

IBM® WebSphere® Commerce



Catalog Manager 使用手冊

第 5.4 版

IBM® WebSphere® Commerce



Catalog Manager 使用手冊

第 5.4 版

注意事項

在使用此資訊和其支援的產品之前，請閱讀第121頁的『注意事項』中的資訊。

第一版，第一次修訂（2002 年 4 月）

本版適用於下列產品：

- IBM WebSphere Commerce 5.4 版（程式 5724-A18）

及其所有後續的版次和修正，除非新版另有提及。請確定您使用的是產品層次的正確版本。

請向 IBM 業務代表或向當地的 IBM 分公司訂購出版品。下列地點恕不供應。

IBM 歡迎您提供意見。請以下列任何一種方法傳送您的批評與建議：

1. 以電子方式送到下列的電子郵件地址。

torrcf@ca.ibm.com

如果您希望獲得答覆，請務必註明您的完整網路位址。

2. 郵件請寄到下列地址：

IBM Canada Ltd. Laboratory B3/KB7/8200/MKM
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7

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

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

目錄

| | |
|--------------------|----|
| 開始之前. | v |
| 本書中使用的慣例. | v |
| 本書的適用對象. | vi |
| 其它相關資訊的位置. | vi |

第1篇 Catalog Manager 概觀 1

第2篇 轉換、載入和擷取資料 3

第1章 簡介. 5

| | |
|---------------|---|
| 公用程式. | 5 |
| 管理工具. | 8 |

第2章 轉換文字. 9

| | |
|---|----|
| 啓動文字轉換工具. | 10 |
| 使用文字綱目編輯檢視畫面 | 11 |
| 建立綱目檔. | 11 |
| 開啓綱目檔. | 11 |
| 儲存綱目檔. | 11 |
| 編輯綱目檔. | 11 |
| 新增元素. | 11 |
| 移除元素. | 12 |
| 取代元素. | 12 |
| 將元素移動至上一列. | 12 |
| 將元素移動至下一列. | 12 |
| 新增屬性. | 12 |
| 移除屬性. | 12 |
| 取代屬性. | 12 |
| 將屬性移動至上一列. | 12 |
| 將屬性移動至下一列. | 13 |
| 變更綱目檔結構. | 14 |
| 準備一個綱目檔，以便將資料從 XML 轉換成以字元界定的變數格式. | 14 |
| 使用「轉換指令編輯檢視畫面」. | 15 |
| 建立指令. | 15 |
| 移除指令. | 15 |
| 編輯或取代指令. | 15 |
| 將指令移動至上一列. | 15 |
| 將指令移動至下一列. | 16 |
| 清除指令. | 16 |
| 使用「轉換程序檢視畫面」. | 16 |

第3章 轉換 XML 資料. 17

| | |
|----------------------|----|
| 啓動 XSL 編輯器. | 18 |
| 使用對應規則建置區. | 19 |
| 建立對映規則建置區. | 19 |
| 修改對映規則建置區. | 19 |
| 刪除對映規則建置區. | 19 |
| 使用「XSL 編輯器」. | 20 |
| 建立對映規則. | 20 |

| | |
|----------------------------|----|
| 元素間的對映. | 20 |
| 屬性間的對映. | 20 |
| 建立自訂對映表示式. | 20 |
| 刪除對映規則. | 21 |
| 處理 XML 轉換. | 21 |
| 自訂對映規則表格. | 21 |
| 顯示完整的 XSL 規則/值表示式. | 21 |

第4章 產生 DTD 和綱目. 23

| | |
|--------------------------|----|
| 設定 DTD 產生器. | 23 |
| 產生 DTD. | 24 |
| 產生綱目和詳細的 XML 檔案. | 27 |

第5章 解析識別碼. 29

| | |
|----------------------------------|----|
| 設定「ID 解析程式」. | 29 |
| 設定「ID 解析程式」處理時間戳記的方式. | 30 |
| 設定「ID 解析程式」處理儲存體的方式. | 30 |
| 設定「ID 解析程式」處理資料庫驅動程式的方式. | 31 |
| 決定資料的處理方式. | 31 |
| 選擇載入方法. | 31 |
| 選擇更新方法. | 32 |
| 選擇混合方法. | 32 |
| 使用 ID 解析技術. | 32 |
| 使用「ID 解析程式」來指定內容檔. | 33 |
| 使用內容檔來產生識別碼. | 34 |
| 透過複合鍵來使用內容檔. | 35 |
| 透過連鎖反應主要鍵來使用內容檔. | 36 |
| 使用內部別名解析. | 37 |
| 使用內部別名 ID 解析的局部範例. | 38 |
| 使用單一索引解析. | 38 |
| 唯一索引解析的局部範例. | 39 |
| 將資料載入至 MEMBER 表格. | 40 |
| 使用 REFKEYS 表格來建立外部關係. | 41 |
| 錯誤疑難排解. | 42 |

第6章 載入資料. 43

| | |
|-----------------------------|----|
| 設定「載入器」. | 43 |
| 忽略輸入檔中的元素. | 44 |
| 在直欄中插入空值 (NULL). | 44 |
| 載入時間戳記和日期資料. | 44 |
| 載入現行時間戳記. | 45 |
| 載入現行時間戳記的範例. | 46 |
| 在現行時間戳記中加入期限的範例. | 47 |
| 管理事件佇列. | 48 |
| 與不同的資料庫軟體和作業系統一起執行. | 48 |
| 替換元件. | 51 |
| 使用「產品顧問」搜尋空間同步化. | 52 |
| 自訂「產品顧問」搜尋空間同步化. | 55 |
| 決定在使用「載入器」時的資料處理方式. | 56 |
| 選擇載入方法. | 56 |
| 選擇匯入方法. | 56 |

| | |
|-----------------------|----|
| 選擇 SQL 匯入方法 | 57 |
| 其他注意事項 | 57 |
| 載入大型文件 | 58 |
| 疑難排解要訣 | 59 |

第7章 擷取資料 61

| | |
|-------------------|----|
| 建立擷取過濾器 | 61 |
| 設定擷取程式 | 65 |

第8章 使用「載入器」套件日誌程式 . . . 67

| | |
|---|----|
| 架構您的環境在 Windows NT、Windows 2000、AIX、Linux 和 Solaris 系統中的日誌記載 . . . 67 | |
| 設定類別路徑變數的範例說明 | 67 |
| 指定 com.ibm.wca.logging.configFile 系統內容的範例說明 | 67 |
| 自訂「載入器」套件的日誌記載 | 68 |
| 處理程式 | 68 |
| 過濾器 | 70 |
| 格式 | 70 |
| 範例：WCALoggerConfig.xml 和 WCALogger.dtd | 71 |
| WCALoggerConfig.xml | 71 |
| WCALogger.dtd | 72 |

第9章 使用「載入器」套件錯誤報告程式 73

第10章 架構載入器套件指令和 Script . . 75

第3篇 使用 Web 編輯器 79

第11章 設置 Web 編輯器 81

| | |
|--------------------------------------|----|
| 架構 Web 編輯器 | 82 |
| 編輯 webeditor.properties 檔案 | 82 |
| 變更暫存檔的位置 | 83 |
| 使用「DTD 產生器」來建立 XML 套表說明檔 | 84 |
| 自訂 XML 套表說明 | 86 |
| 編輯套表名稱 | 88 |
| 變更欄位說明 | 89 |
| 新增下拉式功能表 | 89 |
| 新增欄位說明 | 90 |
| 自訂搜尋結果和工作階段作業清單 | 90 |

| | |
|-------------------------------|----|
| 編輯 weProcessList 檔案 | 91 |
| 編輯 webeditor.xml 檔案 | 95 |

第12章 使用型錄 97

| | |
|-------------------------------|----|
| 使用 Web 編輯器在表格中新增記錄 | 97 |
| 使用 Web 編輯器來修改表格中的記錄 | 98 |
| 使用 Web 編輯器來刪除表格中的記錄 | 99 |

第4篇 指令說明 101

第13章 DTD Generate 指令 103

| | |
|---|-----|
| Windows、AIX、Linux 和 Solaris 系統的 DTD Generate 指令 | 103 |
| 適用於 iSeries 系統的 DTD Generate 指令 | 104 |

第14章 Extract 指令 107

| | |
|--|-----|
| Windows、AIX、Linux 和 Solaris 系統的 Extract 指令 | 107 |
| 適用於 iSeries 系統的 Extract 指令 | 108 |

第15章 ID Resolve 指令 109

| | |
|---|-----|
| Windows、AIX、Linux 和 Solaris 系統的 ID Resolve 指令 | 109 |
| 適用於 iSeries 系統的 ID Resolve 指令 | 110 |

第16章 Load 指令 113

| | |
|---|-----|
| Windows、AIX、Linux 和 Solaris 系統的 Load 指令 | 113 |
| 適用於 iSeries 系統的 Load 指令 | 115 |

第17章 Text Transform 指令 117

| | |
|---|-----|
| Windows、AIX、Linux 和 Solaris 系統的 Text Transform 指令 | 117 |
| 適用於 iSeries 系統的 Text Transform 指令 | 117 |

第18章 XML Transform 指令 119

| | |
|--|-----|
| Windows、AIX、Linux 和 Solaris 系統的 XML Transform 指令 | 119 |
| 適用於 iSeries 系統的 XML Transform 指令 | 119 |

注意事項 121

| | |
|-------------------|-----|
| 商標及服務標示 | 122 |
|-------------------|-----|

開始之前

IBM WebSphere Commerce Catalog Manager 使用手冊 提供了關於 WebSphere Commerce Catalog Manager 的資訊。其中特別提供了以下主題的詳細內容：

- 使用 Catalog Manager 工具和公用程式來轉換、載入和擷取資料
- 使用「Catalog Manager Web 編輯器」來處理型錄資料
- Catalog Manager 指令

本書中使用的慣例

本書使用下列慣例：

| | |
|---|--|
| 粗體字型 | 表示欄位名稱、按鈕或功能表選項等指令或圖形式使用者介面 (GUI) 控制項。 |
| 等框字型 | 表示您所輸入的文字範例、檔案名稱和目錄路徑必須完全相同。 |
| 斜體字型 | 用來表示強調和用輸入值來取代的變數。 |
|  | 表示為 WebSphere Commerce for Windows® 的專屬資訊。 |
|  | 表示為 WebSphere Commerce for Windows NT® 的專屬資訊。 |
|  | 表示為 WebSphere Commerce for Windows 2000® 的專屬資訊。 |
|  | 表示為 WebSphere Commerce for AIX 2000® 的專屬資訊。 |
|  | 表示為 WebSphere Commerce for Solaris™ Operating Environment 軟體的專屬資訊。 |
|  | 表示 WebSphere Commerce for Linux 專屬的資訊。 |
|  | 表示為 WebSphere Commerce for IBM @server™ iSeries™ (先前的名稱為 AS/400®) 的專屬資訊。 |
|  | 表示為 DB2® Universal Database 的專屬資訊。 |
|  | 表示為 Oracle® 資料庫的專屬資訊。 |
|  | 表示 WebSphere Commerce Business Edition 專屬的資訊。 |
|  | 表示 WebSphere Commerce Professional Edition 專屬的資訊。 |

本書的適用對象

需要了解如何使用 WebSphere Commerce Catalog Manager 的網站程式開發人員、管理者和內容提供者，以及其他需要了解其功能的企業使用者，都應該閱讀本書。

需要了解如何使用多種 WebSphere Commerce Catalog Manager 功能中 WebSphere Commerce 的「商店程式開發人員」和「網站管理者」特別應閱讀本書。

要執行程式化延伸功能的「WebSphere Commerce 商店程式開發人員」和「網站管理者」也應了解以下領域的知識：

- 資料庫技術
- Enterprise JavaBeans™ 元件架構
- 超文字標記語言 (Hypertext Markup Language, HTML)
- Java™
- JavaServer Pages™ 技術
- VisualAge® for Java™ Enterprise Edition 3.5 版 (含) 以上
- 可延伸標記語言 (XML)

「型錄管理者」和「產品經理」應閱讀本手冊，以便了解如何與「商店程式開發人員」溝通型錄、產品和項目的基本需求。「型錄管理者」和「產品經理」也應了解如何使用 Web 編輯器來更新商店的型錄。

其它相關資訊的位置

WebSphere Commerce 網站提供了本使用手冊的 Adobe 可攜式文件格式 (PDF) 版本。請查看 WebSphere Commerce Technical Library 網站以便取得本文件以及其他 WebSphere Commerce 相關資訊的最新版本：

-  Business

http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html

-  PRO

http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html

第1篇 Catalog Manager 概觀

WebSphere Commerce Catalog Manager 提供的通用工具箱，而其提供的功能可按照必要的順序合併使用，來解決特定的型錄管理問題。其彈性足以處理針對 WebSphere Commerce 綱目所做的自行設定。

Catalog Manager 可將多個來源的資訊集合至集中化的 WebSphere Commerce 系統，並且使用 XML 作為管理資訊的標準方法，將分散的所有資料重新對映為標準的型錄和產品定義格式。

Catalog Manager 為您提供了執行以下作業的方法：

- 以 ASCII 和 XML 檔案形式將多個輸入來源的資料匯入至 WebSphere Commerce
- 在 ASCII 和 XML 格式之間，進行資料轉換
- 在各種 XML 格式之間重新對映資料
- 將多個輸入串流的資料集合至集中化資料庫
- 透過 Web 瀏覽器建立/編輯/刪除資料

Catalog Manager 包括以下項目：

- 「Catalog Manager 載入器」套件

此套件中包含的主要指令公用程式，可用來準備和將資料載入 WebSphere Commerce 資料庫。您可以利用載入器套件來載入大量資料，並在 WebSphere Commerce 資料庫中更新資料。

「載入器」套件可讓您執行以下項目：

- 從資料庫中擷取資料作為 XML 文件
- 將 XML 資料轉換為其他 XML 格式
- 在以字元界定的變數與 XML 資料格式間轉換資料

- Catalog Manager 管理工具

Catalog Manager 也包含了以下兩種具有使用者介面的工具，來協助您管理其功能：

- 文字轉換工具
- XSL 編輯器

- Catalog Manager Web 編輯器

Web 編輯器可讓您透過 Web 瀏覽器建立、刪除和變更型錄資料。

第2篇 轉換、載入和擷取資料

第1章 簡介

「Catalog Manager 載入器」套件中所包含的公用程式，主要是用來準備並將資料載入至 WebSphere Commerce 資料庫中。您可以利用載入器套件來載入大量資料，並在 WebSphere Commerce 資料庫中更新資料。「載入器」套件可讓您執行以下項目：

- 從資料庫中擷取資料作為 XML 文件
- 將 XML 資料轉換為其他 XML 格式
- 在以字元界定的變數與 XML 資料格式間轉換資料

Catalog Manager 也包含了以下兩種具有使用者界面的工具，來協助您管理其功能：

- 文字轉換工具
- XSL 編輯器

公用程式

「WebSphere Commerce 載入器」套件中包含的公用程式，是用來準備並將資料載入至 WebSphere Commerce 資料庫。您可以使用這些公用程式來載入任何數量的資料，以及更新 WebSphere Commerce 資料庫中的資料。

載入程序包含將資料移入 WebSphere Commerce 資料庫所需的步驟：

1. 使用「DTD 產生器」來產生 DTD
2. 利用「ID 解析程式」解析輸入檔中的識別碼
3. 利用「載入器」載入資料

載入器套件中也包括一些公用程式，可用來從資料庫中擷取資料成為 XML 文件，再將 XML 資料轉換為替代的 XML 格式。

「載入器」套件由下列公用程式組成：

- **文字轉換程式**

文字轉換程式會在以字元界定的變數與 XML 資料格式間轉換資料。

請參閱第9頁的『第2章 轉換文字』以取得其他資訊。

- **XML 轉換程式**

XML 轉換程式會變更、聚集以及重新對映 XML 文件中的資料，再視需要轉換為其他使用者或系統所用的替代 XML 格式。

請參閱第17頁的『第3章 轉換 XML 資料』以取得其他資訊。

- **DTD 產生器**

DTD 用以指定能在 XML 資料文件中使用的結構元素和標示定義。比方說，DTD 中可列出要在文件中使用的元素，並指定每個元素皆可採用的屬性。

「DTD 產生器」是一種公用程式，它可以根據資料庫綱目來建立供載入器使用的 DTD。此 DTD 會說明可供載入器匯入資料的表格與直欄。「DTD 產生器」也可建立用於「Catalog Manager Web 編輯器」的綱目和 XML 詳細文件。

「DTD 產生器」會根據目標資料庫上，資料必須依循的格式來產生 DTD。整個載入程序都會使用這個 DTD。您只需要執行一次「DTD 產生器」。

請參閱第23頁的『第4章 產生 DTD 和綱目』以取得其他資訊。

- **ID 解析程式**

「ID 解析程式」為一種公用程式；它可以為需要識別碼的 XML 元素產生識別碼。如果您的 XML 內容已提供識別碼，即不用執行「ID 解析程式」。

「ID 解析程式」會使用一組 XML 元素的相關識別碼，來更新這組 XML 元素。這個步驟非常重要，因為載入器套件 XML 檔案會直接對映至目標資料庫綱目。因此，它們必須有 ID。

「ID 解析程式」包含可在發生錯誤時產生異常文件的錯誤報告程式。

請參閱第29頁的『第5章 解析識別碼』以取得其他資訊。

- **載入器**

「載入器」會以有效且格式完整的 XML 為輸入，將資料載入資料庫中。XML 文件中的元素會對映至資料庫中的表格名稱，而元素屬性會對映至直欄。載入器是將資料載入系統時最常用的方法。

載入器可更新表格中的直欄層次。它也可以刪除資料庫中的資料。

下列範例是摘錄輸入到載入器中的一段形式完整而有效的 XML：

```
<ADDRBOOK
  ADDRBOOK_ID="11801"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
```

在上例中，ADDRBOOK 是表格名稱，而要更新的直欄是以 ADDRBOOK 元素的屬性指出。

「載入器」公用程式包含以下功能：

- **錯誤報告程式**

「載入器」包含可在發生錯誤時產生異常文件的錯誤報告程式。

- **「產品顧問」搜尋空間同步化**

「產品顧問」搜尋空間的建立方式是從 WebSphere Commerce 型錄擷取資訊，再以適合搜尋的格式來呈現該資訊。當搜尋空間更新時，搜尋空間和型錄會變成不同步。為了避免搜尋空間與型錄之間的同步受延遲，您可以啟用載入器的「產品顧問」搜尋空間同步化特性。

如果已啟用「產品顧問」搜尋空間同步化，載入器即會排定適當的搜尋空間指令，其方法是，如以下表格所示，將指令資訊新增至排程器工作表格 SCHCONFIG 與排程器工作案例表格 SCHSTATUS 中。

| 表格 | 模式 | 搜尋空間指令 | 狀況 |
|------------|----|----------------------------------|----------|
| CATENTRY | 更新 | UpdateSearchSpaces | 更新公佈旗號 |
| | 刪除 | RemoveProductFromAllSearchSpaces | 刪除產品 |
| CATENTDESC | 新增 | UpdateSearchSpaces | 新增說明/語言 |
| | 更新 | UpdateSearchSpaces | 更新公佈旗號 |
| | 刪除 | UpdateSearchSpaces | 更新說明資料 |
| LISTPRICE | 新增 | UpdateSearchSpaces | 新增標價 |
| | 更新 | UpdateSearchSpaces | 更新標價 |
| | 刪除 | UpdateSearchSpaces | 刪除標價 |
| ATTRVALUE | 新增 | UpdateSearchSpaces | 新增使用者定義值 |
| | 更新 | UpdateSearchSpaces | 更新使用者定義值 |
| | 刪除 | UpdateSearchSpaces | 刪除使用者定義值 |
| CATENTATTR | 新增 | UpdateSearchSpaces | 新增屬性 |
| | 更新 | UpdateSearchSpaces | 更新屬性 |
| | 刪除 | UpdateSearchSpaces | 刪除屬性 |
| CATGPENREL | 新增 | AddProductsToSearchSpace | 將產品新增至種類 |
| | 刪除 | RemoveProductsFromSearchSpace | 移除種類中的產品 |

請參閱第43頁的『第6章 載入資料』以取得其他資訊。

- **擷取程式**

「擷取程式」會對資料庫進行查詢，以便從資料庫中將選取的資料子集擷取至 XML 文件。您可以使用擷取過濾 XML 文件來指定要從資料庫中擷取的資料。

擷取程式的功能和載入器正好相反。您可使用「擷取程式」從 WebSphere Commerce 資料庫中，將選取的資料子集擷取至 XML 形式的檔案。例如，您可以取出與即將開始的假日相關的產品資料，也可以從合併資料庫中取出資訊而用於其它系統。

請參閱第61頁的『第7章 擷取資料』以取得其他資訊。

「載入器」套件也有日誌程式的功能。載入器套件中的每個公用程式都會建立訊息，以指出順利完成、失敗與錯誤，並提供程式追蹤資訊。

管理工具

Catalog Manager 所包含的以下工具可協助您管理其功能：

文字轉換工具

文字轉換工具可協助管理者準備必要資訊，以處理以字元界定的變數與 XML 資料格式間的資料轉換。

XSL 編輯器

「XML 轉換」使用「延伸樣式表語言 (XSL)」來定義將 XML 檔案轉換為其他 XML 檔案的規則。XSL 編輯器中的映射功能可提供管理者一個可視介面，以建立來源文件類型 (DTD) 中元素和目標 DTD 中元素間的關聯性。

第2章 轉換文字






Catalog Manager 可讓您將其他工具（例如試算表程式）所輸出的 ASCII 檔案，轉換為可輸入 WebSphere Commerce 資料庫的 XML 資料格式。


文字轉換工具可用來準備必要資訊，以處理以字元界定的變數與 XML 資料格式間的資料轉換。它提供了以下的檢視畫面：

1. 「文字綱目編輯檢視畫面」可讓您建立和修改轉換程序中所用的 XML 綱目檔。
2. 「轉換指令編輯檢視畫面」可讓您建立和修改用來執行轉換程序的實際指令。
3. 「轉換指令程序檢視畫面」可讓您啟動轉換程序。

啓動文字轉換工具

若要啓動「文字轉換」工具，請正確地使用 WebSphere Commerce 目錄中所提供的 Script 或指令：

-  `drive:\WebSphere\CommerceServer\bin\TextTrans.cmd`
-  `drive:\Program Files\WebSphere\CommerceServer\bin\ TextTrans.cmd`
-  `/usr/WebSphere/CommerceServer/bin/texttrans.sh`
-   `/opt/WebSphere/CommerceServer/bin/texttrans.sh`

 在 iSeries 環境中，管理者應該先複製必要的檔案，才能在 Windows NT 或 Windows 2000 機器上執行「文字轉換」工具。以下為執行方法的範例：

1. 在 Windows 機器上建立一個新目錄（例如，`drive:\TextTrans`）。
2. 在 `drive:\TextTrans` 中建立以下的子目錄：

```
\bin
\lib\loader
\wcsadmin
```

3. 請根據以下圖表，將檔案從 iSeries 機器複製至這些目錄中：

| 來源 | | 目的地 |
|----------------------------|--------|------------------------|
| /QIBM/ProdData/WebCommerce | | drive:\TextTrans |
| /bin | | \bin |
| /TextTrans.cmd | -----> | \TextTrans.cmd |
| /lib/loader | | \lib\loader |
| /wcmxmlp.jar | -----> | \wcmxmlp.jar |
| /wcmxslt.jar | -----> | \wcmxslt.jar |
| /wcsadmin | | \wcsadmin |
| /acsxml.ico | -----> | \acsxml.ico |
| /swing.jar | -----> | \swing.jar |
| /TextTransformerUI.zip | -----> | \TextTransformerUI.zip |
| /TextTransform.cnt | -----> | \TextTransform.cnt |
| /TextTransform.hlp | -----> | \TextTransform.hlp |
| /TextTransform.reg | -----> | \TextTransform.reg |

4. 修改 Windows 機器上的 `TextTrans.cmd` 檔，以插入下列文字：

```
set WCS_HOME=drive:\TextTrans
```

插入到下面的文字之前：

```
set lib=%WCS_HOME%\lib\loader
```

5. 在 Windows 機器上執行 `TextTrans.cmd`，以啓動文字轉換工具。

使用文字綱目編輯檢視畫面

下列程序主要是建立與修改在以字元界定的變數格式和 XML 格式間轉換資料時所用的 XML 綱目檔。

建立綱目檔

請執行以下步驟來建立新的綱目檔：

1. 啟動「文字轉換」工具。
2. 選取**檔案 > 新增**，或按下工具列中的「新增」圖示。
3. 選取路徑，然後鍵入所要建立的 XML 綱目檔名稱。

註：預設檔名為 "Default.xml"。

4. 選取 **CSV 格式**。

註：請勿選取 **WebSphere Commerce Suite 格式**。此選項僅適用於舊版 WebSphere Commerce。

5. 在完成時請按下**確定**。

您現在可以按照以下所指示的步驟來建立新的元素和屬性。

開啓綱目檔

請執行以下步驟來開啓綱目檔：

1. 啟動「文字轉換」工具。
2. 選取**檔案 > 開啓**，或按下工具列中的「開啓」圖示。
3. 選取要開啓的綱目檔。
4. 在完成時請按下**確定**。

您現在可以按照以下所指示的步驟來修改元素和屬性。

儲存綱目檔

請執行以下步驟來儲存綱目檔：

1. 啟動「文字轉換」工具。
2. 若要儲存您對綱目所做的任何變更，請選取**檔案 > 儲存**，或按下工具列中的「儲存」圖示。
3. 若要使用新名稱來儲存綱目的複本，請執行以下步驟：
 - a. 選取**檔案 > 另存新檔**。
 - b. 選取路徑然後鍵入要建立的 XML 檔案名稱。
 - c. 在完成時請按下**確定**。

