圖 66 的視窗中，顯示 Web 服務提供者所提供的服務清單。當 UDDI 查詢傳回多個相符項時，WSDL ODA 在顯示這些項目時，會加上底線 ( ) 和序號。例如：StockQuoteService\_1、StockQuoteService\_2 等等。

註: WSDL ODA 顯示只包含 SOAP/JMS 或 SOAP/HTTP 連結的埠。

3. 從埠清單選取一個 (且只能一個) 作業。您不能選取服務或埠節點，這些節點僅供顯示。請注意，WSDL 作業的類型如下：ONE\_WAY、REQUEST\_RESPONSE、SOLICIT\_RESPONSE 及 NOTIFICATION。WSDL ODA 只能支援及顯示 REQUEST\_RESPONSE 和 ONE\_WAY 作業。
4. 按「下一步」，並跳至『確認選項』。

註: 連接器僅支援 UDDI 第 2 版 API。因此，您不能從不支援 UDDI 第 2 版的 UDDI 登錄來擷取 WSDL。

## 確認選項

選取好 Web 服務作業來源之後，「WSDL ODA 商業物件精靈」會顯示一個如圖 67 的確認畫面：

1. 確認您的選項。
2. 按「下一步」，並跳至『產生物件』。

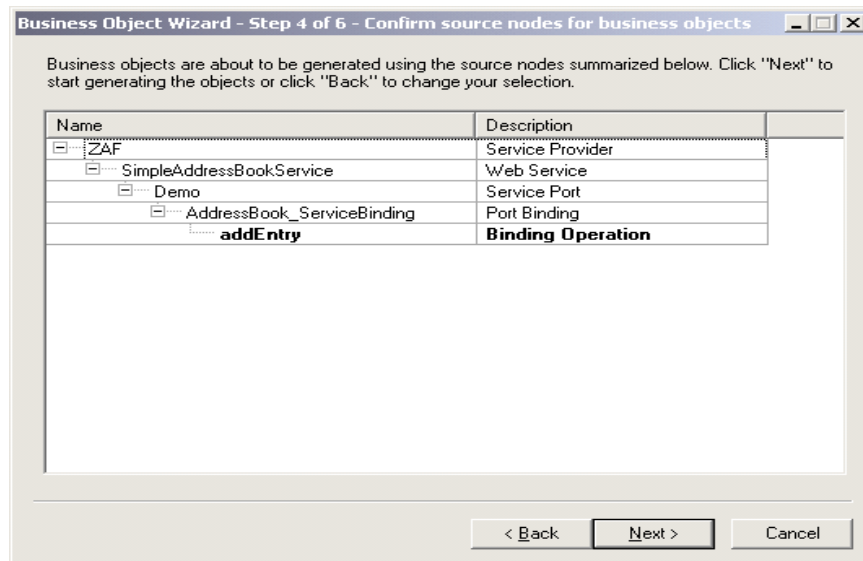


圖 67. 「確認」視窗

## 產生物件

在您確認 WSDL 文件來源之後，WSDL ODA 會為您要呼叫的 Web 服務，或要顯現為 Web 服務的協同作業，產生商業物件及 Meta 物件。

註: WSDL ODA 不會自動選取最上層商業物件的索引鍵屬性。至於所有其他層次的商業物件，WSDL ODA 會將第一個屬性設為索引鍵。因此，當您在「商業物件設計程式」中儲存 WSDL ODA 產生的物件時，會出現錯誤訊息，通知您最上層物件缺少索引鍵屬性。指派索引鍵屬性來反映您的商業物件資料和商業物件基本需求，然後重新儲存物件。選取索引鍵屬性時請小心；該屬性會用在事件序列中，若不小心選取，可能會導致效能問題。

1. 檢查「將商業物件儲存至檔案」，或檢查「用個別的視窗開啓商業物件」。後者會啓動「商業物件設計程式」，並於該應用程式中開啓商業物件。
2. 檢查「關閉 ODA」，並按一下「完成」。

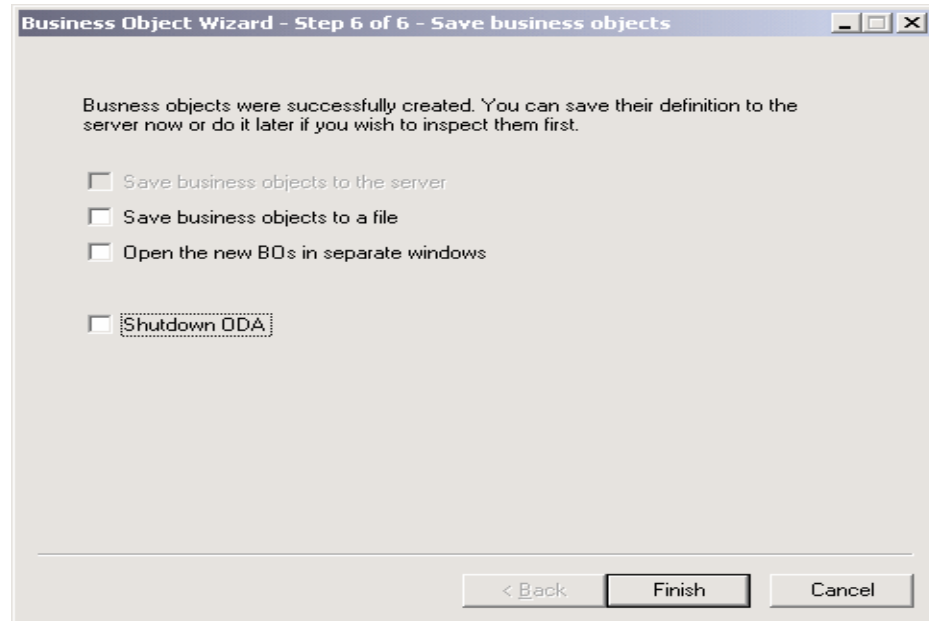


圖 68. 「儲存」視窗

若為要求處理程序，則對 Web 服務的呼叫中必須有要求，而且若是同步的，還要有回應和錯誤訊息。若為事件處理程序，則顯現的協同作業必須有要求，而且若是同步的，還要有回應和錯誤訊息。WSDL ODA 會針對下列每一項來產生商業物件：包括每個層次的應用程式特有資訊 (ASI)，以及 SOAP 資料處理常式和 Protocol Config MO。WSDL 文件中的 SOAP 連結會決定 SOAP 訊息的結構。如需商業物件結構的相關資訊，請參閱第 19 頁的第 3 章，『商業物件基本需求』。

## 限制

表 48 說明 WSDL ODA 支援使用類型和元素之各種組合的屬性 style、use 和 part 定義。

表 48. WSDL ODA 限制

用來定義 Style/Use/Parts 的內容	說明
rpc/encoded/type	支援
rpc/encoded/element	支援
rpc/literal/type	支援
rpc/literal/element	支援
doc/encoded/type	不支援
doc/encoded/element	不支援
doc/literal/type	支援
doc/literal/element	支援

WSDL ODA 可以擷取完全獨立自足 (在一個檔案中) 的 WSDL 檔案，或是分開的檔案，包括一個包含服務元素的實作檔、一個包含所有其他 WSDL 元素 (包括：types、messages、portTypes 及 bindings) 的介面檔，以及一或數個適用於網目的檔案。WSDL ODA 無法順利地擷取含有多個介面檔的 WSDL 檔案，例如，一個介面檔包含 messages 和 portTypes，另一個介面檔包含 bindings。

依據名稱空間字首，WSDL 文件中的綱目必須是獨立自足的。您不能將 WSDL 文件的 <definitions>/<types> 元素所定義的名稱空間字首，用在 <types> 元素的子元素 <schema> 中。若名稱空間字首用於 <schema> 元素的子元素中，您必須在 <schema> 元素中，重新定義名稱空間字首。以下是非自行包含之綱目的範例：

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:NS="NS">
  <types>
    <schema xmlns="http://www.w3.org/1999/XMLSchema">
      <element name="NSElem" type="NS:NSType"/>
    </schema>
  </types>
</definitions>
```

名稱空間字首 NS 定義於 <definitions> 元素中，並且會用於 <schema> 元素中，而且不會重新定義。所以 WSDL ODA 會擲出錯誤。若要解除這個限制，請重新定義 <schema> 元素的名稱空間字首 NS，如下所示：

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:NS="NS">
  <types>
    <schema xmlns="http://www.w3.org/1999/XMLSchema" xmlns:NS="NS">
      <element name="NSElem" type="NS:NSType"/>
    </schema>
  </types>
</definitions>
```



---

## 第 9 章 疑難排解

本章說明您於啟動或執行連接器時可能遇到的問題。

---

### 啓動問題

---

#### 問題

演算法不受支援 / 演算法 'SSL' 不可使用

#### 可能的解決方案 / 說明

當您的 JSSE 提供者不支援連接器配置程式中所指定的 SSL 版本時，就會發生這個錯誤。解決方案：檢查 JSSE 提供者的文件，查詢其所支援的 SSL 版本。若使用 IBM JSSE，請確定 *ProductDir/lib/security* 目錄中的 *java.security* 檔案具有下列項目：

```
security.provider.<number>=com.ibm.jsse.  
IBMJSSEProvider
```

載入金鑰儲存庫時，發生錯誤：所指定的金鑰儲存庫檔路徑： "*<path>*" 不正確：找不到 KeyStore

其中 *<number>* 是載入安全提供者的偏好順序。若您指定的金鑰儲存庫及/或信任儲存庫檔案的路徑不正確，就會發生這個錯誤。解決方案：請於「連接器配置程式」中，檢查 *SSL->KeyStore* 內容所指定的金鑰儲存庫檔路徑。另外，若您使用了信任儲存庫，請於「連接器配置程式」中，檢查 *SSL->TrustStore* 內容所指定的信任儲存庫檔路徑。若您的金鑰儲存庫及/或信任儲存庫遭竄改或損壞，就會發生這個錯誤。若您所指定的密碼值不正確，也會發生這個錯誤。解決方案：確定金鑰儲存庫沒有被竄改。請嘗試重建金鑰儲存庫。另請確定您於 *SSL->KeyStorePassword* 和 *SSL->TrustStorePassword* 連接器內容中輸入的密碼是正確的。若您的憑證及/或金鑰儲存庫、信任儲存庫遭竄改，就會發生這個錯誤。若您所指定的密碼值不正確，也會發生這個錯誤。解決方案：請檢查並查看憑證、金鑰儲存庫或信任儲存庫是否遭竄改。另請確定您於 *SSL->KeyStorePassword* 和 *SSL->TruststorePassword* 連接器內容中指定的密碼是正確的。若 *SOAP/HTTP* 或 *SOAP/HTTPS* 通訊協定接聽器無法連結到連接器內容中所指定的埠，就會發生這個錯誤。解決方案：請檢查指定給所有 *SOAP/HTTP* 及 *SOAP/HTTPS* 通訊協定接聽器的埠。若是將同一個埠指定給多個接聽器，就只有一個接聽器會啟動。另外，請檢查您是否有任何其他服務程式也在該埠上執行。若有，請為通訊協定接聽器選擇其他埠。

KeyManagementError: KeyStore 遭修改，發生 KeyManagement 錯誤

從金鑰儲存庫載入憑證時，發生錯誤。

建立伺服器 Socket 時，發生錯誤，正在終止：錯誤

KeyManagementError:UnrecoverableKeyException，無法回復金鑰

若無法使用金鑰儲存庫或信任儲存庫，就會發生這個錯誤。解決方案：建立新的金鑰儲存庫。若您的信任儲存庫中沒有 CA 憑證，就會發生這個錯誤。解決方案：檢查 CA 的憑證及其自行簽章的憑證是否在信任儲存庫中。另請確定憑證的 DN 具有主機名稱 (最好是 IP 位址)。

SSL 訊息交換異常狀況：不明的 CA

您發現日誌檔中的 JSSE 記載過多。

若您不想在您的主控台上看到所有基礎 JSSE 的詳細資料，請將連接器配置程式中的 *SSL->SSLDebug* 內容值設定為 *false*。

---

## 問題

您已指定通訊協定接聽器，但是接聽器沒有起始設定；您在連接器中看到下列警告訊息：

```
Skipping Protocol Listener Property Set
  "SOME_LISTENER_NAME" with protocol property "":
  unable to determine the protocol listener
  class.]
```

您已指定通訊協定處理常式，但是尚未起始設定；您在連接器中看到下列警告訊息。

```
Unable to determine the type of the
handler; skipping initializing of current
handler. Handler property details:
Name: <Handler Name>;
Value:
  Name: Protocol; Value:
  Name: ResponseWaitTimeout; Value:
  Name: ReplyToQueue; Value: .]
```

```
java.lang.NoClassDefFoundError:
Javax/jms/JMSEException...
```

```
Fail to lookup, queue: "InProgressQueue"
for specified queue name: "<queue name>"
queue using JNDI "<queue name>"
javax.naming.NameNotFoundException:
<queue name>
```

起始設定時，發生錯誤，沒有起始設定「JNDI 環境定義」，使用者無法使用 JMS 通訊協定。

取得起始環境定義時，發生錯誤

---

## 可能的解決方案 / 說明

連接器無法為通訊協定接聽器的「通訊協定」內容擷取有效的值。有效值為：soap/http、soap/https 或 soap/jms。解決方案：這並不是錯誤狀況。但是若您想要讓連接器使用此接聽器，請指定有效的「通訊協定」內容值。

連接器無法為處理常式的「通訊協定」內容擷取有效的值。有效值為：soap/http、soap/https 或 soap/jms。解決方案：這並不是錯誤狀況。但是若您想要讓連接器使用此處理常式，請指定有效的「通訊協定」內容值。

連接器找不到 jms.jar。解決方案：確定 jms.jar 在連接器的類別路徑中。

如果您將 SOAP/JMS Web 服務用於連接器，若您未建立佇列，就會發生這個問題。如果您將

JNDI->LookupQueuesUsingJNDI 設定為 true，且連接器無法使用 JNDI 來查閱佇列，也會發生這個錯誤。解決方案：建立連接器所需的佇列。若 JNDI->LookupQueuesUsingJNDI 的設定為 true，請確定可使用 JNDI 來查閱連接器所需的佇列。若您將連接器配置來使用 SOAP/JMS 通訊協定接聽器或 SOAP/JMS 通訊協定處理常式，則必須指定 JNDI 內容。解決方案：確定您已指定必要的 JNDI 連接器特有內容。請參閱您的 JNDI 提供者文件，以判斷連接 JNDI 提供者所需的程式庫和 Jar 檔。確定所有必要的 Jar 檔皆於連接器的類別路徑中。另請確定所有必要的程式庫都在連接器的路徑中。若您將連接器配置來使用 SOAP/JMS 通訊協定接聽器或 SOAP/JMS 通訊協定處理常式，則必須指定 JNDI 內容。如果您所指定的 JNDI 內容不正確，也會發生這個錯誤。解決方案：檢查 JNDI 內容。確定 JNDI 的配置正確。請參閱您的 JNDI 提供者文件，以判斷連接 JNDI 提供者所需的程式庫和 Jar 檔。確定所有必要的 Jar 檔皆於連接器的類別路徑中。另請確定所有必要的程式庫都在連接器的路徑中。

---

---

## 執行時期錯誤

---

### 問題

剖析 HTTP 回應時，發生錯誤：讀取 HTTP 回應標頭時，已達串流結尾

所提供的 URL 發生錯誤，無法擷取主機和埠的詳細資料，目的地 <destination URL> 錯誤

---

### 可能的解決方案 / 說明

這個錯誤會於連接器呼叫 SOAP/HTTP Web 服務時發生。會發生這個錯誤，是因為您的目標 Web 服務傳送了不正確的 HTTP 回應。解決方案：確定您的目標 SOAP/HTTP Web 服務端點位址是正確的。

這個錯誤會在連接器呼叫 SOAP/HTTP Web 服務時發生。會發生這個錯誤，是因為您指定給 SOAP/HTTP Web 服務的端點位址不正確。解決方案：確定您指定給 Web 服務的端點位址是正確的。

---

---

**問題**

無法將動詞為 <Verb> 的事件商業物件 <BO Name> 傳送給分配管理系統失敗。收到執行狀態 "-1" 及錯誤訊息：

```
MapException: Unable to find the map to map  
business objects <BO Name> for the connector  
controller WebServicesConnector
```

無法將 SOAP 要求轉換成「要求」商業物件。 Soap 錯誤：

```
Failure in generating request object -  
no verb could be set on the request bo
```

**可能的解決方案 / 說明**

會發生這個錯誤，是因為連接器以同步方式來將事件傳送到協同作業，但該協同作業不存在，或是不接受商業物件動詞，而導致整合分配管理系統無法處理事件。解決方案：若您正使用 Web 服務 TLO 來進行事件通知，請查驗 TLO 的 ws\_collab 物件層次 ASI。(錯誤訊息中提供了該 TLO 的名稱。) 檢查 ws\_collab ASI 的值。確定此協同作業存在，且正於執行中。若 ws\_mode BO 層次 ASI 的設定為 synch，則需要 ws\_collab ASI。請檢查 ws\_verb 物件層次 ASI 的值。確定 ws\_verb ASI 中所指定的動詞，可以觸發 ws\_collab ASI 所指定的協同作業。若您正使用非 TLO 來進行事件通知，請查驗 WSCollaborations 連接器內容。尋找這個商業物件要以同步方式來呼叫的協同作業。確定此協同作業存在，且正於執行中。

當連接器無法判斷連接器嘗試要傳送到整合分配管理系統的商業物件動詞時，就會在事件通知期間發生這個錯誤。解決方案：若您正使用 Web 服務 TLO 來進行事件通知，請確定您已為此 TLO 指定 ws\_verb 物件層次 ASI。將動詞指定為這個 ASI 的值。若您正使用非 TLO 來進行事件通知，則 Web 服務用戶端所要傳送的 SOAP 訊息，必須包含動詞元素。SOAP 資料處理常式會使用 SOAP 訊息中的動詞元素值，來設定商業物件的動詞。若 Web 服務用戶端未傳送 SOAP 訊息中的動詞，SOAP 資料處理常式就無法設定商業物件的動詞。於此情形下，連接器無法將商業物件傳遞至整合分配管理系統。如果您懷疑您的 Web 服務用戶端可能沒有將動詞元素包含在 SOAP 訊息中，您可為此商業物件提供 DefaultVerb 動詞層次 ASI。若您這麼做，連接器就會先將這個動詞設定於商業物件上，再將它傳送給整合分配管理系統。

