

6.0.1 版



技術產品概觀

6.0.1 版



技術產品概觀

附註

在使用本資訊及它支援的產品之前，務必先閱讀本書最後之「注意事項」中的相關資訊。

目錄

第 1 章 商業整合 1

業務單位間的整合	1
企業間的整合	2
WebSphere Integration Developer	3
標準	3
整合開發人員的角色	4

第 2 章 服務元件架構 5

服務元件	6
服務資料物件	8
服務限定元	9
模組	10
匯入項目及匯出項目	12
服務的匯入及匯出連結類型	12
選取適當的連結	13
服務實作類型	14
Java 物件	14

BPEL 程序	14
狀態機	15
商業規則	16
選取器	17
人工作業	17
介面對映	18
調解流程	18
獨立式參照	20
相關資訊	20

第 3 章 瞭解工具 21

「歡迎使用」視圖概觀	21
「歡迎使用」視圖中的指導教學	21
「歡迎使用」視圖中的範例	22
資訊中心	23

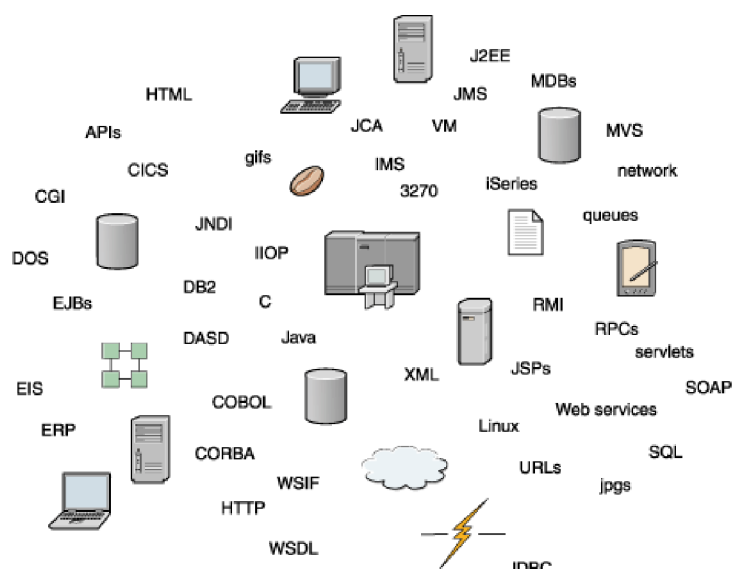
注意事項 25

第 1 章 商業整合

商業整合意指整合企業內部或幾家企業的應用程式、資料和程序。我們將檢視這項作業所面臨的挑戰，以及如何透過 WebSphere Integration Developer 來滿足企業的需求。

您收到指示，必須為客戶建置連接您的商業應用程式的入口網站。這個入口網站必須提供存取散布各業務單位數十種主要商業應用程式及相關資料的途徑。另外，他們還要求您在入口網站中加入您商業夥伴的應用程式。整合也代表開發程序，因為這些組合應用程式有其邏輯順序。您有 20 個業務單位以及 12 個商業夥伴。您的入口網站在 Web 上必須是全天候開放的。公司指派了 6 位開發人員給您，其中還包括您自己在內，而您有 4 個月的時間可以啟動並且運作這個入口網站。

大部分的人在面對這樣的挑戰時，都會先檢視他們過去數十年來所累積的技術，而會看到類似下圖的拼貼圖：



這是個艱難的工作，但卻不是不可能的狀況。除了無條理可循的軟硬體組合外，您面對最困難的問題是您的時間及資源有限。您必須找到功能強大的工具，以迅速集合分散在各相關企業中的應用程式和資料。手動編碼絕對不會是您的選擇。

您是唯一一個面對這種狀況的人嗎？不是。這是一個普遍存在的老問題。根據 2001 年 12 月份的 *CIO Survey* 顯示，應用程式整合始終位居技術優先順位的前三名。根據 *The Business Integrator Journal* 2001 年冬季號的報導，最新調查顯示，目前有三分之二的開發人員使用整合軟體來開發 Web 型解決方案。每位開發人員平均負責開發三種不同的系統。

首先，我們先來討論在商業環境中左右這種問題的兩股力量：企業內部各業務單位間的整合以及企業間的整合。接著，我們將瞭解 WebSphere Integration Developer 如何解決這些問題，尤其是它符合業界標準所彰顯的意義。最後，我們將探討整合專家，這種人員將使用 WebSphere Integration Developer 的工具來解決早先發現的問題。

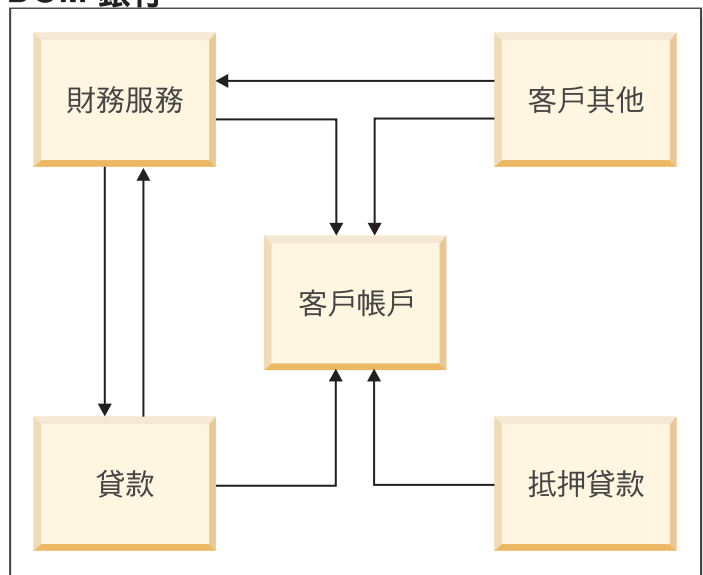
業務單位間的整合

現今業務單位經常合作，因此必須密切整合它們的應用程式。

原本自主發展的業務單位現在都開始整合了，這不只是因為現今的技術能夠將它們連接起來，也因為以效率觀點來看，在運作上必須更進一步合作，才能降低額外負荷並獲得最佳的成果。公司共同目標也是促使這些業務單位攜手合作的原因。市場行銷單位和研發單位都想生產能夠創造利潤的產品。透過整合市場行銷知識與產品開發資訊，成功生產這種產品的機會便可大為提昇。業務單位之間的合作，也讓公司能夠在不同的商業環境中重複使用現有的商業應用程式，而善用許多現有的資源。

業務單位之間的整合比企業之間的整合簡單，因為它的安全風險較低，而且管理各單位間的互動也應該比較容易，因為各業務單位很有可能使用相同的通訊協定、作業系統和電腦語言。換句話說，它是同質性相當高的環境。然而，其關鍵在於使用正確的工具，以快速整合應用程式。在下圖中，DOM 銀行下的幾個業務單位需要彼此分享資訊。幾年前，DOM 銀行透過由一個業務單位列印資訊複本，再徒手交給本身也有系統和應用程式的另一個單位，來進行管理。時至今日，DOM 銀行必須在各業務單位間建立整合性應用程式，才能維持其競爭力。

DOM 銀行



企業間的整合

驅使業務單位間進行應用程式整合的動力，同樣適用於企業之間，因為不論是合作或是接管同樣需要共用的資料和程序。

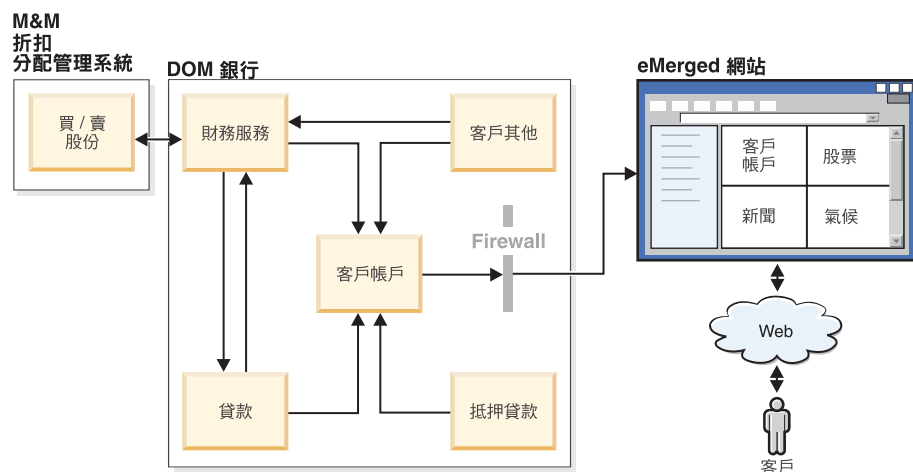
技術的發展使得各企業在互利的基礎下開始結合。例如，汽車製造商可與輪胎供應商合作設定一套整合的程序，使輪胎存貨不足時，可自動通知供應商。促使企業間整合的動機為經濟需求。公司之間如果能夠有較為密切的聯繫，即表示時間延誤的狀況將會減少，而完成工作造成的額外負荷也會降低。這些自動化程序代表處理企業間的交易時間減少，商旅成本和面對面開會的時間可大幅降低。同樣地，管理成本將會降低，而通知、遞送及寄送發票的往返時間也會縮短。

不過不同的企業有不同的發展背景。他們所使用的應用程式是以不同的通訊協定，在不同的平台上以不同的語言編寫。此外，不同的組織在合作時的安全風險也比較高。不論企業之間整合的優勢甚至需求為何，如果沒有使用正確的工具，開發期間的成本都會十分驚人。

