

從 WebSphere Partner Gateway 6.1 版呼叫 WebSphere Transformation Extender 8.2 版對映

本文探討整合 WebSphere Partner Gateway 與 WebSphere Transformation Extender 所需的配置，您可從中瞭解 WebSphere Transformation Extender 所支援的呼叫方法，以及 WebSphere Partner Gateway 應在何時使用何種方法。此外，您將會瞭解如何使用 WebSphere Transformation Extender、Java JNI 和 Java RMI API，以整合 WebSphere Partner Gateway 與 WebSphere Transformation Extender。

前言

IBM WebSphere Transformation Extender（以下稱爲 Transformation Extender）是資料轉換與驗證引擎。已經熟悉 WebSphere Partner Gateway（此後稱爲 Partner Gateway）的架構設計師、系統管理員與開發人員，會樂於見到將 WebSphere Transformation Extender 提供給 Partner Gateway 使用，以延伸其功能的各種型樣。本文探討如何整合 Partner Gateway 與 Transformation Extender，以便執行資料轉換。

本文除了簡介 Transformation Extender 之外，還會說明 Partner Gateway 如何處理文件，從接收文件開始到遞送至目的地。文中將介紹所有的整合型樣，並且深入說明 Java JNI 與 RMI 型樣。閱畢本文之後，您將會了解如何以 Java JNI 或 Java RMI 型樣整合 Partner Gateway 與 Transformation Extender。

建議您先取得 WebSphere Partner Gateway 與 WebSphere Transformation Extender 的必備知識。

必要的產品版本與修正程式

本文的個案研討需要使用下列產品版本與修正程式：

- WebSphere Partner Gateway Enterprise Edition Version 6.1 版

- 與 Fix Pack 1
- WebSphere Transformation Extender Design Studio 8.2 版與所有最新的 iFixes 與 Fix Pack
- WebSphere Transformation Extender SDK 8.2 版與所有最新的 iFixes 與 Fix Pack

WebSphere Transformation Extender

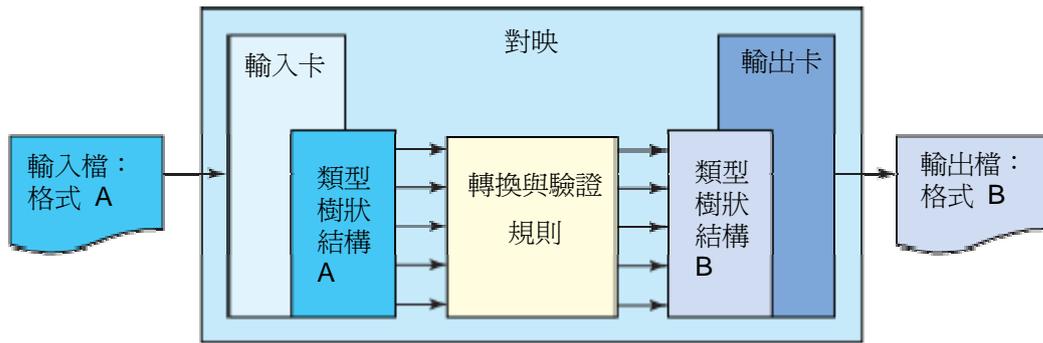
WebSphere Transformation Extender 8.2 版使用無需撰寫程式碼的圖形化方法進行開發。本文把介紹重點放在大量、交易導向與「任意通行」的資料轉換與驗證引擎，以迎合 Transformation Extender 的基礎。請參閱「資源」部分，以進一步瞭解 IBM 為轉換與驗證引擎所新增的各種 Transformation Extender 供應項目。