---



---

## 附錄 A. 連接器的標準配置內容

本附錄說明 WebSphere Business Integration 配接器的連接器元件標準配置內容。本資訊涵蓋執行於下列整合分配管理系統上的連接器：

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker 及 WebSphere Business Integration Message Broker 總稱為 WebSphere Message Brokers (WMQI)。
- WebSphere Application Server (WAS)

並非每個連接器都會利用所有這些標準內容。當您從「連接器配置程式」中選取整合分配管理系統時，會看到您需要配置之標準內容的清單，以使您的配接器與該分配管理系統一起搭配執行。

如需特定連接器內容的相關資訊，請參閱相關的配接器使用手冊。

**註：**於本書中，反斜線 (\) 當做目錄路徑的使用慣例。在 UNIX 安裝方面，以斜線 (/) 取代反斜線，並遵循每一個作業系統的使用慣例。

---

### 新的及已刪除的內容

這些標準內容已新增於此版本中。

#### 新增內容

- XMLNamespaceFormat

#### 已刪除的內容

- RestartCount

---

### 配置標準連接器內容

配接器的連接器有兩種類型的配置內容：

- 標準配置內容
- 特有連接器配置內容

本節說明標準配置內容。如需連接器的特定配置內容之相關資訊，請參閱其配接器使用手冊。

### 使用連接器配置程式

您可從「系統管理程式」存取「連接器配置程式」來配置連接器內容。如需有關使用「連接器配置程式」的詳細資訊，請參照本手冊中有關「連接器配置程式」的小節。

**註：**「連接器配置程式」和「系統管理程式」僅於 Windows 系統上執行。如果您在 UNIX 系統上執行連接器，您必須有一部已安裝這些工具的 Windows 機器。若要對 UNIX 上執行的連接器設定連接器內容，您必須啟動 Windows 機器上的「系統管理程式」，並連接至 UNIX 整合分配管理系統，然後啟動「連接器配置程式」來設定連接器。

## 設定及更新內容值

內容欄位的預設長度為 255 個字元。

連接器依照下列次序來判斷內容的值 (最大號碼會取代其它值)：

1. 預設值
2. 儲存庫 (僅當 WebSphere InterChange Server 為整合分配管理系統時才適用)
3. 本端配置檔
4. 指令行

連接器會在啓動時取得配置值。若您在執行時期階段作業期間變更一或多個連接器內容的值，則內容的**更新方法**會決定變更生效的方式。標準連接器內容有四種不同的更新方法：

- **動態**

變更會在儲存於「系統管理程式」當中之後立即生效。若連接器是以獨立式模式來運作 (與「系統管理程式」無關)，例如與其中一種 WebSphere 訊息分配管理系統搭配使用，您便僅可透過配置檔來變更內容。在此情況下，即無法進行動態更新。

- **代理程式重新啓動 (僅限 ICS)**

僅於您停止並重新啓動特定應用程式元件之後，變更才會生效。

- **元件重新啓動**

僅在「系統管理程式」中停止並重新啓動連接器之後，變更才會生效。您並不需要停止及重新啓動特定應用程式的元件或整合分配管理系統。

- **伺服器重新啓動**

僅於您停止並重新啓動特定應用程式元件和整合分配管理系統之後，變更才會生效。

若要判定特定內容的更新方式，請參閱「連接器配置程式」視窗中的**更新方法**直欄，或參閱下列第 154 頁的表 49 中的「更新方法」直欄。

---

## 標準內容的摘要

表 49 提供標準連接器配置內容的快速參照。並非所有連接器都會利用所有的這些內容，內容設定可能會因整合分配管理系統的不同而有所差異，因為標準內容相依關係是以 RepositoryDirectory 為基礎。

執行連接器之前，您必須先設定其中一部份內容的值。請參閱下面章節，取得每一個內容的說明。

**註：**在表 49 中，"Repository directory is REMOTE" 這個句子表示分配管理系統是 InterChange Server。如果分配管理系統是 WMQI 或 WAS，儲存庫目錄就會設為 LOCAL

表 49. 標準配置內容的摘要

內容名稱	可能值	預設值	更新方法	注意事項
AdminInQueue	有效的 JMS 佇列名稱	CONNECTORNAME /ADMININQUEUE	元件 重新啓動	遞送 傳輸為 JMS

表 49. 標準配置內容的摘要 (繼續)

內容名稱	可能值	預設值	更新方法	注意事項
AdminOutQueue	有效的 JMS 佇列名稱	CONNECTORNAME/ADMINOUTQUEUE	元件 重新啟動	遞送 傳輸為 JMS
AgentConnections	1-4	1	元件 重新啟動	遞送 傳輸為 MQ 或 IDL : 儲存庫 目錄 為 <REMOTE> (分配管理系統是 ICS)
AgentTraceLevel	0-5	0	動態	
ApplicationName	應用程式名稱	指定給連接器 應用程式名稱的值	元件 重新啟動	
BrokerType	ICS、WMQI、WAS		元件 重新啟動	
CharacterEncoding	ascii7, ascii8, SJIS, Cp949, GBK, Big5, Cp297, Cp273, Cp280, Cp284, Cp037, Cp437 註: 這僅是受支援值 的子集。	ascii7	元件 重新啟動	
ConcurrentEventTriggeredFlows	1 至 32,767	1	元件 重新啟動	儲存庫 目錄 為 <REMOTE> (分配管理系統是 ICS)
ContainerManagedEvents	無值或 JMS	無值	元件 重新啟動	遞送 傳輸為 JMS
ControllerStoreAndForwardMode	true 或 false	true	動態	儲存庫 目錄 為 <REMOTE> (分配管理系統是 ICS)
ControllerTraceLevel	0-5	0	動態	儲存庫 目錄 為 <REMOTE> (分配管理系統是 ICS)
DeliveryQueue		CONNECTORNAME/DELIVERYQUEUE	元件 重新啟動	僅限 JMS 傳輸
DeliveryTransport	MQ、IDL 或 JMS	JMS	元件 重新啟動	如果 儲存庫 目錄 為本端，則 值只能 是 JMS

表 49. 標準配置內容的摘要 (繼續)

內容名稱	可能值	預設值	更新方法	注意事項
DuplicateEventElimination	true 或 false	false	元件重新啟動	僅限 JMS 傳輸： 「儲存區管理事件」必須是 <NONE>
FaultQueue		CONNECTORNAME/FAULTQUEUE	元件重新啟動	僅限 JMS 傳輸
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory 或 CxCommon.Messaging.jms.SonicMQFactory 或任何 Java 類別名稱	CxCommon.Messaging.jms.IBMMQSeriesFactory	元件重新啟動	僅限 JMS 傳輸
jms.MessageBrokerName	如果 FactoryClassName 是 IBM，請使用 crossworlds.queue.manager。 如果 FactoryClassName 是 Sonic，請使用 localhost:2506。	crossworlds.queue.manager	元件重新啟動	僅限 JMS 傳輸
jms.NumConcurrentRequests	正整數	10	元件重新啟動	僅限 JMS 傳輸
jms.Password	任何有效的密碼		元件重新啟動	僅限 JMS 傳輸
jms.UserName	任何有效的名稱		元件重新啟動	僅限 JMS 傳輸
JvmMaxHeapSize	資料堆大小 (以 MB 為單位)	128m	元件重新啟動	儲存庫目錄為 <REMOTE> (分配管理系統是 ICS)
JvmMaxNativeStackSize	堆疊大小 (以 KB 為單位)	128k	元件重新啟動	儲存庫目錄為 <REMOTE> (分配管理系統是 ICS)
JvmMinHeapSize	資料堆大小 (以 MB 為單位)	1m	元件重新啟動	儲存庫目錄為 <REMOTE> (分配管理系統是 ICS)
ListenerConcurrency	1- 100	1	元件重新啟動	遞送傳輸必須為 MQ
Locale	en_US、ja_JP、ko_KR、zh_CN、zh_TW、fr_FR、de_DE、it_IT、es_ES、pt_BR 註：這些是支援的語言環境的子集。	en_US	元件重新啟動	



表 49. 標準配置內容的摘要 (繼續)

內容名稱	可能值	預設值	更新方法	注意事項
LogAtInterchangeEnd	true 或 false	false	元件 重新啟動	儲存庫 目錄必須 為 <REMOTE> (分配管理系統是 ICS)
MaxEventCapacity	1-2147483647	2147483647	動態	儲存庫 目錄必須 為 <REMOTE> (分配管理系統是 ICS)
MessageFileName	路徑或檔案名稱	CONNECTORNAMEConnector.txt	元件 重新啟動	
MonitorQueue	任何有效的佇列名稱	CONNECTORNAME/MONITORQUEUE	元件 重新啟動	僅限 JMS 傳輸： DuplicateEvent Elimination 必須為 true
OADAutoRestartAgent	true 或 false	false	動態	儲存庫 目錄必須 為 <REMOTE> (分配管理系統是 ICS)
OADMaxNumRetry	正數	1000	動態	儲存庫 目錄必須 為 <REMOTE> (分配管理系統是 ICS)
OADRetryTimeInterval	正數 (分鐘)	10	動態	儲存庫 目錄必須 為 <REMOTE> (分配管理系統是 ICS)
PollEndTime	HH:MM	HH:MM	元件 重新啟動	
PollFrequency	正整數 (毫秒)  no (以停用輪詢)；  key (僅於連接器的 「指令提示」視窗中 輸入字元 p 時， 才能啟用輪詢)	10000	動態	
PollQuantity	1-500	1	代理程式 重新啟動	僅限 JMS 傳輸： 已指定 儲存區 管理的 事件
PollStartTime	HH:MM(HH 是 0-23，MM 是 0-59)	HH:MM	元件 重新啟動	

表 49. 標準配置內容的摘要 (繼續)

內容名稱	可能值	預設值	更新方法	注意事項
RepositoryDirectory	Meta 資料的位置 儲存庫		代理程式 重新啟動	針對 ICS：設為 <REMOTE> 若為 WebSphere MQ Message Brokers 及 WAS：設為 C:\crossworlds\ repository
RequestQueue	有效的 JMS 佇列名稱	CONNECTORNAME/REQUESTQUEUE	元件 重新啟動	遞送 傳輸為 JMS
ResponseQueue	有效的 JMS 佇列名稱	CONNECTORNAME/RESPONSEQUEUE	元件 重新啟動	遞送 傳輸為 JMS： 只有在儲存庫目錄 為 <REMOTE> 時才需要
RestartRetryCount	0-99	3	動態	
RestartRetryInterval	有理的正數值 (分鐘)： 1 - 2147483547	1	動態	
RHF2MessageDomain	mrm、xml	mrm	元件重新啟動	只有在遞送傳輸 為 JMS 且 WireFormat 為 CwXML 的情況 下。
SourceQueue	有效的 WebSphere MQ 名稱	CONNECTORNAME/SOURCEQUEUE	代理程式 重新啟動	只有在 遞送 傳輸為 JMS 且 已指定 儲存區 管理的 事件
SynchronousRequestQueue		CONNECTORNAME/ SYNCHRONOUSREQUESTQUEUE	元件 重新啟動	遞送 傳輸為 JMS
SynchronousRequestTimeout	0 - 任何數字 (毫秒)	0	元件 重新啟動	遞送 傳輸為 JMS
SynchronousResponseQueue		CONNECTORNAME/ SYNCHRONOUSRESPONSEQUEUE	元件 重新啟動	遞送 傳輸為 JMS
WireFormat	CwXML、CwBO	CwXML	代理程式 重新啟動	只有在儲存庫 目錄不是 <REMOTE> 時， 才是 CwXML： 只有當儲存庫目錄 為 <REMOTE> 才是 CwBO
WsifSynchronousRequestTimeout	0 - 任何數字 (毫秒)	0	元件 重新啟動	僅限 WAS

表 49. 標準配置內容的摘要 (繼續)

內容名稱	可能值	預設值	更新方法	注意事項
XMLNameSpaceFormat	短、長	短	代理程式 重新啓動	WebSphere MQ 訊息 Brokers 及 僅限 WAS

## 標準配置內容

本章節會列出及定義每一個標準連接器的配置內容。

### AdminInQueue

整合分配管理系統用以將管理訊息傳送至連接器的佇列。

預設值是 CONNECTORNAME/ADMININQUEUE。

### AdminOutQueue

連接器用以將管理訊息傳送至整合分配管理系統的佇列。

預設值是 CONNECTORNAME/ADMINOUTQUEUE。

### AgentConnections

只有在 RepositoryDirectory 為 <REMOTE> 時才適用。

AgentConnections 內容會控制由 orb.init[] 所開啓的 ORB (Object Request Broker) 連線數目。

這個內容的預設值設定為 1。您可以視需要變更它。

### AgentTraceLevel

特定應用程式元件的追蹤訊息層次。預設值是 0。連接器會遞送已設定之追蹤層次或更低層次上所有可用的追蹤訊息。

### ApplicationName

連接器的應用程式的獨特識別名稱。系統管理者使用此名稱來監督 WebSphere 商業整合系統環境。這個內容必須有一個值，您才可執行連接器。

### BrokerType

識別您目前使用的整合分配管理系統類型。選項有：ICS、WebSphere 訊息分配管理系統 (WMQI、WMQIB 或 WBIMB) 或 WAS。

### CharacterEncoding

指定從字元（例如，英文字母、數值表示法或標點符號）對映到數值所用的字碼集。

註：Java 型連接器不使用這個內容。C++ 連接器目前對這個內容使用 ascii7 這個值。

依預設，下拉清單中僅顯示一部份受支援的字元編碼。若要於下拉清單中加入其他支援的值，您必須以手動方式修改產品目錄中的 `\Data\Std\stdConnProps.xml` 檔案。如需詳細資訊，請參閱本手冊中有關「連接器配置程式」的小節。

## ConcurrentEventTriggeredFlows

只有在 `RepositoryDirectory` 為 `<REMOTE>` 時才適用。

決定連接器可同時處理多少商業物件，以遞送事件。請將這個屬性的值設為您要同時對映和遞送的商業物件數目。例如，將這個內容的值設為 5 可同時處理五個商業物件。預設值是 1。

將這個內容設為大於 1 的值可讓來源應用程式的連接器同時對映多個事件商業物件，且同時遞送至多個協同作業實例。尤其當商業物件使用複式對映時，可加速將商業物件遞送至整合分配管理系統。增加商業物件至協同作業實例的到達率可改善系統的整體效能。

實作整個流程（從來源應用程式到目的地應用程式）的並行處理，您必須：

- 配置協同作業來使用多重執行緒，方法是將其並行事件的最大數目內容調高至足以使用多重執行緒。
- 請確定目的地應用程式的特定應用程式元件可以並行處理要求。也就是說，它必須為多重執行緒，或者必須能夠使用連接器代理程式平行處理且針對多重處理程序進行配置。將平行處理程度配置內容設為大於 1 的值。

`ConcurrentEventTriggeredFlows` 內容不影響連接器輪詢（單一執行緒且序列化執行）。

## ContainerManagedEvents

這個內容可讓包含 JMS 事件儲存庫且具有 JMS 功能的連接器提供保證事件遞送，在此情形下，會將事件從來源佇列移除並放置到目的地佇列中成為單一 JMS 交易。

沒有預設值。

當 `ContainerManagedEvents` 設為 JMS 時，您必須配置下列內容來啟用保證事件遞送：

- 將 `PollQuantity = 1` 設為 500
- `SourceQueue = /SOURCEQUEUE`

您也必須以 `MimeType`、`DHClass` (資料處理常式類別) 及 `DataHandlerConfigMOName` (選用的 `Meta` 物件) 內容來配置資料處理常式。若要設定這些值，請使用「連接器配置程式」中的**資料處理常式**標籤。

這些內容是配接器特有的，但是**範例**值為：

- `MimeType = text/xml`
- `DHClass = com.crossworlds.DataHandlers.text.xml`
- `DataHandlerConfigMOName = M0_DataHandler_Default`

唯有當您已將 `ContainerManagedEvents` 設為 JMS 後，「資料處理常式」標籤中的這些值欄位才會顯示。

**註：**當 `ContainerManagedEvents` 設為 JMS 時，連接器不呼叫其 `pollForEvents()` 方法，因此會停用該方法的功能。

只有在 DeliveryTransport 內容設為 JMS 這個值時，這個內容才會出現。

## ControllerStoreAndForwardMode

只有在 RepositoryDirectory 為 <REMOTE> 時才適用。

設定連接器控制程式在偵測到目的地特定應用程式元件無法使用之後的行為。

若這個內容設為 true，且當事件到達 ICS 時無可用的目的地特定應用程式元件，則連接器控制程式會暫停對特定應用程式元件的要求。當特定應用程式元件恢復運作時，控制程式就會轉遞要求。

但是當連接器控制程式轉遞服務呼叫要求之後，若目的地應用程式的特定應用程式元件無法使用，則連接器控制程式會讓此要求失敗。

若這個內容設為 false，則當連接器控制程式偵測到目的地特定應用程式元件無法使用時，就會立即讓所有服務呼叫要求失敗。

預設值是 true。

## ControllerTraceLevel

只有在 RepositoryDirectory 為 <REMOTE> 時才適用。

連接器控制程式的追蹤訊息層次。預設值是 0。

## DeliveryQueue

只有在 DeliveryTransport 為 JMS 時才適用。

連接器用以將商業物件傳送至整合分配管理系統的佇列。

預設值為 CONNECTORNAME/DELIVERYQUEUE。

## DeliveryTransport

指定遞送事件的傳輸機制。可能值為 MQ (若為 WebSphere MQ)、IDL (若為 CORBA IIOP) 或 JMS (若為 Java Messaging Service)。

- 如果 RepositoryDirectory 為遠端，DeliveryTransport 內容值可以是 MQ、IDL 或 JMS，預設值為 IDL。
- 如果 RepositoryDirectory 是本端目錄，則值只能是 JMS。

若對 DeliveryTransport 內容配置的值為 MQ 或 IDL，則連接器會透過 CORBA IIOP 來傳送服務呼叫要求和管理訊息。

## WebSphere MQ 及 IDL

在事件遞送傳輸機制上使用 WebSphere MQ 來取代 IDL，除非您只能有一個產品。WebSphere MQ 提供下列優於 IDL 的優點：

- 非同步通訊：  
即使伺服器無法使用，WebSphere MQ 仍可使特定應用程式元件來輪詢並持續地儲存事件。
- 伺服器端效能：  
WebSphere MQ 在伺服器端提供更快的效能。在最佳化模式下，WebSphere MQ 僅