WebSphere Integration Developer

WebSphere Integration Developer 可用來解決組織每日所面對的整合挑戰。它被設計成這些建置整合性應用程式的完整整合開發環境。為了簡化及加快整合性應用程式的開發，這個環境提供一個抽象層，而將您所用而以視覺化方式呈現的元件，和基礎的實作隔開。

整合性應用程式並不簡單。它們可呼叫「企業資訊系統 (EIS)」中的應用程式，包含各部門或企業間的商業程序，以及包含以不同語言寫成並在不同作業系統中執行的本端或遠端應用程式。例如，eMerged Corporation 是由 DOM 銀行和 M&M Discount 仲介商合併而成。這項合併代表上述這些 EIS 系統中的應用程式、商業程序和各個舊公司中的應用程式必須在公司間分享，並平順無礙地提供給一組新客戶。不過，eMerged 達成了這項工作，並且如下圖所示，這兩家以往獨立的企業它們的客戶可線上存取其所有的財經資訊。



eMerged 採用 WebSphere Integration Developer 中的工具為自己和客戶建置整合性應用程式。這些工具以元件方式來呈現應用程式（包括存在於遠端 EIS 系統上的應用程式）。元件是透過視覺化編輯器來建立，並組合到其他整合性應用程式中（亦即，從一組元件來建立應用程式）。視覺化編輯器在元件和其實作間呈現一個抽象層。使用工具的開發人員可建立整合性應用程式，而不需對各元件具備精細的基礎實作知識。

工具容許採用由上而下的設計方法來建置整合性應用程式，亦即，其中可有一或多個元件沒有實作，可日後再新增；也可以採用由下而上的方法，亦即，其中的元件已實作，而開發人員可藉由拖放方式在視覺化編輯器中組合元件，然後使用線條來合併元件，以便在其間建立邏輯流程。除錯及測試環境意味著在將應用程式部署到正式作業伺服器前，可進行完整的測試。設定監視點可讓您查看應用程式如何即時使用，以便細部調整讓它擁有最佳效能。

WebSphere Integration Developer 的工具是以服務導向架構為基礎。元件為一些服務，而涉及多個元件的整合性應用程式即為一種服務。所建立的服務符合先進的的業界標準。同樣地，亦成為元件的商業程序是以好用的視覺化工具來建立，並符合「商業程序執行語言 (BPEL)」業界標準。WebSphere Integration Developer 適用於 Windows 和 Linux 平台。

WebSphere Integration Developer 工具的部分優點如下：

- 易學
- 適用於複雜的整合狀況
- 可快速生產符合業界標準的應用程式

標準

WebSphere Integration Developer 所建立的應用程式符合「服務導向架構」的相關業界標準。

沒有人會想建立和專用程式碼相關聯的應用程式，因為再過幾年可能就不支援或者會牽涉到昂貴的授權費。因此，依循標準的整合是 WebSphere Integration Developer 中的一個基礎觀點。在連線功能方面，所用的是「J2EE 連接器架構」標準。在非同步傳訊方面（通常用在需要保證資料傳遞的大型應用程式中），則使用「Java 訊息服務 (JMS)」標準。WebSphere Integration Developer 可輕易整合以「簡式物件存取通訊協定 (SOAP)」為基礎的 Web 服務。在描述服務方面，使用了普獲肯定的「Web 服務說明語言 (WSDL)」標準。在定義商業程序方面，使用了「商業程序執行語言 (BPEL)」標準。

這些依循標準的介面和元件構成了即插的開放式架構。不過，專用元素並未排除；它們可藉由使用標準化介面來存取。這表示建立在 WebSphere Integration Developer 中的應用程式可和 .NET 應用程式等交談。在「架構」一節中，會提供完整「服務元件架構」的鏈結，以廣泛列出眾多的支援標準。

整合開發人員的角色

整合開發人員是 WebSphere Integration Developer 的主要使用者。這類人員可透過視覺化工具，來建立一個複雜的整合性應用程式，而不需具備廣泛的基礎實作知識。

WebSphere Integration Developer 以元件方式提供一些應用程式和商業程序。元件的實作維持隱藏狀態，且元件透過介面交互運作。如此一來，整合開發人員對於元件不需具備廣泛的基礎實作知識，即可建立一個會用到這些元件的整合性應用程式。不過，整合開發人員在整合領域方面可能具備廣泛的技術知識，這是因為他需要對 EIS 系統、商業程序和採用 Java 或其他語言編碼的應用程式具備一些瞭解。例如，建築師大致明瞭系統的運作方式，但卻不清楚各元件細部用途。如同建築師一樣，整合開發人員可能是組織中的成員，專責設計整個應用程式，而組織中另有成員負責編寫特定元件的實作方式。

第 2 章 服務元件架構

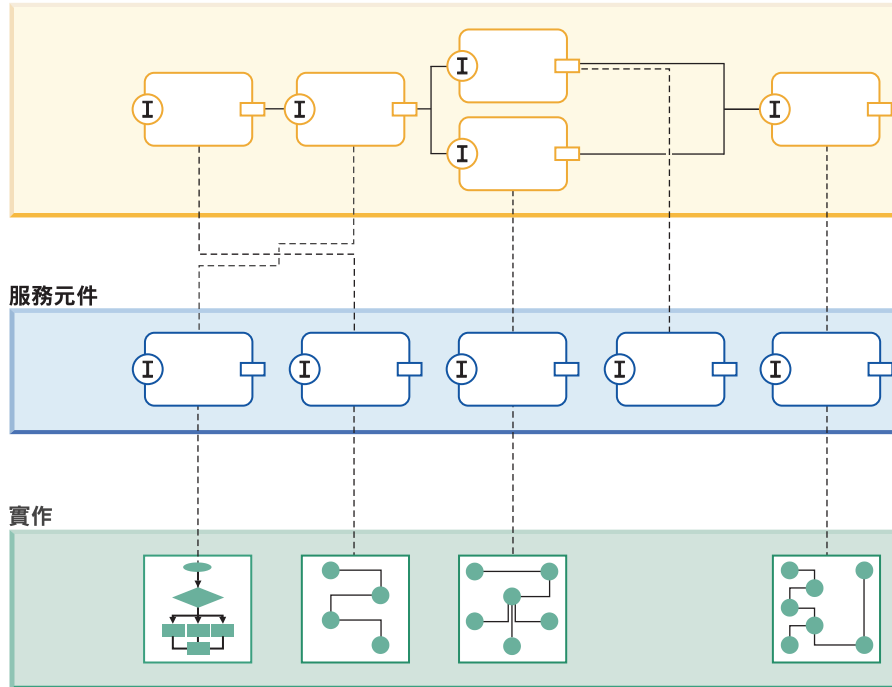
「服務元件架構」是以業界標準的服務導向架構為基礎，採服務導向方式來呈現所有的商業程序，如：Web 服務、「企業資訊系統 (EIS)」服務資產、工作流程、資料庫等。在本節中，我們以較高的層次來檢查這個架構為陳述商業邏輯和參照商業資料而建立的服務和服務資料物件。

「服務元件架構」的目標在於將商業整合邏輯和實作隔開，以便讓整合開發人員可以專注在整合性應用程式的組合上，而不必顧及實作細節。為了達到此目的，而會建立一些內含商業程序所需之個別服務實作的服務元件。如此一來所形成的架構有三層 - 商業整合邏輯、服務元件和實作；請見下圖。



由於服務元件含有實作，整合開發人員可透過圖形方式加以組合，而不必具備低階的實作細節知識。服務元件亦提供選項讓整合開發人員或協助整合開發人員工作者事後再新增實作。您會在本產品中發現，元件是以視覺化方式組合在一起。換句話說，您不需面對元件中的程式碼。在下圖所示的商業邏輯層次中，元件的組合和其實作各自獨立。服務導向的架構可讓您使用和重複使用元件，以專注在商業問題的解決上，而不必分心於您所服務的實作技術上。