編輯綱目檔

若要編輯綱目檔，請按照以上所述來開啓綱目檔，並且遵循程序。

新增元素

請執行以下步驟來新增元素：

1. 在「元素清單」欄位中鍵入新的元素名稱。

2. 按下「新增列」圖示。

移除元素

請執行以下步驟來移除元素：

1. 選取元素名稱。
2. 按下「移除列」圖示。

取代元素

請執行以下步驟來取代元素：

1. 在「元素清單」欄位中鍵入新的元素名稱。
2. 選取要取代的元素名稱。
3. 按下「取代列」圖示。

將元素移動至上一列

請執行以下步驟來將元素移動至上一列：

1. 選取要上移的元素名稱。
2. 按下「上移列」圖示。

將元素移動至下一列

請執行以下步驟來將元素移動至下一列：

1. 選取要下移的元素名稱。
2. 按下「下移列」圖示。

新增屬性

請執行以下步驟來新增屬性：

1. 在「屬性清單」欄位中鍵入新的屬性名稱。
2. 按下「新增列」圖示。

移除屬性

如果要移除屬性，請執行下列步驟：

1. 選取屬性名稱。
2. 按下「移除列」圖示。

取代屬性

如果要取代屬性，請執行下列步驟：

1. 在「屬性清單」欄位中鍵入新的屬性名稱。
2. 選取要取代的屬性名稱。
3. 按下「取代列」圖示。

將屬性移動至上一列

請執行以下步驟來將屬性移動至上一列：

1. 選取要上移的屬性名稱。
2. 按下「上移列」圖示。

將屬性移動至下一列

請執行以下步驟來將屬性移動至下一列：

1. 選取要下移的屬性名稱。
2. 按下「下移列」圖示。

變更綱目檔結構

「文字綱目編輯檢視畫面」之下窗格中的「檔案結構」檢視畫面是說明以字元界定的變數檔案的佈置。預期的檔案結構應該有下列欄位：

欄位分隔字元：

指定分隔屬性值的定界符號。預設為逗點 (",")。

記錄分隔字元：

指定分隔資料記錄的定界符號。預設為 "
" (相當於 `\r\n` 的實體參照)。

字串定界符號：

指定用來指出字串開始和結束邊界的定界符號。預設為單引號 (" ' ")。

內含標頭：

布林值；如果指定為 "true"，則文字資料檔中有標頭行，而若指定為 "false"，則文字資料檔中沒有標頭行。如果有標頭行，則因為在此狀況中會將標頭用來作為 XML 標籤名稱，所以必須符合標籤名稱的 XML 規則。預設為 "false"。

標頭行數：

指定存在於文字資料檔中的標頭行數。預設為零 ("0")。

準備一個綱目檔，以便將資料從 XML 轉換成以字元界定的變數格式

如果您使用以「文字綱目編輯檢視畫面」所建的 XML 綱目檔，來將資料從 XML 格式轉換成以字元界定的變數格式，您必須先使用文字編輯器將該綱目檔中的資料類型規格，從 "CSV 格式" 改為 "XML 格式"，然後再繼續進行轉換。

使用「轉換指令編輯檢視畫面」

透過「轉換指令編輯檢視畫面」，您可以建立一個新指令檔，開啓現有指令檔，或者儲存對指令檔所做的變更。預設指令檔名為 "Manifest.txt"。

您可以建立新指令、移除指令、以編輯過的資訊來取代指令或變更指令的次序。

註： 只要更新指令表格，即會自動儲存指令檔。

建立指令

如果要建立新的指令，請執行下列步驟：

1. 指定來源檔，可以是以字元區隔的變數格式檔（副檔名 .csv）或 XML 格式檔（副檔名 .xml）。
2. 指定要在轉換期間使用的 XML 綱目檔。
3. 指定在轉換程序期間所要建立或修改的輸出檔名稱（亦即，儲存新資料所在），可以是 XML 格式檔（副檔名 .xml），或是以字元區隔的變數格式檔（副檔名 .csv）。
4. 指定指令模式。

如果您要建立輸出檔，請選取**建立**；如果要將輸出資料附加到現有資料檔中，請選取**附加**。

5. 按下「新增列」圖示。

移除指令

如果要移除指令，請執行下列步驟：

1. 選取指令。
2. 按下「移除列」圖示。

編輯或取代指令

如果要編輯或取代指令，請執行下列步驟：

1. 按下「編輯指令」圖示。
列的資料會填入適當的輸入欄位中。
2. 請變更適當輸入欄位中的文字。
3. 按下「取代列」圖示來更新列。

將指令移動至上一列

請執行以下步驟來將指令移動至上一列：

1. 選取指令。
2. 按下「上移列」圖示。

註： 這樣可變更轉換程序的順序。

將指令移動至下一列

請執行以下步驟來將指令移動至下一列：

1. 選取指令。
2. 按下「下移列」圖示。

註： 這樣可變更轉換程序的順序。

清除指令

若要清除指令，請執行以下步驟：

1. 選取指令。
2. 按下「清除輸入欄位」圖示。

這樣會清除來源檔案、綱目檔案和輸出檔案欄位中的文字。

使用「轉換程序檢視畫面」

若要啟動文字轉換程序，請執行以下步驟：

1. 在「檔案」欄位中鍵入或瀏覽參數檔的名稱。
2. 按一下**程序**。

處理按鈕下的輸出區會顯示指出轉換程序狀態的資訊。您可以按下文字區底端的「儲存」按鈕來儲存輸出資訊，或按下「清除」按鈕來清除所有的狀態資訊。

第3章 轉換 XML 資料


「可延伸樣式表語言 (XSL)」可提供以下項目：


1. 指定 XML 文件格式的語言
2. 說明如何將 XML 檔轉換為其他一般結構檔案的語言

XSL 的轉換能利可用來將 XML 檔案轉換為其他符合不同 XML 綱目或 DTD 的 XML 檔案。

如果要轉換 XML 資料成為替代的 XML 格式，您必須利用轉換 XML 規則檔來指定轉換的規則。

以下的範例是將 MemberSubsystemExtracted.xml 中的資料，以 MemberSubsystem.xml 作為轉換 XSL 規則檔，而轉換為使用日文作為國家語言：

- 






```
java com.ibm.wca.XMLTransformer.XMLTransformer -infile MemberSubsystemExtracted.xml  
-transform MemberSubsystem.xml -outfile TransMbrStr.xml -param 'language="-10"'
```
- 


```
QWBCOMM/TRNWCXML INFILE(MemberSubsystemExtracted.xml)  
TRANSFORM(MemberSubsystem.xml) INSTR00T(/QIBM/UserData/WebCommerce/instances  
/my_inst)  
OUTFILE(TransMbrStr.xml) PARAM('language=-10')
```

「XML 轉換程式」是使用 XSL 來定義將 XML 檔案轉換為其他 XML 檔案的規則。XSL 編輯器中的映射功能可提供您一個可視介面，以建立來源 DTD 中元素和目標 DTD 中元素間的關聯性。有了這兩個 DTD 之後，您就可以項 XSL 規則來決定符合第一個（來源）DTD 的 XML 檔要如何轉換成符合第二個（目標）DTD 的檔案。

啓動 XSL 編輯器

若要啓動 XSL 編輯器，請使用 WebSphere Commerce 目錄中所提供的適當 Script 或指令：

-  `drive:\WebSphere\CommerceServer\bin\XSLEditor.cmd`
-  `drive:\Program Files\WebSphere\CommerceServer\bin\ XSLEditor.cmd`
-  `/usr/WebSphere/CommerceServer/bin/xsleditor.sh`
-   `/opt/WebSphere/CommerceServer/bin/xsleditor.sh`

 在 iSeries 環境中，管理者應先複製必要的檔案，才能在 Windows NT 或 Windows 2000 機器上執行 XSL 編輯器。以下爲執行方法的範例：

1. 在 Windows 機器上建立一個新目錄（例如 `drive:\XMLTrans`）。
2. 在 `drive:\TextTrans` 中建立以下的子目錄：

```
\bin
\lib\loader
\wcsadmin
```

3. 請根據以下圖表，將檔案從 iSeries 機器複製至這些目錄中：

| 來源 | | 目的地 |
|----------------------------|--------|-----------------------|
| /QIBM/ProdData/WebCommerce | | drive:\XMLTrans |
| /bin | | \bin |
| /XSLEditor.cmd | -----> | \XSLEditor.cmd |
| /lib/loader | | \lib\loader |
| /wcmxlp.jar | -----> | \wcmxlp.jar |
| /wcmxslt.jar | -----> | \wcmxslt.jar |
| /wcsadmin | | \wcsadmin |
| /config.dtd | -----> | \acsxml.ico |
| /swing.jar | -----> | \swing.jar |
| /XML_transform.ico | -----> | \XML_transform.ico |
| /XMLTransformerUI.zip | -----> | \XMLTransformerUI.zip |
| /XMLTransformWP.xml | -----> | \XMLTransformWP.xml |
| /XMLTransformWP.dtd | -----> | \XMLTransformWP.dtd |
| /XMLRuleConfig.xml | -----> | \XMLRuleConfig.xml |
| /XMLTransform.cnt | -----> | \XMLTransform.cnt |
| /XMLTransform.hlp | -----> | \XMLTransform.hlp |
| /XMLTransform.reg | -----> | \XMLTransform.reg |

4. 修改 Windows 機器上的 XSLEditor.cmd 檔，以插入下列文字：

```
set WCS_HOME=drive:\XMLTrans
```

插入到下面的文字之前：

```
set lib=%WCS_HOME%\lib\loader
```

5. 在 Windows 機器上執行 XSLEditor.cmd，以啓動 XSL 編輯器工具。

使用對應規則建置區

當您啓動 XSL 編輯器時，會顯示「映射規則建置區域」視窗。請使用此視窗來管理對映規則建置區。

建立對映規則建置區

若要建立對映規則建置區，請執行以下步驟：

1. 啓動 XSL 編輯器。
2. 請在下拉式功能表中選取 **[新增]**。
3. 請在「名稱」欄位中鍵入新對映規則建置區的名稱。
4. 請在「說明」欄位中鍵入新對映規則建置區的簡短說明。
5. 請在「來源綱目」欄位中鍵入現有檔案的名稱，或導覽至要用來作為來源綱目的現有檔案。
6. 請在「目標綱目」欄位中鍵入現有檔案的名稱，或導覽至要作為目標綱目的現有檔案。
7. 請在「XSL 規則檔案」欄位中鍵入要建立的新規則檔案名稱。
您可以在此指定完整的路徑。如果您不指定路徑，則會在現行工作目錄中建立檔案。
8. 按下**開啓**可建立和開啓新的對映規則建置區。

修改對映規則建置區

若要修改對映規則建置區，請執行以下步驟：

1. 啓動 XSL 編輯器。
2. 在下拉式功能表中選取您要修改的對映規則建置區。
3. 按下**開啓**以便開啓對映規則建置區。
4. 更新您要變更的欄位。
5. 按下**儲存**以便儲存變更。

刪除對映規則建置區

若要刪除對映規則建置區，請執行以下步驟：

1. 啓動 XSL 編輯器。
2. 在下拉式功能表中選取您要刪除的對映規則建置區。
3. 按下「移除」按鈕來移除項目。

註： 移除對映規則建置區不會刪除磁碟中的實體檔案。

使用「XSL 編輯器」

當您使用 XSL 編輯器開啓一個映射規則建置區域時，映射規則建置區域會出現在 XSL 編輯器主視窗中。

在「XSL 編輯器」主視窗中，左邊的窗格會顯示標示為「來源綱目」的來源 DTD 階層式檢視畫面。右邊的窗格會顯示標示為「目標窗格」的目標 DTD 階層式檢視畫面。

建立對映規則

元素間的對映

在來源階層中選取元素，並且將它拖放至目標階層的元素上。隨即產生 XSL 規則，並且會顯示在視窗底端的「對映規則」檢視畫面。

以下為產生的 XSL 規則範例：

```
<xsl:template match="merchant">
  <xsl:element name="MERCHANT">
    </xsl:element>
  </xsl:template>
```

註：任何必要但不存在的上代關係會自動產生。

屬性間的對映

在來源階層中選取屬性，並且將它拖放至目標階層的屬性上。隨即產生 XSL 規則，並且會顯示在視窗底端的「對映規則」檢視畫面。

以下為產生的 XSL 規則範例：

```
<xsl:attribute name="MEADDR1">
  <xsl:apply-templates select="@mecmail1"/>
</xsl:attribute>
```

註：任何必要但不存在的上代關係會自動產生。

建立自訂對映表示式

若要建立自訂的對映表示式，請先在目標階層中選取元素或屬性。然後按下滑鼠右鍵，並且選取「建立自訂表示式」功能表。隨即顯示「建立自訂表示式」視窗，而其中的兩個下拉式功能表中包含了可用的「範本」和「規則表示式」清單。請執行以下步驟來完成自訂表示式：

1. 選取要加入的自訂表示式之範本。
2. 選取要建立的規則表示式（例如**常數表示式**）。
3. 請在表格中列出的所有參數「值」直欄中鍵入值，然後按下 **Enter** 以確定值。
4. 按下**確定**以便完成建立步驟；或按下**取消**來取消動作而不建立規則。

所產生的 XSL 規則是根據規則架構檔 (XSLRuleConfig.xml) 中所定義的自訂表示式。您可以在需要時修改規則架構檔以及加入新規則。若要在「規則表示式」清單中使用該規則，請將該規則的「可見度」屬性設定為 "true"。

刪除對映規則

若要刪除對映規則，請執行以下步驟：

1. 在「對映規則」表格中選取規則。
2. 按下滑鼠右鍵然後選取**刪除**。
隨即刪除規則和所有衍生項目。

註：更新的對映規則和產生的 XSL 規則會自動持續存在。

處理 XML 轉換

請執行以下步驟來處理 XML 轉換：

1. 選取**工具 > 轉換**以便開啓「處理轉換」視窗。
2. 完成必要的欄位：
 - a. 請在「輸入 XML 檔案」欄位中鍵入或瀏覽來源 XML 資料檔的路徑和名稱。
 - b. 請在「XSL 規則檔」欄位中鍵入或瀏覽要用於轉換的對映規則檔路徑和名稱。
如果對映規則建置區已開啓，則此欄位中會預先填入目前在對映規則建置區中開啓的對映規則檔路徑。
 - c. 請在「輸出 XML 檔案」欄位中鍵入或瀏覽在轉換程序中建立的新 XML 資料檔路徑和名稱。
3. 按下**開始**可啓動 XML 轉換程序；或按一下**關閉**以便結束視窗不處理轉換。

自訂對映規則表格

若要自訂「對映規則」表格，請執行以下步驟：

1. 若要隱藏表格中的直欄，請在表格中的資料格上按下滑鼠右鍵，然後選取**隱藏直欄**。
2. 若要顯示表格中的隱藏直欄，請執行以下步驟：
 - a. 在表格中的資料格上按下滑鼠右鍵。
 - b. 選取**顯示直欄**來開啓隱藏直欄的清單。
 - c. 選取清單中的直欄。

註：請按住 **Shift** 鍵，然後按一下直欄名稱，就可以選取多個直欄。

- d. 按下**確定**可顯示所選取的直欄；按下**取消**可取消作業。
3. 若要顯示表格中所有的隱藏直欄，請在表格中的資料格上按下滑鼠右鍵，然後選取**顯示所有直欄**。
所有的直欄都會按照預設的次序來顯示。

顯示完整的 XSL 規則/值表示式

在「值表示式」或「XSL 規則」直欄的資料格上按一下，就會出現視窗，而且其中包含了所選取列的完整規則內容。

第4章 產生 DTD 和綱目

DTD 產生器可建立一個搭配載入器套件使用的 DTD 和綱目。DTD 產生器會根據您呼叫 DTD Generate 指令的方式來利用含有資料庫表格名稱的輸入檔，並產生 DTD，或產生 DTD 與含詳細 XML 檔（說明資料庫）的綱目。

設定 DTD 產生器

「載入器 DTD」會直接對映至 WebSphere Commerce 資料庫綱目。每個表格都是一項元素，每個直欄都是一個屬性。

範例：將「載入器 DTD」對映至資料庫綱目

| CATENTRY 的 DDL 陳述式 | DTD |
|-----------------------------------|-------------------------------|
| CREATE TABLE | <!ELEMENT CATENTRY EMPTY> |
| "CATENTRY" (| <!ATTLIST CATENTRY |
| "CATENTRY_ID" BIGINT NOT NULL | CATENTRY_ID CDATA #REQUIRED |
| "MEMBER_ID" BIGINT NOT NULL | MEMBER_ID CDATA #REQUIRED |
| "CATENTTYPE_ID" CHAR(16) NOT NULL | CATENTTYPE_ID CDATA #REQUIRED |
| "MARKFORDELETE" INTEGER NOT NULL | MARKFORDELETE CDATA #REQUIRED |
| "PARTNUMBER" VARCHAR(64) NOT NULL | PARTNUMBER CDATA #REQUIRED |
| "MFPARTNUMBER" VARCHAR(64) | MFPARTNUMBER CDATA #IMPLIED |
| "MFNAME" VARCHAR(64) | MFNAME CDATA #IMPLIED |
| "URL" VARCHAR(254) | URL CDATA #IMPLIED |
| "FIELD1" INTEGER | FIELD1 CDATA #IMPLIED |
| "FIELD2" INTEGER | FIELD2 CDATA #IMPLIED |
| "FIELD3" DECIMAL(20,5) | FIELD3 CDATA #IMPLIED |
| "FIELD4" VARCHAR(254) | FIELD4 CDATA #IMPLIED |
| "FIELD5" VARCHAR(254) | FIELD5 CDATA #IMPLIED |
| "LASTUPDATE" TIMESTAMP | LASTUPDATE CDATA #IMPLIED |
| "OID" VARCHAR(64) | OID CDATA #IMPLIED |
| "ONSPECIAL" INTEGER | ONSPECIAL CDATA #IMPLIED |
| "ONAUCTION" INTEGER | ONAUCTION CDATA #IMPLIED |
| "BUYABLE" INTEGER | BUYABLE CDATA #IMPLIED |
| "BASEITEM_ID" INTEGER | BASEITEM_ID CDATA #IMPLIED |
| "CLASSIFGRP_ID" INTEGER | CLASSIFGRP_ID CDATA #IMPLIED |
| "ITEMSPC_ID" INTEGER | ITEMSPC_ID CDATA #IMPLIED |
| "STATE" INTEGER | STATE CDATA "1" |
|); | > |

您可以執行以下步驟來設定「DTD 產生器」的運作方式：

1. 建立新的「DTD 產生器」自訂程式內容檔。



DB2ConnectionCustomizer.properties 是保存在 DTDGenerator.zip。請解開這個檔案，並重新命名（仍採用 .properties 副檔名），並將之置於類別路徑內的目錄中。**重要事項：**請勿移除或修改現有的 DB2ConnectionCustomizer.properties 檔案。

•  400

`ISeries_GENWCSDTD_Customizer.properties` 是在 `/QIBM/ProdData/WebCommerce/properties` 目錄中。將此檔案複製至 `/instroot/xml` 目錄，然後為新檔案重新命名，但是保留 `.properties` 副檔名，接著對新檔案進行任何必要的變更。**重要事項：** 請勿移除或修改原來的 `ISeries_GENWCSDTD_Customizer.properties` 檔案。

2. 修改新檔案中的資料庫驅動程式值。比方說：

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

其中：

- `DBVendorName` 是用來選取資料庫類型。

選項如下：

- DB2 Universal Database for iSeries (DB2/iSeries)
- 其他作業系統上的 DB2 (DB2)
- Oracle 資料庫 (Oracle)

- `DBDriverName` 是用來選取 JDBC 驅動程式。

選項如下：

- DB2 Universal Database for iSeries (`com.ibm.db2.jdbc.app.DB2Driver`)
- 其他作業系統上的 DB2 (`COM.ibm.db2.jdbc.app.DB2Driver`)
- Oracle 資料庫 (`oracle.jdbc.driver.OracleDriver`)

- `DBURL` 是用來指定存資料庫時要使用的 URL。

選項如下：

- DB2 Universal Database for iSeries (`jdbc:db2://`)
- 其他作業系統上的 DB2 (`jdbc:db2:`)
- Oracle 資料庫 (`jdbc:oracle:oci8:@`)

3. 指定新檔案名稱作為 `DTD Generate` 指令的自訂程式參數值。

產生 DTD

`TableNames.txt` 輸入檔包含了以下的資料庫表格名稱，而每一行包含一個名稱：

```
MEMBER
ADDRBOOK
ADDRESS
```

以下是如何呼叫 `DTD` 產生器的範例：

-  NT  2000  AIX  Solaris  Linux

```
java com.ibm.wca.DTDGenerator.GenerateDTD -dbname MALL
-dbuser db2inst1
-dbpwd db2ibm -outfile wc.dtd -infile TableNames.txt
```

•  400



```
QWECOMM/GENWCSDTD DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(my password) OUTFILE(wc.dtd)
INFILE(TableNames.txt)
```


輸出檔 wc.dtd 包含了以下內容：


```
<!ELEMENT MALL (( MEMBER | ADDRBOOK | ADDRESS)*)>
<!ELEMENT MEMBER EMPTY>
<!ATTLIST MEMBER
  MEMBER_ID      CDATA #REQUIRED
  TYPE           CDATA #REQUIRED
  STATE          CDATA #IMPLIED
>
<!ELEMENT ADDRBOOK EMPTY>
<!ATTLIST ADDRBOOK
  ADDRBOOK_ID    CDATA #REQUIRED
  MEMBER_ID      CDATA #REQUIRED
  TYPE           CDATA #IMPLIED
  DISPLAYNAME    CDATA #REQUIRED
  DESCRIPTION    CDATA #IMPLIED
>
<!ELEMENT ADDRESS EMPTY>
<!ATTLIST ADDRESS
  ADDRESS_ID     CDATA #REQUIRED
  ADDRESSTYPE   CDATA #IMPLIED
  MEMBER_ID     CDATA #REQUIRED
  ADDRBOOK_ID  CDATA #REQUIRED
  ORGUNITNAME   CDATA #IMPLIED
  FIELD3        CDATA #IMPLIED
  BILLINGCODE   CDATA #IMPLIED
  BILLINGCODETYPE CDATA #IMPLIED
  STATUS        CDATA #IMPLIED
  ORGNAME       CDATA #IMPLIED
  ISPRIMARY     CDATA #IMPLIED
  LASTNAME     CDATA #IMPLIED
  PERSONTITLE  CDATA #IMPLIED
  FIRSTNAME    CDATA #IMPLIED
  MIDDLENAME   CDATA #IMPLIED
  BUSINESSTITLE CDATA #IMPLIED
  PHONE1       CDATA #IMPLIED
  FAX1         CDATA #IMPLIED
  PHONE2       CDATA #IMPLIED
  ADDRESS1     CDATA #IMPLIED
  FAX2         CDATA #IMPLIED
  NICKNAME     CDATA #REQUIRED
  ADDRESS2     CDATA #IMPLIED
  ADDRESS3     CDATA #IMPLIED
  CITY         CDATA #IMPLIED
  STATE        CDATA #IMPLIED
  COUNTRY      CDATA #IMPLIED
  ZIPCODE      CDATA #IMPLIED
  EMAIL1       CDATA #IMPLIED
  EMAIL2       CDATA #IMPLIED
  PHONE1TYPE   CDATA #IMPLIED
  PHONE2TYPE   CDATA #IMPLIED
  PUBLISHPHONE1 CDATA #IMPLIED
  PUBLISHPHONE2 CDATA #IMPLIED
  BESTCALLINGTIME CDATA #IMPLIED
  PACKAGESUPPRESSION CDATA #IMPLIED
  LASTCREATE   CDATA #IMPLIED
  OFFICEADDRESS CDATA #IMPLIED
  SELFADDRESS  CDATA "0"
  FIELD1       CDATA #IMPLIED
  FIELD2       CDATA #IMPLIED
  TAXGEOCODE   CDATA #IMPLIED
  SHIPPINGGEOCODE CDATA #IMPLIED
>
```

產生綱目和詳細的 XML 檔案

在這個範例中，會依照下列方式來呼叫 DTD 產生器：

- 

```
java com.ibm.wca.DTDGenerator.GenerateDTD -dbname SAMPLE
-dbuser johndoe -dbpwd password -xmlTableDesc c:\sample\sample.xml
-outfile tables.dtd -tablenames "employee,staff"
```
- 

```
java com.ibm.wca.DTDGenerator.GenerateDTD -dbname SAMPLE
-dbuser johndoe -dbpwd password -xmlTableDesc usr/sample/sample.xml
-outfile tables.dtd -tablenames "employee,staff"
```
- 

```
QWEBCOMM/GENWCSDTD DATABASE(MYDB) SCHEMA(SAMPLE)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
OUTFILE(tables.dtd)
TABNAMES('employee,staff') XMLTABDESC(/sample/sample.xml)
```