儲存指向儲存庫資料庫中事件的指標，實際事件留在 WebSphere MQ 佇列中。這可節省將大型事件寫入儲存庫資料庫的可能負荷。

- 代理程式端效能：

WebSphere MQ 在特定應用程式元件端提供更快的效能。運用 WebSphere MQ，連接器的輪詢執行緒會挑選一個事件放入連接器的佇列中，然後再挑選下一個事件。這比 IDL 還要快，因為 IDL 需要連接器的輪詢執行緒挑選一個事件，透過網路進入伺服器程序中，將事件持續地儲存在儲存庫資料庫中，然後再挑選下一個事件。

## JMS

透過「Java 訊息服務 (JMS)」來讓連接器與用戶端連接器組織架構彼此通信。

若您選取 JMS 作為遞送傳輸，則「連接器配置程式」中會顯示額外的 JMS 內容，例如 `jms.MessageBrokerName`、`jms.FactoryClassName`、`jms.Password` 以及 `jms.UserName`。此傳輸需要其中的前兩個內容。

**重要事項:** 若您於下列環境中對連接器使用 JMS 傳輸機制，可能有記憶體方面的限制：

- AIX 5.0
- WebSphere MQ 5.3.0.1
- 當 ICS 為整合分配管理系統時

此環境中，由於 WebSphere MQ 用戶端的記憶體使用情形，您可能難以啟動連接器控制程式（在伺服器端）和連接器（在用戶端）。若您的安裝架構使用小於 768M 的程序資料堆大小，則 IBM 建議您設定：

- CWSHaredEnv.sh Script 中的 LDR\_CNTRL 環境變數。

這個 Script 位於產品目錄的 `\bin` 目錄中。利用文字編輯器，在 CWSHaredEnv.sh Script 中新增下一行作為第一行：

```
export LDR_CNTRL=MAXDATA=0x30000000
```

此行可限制資料堆記憶體用量最大為 768 MB（3 區段 \* 256 MB）。若程序記憶體超過此限制，則對您的系統效能會造成負面影響。

- 將 IPCCBaseAddress 內容設為 11 或 12。如需這個內容的詳細資訊，請參閱系統安裝手冊 UNIX 版。

## DuplicateEventElimination

將這個內容設為 `true`，則可讓具有 JMS 功能的連接器確保不會遞送重複事件到遞送佇列中。若要使用這個功能，連接器必須有一個唯一事件識別碼設為應用程式特定程式碼中商業物件的 `ObjectEventId` 屬性。這會在連接器開發期間內完成。

這個內容亦可設為 `false`。

**註:** 當 DuplicateEventElimination 設為 `true` 時，您也必須配置 MonitorQueue 內容，才能啟用保證事件遞送。

## FaultQueue

若連接器在處理訊息時發生錯誤，則連接器會將訊息移至這個內容所指定的佇列，伴隨著狀態指示器和問題說明。

預設值是 CONNECTORNAME/FAULTQUEUE。

### **JvmMaxHeapSize**

代理程式資料堆大小的最大值 (以 MB 為單位)。只有當 RepositoryDirectory 值設為 <REMOTE> 時，才能使用這個內容。

預設值是 128m。

### **JvmMaxNativeStackSize**

代理程式原生堆疊大小的最大值 (以 KB 為單位)。只有當 RepositoryDirectory 值設為 <REMOTE> 時，才能使用這個內容。

預設值是 128k。

### **JvmMinHeapSize**

代理程式資料堆大小的最小值 (以 MB 為單位)。只有當 RepositoryDirectory 值設為 <REMOTE> 時，才能使用這個內容。

預設值是 1m。

### **jms.FactoryClassName**

指定為 JMS 提供者建立實例的類別名稱。當選擇 JMS 作為遞送傳輸機制 (DeliveryTransport) 時，您必須設定這個連接器內容。

預設值是 CxCommon.Messaging.jms.IBMMQSeriesFactory。

### **jms.MessageBrokerName**

指定用於 JMS 提供者的分配管理系統名稱。當選擇 JMS 作為遞送傳輸機制 (DeliveryTransport) 時，您必須設定這個連接器內容。

預設值是 crossworlds.queue.manager。連接到本端訊息分配管理系統時，請使用預設值。

當您連接到遠端訊息分配管理系統時，這個內容會採用下列 (強制) 值：

QueueMgrName:<Channel>:<HostName>:<PortNumber>，

其中的變數為：

QueueMgrName：佇列管理程式名稱。

Channel：用戶端使用的通道。

HostName：佇列管理程式將常駐的機器名稱。

PortNumber：佇列管理程式將用來接聽的埠號。

例如：

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

### **jms.NumConcurrentRequests**

指定可同時傳送至連接器的並行服務呼叫要求最大數目。一旦到達這個最大值時，新的服務呼叫會暫停執行，等到另一個要求完成之後才會繼續進行。

預設值是 10。





## MaxEventCapacity

控制程式緩衝區中的最大事件數目。這個內容是由流程控制所使用，且只有在 RepositoryDirectory 內容值設為 <REMOTE> 時才能使用。

此值可以是介於 1 和 2147483647 之間的正整數。預設值是 2147483647。

## MessageFileName

連接器訊息檔的名稱。此訊息檔的標準位置為產品目錄中的 \connectors\messages。若訊息檔不在標準位置中，請使用絕對路徑來指定訊息檔案名稱。

若連接器訊息檔不存在，連接器會使用 InterchangeSystem.txt 作為訊息檔。這個檔案位於產品目錄中。

**註：**若要判斷某個特定連接器是否擁有它自己的訊息檔案，請參閱個別的配接器使用手冊。

## MonitorQueue

連接器用來監視重複事件的邏輯佇列。只有當 DeliveryTransport 內容值為 JMS，且 DuplicateEventElimination 設為 TRUE 時，才會使用它。

預設值是 CONNECTORNAME/MONITORQUEUE

## OADAutoRestartAgent

只有在 RepositoryDirectory 為 <REMOTE> 時才有效。

指定連接器是否會使用自動及遠端重新啟動功能。這個功能會使用 MQ 觸發的「物件啟動常駐程式 (OAD)」，在異常關機之後重新啟動連接器，或是從「系統監視器」啟動遠端連接器。

這個內容必須設為 true，才能啟用自動及遠端重新啟動功能。如需如何配置 MQ 觸發的 OAD 功能的相關資訊，請參閱安裝手冊 *Windows 版* 或 *UNIX 版*。

預設值是 false。

## OADMaxNumRetry

只有在 RepositoryDirectory 為 <REMOTE> 時才有效。

指定 MQ 觸發的 OAD 在連接器異常關閉之後自動嘗試重新啟動的最大次數。OADAutoRestartAgent 內容必須設為 true，這個內容才會生效。

預設值是 1000。

## OADRetryTimeInterval

只有在 RepositoryDirectory 為 <REMOTE> 時才有效。

指定 MQ 觸發的 OAD 重試時間間隔的分鐘數。如果連接器代理程式沒有在這個重試時間間隔內重新啟動，連接器控制程式就會要求 OAD 再次重新啟動連接器。OAD 重複這個重試程序的次數就是 OADMaxNumRetry 內容所指定的次數。OADAutoRestartAgent 內容必須設為 true，這個內容才會生效。

預設值是 10。

## PollEndTime

停止輪詢事件佇列的時間。格式為 *HH:MM*，其中 *HH* 代表 0-23 時，*MM* 代表 0-59 秒。

您必須為這個內容提供有效的值。預設值是 *HH:MM*，但必須變更。

## PollFrequency

上次輪詢的結束和下次輪詢的開始之間會有間隔。PollFrequency 可指定結束某個輪詢動作和開始下個輪詢動作之間的時間量 (毫秒)。這不是輪詢動作之間的時間。邏輯如下：

- 輪詢以取得 PollQuantity 值所指定的物件數。
- 處理這些物件。對於某些配接器來說，部份動作可以在個別執行緒上完成，因為這些個別執行緒能夠非同步執行到下一個輪詢動作。
- 延遲由 PollFrequency 所指定的間隔。
- 重複這個週期循環。

請將 PollFrequency 設為下列其中一個值：

- 輪詢動作之間的毫秒數 (整數)。
- key 這個字，僅當您在連接器的「指令提示」視窗中鍵入字母 p 時，連接器才輪詢。請輸入小寫字體。
- no 這個字，使連接器不要輪詢。請輸入小寫字體。

預設值是 10000。

**重要事項：**有些連接器會限制這個內容的使用。有關安裝及配置配接器的那一章會說明這些限制 (如果有的話)。

## PollQuantity

指定來自連接器應該輪詢的應用程式之項目數。如果配接器具有連接器特有內容以供輪詢數量的設定使用，則連接器特有內容中所設定的值將會置換標準內容值。

### **FIX**

電子郵件訊息也會被視為事件。當連接器針對電子郵件進行輪詢時，其行為如下。

輪詢一次 - 連接器會挑選 1. 訊息本文 (因為它也會被視為附件)。因為這個 MIME 類型沒有指定 DH，所以它會忽略本文。 2. 連接器會處理第一個 PO 附件。這個 MIME 類型可以使用 DH，所以它會將商業物件傳送到「視覺化測試連接器」。如果 3. 在 VTC 中再次接受，則 BO 應該無法通過第二次輪詢 1. 連接器會處理第二個 PO 附件。這個 MIME 類型可以使用 DH，所以它會將 BO 傳送到 VTC。 2. 在 VTC 中再次接受，現在第三個 PO 附件應該可以通過。這是正確的行為。

## PollStartTime

開始輪詢事件佇列的時間。格式為 *HH:MM*，其中 *HH* 代表 0-23 時，*MM* 代表 0-59 秒。

您必須為這個內容提供有效的值。預設值是 *HH:MM*，但必須變更。

## RequestQueue

整合分配管理系統用以將商業物件傳送至連接器的佇列。

預設值為 CONNECTOR/REQUESTQUEUE。

## RepositoryDirectory

儲存庫的位置，連接器會從該處讀取用來儲存商業物件定義的 meta 資料之 XML 綱目文件。

如果整合分配管理系統為 ICS，此值必須設為 <REMOTE>，因為連接器是從 InterChange Server 儲存庫來取得這項資訊。

當整合分配管理系統為 WebSphere 訊息分配管理系統或 WAS 時，此值必須設為<本端目錄>。

## ResponseQueue

只有在 DeliveryTransport 為 JMS 時才適用，且只有在 RepositoryDirectory 為 <REMOTE> 時才需要。

指定 JMS 回應佇列，它會將回應訊息從連接器組織架構遞送到整合分配管理系統。如果整合分配管理系統為 ICS，伺服器會傳送要求，並等待 JMS 回應佇列中的回應訊息。

## RestartRetryCount

指定連接器嘗試自我重新啟動的次數。使用於並行連接器時，請指定主要連接器的特定應用程式元件嘗試重新啟動從屬連接器特定應用程式元件的次數。

預設值是 3。

## RestartRetryInterval

指定連接器嘗試自我重新啟動的間隔時間（分鐘）。使用於並行連接器時，請指定主要連接器的特定應用程式元件嘗試重新啟動從屬連接器特定應用程式元件的間隔時間。可能的值範圍從 1 到 2147483647。

預設值是 1。

## RHF2MessageDomain

僅限 WebSphere Message Brokers 及 WAS。

這個內容可讓您配置 JMS 標頭中領域名稱欄位的值。透過 JMS 傳輸將資料傳送到 WMQI 時，連接器組織架構會寫入 JMS 標頭資訊，其中包含領域名稱及 mrm 的固定值。可配置的領域名稱可讓使用者追蹤 WMQI 分配管理系統處理訊息資料的方式。

標頭的範例如下：

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

預設值為 mrm，但也可設為 xml。當 DeliveryTransport 設為 JMS 且 WireFormat 設為 CwXML 時，才會出現這個內容。

## SourceQueue

只有在 `DeliveryTransport` 為 `JMS` 且指定 `ContainerManagedEvents` 時才適用。

指定連接器組織架構的 `JMS` 來源佇列，以支援使用 `JMS` 事件儲存庫並且具有 `JMS` 功能的連接器的保證事件遞送。如需進一步資訊，請參閱第 160 頁的『`ContainerManagedEvents`』。

預設值為 `CONNECTOR/SOURCEQUEUE`。

## SynchronousRequestQueue

只有在 `DeliveryTransport` 為 `JMS` 時才適用。

將需要同步回應的要求訊息從連接器組織架構傳遞至分配管理系統。只在連接器採用同步執行時，才需要這個佇列。透過同步執行，連接器組織架構將訊息傳送至 `SynchronousRequestQueue`，然後在 `SynchronousResponseQueue` 等待來自分配管理系統的回應。傳送至連接器的回應訊息伴隨一個符合原始訊息 ID 的交互關聯 ID。

預設值為 `CONNECTORNAME/SYNCHRONOUSREQUESTQUEUE`

## SynchronousResponseQueue

只有在 `DeliveryTransport` 為 `JMS` 時才適用。

將回覆同步要求而送回的回應訊息從分配管理系統傳遞至連接器組織架構。只在連接器採用同步執行時，才需要這個佇列。

預設值為 `CONNECTORNAME/SYNCHRONOUSRESPONSEQUEUE`

## SynchronousRequestTimeout

只有在 `DeliveryTransport` 為 `JMS` 時才適用。

指定連接器等待同步要求的回應時間（分鐘）。若在指定的時間內未接收回應，則連接器會將原始同步要求訊息移至錯誤佇列中，隨伴著錯誤訊息。

預設值是 0。

## WireFormat

傳輸的訊息格式。

- 如果 `RepositoryDirectory` 是本端目錄，則設定為 `CwXML`。
- 如果 `RepositoryDirectory` 的值為 `<REMOTE>`，則設定為 `CwBO`。

## WsifSynchronousRequestTimeout

僅限 `WAS` 整合分配管理系統。

指定連接器等待同步要求的回應時間（分鐘）。若在指定的時間內未接收回應，則連接器會將原始同步要求訊息移至錯誤佇列中，隨伴著錯誤訊息。

預設值是 0。

## **XMLNamespaceFormat**

僅限 WebSphere 訊息分配管理系統及 WAS 整合分配管理系統。

健全的內容可讓使用者以 XML 格式的商業物件定義來指定短的或長的名稱空間。

預設值為短名稱。



---

## 附錄 B. 連接器配置程式

本附錄說明如何使用「連接器配置程式」來設定配接器的配置內容值。

您可以使用「連接器配置程式」執行下列作業：

- 建立一個連接器特有的內容範本來配置您的連接器
- 建立配置檔
- 設定配置檔內容

### 註：

於本書中，反斜線 (\) 當做目錄路徑的使用慣例。在 UNIX 安裝方面，以斜線 (/) 取代反斜線，並遵循每一個作業系統的使用慣例。

本附錄涵蓋的主題如下：

- 第 171 頁的『連接器配置程式概觀』
- 第 172 頁的『啟動連接器配置程式』
- 第 173 頁的『建立連接器特有的內容範本』
- 第 175 頁的『建立新配置檔』
- 第 177 頁的『設定配置檔內容』
- 第 183 頁的『在全球化環境下使用連接器配置程式』

---

### 連接器配置程式概觀

「連接器配置程式」可讓您配置配接器的連接器元件，以搭配下列這些整合分配管理系統一起使用：

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker 及 WebSphere Business Integration Message Broker 總稱為 WebSphere Message Brokers (WMQI)
- WebSphere Application Server (WAS)

您可以使用「連接器配置程式」執行下列作業：

- 建立**連接器專屬的內容範本**來配置您的連接器。
- 建立**連接器配置檔**；您必須為您所安裝的每一個連接器，各建立一個配置檔。
- 於配置檔中設定內容。  
您可能需要修改連接器範本中所設定的內容預設值。您也必須指定所支援的商業物件定義，及具有 ICS 的對映表，以用於協同作業，並依需要來指定傳訊、記載與追蹤，以及資料處理常式參數。

根據執行的整合分配管理系統，「連接器配置程式」的執行模式以及使用的配置檔類型會有所不同。比方說，如果分配管理系統是 WMQI，則必須直接執行「連接器配置程式」，而不是從「系統管理程式」執行「連接器配置程式」（請參閱第 172 頁的『以獨立模式執行配置程式』）。

連接器配置內容包含標準配置內容 (所有連接器擁有的內容) 和連接器特有內容 (連接器針對特定應用程式或技術所需的內容)。

由於所有的連接器都會使用**標準內容**，所以您不需要從頭定義這些內容；只要您建立了配置檔，「連接器配置程式」就會將這些定義納入您的配置檔中。不過，您必須在「連接器配置程式」設定每個標準內容的值。

標準內容的範圍對每個分配管理系統及配置都不盡相同。部分內容只有在其他內容指定了特定值才可用。「連接器配置程式」的「標準內容」視窗中，將會顯示您的特定配置可使用的內容。

不過，若是**連接器特有內容**，您必須先定義內容，然後再設定其值。做法是為您的特定配接器建立連接器特有內容範本。若您的系統有設定好的範本，即可使用之。否則，請遵循第 173 頁的『建立新範本』中的步驟，設定新範本。

**註：**「連接器配置程式」僅適合在 Windows 環境下執行。如果您要在 UNIX 環境下執行連接器，請在 Windows 系統中使用「連接器配置程式」來修改配置檔，然後將檔案複製到您的 UNIX 環境。

---

## 啓動連接器配置程式

您可以兩個模式啓動及執行「連接器配置程式」：

- 以獨立模式執行
- 從系統管理程式執行

### 以獨立模式執行配置程式

不論分配管理系統，您都可以單獨執行「連接器配置程式」並使用連接器配置檔。

做法是：

- 從**開始 > 程式集**，按一下 **IBM WebSphere InterChange Server > IBM WebSphere Business Integration Tools > 連接器配置程式**。
- 選取**檔案 > 新建 > 連接器配置**。
- 當您按一下**系統連線功能整合分配管理系統**旁的下拉功能表時，您可以根據您的分配管理系統來選取 ICS、WebSphere Message Brokers 或 WAS。