商業整合邏輯

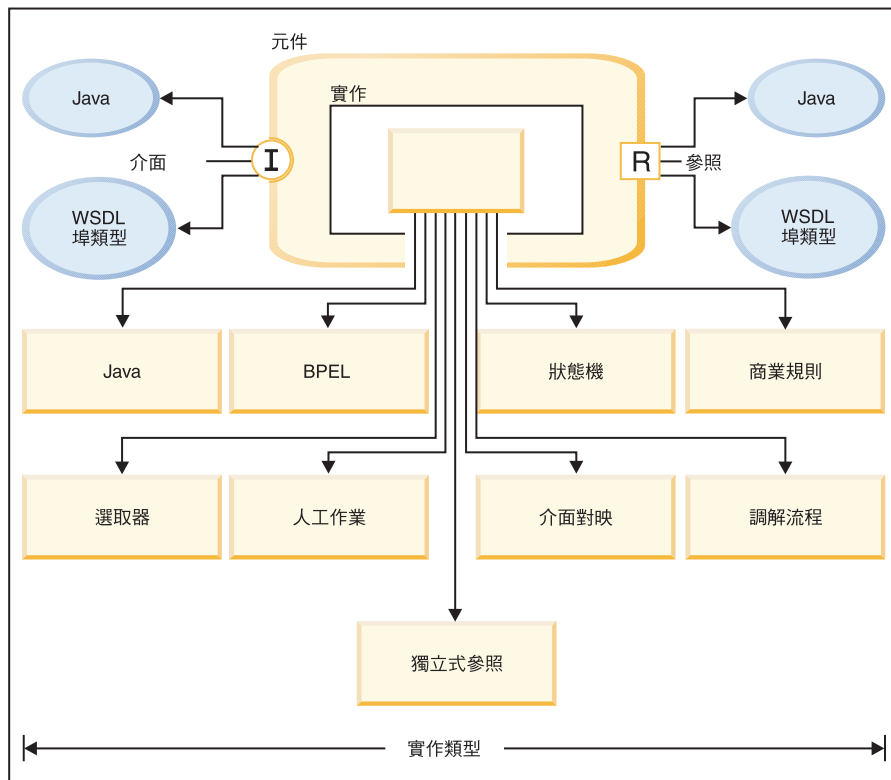


服務元件

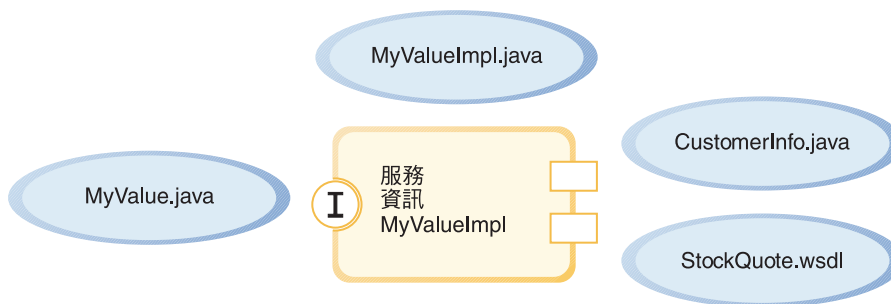
服務元件用以配置服務的實作。服務元件會出現在標準方塊圖中。

元件由一項實作構成，會在使用 WebSphere® Integration Developer 的工具、一或多個介面時隱藏，用以定義其輸入、輸出和錯誤，以及零或多項參照。參照用來識別這個元件所需或所使用之另一個服務或元件的介面。介面可使用下列兩種語言之一來定義：WSDL 埠類型或 Java™。介面可支援同步和非同步交談樣式。元件的實作可採用各種不同語言。

介面類型可以是 WSDL 或 Java，但是如果有多種介面時，WSDL 和 Java 不能混用。

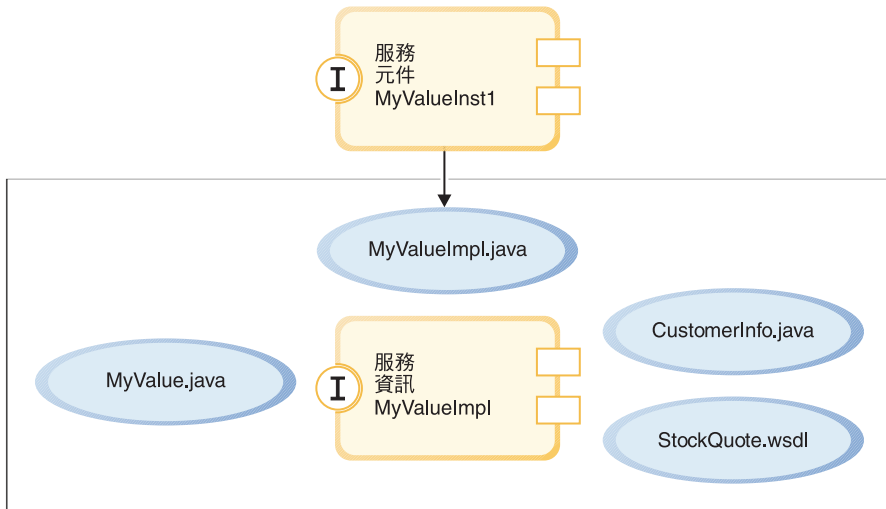


在下圖的中央有一個元件。其實作為 `MyValueImpl`，在 `Java` 中猶如其介面。它有下列兩項參照：另一個 `Java` 介面和 `WSDL` 介面。



如下所示，當搭配這個元件使用時，實際上您只會看到元件本身。另一個元件對這個元件的參照，是以連接其介面的一條線表示。這個元件對其他元件的參照，則是以其參照點到其他元件介面的一條線表示。參照代表這個元件所使用的服務。藉由為參照命名以及只指定其介面，它可讓元件實作作者延緩將該參照連結至實際的服務，而等日後再做。之後，再由整合專家將參照佈線至另一個元件或匯入項目的介面，以完成這項工作。如此將鬆動聯結關係，而延後連結和重複使用實作，這也是為何要使用 `WebSphere Integration Developer` 之「服務元件架構」的重要原因之一。

元件可能也有一些內容和限定元。限定元是在執行時期時有關介面和參照的服務品質 (QoS) 指引。

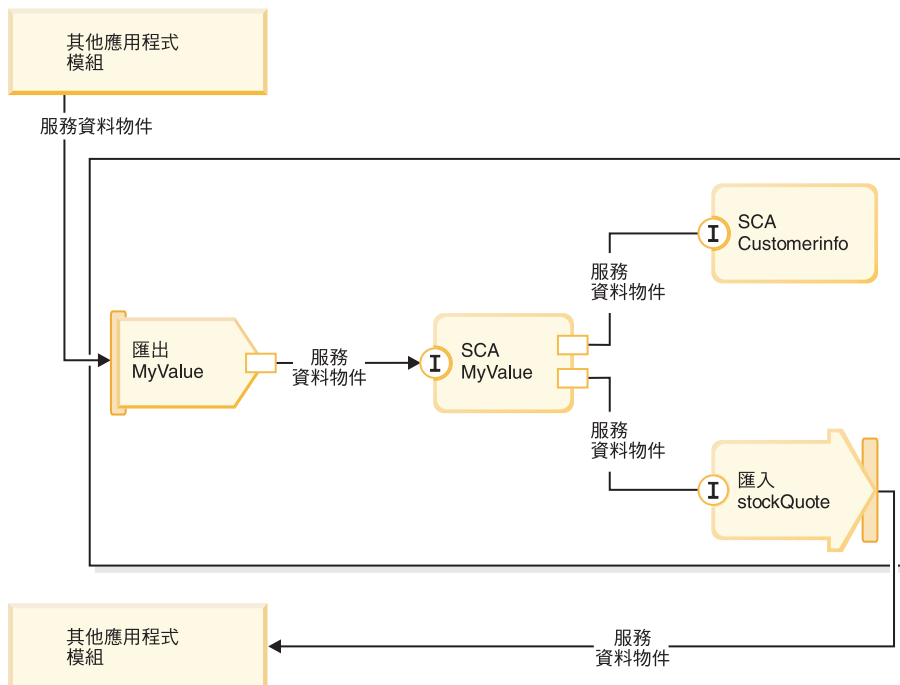


服務資料物件

服務資料物件用來補足「服務元件架構」。「服務元件架構」將服務定義成元件，以及定義其間的連線功能。服務資料物件是定義元件間的資料流程。

每一個元件會將資訊當成輸入及輸出傳遞。當呼叫服務時，如果採用 WSDL 埠類型，則會使用文件文字編碼方式，將資料物件當成 XML 文件傳遞，如果採用 Java 介面，則會當成 Java 物件傳遞。對於「服務元件架構」之服務中的資料和 Meta 資料來說，資料物件是較理想的格式。類似於元件，服務資料物件會將資料物件和其實作隔開。例如，當採購單本身使用 JDBC、EJB 等來執行資料的更新時，元件會和採購單產生互動。服務資料物件可讓整合開發人員專注在商業成品的使用上。事實上，服務資料物件對整合開發人員而言為公開的。它們是由服務資料物件 Java 規格需求 (JSR) 所定義。

在下圖中，服務資料物件從外部服務傳遞匯出項目、從匯出項目傳給元件、從元件傳給元件、從元件傳給匯入項目，以及從匯入項目傳給服務。有關匯入項目及匯出項目的說明，請參閱後續的匯入項目及匯出項目一節。



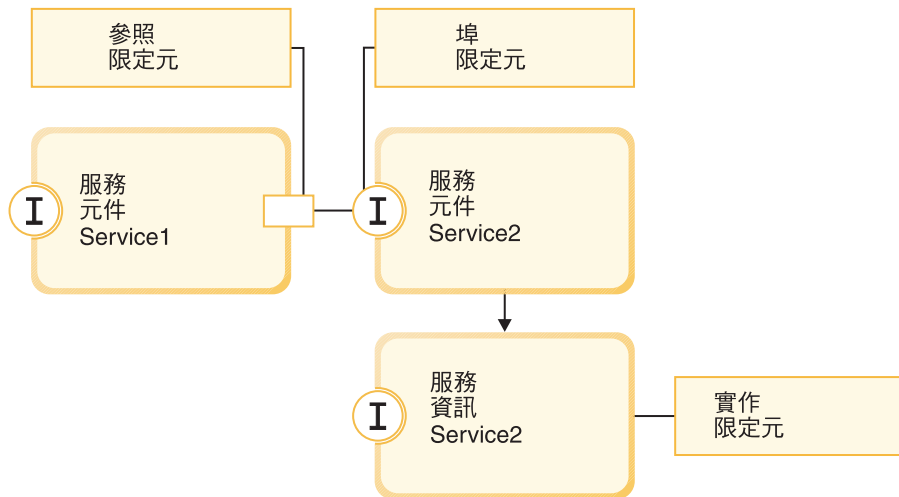
服務限定元

應用程式會指定服務限定元，藉以將其服務品質 (QoS) 傳達給執行時期。它們負責控管服務用戶端和目標服務間的交談。

限定元可指定在服務元件的參照、介面和實作中。由於 QoS 值的宣告是在實作的外部進行，您可以變更這些值而不需變更實作；此外，當同一實作的幾個實例用在不同的環境定義中時，也可以設定不同的值。