綱目檔會建立在 sample 目錄，而檔名為 WCAWebForm.xsd。sample.xml 輸出檔中含有下列內容：

```
<?xml version="1.0" encoding="UTF-8"?>
<formList xmlns="WCAWebForm.xsd" dbname="SAMPLE" dtdname="tables.dtd">
  <form name="EMPLOYEE">
    <uniqueIndex name="U2" columns="FIRSTNAME, LASTNAME"/>
    <uniqueIndex name="U3" columns="MIDINIT, LASTNAME"/>
    <field name="EMPNO" type="string" maxLength="6" minOccurs="1"
      uniqueKey="true" showColumnInList="true" />
    <field name="FIRSTNAME" type="string" maxLength="32" minOccurs="1"
      showColumnInList="true" />
    <field name="MIDINIT" type="string" maxLength="1" minOccurs="1"
      showColumnInList="true" />
    <field name="LASTNAME" type="string" maxLength="15" minOccurs="1"
      showColumnInList="true" />
    <field name="WORKDEPT" type="string" maxLength="3" showColumnInList="true" />
    <field name="PHONENO" type="string" maxLength="4"/>
    <field name="HIREDATE" type="date" maxLength="10"/>
    <field name="JOB" type="string" maxLength="8"/>
    <field name="EDLEVEL" type="integer" maxLength="5" minOccurs="1" />
    <field name="SEX" type="string" maxLength="1"/>
    <field name="BIRTHDATE" type="date" maxLength="10"/>
    <field name="SALARY" type="decimal" maxLength="9"/>
    <field name="BONUS" type="decimal" maxLength="9"/>
    <field name="COMM" type="decimal" maxLength="9"/>
  </form>
  <form name="STAFF">
    <field name="ID" type="integer" maxLength="5" minOccurs="1"
      uniqueKey="true" showColumnInList="true" />
    <field name="NAME" type="string" maxLength="9" showColumnInList="true" />
    <field name="DEPT" type="integer" maxLength="5" showColumnInList="true" />
    <field name="JOB" type="string" maxLength="5" showColumnInList="true" />
    <field name="YEARS" type="integer" maxLength="5" showColumnInList="true" />
    <field name="SALARY" type="decimal" maxLength="7"/>
    <field name="COMM" type="decimal" maxLength="7"/>
  </form>
</formList>
```

第5章 解析識別碼

要載入到目標資料庫的 XML 資料必須取得需要它們的 XML 元素的識別碼。如果要產生或找到 XML 文件中之型錄項目的識別碼，請呼叫 ID Resolve 指令。

ID 解析程式只會解析主要表格的識別碼。主要表格會列於 KEYS 或 SUBKEYS 表格之中。如果您需要解析未列在 KEYS 或 SUBKEYS 中之表格的識別碼，請先將該表格新增到 SUBKEYS 表格中，然後再執行 ID 解析程式。

舉例來說，在下列情況下您或許會想使用 ID 解析程式：

- 當您採用 XML 格式來載入新內容，且需要資料的識別碼時
- 當您要更新內容，而資料庫中已有物件的識別碼時

ID 解析程式可提供實際的識別碼，您也可以採用下列方式來解析識別碼：

- 內部別名解析
- 內容檔規格
- 唯一索引解析

設定「ID 解析程式」

您可以採取下列步驟來設定「ID 解析程式」處理時間戳記、儲存體及資料庫驅動程式的方式：

1. 建立新的「ID 解析程式」自訂程式內容檔。



DB2ConnectionCustomizer.properties 是保存在 IdResGen.zip。請解開這個檔案，並重新命名（仍採用 .properties 副檔名），並將之置於類別路徑內的目錄中。**重要事項：**請勿移除或修改現有的 DB2ConnectionCustomizer.properties 檔。



ISeries_RESWCSID_Customizer.properties 是在 /QIBM/ProdData/WebCommerce/properties 目錄中。將此檔案複製至 *instroot/xml* 目錄，然後為新檔案重新命名，但是保留 .properties 副檔名，接著對新檔案進行任何必要的變更。**重要事項：**請勿移除或修改原來的 iSeries_RESWCSID_Customizer.properties 檔案。

2. 修改新檔案中所指定的內容值。
3. 指定新檔案名稱作為 ID Resolve 指令的自訂程式參數值。

設定「ID 解析程式」處理時間戳記的方式

在「ID 解析程式」的自訂程式內容檔中，已提供了以下的預設輸入時間戳記遮罩：

```
InputTimeStampFormat.1 = yyyy-DD hh:mm:ss.SSSSSS
InputTimeStampFormat.2 = yyyy-MM-dd hh:mm:ss.SSSSSS
InputTimeStampFormat.3 = yyyy-DD-hh.mm.ss.SSSSSS
InputTimeStampFormat.4 = yyyy-MM-dd-HH.mm.ss.SSSSSS
InputTimeStampFormat.5 = yyyy-MM-dd-hh.mm.ss.SSSSSS
InputTimeStampFormat.6 = yyyy-MM-dd HH:mm:ss.SSSSSS
InputTimeStampFormat.7 = yyyy-DD HH:mm:ss.SSSSSS
```

您可以修改這些時間戳記遮罩，或在「ID 解析程式」的自訂程式內容檔中任意加入遮罩。如果您新增輸入時間戳記，就必須利用目前順序中的下個號碼。（比方說，如果您新增到上述清單中，下個輸入時間戳記遮罩就會是 `InputTimeStampFormat.8`。）

您也可以在此「ID 解析程式」的自訂程式內容檔中修改下列內容值，來自訂輸出時間戳記格式、毫秒遮罩及資料庫專用格式：

```
TargetTimeStampFormat = yyyy-MM-dd HH:mm:ss.SSSSSS
MicroSecondMask = SSSSSS
DatabaseSpecificFormat = YYYY-MM-DD HH24:MI:SS
```

設定「ID 解析程式」處理儲存體的方式

以下為「ID 解析程式」自訂程式內容檔的段落，而其中指定與持續性雜湊對映有關的內容預設值：

```
////////////////////////////////////
/// 0 = 沒有後端儲存體的一般雜湊對映
/// 1 = JDBM
////////////////////////////////////

PersistentStorageType = 0

////////////////////////////////////
/// 如果 PersistentStorageType != 0，請將 MemoryStorageSize 設為記憶體
/// 資料中之雜湊映射的最大大小，而在該雜湊映射後，會將資料引至
/// 指定的永久儲存體中。
/// 若為 -1，則會使用正常的雜湊映射，而不使用後端儲存體
////////////////////////////////////

MemoryStorageSize = 1
```

您可以在「ID 解析程式」自訂程式內容檔中設定 `PersistentStorageType` 的值，以便指定「ID 解析程式」處理持續性儲存體的方式。

- 如果您設定了 `PersistentStorageType = 0`，則「ID 解析程式」會以「一般」方式運作（符號雜湊對映存在於記憶體中）。
- 如果設定 `PersistentStorageType = 1`，JDBM 會用來持續儲存符號和鍵值。

您可以設定「ID 解析程式」自訂程式內容檔的 `MemoryStorageSize` 值，以指定儲存在記憶體中的記錄數量。

- `MemoryStorageSize` 的值為 1 表示只在記憶體中保留一筆記錄。
- `MemoryStorageSize` 的值為 -1 時則有特殊意義；他表示所有的記錄都會保存在記憶體中。

在此狀況下，「ID 解析程式」會回復至「一般」行為。

設定「ID 解析程式」處理資料庫驅動程式的方式

「ID 解析程式」自訂程式內容檔的以下行內容，指定了資料庫驅動程式的預設值：

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

其中：

- DBVendorName 是用來選取資料庫類型。
選項如下：
 - DB2 Universal Database for iSeries (DB2/iSeries)
 - 其他作業系統上的 DB2 (DB2)
 - Oracle 資料庫 (oracle)
- DBDriverName 是用來選取 JDBC 驅動程式。
選項如下：
 - DB2 Universal Database for iSeries (com.ibm.db2.jdbc.app.DB2Driver)
 - 其他作業系統上的 DB2 (COM.ibm.db2.jdbc.app.DB2Driver)
 - Oracle 資料庫 (oracle.jdbc.driver.OracleDriver)
- DBURL 是用來指定存資料庫時要使用的 URL。
選項如下：
 - DB2 Universal Database for iSeries (jdbc:db2://)
 - 其他作業系統上的 DB2 (jdbc:db2:)
 - Oracle 資料庫 (jdbc:oracle:oci8:@)


決定資料的處理方式

ID Resolve 指令可讓您選擇用載入、更新或混合方法來處理輸入檔。

- 如果檔案中的**所有**記錄在資料庫中都不存在，請使用載入方法來處理輸入檔。
- 如果檔案中的**所有**記錄都**存在**於資料庫中，請使用更新方法來處理輸入檔。
- 如果只有檔案中的**部份**記錄**存在**於資料庫中，請使用混合方法來處理輸入檔。

選擇載入方法

「ID 解析程式」的載入方法用來產生載入到資料庫的記錄的新識別碼。在這個模式下，會建立記錄的新識別碼。下列範例用來產生新資料的識別碼：

```
• 
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method load -customizer customizer -schemaname mall
```

載入方法是預設值。

```
• 
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method load -customizer customizer -schemaname mall
```

載入方法是預設值。

- ▶ 400

```
QWEBCOMM/RESWCSID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(mypassword) INFILE(input.xml) OUTFILE(output.xml)
METHOD(*LOAD)
```

選擇更新方法

如果您指定「ID 解析程式」的更新方法，輸入檔中的記錄應該已在資料庫中。ID 解析程式會在資料庫中找出識別碼。如果有記錄不在資料庫中，ID 解析程式便無法解析這項記錄的識別碼，而會指出發生錯誤。下列範例用來尋找已在資料庫中之資料的識別碼：

- ▶ NT ▶ 2000

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method update -customizer customizer -schemaname mall
```

- ▶ AIX ▶ Solaris ▶ Linux

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method update -customizer customizer -schemaname mall
```

- ▶ 400

```
QWEBCOMM/RESWCSID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
INFILE(input.xml) OUTFILE(output.xml) METHOD(*UPD)
```

選擇混合方法

如果輸入資料檔含有已在資料庫中的記錄及部份新資料，就必須使用混合方法來執行「ID 解析程式」。在這個模式下，記錄必須不在資料庫中，ID 解析程式才會建立記錄的識別碼。否則，會從資料庫中取得現有的識別碼。下列範例用來產生新資料的識別碼及尋找已在資料庫中之資料的識別碼：

- ▶ NT ▶ 2000

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method mixed -customizer customizer -schemaname mall
```

- ▶ AIX ▶ Solaris ▶ Linux

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method mixed -customizer customizer -schemaname mall
```

- ▶ 400

```
QWEBCOMM/RESWCSID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
INFILE(input.xml) OUTFILE(output.xml) METHOD(*MIX)
```

使用 ID 解析技術

「ID 解析程式」可以使用 Java 內容檔來決定要使用哪些主要表格的直欄，來查閱需要主要表格識別碼的表格。如果表格列於 KEYS 或 SUBKEYS 表格，則為主要表格。

若要使用 ID 解析程式的內部別名解析，別名會放在 XML 檔的主要鍵屬性（識別碼）中。之後，XML 檔可以利用別名來指向這個元素。這個程序使程式不需要決定建置 XML 檔所需要的唯一索引。

「ID 解析程式」也可以分析資料庫綱目來判斷有沒有唯一索引能滿足它的需求。只有在表格之內容檔中沒有要分析的項目時，或沒有內容檔時，ID 解析程式才會尋找唯一索引。如果符合這些情況，則會進行唯一索引檢查。只要唯一索引存在，且不含表格的主要鍵，則該唯一索引即為有效的。

使用「ID 解析程式」來指定內容檔

「ID 解析程式」可讓您使用其他的 Java 內容檔，來說明要使用哪些主要項目的直欄來查詢需要主要列識別碼的表格。

預設內容檔是 `IdResolveKeys.properties`，但您可視需要在呼叫 `ID Resolve` 指令時修改這個內容檔或者指定自己的檔案。

- 

`IdResolveKeys.properties` 位於下列目錄中：

-  `drive:\WebSphere\CommerceServer\properties`
-  `drive:\Program Files\WebSphere\CommerceServer\properties`
-  `/usr/WebSphere/CommerceServer/properties`
-   `/opt/WebSphere/CommerceServer/properties`

在您執行 ID 解析程式時，如果您不是將此檔案置於現行目錄中，您可將之置於類別路徑環境變數中所定義的目錄中。您也可以指定此檔案的完整路徑。

- 

若要變更 `IdResolveKeys.properties`，請從 `/QIBM/ProdData/WebCommerce/properties` 目錄複製該檔，然後將它儲存至 `/instroot/xml` 目錄，接著在對新檔案進行必要的變更。

註： 上述目錄位於 `RESWC SID` 指令所用的類別路徑中。

使用內容檔來產生識別碼

在下列範例中，您需針對 ADDRBOOK 與 ADDRESS 記錄分別解析其 ADDRBOOK_ID 與 ADDRESS_ID 識別碼。我們已知 MEMBER 記錄的識別碼。每一筆記錄在 WebSphere Commerce 資料庫中都需要有一個有效的識別碼。此外 ADDRESS 記錄中的 ADDRBOOK_ID 需採用主要表格中的識別碼，以滿足其外來鍵限制。

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  MEMBER_ID="100"
  DISPLAYNAME="Friends"           DISPLAYNAME 直欄的實際值
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRBOOK_ID="@Friends"         使用 DISPLAYNAME 值做為查閱，以參照 ADDRBOOK
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>
```

您需要一個內容檔來指出關係列會利用主要列中的哪些直欄來解析外來鍵直欄的識別碼。下列程序可確保上述檔案的剖析能正確進行。

請在 IdResolveKeys.properties 中指定以下內容：

```
NAMEDELIMITER=@
SELECTDELIMITER=:

ADDRBOOK=@DISPLAYNAME:DISPLAYNAME
ADDRESS=@NICKNAME:NICKNAME
```

NAMEDELIMITER 與 SELECTDELIMITER 用以設定整個內容檔中所用的定界符號，且其使用必須一致。

ADDRBOOK=@DISPLAYNAME:DISPLAYNAME 是說明在收到通訊錄記錄時，建立通訊錄列的識別碼。DISPLAYNAME 欄位擷取自輸入記錄，且用以構成和新識別碼間的連結。DISPLAYNAME 字串是用來比對通訊錄列 DISPLAYNAME，以解析外來鍵所需的識別碼。

在上述輸入範例中，DISPLAYNAME 為 Friends，並假定建給這個記錄的識別碼是 12951。DISPLAYNAME 用來作為 129512 的鍵值參照。繼續處理下筆記錄，ADDRESS；其中 ADDRBOOK_ID 的格式為 "@..."（表示定界符號後的字元用來查閱通訊錄識別碼）。這個字串會比對 DISPLAYNAME，並傳回 12951，且置於 ADDRBOOK_ID 屬性中。

```

<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="12951"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="13051"
  ADDRBOOK_ID="12951"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

產生的主要鍵
未變更的 ADDRBOOK DISPLAYNAME 值
產生的主要鍵
ADDRESS 指向正確的 ADDRBOOK

透過複合鍵來使用內容檔

由兩個以上直欄組成的鍵值即稱為複合鍵值。您可以在欄位名稱前面指定 NAMEDELIMITER 和 SELECTDELIMITER，以在內容檔中定義複合鍵值的查閱。比方說，如果要顯示名稱和成員 ID 的組合作為 ADDRBOOK 記錄的查閱準則，請在內容檔中指定下列內容：

```
ADDRBOOK=@DISPLAYNAME@MEMBER_ID:DISPLAYNAME MEMBER_ID
```

之後，XML 輸入檔的片段如下：

```

<ADDRBOOK
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRBOOK
  MEMBER_ID="101"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>

```

第 100 號成員的 ADDRBOOK "Friends"
第 101 號成員的 ADDRBOOK "Friends"

```

<ADDRESS
  ADDRBOOK_ID="@Friends@100"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

**查閱成員 100 之 ADDRBOOK
"Friends" 的主要鍵**

會在解析之後，產生如下結果：

```

<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="12951"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRBOOK
  ADDRBOOK_ID="12952"
  MEMBER_ID="101"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="13051"
  ADDRBOOK_ID="12951"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

需要的 ADDRBOOK

ADDRESS 指向正確的 ADDRBOOK

透過連鎖反應主要鍵來使用內容檔

主要表格 STOREENT 定義了主要鍵 STOREENT_ID。參照 STOREENT 的外部表格 STORE，定義了作為主要表格 STOREENT 外部鍵的主要鍵 STORE_ID。這表示 STORE_ID 值必須是某個 STOREENT_ID 值。因此，外部表格 STORE 的主要鍵 STORE_ID 扮演雙重角色 -- 主要和外來。

我們假設另一份表格 CONTRACT 是 STORE 中的外部表格且 CONTRACT 的外來鍵 STORE_ID 會參照 STORE 中的主要鍵 STORE_ID。因此，STORE 表格是 CONTRACT 表格的主要表格。

由於 STORE 表格的 STORE_ID 是參照 STOREENT_ID 而來，而非建立而來，因此，ID 解析程式不會為 STORE 表格建立一個「內部別名/ID 值」連結。當 CONTRACT 表格試圖從 STORE 表格解析 STORE_ID 時，它會得到空值。

由於這個特殊狀況，您必須在內容檔中建立一個項目來明確指定建立內部別名。在 IdResolveKeys.properties 中指定以下內容：

```
"STORE=@STORE_ID:STORE_ID"
```

這樣會強制「ID 解析程式」執行以下步驟：

- 在將 STORE_ID 解析成外部參照時，建立內部別名和 ID 值的關聯
- 在解析 CONTRACT 表格的 STORE_ID 時使用這個關聯

使用內容檔中的 STORE=@STORE_ID:STORE_ID 項目及下列 XML 輸入檔片段：

```
<STOREENT
  IDENTIFIER="Out Fashions"
  MEMBER_ID="-2000"
  STOREENT_ID="@storeent_id_1"
  TYPE="G"
/>
<STORE
  STORE_ID="@storeent_id_1"
  STOREGRP_ID="1"
  STORELEVEL="store_level"
/>
<CONTRACT
  CONTRACT_ID="@contract_id_1"
  STATE="0"
  STORE_ID="@storeent_id_1"
/>
```

會在解析之後，產生如下結果：

```
<STOREENT
  IDENTIFIER="Out Fashions"
  MEMBER_ID="-2000"
  STOREENT_ID="10501"
  TYPE="G"
/>
<STORE
  STORE_ID="10501"
  STOREGRP_ID="1"
  STORELEVEL="store_level"
/>
<CONTRACT
  CONTRACT_ID="@contract_id_1"
  STATE="0"
  STORE_ID="10501"
/>
```

使用內部別名解析

若要使用 ID 解析程式的內部別名解析，別名會放在 XML 檔的主要鍵屬性（識別碼）中。之後，XML 檔可以利用別名來指向這個元素。這個程序使程式不需要決定建置 XML 檔所需要的唯一索引。

在檔案中，內部別名的使用必須一致。如果將通訊錄 ID ADDRBOOK_ID 的別名設為 @addrbook_1，檔案內所有參照這個 ID 的外來鍵都必須使用 @addrbook_1。請注意，別名是暫時的。它們不會儲存；如果沒有再次引進別名的話，在個別的 XML 檔中也無法使用它們。

使用內部別名 ID 解析的局部範例

解析前：

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="@addrbook_1"           ADDRBOOK 的別名
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="@address_1"             ADDRESS 的別名
  ADDRBOOK_ID="@addrbook_1"         參照 ADDRBOOK 的別名
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="1 Brave Developer St."
  CITY="Toronto"
  ZIPCODE="A3B0F4"
  COUNTRY="Canada"
  STATUS="P"
/>
```

解析後：

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="11801"                 產生的主要鍵
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="11901"                 產生的主要鍵
  ADDRBOOK_ID="11801"               參照 ADDRBOOK 項目
  MEMBER_ID="100"
  NICKNAME="Bob"
  ADDRESS1="1 Brave Developer St."
  CITY="Toronto"
  ZIPCODE="A3B0F4"
  COUNTRY="Canada"
  STATUS="P"
/>
```

使用單一索引解析

只有在內容檔沒有分析中的表格的項目或沒有內容檔時，才使用單一索引解析，也就是 ID 解析程式的預設行為。單一索引解析會使用在表格中所指定的任何唯一索引來找

出識別碼。比方說，MEMBER_ID 配上 IDENTIFIER 是 CATALOG 表格中的唯一索引，因而可作為 CATALOGDSC 表格之主要鍵 CATALOG_ID 的解析點。

如果要更新資料庫的內容，您必須知道資料庫中之主要表格的唯一鍵值。您可以查詢您的資料庫來找出它。比方說，擷取唯一鍵值的 DB2 指令如下：

```
db2 describe indexes for table schema.tablename show detail
```

唯一索引解析的局部範例

解析前:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="0"
  STATE="1"
/>

<CATALOG
  DESCRIPTION="Winter Catalog"
  IDENTIFIER="WC2001"
  MEMBER_ID="100"
  TPCLEVEL="2"
/>

<CATALOGDSC
  CATALOG_ID="@WC2001@100"           回頭參照成員 "100" 的型錄 "WC2001"

  (附註：務請注意次序。 )
  FULLIMAGE="c:\store\img\wc.gif"
  LANGUAGE_ID="-1"
  LONGDESCRIPTION="2001 Winter Catalog"
  SHORTDESCRIPTION="2001 Winter Catalog"
  NAME="InFashion 2001 Winter Catalog"
  THUMBNAIL="c:\store\img\wc_th.gif"
/>
```

解析後:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="0"
  STATE="1"
/>

<CATALOG
  CATALOG_ID="10351"                 自動產生的主要鍵
  DESCRIPTION="Winter Catalog"
  IDENTIFIER="WC2001"
  MEMBER_ID="100"
  TPCLEVEL="2"
/>

<CATALOGDSC
  CATALOG_ID="10351"                 參照正確的型錄
  FULLIMAGE="c:\store\img\wc.gif"
  LANGUAGE_ID="-1"
  LONGDESCRIPTION="2001 Winter Catalog"
  SHORTDESCRIPTION="2001 Winter Catalog"
  NAME="InFashion 2001 Winter Catalog"
  THUMBNAIL="c:\store\img\wc_th.gif"
/>
```

將資料載入至 MEMBER 表格

ID 解析程式用以處理有產生識別碼（由系統產生）之表格的解析。其中包括 KEYS 或 SUBKEYS 表格中登錄的任何表格與直欄。此項解析有兩個元件：

1. 判斷資料庫中是否有主要表格（亦即，KEYS 或 SUBKEYS 中列出的表格）存在。
此項解析是根據該元素的 XML 資料內容，而採用唯一索引解析方式或內容檔規格方式。
2. 判斷主要表格是否有外來鍵。
其做法是解析相關表格之外來鍵屬性中的規格。

MEMBER 表格是作為 ORGENTITY、MBRGRP 與 USER 表格的「超類別」。這會建立一種「is-a」型樣，當表格對 MEMBER 的子類型有外來鍵方面的限制時，此種型樣有助於維護參照完整性。但由於所有 MEMBER 子類型都共用一個共通的基礎類型，識別碼在子類型間必須是唯一的。也就是說，在 MBRGRP_ID 與 USER_ID 組中，ORGENTITY_ID 必須是唯一的。也因此，KEYS 表格只能參照 ORGENTITY、MBRGRP 與 USER 表格，並針對它們的識別碼指定互斥範圍。每一個子類型都有一個主要鍵；每一個主要鍵也各有一個參照 MEMBER 表格主要鍵的外來鍵。

因 MEMBER 與其子類型間的限制，導致 MEMBER 與子類型不能有一個同步化的 ID。為了將 ORGENTITY、MBRGRP 和 USER 表格載入至系統，「ID 解析程式」會辨識「is-a」型樣，並且適當地進行處理。ID 解析程式的 XML 語法如下：

```
<ORGENTITY
  ORGENTITY_ID="@orgAlias"
  ORGENTITYNAME="Test Org"
  ORGENTITYTYPE="0">
  <ISA>
  <MEMBER
    TYPE="0"
    STATE="1"
  />
  </ISA>
</ORGENTITY>
```

產生的結果如下：

```
<MEMBER
  MEMBER_ID="12345"
  TYPE="0"
  STATE="1"
/>
<ORGENTITY
  ORGENTITY_ID="12345"           與成員元素同步
  ORGENTITYNAME="Test Org"
  ORGENTITYTYPE="0"
/>
```

透過此方式，ID 解析程式處理 <isa> 子元素並建立一個同步化的識別碼。

使用 REFKEYS 表格來建立外部關係

REFKEYS 表格用來代表一些未存在於資料庫中之表格間的外來關係。一般而言，資料庫綱目會建立一項外來鍵宣告（以便讓表格中的直欄鏈結至另一表格），來說明外來鍵關係。如果資料庫綱目中並未定義外來鍵關係，而識別碼必須解析成外來鍵時，請執行下列步驟：

1. 按照以下範例 DDL 所示建立 REFKEYS 表格：

```
CREATE TABLE "REFKEYS" (  
    "FKTABLE_NAME" CHAR(18) NOT NULL ,  
    "FKCOLUMN_NAME" CHAR(18) NOT NULL ,  
    "TABLENAME" CHAR(18) NOT NULL  
);
```

其中：

FKTABLE_NAME 是指外部（或「子項」）表格名稱

FKCOLUMN_NAME 是指外部直欄名稱

TABLENAME 是指主要（或「母項」）表格名稱

2. 在 REFKEYS 表格中建立一個項目，以說明必要的外來關係。

錯誤疑難排解

如果在解析識別碼時發生錯誤，請參閱下表：

| 錯誤 | 使用的方法 | 可能的原因 | 可能的解決方案 |
|---------|-------|---|---|
| 未解析的主要鍵 | All | 對於 KEYS 或 SUBKEYS 表格中未指定的主要鍵（識別碼），「ID 解析程式」將不會解析表格中的主要鍵（識別碼）。 | <p>將要解析主要鍵的表格名稱新增至 SUBKEYS 表格中，然後再執行「ID 解析程式」。</p> <p>此外，確定資料庫綱目中已有外來鍵關係。</p> |
| | 更新 | 主要鍵是藉由查詢資料庫來解析。使用 properties-file 項目資訊或給定表格的唯一索引來產生資料庫查詢。properties-file 項目有優先順序。 | <p>確定輸入檔中的 unique-index 資訊是正確的。</p> <p>您可能要建立或修改內容檔中的相關項目，產生適當的資料庫查詢來解析主要鍵。</p> |
| 未解析的外來鍵 | All | 對於 KEYS 或 SUBKEYS 表格中未指定的外來鍵，「ID 解析程式」將不會解析表格中的外來鍵。 | <p>將參照的表格名稱新增至 SUBKEYS 表格中，然後再執行「ID 解析程式」。</p> <p>此外，確定資料庫綱目中已有外來鍵關係。</p> |
| | | 外來鍵是藉由使用內部別名或查詢資料庫來解析。只有當內部別名無法解析外來鍵時，才會執行資料庫查詢。 | <p>內部別名是使用主要鍵和 properties-file 項目所產生。</p> <p>請確定在輸入檔中作為來源和目標的 internal-alias 值是正確的。</p> |
| | | 使用 properties-file 項目資訊或給定表格的唯一索引來產生資料庫查詢。properties-file 項目有優先順序。 | <p>確定輸入檔中的 unique-index 資訊是正確的。</p> <p>您可能要建立或修改內容檔中的相關項目，產生適當的資料庫查詢來解析主要鍵。</p> |