您可以選擇先單獨執行「連接器配置程式」產生配置檔，然後再連接「系統管理程式」，將該檔案儲存在「系統管理程式」專案中 (請參閱第 177 頁的『完成配置檔』)。

---

## 從系統管理程式執行配置程式

您可以從「系統管理程式」執行「連接器配置程式」。

若要執行「連接器配置程式」：

1. 開啓「系統管理程式」。
2. 於「系統管理程式」視窗中，展開**整合元件程式庫**圖示，並強調顯示**連接器**。
3. 從「系統管理程式」功能表列中，按一下**工具 > 連接器配置程式**。即開啓「連接器配置程式」視窗並顯示**新建連接器對話框**。



4. 當您按一下**系統連線功能整合分配管理系統**旁的下拉功能表時，您可以根據您的分配管理系統來選取 ICS、WebSphere Message Brokers 或 WAS。

若要編輯現有的配置檔：

- 在「系統管理程式」視窗中，選取任何一個「連接器」資料夾中所列出的配置檔，然後用滑鼠右鍵按一下該檔案。即會開啓「連接器配置程式」，並顯示此配置檔，且頂端會有整合分配管理系統類型與檔案名稱。
- 從「連接器配置程式」中選取**檔案 > 開啓**。從儲存連接器配置檔的專案或目錄中，選取連接器配置檔的名稱。
- 請按一下「標準內容」標籤，查看哪些內容包含在此配置檔中。

---

## 建立連接器特有的內容範本

若要為您的連接器建立配置檔，您需要一個連接器特定的內容範本以及系統提供的標準內容。

您可以為連接器的特有內容建立全新的範本，也可以將現有的連接器定義當作範本。

- 若要建立範本，請參閱第 173 頁的『建立新範本』。
- 若要使用現有的檔案，僅需要修改現有的範本，並用新名稱來儲存它。您可以在 `\WebSphereAdapters\bin\Data\App` 目錄中找到現有的範本。

### 建立新範本

本節說明如何在範本中建立內容、定義這些內容的一般性質和值，以及指定內容之間的任何相依關係。然後，儲存範本並使用它來作為建立新連接器配置檔的基礎。

若要在「連接器配置程式」中建立範本：

1. 按一下**檔案 > 新建 > 連接器特有內容範本**。
2. 即會顯示**連接器特有內容範本**對話框。
  - 在**輸入新範本名稱**下的**名稱**欄位中，輸入新範本的名稱。當您再次開啓對話框來從範本建立新配置檔時，將會再次看到這個名稱。
  - 若要查看任何範本的連接器特有內容定義，請在**範本名稱**顯示畫面中選取該範本的名稱。該範本內含的內容定義清單會出現在**範本預覽**顯示畫面中。
3. 若現有範本的內容定義類似於連接器所需的內容定義，則您可以使用它當作範本的好起點。如果看不到顯示您連接器所使用特定內容的任何範本，您必須自行建立。
  - 如果您規劃修改現有的範本，請在**選取要修改的現有範本：尋找範本** 下的**範本名稱**表格的清單中，選取範本名稱。
  - 此表格會顯示所有目前可用的範本的名稱。您也可以搜尋範本。

### 指定一般性質

當您按**下一步**選取範本時，即顯示**內容 - 連接器特有內容範本**對話框。此對話框具有已定義內容的「一般」性質和「值」限制的標籤。「一般」顯示畫面具有下列欄位：

- **一般：**
  - 內容類型
  - 已更新的方法
  - 說明

- **旗標**  
標準旗標
- **自訂旗標**  
旗標

完成選取內容的一般性質之後，按一下**值**標籤。

## 指定值

**值**標籤可讓您設定內容的最大長度、最大多重值、預設值或值範圍。它也容許可編輯的值。做法是：

1. 按一下**值**標籤。「**值**」顯示畫面會取代「**一般**」顯示畫面。
2. 在**編輯內容**顯示畫面中選取內容名稱。
3. 在**最大長度**和**最大多重值**欄位中輸入值。

若要建立新內容值：

1. 在**編輯內容**清單中選取內容，然後用滑鼠右鍵按一下它。
2. 從對話框中選取**新增**。
3. 輸入新內容值的名稱，然後按一下「**確定**」。值會顯示在右側的**值**畫面中。

**值**畫面會顯示包含三個直欄的表格：

**值**直欄，它顯示您在**內容值**對話框中所輸入的值，以及您先前所建立的任何值。

**預設值**直欄，它可讓您指定任何值作為預設值。

**值範圍**，它顯示您在**內容值**對話框中所輸入的範圍。

如果方格中出現所建立的**值**，您就可以從表格顯示畫面內來編輯該值。

若要變更表格中現有的值，請按下列號來選取整列。然後，用滑鼠右鍵按一下**值**欄位並按一下**編輯值**。

## 設定相依關係

在您變更了**一般**標籤與**值**標籤，請按**下一步**。即會顯示**相依關係 - 連接器特有內容範本**對話框。

相依項內容是指只有當另一內容的值符合特定條件時才併入範本及使用於配置檔中的內容。舉例來說，只有當 **JMS** 為傳輸機制時，以及 **DuplicateEventElimination** 設為 **True**時，**PollQuantity** 才會出現在範本中。

若要將某個內容指定為相依項並根據它來設定條件，請執行下列步驟：

1. 在**可用的內容**顯示畫面中，選取要建立成相依項的內容。
2. 在**選取內容**欄位中，使用下拉功能表選取將保留其條件值的內容。
3. 在**條件運算子**欄位中，選取下列其中一項：

== (等於)

!= (不等於)

> (大於)

< (小於)

>= (大於或等於)

<= (小於或等於)

4. 在**條件值**欄位中，輸入要讓相依項內容併入範本的必要值。
5. 在**可用的內容**顯示畫面中強調顯示相依項內容後，按一下箭頭將它移至**相依項內容**顯示畫面中。
6. 按一下**完成**。「連接器配置程式」會將您所輸入的資訊儲存為 XML 文件，儲存位置是在您「連接器配置程式」安裝目錄下 \bin 目錄的 \data\app。

---

## 建立新配置檔

當您建立新的配置檔時，必須為它命名，然後選取整合分配管理系統。

- 在「系統管理程式」視窗中，用滑鼠右鍵按一下**連接器**資料夾，並選取**建立新的連接器**。即會開啓「連接器配置程式」，並顯示**新建連接器**對話框。
- **獨立式模式**：從「連接器配置程式」中選取**檔案 > 新建 > 連接器配置程式**。在「新建連接器」視窗中輸入新連接器的名稱。

您也必須選取整合分配管理系統。您所選取的分配管理系統將決定出現在配置檔中的內容。若要選取分配管理系統：

- 在**整合分配管理系統**欄位中，選取 ICS、WebSphere Message Brokers 或 WAS 連線功能。
- 完成**新建連接器**視窗中剩餘的欄位，本章稍後將說明。

## 從連接器特有範本建立配置檔

一旦建立了連接器特有範本，您可以使用它來建立配置檔：

1. 按一下**檔案 > 新建 > 連接器配置**。
2. 即顯示**新建連接器**對話框，其中包含下列欄位：
  - **名稱**

請輸入連接器名稱。名稱需要區分大小寫。您輸入的名稱必須是唯一的，而且必須與系統上所安裝的連接器檔名一致。

**重要事項：**「連接器配置程式」不會檢查您所輸入的名稱拼法。您必須確定名稱無誤。

- **系統網路功能**

按一下 ICS、WebSphere Message Brokers 或 WAS。

- **選取連接器特有內容範本**

請鍵入針對連接器所設計的範本名稱。所有可用的範本會顯示於**範本名稱**顯示畫面中。當您在「範本名稱」顯示畫面中選取名稱時，**內容範本預覽**顯示畫面會顯示已在該範本中定義的那些連接器特有內容。

選取所要使用的範本，然後按一下**確定**。

3. 此時會針對您配置的連接器顯示一個配置畫面。標題列會顯示整合分配管理系統及連接器名稱。您可以現在填入所有欄位值來完成定義，或儲存檔案並於稍後完成這些欄位。
4. 若要儲存檔案，請按一下**檔案 > 儲存 > 儲存至檔案**，或**檔案 > 儲存 > 儲存至專案**。若要儲存至專案，則必須執行「系統管理程式」。  
如果是儲存為檔案，則會出現**儲存檔案連接器**對話框。選擇 \*.cfg 做為檔案類型，在檔案名稱欄位中驗證其名稱拼法是否正確，並注意大小寫的正确性，導覽至要放置檔案的目錄，並按一下**儲存**。「連接器配置程式」訊息畫面中顯示的狀態，指示配置檔已順利建立完成。  
**重要事項:** 您在此建立的目錄路徑和名稱必須符合您在啟動檔中為連接器提供的連接器配置檔路徑和名稱。
5. 若要完成連接器定義，請在「連接器配置程式」視窗的每一個標籤的欄位中分別輸入想要使用的值，本章稍後會提供說明。

---

## 使用現存檔

您可以有下列其中一或多種格式的可用現存檔：

- 連接器定義檔。  
這是一種文字檔，它列示特定連接器的內容和適合的預設值。某些連接器將這種檔案併入於其遞送套件中的 \repository 目錄下 (這種檔案的副檔名通常是 .txt；例如，XML 連接器的 CN\_XML.txt)。
- ICS 儲存庫檔。  
先前連接器的 ICS 實作所用的定義可能出現在該連接器配置中所使用的儲存庫檔案。這種檔案的副檔名通常是 .in 或 .out。
- 連接器先前的配置檔。  
這種檔案的副檔名通常是 \*.cfg。

雖然這些檔案來源可能包含您連接器大部份或全部的連接器特有內容，但除非您開啓檔案並設定內容，否則連接器配置檔將不會完成，本章稍後將做說明。

若要使用現存檔來配置連接器，您必須在「連接器配置程式」中開啓檔案、修改配置，然後重新儲存檔案。

請遵循下列步驟來從目錄中開啓 \*.txt、\*.cfg 或 \*.in 檔：

1. 在「連接器配置程式」中，按一下**檔案 > 開啓 > 從檔案**。
2. 在**開啓檔案連接器**對話框中，選取下列其中一個檔案類型來查看可用的檔案：
  - 配置 (\*.cfg)
  - ICS 儲存庫 (\*.in、\*.out)

如果使用儲存庫檔在 ICS 環境下配置連接器，請選擇這個選項。儲存庫檔可能包含多個連接器定義，當您開啓檔案時，所有這些定義都會顯示出來。

- 全部檔案 (\*.\*)

如果連接器的配接器套件中有提供 \*.txt 檔，或有提供另一個副檔名的定義檔，請選擇本選項。

3. 在目錄顯示畫面中，導覽至適當的連接器定義檔、選取它，然後按一下**開啓**。

請遵循下列步驟來從「系統管理程式」專案中開啓連接器配置：

1. 啓動「系統管理程式」。唯有在「系統管理程式」已啓動的情況下，才可以從「系統管理程式」開啓配置或將配置儲存到「系統管理程式」。
2. 啓動「連接器配置程式」。
3. 按一下 **檔案 > 開啓 > 從專案**。

---

## 完成配置檔

當您從專案開啓配置檔或連接器時，「連接器配置程式」視窗會顯示配置畫面，其中包含現行屬性和值。

配置畫面標題顯示檔案中所指定的整合分配管理系統和連接器名稱。確定分配管理系統是您所要的。如果不是，請先變更分配管理系統值再配置連接器。做法是：

1. 在**標準內容**標籤下，選取 `BrokerType` 內容的值欄位。在下拉功能表中，選取值 `ICS`、`WMQI` 或 `WAS`。
2. 「標準內容」標籤會顯示與已選取分配管理系統相關的內容。您可以立刻儲存檔案或完成剩餘的配置欄位，如第 179 頁的『指定支援的商業物件定義』中所述。
3. 完成配置後，請按一下 **檔案 > 儲存 > 至專案**，或 **檔案 > 儲存 > 至檔案**。

如果您要儲存至檔案，請選取 `*.cfg` 作為副檔名、選取正確的檔案位置，然後選取**儲存**。

如果開啓了多個連接器配置，請按一下**全部儲存至檔案**，以將所有配置儲存至檔案，或按一下**全部儲存至專案**，以將所有連接器配置儲存至「系統管理程式」專案。

在儲存檔案之前，「連接器配置程式」會檢查所有必要的標準內容值是否皆已設定。如果有必要的標準內容遺漏了值，「連接器配置程式」便會顯示一則指示驗證失敗的訊息。您必須為該內容提供值，才能儲存配置檔。

---

## 設定配置檔內容

當您建立並命名新的連接器配置檔，或當您開啓現存連接器配置檔時，「連接器配置程式」會顯示一個配置畫面，其中具有一些用於必要配置值種類的標籤。

對於所有分配管理系統上執行的連接器，「連接器配置程式」需要這些種類的內容值：

- 標準內容
- 連接器特有內容
- 支援的商業物件
- 追蹤/日誌檔值
- 資料處理常式 (適用於使用含保證事件遞送的 `JMS` 傳訊之連接器)

**註：**若是使用 `JMS` 傳訊的連接器，則可能會針對將資料轉換成商業物件的資料處理常式配置來顯示其他的種類。

針對執行於 `ICS` 的連接器，也需要這些內容值：

- 相關對映

- 資源
- 傳訊 (在可適用情況下)

**重要事項:** 「連接器配置程式」接受英文或非英文字集的內容值。不過，標準和連接器特有內容的名稱，以及支援的商業物件名稱，只能使用英文字集。

標準內容不同於連接器特有內容，如下所示：

- 連接器的標準內容由連接器的特定應用程式元件及其分配管理系統元件共用。所有連接器都有相同的標準內容集。這些內容說明於每一個配接器手冊的「附錄 A」中。您可以變更一部份值，但不能變更所有的值。
- 特定應用程式專用內容僅套用到連接器的特定應用程式元件，亦即，直接與應用程式互動的元件。每一個連接器都有專屬於其應用程式的特定應用程式內容。並非所有內容都有提供預設值；您可以修改某些預設值。每一個配接器手冊的「安裝與配置」一章皆有提供特定應用程式內容及建議值的說明。

標準內容與連接器特有內容的欄位皆以色彩標示來顯示哪些是可配置的：

- 灰色背景的欄位指示標準內容。您可以變更其值，但不能變更名稱或移除內容。
- 白色背景的欄位指示特定應用程式內容。這些內容會隨應用程式或連接器的特定需求而有不同。您可以變更其值和刪除這些內容。
- 值欄位是可配置的。
- 每個內容都會顯示**更新方法**欄位。它指出元件或代理程式是否需要重新啟動以使變更值生效。您無法配置此設定。

## 設定標準連接器內容

若要變更標準內容的值：

1. 按一下想要設定其值的欄位。
2. 輸入所要設定的值，或從下拉功能表 (若有出現) 中選取值。
3. 輸入標準內容的所有值之後，您可以執行下列其中一項作業：
  - 若要捨棄變更、保留原始值並結束「連接器配置程式」，請按一下**檔案 > 結束** (或關閉視窗)，然後當提示您儲存變更時，請按一下**否**。
  - 若要輸入「連接器配置程式」中其他種類的值，請選取代表該種類的標籤。當您移至下一個種類時，您對**標準內容** (或其他種類) 所輸入的值會被保留下來。當您關閉視窗時，系統會提示您儲存或捨棄您在所有種類中輸入的值。
  - 若要儲存已修訂的值，請按一下**檔案 > 結束** (或關閉視窗)，然後當提示您儲存變更時，請按一下**是**。此外，您也可以從「檔案」功能表或工具列中按一下**儲存 > 至檔案**。

## 設定特定應用程式專用的配置內容

針對特定應用程式專用的配置內容，您可以新增或變更內容名稱、配置值、刪除內容及加密內容。預設內容長度是 255 個字元。

1. 用滑鼠右鍵按一下方格左上方。即出現一個蹦現式功能表列。按一下**新增**來新增內容。若要新增子項內容，請用滑鼠右鍵按一下母項列號，並按一下**新增子項**。
2. 輸入內容或子項內容的值。
3. 若要加密內容，請選取**加密**方框。
4. 選擇儲存或捨棄變更，如『設定標準連接器內容』所述。

針對每一個內容顯示的「更新方法」，可指出元件或代理程式是否需要重新啓動以使變更值生效。

**重要事項：**變更預設的特定應用程式連接器內容名稱可能會導致連接器失敗。連接器可能需要某些特定的內容名稱，才能連接應用程式或正確地執行。

### 為連接器內容加密

您可以選取「連接器特有內容」視窗中的**加密**勾選框來為特定應用程式內容加密。若要對某個值解密，請按一下以清除**加密**勾選框、在**驗證**對話框中輸入正確值，然後按一下**確定**。如果輸入的是正確值，則該值會被解密並顯示出來。

每一個連接器的配接器使用手冊都有每一個內容及其預設值的清單和說明。

如果內容具有多重值，則會針對內容的第一個值顯示**加密**勾選框。當您選取**加密**時，該內容的所有值都會被加密。若要對內容的多重值進行解密，請按一下來清除該內容第一個值的**加密**勾選框，然後在**驗證**對話框中輸入一個新的值。如果輸入值符合，所有多重值都會被解密。

### 更新方法

請參閱第 154 頁的『設定及更新內容值』下的**連接器的標準配置內容附錄**中的更新方法說明。

## 指定支援的商業物件定義

使用「連接器配置程式」中**支援的商業物件**標籤，指定此連接器將使用的商業物件。您必須同時指定通用商業物件和特定應用程式專用的商業物件，而且必須指定商業物件之間的對映關係。

**註：**某些連接器需要指定一些支援的商業物件，以便搭配其應用程式來執行事件通知或其他的配置（使用 Meta 物件）。如需詳細資訊，請參閱 *Connector Development Guide for C++* 或 *Connector Development Guide for Java*。

### 如果 ICS 是您的分配管理系統

若要指定連接器支援的商業物件定義，或變更現有商業物件定義的支援設定，請按一下**支援的商業物件**標籤並使用下列欄位。

**商業物件名稱：** 若要使用「系統管理程式」來指定連接器支援的商業物件定義，請執行：

1. 按一下**商業物件名稱**清單中的空欄位。即顯示一個下拉清單，內容是目前在「系統管理程式」專案中的所有商業物件定義。
2. 按一下商業物件來新增它。
3. 為商業物件設定**代理程式支援**（如下所述）。
4. 在「連接器配置程式」視窗的「檔案」功能表中，按一下**儲存至專案**。已修訂的連接器定義（包括對新增的商業物件定義指定的支援）會儲存到「系統管理程式」中的 ICL 專案。