讓我們先了解基本知識：

- Transformation Extender 轉換與驗證對映，為 Transformation Extender Design Studio 所設計，編譯成平台專用檔案後，供轉換與驗證引擎於執行時使用。
- 在 Transformation Extender 中，使用類型樹狀結構說明資料格式。Transformation Extender Industry Packs 提供適用多種標準格式（EDI、HIPAA 等）的預先建置「類型樹狀結構」，同時提供各式精靈，以便您從 XML 綱目、COBOL Copy Books 與多種其他格式產生「類型樹狀結構」。
- 卡片 (Card) 提供將 meta 資料附加至「類型樹狀結構」的機制。此 meta 資料是由各種參數所組成，其中包括輸入檔案與位置（在「輸入卡」上）或輸出檔名與位置（在「輸出卡」上）。此外，卡片也會指定供轉換與驗證引擎在執行時使用的參數，包括驗證層次，以及在處理過程中遇到警告或錯誤時會遇到的狀況。
- Transformation Extender 對映通常具有一個輸入卡與一個輸出卡，以及一組轉換與驗證規則。

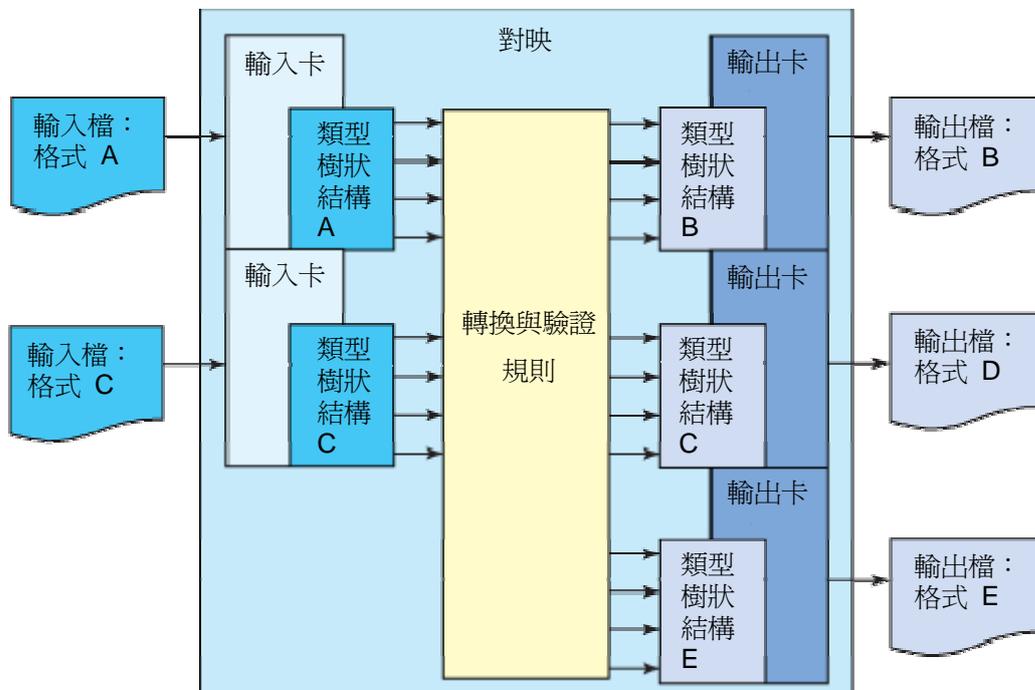
下方圖 1 中的圖表顯示可從格式 A 轉換成格式 B 的 Transformation Extender 對映。

圖 1：Transformation Extender 對映將資料從格式 A 轉換成格式 B



下方圖 2 的圖表顯示 Transformation Extender 支援擁有多個「輸入卡」與「輸出卡」的對映。

圖 2：擁有多個輸入卡與輸出卡的 Transformation Extender 對映



使用 JNI 呼叫 Transformation Extender 對映

Transformation Extender 提供 Java API，可讓您從 Java 呼叫 Transformation Extender 對映。假定 Transformation Extender 是 C++ 應用程式，此 Java API 使用 JNI (Java 原生介面)，而且要求您將 Transformation Extender 安裝至與 Partner Gateway 的相同實體機器上。