第6章 載入資料

在您載入資料前，您必須執行以下步驟：

1. 產生要用於「載入器」的 DTD 和綱目（第一次載入資料時）

註：如果您正在載入商店記錄的資料，而且您以商店記錄所提供的 DTD 來建立 XML 檔，就不需要執行這個步驟。

2. 解析識別碼（必要時）

資料必須採用關聯於 DTD 的 XML 格式。如果要載入資料，請呼叫 Load 指令。

設定「載入器」

「載入器」套件可讓您執行以下項目來設定「載入器」的運作方式：

- 忽略輸入檔中的元素
- 在直欄中插入空值 (NULL)
- 載入時間戳記和日期資料
- 載入現行時間戳記
- 管理事件佇列
- 與不同的資料庫軟體和作業系統一起執行
- 替換元件
- 使用「產品顧問」搜尋空間同步化

您可以執行以下步驟來自訂這些「載入器」的功能：

1. 建立新的「載入器」自訂程式內容檔。



• `MassLoadCustomizer.properties` 位於 `MassLoader.zip` 保存檔中。請解開這個檔案，並重新命名（仍採用 `.properties` 副檔名），並將之置於類別路徑內的目錄中。**重要事項：**請勿移除或修改現有的 `MassLoadCustomizer.properties` 檔案。



• `ISeries_LODWCSDTA_Customizer.properties` 是位於 `/QIBM/ProdData/WebCommerce/properties` 目錄中。將此檔案複製至 `/instroot/xml` 目錄，然後為新檔案重新命名，但是保留 `.properties` 副檔名，接著對新檔案進行任何必要的變更。**重要事項：**請勿移除或修改原來的 `ISeries_LODWCSDTA_Customizer.properties` 檔案。

2. 修改新「載入器」自訂程式內容檔中指定的內容值。
3. 指定新檔案名稱作為 Load 指令的自訂程式參數值。

忽略輸入檔中的元素

如果輸入檔中包含的元素無法對映至目標資料庫，您可以將「載入器」設定為忽略「載入器」自訂程式內容檔中的那些元素。請利用 `IgnoreElements` 來指定要忽略的元素，以及利用分號來分開這些元素 (;)。如若要忽略匯入、文字和 `ProductRepository` 元素，請在「載入器」自訂程式內容檔中指定以下內容：

```
IgnoreElements = import;literals;ProductRepository
```

在直欄中插入空值 (NULL)

您可以在「載入器」自訂程式內容檔中將 `EnableNULLCheck` 內容設定為 "true"，而讓「載入器」可將空值 (NULL) 插入直欄中。比方說：

```
EnableNULLCheck = true
```

考慮到效能的因素，此功能預設停用。

請使用 `NULLStringLiteral` 內容來決定資料中空值的字串表示法。如若要設定「載入器」使用 "-" 字串來代表空值，請在「載入器」自訂程式內容檔中指定以下的內容和值：

```
NULLStringLiteral = -
```

在預設的情況下，此內容的值為 "NULL"（不含引號）。

載入時間戳記和日期資料

載入器可以將資料載入含時間戳記和日期資料類型的直欄中。文件中的時間戳記和日期資料的資料格式是由可自訂的型樣來決定。使用者可以編輯現有的型樣或新增其它型樣到現有的型樣清單中。

時間戳記或日期的資料是就可用的型樣（遮罩）來檢查的。符合資料的第一個型樣用以在將資料載入到資料庫之前，將資料轉換成目標時間戳記格式。

可自訂的輸出時間戳記型樣有兩個：`TimeStampFormat.JDBC` 和 `TimeStampFormat.Load`。

1. 當「載入器」使用 `JDBC` 連線來執行作業時，則使用 `TimeStampFormat.JDBC`。
「載入器」的 `SQL` 匯入和刪除方法是使用 `JDBC` 連線來更新資料庫。
2. 當「載入器」使用原生公用程式時，就會使用 `TimeStampFormat.Load`。
載入器的匯入與載入方法會使用原生的公用程式。

您可以在「載入器」自訂程式內容檔中修改或新增遮罩，以便自訂時間戳記格式。

所提供的輸入時間戳記遮罩如下：

```
InputTimeStampFormat.1 = yyyy-DD hh:mm:ss.SSSSSS
InputTimeStampFormat.2 = yyyy-MM-dd hh:mm:ss.SSSSSS
InputTimeStampFormat.3 = yyyy-DD-hh.mm.ss.SSSSSS
InputTimeStampFormat.4 = yyyy-MM-dd-HH.mm.ss.SSSSSS
InputTimeStampFormat.5 = yyyy-MM-dd-hh.mm.ss.SSSSSS
InputTimeStampFormat.6 = yyyy-MM-dd HH:mm:ss.SSSSSS
InputTimeStampFormat.7 = yyyy-DD HH:mm:ss.SSSSSS
```

以下是輸入日期格式的預設型樣：

```
InputDateFormat.1 = MM-dd-yyyy
InputDateFormat.2 = yyyy-dd-MM
InputDateFormat.3 = yyyy-MM-dd
InputDateFormat.4 = MM/dd/yyyy
InputDateFormat.5 = yyyy/dd/MM
InputDateFormat.6 = yyyy-DD
```

您可以修改這些時間戳記和日期遮罩，也可以您新增需要的任意數目的遮罩。請依照與輸入時間戳記比較時所要採用的數值順序，在「載入器」自訂程式內容檔中指定這些遮罩。如果您新增輸入時間戳記，就必須利用目前順序中的下個號碼。（比方說，如果您新增到上述清單中，下個輸入時間戳記遮罩就會是 `InputTimeStampFormat.8`。）

以下是將輸入資料格式化成為時間戳記和日期之輸出的型樣：

```
TimeStampFormat.JDBC = yyyy-MM-dd hh:mm:ss.SSSSSS
TimeStampFormat.Load = yyyy-MM-dd-hh.mm.ss.SSSSSS

DateFormat.JDBC = yyyy-MM-dd
DateFormat.Load = yyyy-MM-dd
```

一般而言，輸出日期和時間戳記格式並**不會**進行自訂。

載入現行時間戳記

載入器可以根據 `TOD` 計時鐘的讀取，將值插入含時間戳記資料類型的直欄中。比方說，`WebSphere Commerce` 中的優惠 `STARTDATE` 和 `ENDDATE` 可有以優惠插入表格中的時間為根據的值。為了支援這項功能，載入器套件會利用 `MLTIME` 表格來保留時間戳記案例。這份表格的綱目如下：

```
table MLTIME
(
  INSTANCEID BIGINT not null,
  MLTIMESTAMP TIMESTAMP
)
```

您可以變更「載入器」自訂程式內容檔中的以下內容，以便自訂表格的名稱和直欄：

```
TimestampTableName = MLTIME
TimestampIdColumn = INSTANCEID
TimestampValueColumn = MLTIMESTAMP
```

指定現行時間戳記值的輸入資料是以時間戳記字串型樣為基礎。下列遮罩是用來指定時間戳記的期限：

```
%D for days
%M for months
```

%Y for years
%H for hours
%m for minutes
%s for seconds

您可以在「載入器」自訂程式內容檔中修改或新增遮罩，以便自訂現行時間戳記格式。所提供的輸入遮罩如下：

```
InputCurrentTimestampFormat.1 = CURRENT TIMESTAMP  
InputCurrentTimestampFormat.2 = CURRENT TIMESTAMP %D DAYS  
InputCurrentTimestampFormat.3 = CURRENT TIMESTAMP %D DAYS %M MONTHS  
InputCurrentTimestampFormat.4 = CURRENT TIMESTAMP %D DAYS %M MONTHS %Y YEARS  
InputCurrentTimestampFormat.5 = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS  
InputCurrentTimestampFormat.6 = SYSDATE  
InputCurrentTimestampFormat.7 = ADDDAYS(SYSDATE,%D)  
InputCurrentTimestampFormat.8 = ADDDAYS(ADDMONTHS(SYSDATE,%M),%D)  
InputCurrentTimestampFormat.9 = ADDDAYS(ADDMONTHS(ADDEYEARS(SYSDATE,%Y),%M),%D)
```

現行時間戳記的輸入資料會比對指定的型樣。如果資料符合指定的型樣，則會利用這個型樣來剖析輸入資料，且載入器會將資料轉換成適當的輸出格式，然後再將資料插入資料庫中。如果下註標號碼依序排列的話，新型樣就可以新增到上述清單中。

總共有兩個目標輸出格式可用來指定現行時間戳記：

1. 當載入器是在載入或匯入模式下運作時，則會使用 `CurrentTimestampFormat.Load`。
2. 當載入器利用 `JDBC` 在資料庫中插入、更新或刪除值時，會使用 `CurrentTimestampFormat.JDBC`。

「載入器」中的預設目標型樣如下：

```
CurrentTimestampFormat.Load = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS  
%h HOURS %m MINUTES %s SECONDS  
CurrentTimestampFormat.JDBC = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS  
%h HOURS %m MINUTES %s SECONDS
```

您也可以在此「載入器」自訂程式內容檔中自訂這些型樣的內容。在您自訂 `CurrentTimestampFormat.Load` 和 `CurrentTimestampFormat.JDBC` 內容時，您應確定結果陳述式的語法在指定的資料庫管理系統中是有效的。

載入器會利用 `CurrentTimestampLiteral` 內容及早判斷時間戳記直欄的值是否採用現行的時間戳記格式，從而避免浪費時間在判斷該值是否代表時間的字串。

```
CurrentTimestampLiteral = CURRENT TIMESTAMP
```



此內容對於 DB2 的預設值為 `CURRENT TIMESTAMP`。



對於 Oracle 資料庫的預設值為 `SYSDATE`。

載入現行時間戳記的範例

「載入器」會收到以下的資訊以便更新 `OFFER_ID 10123` 的報價。起始日期的值為 `"CURRENT TIMESTAMP"`，而結束日期的值為 `"CURRENT TIMESTAMP + 14 DAYS"`。

```
<OFFER
  OFFER_ID="10123"
  STARTDATE="CURRENT TIMESTAMP">
  ENDDATE="CURRENT TIMESTAMP + 14 DAYS"
/>
```

載入器會認定 STARTDATE 和 ENDDATE 直欄為資料庫中的時間戳記資料類型。根據 CurrentTimeStampLiteral 內容而定，這些值確定會有以現行時間戳記格式所指定的值。STARTDATE 的值會符合 InputCurrentTimeStampFormat.1 型樣，且會轉換成 CurrentTimeStampFormat.JDBC 內容所指定的型樣。ENDDATE 的值會符合 InputCurrentTimeStampFormat.2 內容的格式，且會轉換成 CurrentTimeStampFormat.JDBC 內容所指定的型樣。

在現行時間戳記中加入期限的範例

您可以利用載入器來新增期限到現行時間戳記中。比方說，您可能會想載入優惠，但不要輸入特定日期。如果要做到這一點，您必須建立一個本身是起始日期後之某段期限的結束日期。以下是 DB2 所適用的範例：

```
<Offer
  Startdate="Current Timestamp"
  Enddate="Current Timestamp +14 Days +4 Months +1 Year +0 Hours
    +0 Minutes +0 Seconds"
/>
```

但若要處理現行時間戳記期限而不考慮平台的因素，您必須修改「載入器」自訂程式內容檔的遮罩，以便自訂現行時間戳記格式。以下是自訂現行時間戳記內容規格的範例：

```
CurrentTimestampLiteral=Current Timestamp

InputCurrentTimeStampFormat.0=Current Timestamp
InputCurrentTimeStampFormat.1=Current Timestamp %D Days
InputCurrentTimeStampFormat.2=Current Timestamp %M Months
InputCurrentTimeStampFormat.3=Current Timestamp %Y Years
InputCurrentTimeStampFormat.4=Current Timestamp %D Days %M Months
InputCurrentTimeStampFormat.5=Current Timestamp %D Days %M Months %Y Years
InputCurrentTimeStampFormat.5=Current Timestamp %H Hours %m Minutes %s Seconds

CurrentTimeStampFormat.JDBC=Current Timestamp %D Days %M Months %Y Years
  %H Hours %m Minutes %s Seconds
```

使用優惠範例和這些內容規格，優惠的結束日期符合 InputCurrentTimeStampFormat.5 型樣。這會利用 CurrentTimeStampFormat.JDBC 來產生下列優惠資訊。

```
<Offer
  Startdate="Current Timestamp"
  Enddate="Current Timestamp +14 Days +4 Months +1 Year +0 Hours +0 Minutes +0 Seconds"
/>
```

上述範例顯示載入器可如何輸入多個現行時間戳記格式，並依照適當的輸出格式來建立它們的格式。下列範例顯示您可以如何處理非特定平台專用的格式及將它們對映至特定平台專用的輸出格式。

```
<offer
  Startdate="Now"
  Enddate="Now +14D +4M +1Y"
/>
```

```
CurrentTimestampLiteral=Now
```

```
InputCurrentTimestampFormat.0=Now  
InputCurrentTimestampFormat.1=Now %DD  
InputCurrentTimestampFormat.2=Now %MM  
InputCurrentTimestampFormat.3=Now %YY  
InputCurrentTimestampFormat.4=Now %DD %MM  
InputCurrentTimestampFormat.5=Now %DD %MM %YY  
InputCurrentTimestampFormat.5=Sysdate %HH %mm %ss
```

```
CurrentTimestampFormat.JDBC=AddYears (AddMonths (AddDays (AddHours (AddMinutes (AddSeconds  
(Sysdate,%s),%m),%H),%D),%M),%Y)
```

註： 以上的陳述式僅作為範例。它只是用來說明針對虛擬資料庫管理系統的自訂功能。它並不適用於 DB2 或 Oracle 資料庫。

使用優惠範例和這些內容規格，優惠的結束日期會符合 `InputCurrentTimestampFormat.5` 型樣。這會利用 `CurrentTimestampFormat.JDBC` 來產生下列優惠資訊。

```
<Offer  
  Startdate="Current Timestamp"  
  Enddate="AddYears (AddMonths (AddDays (AddMinutes (AddSeconds (Sysdate,0),0),0),14),4),1)"  
>
```

管理事件佇列

您可以修改「載入器」自訂程式內容檔的設定，以便自訂「載入器」的事件佇列。比方說：

```
QueueLowCount = 35  
QueueHighCount = 90
```

當佇列中的元素數目到達上限時，會鎖定填入佇列的事件來源，以防止其它事件進入佇列中。當佇列中的元素數目低於下限值時，佇列會開始接受事件。

與不同的資料庫軟體和作業系統一起執行

您可以在「載入器」自訂程式內容檔中修改以下元素的參數，而指定不同的資料庫軟體和作業系統，以便自訂「載入器」來用於不同的資料庫軟體和作業系統：

- 資料庫連線指令
- 資料庫載入表格指令
- 資料庫匯入指令
- 呼叫載入的系統指令

如果要自訂這些項目，請在「載入器」自訂程式內容檔中，將指令前面的雙斜線備註字元 (`//`) 移除，再修改預設值。

資料庫連線指令： 如果您要修改預設值（假設您使用 DB2），您可以變更資料庫連線指令的參數。

```
DBConnectCommand = connect to {0} user {1} using {2};
```

其中：

- 0 = 資料庫名稱
- 1 = 資料庫使用者
- 2 = 使用者密碼

資料庫載入表格指令：如果您要變更預設值，您可以變更資料庫載入表格指令的參數。

```
DBLoadTableCommand = load from {0} of del modified by coldel{1}
charcodel{2} insert into {3} ({4});
```

其中：

- 0 = 檔案名稱
- 1 = 直欄定界符號
- 2 = 字元定界符號
- 3 = 表格名稱
- 4 = 直欄名稱，其間用逗號分開 (,)

資料庫匯入指令：如果您要變更預設值，您可以變更資料庫匯入指令的參數。

```
DBImportCommand = import from {0} of del modified by colcodel{1} charcodel{2}
insert_update into {3} ({4});
```

其中：

- 0 = 檔案名稱
- 1 = 直欄定界符號
- 2 = 字元定界符號
- 3 = 表格名稱
- 4 = 直欄名稱，其間用逗號分開 (,)

呼叫載入的系統指令：如果您要變更預設值，您可以變更呼叫載入的系統指令參數。這個指令會執行載入器所產生的原生載入及匯入 Script。

- 

```
DBLoadCommand = db2c1pex DB2 -z {0} -astvf {1}
```

其中：

0 = 日誌檔名稱
1 = 指令檔名稱

例如對於在 AIX 上執行的 DB2，DBLoadCommand 內容的值如下：

```
db2 -tvf {1} -z {0}
```



```
DBLoadCommand = sqlldr log={0} control={1} USERID={2}
```

其中：

0 = 日誌檔名稱
1 = 指令檔名稱
2 = 資料庫使用者名稱

您可使用下列各種資料庫與作業系統組合的設定：



針對在 Windows NT 或 Windows 2000 上執行，並且使用 sqlimport、load、import 或 delete 方法的 DB2，設定類別路徑系統環境變數並且加入 db2/dbconnect.zip。



請針對在 AIX、Solaris 或 Linux 環境中執行的 DB2 執行以下步驟：

- 若使用 sqlimport、load、import 或 delete 方法，請設定類別路徑系統環境變數並且加入 db2/dbconnect.zip。
- 若要使用 load 或 import 方法，請修改「載入器」自訂程式內容檔中的以下內容：

```
/**
 * 「連線」指令。(預設值是針對 DB2)
 * 參數 0 = 資料庫名稱
 * 參數 1 = 資料庫使用者
 * 參數 2 = 使用者密碼
 */
DBConnectCommand = connect to {0} user {1} using {2};

/**
 * 「將資料載入至表格」指令。(預設值是針對 DB2)
 * 參數 0 = 檔名
 * 參數 1 = 直欄定界符號
 * 參數 2 = 字元定界符號
 * 參數 3 = 表格名稱
 * 參數 4 = 直欄名稱，以逗點區隔
 */
DBLoadTableCommand = load from {0} of del modified by coldel{1}
insert into {3} ({4});

/**
 * 「在表格中插入資料」指令 (預設值是針對 DB2)
 * 參數 0 = 檔名
 * 參數 1 = 直欄定界符號
 * 參數 2 = 字元定界符號
```



```

    * 參數 3 = 表格名稱
    * 參數 4 = 直欄名稱，以逗點區隔
*/

DBUpdateTableCommand = import from {0} of del modified by coldel{1}
insert_update into {3} ({4});
/**
 * 會呼叫載入的系統指令（預設值是針對 DB2）
 * 參數 0 = 日誌檔名稱
 * 參數 1 = 指令檔名稱
*/

DBLoadCommand = db2 -z {0} -tf {1}

```

- 
 針對在 iSeries 執行並且使用 sqlimport、load、import 或 delete 方法的 DB2，修改「載入器」自訂程式內容檔的以下內容：

```

/**
 * 連接字串。
*/

ConnectionStringID = jdbc:db2://

/**
 * JDBC 驅動程式資訊。
*/

JDBCDriverName = com.ibm.db2.jdbc.app.DB2Driver
DbVendorName=DB2/iSeries

/**
 * 自訂載入/匯入方法的寫出器。
*/

WriterName=com.ibm.wca.MassLoader.Writer.ISeriesWriter

```

- 






針對在 Windows NT、Windows 2000、AIX、Solaris 或 Linux 環境中執行，而且使用 sqlimport、load、import 或 delete 方法的 Oracle 資料庫，設定 classpath system-environment 變數以併入 oracle/dbconnect.zip。

替換元件

您可以在「載入器」自訂程式的內容檔中，為以下的元素提供您要取代預設實作的類別值，以取代「載入器」元件：

ParserName

所要使用的剖析器名稱

ValidatorName

要使用的驗證器名稱

FormatterName

要使用的格式製作器名稱

JDBCFormatterName

當使用 SQL 匯入方法時所用的格式製作器名稱

WriterName

要使用的寫入器名稱

JDBCWriterName

當使用 SQL 匯入方法時所用的寫出器名稱

例如，若要用 `com.abc.writer.SpecialWriter` 寫入器來取代「載入器」的預設寫入器 (DefaultWriter)，請在「載入器」自訂程式內容檔中指定以下內容：

```
WriterName = com.abc.writer.SpecialWriter
```

「載入器」會利用 `"com.abc.writer.SpecialWriter"` 來執行寫入功能。

使用「產品顧問」搜尋空間同步化

若要使用「產品顧問」的搜尋空間同步化，請執行以下步驟：

1. 建立一個用以同步化的 XML 架構資訊檔，並命名為 `"PASyncInfo.xml"`。
2. 請在 `PASyncInfo.xml` 中指定 `PASync.xsd` 作為要使用的 XML 綱目。比方說：

```
<PASync
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='PASync.xsd'
```

系統會提供 `PASync.xsd` 檔案。以下的文字顯示了 `PASync.xsd` 的內容：

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="PASync">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="SearchScheme" />
        <xsd:element ref="Command" minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name="member" type="xsd:string" use="required" />
      <xsd:attribute name="store" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="SearchScheme">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="RelatedTable" />
        <xsd:element ref="Search" minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name="tableName" type="xsd:string" use="required" />
      <xsd:attribute name="primary" type="xsd:string" use="required" />
      <xsd:attribute name="colName" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="RelatedTable">
    <xsd:complexType>
      <xsd:attribute name="tableName" type="xsd:string" use="required" />
      <xsd:attribute name="from" type="xsd:string" use="required" />
      <xsd:attribute name="to" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Search">
    <xsd:complexType>
      <xsd:attribute name="value" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Command">
    <xsd:complexType>
```

```

        <xsd:attribute name = "tableName" type="xsd:string" use="required" />
        <xsd:attribute name = "idColumnName" type="xsd:string" use="required" />
        <xsd:attribute name = "addCommand" type="xsd:string" />
        <xsd:attribute name = "updateCommand" type="xsd:string" />
        <xsd:attribute name = "deleteCommand" type="xsd:string" />
    </xsd:complexType>
</xsd:element>

```

```
</xsd:schema>
```

3. 請在 PASyncInfo.xml 中指定完成同步化所需的成員 ID 和商店實體 ID。比方說：

```

member = "-2000"
store = "10351"

```

4. 在 PASyncInfo.xml 中的 search-scheme 元素下，指定構成搜尋空間的 CATGROUP 識別碼。比方說：

```

<SearchScheme
  tableName = "catgroup"
  primary = "CATGROUP_ID"
  colName = "identifier" >

  <RelatedTable
    tableName = "catgpenrel"
    from = "CATGROUP_ID"
    to = "CATENTRY_ID" />

  <Search value="Pants" />
  <Search value="Shirts" />

</SearchScheme>

```

範例中指定了 "Pants" 和 "Shirts"。您可以視需要指定多個 CATGROUP 識別碼。

5. 請在 PASyncInfo.xml 中指定用來決定排定指令的屬性。比方說：

```

<Command tableName = "CATENTRY" idColumnName = "CATENTRY_ID"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "RemoveProductsFromAllSearchSpaces"
/>

<Command tableName = "CATENTDESC" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "LISTPRICE" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "ATTRVALUE" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "CATENTATTR" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "CATGPENREL" idColumnName = "CATENTRY_ID"
  addCommand = "AddProductsToSearchSpace"

```

```
        deleteCommand = "RemoveProductsFromSearchSpace"  
    />
```

```
</PASync>
```

6. 請在新的「載入器」自訂程式內容檔中指定 XML 架構資訊檔。比方說：

```
PASyncDocumentURL = PASyncInfo.xml
```

7. 請在新的「載入器」自訂程式內容檔中啟用同步化。比方說：

```
PASyncEnabled = true
```

8. 在 Load 指令中使用 SQL 匯入或刪除方法。以下為「載入器」的正確 XML 輸入範例：

```
<store-asset>
```

```
    <catentry  
      CATENTRY_ID="10351"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000051"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000051"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10352"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000052"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000052"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10353"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000053"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000053"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10358"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000058"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000058"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10365"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000065"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000065"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10372"
```

```

MEMBER_ID="-2000"
PARTNUMBER="000072"
CATENTTYPE_ID="ProductBean"
MFPARTNUMBER="m000072"
MARKFORDELETE="0"
BUYABLE="1"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10372"
  CATALOG_ID="10351"
  SEQUENCE="3"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10365"
  CATALOG_ID="10351"
  SEQUENCE="4"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10358"
  CATALOG_ID="10351"
  SEQUENCE="5"
/>

<catgpenrel
  CATGROUP_ID="10355"
  CATENTRY_ID="10372"
  CATALOG_ID="10351"
  SEQUENCE="3"
/>

</store-asset>

```

註: 停用「產品顧問」的搜尋空間同步化功能，可改善「載入器」的效能；因此，請在需要時才使用此功能。

自訂「產品顧問」搜尋空間同步化

「載入器」套件可讓您修改「載入器」自訂程式內容檔，而自訂「產品顧問」的搜尋空間同步化功能，以便執行以下項目：

- **啟用或停用同步化**

您可以在「載入器」自訂程式內容檔的以下內容中，指定 `true` 或 `false` 值來啟用或停用同步化：