若要從支援的清單中刪除商業物件：

1. 要選取商業物件欄位，請按一下商業物件左邊的號碼。

2. 從「連接器配置程式」視窗的**編輯**功能表中，按一下**刪除列**。如此便會從清單顯示畫面移除商業物件。
3. 從**檔案**功能表中，按一下**儲存至專案**。

從支援的清單中刪除商業物件會變更連接器定義，並使刪除的商業物件無法用於此連接器的這個實作中。它並不會影響連接器的程式碼，也不會從「系統管理程式」中移除商業物件定義本身。

**代理程式支援：** 如果商業物件有「代理程式支援」，這表示系統將會嘗試使用該商業物件，透過連接器代理程式傳遞資料給應用程式。

一般而言，連接器的特定應用程式商業物件是受到連接器代理程式的支援，但通用商業物件不受其支援。

若要指示商業物件是受到連接器代理程式的支援，請勾選**代理程式支援**方框。「連接器配置程式」視窗不會驗證您的「代理程式支援」選項。

**最大交易層次：** 連接器的最大交易層次就是連接器支援的最高交易層次。

對大部份的連接器而言，最佳嘗試是唯一的選擇。

您必須重新啟動伺服器才能使交易層次的變更生效。

## 如果 WebSphere Message Broker 是您的分配管理系統

如果您是在獨立式的模式作業 (而非連接到系統管理程式)，必須以手動方式輸入商業物件名稱。

如果執行「系統管理程式」，您可以在**支援的商業物件**標籤中的**商業物件名稱**欄位下，選取空的方框。此時會出現組合框，並列出連接器所屬的「整合元件程式庫」專案中可用的商業物件。請在清單中選取想要的商業物件。

**訊息集 ID** 是 WebSphere Business Integration Message Broker 5.0 的選用欄位，不一定是唯一值 (如果您提供此值)。然而，您必須為 WebSphere MQ Integrator 及 Integrator Broker 2.1 提供唯一的 **ID**。

## 如果 WAS 是您的分配管理系統

當選取 WebSphere Application Server 做為您的分配管理系統類型，則「連接器配置程式」不需要訊息集 ID。只有在支援商業物件時，**支援的商業物件**標籤才會顯示出**商業物件名稱**欄位。

如果您是在獨立式的模式作業 (而非連接到系統管理程式)，必須以手動方式輸入商業物件名稱。

如果執行「系統管理程式」，您可以在「支援的商業物件」標籤中的「商業物件名稱」欄位下，選取空對話框。組合框會出現，列出連接器所屬「整合元件程式庫」專案中可用的商業物件。請在此清單中選取想要的商業物件。

## 相關對映 (僅限 ICS)

每一個連接器皆支援目前在 WebSphere InterChange Server 作用中的商業物件定義及其相關對映的清單。當您選取**相關對映**標籤時便會顯示此清單。



商業物件清單包含代理程式支援的特定應用程式商業物件，和控制程式傳至訂閱協同作業的對應通用物件。對映關聯可決定使用哪一種對映來將特定應用程式商業物件轉換成通用商業物件，或將通用商業物件轉換成特定應用程式商業物件。

如果您使用一些專門為特定來源和目標商業物件來定義的對映，則當您開啓顯示畫面時，這些對映已經與其適當的商業物件相關聯，您不需要 (或不可以) 變更它們。

如果支援的商業物件可使用多個對映，您必須明確地連結該商業物件與其應使用的對映。

**相關對映**標籤顯示下列欄位：

- **商業物件名稱**

這些商業物件是指**支援的商業物件**標籤中所指出由此連接器支援的商業物件。如果您在「支援的商業物件」標籤下指定其他的商業物件，則在「連接器配置程式」視窗的**檔案**功能表中選擇**儲存至專案**來儲存變更之後，這些指定的商業物件將反映到此清單中。

- **相關對映**

此顯示畫面顯示所有已安裝到系統中來搭配連接器受支援商業物件使用的對映。在**商業物件名稱**顯示畫面中，每一個對映的來源商業物件會在對映名稱左邊顯示。

- **明確**

在某些情形下，您需要明確地連結相關對映。

唯有在特定支援的商業物件有多個對映時，才需要明確連結。當 ICS 啓動時，它會嘗試自動將某個對映連結到每個連接器的每一個受支援商業物件。如果不止一個對映使用相同商業物件作為它的輸入，則伺服器會嘗試尋找和連結本身是其他對映超集的一個對映。

如果沒有對映的超集，則伺服器無法將商業物件連結到單一對映，所以您必須明確地設定連結。

若要明確地連結對映：

1. 在**明確**直欄中，針對您想要連結的對映，在勾選框中勾選。
2. 選取您想要與商業物件相關聯的對映。
3. 在「連接器配置程式」視窗的**檔案**功能表中，按一下**儲存至專案**。
4. 將專案部署到 ICS。
5. 重新啓動伺服器以使變生效。

## 資源 (ICS)

**資源**標籤可讓您設定值，來決定連接器代理程式是否使用連接器代理程式平行處理方式來同時處理多重程序，以及決定其處理範圍。

並非所有連接器都支援此項功能。如果您是使用 Java 設計成多執行緒的連接器代理程式，通常不建議使用此功能，因為使用多執行緒的通常比使用多重程序更有效率。

## 傳訊 (ICS)

唯有在您將 MQ 設為 DeliveryTransport 標準內容的值且將 ICS 設為分配管理系統類型時，才能使用傳訊內容。這些內容將影響您連接器使用佇列的方式。

## 設定追蹤/日誌檔案值

當開啓連接器配置檔或連接器定義檔時，「連接器配置程式」會使用該檔案的日誌記載和追蹤值作為預設值。您可以在「連接器配置程式」中變更這些值。

若要變更日誌記載和追蹤值：

1. 按一下**追蹤/日誌檔**標籤。
2. 您可以選擇日誌記載或追蹤來將訊息寫入下列其中之一或兩者：

- 至主控台 (STDOUT)：  
將日誌記載或追蹤訊息寫入 STDOUT 顯示畫面。

**註：**對於在 Windows 平台上執行的連接器，您僅能使用**追蹤/日誌檔**標籤的 STDOUT 選項。

- 至檔案：  
將日誌記載或追蹤訊息寫入指定的檔案。若要指定檔案，請按一下目錄按鈕 (省略符號)、導覽至偏好的位置、提供檔案名稱，然後按一下**儲存**。日誌記載或追蹤訊息會被寫入您指定的檔案和位置。

**註：**日誌記載和追蹤檔為純文字檔。當設定其檔名時，您可以使用偏好的副檔名。不過，若是追蹤檔的話，建議使用的副檔名為 .trace 而非 .trc，以免和可能位於系統中的其他檔案產生混淆。若是日誌檔的話，.log 和 .txt 是典型的副檔名。

## 資料處理常式

唯有當您對 DeliveryTransport 指定了 JMS 的值並對 ContainerManagedEvents 指定了 JMS 的值，資料處理常式區段才可用於配置。並非所有的配接器都使用資料處理常式。

有關這些內容使用的值，請參閱「附錄 A. 標準內容」中 ContainerManagedEvents 下的說明。如需其他詳細資料，請參閱 *Connector Development Guide for C++* 或 *Connector Development Guide for Java*。

---

## 儲存配置檔

當您完成配置連接器時，請儲存連接器配置檔。「連接器配置程式」會以您在配置時所選取的分配管理系統模式來儲存檔案。「連接器配置程式」標題列恆顯示目前使用的分配管理系統模式 (ICS、WMQI 或 WAS)。

此檔案會被儲存為 XML 文件。有三個方法可以儲存 XML 文件：

- 從「系統管理程式」中，當作「整合元件程式庫」中具有 \*.con 副檔名的檔案，或是
- 在您所指定的目錄中。
- 在獨立式模式當中，當作目錄資料夾中具有 \*.cfg 副檔名的檔案。依預設，檔案會儲存到 \WebSphereAdapters\bin\Data\App。
- 如果您已經設定檔案，則也可以將它儲存到 WebSphere Application Server 專案。

有關如何使用「系統管理程式」專案的詳細資料以及部署的進一步資料，請參閱下列實作手冊：

- ICS : *Implementation Guide for WebSphere InterChange Server*
- WebSphere Message Brokers : *Implementing Adapters with WebSphere Message Brokers*
- WAS : *Implementing Adapters with WebSphere Application Server*

---

## 變更配置檔

您可以針對現有的配置檔變更整合分配管理系統設定。如此可讓您將此檔案當作範本，用來建立新的配置檔，與另一個分配管理系統一起搭配使用。

**註：**若您切換整合分配管理系統，您將需要變更其他配置內容以及分配管理系統模式的內容。

若要變更現有配置檔內的分配管理系統選項 (選用)：

- 在「連接器配置程式」中開啓現有的配置檔。
- 選取**標準內容**標籤。
- 在「標準內容」標籤的 **BrokerType** 欄位中，選取適合您分配管理系統的值。當您變更現有的值，內容畫面的可用標籤和欄位選項會立刻變更，僅顯示專屬於您所選取的新建分配管理系統之標籤和欄位。

---

## 完成配置

在建立連接器的配置檔並修改它之後，請確定連接器啓動後能夠找到該配置檔。

做法是，開啓連接器使用的啓動檔，然後驗證連接器配置檔使用的位置和檔名是否完全符合您指定給檔案的名稱，以及檔案所在的目錄和路徑。

---

## 在全球化環境下使用連接器配置程式

「連接器配置程式」已全球化，可以處理配置檔和整合分配管理系統之間的字元轉換。「連接器配置程式」使用原生編碼方式。當它寫入配置檔時，它是使用 UTF-8 編碼方式。

「連接器配置程式」支援下列項目中的非英文字元：

- 所有值欄位
- 日誌檔和追蹤檔路徑 (在**追蹤/日誌檔**標籤中指定)

CharacterEncoding 和 Locale 標準配置內容的下拉清單僅顯示受支援值的子集。若要將其他的值加入下拉清單中，您必須手動修改產品目錄中的 `\Data\Std\stdConnProps.xml` 檔。

例如，若要將語言環境 `en_GB` 加入 Locale 內容的值清單中，請開啓 `stdConnProps.xml` 檔，並新增下面以粗體字型來顯示的那一行：

```
<Property name="Locale"
isRequired="true"
updateMethod="component restart">
  <ValidType>String</ValidType>
  <ValidValues>
    <Value>ja_JP</Value>
    <Value>ko_KR</Value>
```

```
<Value>zh_CN</Value>
<Value>zh_TW</Value>
<Value>fr_FR</Value>
<Value>de_DE</Value>
<Value>it_IT</Value>
<Value>es_ES</Value>
<Value>pt_BR</Value>
<Value>en_US</Value>
<Value>en_GB</Value>
<DefaultValue>en_US</DefaultValue>
</ValidValues>
</Property>
```

---

## 附錄 C. Web 服務配接器指導教學

- 『關於指導教學』
- 第 186 頁的『開始之前』
- 第 186 頁的『安裝及配置』
- 第 192 頁的『執行非同步範例實務』
- 第 194 頁的『執行同步範例實務』

本附錄包含下列逐步式程序：

- 示範要求及事件處理程序的非同步及同步事件傳輸
- 說明如何為 SOAP/HTTPS 範例配置 Web 服務連接器
- 說明如何為 SOAP/HTTP 範例配置 Web 服務連接器
- 說明如何為 SOAP/JMS 範例配置 Web 服務連接器

---

### 關於指導教學

此指導教學的目的，是要示範如何針對「Web 服務配接器」的要求及事件處理程序層面，搭配下列各種受支援的通訊協定，來進行同步及非同步的事件傳輸作業：SOAP/HTTP、SOAP/HTTPS 及 SOAP/JMS。於各範例實務中，配接器的作用如下：

- 作為 Web 服務用戶端，讓協同作業能夠呼叫 Web 服務
- 作為 Proxy，將 WebSphere ICS 協同作業顯現為 Web 服務

指導教學的目的是要顯示範例實務中之配接器的基本功能：

- **非同步範例實務**說明非同步 (只有要求) Web 服務及其用戶端和連接器。此範例實務中有二個範例--為方便配置，其使用相同的「Web 服務」連接器，將協同作業顯現為「Web 服務」，並將「Web 服務」當成用戶端來呼叫。
  - **顯現為 Web 服務的協同作業** 在此範例中，Web 服務只是 WebSphere ICS 中的一個協同作業 SERVICE\_ASYNC\_Order\_Collab，被連接器顯現為 Web 服務。該 Web 服務即所謂的 Asynch Order Service。若連接器的配置正確，就可以用下列任何 (一種)「Web 服務」通訊協定來呼叫此「Web 服務」：SOAP/HTTP、SOAP/HTTPS 或 SOAP/JMS。SERVICE\_ASYNC\_Order\_Collab 是採用 SERVICE\_ASYNC\_TLO\_Order 的一種簡式透通協同作業。在此協同作業中負責觸發的埠 (From) 連結至「Web 服務」連接器。服務埠 (To) 則連結至 SampleSiebelConnector。
  - **Web 服務用戶端呼叫的協同作業**在此範例中，Web 服務用戶端是 WebSphere ICS 中的另一個協同作業 CLIENT\_ASYNC\_Order\_Collab，它將會使用「Web 服務」連接器來呼叫 Web 服務 Asynch Order Service。若連接器的配置正確，這個 Web 服務用戶端就可以透過下列任何 (一種)「Web 服務」通訊協定來呼叫「Web 服務」：SOAP/HTTP、SOAP/HTTPS 或 SOAP/JMS。CLIENT\_ASYNC\_Order\_Collab 是採用 CLIENT\_ASYNC\_TLO\_Order 的一種簡式透通協同作業。於此協同作業中負責觸發的埠 (From) 連結至 SampleSAPConnector。服務埠 (To) 則連結至「Web 服務」連接器。

非同步範例實務中的這二個範例，都會用到下列二種應用程式：

- SampleSiebel：為其用戶端建立順序。

- SampleSAP：建立順序。
- **同步範例實務**說明同步 (要求-回應) Web 服務及其用戶端和連接器。此範例實務中有二個範例--為方便配置，其使用相同的「Web 服務」連接器，將協同作業顯現為「Web 服務」，並將「Web 服務」當成用戶端來呼叫。
  - 顯現為 Web 服務的協同作業。在此範例中，「Web 服務」只是 WebSphere ICS 中的一個協同作業 SERVICE\_SYNCH\_OrderStatus\_Collab，被連接器顯現為 Web 服務。在此範例中，該 Web 服務即所謂的 Synch OrderStatus Service。如果連接器的配置正確，就可以用下列任何 Web 服務通訊協定來呼叫 Web 服務：SOAP/HTTP、SOAP/HTTPS 或 SOAP/JMS。SERVICE\_SYNCH\_OrderStatus\_Collab 是採用 SERVICE\_SYNCH\_TLO\_OrderStatus 的一種簡式透通協同作業。在此協同作業中負責觸發的埠 (From) 連結至「Web 服務」連接器。服務埠 (To) 則連結至 SampleSiebelConnector。
  - **Web 服務用戶端呼叫的協同作業**在此範例中，Web 服務用戶端是 WebSphere ICS 中的另一個協同作業 CLIENT\_SYNCH\_OrderStatus\_Collab，它將會使用「Web 服務」連接器來呼叫 Web 服務 Synch OrderStatus Service。若連接器的配置正確，此 Web 服務用戶端就可以透過下列任何 Web 服務通訊協定來呼叫 Web 服務：SOAP/HTTP、SOAP/HTTPS 或 SOAP/JMS。CLIENT\_SYNCH\_OrderStatus\_Collab 是採用 CLIENT\_SYNCH\_TLO\_OrderStatus 的一種簡式透通協同作業。在此協同作業中負責觸發的埠 (From) 連結至 SampleSAPConnector。服務埠 (To) 則連結至「Web 服務」連接器。

同步範例實務中的這二個範例，都會用到下列二種應用程式：

- SampleSiebel：為其用戶端擷取順序的狀態。
- SampleSAP：要求順序的狀態。

這二種範例實務都要使用二個「測試連接器」來模擬 SampleSiebelConnector 及 SampleSAPConnector。

---

## 開始之前

在您開始使用指導教學之前，請先確定：

- 您已安裝 WebSphere ICS 4.2.x 或較新版本，具有這方面的經驗。
- 您已將 WebSphere Business Integration Adapter for Web Services 安裝於 WebSphere ICS 起始目錄中。
- 您具有「Web 服務」這方面技術的經驗。
- 您具有 SOAP 這方面技術的經驗。

---

## 安裝及配置

在接下來的各節中，*WBI\_folder* 是指包含您現行 WebSphere ICS 安裝主體的資料夾。所有環境變數及檔案分隔字元，都是以 Windows NT/2000 的格式來指定。若要在 AIX 或 Solaris 上執行，請做適當的變更。(例如，*WBI\_folder\connectors* 要改成 *WBI\_folder/connectors*)。

### 啟動伺服器及工具

1. 從捷徑來啟動 WebSphere InterChange Server (ICS)。
2. 啟動 WebSphere Business Integration System Manager，並開啓「元件導覽器視景」。

3. 將您的伺服器登錄為 InterChange Server 檢視畫面中的「伺服器實例」，並連接該伺服器。

## 載入範例內容

從「元件瀏覽器視景」：

1. 建立新的「整合元件程式庫」。
2. 匯入 repos 檔案，檔名為 WebServicesSample.jar，位於：  
WBI\_folder\connectors\WebServices\samples\WebSphereICS\

## 編譯協同作業範本

使用 WebSphere Business Integration System Manager：

- 將從 WebServicesSample.jar repos 檔案匯入的「協同作業範本」全部編譯。

## 配置連接器

1. 若您尚未這麼做，請依照本手冊中的說明，並根據您的系統來配置連接器。
2. 使用 WebSphere Business Integration System Manager，開啓「連接器配置程式」中的 WebServicesConnector。
3. 您必須針對您要用於範例的通訊協定，來配置 WebServicesConnector。
  - 如果您要使用 SOAP/HTTP，請參閱『配置 SOAP/HTTP 通訊協定範例實務』來配置 SOAP/HTTP 的連接器。
  - 如果您要使用 SOAP/HTTPS，請參閱第 188 頁的『配置 SOAP/HTTPS 通訊協定範例實務』來配置 SOAP/HTTPS 的連接器。
  - 如果您要使用 SOAP/JMS，請參閱第 189 頁的『配置 SOAP/JMS 通訊協定範例實務』來配置 SOAP/JMS 的連接器。