限定元的種類如下：

- 交易 - 交易類型的規則
- 活動階段作業 - 合併作用中階段作業時的規則
- 安全 - 許可權的規則
- 非同步可靠性 - 非同步訊息的遞送規則



模組

模組是一種部署單元，用以決定要將那些成品一起套裝在企業保存檔 (EAR) 中。模組元件的收集是爲了效能，且可以依參照來傳遞其資料。模組可視爲一種範圍設定機制；亦即，它會爲成品設定一個組織界限。

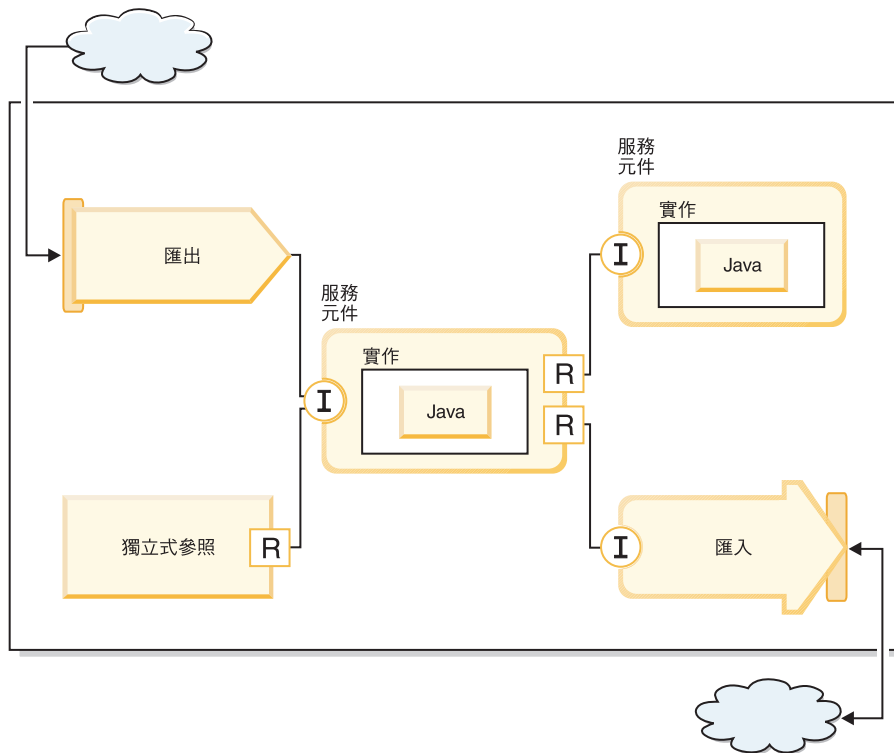
模組是由服務元件、匯入項目及匯出項目構成。服務元件、匯入項目及匯出項目常駐在相同的專案和根資料夾中，此外，其中亦含有佈線，以便鏈結元件和匯入項目及匯出項目所需的連結。模組也可能含有其元件、匯入項目及匯出項目所參照的實作和介面，或者，這些也可位於程式庫專案等其他專案中。

模組類型有兩種。第一種模組稱爲模組 (*module*) (有時亦稱爲商業整合模組)，內有各種元件類型可選，通常用來支援商業程序。第二種模組稱爲調解模組 (*mediation module*)，最多只含有一個元件、一個調解流程元件，外加零或多個用以增強調解流程元件的 Java 元件。

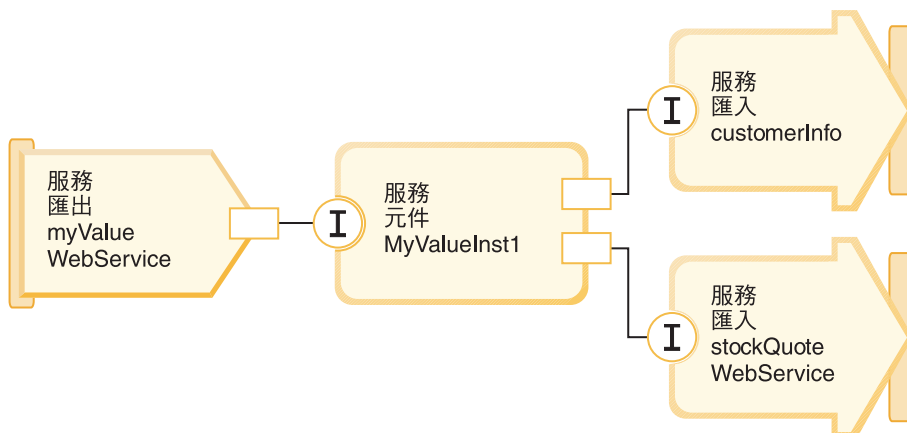
爲何會有兩種模組類型？第一種模組類型旨在供商業程序使用。調解模組就如同現有外部服務的閘道，常見於 Enterprise Service Bus 架構中。這些外部服務或匯出項目在調解模組中是由匯入項目或服務提供者所存取。透過調解流程，崩解用戶端服務要求者和服務提供者間聯結關係，這讓您的應用程式相當具有彈性和恢復力，從而達到服務導向架構的目標。舉例來說，您的調解流程可記載送入的訊息，將訊息傳給在執行時期所決定的特定服務，或轉換資料以便能正確傳給另一個服務。您可以隨時新增和變更這些功能，而不需修改要求者或提供者服務。

模組會讓服務應用程式進行測試並部署到 WebSphere Process Server 中。調解模組會讓服務應用程式進行測試，並部署到 WebSphere Process Server 或 WebSphere Enterprise Service Bus 伺服器中。這兩種模組類型皆支援匯入項目及匯出項目。

在模組之間，精誠需要共用實作、介面、商業物件、商業物件對映、角色、關係和其他成品。程式庫是一種用來儲存這些共用資源的專案。



在下圖中，模組含有一個匯出項目、兩個匯入項目以及一個使用它們的服務元件。所示的佈線鏈結了介面和參照。



模組和調解模組成品包括：

- 模組定義 - 定義模組。
- 服務元件 - 模組中之服務的定義。模組內之服務元件的名稱為唯一的。不過，服務元件可有一個任意的顯示名稱，這通常是一個對使用者有用的名稱。
- 匯入項目 - 匯入項目的定義，用以呼叫這個模組的外部服務。匯入項目含有一些連結；相關說明請參閱匯入項目及匯出項目一節。
- 匯出項目 - 匯出項目的定義，用來顯現這個模組的外部元件給呼叫者。匯出項目含有一些連結；相關說明請參閱匯入項目及匯出項目一節。
- 參照 - 模組中某元件對另一個元件的參照。

- 獨立式參照 - 參照未定義成「服務元件架構」元件的應用程式（如 JavaServer Pages），以便讓這些應用程式能和「服務元件架構」元件互動。每一個模組只能有一個獨立式參照成品。
- 其他成品 - 這些成品包括 WSDL 檔、Java 類別、XSD 檔、BPEL 程序等。

匯入項目及匯出項目

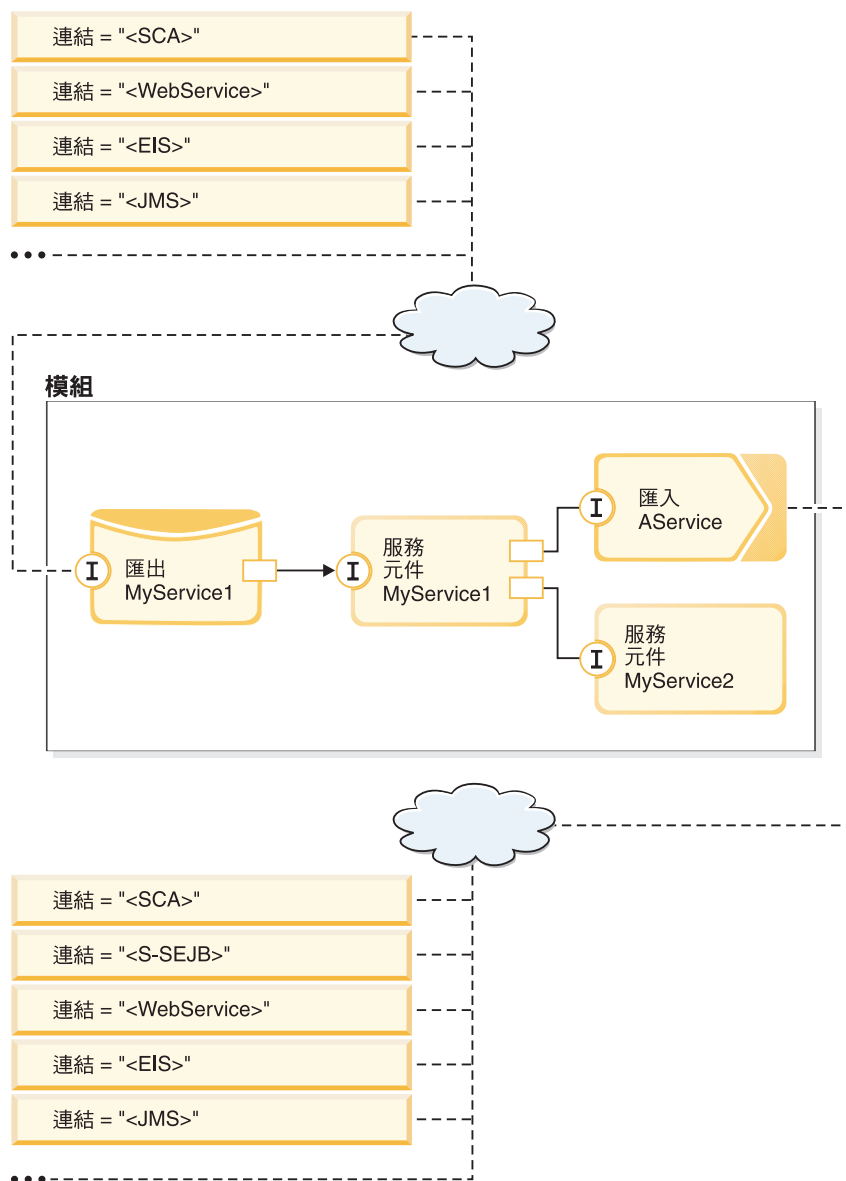
匯入項目及匯出項目是定義模組的外部介面或存取點。匯入項目可識別模組外的服務，以便可從模組中來呼叫它們。匯出項目可讓元件為外部用戶端提供服務。匯入或匯出時需要連結資訊。有數個連結可供使用，而且有一些建議會告訴您哪一種類型可能適合於您的應用程式。