```
PASyncEnabled = true
```

- **指定同步化的架構資訊檔**

您可以在「載入器」自訂程式內容檔中設定以下內容的值，以便指定要用於同步化的 XML 架構資訊檔：

```
PASyncDocumentURL = PASyncInfo.xml
```

- **指定排程查詢長度**

您可以在「載入器」自訂程式內容檔中設定以下內容的值，以指定排程查詢長度：

```
PAScheduleQueryLength = 30
```

此內容的值應在 20 到 900 間。

- **指定排定開始時間**

您可以在「載入器」自訂程式內容檔中，為 `PAScheduledStartTime` 內容指定絕對時間戳記、現行時間戳記或具有期限的現行時間戳記，以指定排定開始時間。

註： 時間戳記的格式必須能適用於您的資料庫。

以下以 DB2 為例，其在載入後將會執行排定的工作 5 分鐘。

```
PAScheduledStartTime = CURRENT TIMESTAMP + 5 MINUTES
```

 以下為 Oracle 資料庫立即執行工作的範例：

```
PAScheduledStartTime = SYSDATE
```

決定在使用「載入器」時的資料處理方式

載入器會在 Load 指令中提供下列選項，讓您用以處理資料：


- 載入
- 匯入
- 使用 SQL 匯入特性

在載入資料之前，您應該判斷哪個處理方法會產生最好的結果。




選擇載入方法

請考慮以下狀況中的載入方法：

- 如果您知道資料沒有問題而且資料庫中沒有任何資料
- 如果您知道資料沒有問題而且資料庫中沒有任何正在載入的資料
- 如果您知道資料沒有問題、目標表格沒有任何主要鍵，而且資料庫中沒有正要載入的資料
- 如果載入時間為您的主要考量
- 如果資料庫為本端 DB2 資料庫

 若使用載入方法，則資料會載入至資料庫。如果資料已存在，則指令會失敗，並且會顯示重複鍵錯誤和重複錯誤訊息。

選擇匯入方法

     利用 DB2 的匯入方法，也會將資料載入資料庫中。如果資料已經存在，這時不會刪除它，但會用新值來更新它。在下列情況下，則可考慮使用這個方法：

- 如果資料庫管理系統是 DB2
- 如果您不知道資料有沒有問題

- 如果您必須在直欄層次更新許多組異質資料
- 如果載入時間不是主要考量
- 如果資料匯入其中的表格有主要鍵

400 若使用匯入方法，則資料也會載入至資料庫。如果資料已經存在，這時不會刪除它，但會用新值來更新它。在下列情況下，則可考慮使用這個方法：

- 如果您不知道資料有沒有問題
- 資料已在資料庫中
- 如果載入時間不是主要考量
- 如果資料匯入其中的表格有主要鍵

選擇 SQL 匯入方法

透過 SQL 匯入方法，可使用 JDBC 或 SQL 陳述式來更新或將資料插入資料庫中。如果資料還不存在，這時會插入資料；如果資料已經存在，這時會更新資料。在下列情況下，則可考慮使用這個方法：

- 如果您是要更新現有資料，而且需要進行直欄層次的更新
此方法可提供有關限制違規與資料類型錯誤的錯誤報告。
- 如果您知道某些資料有問題
- 如果資料完整性是主要考量
- 如果資料庫不在本端
- 如果您使用「產品顧問」搜尋空間同步化

其他注意事項

- **使用載入方法的限制**

載入方法無法在位元資料欄位中插入或更新資料。

DB2 若使用載入方法，則只會將新記錄插入資料庫；現有的記錄不會更新。

DB2 載入方法僅適用於本端（而非遠端）DB2 資料庫。

- **使用匯入方法的限制**

資料庫管理系統必須是 DB2，才能使用匯入方法。

匯入方法無法在位元資料欄位中插入或更新資料。

若使用匯入方法，則載入器只會插入或更新有定義主要鍵的表格；匯入方法無法插入或更新沒有主要鍵之表格中的資料。如果輸入記錄中只含主要直欄的值，則會拒絕該記錄。

- **SQL 匯入與載入方法的比較**

SQL 匯入方法會檢查資料是否一致（其中包括外來參照），且可讓您更新現有資料。載入方法則不會。

- **匯入與 SQL 匯入方法的比較**

匯入與 SQL 匯入方法所執行的功能類似。一般而言，匯入方法執行速度較快，但需要暫存檔的磁碟空間。

匯入方法只能插入或更新有定義主要鍵的表格；而 SQL 匯入方法則不會要求表格中必須有主要鍵。

- **以所用資料庫產品為基礎的方法比較**

匯入與載入方法採用經過最佳化（DB2 方面）的原生公用程式，而 SQL 匯入方法則是採用 JDBC 呼叫方式（許多資料庫產品都通用 JDBC 呼叫）。

- **進一步的注意事項**

刪除方法用以從資料庫中刪除輸入 XML 文件中的資料。元素必須含有表格的主要鍵或唯一索引值。如果要刪除的資料和啓用了「連帶刪除」之另一個表格中的資料相依，則相依資料也會同時刪除。

如果您使用「產品顧問」搜尋空間同步化，您必須使用 SQL 匯入方法來載入資料。

載入大型文件

在使用「載入器」套件公用程式將大型文件載入至資料庫時，請注意以下項目：

- **Java 虛擬機器 (JVM) 資料堆大小**

在預設的情況下，配置給 JVM 資料堆的記憶體數量上限是 64 MB。如果這沒有增加的話，在載入過程中，JVM 最後可能會將記憶體用完。您可以在 Java 指令中設定 JVM -mx 選項，來改變配置給 Java 資料堆的記憶體數量上限。

- **追蹤日誌記載**

當載入大型 XML 文件時，追蹤日誌程式可能會用完 JVM 資料堆。追蹤資訊主要是在執行失敗的情況下，用來進行執行的除錯。如果不需要追蹤載入程序的話，應該將追蹤關閉。當停用追蹤時，效能會顯著提升。修改日誌記載配置 XML 文件可關閉追蹤。

預設日誌配置檔為 WCALoggerConfig.xml。若要關閉追蹤日誌記載，請將「載入器」的追蹤日誌記載配置從：

```
<logger type="trace">
  <handler type="file">
    <filePath>MassLoadTrace.log</filePath>
    <filter type="Any">
      <messageType name="PUBLIC" />
    </filter>
  </handler>
</logger>
```

改為

```
<logger type="trace">
  <handler type="file">
    <filePath>MassLoadTrace.log</filePath>
  </handler>
</logger>
```

關於修改 WCALoggerConfig.xml 檔案的其他資訊，請參閱第68頁的『自訂「載入器」套件的日誌記載』。

- **確定數**

當在 SQL 匯入模式下操作時的載入器預設確定數是 1。因而在預設的情況下，每次更新或插入資料庫時，都會確定交易。如果要改進大型文件的載入器效能，您應該增加確定數。建議使用值 "100"；不過，可視伺服器中的實體記憶體量、DBMS 交易日誌大小等而增加。

「載入器」的確定數是利用 Load 指令的 `-commitcount count` 選項來變更的（其中 `count` 是確定交易之前所執行的陳述式數目）。

疑難排解要訣

如果資料的載入進度異常變慢，則可能是載入器之日誌程式的檔案處理程式架構不正確。這可能是起因於以下其中一個狀況：


- 呼叫載入器的使用者無權寫入目錄或更新日誌記載架構文件中指定的檔案。
- 找不到日誌記載配置文件中指定作為檔案位置的目錄。
- 在日誌記載架構文件中指定為檔案位置的磁碟機沒有足夠的空間。


在您更正了這些問題後，您可以需要修改日誌記載架構文件（預設值為 WCALoggerConfig.xml）來變更指定的檔案位置。關於檔案處理和 WCALoggerConfig.xml 檔案的其他資訊，請參閱第68頁的『自訂「載入器」套件的日誌記載』。

第7章 擷取資料

如果要利用「擷取程式」在資料庫中擷取拉出資料，您必須利用擷取過濾檔來指定要從資料庫中擷取的資料。您所使用的擷取過濾器取決於您要擷取的資料類型。

以下的範例是使用 MemberSubsystemFilter.xml 作為擷取過濾器，而從資料庫中擷取「成員」子系統的資料。

- 

```
java com.ibm.wca.MassExtract.Extract -filter MemberSubsystemFilter.xml
-outfile MemberSubsystemExtracted.xml -dbname mall -dbuser myname
-dbpwd mypassword -customizer MemberSubsystemCustomizer
```
- 

```
QWEBCOMM/EXTWCSDTA FILTER(MemberSubsystemFilter.xml)
OUTFILE(MemberSubsystemExtracted.xml) DATABASE(database_name)
SCHEMA(mall) INSTRROOT(/QIBM/UserData/WebCommerce/instances/msr)
PASSWD(mypassword)
```

建立擷取過濾器

以下的擷取過濾器範例是從 CATGROUP、CATGRPDESC、CATGRPREL、CATENTRY、CATENTSHIP、OFFER、CATENTREL、CATGPENREL、CATENTDESC 和 ATTRVALUE 表格中擷取種類和產品資訊：

```
<sqlx>
<!-- ***** -->
<!-- 擷取種類資訊 -->
<!-- ***** -->

<functionDef id="Category" description="擷取種類" schemaentity="catgroup">
  <paramDef name=":lastRecord" type="string" value="10301"

  description="載入新資料前的最新記錄" />
  <body>
    select * from catgroup where catgroup_id > :lastRecord
  </body>
</functionDef>

<execute id="Category" description="擷取種類" schemaentity="catgroup">
  <param name=":lastRecord" type="string" value="10300"

  description="載入新資料前的最新記錄" />
</execute>

<functionDef id="Category Description" description="擷取語言環境的種類說明"

schemaentity="catgrpdesc">
  <paramDef name=":lastRecord" type="string" value="10300"

  description="載入新資料前的最新記錄" />
  <body>
    select * from catgrpdesc where catgroup_id > :lastRecord
  </body>
```

```

</functionDef>

<execute id="Category Description" description="擷取語言環境的種類說明"
schemaentity="catgrpdesc">
  <param name=":lastRecord" type="string" value="10300"

  description="載入新資料前的最新記錄" />
</execute>

<functionDef id="Category Relationship" description="擷取語言環境的種類關係"

schemaentity="catgrprel">
  <paramDef name=":lastRecord" type="string" value="10300"

  description="載入新資料前的最新記錄" />
  <body>
    select * from catgrprel where catgroup_id_child > :lastRecord
  </body>
</functionDef>

<execute id="Category Relationship" description="擷取語言環境的種類關係"

schemaentity="catgrprel">
  <param name=":lastRecord" type="string" value="10300"

  description="載入新資料前的最新記錄" />
</execute>

<!-- ***** -->
<!-- 擷取產品資訊 -->
<!-- ***** -->

<functionDef id="Product" description="擷取產品"
schemaentity="catentry">
  <paramDef name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
  <body>
    select * from catentry where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Product" description="擷取產品"
schemaentity="catentry">
  <param name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
</execute>

<functionDef id="Product Relationship" description="擷取產品出貨資訊"
schemaentity="catentrel">
  <paramDef name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
  <body>
    select * from catentrel where catentry_id_child > :lastrecord
  </body>
</functionDef>

<execute id="Product Relationship" description="擷取產品出貨資訊"

schemaentity="catentrel">
  <param name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
</execute>

<functionDef id="Product Description" description="擷取產品說明"

schemaentity="catentdesc">

```

```

        <paramDef name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
        <body>
            select * from catentdesc where catentry_id > :lastrecord
        </body>
    </functionDef>

    <execute id="Product Description" description="擷取產品說明"

        schemaentity="catentdesc">
        <param name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
        </execute>

    <functionDef id="Product Ship" description="擷取產品出貨資訊"

        schemaentity="catentship">
        <paramDef name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
        <body>
            select * from catentship where catentry_id > :lastrecord
        </body>
    </functionDef>

    <execute id="Product Ship" description="擷取產品出貨資訊"

        schemaentity="catentship">
        <param name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
        </execute>

    <functionDef id="Category Product Relationship" description="擷取種類對產品關係"
        schemaentity="catgpenrel">
        <paramDef name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
        <body>
            select * from catgpenrel where catgroup_id > :lastrecord
        </body>
    </functionDef>

    <execute id="Category Product Relationship" description="擷取種類對產品關係關係"
        schemaentity="catgpenrel">
        <param name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
        </execute>

    <!-- ***** -->
    <!-- 擷取產品屬性資訊 -->
    <!-- ***** -->

    <functionDef id="Product Attribute Values" description="擷取語言環境的產品屬性值"
        schemaentity="attrvalue">
        <paramDef name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
        <body>
            select * from attrvalue where catentry_id > :lastrecord
        </body>
    </functionDef>

    <execute id="Product Attribute Values" description="擷取語言環境的產品屬性值"
        schemaentity="attrvalue">
        <param name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
        </execute>

    <!-- ***** -->
    <!-- 擷取產品價格資訊 -->

```

```
<!-- ***** -->

<functionDef id="Offer" description="擷取報價" schemaentity="offer">
  <paramDef name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
  <body>
    select * from offer where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Offer" description="擷取報價" schemaentity="offer">
  <param name=":lastrecord" type="string" value="10300"
description="載入新資料前的最新記錄" />
</execute>

</sqlx>
```


設定擷取程式

您可以執行以下步驟來變更「擷取程式」使用的資料庫驅動程式：

1. 建立新的「擷取程式」自訂程式內容檔。



DB2ConnectionCustomizer.properties 是保存在 MassExtract.zip 檔中。請解開這個檔案，並重新命名（仍採用 .properties 副檔名），並將之置於類別路徑內的目錄中。**重要事項：** 請勿移除或修改現有的 DB2ConnectionCustomizer.properties 檔案。



iSeries_EXTWCSDTA_Customizer.properties 是位於 /QIBM/ProdData/WebCommerce/properties 目錄中。將此檔案複製至 /instroot/xml 目錄，然後為新檔案重新命名，但是保留 .properties 副檔名，接著對新檔案進行任何必要的變更。**重要事項：** 請勿移除或修改原來的 iSeries_EXTWCSDTA_Customizer.properties 檔案。

2. 修改新檔案中的資料庫驅動程式值。
3. 指定新檔案名稱作為 Extract 指令的自訂程式參數值。

以下引用了「擷取程式」自訂程式內容檔：

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

其中：

- DBVendorName 是用來選取資料庫類型。
選項如下：
 - DB2 Universal Database for iSeries (DB2/iSeries)
 - 其他作業系統上的 DB2 (DB2)
 - Oracle 資料庫 (Oracle)
- DBDriverName 是用來選取 JDBC 驅動程式。
選項如下：
 - DB2 Universal Database for iSeries (com.ibm.db2.jdbc.app.DB2Driver)
 - 其他作業系統上的 DB2 (COM.ibm.db2.jdbc.app.DB2Driver)
 - Oracle 資料庫 (oracle.jdbc.driver.OracleDriver)
- DBURL 是用來指定存資料庫時要使用的 URL。
選項如下：
 - DB2 Universal Database for iSeries (jdbc:db2://)
 - 其他作業系統上的 DB2 (jdbc:db2:)
 - Oracle 資料庫 (jdbc:oracle:oci8:@)

第8章 使用「載入器」套件日誌程式

載入器套件中的每個公用程式都會建立訊息，以指出順利完成、失敗與錯誤，並提供程式追蹤資訊。

載入器套件中的公用程式會參照 `WCALoggerConfig.xml` 檔。

- 

此檔案存在於類別路徑系統環境變數中指定的目錄中。您也可以使用 `com.ibm.wca.logging.configFile` Java 系統內容來指定。

- 

這個檔案是在 `/instroot/xml` 目錄中。

`WCALoggerConfig.xml` 決定每個公用程式所要提供的日誌記載資訊，以及資訊的顯示或儲存位置。您可以自訂這個檔案，並指定所要建立的日誌類型，以及所要記載的訊息類型。

架構您的環境在 Windows NT、Windows 2000、AIX、Linux 和 Solaris 系統中的日誌記載

若要設定環境中的日誌記載，您必須設定包含 `WCALoggerConfig.xml` 檔案的類別路徑系統環境變數，或指定 `com.ibm.wca.logging.configFile` 系統內容。

設定類別路徑變數的範例說明

假設 `WCALoggerConfig.xml` 檔位於 Windows NT 機器的 `d:\WebSphere\CommerceServer\xml\loader` 目錄中，您可使用下列陳述式來設定類別路徑變數：

```
SET CLASSPATH=%CLASSPATH%;D:\WebSphere\CommerceServer\xml\loader
```

指定 `com.ibm.wca.logging.configFile` 系統內容的範例說明

如果要指定 `com.ibm.wca.logging.configFile` 系統內容，請在呼叫 Java 直譯器時，使用 `-D` 選項。範例如下：

```
java -Dcom.ibm.wca.logging.configFile=D:\ice_tea\src\classlib\logger\xml\WC.xml  
com.ibm.wca.DTDGenerator.GeneratedDTD
```

自訂「載入器」套件的日誌記載

若要自訂「載入器」套件的日誌記載，請使用 WCALoggerConfig.xml 檔案。

- ▶ NT ▶ 2000 ▶ AIX ▶ Solaris ▶ Linux

此檔案存在於類別路徑系統環境變數中指定的目錄中。您也可以使用 com.ibm.wca.logging.configFile Java 系統內容來指定。

- ▶ 400

這個檔案是在 `/instroot/xml` 目錄中。

WCALoggerConfig.xml 包含了 `<component name="DTDGenerator">` 等一或多個元件標籤。在每個這些標示中，您都可以加入日誌程式和處理程式。您不應改變系統所提供的公用程式和日誌程式標籤；不過，您可以在日誌程式中加入處理程式標籤。有關您可在該檔案中加入什麼內容的詳細資訊，請參閱 WCALOGGER.DTD 檔。

「載入器」套件日誌位於下列目錄中的 messages.txt 檔內：

- ▶ NT `drive:\WebSphere\CommerceServer\instances\instance_name\logs`
- ▶ 2000 `drive:\Program Files\WebSphere\CommerceServer\instances\instance_name\logs`
- ▶ AIX `/usr/WebSphere/CommerceServer/instances/instance_name/logs`
- ▶ Solaris ▶ Linux `/opt/WebSphere/CommerceServer/instances/instance_name/logs`
- ▶ 400 `/QIBM/UserData/WebCommerce/instances/instance_name/logs`

處理程式

若要將處理程式新增至日誌程式，請指定 WCALoggerConfig.xml 檔案中的處理程式類型。您可以新增多個處理程式到日誌程式中。請注意，每個處理程式都有它本身的屬性和次層標籤，它們不一定會套用到其它處理程式上。處理程式類型包括：

| 處理程式類型 | 說明和屬性 |
|--|---|
| console | 將訊息傳送到標準輸出，通常是指令行 |
| file | 將訊息儲存在文字檔中 您必須在此處理程式中加入 "<filePath>log_path</filePath>"，作為次層標籤。 |
| multifile | 建立檔案的循環式日誌 您必須指定 "<filePath>log_path</filePath>"。現在會建立 1 至 <i>n</i> 的日誌檔。您可以新增下列屬性： MaxFiles 這是指出在使用多少日誌檔之後要清除第一個日誌檔的整數。 MaxKBFileSize 這是指出每個日誌檔最多儲存多少 KB 的整數。 |
|      資料庫 | 將 DB2 表格中的訊息儲存到循環式日誌中。您可以新增下列屬性： brand 這是資料庫品牌名稱。DB2 是目前唯一支援的資料庫。 maxRows 這是在清除最舊的項目之前，可儲存在表格中的記錄數目上限 您可以併入 "<jdbc/>" 作為次層標籤，再併入下列屬性： url 在 JDBC 中用來存取資料庫的 URL（例如 "Jdbc:db2:"wcm"；其中 "wcm" 是指資料庫的名稱）。資料庫必須先存在，您才能執行公用程式。 table 這是要記錄訊息的資料庫表格名稱。您必須使用以下的 DB2 陳述式來建立它： "CREATE TABLE"tablename (KEY char(13) FOR BIT DATA NOT NULL, COMPONENTNAME VARCHAR(30), ENTRY VARCHAR(2000), PRIMARY KEY(key))" userid 資料庫使用者名稱。您必須指定更新表格的許可權給這個使用者。以下的 DB2 陳述式將執行此動作： "GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE"tablename TO USER userid" password 指定的使用者名稱之資料庫密碼 |

以下的範例是將 "database" 類型的處理程式新增至日誌程式：

```
<handler type="database" brand="DB2" maxRows="50">
  <jdbc url="jdbc:db2:wcm"
    table="wcm.log"
    userid="wasuser"
    password="123456"/>
  <filter type="Any">
    <messageType name="FATAL"/>
  </filter>
</handler>
```

```

        <messageType name="ERROR"/>
        <messageType name="WARNING"/>
    </filter>
</handler>

```

過濾器

您可以在處理程式中新增或移除過濾器，以併入和排除訊息類型。如果日誌程式沒有過濾器，就不會記載任何日誌。每個過濾器標籤都會有列出訊息類型的次層 `messageType` 標籤，它通常是下列中的一項：

- INFO
- ERROR
- FATAL
- WARNING

WCALogger.dtd 檔案會列出其他的訊息類型，但是大部份都不用於「載入器」套件。

過濾器的類型如下：

| 過濾器類型 | 說明和屬性 |
|---------|---|
| Any | 在日誌檔中併入旗號標示為指定的 <code>messageType</code> 類型之一的任何訊息。 比方說，如果 <code>messageType</code> 清單包括 <code>ERROR</code> ，且應用程式產生 <code>ERROR</code> 類型訊息，這時會將訊息記錄下來。 |
| All | 訊息必須有所有指定 <code>messageType</code> 類型的屬性，才能將它併入日誌中。 |
| Exclude | 記錄 <code>messageType</code> 標籤清單中所未指定的所有訊息。 |

下列新增過濾器到處理程式的範例，可讓您記錄 `FATAL` 和 `ERROR` 訊息類型，並且忽略其它訊息：

```

<handler type="file">
  <filter type="Any">
    <messageType name="FATAL"/>
    <messageType name="ERROR"/>
  </filter>
</handler>

```

格式

您可以指定下列其中一種格式製作器類型，來製作訊息格式：

| 格式製作器類型 | 說明和屬性 |
|------------|--|
| safe (預設值) | 防止在內容檔中找不到訊息時設定異常狀況。 這個格式製作器會建立一則訊息來指出資源已遺漏。 |
| xml | 以 XML 格式來製作訊息格式。 如果找不到訊息，這個格式製作器也會寫入訊息，而不會設定異常狀況。 |

範例：WCALoggerConfig.xml 和 WCALogger.dtd

WCALoggerConfig.xml

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE WCALoggerConfig SYSTEM "WCALogger.dtd">
<WCALoggerConfig>
  <component name="MassLoader">
    <logger type="message">
      <handler type="file">
        <filePath>c:\temp\out.txt</filePath>
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
          <messageType name="INFO"/>
        </filter>
      </handler>
    </logger>
    <logger type="trace">
      <handler type="file">
        <filePath>out2.txt</filePath>
        <filter type="Any">
          <messageType name="PUBLIC"/>
        </filter>
      </handler>
    </logger>
    <logger type="typedMessage">
      <handler type="file">
        <filePath>tout.txt</filePath>
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
          <messageType name="INFO"/>
        </filter>
      </handler>
    </logger>
    <logger type="progress">
      <handler type="console" format="safe">
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
          <messageType name="INFO"/>
        </filter>
      </handler>
    </logger>
  </component>
  <component name="DTDGenerator">
    <logger type="message">
      <handler type="console">
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
          <messageType name="INFO"/>
        </filter>
      </handler>
    </logger>
    <logger type="trace">
      <handler type="console">
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
        </filter>
      </handler>
    </logger>
  </component>
</WCALoggerConfig>
```

```

        <messageType name="INFO"/>
    </filter>
</handler>
</logger>
</component>
</WCALoggerConfig>

```

WCALogger.dtd

```

<!-- 此 DTD 說明如何建構 WCALoggerConfig XML。
WCALoggerConfig XML 文件為 WCALoggerFactory 類別的輸入架構檔。
-->

<!ELEMENT WCALoggerConfig (component)+>

<!ELEMENT component (logger)+>
<!ATTLIST component name CDATA #REQUIRED>
<!ELEMENT logger (handler+,messageFile?)>
<!ATTLIST logger type (message | trace | typedMessage | progress) "typedMessage">

<!-- messageFile 為選用的預設內容檔，可用來建立特定語言環境的訊息
-->
<!ELEMENT messageFile (#PCDATA)>
<!ELEMENT handler (filePath?, filter, jdbc?)>
<!ATTLIST handler
type ( file|multiFile|console|error|textArea|database|ejbQueue|queue ) "console">

<!-- maxFiles & maxKBFileSize 僅可適用於處理程式的 multiFile 類型
-->
<!-- filePath & 編碼制只有在處理程式為 file 或 multiFile 類型時才適用
-->
<!ATTLIST handler maxFiles CDATA #IMPLIED>
<!ATTLIST handler maxKBFileSize CDATA #IMPLIED>
<!ATTLIST handler encoding CDATA #IMPLIED>
<!ATTLIST handler format (safe | xml) "safe">
<!-- maxRecords & 卡別僅適用於資料庫處理程式類型
-->
<!ATTLIST handler maxRecords CDATA #IMPLIED>
<!ATTLIST handler brand (DB2) #IMPLIED>
<!-- 資料庫處理程式類別標籤中必須有 jdbc 標籤
-->
<!ELEMENT jdbc EMPTY>
<!ATTLIST jdbc url CDATA #IMPLIED>
<!ATTLIST jdbc table CDATA #IMPLIED>
<!ATTLIST jdbc userid CDATA #IMPLIED>
<!ATTLIST jdbc password CDATA #IMPLIED>

<!ELEMENT filter (messageType+)>
<!ATTLIST filter type (Any | All | Exclude ) "Any">

<!-- messageType 屬性名稱為這些 JLog IRecordType
常數之一
-->
<!ELEMENT messageType EMPTY>
<!ATTLIST messageType name ( NONE | ALL | INFO |
INFORMATION | WARN | WARNING | ERR | ERROR |
FATAL | DEFAULT MESSAGE | API | CALLBACK |
ENTRY_EXIT | ENTRY | EXIT | ERROR_EXC |
MISC_DATA | OBJ_CREATE | OBJ_DELETE |
PRIVATE | PUBLIC | STATIC | SVC | PERF |
LEVEL1 | LEVEL2 | LEVEL3 ) "ALL">
<!ELEMENT filePath (#PCDATA)>