## 配置 SOAP/HTTP 通訊協定範例實務

本節說明如何配置針對 SOAP/HTTP 範例實務來配置連接器。如本書中的主文所述，連接器包含 SOAP/HTTP 通訊協定接聽器及 SOAP/HTTP-HTTPS 通訊協定處理常式。範例實務將 SERVICE\_ASYNC\_Order\_Collab 及 SERVICE\_SYNC\_OrderStatus\_Collab 協同作業顯現成 SOAP/HTTP Web 服務。若要將協同作業顯現成 SOAP/HTTP Web 服務，連接器會使用 SOAP/HTTP 通訊協定接聽器。CLIENT\_ASYNC\_Order\_Collab 及 CLIENT\_SYNC\_OrderStatus\_Collab 協同作業是 SOAP/HTTP Web 服務的 SOAP/HTTP 用戶端，附有範例實務。若要呼叫 SOAP/HTTP Web 服務，連接器會使用「SOAP/HTTP-HTTPS 通訊協定處理常式」。

在接下來的步驟和說明中，會以 ” 符號來表示階層式連接器配置內容。例如，A” B 表示 A 是階層式內容，B 是 A 的子項內容。

若要為此範例來配置 SOAP/HTTP 通訊協定接聽器，請：

1. 在「連接器配置程式」中，按一下 WebServicesConnector 的**連接器特有內容**。
2. 展開 **ProtocolListenerFramework** 內容，以顯示 ProtocolListeners 子項內容。
3. 展開 **ProtocolListeners** 子項內容，以顯示 **SOAPHTTPListener1** 子項內容。
4. 檢查 **SOAPHTTPListener1”Host** 及 **SOAPHTTPListener1”Port** 內容的值。請確定您的主機上沒有其他處理程序在進行，或是正在接聽此 TCP/IP 埠。您可選擇將 **SOAHTTPListener1”Host** 的值設定成要用來執行連接器的機器名稱。

您不需要為此範例配置 SOAP/HTTP-HTTPS 通訊協定處理常式。

## 配置 SOAP/HTTPS 通訊協定範例實務

本節說明如何配置針對 SOAP/HTTPS 範例實務來配置連接器。連接器包含 SOAP/HTTPS 通訊協定接聽器及 SOAP/HTTP-HTTPS 通訊協定處理常式。範例實務將 SERVICE\_ASYNC\_Order\_Collab 及 SERVICE\_SYNC\_OrderStatus\_Collab 協同作業顯現成 SOAP/HTTPS Web 服務。若要將協同作業顯現成 SOAP/HTTPS Web 服務，連接器會使用 SOAP/HTTPS 通訊協定接聽器。 CLIENT\_ASYNC\_Order\_Collab 及 CLIENT\_SYNC\_OrderStatus\_Collab 協同作業是 SOAP/HTTPS Web 服務的 SOAP/HTTPS 用戶端，附有範例實務。若要呼叫 SOAP/HTTPS Web 服務，連接器會使用 SOAP/HTTPHTTPS 通訊協定處理常式。

在接下來的步驟和說明中，會以 ” 符號來表示階層式連接器配置內容。例如，A” B 表示 A 是階層式內容，B 是 A 的子項內容。

**註：**除了第 186 頁的『開始之前』中所列的預先安裝項目之外，您應該也已經使用您的「金鑰」和「憑證」管理軟體來建立並測試您的金鑰儲存庫及信任儲存庫。

**配置 SSL 連接器特有的內容：**若為 SOAP/HTTPS，則連接器需要您配置 SSL 連接器特有的階層式內容。

1. 在「連接器配置程式」中，按一下 WebServicesConnector 的**連接器特有內容**標籤。
2. 展開 **SSL** 階層式內容，以檢視其所有的子項內容。另外，請檢查或變更階層式 SSL 連接器特有內容的下列子項內容。
  - **SSL” KeyStore** 設定為金鑰儲存庫檔的完整路徑 (您必須用您的「金鑰」和「憑證」管理軟體來建立金鑰儲存庫檔)。
  - **SSL”KeyStorePassword** 設定為存取 KeyStore 所必需的密碼。
  - **SSL”KeyStoreAlias** 設定為 KeyStore 中之私密金鑰的別名。
  - **SSL”TrustStore** 設定為信任儲存庫檔的完整路徑 (您用「金鑰」和「憑證」管理軟體來建立的信任儲存庫檔)。
  - **SSL”TrustStorePassword** 設定為存取 TrustStore 所必需的密碼。

**註：**請不要忘記在「連接器配置程式」中儲存變更。

## 配置 SOAP/HTTPS 通訊協定接聽器：

1. 在「連接器配置程式」中，按一下 WebServicesConnector 的**連接器特有內容**。
2. 展開 **ProtocolListenerFramework** 內容，以顯示 **ProtocolListeners** 子項內容。
3. 展開 **ProtocolListeners** 子項內容，以顯示 **SOAPHTTPSListener1** 子項內容。檢查 **SOAPHTTPSListener1”Host** 及 **SOAPHTTPSListener1”Port** 內容的值。請確定您的主機上沒有其他處理程序在進行，或是正在接聽此 TCP/IP 埠。您可以選擇將 **SOAPHTTPSListener1”Host** 的值設定成用來執行連接器的機器名稱。

您不需要為此範例配置 SOAP/HTTP-HTTPS 通訊協定處理常式。

**設定 KeyStore 及 TrustStore：**您可以快速地設定 KeyStore 及 TrustStore，以用於範例實務。若是正式作業系統，則必須使用協力廠商的軟體來設定及管理金鑰儲存庫，及憑證和金鑰的產生作業。「Web 服務配接器」並未提供任何工具來設定及管理這些資源。



本節假設「Java 虛擬機器」已安裝在您的系統上，且您已熟悉您的 JVM (Java 虛擬機器) 所附的 Keytool。如需詳細資訊，或是有關於 Keytool 的疑難排解問題，請參閱您的 JVM 所附的文件。

### 設定 KeyStore :

1. 使用 Keytool 來建立 KeyStore。您必須在 KeyStore 中建立一對金鑰組。若要這麼做，請在指令行中輸入下列指令：

```
keytool -genkey -alias wsadapter -keystore c:\security\keystore
```

2. Keytool 會立即提示您輸入密碼。請指定您在 SSL”KeyStorePassword 連接器內容中輸入的密碼。

請注意，在上述範例中，若您在指令行中指定了 -keystore c:\security\keystore，就要輸入 c:\security\keystore，以作為 SSL”KeyStore 內容的值。此外，若您在指令行中指定了 -alias wsadapter，就要輸入 wsadapter，以作為 SSL”KeyStoreAlias 連接器內容的值。然後 Keytool 會提示您輸入憑證的詳細資料。下面說明您可以在各個提示中輸入的內容，但這只是範例，請務必參照並遵循 Keytool 文件的指示。

您的姓名為何？

[不明]: HostName

您組織單位的名稱為何？

[不明]: myunit

您組織的名稱為何？

[不明]: myorganization

您的城市或地區名稱為何？

[不明]: mycity

您的省名或市名為何？

[不明]: mystate

此單位的二字母國碼為何？

[不明]: mycountryIs <CN=HostName, OU=myunit, O=myorganization,  
L=mycity, ST=mystate, C=mycountry> 正確嗎？

[no]: 是

3. 請注意您的姓名為何？這個問題，您應輸入您執行連接器所在之機器的名稱。然後 Keytool 會提示您：

請輸入 <wsadapter> 的金鑰密碼 (如果和金鑰儲存庫密碼相同，則按 RETURN)：

4. 若要使用相同的密碼，請按 **Return** 鍵。如果您要使用自行簽章的憑證，您也許會想要匯出上面所建立的憑證。若要這麼做，請在指令行中輸入下列指令：

```
C:\security>keytool -export -alias wsadapter -keystore c:\security\keystore  
-file c:\security\wsadapter.cer
```

5. 現在 Keytool 會提示您輸入金鑰儲存庫密碼。請輸入您在上邊輸入的密碼。

### 設定 TrustStore :

1. 若要將受信任的憑證匯入 TrustStore，請輸入下列指令：

```
keytool -import -alias trusted1 -keystore c:\security\truststore  
-file c:\security\wsadapter.cer
```

2. 現在 Keytool 會提示您輸入金鑰儲存庫密碼。如果您輸入 -keystore c:\security\truststore，請確定 SSL”TrustStore 內容的設定為 c:\security\truststore。此外，也請將 SSL ” TrustStorePassword 內容的值設定為您在上邊輸入的密碼。

## 配置 SOAP/JMS 通訊協定範例實務

本節說明如何配置針對 SOAP/JMS 範例實務來配置連接器。範例實務將 SERVICE\_ASYNC\_Order\_Collab 及 SERVICE\_SYNC\_OrderStatus\_Collab 協同作業顯現成 SOAP/JMS Web 服務。若要將協同作業顯現成 SOAP/JMS Web 服務，連接器會使

用 SOAP/JMS 通訊協定接聽器。 CLIENT\_ASYNCH\_Order\_Collab 及 CLIENT\_SYNCH\_OrderStatus\_Collab 協同作業是 SOAP/JMS Web 服務的 SOAP/JMS 用戶端，附有範例實務。若要呼叫 SOAP/JMS Web 服務，連接器會使用 SOAP/JMS 通訊協定處理常式。

在接下來的步驟和說明中，會以 ” 符號來表示階層式連接器配置內容。例如，A” B 表示 A 是階層式內容，B 是 A 的子項內容。

**註：**除了第 186 頁的『開始之前』中所列的預先安裝項目之外，您應該也已經安裝 JMS 服務提供者，並已安裝及配置您的 JNDI。

**配置 JNDI 內容：** 若為 SOAP/JMS，則您必須配置 JNDI 連接器配置內容：

1. 在「連接器配置程式」中，按一下 WebServicesConnector 的**連接器特有內容**。
2. 展開 JNDI 階層式內容，以顯示其子項內容。然後檢查或變更子項內容，以符合下面所列的值。
  - **JNDI”JNDIProviderURL** 將這個內容設定為「JNDI 服務」提供者的 URL。請參閱您的 JNDI 提供者文件。
  - **JNDI”InitialContextFactory** 將這個內容設定為要用來建立 JNDI 起始環境定義之 Factory 類別的完整類別名稱。請參閱您的 JNDI 提供者文件。
  - **JNDI”JNDIConnectionFactoryName** 將這個內容設定為所要查閱之 Connection Factory 的 JNDI 名稱 (使用 JNDI 環境定義)。請確定可使用 JNDI 來查閱此名稱。
  - 請參閱您的 JNDI 文件，以查看您的 JNDI 提供者是否需要下列任何內容：
    - **JNDI”CTX\_ObjectFactories**
    - **JNDI”CTX\_ObjectFactories**
    - **JNDI”CTX\_StateFactories**
    - **JNDI”CTX\_URLPackagePrefixes**
    - **JNDI”CTX\_DNS\_URL**
    - **JNDI”CTX\_Authoritative**
    - **JNDI”CTX\_Batchsize**
    - **JNDI”CTX\_Referral**
    - **JNDI”CTX\_SecurityProtocol**
    - **JNDI”CTX\_SecurityAuthentication**
    - **JNDI”CTX\_SecurityPrincipal**
    - **JNDI”CTX\_SecurityCredentials**
    - **JNDI”CTX\_Language**
3. 在「連接器配置程式」中儲存變更。

**配置 JMS 佇列及 SOAP/JMS 通訊協定接聽器：** 此範例實務需要向您的 JMS 服務提供者定義 6 個佇列。在執行之前，請先檢查您的 JMS 提供者文件；定義佇列的方式會因提供者而有所不同。

1. 定義下列佇列 (或透過 JNDI 查閱功能來啟用)：
  - ORDER\_INPUT
  - ORDER\_INPROGRESS

- ORDER\_ERROR
  - ORDER\_ARCHIVE
  - ORDER\_UNSUBSCRIBED
  - ORDER\_REPLYTO
2. 從 CSM 來開啓「連接器配置程式」中的 **WebServicesConnector**。若您尚未這麼做，請依照您系統所適用之安裝手冊的說明來配置連接器。
  3. 按一下「連接器配置程式」中的**應用程式配置內容**。
  4. 展開 **ProtocolListenerFramework** 內容，以顯示 ProtocolListeners 子項內容。
  5. 展開 **ProtocolListeners** 內容，以顯示 SOAPJMSListener1 子項內容。
  6. 檢查或變更 **SOAPJMSListner1** 子項內容的值，以符合下列各項：
    - **SOAPJMSListener1”Protocol** 設定為 soap/jms
    - **SOAPJMSListener1”Protocol** 設定為 soap/jms
    - **SOAPJMSListener1”InputQueue** 設定為 ORDER\_INPUT
    - **SOAPJMSListener1”InProgressQueue** 設定為 ORDER\_INPROGRESS
    - **SOAPJMSListener1”ArchiveQueue** 設定為 ORDER\_ARCHIVE
    - **SOAPJMSListener1”UnsubscribedQueue** 設定為 ORDER\_UNSUBSCRIBED
    - **SOAPJMSListener1”ErrorQueue** 設定為 ORDER\_ERROR
    - **SOAPJMSListener1”ReplyToQueue** 設定為 ORDER\_REPLYTO
  7. 在「連接器配置程式」中儲存變更。

#### 配置 SOAP/JMS 通訊協定處理常式:

1. 從「系統管理程式」來開啓「連接器配置程式」中的 **WebServicesConnector**。若您尚未這麼做，請依照您系統所適用之安裝手冊的說明來配置連接器。
2. 按一下「連接器配置程式」中的**連接器配置內容**。
3. 展開 **ProtocolHandlerFramework** 內容，以顯示 ProtocolHandlers 子項內容。
4. 展開 **ProtocolHandlers** 子項內容，以顯示 SOAPJMShandler 子項內容。檢查或變更 SOAPJMShandler 子項內容的值，以符合下列各項：
  - **SOAPJMShandler”Protocol** 設定為 soap/jms
  - **SOAPJMShandler”ReplyToQueue** 設定為 ORDER\_REPLYTO 值
5. 在「連接器配置程式」中儲存變更。

## 建立使用者專案

- 使用 WebSphere Business Integration System Manager，建立新的**使用者專案**。選取在第 187 頁的『載入範例內容』中建立之「整合元件程式庫」中的所有元件。

## 新增及部署專案

1. 在「伺服器實例」檢視畫面中，將在『建立使用者專案』中建立的**使用者專案**新增至 WebSphere ICS。
2. 將此「使用者專案」中的所有元件部署至 ICS。

## 重新啓動 ICS

1. 重新啓動 ICS，以確定所有變更皆生效。

2. 使用「系統監督」工具來確定所有協同作業物件、連接器控制程式和對映都在綠色狀態。

---

## 執行非同步範例實務

此範例實務會呼叫 Asynch Order Service Web 服務。在執行範例實務之前，請先檢閱其資料流程的這個逐步概要說明。

1. 在某個「測試連接器」實例中執行的應用程式 SampleSAP 中，產生一個 CLIENT\_ASYNC\_TLO\_Order.Create 事件。
2. 該事件從 SampleSAP 傳送到協同作業 CLIENT\_ASYNC\_Order\_Collab。
3. 然後該事件從協同作業傳送到「Web 服務」連接器。
4. 「Web 服務」連接器找到 CLIENT\_ASYNC\_TLO\_Order 物件的子項 CLIENT\_ASYNC\_Order 物件。
5. 使用 SOAP 資料處理常式，將「要求」商業物件被轉換成 SOAP 訊息。
6. 「Web 服務」連接器將「SOAP 訊息」傳送到 Web 服務 Asynch Order Service 的端點 (Destination)。端點是由 Protocol Config Meta 物件 (MO) 的 Destination 屬性來提供。連接器所使用的 Protocol Config MO，需視 CLIENT\_ASYNC\_TLO\_Order 的 Handler 屬性值而定。若設定為 soap/http 或 soap/https，CLIENT\_ASYNC\_Order\_SOAP\_HTTP\_CfgMO 的 Destination 屬性就會以 Web 服務的 URL 來提供端點。若 Handler 屬性的設定為 soap/jms，CLIENT\_ASYNC\_Order\_SOAP\_JMS\_CfgMO 的 Destination 屬性就會以目的地佇列名稱來提供端點。
7. Asynch Order Service Web 服務接收 SOAP 要求。如前述，「Web 服務」連接器是這個 Web 服務的端點。在該端點 (之前要求被傳送到這個端點) 上接聽之連接器的通訊協定接聽器會接收 SOAP 訊息。
8. 連接器將 SOAP 訊息轉換成 SERVICE\_ASYNC\_Order，然後建立 SERVICE\_TLO\_Order 物件。SERVICE\_ASYNC\_Order 物件被設定成 SERVICE\_TLO\_Order 物件的子項。
9. 「Web 服務」連接器現在會以非同步的方式，將 SERVICE\_TLO\_Order 物件發佈至 ICS。如此便完成非同步 Web 服務呼叫作業。

因為這是非同步的 Web 服務 (只有要求)，所以不會傳回回應至 Web 服務用戶端。SERVICE\_ASYNC\_Order\_Collab 收到這個物件之後，協同作業會將該商業物件傳送到一個名為 SampleSiebel 的應用程式，該應用程式是當作「測試連接器」的第二個實例來執行。物件會顯示在「測試連接器」上。若從 SampleSiebel 應用程式選取 Reply Success，就會將該事件傳回 SERVICE\_ASYNC\_Order\_Collab。

### 執行非同步範例實務：

1. 啟動您的 ICS 整合分配管理系統 (若尚未執行的話)。
2. 啟動「Web 服務」連接器。
3. 啟動兩個「測試連接器」實例。
4. 使用「測試連接器」，定義 SampleSAPConnector 及 SampleSiebelConnector 的設定檔。
5. 在各個「測試連接器」中選取 **FILE"CONNECT AGENT**，開始模擬代理程式。
6. 在使用「測試連接器」來模擬 SampleSAPConnector 時，請從功能表選取 **EDIT"LOAD BO**。請載入下列檔案：

WBI\_folder\connectors\WebServices\samples\WebSphereICS\OrderStatus  
\CLIENT\_ASYNC\_TLO\_Order.bo