服務的匯入及匯出連結類型

匯入項目及匯出項目需要連結資訊，以指定傳輸模組中的資料時所用的方法。匯入連結說明外部服務連結匯入元件時所用的特定方法。匯出連結說明提供模組的服務給用戶端時所用的特定方法。

SCA 或預設連結可讓您的服務與其他模組中的其他服務通訊。具有 SCA 連結的匯入可讓您存取另一個模組中的服務。具有 SCA 連結的匯出可讓您為其他模組提供服務。Web 服務匯入連結可讓您將外部 Web 服務連結匯入。Web 服務匯出連結可讓您以 Web 服務方式為外部用戶端提供服務。企業服務探索精靈會建立匯入及匯出來代表 EIS 系統上的服務。所建立的連結為 EIS 或「Java 訊息服務」類型。EIS 連結提供與 EIS 系統上之服務進行的同步通訊。一般而言，JMS 連結會用來與需要進行非同步通訊（透過訊息佇列）以求可靠性的大型 EIS 系統交談。匯入（雖非匯出）可能也有一個 Stateless Session EJB 連結。

組合編輯器會列出支援的連結，並可在您想建立匯入或匯出時，簡化連結的建立。組合編輯器中的內容視圖會顯示任何匯入或匯出的連結資訊。



選取適當的連結

本節討論在何種時機下，某種連結可能較適合您應用程式的需求。

WebSphere Integration Developer 中的可用連結可提供一系列的選擇。本清單有助您判別在何種情況下，某種連結類型可能比另一種連結類型更適合您應用程式的需求。

當符合下列因素時，請考量使用 SCA 連結：

- 所有服務皆包含在 WebSphere Integration Developer 模組中；亦即，沒有外部服務
- 效能很重要
- 模組為緊密耦合

當符合下列因素時，請考量使用 Web 服務連結：

- 您需要透過網際網路存取外部服務，或透過網際網路提供服務
- 服務為鬆散耦合

- 您所存取之外部服務的通訊協定或您想提供的服務是 SOAP/HTTP 或 JMS/HTTP

當符合下列因素時，請考量使用 *EIS* 連結：

- 您需要使用資源配接器來存取 *EIS* 系統上的服務
- 效能比可靠性重要；亦即，偏向於捨非同步而採同步資料傳輸

當符合下列因素時，請考量使用 *JMS* 連結：

- 您需要存取傳訊系統
- 服務為鬆散耦合
- 可靠性比效能重要；亦即，偏向於捨同步而採非同步資料傳輸

當符合下列因素時，請考量使用 *Stateless Session EJB* 連結：

- 連結是用在本身是 *EJB* 的所匯入服務上
- 所匯入的服務為鬆散耦合
- *EJB* 的狀態無關緊要

服務實作類型

服務實作類型是指服務元件的實作。

本節將說明服務的標準實作。這些實作將出現在組合編輯器或 *BPEL* 程序中的服務中。

Java 物件

在 *Java* 中，元件的實作稱為 *Java* 物件。

以 *Java* 寫成的元件即是一種常見的實作。有時這種實作的暱稱為「單純傳統的 *Java* 物件」或 *POJO*。一般而言，這種實作採用 *WSDL* 介面類型，不過，這種實作也可以採用 *Java* 介面。如果指定了多個介面，則 *WSDL* 介面不可和 *Java* 介面混用。不過，您可以將使用一組 *WSDL* 介面所建立的應用程式合併至使用 *Java* 介面所建立的應用程式中。「歡迎使用」視圖之範例展示區中的範例即會顯示其做法。

當搭配 *Java* 物件使用時，程式碼在編輯器的環境定義中仍舊隱藏不讓您看到。

Java 物件可以用在調解模組中。它可部署至 *WebSphere Process Server* 或 *WebSphere Enterprise Service Bus* 伺服器中。

BPEL 程序

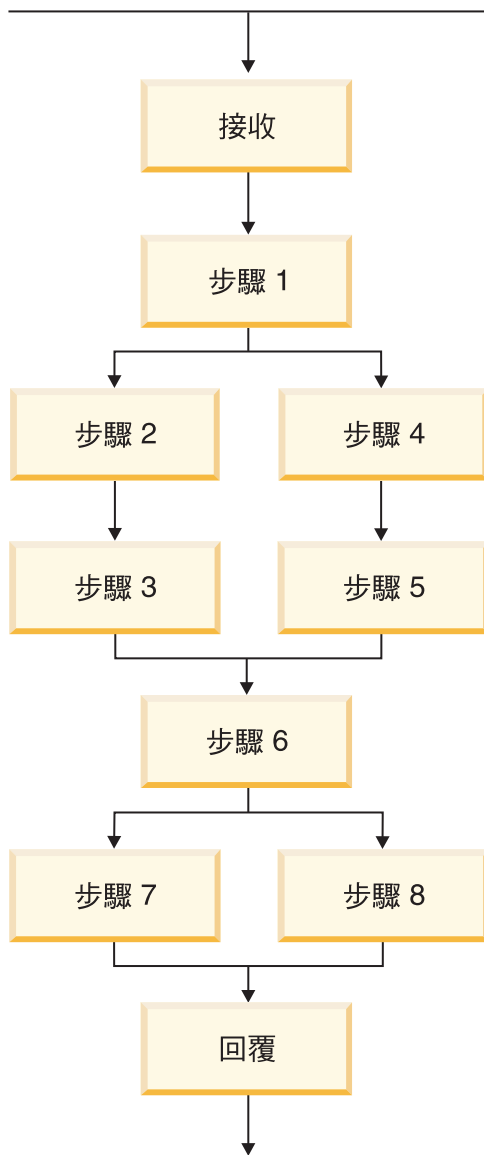
BPEL 程序元件用以實作商業程序。

其實作語言為業界標準 *Web* 服務的商業程序執行語言 (*BPEL4WS*) 以及其 *IBM* 延伸。*BPEL* 程序是藉由使用多項基本服務，來實作可能長時間執行的服務。在程序編輯器中所建立的 *BPEL* 程序可執行下列事項：

- 使用控制流程圖來說明其他服務的編排
- 使用變數來維持程序狀態
- 透過錯誤處理以更準確處理錯誤
- 支援非同步事件
- 使用相互關係集來標示識別實例之要求中的商業資料（如：客戶 ID），以便讓入埠要求和特定程序的正確實例產生關聯
- 透過更準確的補償支援，提供擴充的交易

除了這些標準 BPEL 項目外，透過人工作業支援，WebSphere Integration Developer 還將 BPEL 擴充成可將人員併入程序中。例如，這項擴充可將個人核准貸款的需求條件新增到程序中。