在下方清單 1 中的程式碼片段 (Code snippet) 顯示如何使用 Java API 呼叫 Transformation Extender 對映：

清單 1：使用 JNI 的 Transformation Extender Java API snippet

```
import com.ibm.websphere.dtx.dtxpi.MConstants;
import com.ibm.websphere.dtx.dtxpi.MException;
import com.ibm.websphere.dtx.dtxpi.MMap;

...

// Initialize the API
MMap.initializeAPI(null);

// Create a map
MMap map = new MMap("test1.mmc");

// Run the map
map.run();

// Check the return status
int iRC = map.getIntegerProperty(MConstants.MPIP_OBJECT_ERROR_CODE, 0);
String szMsg = map.getTextProperty(MConstants.MPIP_OBJECT_ERROR_MSG, 0);
System.out.println("Map status: " + szMsg + " (" + iRC + ")");

// Clean up
map.unload();
MMap.terminateAPI();
```

使用 RMI 呼叫 Transformation Extender 對映

Transformation Extender 也提供 Java RMI API，讓您可改用 RMI（遠端方法呼叫）介面而非 JNI 介面，以呼叫 Transformation Extender 對映。您也可以使用 RMI 介面，應用於 Transformation Extender 轉換與驗證引擎，及安裝於不同機器的 Partner Gateway 實例。

在下方清單 2 中的程式碼片段 (Code snippet) 顯示如何使用 Java RMI API 呼叫 Transformation Extender 對映：

清單 2：使用 RMI 的 Transformation Extender Java API snippet

```
import java.rmi.*;
import com.ibm.websphere.dtx.dtxpi.MConstants;
import com.ibm.websphere.dtx.dtxpi.rmi.*;

...

IMMap map = null;

// Initialize the RMI Connection
IMFactory factory = MRMIAPI.getFactory(hostname, port);

// Load the Map
map = factory.loadMapFile("Z:\\ test1.mmc");

// Run the Map
map.run();

// Check the return status
String responseMessage = map.getTextProperty(MConstants.MPIP_OBJECT_ERROR_MSG, 0);
int resultCode = map.getIntegerProperty(MConstants.MPIP_OBJECT_ERROR_CODE, 0);
System.out.println("Map status: " + szMsg + " (" + iRC + ")");

// Clean up
map.unload();
```

使用 Transformation Extender 與 Partner

Gateway

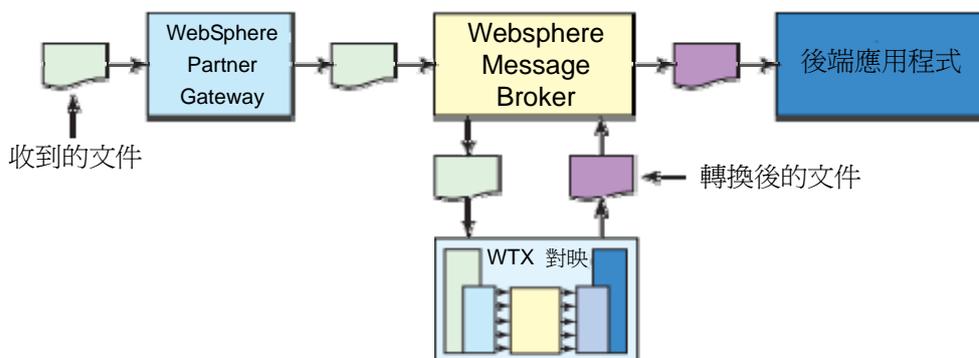
雖然本文提到兩種支援整合 WebSphere Transformation Extender 與 Partner Gateway 6.1 版的型樣，但其中只介紹第二種方法。

使用 WebSphere Message Broker

在本型樣中，會透過間接方式整合 Partner Gateway 與 Transformation Extender。Partner Gateway 傳遞需要轉換成 WebSphere Message Broker（以下稱爲 Message Broker）的文件，Transformation Extender 則會呼叫必要的 Transformation Extender 對