```


第9章 使用「載入器」套件錯誤報告程式

「載入器」和「ID 解析程式」包含一個錯誤報告程式，可在發生錯誤時產生異常狀態的文件報告。

在預設的情況下，異常狀況的文件會寫入到下列目錄中：

-  輸入文件常駐的目錄
-  `/instroot/logs`

如果要指定寫入異常狀況文件的目錄，請使用 Java 內容 `com.ibm.wcm.ErrorReporterDir`。在 Windows NT 環境中，「載入器」的範例應該以下列項目作為開頭：

```
java -Dcom.ibm.wcm.ErrorReporterDir=d:\massloadererrors
com.ibm.wca.MassLoader.MassLoad -dbname . . .
```

附註：使用者應該要有可以寫入指定目錄的許可權。

以下錯誤產生程式的範例 DTD (`store-all-error.dtd`)：

```
<!ENTITY % TABLE "calrule | catentry">
<!ELEMENT store-asset (error, (%TABLE;)*)>
<!ELEMENT message (#PCDATA) >
<!ELEMENT error ( message ) >
<!ATTLIST error
    locus          CDATA      #REQUIRED
    id             CDATA      #REQUIRED
>
<!ELEMENT calrule (error)>
<!ATTLIST calrule
    identifier      CDATA      #REQUIRED
    calrule_id     CDATA      #REQUIRED
    calcode_id     CDATA      #REQUIRED
    startdate      CDATA      #IMPLIED
    taxcgry_id     CDATA      #IMPLIED
    enddate        CDATA      #IMPLIED
    sequence       CDATA      #REQUIRED
    combination    CDATA      #REQUIRED
    calmethod_id   CDATA      #REQUIRED
    calmethod_id_qfy CDATA      #REQUIRED
    flags          CDATA      #REQUIRED
    field1         CDATA      #IMPLIED
    field2         CDATA      #IMPLIED
>
<!ELEMENT catentry (error)>
<!ATTLIST catentry
    catentry_id    CDATA      #REQUIRED
    member_id     CDATA      #REQUIRED
    catenttype_id  CDATA      #REQUIRED
    partnumber    CDATA      #IMPLIED
    mfpartment    CDATA      #IMPLIED
    mfname        CDATA      #IMPLIED
    markfordelete CDATA      #REQUIRED
    url           CDATA      #IMPLIED
    field1        CDATA      #IMPLIED
    field2        CDATA      #IMPLIED
    lastupdate    CDATA      #IMPLIED
    field3        CDATA      #IMPLIED
```

```

onspecial      CDATA      #IMPLIED
onauction      CDATA      #IMPLIED
field4         CDATA      #IMPLIED
field5         CDATA      #IMPLIED
buyable        CDATA      #IMPLIED
>

```

以下為「載入器」中的錯誤報告文件範例：

```

<?xml version="1.0"?>
<!DOCTYPE store-asset SYSTEM "store-all-error.dtd">
<store-asset>
  <error
    locus="Parser"
    id="SAXParseFatalError" >
    <message>
      錯誤字串 "--" 在備註中是不允許的。: 155 : 18
    </message>
  </error>
  <calrule
    calcode_id="30"
    enddate="2100-01 10:20:30.000000"
    calmethod_id="-47"
    identifier="7"
    taxcgry_id="9"
    calmethod_id_qfy="-46"
    startdate="1900-01-01-00.00.00.000000"
    flags="1"
    combination="2"
    calrule_id="44"
    sequence="9.0E+1">
    <error
      locus="Writer"
      id="SQLException" >
      <message>
        收到 SQL 異常狀況 [IBM][CLI Driver][DB2/NT] SQL0530N
        外來鍵
        "JANTONY.CALRULE.F_CALRULE4" 的 insert 或 update 值
        不等於上層表格之親代鍵的任何值。SQLSTATE=23503
      </message>
    </error>
  </calrule>
  <catentry
    catentry_id="10118"
    member_id="-2001"
    partnumber="1254"
    mfpartnumber="sku-163"
    mfname="InFashion"
    markfordelete="0"
    buyable="1"
    field1="abc" >
    <error
      locus="Formatter"
      id="FormattingError" >
      <message>
        在格式化 CATENTRY.FIELD1 的值時發生錯誤：abc 有錯誤
        [class java.lang.NumberFormatException(abc)].
      </message>
    </error>
  </catentry>
</store-asset>

```

第10章 架構載入器套件指令和 Script

如果要啟動載入器套件及執行其指令，請利用 WebSphere Commerce 目錄中所提供的 Script 或指令：

-  `drive:\WebSphere\CommerceServer\bin`
-  `drive:\Program Files\WebSphere\CommerceServer\bin`
-  `/usr/WebSphere/CommerceServer/bin`
-   `/opt/WebSphere/CommerceServer/bin`
-  QWEBCOMM 原生程式庫

這些 Script 與指令如下：



GENWCSDTD

DTD Generate 指令

RESWC SID

ID Resolve 指令

EXTWCSDTA

Extract 指令

LODWCSDTA

Load 指令

TRNWCSTXT

Text Transform 指令

TRNWCXML

XML Transform 指令

dttdgen.cmd

DTD Generate 指令

idresgen.cmd

ID Resolve 指令

massextract.cmd

Extract 指令

massload.cmd

Load 指令

txttransform.cmd

Text Transform 指令

xmltransform.cmd

XML Transform 指令

**dtdgen.sh**

DTD 產生 Shell Script

idresgen.sh

ID 解析 Shell Script

massextract.sh

擷取 Shell Script

massload.sh

載入 Shell Script

txttransform.sh

「文字轉換」Shell Script

xmltransform.sh

XML 轉換 Shell Script

第3篇 使用 Web 編輯器

本篇說明管理和使用 Catalog Manager Web 編輯器的方法。







Web 編輯器可讓您透過 Web 瀏覽器建立、刪除和變更型錄資料。用以檢視和更新資訊的資料輸入套表，集中置於 Web 編輯器中。在最簡單的情況下，套表會對應至 WebSphere Commerce 資料庫的表格。管理者可以選擇使用所提供的預設套表，或自訂可用的套表。

註： Web 編輯器使用 Internet Explorer 5（含）以上版本。

第11章 設置 Web 編輯器







管理者可將 Web 編輯器架構成提供資料給 WebSphere Commerce，其中包括擴充和自訂 WebSphere Commerce 綱目。Web 編輯器的設計，並不是要作為專屬於特定 WebSphere Commerce 案例的 Web 輸入套表組。Web 編輯器採用彈性且能夠自訂的設計，因而能夠支援不同組織的個別需求和職務。

資料庫檢視畫面是針對資料庫中一或多個表格所儲存的查詢。在安裝 Catalog Manager 期間，會將一個用以建立 WebSphere Commerce 產品的邏輯檢視畫面的範例檔置於下列目錄之各系統特定的子目錄（db2、oracle 與 os400）中：

-  *drive:\WebSphere\CommerceServer\schema*
-  *drive:\Program Files\WebSphere\CommerceServer\schema*
-  */usr/WebSphere/CommerceServer/schema*
-   */opt/WebSphere/CommerceServer/schema*
-  */QIBM/ProdData/WebCommerce/schema*

此 *wcs.view.sql* 範例檔含有一個產品檢視畫面，其中從多個表格合併出產品的相關資訊。它包含了產品檢視畫面的 SQL 資料定義。管理者可以研讀此檔案來規劃資料庫檢視畫面的開發。

預設 XML 套表說明檔 (*forms51_be.xml*) 中的套表設計，是要能在 WebSphere Commerce 資料庫中輕鬆地新增、編輯和刪除資料。此檔案的副本位於下列目錄中：

-  *drive:\WebSphere\CommerceServer\xml\wcwebeditor\xml*
-  *drive:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml*
-  */usr/WebSphere/CommerceServer/xml/wcwebeditor/xml*
-   */opt/WebSphere/CommerceServer/xml/wcwebeditor/xml*
-  */instroot/xml/wcwebeditor/xml*

此架構檔可直接沿用，或可供設置 Web 編輯器的管理者加以變更和加強。若要自訂 XML 套表說明檔，請參閱本章節後面所附的說明。


註： DTD 產生器可自動建立 Web 編輯器要使用的套表。

架構 Web 編輯器

本節提供了如何架構 Web 編輯器的資訊。雖然安裝程序在一開始就會處理此架構，管理者仍可透過此資訊，將 Web 編輯器重新架構成使用其他資料庫等。

編輯 `webeditor.properties` 檔案

Web 編輯器有部份應用程式參數是設定在 `webeditor.properties` 檔案中。此檔案位於以下的目錄中：

-  `drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
-  `drive:\Program Files\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
-  `/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wcwebeditor.war/WEB-INF/classes/webeditor.properties`
-   `/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wcwebeditor.war/WEB-INF/classes/webeditor.properties`
-  `/QIBM/UserData/WEBASADV4/was_instance_name/installedApps/WC_Enterprise_App_wcs_instance_name.ear/wcwebeditor.war/WEN-INF/classes`

管理者可藉由編輯 `webeditor.properties` 檔，來變更要使用哪個檔案來說明出現在 Web 編輯器中的套表。以下為 `webeditor.properties` 檔案內容的範例：

```
# WebEditor 的內容檔
(以下說明自訂程序清單的位置，以及 Catalog Manager 公用程式架構封套的位置。)
# 程序和 WCM 子系統架構封套的 URI 位置
ProcessConfigFile=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xml/weProcessList.xml

(以下指定了套表的定義位置。)
# 套表檔案的位置
FormsURL=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xml/forms51_be.xml

(以下的設定指定了 XML 對 HTML 樣式表的位置)
# XML 對 HTML 樣式表的位置
StyleSheetURI=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xml/webeditor.xml

(以下的設定指定了暫存檔的位置)
# 暫時目錄的位置
#temp.dir=

# WebSphere 資料來源。用來建置套表
# 下拉清單與欄位預設值。不過當公佈與搜尋時，Web 編輯器會利用 WCM
# 子系統 ID 解析程式、大量載入器與大量擷取程式。這些 WCM 子系統的
# 資料庫存取必須分別架構。
# WAS 資料庫來源的名稱
dbsource=jdbc/WebSphere Commerce DB2 DataSource demo

# WAS 資料庫的名稱。若有指定，則會改寫套表 XML 檔案中的資料庫名稱值。
# 當呼叫「大量載入器」等 WCM 子系統時，此值將作為參數。
dbname=mall

(下列設定指出所要使用的字集。依預設，會設為單位元組字集。
若為其他國家，請從被標成註解行之值中選取。)
# 編碼字集
```

```
Encoding=ISO-8859-1
```

```
# CN
#Encoding=gb2312
# TW
#Encoding=Big5
# KR
#Encoding=EUC-KR
# JP
#Encoding=Shift_JIS
```

(下列設定用以容許預覽影像。應在此指定伺服器的主電腦名稱，外加儲存在 WebSphere Commerce 中之影像資訊前的任何前導目錄資訊。WebSphere Commerce 型錄資訊則會附加於此值後面，以建構一個影像 URL。)

```
# 指定影像的基本 href 位置
# 應設定此項，以便將此資訊和儲存在 WCS 中的資訊
# (例如：/image/char.gif) 合併構成影像的 URL。
```

```
#imageRootURL=http://%HOSTDOMAIN
imageRootURL=http://localhost/webeditor
```

(下列設定用以設定應用程式的日期格式。如果這些內容無法使用 (亦即，被標成註解行)，則將使用 Java 語言環境的特定格式。以下是設定這些值的參照表。)

```
# 如果 Java 語言環境的特定格式不適用，請使用以下內容來指定日期格式
```

```
#dateFormat=yyyy-MM-dd
#dateTimeFormat=yyyy-MM-dd HH:mm:ss
```

| # 符號 | 意義 | 表示方式 | 範例 |
|---------|-----------|-------------|-----------------------|
| # ----- | ----- | ----- | ----- |
| # G | 紀元指定元 | (文字) | AD |
| # y | 年 | (數字) | 1996 |
| # M | 月 (年) | (文字 & 數字) | July & 07 |
| # d | 日 (月) | (數字) | 10 |
| # h | 時 (am/pm) | (1_12) (數字) | 12 |
| # H | 時 (日) | (0_23) (數字) | 0 |
| # m | 分 (時) | (數字) | 30 |
| # s | 秒 (分) | (數字) | 55 |
| # S | 毫秒 | (數字) | 978 |
| # E | 日 (週) | (文字) | Tuesday |
| # D | 日 (年) | (數字) | 189 |
| # F | 日之週 (月) | (數字) | 2 (七月的第二個星期三) |
| # w | 週 (年) | (數字) | 27 |
| # W | 週 (月) | (數字) | 2 |
| # a | am/pm 記號 | (文字) | PM |
| # k | 時 (日) | (1_24) (數字) | 24 |
| # K | 時 (am/pm) | (0_11) (數字) | 0 |
| # z | 時區 | (文字) | Pacific Standard Time |
| # ' | 文字結束 | (定界符號) | ' |
| # '' | 單引號 | (文字) | ' |







變更暫存檔的位置

如果您要變更暫存檔的位置，請移除 `webeditor.properties` 檔案中 `temp.dir` 內容的標註，並且加上值。

另外，`java.io.tmpdir` Java 內容是用來決定暫存檔的建立位置。

使用「DTD 產生器」來建立 XML 套表說明檔

forms51_be.xml 檔案是一個 XML 套表說明檔的範本。它提供了一組可供 Web 編輯器使用的套表。此檔案的副本位於下列目錄中：

-  `drive:\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `drive:\Program Files\WebSphere\CommerceServer\xml\ wcwebeditor\xml`
-  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-  `/instroot/xml/wcwebeditor/xml`

以下的步驟可說明系統管理者如何使用「DTD 產生器」來加入新的 XML 套表。

註： 在執行以下所說明的程序之前，請先為 Web 編輯器所在目錄中的現有 forms51_be.xml 變更名稱。另外，請在以下的程序中使用新名稱來建立輸出檔，然後將 Web 編輯器重新架構成使用新檔案。請參閱先前的章節以取得執行此步驟的說明。

若要建立 XML 套表，請執行 DTD Generate 指令。

下列步驟說明了如何建立 XML 套表：

1. 建立名稱為 "tables.txt" 的暫存檔，而其中要包含您要在套表中使用的表格名稱。依照下面的範例，一行輸入一個表格名稱：

```
catentry
catentdesc
catentship
inventory
```

2. 請將 tables.txt 儲存在 DTD Generate 指令找出的目錄。（有關此指令的安裝位置，請參閱第75頁的『第10章 架構載入器套件指令和 Script』。）
3. 請在作業系統的命令提示中，將現行目錄變更至 DTD Generate 找出的目錄。

4. 請輸入以下指令以執行 DTD Generate 指令：



```
dtdgen -infile tables.txt -outfile tables51.dtd
-dbname dbname -dbuser userid -dbpwd password
-xmlTableDesc tableFORMS.xml -schemaname schema -propfile filename
```



```
./dtdgen.sh -infile tables.txt -outfile tables51.dtd
-dbname dbname -dbuser userid -dbpwd password
-xmlTableDesc tableFORMS.xml -schemaname schema -propfile filename
```



```
QWEBCOMM/GENWCSDTD DATABASE(database) SCHEMA(schema)
INSTROOT(instroot) PASSWD(password) OUTFILE(tables51.dtd)
INFILE(tables.txt) XMLTABDESC(tableFORMS.xml)
```

表格說明切換指令 (-xmlTableDesc 或 XMLTABDESC) 可讓「DTD 產生器」建立 DTD 和新的表格套表說明。



-propFile 選項可指定

用來存放說明文字、預設值和欄位說明資訊的外部內容檔名稱。

5. 依照上一節中的說明，將 Web 編輯器重新架構成使用新建的檔案。
6. 在 WebSphere 進階管理主控台中，重新啓動 Web 編輯器。要執行此動作時，請遵循下列步驟。
 - a. 展開 **WebSphere 管理網域**。
 - b. 展開**企業應用程式**。
 - c. 用滑鼠右鍵按下 **WebSphere Commerce 企業應用程式 - demo**，並且選取**停止**。
 - d. 請稍後，直到訊息指示應用程式已停止。
 - e. 用滑鼠右鍵按下 **WebSphere Commerce 企業應用程式 - demo**，並且選取**啓動**。
 - f. 請稍後，直到訊息指示企業應用程式已啓動。
7. 若要在 Web 瀏覽器中查看新的套表，請開啓以下的 URL：

```
https://host_name:8000/wcm/webeditor
```

其中 *host_name* 是指 WebSphere Application Server 的完整 HTTP 主電腦名稱。

Web 編輯器會顯示在瀏覽器視窗中，並列出所有表格名稱。

自訂 XML 套表說明

本節說明管理者可如何強化 Web 編輯器所顯示的套表。

您可以在 XML 套表說明檔或另外的內容檔中設定屬性和值，以便自訂和改進 XML 套表說明。

註：指定給此內容檔的名稱，應該與 formList 標籤中的 resourcePackage 屬性值相同。如果檔案名稱是出現在類別路徑中之目錄的子目錄，則應使用套件（點）規格。

下表列出管理者可改變的 Web 編輯器套表欄位屬性。

| 欄位屬性 | 說明 |
|-------------------------|--|
| Currency | 讓值以語言環境專用的數字分隔字元來顯示（例如美國使用逗點作為千位分隔字元） |
| DbColumn | 用來將欄位名稱對映至內容檔鍵值 如果已使用了語言環境專用的內容檔，則此項目中的值必須符合內容檔中所輸入的值。「DTD 產生器」會將綱目附加至此項目。 |
| DefaultValue | 指定在使用者填入新套表時，資料輸入套表上要顯示的值 「DTD 產生器」可設定此參數作為資料庫的預設值。它可以是靜態的字串，但是也可以包含針對單行表格的 SQL 純量查詢。如果要擷取外部資料，查詢中可以使用使用者定義的函數。比方說： DefaultValue="SELECT CURRENT TIMETAMP FROM EXEC" 其中會定義 EXEC 而且會移入為： CREATE TABLE EXEC (A CHAR(1)); INSERT INTO A VALUES('A'); |
| dynamicSqlSelectionList | 為每一個套表重新建立下拉式功能表 |
| FieldDescription | 提供套表中輸入欄位旁邊所顯示的說明 如果建立這個屬性時有任何直欄註解，DTD 產生器會使用這些註解。如果沒有直欄註解的話，預設值為直欄名稱。 您可在語言環境專屬的內容檔或 XML 套表說明檔中設定此參數。如果內容檔已指定了值，則有較高的優先順序。 |

| 欄位屬性 | 說明 |
|-------------------|---|
| FieldHelp | <p>當欄位成為套表的焦點時，在瀏覽器底部訊息列提供所顯示的欄位簡短說明。</p> <p>依預設，這包含一則隨資料直欄類型來輸入給定欄位之資料的簡單訊息。</p> <p>您可在語言環境專屬的內容檔或 XML 套表說明檔中設定此參數。如果內容檔已指定了值，則有較高的優先順序。</p> |
| formatNumber | <p>告知 Web 編輯器不要處理數字</p> <p>將 formatNumber 屬性設定為 "false"，可使輸入的值在擷取查詢時被視為字串（其中沒有放在雙引號中的值會被視為字串）。此參數的預設值為 "true"。</p> |
| Hidden | 表示此值不會顯示在套表中，但是仍會是 HTML 的隱藏欄位 |
| HideOnCreate | <p>表示在撰寫新套表時可使用此欄位</p> <p>與 showInCreateMode="false" 類似，但是為欄位名稱新增了隱藏類型。</p> |
| Maxlength | <p>指定資料庫直欄的的長度</p> <p>這是用來確定使用者輸入的值不會超過資料庫可儲存的長度。</p> |
| MinOccurs | <p>指示是否為必要欄位</p> <p>值為 "1" 表示為必要欄位；"0" 則表示為選用欄位。</p> |
| 名稱 | 指定資料庫直欄的名稱。 |
| readOnly | 控制編輯時可使用的欄位，和使用者只能讀取的時間。 |
| readOnlyForCreate | readOnly="true" 表示欄位固定為唯讀模式。 |
| readOnlyForEdit | readOnlyForCreate="true" 則表示在撰寫新套表時，欄位處於唯讀模式。readOnlyForEdit="true" 表示在編輯現有套表時，欄位處於唯讀模式。 |
| ShowColumnInList | <p>在設定為 "true" 時，則是指定欄位為構成資料多記錄檢視畫面的直欄。</p> <p>「DTD 產生器」會將前六個直欄設定為 "true"。</p> |

| 欄位屬性 | 說明 |
|------------------|--|
| showInCreateMode | 用來隱藏已建立套表中的欄位 將 showInCreateMode 屬性設定為 "false" 可隱藏顯示的已建立套表中的欄位。此參數的預設值為 "true"。 |
| ShowInSearchMode | 用來隱藏搜尋準則頁面中的某些直欄 如果此屬性設定為 "false"，則指定的欄位不會顯示在搜尋套表中。 |
| SqlSelectionList | 在撰寫新套表時為使用者建立可選取的下拉式選單。 查詢必須傳回一或兩個直欄的結果集。第一個直欄是使用者可以選擇的標籤清單；第二個直欄是儲存在資料庫表格中的實際值清單。如果只使用一個直欄的話，顯示的值會儲存在表格中。此功能在實施外來鍵關係時非常重要。 |
| 類型 | 指示資料庫直欄的類型，以及要對資料執行何種驗證檢查。 例如，若類型為 "integer"，則應用程式會確定只在此欄位中輸入有效的整數。 |
| UniqueKey | 表示欄位為表格的主要鍵 如果此屬性設定為 "true"，則應用程式會對在此直欄中輸入的資料實施唯一限制。此項檢查只會針對載入至應用程式的記錄執行。這並不會包含資料庫中的記錄。 |
| ValidateInput | 用來關閉驗證檢查 這個屬性可讓使用者在 ID 解析程式或 XML 轉換程式所處理的數值欄位中輸入文字。 |

編輯套表名稱

XML 套表說明中的套表標籤有 name 和 displayName 屬性。

- name 屬性必須設為它所代表的資料庫表格或概略表的名稱。
- displayName 屬性可在 Web 編輯器應用程式中提供易用的名稱。另外，您可以在 formList 標籤的 resourcePackage 屬性所指定的內容檔中，設定套表的 displayName 屬性。（resourcePackage 屬性是用來提供多語言的支援。）

註： 如果內容檔中有項目存在，則會改寫在 XML 套表說明檔中所輸入的項目。

變更欄位說明

以下為 XML 套表說明檔的欄位說明範本：

```
<form name = "CATALOG.CATENTRY"  
.  
.  
<field name="MEMBER_ID"  
  showInCreateMode="false"  
  fieldDescription="MEMBER_ID"  
  type="integer"  
  maxLength="19"  
  defaultValue=""  
.  
.  
<field name="CATENTTYPE_ID"  
  fieldDescription="CATENTTYPE_ID"  
  type="string"  
  maxLength="16"  
  defaultValue=""  
.  
.
```

您可以將它變更為：

```
<form name = "CATALOG.CATENTRY"  
  displayName="Product"  
.  
.  
<field name="MEMBER_ID"  
  showInCreateMode="false"  
  fieldDescription="Member Identifier"  
  type="integer"  
  maxLength="19"  
  defaultValue=""  
.  
.  
<field name="CATENTTYPE_ID"  
  fieldDescription="Product Type"  
  type="string"  
  maxLength="16"  
  defaultValue=""  
.  
.
```

新增下拉式功能表

您可以採下拉式功能表形式在 Web 編輯器套表中新增一份選單，讓套表的填寫更加方便，並限制使用者只能從一組有效的選項中挑選。

若要重新整理選單，管理者可以重新啟動 Web 編輯器。管理者還可以將另一個套表說明欄位屬性 `dynamicSqlSelectionList` 設定為 `"true"`，以便確定每一次都會重新載入下拉式功能表。

選單的查詢可能有傳回一或兩個直欄的結果集。如果結果傳回兩個直欄，則第二個直欄會包含實際儲存的值，而第一個直欄會包含使用者標籤。

您可以在欄位標籤的 `sqlSelectionList` 屬性中輸入 SQL 查詢，或在 XML 套表說明檔中建立列舉，就可以建立選單。以下的範例中顯示了這兩種方法：

```
<field name="MEMBER_ID"  
  showInCreateMode="false"
```

```

fieldDescription="Member Identifier"
.
.
readOnly="false"
sqlSelectionList="select orgentityname,orgentity.orgentity_id
from member,orgentity where member.type='0'
and member.member_id=orgentity.orgentity_id"
.
.
fieldHelp=""
.
.
</field>
<field name="MARKFORDELETE"
fieldDescription="Mark for Delete"
type="NMOKEN"
sqlSelectionList=""
>
<datatype source="integer">
<enumeration label="No" value="0"/>
<enumeration label="Yes" value="1"/>
</datatype>
</field>
>

```

WebSphere Commerce 資料庫中有些表格是用 1 和 0 來表示 true 和 false 的。如果要建立使用者更容易瞭解的欄位，您可以建立這些欄位的列舉表。Catalog Manager 提供的 SQL Script 可用來建立名為 "NUMDESC" 的列舉表。此 Script 的名稱為 "createEnum.sql"。管理者可以按照以下範例中所顯示的方式來編輯 sqlSelectionList 屬性，以便使用 ENUMDESC 表格來建立選單。

```

fieldDescription="On Special"
.
.
readOnly="false"
sqlSelectionList="select description,value from
enumdesc where columnname='ALL' and type='YESNO'"
fieldHelp=""
.
.