程序編輯器使用視覺化的 BPEL 建構表示法，讓您能快速輕鬆地建置您的商業程序。

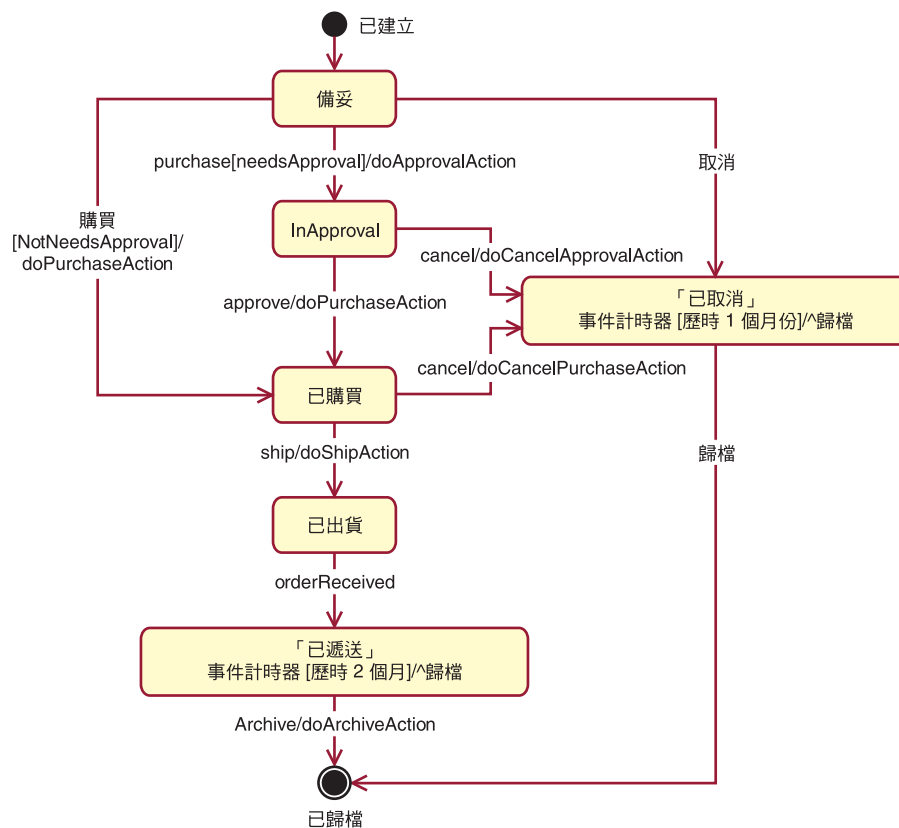


BPEL 程序無法用在調解模組中。它只能部署到 WebSphere Process Server 中。

狀態機

狀態機是另一種用來建立商業程序的方式。狀態機適用於和變更狀態（而非控制流程）有關的程序。狀態是定義該成品在某個時間點所能執行的作業。狀態機即是這組狀態的實作。

狀態機是顯示程序中的一組相關狀態時經常使用的方法。飲料機即為一種常見的狀態機。當您投幣並選取您的飲料時，它會提供您要的飲料並精確找零，這是因為狀態機會機械式地針對您所投入的硬幣退回零錢。下圖是一個由狀態機編輯器所建立的典型狀態機。在狀態機中，客戶購買某項目，並且將該項目出貨給該客戶。



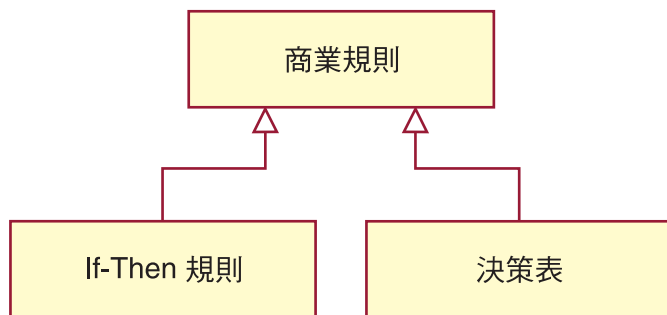
狀態機無法用在調解模組中。它只能部署到 WebSphere Process Server 中。

商業規則

商業規則用以補足商業程序和狀態機。舉例來說，假設某變數有狀況，商業規則即可在執行時期變更該變數中的值。以視覺化程式語言建立的商業規則，是根據環境定義來下決策。決策可以簡單也可以複雜。商業規則為非程序化的，且規則可以變更，而不受任何應用程式影響。

商業規則則是根據環境定義來決定程序的輸出。商業規則用於每日的商業狀況中，以根據一組特定的情況來下決策。此決策可能需要許多規則以涵蓋所有的情況。商業程序中的商業規則可讓應用程式面對變動的商業情況迅速做出回應。以保險公司為例，以下是核准申請者車險的商業規則：如果申請者為男性，年齡超過 25 歲，車種為跑車，只要過去 5 年皆在本公司投保，每月可核准保險申請費用 \$100。

WebSphere Integration Developer 提供許多方法來建立商業規則。您可以建立 if-then 規則或決策表，而這些全都可以塑造您程序的輸出。請注意，這些規則和程序本身各自獨立，亦即，您可以隨時變更規則，而不必重新執行您的程序。例如，您可以根據您的生意所在，訂出下列規則：每逢 12 月 26 日到 1 月 1 日，提供假期後折扣 20%。不過，如果賣況持續低迷，您隨時可將折扣下修到 40%。

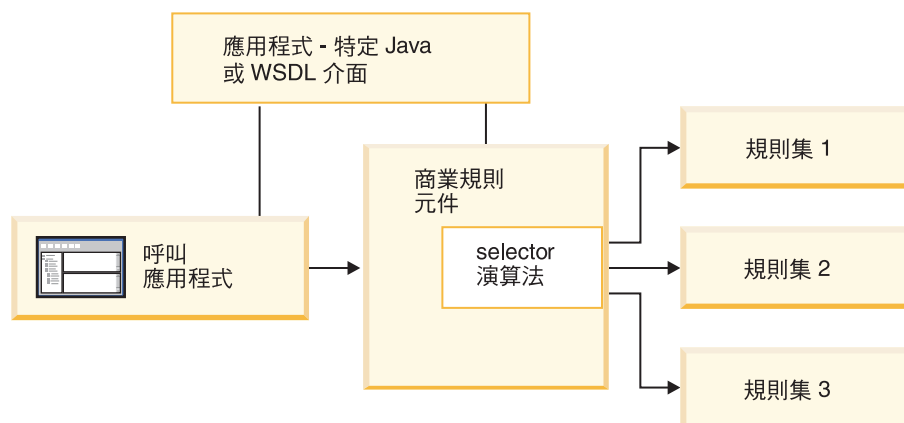


商業規則無法用在調解模組中。它們只能部署到 WebSphere Process Server 中。

選取器

整合性應用程式含有多種互動方式。選取器用以將用戶端應用程式中的作業遞送給其中一個可能的元件，以便實作。

「遞送給元件」這項動作是以日期為依據。以下即為一種以日期為依據的遞送方式：開學前兩週，學用品提供返校特惠價。企業可能有許多這類以日期為依據的遞送方式。選取器會做出決定，以便在以日期為依據的執行時期間，選取一項遞送以取代另一項。比方說，如果時間剛好是在開學前，則呼叫前述的返校報價。如果是在學期結束的季節，則換成學童暑假用品的報價。

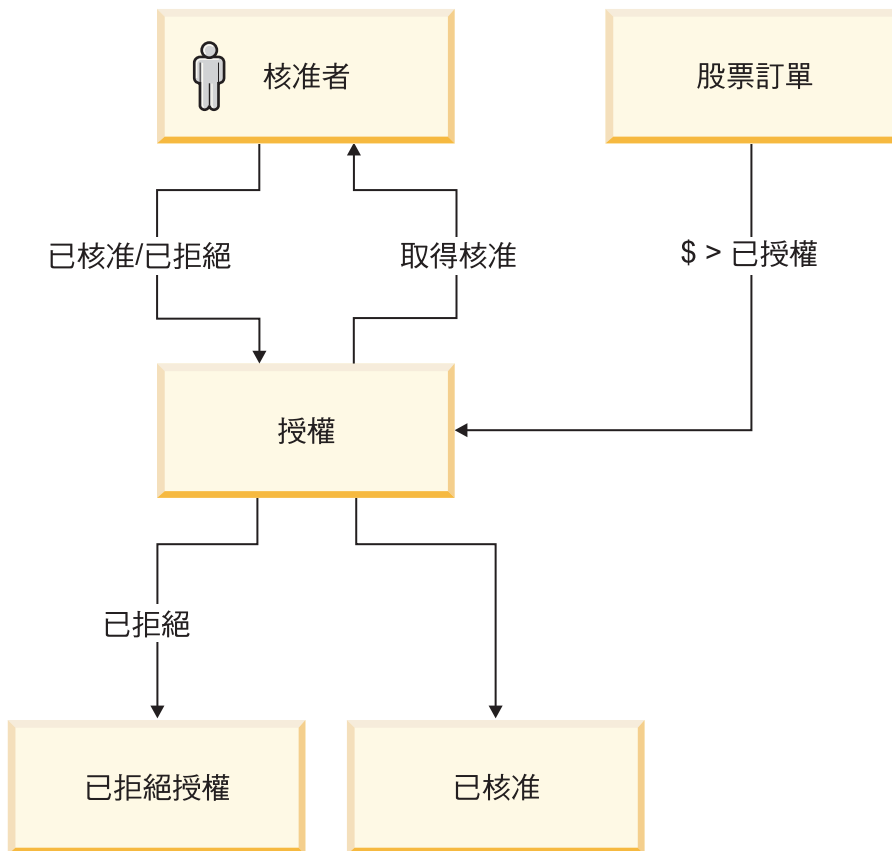


選取器無法用在調解模組中。它只能部署到 WebSphere Process Server 中。

人工作業

人工作業元件是指實作一項由人執行的作業。它代表人員參與商業程序。

在商業程序中有時需要人員介入。例如，客戶想購買一項超出其信用上限的項目。人工作業可讓您介入並置換防止客戶採購的商業規則。人工作業可以有一些屬性，例如，設定作業的擁有者以及提供呈報程序（如果指定的人員不適用的話）。人工作業元件用以辨識一些程序在某些作業（如：檢視、研究和核准）方面是否真的需要人為介入。