映以執行轉換，然後提供轉換後文件給目標後端應用程式。

圖 3：使用 WebSphere Message Broker



本文並未介紹此種整合方法。

使用 Partner Gateway "Action" Exit

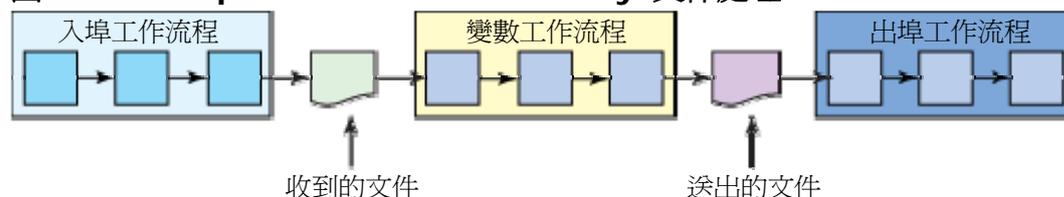
在此型樣中，Partner Gateway 會使用 Java JNI 或 Java RMI API 直接呼叫 Transformation Extender 對映。

爲了瞭解 Partner Gateway 如何呼叫 Transformation Extender，請研讀 WebSphere Partner Gateway Document Manager 的文件處理步驟說明，以便您以交易夥伴預期的格式與通訊協定，遞送文件給對方。

商業程序處理引擎會執行一系列的工作流程（包含許多步驟），其順序如下：

- 固定入埠工作流程在文件上執行通訊協定解開包裝、通訊協定處理，然後執行連線查閱。
- 變數工作流程在文件上執行動作，包括驗證、轉換及/或轉譯。
- 固定出埠工作流程執行通訊協定包裝，然後傳遞文件給交易夥伴。

圖 4：WebSphere Partner Gateway 文件處理



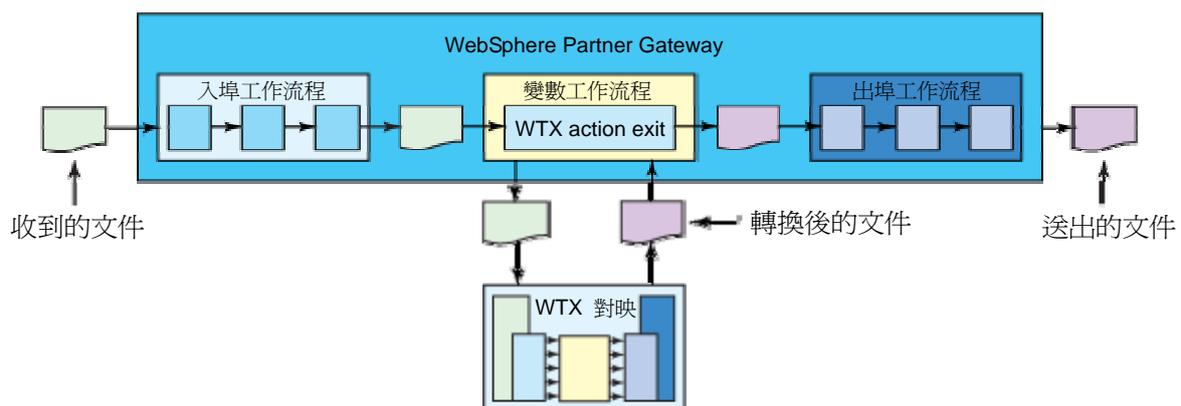
您可以使用 *Partner Gateway User Exit* 延伸每一種工作流程，此自訂 Java 片段可插入 Partner Gateway 以提供額外功能。

Transformation Extender 對映的呼叫要配合文件種類 — 驗證、轉換及 / 或轉譯，因此需要變動工作流程 *User Exit*，也稱之為 *Action Exit*。Action Exit 可完整存取收到的文件，並可以（若有需要）建立新的轉換後文件，然後傳遞至「出埠工作流程」，以便進行包裝並遞送給合作夥伴。關於開發 Partner Gateway User Exit，請詳見 WebSphere Partner Gateway Information Center（請參閱本文最後的「資源」部分）。