「測試連接器」應該會顯示 CLIENT\_ASYNC\_TLO\_Order 已載入。

7. 驗證 Web 服務端點位址：

- 若是 **SOAP/HTTP Web 服務**如果您想要使用 SOAP/HTTP：
    - a. 請確定您已為 SOAP/HTTP 配置「Web 服務」連接器。在您的「測試連接器」中，確定 CLIENT\_ASYNC\_TLO\_Order 商業物件的 Handler 屬性值設定為 soap/http。此值中不允許使用引號。
    - b. 展開 CLIENT\_ASYNC\_TLO\_Order 的 Request 屬性。此屬性的類型為 CLIENT\_ASYNC\_Order 商業物件。
    - c. 展開 CLIENT\_ASYNC\_Order 的 SOAPHTTPCfgMO 屬性。此屬性的類型為 CLIENT\_ASYNC\_Order\_SOAP\_HTTP\_CfgMO。
    - d. 請確定 CLIENT\_ASYNC\_Order\_SOAP\_HTTP\_CfgMO 的 Destination 屬性值設定為 http://localhost:8080/wbia/webservices/samples。此值中不允許使用引號。
  - 若是 **SOAP/HTTPS Web 服務** 如果您想要使用 SOAP/HTTPS：
    - a. 請確定您已為 SOAP/HTTPS 配置「Web 服務」連接器。在您的「測試連接器」中，確定 CLIENT\_ASYNC\_TLO\_Order 商業物件的 Handler 屬性值設定為 soap/https。此值中不允許使用引號。
    - b. 展開 CLIENT\_ASYNC\_TLO\_Order 的 Request 屬性。此屬性的類型為 CLIENT\_ASYNC\_Order 商業物件。
    - c. 展開 CLIENT\_ASYNC\_Order 的 SOAPHTTPCfgMO 屬性。此屬性的類型為 CLIENT\_ASYNC\_Order\_SOAP\_HTTP\_CfgMO。
    - d. 請確定 CLIENT\_ASYNC\_Order\_SOAP\_HTTP\_CfgMO 的 Destination 屬性值設定為 https://localhost:8443/wbia/webservices/samples。此值中不允許使用引號。
  - 若是 **SOAP/JMS Web 服務**如果您想要使用 SOAP/JMS：
    - a. 請確定您已為 SOAP/JMS 配置「Web 服務」連接器。在您的「測試連接器」中，確定 CLIENT\_ASYNC\_TLO\_Order 商業物件的 Handler 屬性值設定為 soap/jms。此值中不允許使用引號。
    - b. 展開 CLIENT\_ASYNC\_TLO\_Order 的 Request 屬性。此屬性的類型為 CLIENT\_ASYNC\_Order 商業物件。
    - c. 展開 CLIENT\_ASYNC\_Order 的 SOAPJMSCfgMO 屬性。此屬性的類型為 CLIENT\_ASYNC\_Order\_SOAP\_JMS\_CfgMO。
    - d. 請確定 CLIENT\_ASYNC\_Order\_SOAP\_JMS\_CfgMO 的 Destination 屬性值設定為 ORDER\_INPUT。此值中不允許使用引號。
8. 在以「測試連接器」來模擬 SampleSAPConnector 時，請按一下所載入的**測試 BO**。從功能表選取 **REQUEST”SEND**。請參閱本節前面的逐步概要說明，以取得有關事件流程的詳細資料。
9. 在以「測試連接器」來模擬 SampleSiebelConnector 時，請選取 **REQUEST”ACCEPT REQUEST**。「測試連接器」的右邊畫面中，會顯示標示為 SERVICE\_ASYNC\_TLO\_Order.Create 的事件。
10. 按兩下商業物件。商業物件會在視窗中開啓。

11. 展開商業物件的 Request 屬性。Request 屬性的類型為 SERVICE\_ASYNC\_Order。檢查 SERVICE\_ASYNC\_Order 的 OrderId、自訂及其他屬性，以驗證所收到的「順序」。如此便完成執行非同步範例實務。
12. 一旦檢查完商業物件之後，請關閉視窗。選取 **REQUEST "REPLY" SUCCESS**。

---

## 執行同步範例實務

此範例實務會呼叫 Synch OrderStatus Service Web 服務。執行範例實務之前，請先檢閱其資料流程的這個逐步概要說明。

1. 在某個「測試連接器」實例中執行的應用程式 SampleSAP 中，產生一個 CLIENT\_SYNC\_TLO\_OrderStatus.Retrieve 事件。
2. 該事件從 SampleSAP 傳送到協同作業 CLIENT\_SYNC\_OrderStatus\_Collab。
3. 然後該事件從協同作業傳送到「Web 服務」連接器。
4. 「Web 服務」連接器找到 CLIENT\_SYNC\_TLO\_OrderStatus 物件的子項 CLIENT\_SYNC\_OrderStatus\_Request 物件。
5. 「Web 服務」連接器會呼叫 SOAP 資料處理常式，將 CLIENT\_SYNC\_OrderStatus\_Request 商業物件轉換成 SOAP 訊息。
6. 「Web 服務」連接器將 SOAP 訊息傳送到 Web 服務 Synch OrderStatus Service 的端點 (Destination)。端點是由 Protocol Config MO 的 Destination 屬性來提供。連接器所使用的 Protocol Config MO，需視 CLIENT\_SYNC\_TLO\_OrderStatus 的 Handler 屬性值而定。若設定為 soap/http 或 soap/https，CLIENT\_SYNC\_OrderStatus\_Request\_SOAP\_HTTP\_CfgMO 的 Destination 屬性就會以 Web 服務的 URL 來提供端點。若 Handler 屬性的設定為 soap/jms，CLIENT\_SYNC\_OrderStatus\_Request\_SOAP\_JMS\_CfgMO 的 Destination 屬性就會以 Web 服務的目的地佇列名稱來提供端點。
7. Web 服務 Synch OrderStatus Service 接收 SOAP 要求。如前述，「Web 服務」連接器是目標端點。在該端點 (之前要求被傳送到這個端點) 上接聽之連接器的通訊協定接聽器會接收 SOAP 訊息。
8. 連接器以 SOAP 訊息來呼叫 SOAP 資料處理常式。SOAP 資料處理常式將 SOAP 訊息轉換成 SERVICE\_SYNC\_OrderStatus\_Request 物件。然後「Web 服務」連接器會建立 SERVICE\_TLO\_OrderStatus 物件。SERVICE\_SYNC\_OrderStatus\_Request 物件被設定成 SERVICE\_TLO\_OrderStatus 物件的子項。
9. 「Web 服務」連接器現在會以同步的方式，將 SERVICE\_TLO\_OrderStatus 物件發佈到在 WebSphere ICS 中執行的 SERVICE\_SYNC\_OrderStatus\_Collab 協同作業。由於這是同步的執行作業，所以「Web 服務」連接器會維持暫停執行的狀態，直到協同作業執行並傳回回應。
10. SERVICE\_SYNC\_OrderStatus\_Collab 接收 SERVICE\_TLO\_OrderStatus 物件。然後協同作業會將商業物件傳送到應用程式 SampleSiebel，該應用程式是當作「測試連接器」的第二個實例來執行。
11. 若您從 SampleSiebel 應用程式選取 Reply Success，就會將事件傳回 SERVICE\_SYNC\_OrderStatus\_Collab 協同作業。
12. SERVICE\_SYNC\_OrderStatus\_Collab 接收 SERVICE\_TLO\_OrderStatus 物件。然後協同作業會將商業物件傳送到「Web 服務」連接器。
13. 「Web 服務」連接器找到 SERVICE\_SYNC\_OrderStatus\_TLO 的子項 SERVICE\_SYNC\_OrderStatus\_Response 商業物件 (或

SERVICE\_SYNCH\_OrderStatus\_Fault，若已移入)。SOAP 資料處理常式會將此商業物件轉換成 SOAP 回應訊息 (或 SOAP 錯誤訊息)。

14. 「Web 服務」連接器將 SOAP 回應訊息 (或 SOAP 錯誤訊息) 傳回 Web 服務用戶端。
15. Web 服務用戶端 (在本例中是連接器) 接收回應。連接器以回應訊息來呼叫 SOAP 資料處理常式。
16. SOAP 資料處理常式將回應訊息轉換成 CLIENT\_SYNCH\_OrderStatus\_Response 或 CLIENT\_SYNCH\_OrderStatus\_Fault 商業物件 (需視 Order Synch Service 傳回的內容而定)。「Web 服務」連接器將這個物件設定成 CLIENT\_SYNCH\_OrderStatus\_TLO 的子項。CLIENT\_SYNCH\_OrderStatus\_TLO 被傳回 CLIENT\_SYNCH\_OrderStatus\_Collab 協同作業。
- 17) 然後 CLIENT\_SYNCH\_OrderStatus\_Collab 將 CLIENT\_SYNCH\_OrderStatus\_TLO 傳送到 SampleSAP 應用程式 (此應用程式是當作「測試連接器」的第一個實例來執行)。「測試連接器」會顯示這個物件。

#### 執行同步範例實務：

1. 啟動您的 ICS 整合分配管理系統 (若尚未執行的話)。
2. 啟動「Web 服務」連接器。
3. 啟動兩個「測試連接器」實例。
4. 使用「測試連接器」，定義 SampleSAPConnector 及 SampleSiebelConnector 的設定檔。
5. 在各個「測試連接器」中選取 **FILE”CONNECT AGENT**，開始模擬代理程式。
6. 當使用「測試連接器」模擬 SampleSAPConnector 時，請從功能表選取 **EDIT”LOAD BO**。請載入下列檔案：

```
WBI_folder\connectors\WebServices\samples\WebSphereICS\OrderStatus  
\CLIENT_SYNCH_TLO_OrderStatus.bo
```

「測試連接器」應該會顯示 CLIENT\_SYNCH\_TLO\_OrderStatus 已載入。

7. 驗證 Web 服務端點位址：
  - 若是 **SOAP/HTTP Web 服務**如果您想要使用 SOAP/HTTP：
    - a. 請確定您已為 SOAP/HTTP 配置「Web 服務」連接器。在您的「測試連接器」中，確定 CLIENT\_SYNCH\_TLO\_OrderStatus 商業物件的 Handler 屬性值設定為 soap/http。此值中不允許使用引號。
    - b. 展開 CLIENT\_SYNCH\_TLO\_OrderStatus 的 Request 屬性。此屬性的類型為 CLIENT\_SYNCH\_OrderStatus 商業物件。
    - c. 展開 CLIENT\_SYNCH\_OrderStatus 的 SOAPHTTPCfgMO 屬性。此屬性的類型為 CLIENT\_SYNCH\_OrderStatus\_SOAP\_HTTP\_CfgMO。
    - d. 請確定 CLIENT\_SYNCH\_OrderStatus\_SOAP\_HTTP\_CfgMO 的 Destination 屬性值設定為 http://localhost:8080/wbia/webservices/samples。此值中不允許使用引號。
  - 若是 **SOAP/HTTPS Web 服務**如果您想要使用 SOAP/HTTPS：
    - a. 請確定您已為 SOAP/HTTPS 配置「Web 服務」連接器。在您的「測試連接器」中，確定 CLIENT\_SYNCH\_TLO\_OrderStatus 商業物件的 Handler 屬性值設定為 soap/https。此值中不允許使用引號。

- b. 展開 CLIENT\_SYNCH\_TLO\_OrderStatus 的 Request 屬性。此屬性的類型為 CLIENT\_SYNCH\_OrderStatus 商業物件。
  - c. 展開 CLIENT\_SYNCH\_OrderStatus 的 SOAPHTTPCfgMO 屬性。此屬性的類型為 CLIENT\_SYNCH\_OrderStatus\_SOAP\_HTTP\_CfgMO。
  - d. 請確定 CLIENT\_SYNCH\_OrderStatus\_SOAP\_HTTP\_CfgMO 的 Destination 屬性值設定為 https://localhost:8443/wbia/webservices/samples。此值中不允許使用引號。
- 若是 **SOAP/JMS Web 服務**如果您想要使用 SOAP/JMS：
    - a. 請確定您已為 SOAP/JMS 配置「Web 服務」連接器。在您的「測試連接器」中，確定 CLIENT\_SYNCH\_TLO\_OrderStatus 商業物件的 Handler 屬性值設定為 soap/jms。此值中不允許使用引號。
    - b. 展開 CLIENT\_SYNCH\_TLO\_OrderStatus 的 Request 屬性。此屬性的類型為 CLIENT\_SYNCH\_OrderStatus 商業物件。
    - c. 展開 CLIENT\_SYNCH\_OrderStatus 的 SOAPJMSCfgMO 屬性。此屬性的類型為 CLIENT\_SYNCH\_OrderStatus\_SOAP\_JMS\_CfgMO。
    - d. 請確定 CLIENT\_SYNCH\_OrderStatus\_SOAP\_JMS\_CfgMO 的 Destination 屬性值設定為 ORDER\_INPUT。此值中不允許使用引號。
8. 在以「測試連接器」來模擬 SampleSAPConnector 時，請按一下所載入的**測試 BO**。從功能表選取**要求”傳送**。請參閱本節前面的逐步概要說明，以取得有關資料流程的詳細資料。
  9. 模擬 SampleSiebelConnector 之「測試連接器」實例的右邊畫面中，會顯示標示為 SERVICE\_SYNCH\_TLO\_OrderStatus.Retrieve 的事件。按兩下商業物件，以將其顯示於視窗中。
  10. 展開商業物件的 Request 屬性。Request 屬性的類型為 SERVICE\_SYNCH\_OrderStatus\_Request。檢查 SERVICE\_ASYNC\_Order 的 OrderId 屬性，以驗證這是所需的順序狀態。
    - 如果您知道順序的狀態，請：
      - a. 按一下 SERVICE\_SYNCH\_TLO\_OrderStatus 的 **Response** 屬性。Response 屬性的類型為 CLIENT\_SYNCH\_OrderStatus\_Response。
      - b. 用滑鼠右鍵按一下 **Response** 屬性。
      - c. 按一下**新增實例**選項，就會建立一個新的 CLIENT\_SYNCH\_OrderStatus\_Response 商業物件實例。
      - d. 輸入 **OrderId**、**CustomerId** 的值，以及您知道有關此順序的所有其他詳細資料。輸入完此順序的所有詳細資料之後，關閉此視窗。
    - 如果您不知道順序的狀態，請：
      - a. 按一下 SERVICE\_SYNCH\_TLO\_OrderStatus 的 **Fault** 屬性。Fault 屬性的類型為 CLIENT\_SYNCH\_OrderStatus\_Fault。
      - b. 用滑鼠右鍵按一下 **Fault** 屬性。
      - c. 按一下**新增實例**選項，就會建立一個新的 CLIENT\_SYNCH\_OrderStatus\_Fault 實例。
      - d. 輸入 faultcode、faultstring 的值，以及要放在 SOAP 錯誤訊息中以傳送的所有其他詳細資料。輸入完此錯誤的所有值之後，關閉此視窗。



11. 選取 **REQUEST”REPLY”SUCCESS**。模擬 SampleSAPConnector 之「測試連接器」的右邊畫面中，會顯示標示為 SERVICE\_SYNCH\_TLO\_OrderStatus.Retrieve 的事件。
12. 按兩下 **SERVICE\_SYNCH\_TLO\_OrderStatus.Retrieve** 商業物件，就會在視窗中顯示該商業物件。
  - 如果您的 SampleSiebelConnector 傳回順序狀態，應該會看到商業物件的 **Response** 屬性已移入資料。展開 **Response** 屬性，以驗證順序狀態。
  - 如果您的 SampleSiebelConnector 傳回錯誤，應該會看到商業物件的 **Fault** 屬性已移入資料。展開 **Fault** 屬性，以判斷錯誤。
13. 一旦檢查完商業物件之後，請關閉視窗。選取 **REQUEST ”REPLY” SUCCESS**。如此便完成執行同步範例實務。



---

## 附錄 D. 移轉至 3.0.x

- 『反向相容性』
- 『升級作業』
- 『Web 服務產生公用程式』
- 『SOAP 資料處理常式』
- 第 200 頁的『SOAP 連接器』
- 第 200 頁的『Web 服務連接器』

此附錄說明 Web 服務配接器 3.0.x 版的反向相容性，及如何從 Web 服務配接器 1.x 及 2.x 版來移轉。

---

### 反向相容性

Web 服務配接器 3.0.x 版無法反向相容：

- 不管是一起使用，還是分開使用，所有新元件 (Web 服務連接器、SOAP 資料處理常式、WSDL ODA) 皆無法與本產品的舊版元件搭配使用。
- 不管是一起使用，還是分開使用，本產品的所有舊版元件 (SOAP 連接器、SOAP 資料處理常式、Web 服務產生公用程式) 皆無法與 3.0.x 搭配使用。
- 以本產品舊版解決方案來建立或使用的成品，可能亦無法使用於 3.0.x 中。
- 本產品的 3.0.x 版無法使用於 4.2 以前的 WebSphere ICS 版本。

---

### 升級作業

下面各段將說明如何將本產品 1.x 及 2.x 版的元件升級。

#### Web 服務產生公用程式

已不再提供「Web 服務產生公用程式」。請改用「系統管理程式」工具將協同作業顯現為 Web 服務。「Web 服務產生公用程式」產生的所有成品皆無法使用於此版本：

- **Proxy 類別**現在 Web 服務連接器可以支援事件通知。所以不需要 Proxy 類別，便可將協同作業顯現為 Web 服務了。以「Web 服務產生公用程式」產生的 Proxy 類別，無法呼叫以「系統管理程式 4.2 版」工具來顯現成 Web 服務的協同作業。
- **WSDL 文件**「Web 服務產生公用程式」無法使用產生 ICS 4.2 版協同作業的 WSDL 文件。您必須改用「系統管理程式」工具來產生 WSDL 文件。如果需要詳細資訊，請參閱第 134 頁的『執行精靈』。  
當您為要求處理程序啟用協同作業時，您使用「Web 服務產生公用程式」來產生的 WSDL 文件可能無法用於 3.0.x 版連接器所提供的 WSDL ODA。

#### SOAP 資料處理常式

您只能將 SOAP 資料處理常式用於 Web 服務連接器。任何其他連接器或「伺服器存取介面」皆無法使用這個資料處理常式。下面各段將討論其他的支援問題。

## Meta 物件

不再支援用於舊版的最上層 SOAP 資料處理常式 Meta 物件。您必須建立新的最上層 Meta 物件，以用於 3.0.x 版的 SOAP 資料處理常式。這個 Meta 物件必須只有 Classname 及 SOAPNameHandler 屬性。