人工作業無法用在調解模組中。它只能部署到 WebSphere Process Server 中。

介面對映

介面對映會解析互動元件之介面間的差異。

需要彼此互動之元件的介面間出現差異是很平常的。會出現這些差異的原因在於在 WebSphere Integration Developer 中，您經常要組合原本是為不同應用程式建立的元件。重複使用這些元件以建立新應用程式，是 WebSphere Integration Developer 的強項之一，這是因為您所記錄的反而是類似的元件。不過，一般而言，您仍需進行一些調整。

舉例來說，有兩個元件同樣使用了基本上會執行相同動作但名稱有別（如 `getCredit` 和 `getCreditRating`）的方法。它們也使用了不同的作業名稱，且這些作業的參數類型可能不同。介面對映會對映這些作業和這些方法的參數，以便解析差異，讓這兩個元件可以互動。介面對映就如同這兩個元件之介面間的橋樑，讓它們儘管存在差異也能佈線在一起。

介面對映的存在不受使用它之元件的影響，也就是說，不需變更元件本身。

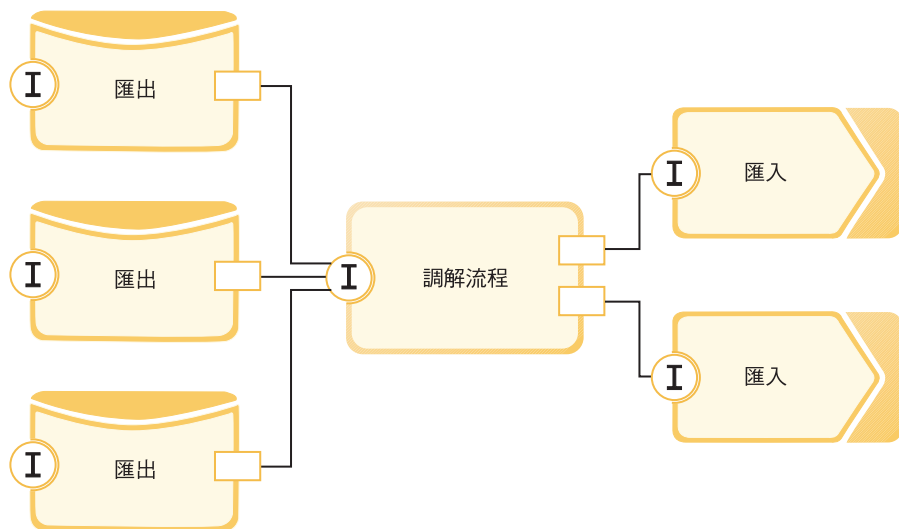
介面對映無法用在調解模組中。它只能部署到 WebSphere Process Server 中。

調解流程

調解是一種動態調解或介入服務之間的方式。調解流程會實作調解。

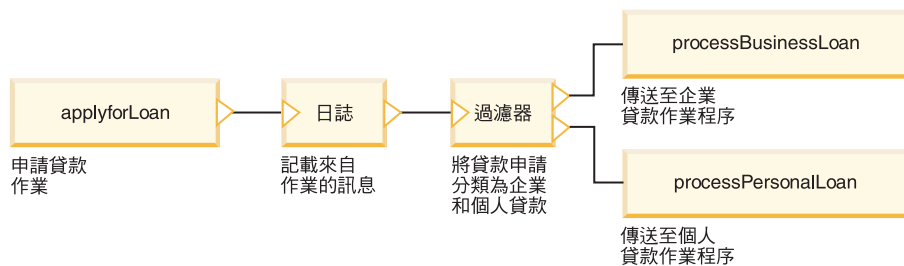
調解具有一些好用的功能。例如，當您需要將某服務中的資料轉換成適合後續服務的格式時，即可使用調解。「日誌記載」可在將服務中的訊息傳給下一個服務前，先記載訊息。「遞送」可讓您將某服務中的資料遞送至調解流程所決定的適當服務。調解的運作和它所連接的服務各自獨立。組合編輯器中的調解會以匯出項目和匯入項目間的調解流程元件出現。

在下圖中，三個服務要求者或匯出項目將其輸出資料傳給調解流程元件的介面。接著，調解流程元件將適當資料遞送給兩個服務提供者或匯入項目。



調解流程是一種使用調解流程編輯器來建立的流程型建構。當在組合編輯器中選取一個調解流程元件時，會啟動調解流程編輯器。在調解流程編輯器中，某服務（服務要求者或匯出項目）中的作業會對映至另一個服務（服務提供者或匯入項目）以及調解流程編輯器所提供的一些功能。這些功能稱為調解基本元素，且會佈線在調解流程中（如下圖所示）。調解基本元素可以是 IBM 提供的，而您也可以建立自己的自訂基本元素。調解基本元素可同時在訊息內容和訊息環境定義中處理，其中，環境定義是指連結特有的資訊，如：SOAP 或 JMS 標頭、使用者定義的內容等。

在下圖中，`applyforLoan` 作業先傳送訊息給一個記載基本元素，以記錄訊息。「日誌」將訊息傳給「過濾器」基本元素，而這個功能會根據訊息，將訊息遞送給 `processBusinessLoan` 作業或 `processPersonalLoan` 作業。



如同模組一節中所述，調解流程元件有一個調解模組。它最多只含有一個調解流程元件，外加零或多個用以增強調解流程元件的 Java 元件。調解模組可部署至 WebSphere Process Server 或 WebSphere Enterprise Service Bus 伺服器中。

獨立式參照

獨立式參照是指未定義成「服務元件架構」元件之應用程式（如 JavaServer Pages 或 Servlet）的參照。獨立式參照可讓這些應用程式和「服務元件架構」元件互動。

獨立式參照沒有介面，也沒有實作（這是因為實作不在模組範圍內）。模組可不含任何獨立式參照，或有一個獨立式參照成品。獨立式參照可有實際值，這可讓您現有的應用程式搭配 WebSphere Integration Developer 中所建立的「服務元件架構」元件一起使用。

獨立式參照可用在調解模組中。它們可部署至 WebSphere Process Server 或 WebSphere Enterprise Service Bus 伺服器中。

相關資訊

本節討論幾個和架構有關的主題。

這些主題提供和產品架構相關聯的其他資訊。

- 『開發要搭配 .NET 服務使用的服務』
- 『雙向支援』

開發要搭配 .NET 服務使用的服務

如果您想開發要搭配 .NET 服務使用的服務，您必須瞭解某些特殊考量。考量的層面有下列兩項：將您在 .NET 開發環境中所開發的 WSDL 檔帶到 WebSphere Integration Developer 開發環境中，以及從 WebSphere Integration Developer 匯出 WSDL 檔，以便搭配 .NET 服務使用。這兩種開發環境的關鍵差異在於 .NET 環境採用行內綱目，而 WebSphere Integration Developer 不是。行內綱目是一種方法，藉以將綱目包含在 WSDL 檔中，而不指定將它匯入成個別的檔案。WebSphere Integration Developer 已提供資訊，協助您如何處理這兩種案例。

「建立介面」一節中的行內綱目會說明如何匯入內含 .NET WSDL 檔將具有之行內綱目的 WSDL 檔。

「組合商業服務」一節中的建立 Proxy 來使用 .NET 服務會說明如何從外部發佈服務，並以此方式使用 .NET 服務。這個主題特別將焦點放在如何建立要搭配 .NET 服務使用的 Proxy 上。

雙向支援

WebSphere Integration Developer 是在多語系環境中運作。這表示它可以顯示和操作以不同語言呈現的資料。這項支援還包括一些採用雙向書寫的語言（如阿拉伯文或希伯來文）在內。這些語言的書寫方向為由右至左，而拉丁文（或斯拉夫文、希臘文等）中的數字和內嵌區段卻採由左至右內嵌在這段文字中。

在 IBM WebSphere Integration Developer 雙向書寫支援概觀中所討論的雙向語言支援包括：所需的配置、使用支援時的一些特定技術點，以及限制事項。

第 3 章 瞭解工具

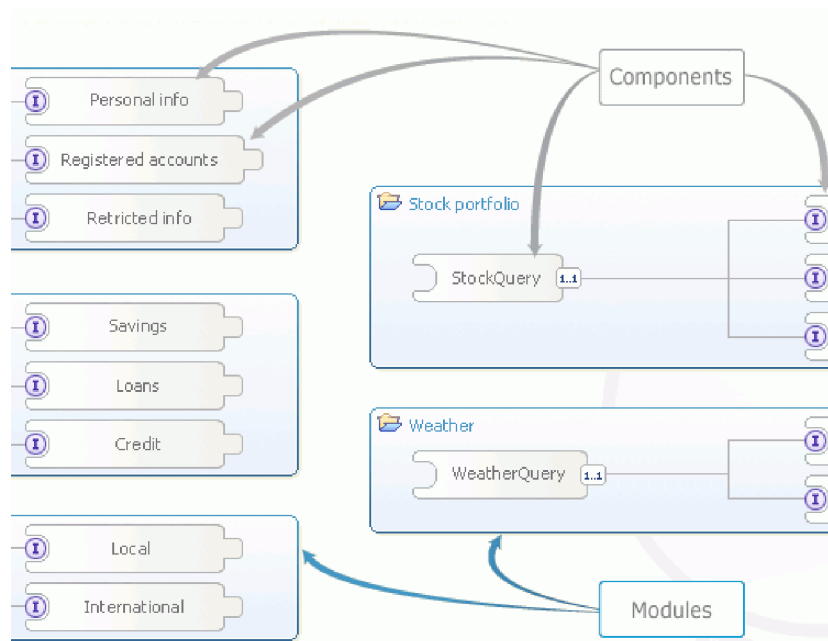
WebSphere Integration Developer 使用視覺化工具來建立以服務導向為架構的整合性應用程式。