```

新增欄位說明

現行作用中欄位的欄位說明會顯示在 Web 瀏覽器視窗的底部。XML 套表說明檔中欄位標籤的 fieldHelp 屬性，或內容檔中的 fieldHelp 件都可以用來設定此值。

例如，內容檔可能包含以下的欄位說明規格：

```

CATEGORY.MARKFORDELETE.defaultValue=No
CATEGORY.MARKFORDELETE.fieldDescription=Delete Entry

```

如果未指定任何值，則會建立預設的訊息："請在此輸入 *field_name* 值，*field_type*"。

註： 如果內容檔中包含了套表資訊，則會改寫 XML 套表說明檔中的項目。

自訂搜尋結果和工作階段作業清單

您可以自訂顯示在搜尋結果頁面的直欄集。XML 套表說明檔中的欄位標籤中有 showColumnInList 屬性。若要在搜尋結果中加入欄位，請將此屬性設定為 "true"；否則，它不會在搜尋結果檢視畫面中顯示為直欄。在顯示套表時，該欄位會出現。

編輯 weProcessList 檔案

weProcessList 檔案可讓管理者自訂在處理 Web 編輯器工作階段作業時所要執行的 Catalog Manager 公用程式。

- ▶ DB2 ▶ NT ▶ 2000 ▶ AIX ▶ Solaris

▶ Linux weProcessList.xml 是在下列目錄中：

- ▶ NT drive:\WebSphere\CommerceServer\xml\wcwebeditor\xml
- ▶ 2000 drive:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml
- ▶ AIX /usr/WebSphere/CommerceServer/xml/wcwebeditor/xml
- ▶ Solaris ▶ Linux /opt/WebSphere/CommerceServer/xml/wcwebeditor/xml

- ▶ Oracle ▶ NT ▶ 2000 ▶ AIX ▶ Solaris

▶ Linux weProcessListOracle.xml 檔案是在下列目錄中：

- ▶ NT drive:\WebSphere\CommerceServer\xml\wcwebeditor\xml
- ▶ 2000 drive:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml
- ▶ AIX /usr/WebSphere/CommerceServer/xml/wcwebeditor/xml
- ▶ Solaris ▶ Linux /opt/WebSphere/CommerceServer/xml/wcwebeditor/xml

- ▶ 400 weProcessListAS400.xml 檔案是在下列目錄中：

- /instroot/xml/wcwebeditor/xml

此檔案包含多種公用程式的封套範本。它也可能包含了參照，而可自訂管理者要執行的應用程式。

您可以在此檔案中使用一組系統變數。例如，%-dbname% 系統變數可將資料庫名稱插入讓「載入器」等公用程式進行呼叫的封套。XML 套表說明檔包含了這些程序的參照，而可指示 add、edit 或 delete 應呼叫何者。

以下為 weProcessList.xml 檔案的範例：

```
<?xml version="1.0" encoding="UTF-8"?>
<processSet>
  <!-- Do not change name of extract -->
  <process name="extract"
    subsystem="com.ibm.wca.MassExtract.extract.ExtractSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
      <param name="-dbname" type="scalar"
value="%-dbname%"/>
      <param name="-dbuser" type="scalar"
value="%-dbuser%"/>
      <param name="-dbpwd" type="scalar"
value="%-dbpwd%"/>
      <param name="-outfile" type="file" reside="local"
```

```

value="%tempFilePath%"/>
  <param name="-filter" type="file" reside="local"
value="%tempFileURI1%"/>
  </envelope-input>
</process>
<process name="transformer"
  subsystem="com.ibm.wca.XMLTransformer.XML
TransformerSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-infile" type="file" reside="local"
value="%tempFileURI%"/>
    <param name="-transform" type="file" reside="local"
      value="%webEditorDir%/xsl/ViewsToWCS51.XSL"/>
    <param name="-outfile" type="file" reside="local"
value="%tempFilePath1%"/>
    <param name="-param" value="root=%-dbname%"/>
    <param name="-param" value="dtdname=%-dtdname%"/>
  </envelope-input>
</process>
<process name="transformerForDelete"
  subsystem="com.ibm.wca.XMLTransformer.XMLTransformerSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-infile" type="file" reside="local"
value="%tempFileURI%"/>
    <param name="-transform" type="file" reside="local"
      value="%webEditorDir%/xsl/ViewsToWCS51.XSL"/>
    <param name="-outfile" type="file" reside="local"
value="%tempFilePath1%"/>
    <param name="-param" value="root=%-dbname%"/>
    <param name="-param" value="dtdname=%-dtdname%"/>
    <param name="-param" value="forDelete=true"/>
  </envelope-input>
</process>
<process name="resolver"
  subsystem="com.ibm.wca.IdResGen.IdResGenSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-infile" type="file" reside="local"
      value="%previousOutFileAsURI%"/>
    <param name="-outfile" type="file" reside="local"
      value="%tempFilePath2%"/>
    <param name="-propfile" type="file" reside="local"
      value="propertyFiles.IdKeys"/>
    <param name="-method" type="scalar" value="mixed"/>
  </envelope-input>
</process>
<!-- Resolver as first process -->
<process name="resolverFirstProcess"
  subsystem="com.ibm.wca.IdResGen.IdResGenSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-infile" type="file" reside="local"
value="%tempFileURI%"/>
    <param name="-outfile" type="file" reside="local"
value="%tempFilePath2%"/>
    <param name="-propfile" type="file" reside="local"
value="propertyFiles.IdKeys"/>
    <param name="-method" type="scalar" value="mixed"/>
  </envelope-input>
</process>
<process name="loader"
  subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
  <envelope-input xmlns='saf_params.xsd'>

```

```

        <param name="-dbname" type="scalar" value="%-dbname%"/>
        <param name="-dbuser" type="scalar" value="%-dbuser%"/>
        <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
        <param name="-commitcount" type="scalar" value="1000"/>
        <param name="-infile" type="file" reside="local"
value="%previousOutFileAsURI%"/>
        <param name="-method" type="scalar" value="sqlimport"/>
        <param name="-nopriamry" type="scalar" value="insert"/>
    </envelope-input>
</process>
<process name="loaderFirstProcess"
subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
        <param name="-dbname" type="scalar" value="%-dbname%"/>
        <param name="-dbuser" type="scalar" value="%-dbuser%"/>
        <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
        <param name="-commitcount" type="scalar" value="1000"/>
        <param name="-infile" type="file" reside="local"
value="%tempFileURI%"/>
        <param name="-method" type="scalar" value="sqlimport"/>
        <param name="-nopriamry" type="scalar" value="insert"/>
    </envelope-input>
</process>
<process name="loaderForDelete"
subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
        <param name="-dbname" type="scalar" value="%-dbname%"/>
        <param name="-dbuser" type="scalar" value="%-dbuser%"/>
        <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
        <param name="-commitcount" type="scalar" value="1000"/>
        <param name="-infile" type="file" reside="local"
value="%previousOutFileAsURI%"/>
        <param name="-delete" type="scalar" value=""/>
    </envelope-input>
</process>
<process name="loaderForDeleteFirstProcess"
subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
        <param name="-dbname" type="scalar" value="%-dbname%"/>
        <param name="-dbuser" type="scalar" value="%-dbuser%"/>
        <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
        <param name="-commitcount" type="scalar" value="1000"/>
        <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
        <param name="-delete" type="scalar" value=""/>
    </envelope-input>
</process>
<process name="saveToFile"
    cmd="cmd.exe /c c:\temp\theBatchFile.bat"
    args="-infile %tempFilePath% -dbname %-dbname%"
/>
</processSet>

```

註：檔案參照會區分大小寫。







下表包含了應用程式可辨識的有效替代變數清單。

| % 替代變數 | 傳回值 |
|------------|--------------------|
| %-dbname% | 現行資料庫的名稱 |
| %-dbuser% | 資料庫使用者名稱 |
| %-dtdname% | XML 檔案的 DTD URI 位置 |
| %-dbpwd% | 資料庫使用者名稱的密碼 |

| | |
|--|---|
| <p>%tempFilePath%</p> <p>%tempFilePath1%</p> <p>%tempFilePath2%</p> | <p>暫存檔的完整路徑</p> <p>這些是專屬的暫存檔名稱。它們可以放在封套定義的語法範本或指令行中。舉例來說，假設 %tempFilePath% 是放在封套範本的 <code>-infile</code> 參數屬性值中，則 Web 編輯器會將工作階段作業的資料寫入暫存檔的位置。</p> |
| <p>On SubSystem</p> <p>%tempFileURI%</p> <p>%tempFileURI1%</p> <p>%tempFileURI2%</p> | <p>暫時 URI</p> <p>暫時 URI 是指 %tempFilePath%..%tempFilePath2% 所代表的相同檔案 URI。這不是檔案的附加集，但可以擷取不同語法傳回而建立的相同暫存檔。</p> |
| <p>%previousOutFileAsURI%</p> | <p>在先前作業的 <code>-outfile</code> 參數中表示為 URI</p> |
| <p>%webEditorDir%</p> | <p>Web 編輯器的安裝位置</p> |

編輯 webeditor.xml 檔案

Web 編輯器使用 webeditor.xml 檔案作為預設 XSL 樣式表。此檔案位於以下的目錄中：

-  *drive:\WebSphere\CommerceServer\xml\wcwebeditor\xsl*
-  *drive:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xsl*
-  */usr/WebSphere/CommerceServer/xml/wcwebeditor/xsl*
-   */opt/WebSphere/CommerceServer/xml/wcwebeditor/xsl*
-  */instroot/xml/wcwebeditor/xsl*

管理者可編輯 webeditor.xml 檔，以變更 Web 編輯器輸出的格式。以下為 webeditor.xml 檔案的內容範例：

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html"/>
  <!-- Largest single line entry field value for fields larger than this a
  TEXTAREA is created -->
  <xsl:variable name="maxEntryFieldSize" select="80"/>
  <xsl:template match ="/>
    <xsl:apply-templates/>
  </xsl:template>
  <!-- Read Only field -->
  <xsl:template match="readOnly" name="readOnly">
    <xsl:element name="input">
      <xsl:attribute name="name"><xsl:value-of select="@name"/></xsl:attribute>
      <xsl:attribute name="type">hidden</xsl:attribute>
      <xsl:attribute name="value"><xsl:value-of select="@defaultValue"/></xsl:attribute>
      <!--
      <xsl:attribute name="onFocus">this.blur()</xsl:attribute>
      <xsl:attribute name="style">border-style:groove</xsl:attribute>
      -->
    </xsl:element>
    <table border="1" cellpadding="0" cellspacing="0" width="155" bgcolor="#C0C0C0">
      <tr>
        <td>
          <xsl:value-of select="@defaultValue"/>
        </td>
      </tr>
    </table>
  </xsl:template>
</xsl:stylesheet>
```

第12章 使用型錄

下列程序說明如何使用 Web 編輯器來新增、修改和刪除包含型錄資料的資料庫表格記錄，以便處理型錄。

使用 Web 編輯器在表格中新增記錄

若要使用 Web 編輯器在表格中新增記錄，請遵循以下的步驟：

1. 請在 Web 瀏覽器中開啓以下的 URL：

```
https://host_name:8000/wcm/webeditor
```

其中 *host_name* 是指 WebSphere Application Server 的完整 HTTP 主電腦名稱。

會出現「Web 編輯器資料庫登入」視窗。

2. 輸入您的資料庫使用者名稱與密碼，然後按一下**登入**。
Web 編輯器會顯示在瀏覽器視窗中，並在左功能表列中列出表格名稱。
3. 請在左側功能表列的「新增」子功能表中按下適當的超鏈結。
隨即顯示正確的套表。
4. 在套表中鍵入所有的必要資料。
5. 按下**移動至工作階段作業**。
隨即顯示套表的工作階段作業結果。這些結果包含您在此 Web 編輯器工作階段作業中，進行但還未清除或處理的所有編輯、新增和刪除動作。
6. 如果您要移除工作階段作業中的任何記錄變更，請勾選您要移除的記錄變更前方之方框，然後按下**清除已選取項目**。
7. 按下**處理工作階段作業**以將選取的變更提交至資料庫。
狀態頁面會顯示訊息說明程序已完成。
8. 請瀏覽網站，按下適當的超鏈結，並且驗證您所做的變更。

使用 Web 編輯器來修改表格中的記錄

若要使用 Web 編輯器來修改表格中的記錄，請遵循以下的步驟：

1. 請在 Web 瀏覽器中開啓以下的 URL：

`https://host_name:8000/wcm/webeditor`

其中 *host_name* 是指 WebSphere Application Server 的完整 HTTP 主電腦名稱。

會出現「Web 編輯器資料庫登入」視窗。

2. 輸入您的資料庫使用者名稱與密碼，然後按一下**登入**。
Web 編輯器會顯示在瀏覽器視窗中，並在左功能表列中列出表格名稱。
3. 請在左側功能表列的「搜尋」子功能表中按下適當的超鏈結。
隨即顯示適當的搜尋頁面。
4. 請執行以下步驟來指定搜尋準則：
 - a. 請選取您要在搜尋中指定之所有屬性旁的勾選框。
 - b. 請使用所選取屬性的正確下拉式功能表來選取您要在搜尋中使用的邏輯。
 - c. 請在選取的每一個屬性旁的欄位中鍵入或選取您要在搜尋中使用的值。
5. 按下**尋找**。
這會將搜尋準則提交給 Web 編輯器。
狀態頁面會顯示符合搜尋準則的記錄數目。
6. 執行下列動作之一：
 - 按下**載入資料**以查閱找到的記錄清單。
Web 編輯器會列出取自查詢中的記錄。
跳至步驟 6。
 - 按下**新搜尋**以返回搜尋頁面。
跳回至步驟 3。
7. 選取您要編輯的記錄。
隨即顯示正確的套表。
8. 向下捲動至您要編輯的欄位，並且變更其內容。
9. 按下**移動至工作階段作業**。
隨即顯示套表的工作階段作業結果。這些結果包含您在此 Web 編輯器工作階段作業中，進行但還未清除或處理的所有編輯、新增和刪除動作。
10. 如果您要移除工作階段作業中的任何記錄變更，請勾選您要移除的記錄變更前方之方框，然後按下**清除已選取項目**。
11. 按下**處理工作階段作業**以將選取的變更提交至資料庫。
狀態頁面會顯示訊息說明程序已完成。
12. 請瀏覽網站，按下適當的超鏈結，並且驗證您所做的變更。

使用 Web 編輯器來刪除表格中的記錄

若要使用 Web 編輯器來刪除表格中的記錄，請遵循以下的步驟：

1. 請在 Web 瀏覽器中開啓以下的 URL：

`https://host_name:8000/wcm/webeditor`

其中 *host_name* 是指 WebSphere Application Server 的完整 HTTP 主電腦名稱。

會出現「Web 編輯器資料庫登入」視窗。

2. 輸入您的資料庫使用者名稱與密碼，然後按一下**登入**。

Web 編輯器會顯示在瀏覽器視窗中，並在左功能表列中列出表格名稱。

3. 請在左側功能表列的「搜尋」子功能表中按下適當的超鏈結。

隨即顯示適當的搜尋頁面。

4. 請執行以下步驟來指定搜尋準則：

- a. 請選取您要在搜尋中指定之所有屬性旁的勾選框。
- b. 請使用所選取屬性的正確下拉式功能表來選取您要在搜尋中使用的邏輯。
- c. 請在選取的每一個屬性旁的欄位中鍵入或選取您要在搜尋中使用的值。

5. 按下**尋找**。

這會將搜尋準則提交給 Web 編輯器。

狀態頁面會顯示符合搜尋準則的記錄數目。

6. 執行下列動作之一：

- 按下**載入資料**以查閱找到的記錄清單。
Web 編輯器會列出取自查詢中的記錄。
跳至步驟 6。
- 按下**新搜尋**以返回搜尋頁面。
跳回至步驟 3。

7. 選取您要刪除的所有記錄旁之勾選框。

8. 按下**移動至刪除清單**。

9. 請在左側功能表列的「工作階段作業」子功能表中按下適當的超鏈結。

隨即顯示套表的工作階段作業結果。這些結果包含您在此 Web 編輯器工作階段作業中，進行但還未清除或處理的所有編輯、新增和刪除動作。

10. 如果您要移除工作階段作業中的任何記錄變更，請勾選您要移除的記錄變更前方之方框，然後按下**清除已選取項目**。

11. 按下**處理工作階段作業**以將變更提交至資料庫。

狀態頁面會顯示訊息說明程序已完成。

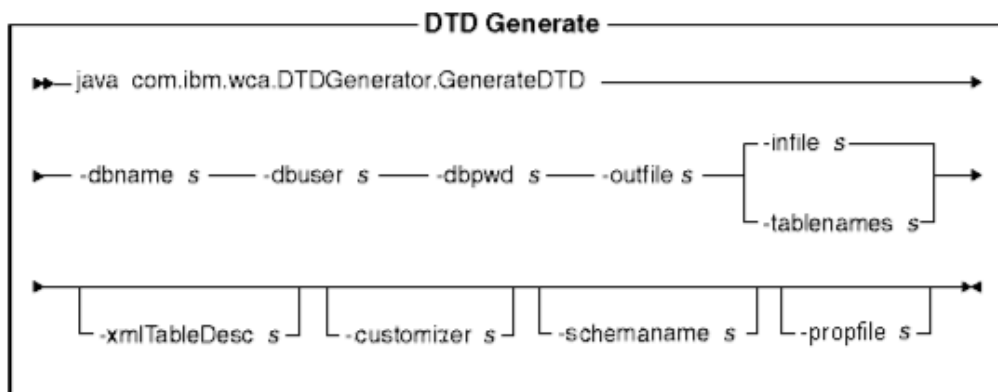
12. 請瀏覽網站，按下適當的超鏈結，並且驗證您所做的變更。

第4篇 指令說明

第13章 DTD Generate 指令

此指令可建立用於「載入器」套件的 DTD 和綱目檔。

Windows、AIX、Linux 和 Solaris 系統的 DTD Generate 指令



註:

1. 上圖主要做為指令參數的參考用。針對此指令所提供以及第75頁的『第10章 架構載入器套件指令和 Script』下所列的指令檔或 Script 可視為實際 Java 指令的外層；因此，建議您使用指令檔或 Script，而不要直接呼叫 Java 指令。
2. 此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

-dbname

目標資料庫的名稱

-dbuser

連接至資料庫的使用者名稱

-dbpwd

連接至資料庫的使用者密碼

-outfile

輸出 DTD 檔的名稱

-infile 輸入檔的名稱，檔案中每行都有一個資料庫表格名稱

-tablename

表格名稱，以逗點隔開。

-xmlTableDesc

要建立的綱目檔的檔案路徑

-customizer

要用的自訂程式內容檔的名稱。DB2ConnectionCustomizer.properties 是預設檔。customizer 內容檔可以根據下列範例中的方式來指定：

-customizer d:\wc\prop\dtdgen.properties

如果這個檔案是在 `classpath system-environment` 變數中指定的目錄裡，就可以根據下列範例中的方式來指定同一個檔案：

```
-customizer dtdgen
```

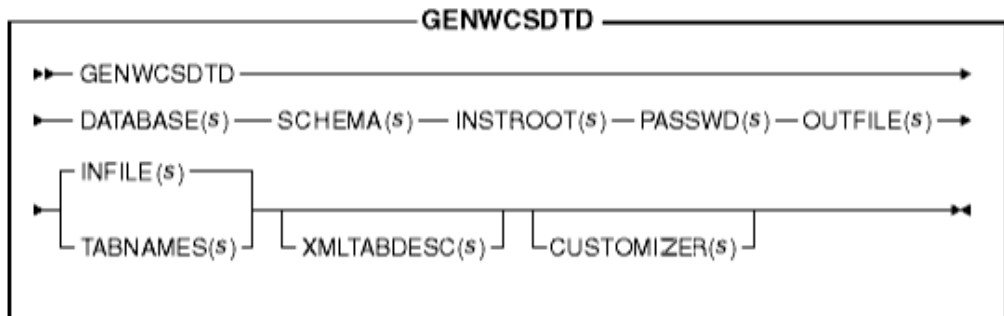
-schemaname

目標資料庫綱目的名稱

-propFile

包含外部內容檔等內容的檔案；其中可儲存 Web 編輯器套表說明的說明文字、預設值和欄位說明資訊

適用於 iSeries 系統的 DTD Generate 指令



註：此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

DATABASE

目標資料庫的名稱，如關聯式資料庫目錄中所顯示

SCHEMA

目標資料庫綱目的名稱；這與案例名稱相同

INSTROOT

WebSphere Commerce 案例起始位置路徑的完整名稱，如 `/QIBM/UserData/WebCommerce/instances/instance_name`

PASSWD

WebSphere Commerce 案例的密碼

OUTFILE

輸出 DTD 檔的名稱

INFILE

輸入檔的名稱，檔案中每行都有一個資料庫表格名稱

TABNAMES

表格名稱，以逗點隔開。

XMLTABDESC

要建立的綱目檔的檔案路徑。此參數為選用的。

CUSTOMIZER

要用的自訂程式內容檔的名稱。預設檔是 `ISeries_GENWCSDTD_Customizer.properties`。customizer 內容檔可以根據下列範例中的方式來指定：

```
CUSTOMIZER(/wc/prop/dtdgen.properties)
```

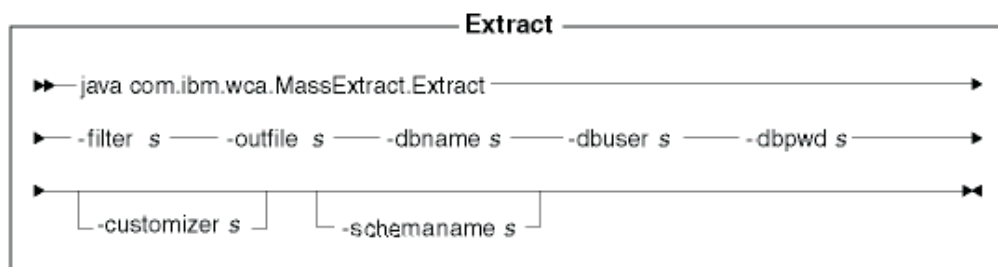
如果這個檔案是在 `classpath system-environment` 變數中指定的目錄裡，就可以根據下列範例中的方式來指定同一個檔案：

```
CUSTOMIZER(dtdgen)
```

第14章 Extract 指令

這個指令會採用 XML 檔，從資料庫中擷取所選的資料子集。

Windows、AIX、Linux 和 Solaris 系統的 Extract 指令



註:

1. 上圖主要做為指令參數的參考用。針對此指令所提供以及第75頁的『第10章 架構載入器套件指令和 Script』下所列的指令檔或 Script 可視為實際 Java 指令的外層；因此，建議您使用指令檔或 Script，而不要直接呼叫 Java 指令。
2. 此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

-filter 擷取過濾器檔的名稱

-outfile
擷取的資料要儲存其中之輸出 XML 檔的名稱

-dbname
要從中擷取資料之資料庫的名稱

-dbuser
要從中擷取資料之資料庫的資料庫使用者名稱

-dbpwd
要從中擷取資料之資料庫的使用者名稱所關聯的密碼

-customizer
要用的自訂程式內容檔的名稱。DB2ConnectionCustomizer.properties 是預設檔。customizer 內容檔可以根據下列範例中的方式來指定：

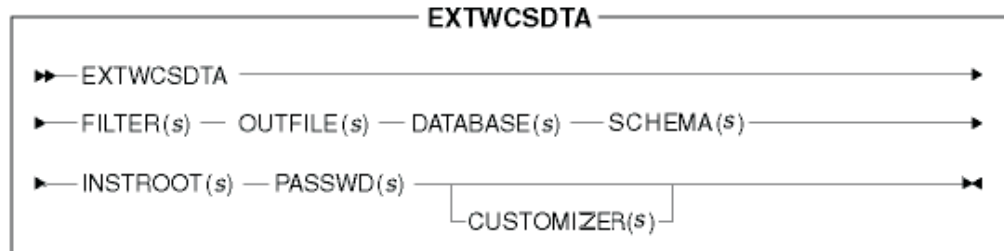
```
-customizer d:\wc\prop\extract.properties
```

如果這個檔案是在 classpath system-environment 變數中指定的目錄裡，就可以根據下列範例中的方式來指定同一個檔案：

```
-customizer extract
```

-schemaname
目標資料庫綱目的名稱

適用於 iSeries 系統的 Extract 指令



註：此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

FILTER

擷取過濾器檔的名稱

OUTFILE

擷取的資料要儲存其中之輸出 XML 檔的名稱

DATABASE

從中擷取資料之資料庫的名稱，如關聯式資料庫目錄中所顯示

SCHEMA

從中擷取資料之資料庫綱目的名稱；這與案例名稱相同

INSTROOT

WebSphere Commerce 案例起始位置路徑的完整名稱，如
`/QIBM/UserData/WebCommerce/instances/instance_name`

PASSWD

WebSphere Commerce 案例的密碼

CUSTOMIZER

要用的自訂程式內容檔的名稱。預設檔是

`ISeries_EXTWCSDTA_Customizer.properties`。customizer 內容檔可以根據下列範例中的方式來指定：