新的 Meta 物件不再需要子項 Meta 物件，就可進行從 SOAP 訊息轉換成商業物件的作業，以及從商業物件轉換成 SOAP 訊息的作業。因此，請確定您的最上層 Meta 物件沒有 SOAPToBOConfigMO 及 BOToSOAPConfigMO 屬性。

於上述從 SOAP 訊息轉換成商業物件和從商業物件轉換成 SOAP 訊息的作業中提到的子項 Meta 物件，現在必須新增為 SOAP 要求、SOAP 回應及 SOAP 錯誤商業物件的子項。如需進一步資訊，請參閱第 97 頁的第 5 章，『SOAP 資料處理常式』及第 19 頁的第 3 章，『商業物件基本需求』。

## 應用程式特有資訊

新 SOAP 資料處理常式的特點，就是有新的應用程式特有資訊 (ASI) 功能。您可將特定的 ASI 新增至 SOAP 商業物件，以利用這項增強功能，但這並不是必要的作業。除了不能將子項 SOAP Config MO 新增至商業物件外，您還可部署您以舊版連接器來建立的 SOAP 商業物件，以用於 3.0.x 版。

## 連接器相容性

您只能將新的 SOAP 資料處理常式用於 3.0.x 版的 Web 服務連接器。新的 SOAP 資料處理常式無法使用於舊版的元件，例如：SOAP 連接器或「伺服器存取介面」。

您不可將舊的 SOAP 資料處理常式用於 3.0.x Web 服務連接器。

## SOAP 連接器

3.0.x 版不支援 SOAP 連接器。您必須使用 Web 服務連接器來呼叫 Web 服務。

## Web 服務連接器

有了 3.0.x 版，您可以使用 Web 服務連接器將協同作業顯現為 Web 服務，並呼叫 Web 服務。新事件通知功能可用來將協同作業顯現成 Web 服務。新要求處理程序的特性現在可用來呼叫 Web 服務。下面各段會強調顯示使用 Web 服務所必須完成的移轉作業。

**註：** 下面所討論的移轉作業可能不夠詳盡。此外，您可使用任何順序來完成這些作業。

## 事件通知

3.0.x 連接器可以同步或非同步的方式來呼叫協同作業，不需要在 Web 伺服器上建立及部署 Proxy 類別。連接器現在有接聽器組織架構，可通知連接器有事件發生。接聽器組織架構可支援 SOAP/HTTP、SOAP/HTTPS 及 SOAP/JMS 連結。若接聽器的配置正確，連接器就可代表已顯現為 Web 服務的協同作業，來偵測及回應 Web 服務。如需有關接聽器組織架構的進一步資訊，及其配置方法，請參閱第 55 頁的『通訊協定接聽器』。

**事件通知的商業物件：** 您必須建立事件通知最上層物件 (TLO)。TLO 是 SOAP 要求商業物件的儲存區，亦可為 (針對同步的要求處理程序) SOAP 回應及 SOAP 錯誤商業物件的儲存區。TLO 的結構元件會預期一項單一的 Web 服務作業：SOAP 要求商業物

件對應於 SOAP 要求訊息，SOAP 回應商業物件對應於 SOAP 回應訊息，SOAP 錯誤商業物件對應於 SOAP 錯誤訊息。如需進一步資訊，請參閱第 20 頁的『處理 TLO 的同步事件』。

**事件通知及 SOAP 商業物件：** 舊版「伺服器存取介面」所使用的 SOAP 商業物件，可用於 3.0.x 版，但需經過修改，如需說明，請參閱第 199 頁的『SOAP 資料處理常式』。請注意，您必須將 SOAP 商業物件指定成事件通知 TLO 中的子項。

## 要求處理程序

就像舊版中的 SOAP 連接器一樣，3.0.x Web 服務連接器可呼叫 Web 服務。此外，新連接器可支援以 SOAP/JMS 連結來呼叫 Web 服務。下面各段將進一步討論連接器要求處理程序中的變更。

**最上層物件要求處理程序：** 您必須建立要求處理程序 TLO。TLO 是 SOAP 要求商業物件的儲存區，亦可為 (針對同步的要求處理程序) SOAP 回應及 SOAP 錯誤商業物件的儲存區。TLO 的結構元件會預期一項單一的 Web 服務作業：SOAP 要求商業物件對應於 SOAP 要求訊息，SOAP 回應商業物件對應於 SOAP 回應訊息，SOAP 錯誤商業物件對應於 SOAP 錯誤訊息。以此概念來看，3.0.x 的要求處理程序 TLO 要對應於舊版 SOAP 連接器所使用的 TLO。如需有關要求處理程序 TLO 的進一步資訊，請參閱第 35 頁的『處理 TLO 的同步要求』。

**SOAP 商業物件：** 可使用舊版 SOAP 連接器所使用的 SOAP 商業物件，但需經過修改，如需說明，請參閱第 199 頁的『SOAP 資料處理常式』。您必須將這些商業物件指定成要求處理程序 TLO 的子項。請注意，在舊版中，這些商業物件是用於 SOAP 連接器之 TLO 的子項。

3.0.x Web 服務連接器對 SOAP 要求商業物件還有一項額外需求。每個 SOAP 要求商業物件都必須要有一個 Protocol Config MO (Meta 物件) 類型的屬性。連接器使用 Protocol Config MO 來判斷要求訊息的目的地。每個 Protocol Config MO 皆有一個 Destination 屬性，以提供目標 Web 服務的位址。若您正使用 SOAP/HTTP 或 SOAP/HTTPS 來呼叫目標 Web 服務，則 Destination 屬性對應於用於舊版 SOAP 連接器之 TLO 的 URL 屬性。如需進一步資訊，請參閱第 25 頁的『Protocol Config MO』。



---

## 附錄 E. 配置 HTTPS/SSL

- 『Keystore 設定』
- 第 204 頁的 『TrustStore 設定』
- 第 204 頁的 『產生公開金鑰憑證的憑證簽章要求 (CSR)』

若您計劃使用 SSL，則必須使用協力廠商的軟體來管理您的金鑰儲存庫、憑證及金鑰產生作業。這些作業的工具開發並不包含 Web 服務連接器。但是您可選擇使用 Keytool (隨附於 IBM JRE) 來建立自行簽章的憑證，以及管理金鑰儲存庫。

Keytool 是一種金鑰和憑證管理公用程式，可讓您管理您自己的公用/私密金鑰組，以及相關的憑證。這些金鑰組和憑證的目的是要用來自行鑑別 (向其他使用者或服務鑑別您自己)，或是用於使用數位簽章的資料完整性和鑑別服務。Keytool 公用程式亦可讓您儲存與您通訊之夥伴的公開金鑰 (以憑證的格式來儲存)。

此附錄說明如何使用 Keytool 來設定金鑰儲存庫。請注意，此附錄僅提供範例來說明；其目的並不是要用來取代 Keytool 或相關產品的文件。請務必參照您用來設定金鑰儲存庫之工具的原始文件。如需 Keytool 的進一步資訊，請參閱：

- <http://java.sun.com/j2se/1.3/docs/tooldocs/tools.html#security>

---

### Keystore 設定

若要使用 Keytool 來建立 KeyStore，您必須先在 KeyStore 中建立金鑰組。假設您輸入下列指令行：

```
keytool -genkey -alias wsadapter -keystore c:\security\keystore
```

Keytool 會立即提示您輸入密碼。您可輸入您所選擇的密碼 (在 Keytool 參數中)，但您應要將輸入 Keytool 中的密碼，指定為 SSL ” KeyStorePassword 連接器內容的值。如需進一步資訊，請參閱第 89 頁的 『KeyStorePassword』。

該範例指令會在 c:\security\keystore 目錄中建立一個名為 keystore 的金鑰儲存庫。因此，您要輸入 c:\security\keystore 作為 SSL ” KeyStore 連接器階層式內容的值。此外，您也要從上述的指令行範例輸入 -alias wsadapter，以作為 SSL ” KeyStoreAlias 連接器階層式內容的值。然後 Keytool 公用程式會提示您輸入憑證的詳細資料。以下說明您可以針對各項提示來輸入的內容。(請參閱 Keytool 文件。)

```
您的姓名為何？
  [不明]:  HostName
您組織單位的名稱為何？
  [不明]:  wbi
您組織的名稱為何？
  [不明]:  IBM
您的城市或地區名稱為何？
  [不明]:  Burlingame
您的省名或市名為何？
  [不明]:  CA
此單位的二字母國碼為何？
  [不明]:  US
<CN=HostName, OU=wbi, O=IBM, L=Burlingame,
ST=CA, C=US> 正確嗎？
  [no]:  是
```

然後 Keytool 會提示您輸入密碼：

請輸入 <wsadapter> 的金鑰密碼 (若與金鑰儲存庫密碼相同，則按 RETURN)：

若要使用相同的密碼，請按 Return 鍵。若您要使用自行簽章的憑證，您也許會想要匯出上面所建立的憑證。若是如此，請在指令行中輸入下列指令：

```
keytool -export -alias wsadapter -keystore c:\security\keystore -file wsadapter.cer
```

現在 Keytool 會提示您輸入金鑰儲存庫密碼。請輸入您在上面輸入的密碼。

---

## TrustStore 設定

您可能會因為下列原因而想要設定 TrustStore：如果您想要由 SOAP/HTTPS 通訊協定接聽器來鑑別 Web 服務用戶端，請將 SSL ” UseClientAuth 連接器配置內容設定為 true。若是如此，SOAP/HTTPS 通訊協定接聽器會預期 TrustStore 包含所有受信任之 Web 服務用戶端的憑證。請注意，連接器會使用 JSSE 預設機制來信任用戶端。若您正呼叫 SOAP/HTTPS Web 服務，則 SOAP/HTTP-HTTPS 通訊協定處理常式會要求 TrustStore 信任 Web 服務。這表示 TrustStore 必須包含所有受信任之 Web 服務的憑證。請注意，連接器會使用 JSSE 預設機制來信任用戶端。若要將受信任的憑證匯入 TrustStore，請輸入如下列的指令：

```
keytool -import -alias trusted1 -keystore c:\security\truststore -file  
c:\security\trusted1.cer
```

現在 Keytool 會提示您輸入金鑰儲存庫密碼。若您輸入 -keystore c:\security\truststore，請確定 SSL ” TrustStore 階層式內容的設定為 c:\security\truststore。此外，亦請將 SSL ” TrustStorePassword 階層式內容的值設定為您之前輸入的密碼。

---

## 產生公開金鑰憑證的憑證簽章要求 (CSR)

若於獲得彼此信任的夥伴之間進行 SSL 資料交換，則自行簽章的憑證即已足夠。但若是由認證管理中心 (CA) 來簽署憑證，較易取得其他人的信任。

若要使用 Keytool 公用程式，以取得由 CA 簽署的憑證，您必須先產生「憑證簽章要求 (CSR)」，再將 CSR 交給 CA。然後 CA 會簽署憑證，並交還給您。

請輸入下列指令，以產生 CSR：

```
keytool -certreq -alias wsadapter -file wsadapter.csr  
-keystore c:\security\keystore
```

於該指令中，alias 是您為私密金鑰所建立的金鑰儲存庫別名。Keytool 公用程式會產生 CSR 檔案，您要將它提供給您的 CA。然後您的 CA 會提供您已簽章的憑證。您必須將此憑證匯入您的金鑰儲存庫。您可輸入下列指令來完成此動作：

```
keytool -import -alias wsadapter -keystore c:\security\keystore -trustcacerts  
-file casignedcertificate.cer
```

一旦匯入了，金鑰儲存庫中的自行簽章憑證就會被取代成 CA 簽章的憑證。



---

## 注意事項

在所有國家中，IBM 不見得有提供本書中所提的各項產品、服務或功能。要知道在您所在之區是否可用到這些產品與服務時，請向當地的 IBM 業務代表查詢。本書在提及 IBM 的產品、程式或服務時，不表示或默示只能使用 IBM 的產品、程式或服務。只要未侵害 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。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。

在這本書或文件中可能包含著 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.

下列段落不適用於英國或任何其他與當地法律相抵觸之國家：

IBM 僅以「現狀」提供本書，而不提供任何明示或默示之保證 (包括但不限於為未涉侵權、可售性或符合特定效用的保證)。有些地區在某些固定的交易上並不接受明示或默示保證的放棄聲明，因此此項聲明可能不適用。

本書中可能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。同時，IBM 得隨時改進及 (或) 變動本書中所提及的產品及 (或) 程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供保證。該網站上的資料，並非本 IBM 產品所用資料的一部分，如因使用該網站而造成損害，其責任由貴客戶自行負責。

IBM 得以各種適當的方式使用或散佈由 貴客戶提供的任何資訊，而無需對您負責。

本程式之獲授權者若希望取得本程式之相關資訊，以便達到下列目的：(i) 在獨立建立的程式與其它程式 (包括本程式) 之間交換資訊；以及 (ii) 相互使用已交換之資訊。則請與位於下列地址之人員聯絡：

IBM Burlingame Laboratory Director  
IBM Burlingame Laboratory  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
U.S.A

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 基於雙方之「IBM 客戶合約」、「國際程式授權合約」或任何同等合約之條款，提供本書中所述之授權程式與其所有適用的授權資料。

任何此處涵蓋的執行效能資料都是在一個受控制的環境下決定出來的。因此，若在其他作業環境下，所得的結果可能會大大不同。有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。再者，有些測定可能已透過推測方式評估過。但實際結果可能並非如此。本書的使用者應依自己的特定環境，查證適用的資料。

本書所提及之非 IBM 產品資訊，係一由產品的供應商，或其出版的聲明或其他公開管道取得。IBM 未必測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。如果您對非 IBM 產品的性能有任何的疑問，請逕向該產品的供應商查詢。

此資訊包含日常企業運作所使用的資料和報告的範例。為求儘可能地完整說明，範例可能包括了個人、公司、品牌和產品的名稱。這些名稱全為虛構，如有雷同，純屬巧合。

有關 IBM 未來動向的任何陳述，僅代表 IBM 的目標而已，並可能於未事先聲明的情況下有所變動或撤回。

#### 版權授權合約

此資訊可能包含原始語言格式的範例應用程式，解說各種作業平台上的程式設計技術。貴客戶得為開發、使用、銷售並散佈運用範例程式之作業平台之應用程式程式介面所撰寫的應用程式之目的，免費複製、修改並散佈這些範例程式。這些範例尚未在所有狀況下完全測試。因此 IBM 不保證或默示這些程式的可靠性、服務性或功能。

---

## 程式設計介面資訊

程式設計介面資訊（如有提供）主要是協助您使用這個程式來建立應用軟體。

一般用途的程式設計介面可讓您撰寫應用軟體來取得這個程式的工具服務。

然而，這項資訊亦可能包含診斷、修改及調整資訊。診斷、修改及調整資訊是為了協助您對應用軟體除錯。

**警告：** 請勿將這項診斷、修改及調整資訊當做程式設計介面來使用，因為隨時有變動。

---

## 商標與服務標記

下列詞彙是 International Business Machines Corporation 在美國及 (或) 其他國家的商標或註冊商標：

IBM  
IBM 標誌  
AIX  
CrossWorlds  
DB2  
DB2 Universal Database  
Domino  
Lotus  
Lotus Notes

MQIntegrator  
MQSeries  
Tivoli  
WebSphere

Microsoft、Windows、Windows NT 及 Windows 標誌皆為 Microsoft Corporation 在美國及 (或) 其他國家的商標。

MMX、Pentium 以及 ProShare 是 Intel Corporation 在美國及 (或) 其他國家的商標或註冊商標。

Java 及所有以 Java 為基礎的商標是 Sun Microsystems, Inc. 在美國及 (或) 其他國家的商標。

其他公司、產品及服務名稱，可能是其他公司的商標或服務標誌。



IBM WebSphere Business Integration Adapter Framework V2.4.0



## 讀者意見表

為使本書盡善盡美，本公司極需您寶貴的意見；懇請您閱讀後，撥冗填寫下表，惠予指教。

請於下表適當空格內，填入記號(√)；我們會在下一版中，作適當修訂，謝謝您的合作!

評估項目	評 估 意 見	備 註
正 確 性	內容說明與實際程序是否符合	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	參考書目是否正確	<input type="checkbox"/> 是 <input type="checkbox"/> 否
一 致 性	文句用語及風格，前後是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	實際產品介面訊息與本書中所提是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
完 整 性	是否遺漏您想知道的項目	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	字句、章節是否有遺漏	<input type="checkbox"/> 是 <input type="checkbox"/> 否
術語使用	術語之使用是否恰當	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	術語之使用，前後是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
可 讀 性	文句用語是否通順	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	有否不知所云之處	<input type="checkbox"/> 是 <input type="checkbox"/> 否
內容說明	內容說明是否詳盡	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	例題說明是否詳盡	<input type="checkbox"/> 是 <input type="checkbox"/> 否
排版方式	本書的形狀大小，版面安排是否方便閱讀	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	字體大小，顏色編排，是否有助於閱讀	<input type="checkbox"/> 是 <input type="checkbox"/> 否
目錄索引	目錄內容之編排，是否便於查找	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	索引語錄之排定，是否便於查找	<input type="checkbox"/> 是 <input type="checkbox"/> 否
※評估意見為"否"者，請於備註欄提供建議。		

其他：(篇幅不夠時，請另外附紙說明。)

---



---



---



---



---



---



---



---



---



---

上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。  
註：您也可將寶貴的意見以電子郵件寄至 [NLSC01@tw.ibm.com](mailto:NLSC01@tw.ibm.com)，謝謝。

IBM WebSphere Business Integration Adapters

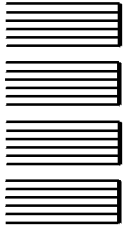
Web 服務配接器使用手冊

V 3.3.x

折疊線

110 台北市信義區松仁路 7 號 3 樓

臺灣國際商業機器股份有限公司  
大中華研發中心 軟體國際部 啟



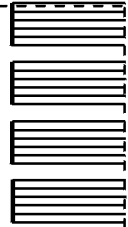
廣 告 回 信
台灣北區郵政管理局 登記
北台字第 00176 號

(免貼郵票)

寄件人 姓名：  
地址：

寄

折疊線





**IBM**