「歡迎使用」視圖中的「概觀」一節提供最高階的 WebSphere Integration Developer 工具說明。在此情況下，最高階意指以最抽象的方式來呈現工具。圖形中會說明工具的用途。如果您初次使用本產品，而想迅速瞭解其工具的用途，請參閱「概觀」一節。「歡迎使用」視圖中的「指導教學」章節，可讓您以動態方式來觀看工具。「範例」章節可讓您在失敗安全方式下開始使用工具。最後，當您開始使用 WebSphere Integration Developer 來開發應用程式時，請使用資訊中心來尋找各工具的深度資訊，包括：概念資訊、您可執行的作業，以及各工具的參考資訊等。

「歡迎使用」視圖概觀

「歡迎使用」視圖中的「概觀」一節提供產品中之工具和特性的最高階探討。在「概觀」中，是以圖形來闡述產品中之重要工具的用途。

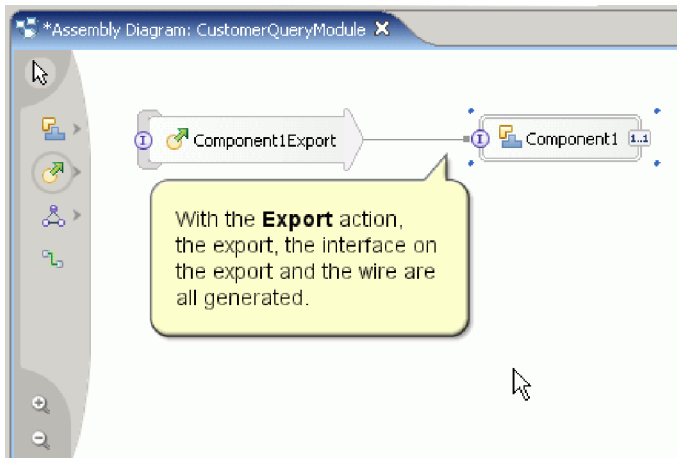
「歡迎使用」視圖的「概觀」一節，可讓您開始使用產品之前，快速瞭解該產品。大部分初次接觸產品的使用者應從這裡著手。當您從「概觀」中選取一個圖示時，會啟動某工具或主題的圖形。例如，以下顯示元件和模組的關係。您也可以展開一段產品的影音之旅。



「歡迎使用」視圖中的指導教學

指導教學可讓您在失敗安全方式下查看動態工具。您可以選取一項指導教學，並觀看它如何像使用者操作般，持續進行一些作業。

一旦您瞭解產品的概念層次，可啓動指導教學以執行工具。每一項指導教學是以電影方式來顯示特定工具。除了觀看指導教學外，氣泡文字會說明您目前所看到的內容。WebSphere Integration Developer 的指導教學可在「歡迎使用」視圖之指導教學展示區的「看中學」一節中找到。



「歡迎使用」視圖中的範例

這些範例可讓您親手操作 WebSphere Integration Developer，以便在失敗安全環境下開發商業服務解決方案。

這些範例可在「歡迎使用」視圖的「範例展示區」中找到。當您選取範例時，會提供您一些選項。您可以遵循步驟指示自行建置，也可以讓它為您建置。

範例因範圍而異。技術範例著重在執行特定作業的特定工具上。應用程式範例則會用到一些工具來完成一個較複雜的結果。實務是一個冗長的範例，將透過許多工具來建置一個大型的應用程式。每一個範例會列出建置所花的時間。

Business state machine (simple)

This sample demonstrates how a business state machine can be used to moderate a sales order transaction. Specifically, the state machine emulates an on-line brokerage that manages the selling of a share.

The following tools are used in this application:

- Business state machine editor
- Human task editor

To import the ready-made sample, click the **Import** link below and click **Finish** in the opened wizard. See **Running instructions** to run the imported code.

or

If you want to build the sample for yourself, click **Step-by-step instructions**.

Ready-made sample

[Import](#) [Running instructions](#)

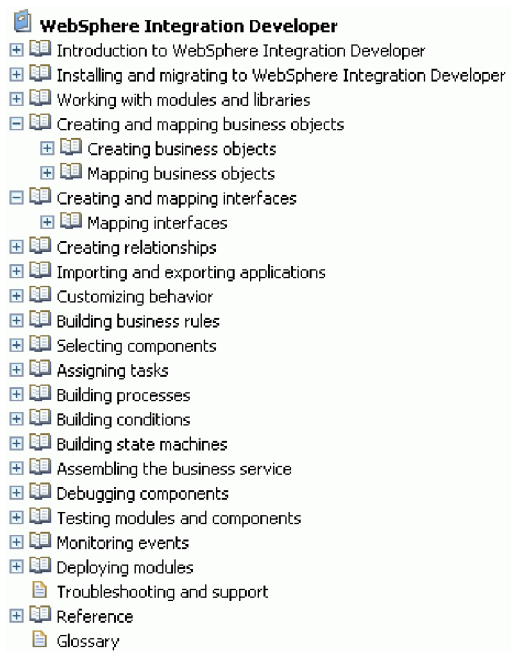
Build it yourself

[Step-by-step instructions](#)

資訊中心

資訊中心提供產品的完整資訊。除了提供指導教學和範例外，也可找到 WebSphere Integration Developer 中每一項工具的其他概念、作業和參考資訊。

資訊中心的導覽可讓您快速存取您要的資訊。您也可以搜尋整個資訊中心。請從搜尋結果中選取，以查看各種內含深度主旨資訊的主題，以及相關資訊的鏈結。



注意事項

本 IBM® 產品所檢附的 XDoclet 文件，必須具備許可權才能使用，而且必須遵守下列著作權聲明：Copyright (c) 2000-2004, XDoclet Team. All rights reserved.

本軟體之若干部分係以 Addison-Wesley Publishing Company, Inc. 出版之 *Design Patterns: Elements of Reusable Object-Oriented Software* 為基礎，其作者為 Erich Gamma、Richard Helm、Ralph Johnson 及 John Vlissides，其著作權標示如下：Copyright (c) 1995 by Addison-Wesley Publishing Company, Inc. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

本資訊係針對 IBM 在美國所提供之產品與服務所開發；而在其他國家中，IBM 不見得有提供本件中所提及的各項產品、服務或功能。請洽詢當地的 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。本文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 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，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

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

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

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

下列段落若與該國之法律條抵觸時，即視為不適用：International Business Machines Corporation 只依「現況」提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不違反規定、可商用性或特定目的之適用性的隱含保證。有些國家在某些交易上並不接受明示或默示保證的排除，因此，這項聲明對 貴客戶不見得適用。

本資訊可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/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。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式之獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

Intellectual Property Dept. for Rational Software
IBM Corporation
20 Maguire Road
Lexington, Massachusetts 02421-3112
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 IBM 客戶合約、IBM 國際程式授權合約或雙方之任何同等合約的條款，提供本文件所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料是在控制環境中得出。因此，在其他作業環境中獲得的結果可能有明顯的差異。部分測量可能是在開發階段的系統上測定，無法保證這些測量在一般可用的系統上維持不變。再者，有些測定可能是透過推測方式來評估。實際結果可能不同。本文件的使用者應驗證適用於其特定環境的資料。

本書所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 未測試過這些產品，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。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

所有關於 IBM 未來方針或目的之聲明，隨時可能更改或撤銷，不必另行通知，且僅代表目標與主旨。

本資訊含有日常商業運作所用之資料和報告範例。為了盡可能地加以完整說明，範例中含有個人、公司、品牌及產品的名稱。所有這些名稱全為虛構，任何與實際商場企業使用的名稱及地址類似之處，純屬巧合。

著作權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。貴客戶可以為了研發、使用、銷售或散佈符合範例應用式所適用的作業平台之應用程式介面的應用程式，以任何形式複製、修改及散佈這些範例程式，不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。您可以免費為了開發、使用、銷售或散布應用程式，利用任何形式來複製、修改和散布這些程式範例，不過，這些應用程式必須符合 IBM 的應用程式設計介面。

這些範例程式或任何衍生成果的每份複本或任何部分，都必須依照下列方式併入著作權聲明：

(C) (您的公司名稱) (年份)。本程式的若干部分係衍生自 IBM 公司的範例程式。(C) Copyright IBM Corp. 2000, 2005. All rights reserved.

若 貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

程式設計介面資訊

程式設計介面資訊之目的，是要協助您利用這個程式來建立應用軟體。

通用程式設計介面可讓您撰寫應用軟體，以取得本程式工具的服務。

不過，這份資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

警告：請勿將這份診斷、修正和調整資訊當作程式設計介面使用，因為它隨時會改變。

商標和服務標記

請參閱 <http://www.ibm.com/legal/copytrade.shtml>。

讀者意見表

爲使本書盡善盡美，本公司極需您寶貴意見；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	
	※評估意見爲"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 tscadmin@tw.ibm.com，謝謝。

WebSphere® Integration Developer

SC40-2056-01

技術產品概觀

6.0.1 版

折疊線

台北市 110 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	信 號
臺灣北區郵政管理局 登記	
北台字第	

(免貼郵票)

寄件人 姓名：
地址：

寄

折疊線

讀者意見表



SC40-2056-01