WebSphere Partner Gateway 開發團隊已開發出範例 Action User Exit，可使用 Transformation Extender Java JNI 介面或 Transformation Extender Java RMI 介面，以呼叫 Transformation Extender。您可以在 Partner Gateway 安裝 CD 中找到此範例。

圖 5（下方）顯示 WTX Action Exit 如何配合工作流程系列。

圖 5：使用 WTX Action Exit 的 WebSphere Partner Gateway 文件處理



本型樣容許 Partner Gateway 控制整份文件處理，以便將所需的配置數量降至最低，而且存放在單一位置。本文稍後將會詳細探討此型樣。

如何判斷該使用何種 Transformation Extender

Java API

首先，您必須決定 Action Exit 要使用何種 Transformation Extender API。這是很簡單的決策，因為只要根據 Transformation Extender/Partner Gateway 環境的配置即可決定。假如 Transformation Extender 與 Partner Gateway 位於不同的實體機器上，則可供使用的 API 只有 Transformation Extender Java RMI API。如果 Transformation Extender

與 Partner Gateway 位於相同的實體機器上，則任何 API 皆可使用，我們建議使用 Java™ JNI API，因為其速度快於 Java RMI API。

置換 Transformation Extender 卡片

對映編譯成執行檔後，輸入卡與輸出卡中的值將會設定做為預設值，然後在該對映被呼叫時供 Transformation Extender 的轉換與驗證引擎使用。為了讓 Partner Gateway 能夠使用對映來執行轉換與驗證，以下的值必須動態進行設定：

- 輸入卡 輸入檔檔名：以檔案的完整名稱做為對映的輸入。
- 輸出卡 輸出檔檔名：以檔案的完整名稱做為對映的輸出。

Transformation Extender Java API 容許您在執行時期置換「輸入卡」與「輸出卡」中的預設值。

在「Partner Gateway BPE 變數工作流程」進行處理期間，經過處理的文件將儲存在「Partner Gateway 共用檔案系統」目錄下的實體檔中。在處理期間使用 Partner Gateway API 所建立的任何新文件，也會在 Partner Gateway 共用檔案系統目錄下的實體檔中建立一份。每一次呼叫「Transformation Extender 動作處理程式」都是為了處理不同文件，因此有必要在執行時期置換 Transformation Extender 對映的輸入卡與輸出卡，以便指向 WPG 共用檔案系統目錄中的正確檔案。

圖 6：使用 WPG 共用檔案系統文件的 Transformation Extender 對映

WTX_Map_Root	c:/Maps
WTX_RMIServer_Port	2500
WTX_RMIServer_Hostname	localhost
WTX_RMIServer_Use_Server	yes

您可以使用清單 3 中的程式碼片段以搭配 Transformation Extender Java JNI API 與 Transformation Extender Java RMI API，以便在執行時期設定完整的輸入檔與輸出檔檔名，以指向 WPG 共用檔案系統中的檔案。

清單 3：置換 Transformation Extender 輸入卡與輸出卡

```
// Get the first Input Card
MCard inputCard = map.getInputCardObject(1);
```

```

// Get the Adapter for the Input Card
MAdapter inputAdapter = inputCard.getAdapter();

// Modify the Input Adapter Command Line
String xml = "<Adapter>" +
    "<CommandLine id=\"" + MConstants.MPIP_ADAPTER_COMMANDLINE + "\"
type=\"text\">" +
    inputFilename +
    "</CommandLine>" +
    "</Adapter>";

// Set the Adapter Command Line
inputAdapter.setPropertiesXML( xml );

// Get the first Output Card
MCard outputCard = map.getOutputCardObject(1);
MAdapter outputAdapter = outputCard.getAdapter();

// Modify the Output Adapter Command Line
xml = "<Adapter>" +
    "<CommandLine id=\"" + MConstants.MPIP_ADAPTER_COMMANDLINE +
    "\" type=\"text\">" +
    outputFilename +
    "</CommandLine>" +
    "</Adapter>";

// Set the Adapter Command Line
outputAdapter.setPropertiesXML( xml );

// Run the Map
map.run();