```
CUSTOMIZER(/wc/prop/extract.properties)
```

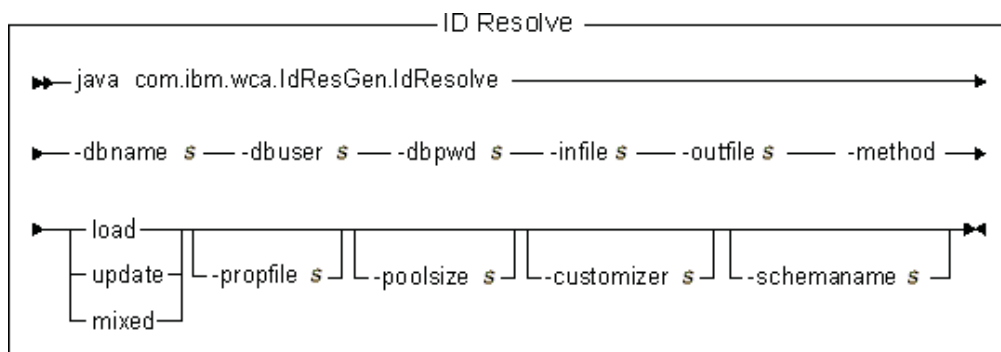
如果這個檔案是在 `classpath system-environment` 變數中指定的目錄裡，就可以根據下列範例中的方式來指定同一個檔案：

```
CUSTOMIZER(extract)
```

第15章 ID Resolve 指令

這個指令會在載入到資料庫之前，產生需要它們之 XML 元素的識別碼。

Windows、AIX、Linux 和 Solaris 系統的 ID Resolve 指令



註:

1. 上圖主要做為指令參數的參考用。針對此指令所提供以及第75頁的『第10章 架構載入器套件指令和 Script』下所列的指令檔或 Script 可視為實際 Java 指令的外層；因此，建議您使用指令檔或 Script，而不要直接呼叫 Java 指令。
2. 此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

-dbname

目標資料庫的名稱

-dbuser

連接至資料庫的使用者名稱

-dbpwd

連接至資料庫的使用者密碼

-infile 含有表格記錄描述子的輸入 XML 文件的名稱

-outfile

要產生的輸出 XML 檔名稱；這個檔案可作為載入器的輸入

-method

用來處理輸入檔的方法

- 如果檔案中的**所有**記錄在資料庫中都不存在，請使用載入方法來處理輸入檔。
- 如果檔案中的**所有**記錄都**存在**於資料庫中，請使用更新方法來處理輸入檔。
- 如果只有檔案中的**部份**記錄**存在**於資料庫中，請使用混合方法來處理輸入檔。

預設方法是載入。

-profile

含有 Java 內容（格式為「名稱=值」配對）的文字檔。這個檔案用來定義類似的直欄名稱，以查閱外來鍵的識別碼，以及定義 `select` 述語，以查詢主要表格（如 `CATEGORY` 和 `PRODUCT`）。在這個檔案中，您可以省略定義了不含識別碼之唯一索引的表格項目。預設檔案為 `IdResolveKeys.properties`。雖然這個檔案的副檔名是 `.properties`，但在指定值時，請勿使用副檔名。

-poolsize

要保留的識別碼數目。預設值是 50 個。

-customizer

要用的自訂程式內容檔的名稱。 `DB2ConnectionCustomizer.properties` 是預設檔。 `customizer` 內容檔可以根據下列任一範例中的方式來指定：

```
-customizer d:\wc\prop\idres.properties
```

```
-customizer d:\wc\prop\idres
```

如果這個檔案存在於現行目錄中，您可按照下列範例指定同一檔案：

```
-customizer idres.properties
```

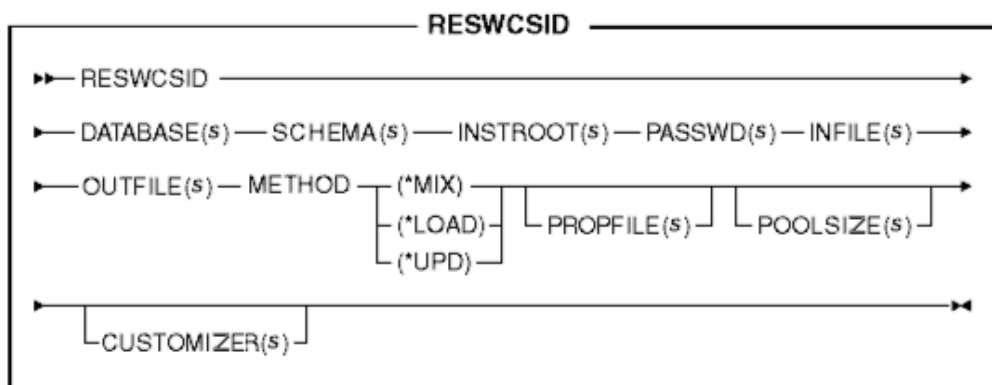
如果這個檔案是在 `classpath system-environment` 變數中指定的目錄裡，就可以根據下列範例中的方式來指定同一個檔案：

```
-customizer idres
```

-schemaname

目標資料庫綱目的名稱

適用於 iSeries 系統的 ID Resolve 指令



註：此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

DATABASE

目標資料庫的名稱，如關聯式資料庫目錄中所顯示

SCHEMA

目標資料庫綱目的名稱；這與案例名稱相同

INSTROOT

WebSphere Commerce 案例起始位置路徑的完整名稱，如
`/QIBM/UserData/WebCommerce/instances/instance_name`

PASSWD

WebSphere Commerce 案例的密碼

INFILE

含有表格記錄描述子的輸入 XML 文件的名稱

OUTFILE

要產生的輸出 XML 檔名稱；這個檔案可作為載入器的輸入

METHOD

用來處理輸入檔的方法

- 如果檔案中的**所有**記錄皆**不存在**於資料庫中，請使用載入方法 (*LOAD) 來處理輸入檔。
- 如果檔案中的**所有**記錄都**存在**於資料庫中，請使用更新方法 (*UPD) 來處理輸入檔。
- 如果檔案中只有**部份**記錄**存在**於資料庫中，請使用混合方法 (*MIX) 來處理輸入檔。

PROFILE

含有 Java 內容（格式為「名稱=值」配對）的文字檔。這個檔案用來定義類似的直欄名稱，以查閱外來鍵的識別碼，以及定義 select 述語，以查詢主要表格（如 CATEGORY 和 PRODUCT）。在這個檔案中，您可以省略定義了不含識別碼之唯一索引的表格項目。預設檔案為 `IdResolveKeys.properties`。雖然這個檔案的副檔名是 `.properties`，但在指定值時，請勿使用副檔名。

POOLSIZE

要保留的識別碼數目。預設值是 50 個。

CUSTOMIZER

要用的自訂程式內容檔的名稱。預設檔是

`ISeries_RESWCSID_Customizer.properties`。customizer 內容檔可以根據下列任一範例中的方式來指定：

```
CUSTOMIZER(/wc/prop/idres.properties)
```

```
CUSTOMIZER(/wc/prop/idres)
```

如果這個檔案存在於現行目錄中，您可按照下列範例指定同一檔案：

```
CUSTOMIZER(idres.properties)
```

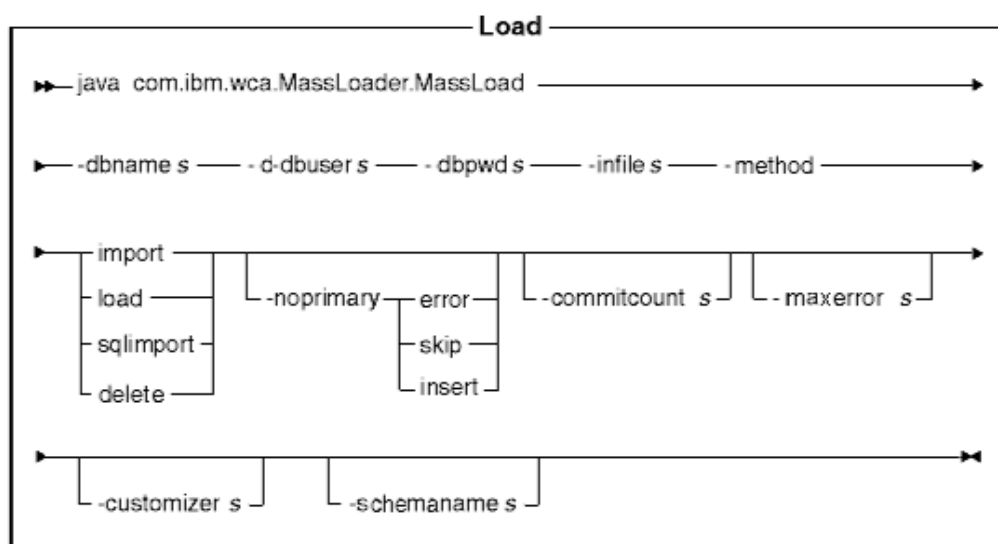
如果這個檔案是在 `classpath system-environment` 變數中指定的目錄裡，就可以根據下列範例中的方式來指定同一個檔案：

```
CUSTOMIZER(idres)
```


第16章 Load 指令

這個指令會將 XML 輸入檔載入目標資料庫中。

Windows、AIX、Linux 和 Solaris 系統的 Load 指令



註:

1. 上圖主要做為指令參數的參考用。針對此指令所提供以及第75頁的『第10章 架構載入器套件指令和 Script』下所列的指令檔或 Script 可視為實際 Java 指令的外層；因此，建議您使用指令檔或 Script，而不要直接呼叫 Java 指令。
2. 此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

-dbname

目標資料庫的名稱

-dbuser

連接至資料庫的使用者名稱

-dbpwd

連接至資料庫的使用者密碼

-infile 輸入 XML 檔的名稱

-method

將資料插入到資料庫中時，載入器所用的作業模式

- load 方法會採用資料庫供應商所提供的原生載入器。您可以針對本端和遠端 Oracle 資料庫使用 load 方法；但是 load 方法只能用於本端 DB2 資料庫。

- 使用匯入方法將資料載入至遠端 DB2 資料庫中。如果資料庫供應商有提供的话，匯入方法會使用 `import` 或 `update` 選項。如果要採取此方法的資料庫並不適用於 `import` 或 `update` 選項（如 Oracle），則會以採用 JDBC 的 SQL 陳述式來更新資料庫。
- SQL `import` (`sqlimport`) 方法適用於本端和遠端資料庫。
- `delete` 方法會刪除資料庫中的資料。

如果您使用「產品顧問」搜尋空間同步化，您必須使用 `sqlimport` 或 `delete` 方法。

-noprimary

當輸入檔中有記錄遺漏主要鍵時，載入器必須採取的動作。 `error` 選項表示載入器應將遺漏主要鍵當成一則錯誤提出報告並且終止執行。 `skip` 選項會略過輸入檔中沒有主要鍵的記錄。插入選項會嘗試處理（插入或刪除）資料。預設動作為 `error`。

-commitcount

當採用作業的 SQL 更新方法時，在處理了多少筆記錄之後，要進行資料庫確定。預設值是 1。

-maxerror

當採用作業的 SQL 更新方法時，在發生多少錯誤之後，載入器就要終止執行。

-customizer

要用的自訂程式內容檔的名稱。 `MassLoadCustomizer.properties` 是預設檔。 `customizer` 內容檔可以根據下列範例中的方式來指定：

```
-customizer d:\wc\prop\ml.properties
```

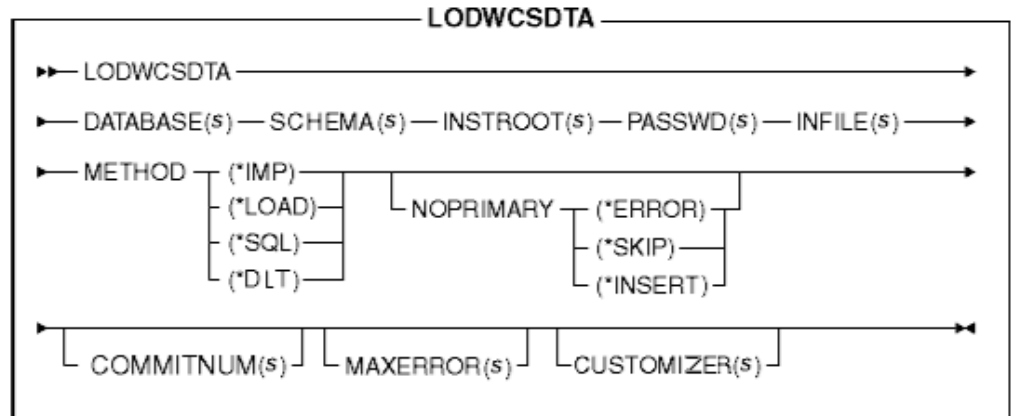
如果這個檔案是在 `classpath system-environment` 變數中指定的目錄裡，就可以根據下列範例中的方式來指定同一個檔案：

```
-customizer ml
```

-schemaname

目標資料庫綱目的名稱

適用於 iSeries 系統的 Load 指令



註：此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

DATABASE

目標資料庫的名稱，如關聯式資料庫目錄中所顯示

SCHEMA

目標資料庫綱目的名稱；這與案例名稱相同

INSTROOT

WebSphere Commerce 案例起始位置路徑的完整名稱，如
`/QIBM/UserData/WebCommerce/instances/instance_name`

PASSWD

WebSphere Commerce 案例的密碼

INFILE

輸入 XML 檔的名稱

METHOD

將資料插入到資料庫中時，載入器所用的作業模式

- load 方法 (*LOAD) 會採用資料庫供應商所提供的原生載入器。您可以針對本端和遠端 Oracle 資料庫使用 load 方法 (*LOAD)；但是 load 方法 (*LOAD) 只能用於本端 DB2 資料庫。
- 使用匯入選項 (*IMP) 將資料載入至遠端 DB2 資料庫。如果資料庫供應商有提供的話，import (*SQL) 方法會採用 import 或 update 選項。如果無 import 或 update 選項可用，則會以採用 JDBC 的 SQL 陳述式來更新資料庫。
- SQL import (*SQL) 方法適用於本端和遠端資料庫。
- delete (*DLT) 方法會刪除資料庫中的資料。

NOPRIMARY

當輸入檔中有記錄遺漏主要鍵時，載入器必須採取的動作。error 選項 (*ERROR) 表示載入器應將遺漏主要鍵當成一則錯誤提出報告並且終止執行。skip 選項 (*SKIP) 會略過輸入檔中沒有主要鍵的記錄。insert 選項 (*INSERT) 會嘗試處理（插入或刪除）資料。預設動作為 error。

COMMITNUM

當採用作業的 SQL 更新方法時，在處理了多少筆記錄之後，要進行資料庫確定。預設值是 1。

MAXERROR

當採用作業的 SQL 更新方法時，在發生多少錯誤之後，載入器就要終止執行。

CUSTOMIZER

要用的自訂程式內容檔的名稱。預設檔是

ISeries_LODWCSDTA_Customizer.properties。customizer 內容檔可以根據下列範例中的方式來指定：

```
CUSTOMIZER(/wc/prop/ml.properties)
```

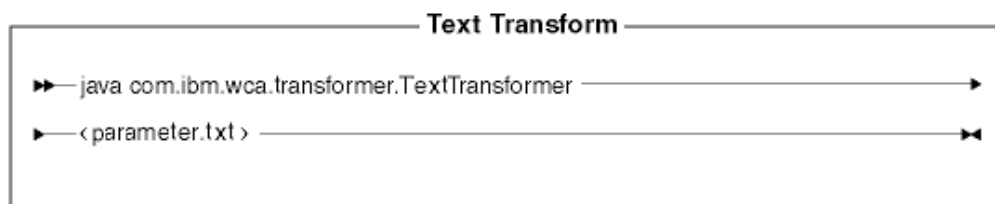
如果這個檔案是在 classpath system-environment 變數中指定的目錄裡，就可以根據下列範例中的方式來指定同一個檔案：

```
CUSTOMIZER(ml)
```

第17章 Text Transform 指令

此指令可在以字元界定的變數格式與 XML 格式間轉換資料。

Windows、AIX、Linux 和 Solaris 系統的 Text Transform 指令



註: 上圖主要做為指令參數的參考用。針對此指令所提供以及第75頁的『第10章 架構載入器套件指令和 Script』下所列的指令檔或 Script 可視為實際 Java 指令的外層；因此，建議您使用指令檔或 Script，而不要直接呼叫 Java 指令。

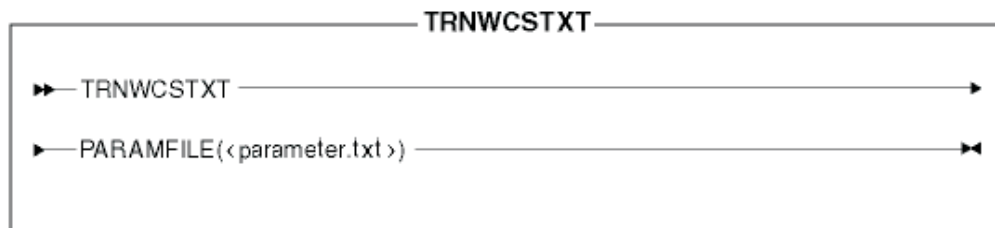
參數值

您可在參數檔 (*parameter.txt*) 中指定和以逗點分隔下列值：

- input file
要轉換的檔案名稱
- schema file
轉換中要使用的 XML 綱目檔名稱
- output file
將儲存已轉換資料的輸出檔名稱
- transformation method
用來將資料新增至輸出檔的方法。如果要建立新檔案，請指定**建立**；如果要將輸出資料附加到現有資料檔中，請指定**附加**。

註: 此檔案也可稱為 "manifest" 或 "command" 檔。

適用於 iSeries 系統的 Text Transform 指令



參數值

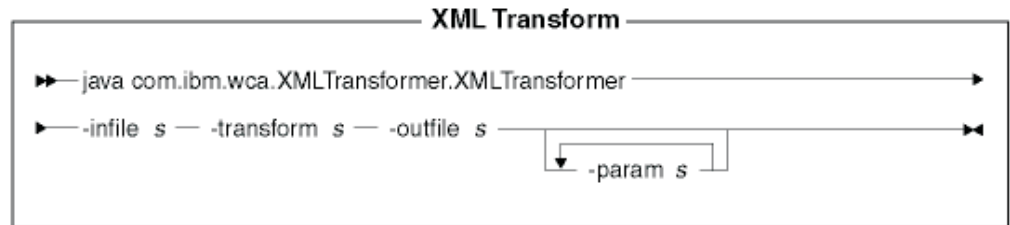
您可在參數檔 (*parameter.txt*) 中指定和以逗點分隔下列值：

- **input file**
要轉換的檔案名稱
- **schema file**
轉換中要使用的 XML 綱目檔名稱
- **output file**
將儲存已轉換資料的輸出檔名稱
- **transformation method**
用來將資料新增至輸出檔的方法。如果要建立新檔案，請指定**建立**；如果要將輸出資料附加到現有資料檔中，請指定**附加**。

第18章 XML Transform 指令

這個指令會將 XML 檔轉換成替代的 XML 格式。

Windows、AIX、Linux 和 Solaris 系統的 XML Transform 指令



註:

1. 上圖主要做為指令參數的參考用。針對此指令所提供以及第75頁的『第10章 架構載入器套件指令和 Script』下所列的指令檔或 Script 可視為實際 Java 指令的外層；因此，建議您使用指令檔或 Script，而不要直接呼叫 Java 指令。
2. 此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

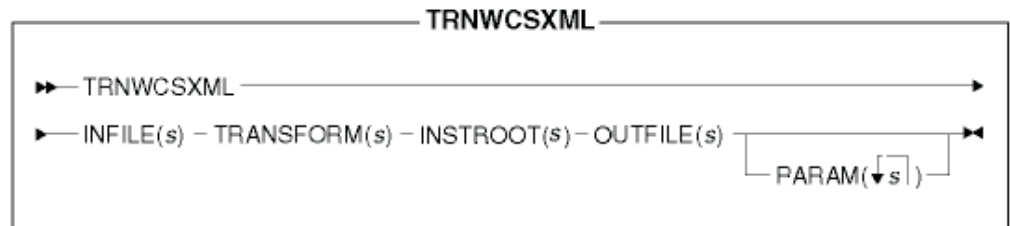
-infile 要轉換的檔案名稱

-transform
轉換 XSL 規則檔的名稱

-outfile
要儲存轉換資料的 XML 輸出檔的名稱

-param
要傳遞給 XSL 規則檔的參數。**此為選用參數**。這個參數可以重複指定，以傳送多重 "name=value" 配對。

適用於 iSeries 系統的 XML Transform 指令



註: 此指令中所指定的檔名參數前可加上相對或絕對路徑。

參數值：

INFILE

要轉換的檔案名稱

TRANSFORM

轉換 XSL 規則檔的名稱

INSTROOT

WebSphere Commerce 案例起始位置路徑的完整名稱，如
`/QIBM/UserData/WebCommerce/instances/instance_name`

OUTFILE

要儲存轉換資料的 XML 輸出檔的名稱

PARAM

要傳遞給 XSL 規則檔的參數。**此為選用參數**。字串可含有多個值，以傳遞多個
"name=value" 配對。

注意事項

本資訊是針對 IBM 在美國所提供之產品與服務開發出來的。而在其他國家中，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 服務代表查詢。凡提及 IBM 產品、程式或服務項目時，亦不表示只可用 IBM 的產品、程式或服務項目。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 產品、程式或服務的作業，使用者都必須自負評估和驗證責任。

IBM 可能已經申請與本書有關（包括本書的主題內容）的各項專利權，或者具有正在審理中的專利權申請書。使用者不得享有本書內容之專利權。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.

若要查詢有關二位元組 (DBCS) 資訊的特許權限事宜，請聯絡您國家的 IBM 智慧財產部門，或者用書面方式寄到：

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

IBM 僅以現狀提供本書，而不提供任何明示或默示之保證（包括但不限於可售性或符合特定效用的保證）。若有些地區在某些交易上並不允許排除上述保證，則該排除無效。

本書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。同時，IBM 得隨時改進並（或）變動本書中所提及的產品及（或）程式。

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。這些網站中的材料不在本 IBM 產品的材料中，您在使用這些網站時，要自行承擔風險。

IBM 得以各種適當的方式使用或散布由 貴客戶提供的任何資訊，而無需對您負責。

本程式之獲授權者若希望取得相關資料，以便使用下列資訊者可洽詢 IBM。其下列資訊指的是：(1) 獨立建立的程式與其他程式（包括此程式）之間更換資訊的方式 (2) 相互使用已交換之資訊方法若有任何問題請聯絡：

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

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之「IBM 客戶合約」、「國際程式授權合約」或任何同等合約之條款，提供本文件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 並未測試過這些產品，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。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。

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。

所有顯示的 IBM 售價都是 IBM 的建議零售價，這些都是目前的售價且隨時會不經過通知即變更。 經銷商提供的價格可會有不同。

本資訊只供作規劃之用。在說明的產品上市之前，這些資訊仍會變更。

本書中的範例包含了用於日常商業活動的資料及報告。為了盡可能詳細，範例中涵蓋了個人、公司、品牌及產品的名稱。所有名稱純屬虛構，如有雷同純屬巧合。

著作權授權：

本書包含範例應用程式的來源語言，用來說明在各種作業平台上的程式設計技術。 客戶不需付費給 IBM 即可以任何格式複製、修改或散布這些範例程式，作為開發、使用、行銷或散布符合範例程式針對的目標作業平台應用程式設計介面的應用程式。這些範例並未在所有條件下進行完整測試。因此，IBM 無法保證或暗示這些程式的可靠性、有用性或功能。客戶不需付費給 IBM 即可以任何格式複製、修改或散布這些範例程式，作為開發、使用、行銷或散布符合 IBM 應用程式設計介面的應用程式。

這些範例程式或任何衍生程式的全部或部分拷貝，都必須包括下列版權聲明：

©Copyright International Business Machines Corporation 2001. 此程式碼衍生自 IBM Corp. 範例程式。©Copyright IBM Corp. 2000, 2001. All rights reserved.

如果您看到的是本資訊的軟本。其圖片和顏色圖例可能不會出現。

商標及服務標示

下列詞彙為 IBM 公司在美國及（或）其他國家的商標：

AIXDB2

DB2 Universal DatabaseIBMiSeriesOS/400

WebSphere

Microsoft、Windows、Windows NT 和 Windows 2000 是 Microsoft Corporation 在美國或其他國家的註冊商標。

Oracle 是 Oracle Corporation 在美國或其他國家的註冊商標。

Solaris、Java 和所有以 Java 為基礎的商標和標誌圖是 Sun Microsystems, Inc. 在美國或其他國家的商標或註冊商標。

其他公司、產品和服務名稱可能是第三者的商標或服務標記。

讀者意見表

為使本書盡善盡美，本公司極需您寶貴的意見；懇請您使用過後，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 |
| | 實際畫面訊息與本書所提之畫面訊息是否一致 | <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，版面安排是否方便使用 | <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，是否便於查考 | <input type="checkbox"/> 是 <input type="checkbox"/> 否 |
| | 索引語錄之排定，是否便於查考 | <input type="checkbox"/> 是 <input type="checkbox"/> 否 |
| ※評估意見為"否"者，請於備註欄說明。 | | |

其他：（篇幅不夠時，請另紙說明。）

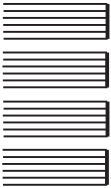
上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。

IBM® WebSphere® Commerce
Catalog Manager 使用手冊
第 5.4 版

折疊線

台北市 110 基隆路一段 206 號

臺灣國際商業機器股份有限公司 啟
大中華研發中心 軟體國際部



廣告回信
台灣北區郵政管理局
登記
北台字第 0587 號

(免貼郵票)

寄件人 姓名：
地址：

寄

折疊線

讀者意見表

IBM