```

瞭解範例 WTX Action Exit

隨 Partner Gateway 安裝媒體出貨的範例，WTX Action Exit 擁有可延伸設計，對於所有希望使用 Transformation Extender Java JNI 或 Transformation Extender Java RMI API 的 Partner Gateway/Transformation Extender 整合而言，是絕佳的起始點。本文僅探

討論範例中，Transformation Extender 專用部分。您可以在 Partner Gateway 資訊中心尋找完整且詳細的範例說明。

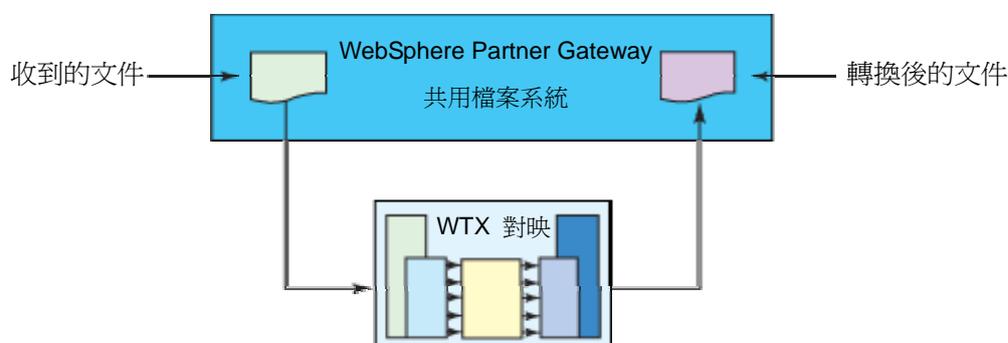
範例 WTX Action Exit 是由四個 Java 類別所組成：

- *WTXTransformationFactory*：這是由 Partner Gateway 呼叫的動作處理程式類別。其負責從 Partner Gateway 擷取動作處理程式與商業文件，然後呼叫 *WTXProcess* 以使用 Transformation Extender 進行轉換文件。
- *WTXProcess*：此類別負責判斷輸入檔與輸出檔的名稱與位置，以及 WTX 對映執行檔的名稱與位置，然後呼叫 *WTXProcessRMI* 類別或 *WTXProcessLocal* 類別，以呼叫 Transformation Extender 並且執行轉換。
- *WTXProcessRMI*：此類別負責使用 Transformation Extender Java RMI API 以呼叫 Transformation Extender。
- *WTXProcessLocal*：此類別負責使用 Transformation Extender Java JNI API 以呼叫 Transformation Extender。

無論有多少個不同對映，您在 Partner Gateway 中只需要配置一個 WTX Action Exit，因為 WTX Action Exit 與 Transformation Extender 對映之間是一對多的關係（單一的 WTX Action Exit 實例可以呼叫許多個不同的 Transformation Extender 對映）。

WTXProcess、*WTXProcessRMI* 與 *WTXProcessLocal* 類別所需的動作處理程式配置值，位於 Partner Gateway 主控台的「動作處理程式」本身設定（如圖 7 所示）。

圖 7：WTX 動作處理程式配置



WTXProcess 類別使用 *WTX_RMIServer_Use_Server* 旗標，以決定呼叫 *WTXProcessRMI* (yes) 類別或 *WTXProcessLocal* (no) 類別執行轉換。如

果選用 WTXProcessRMI 類別，則主機名稱 (*WTX_RMIServer_Hostname*) 與連接埠 (*WTX_RMIServer_Port*) 等值將會用來建立 Transformation Extender RMI 伺服器連線。JNI 與 RMI 連線都需要 WTX 對映執行檔的位置，該位置位於 *WTX_Map_Root* 配置參數中。對映專用配置值是由對映配置檔提供，您可以使用 Partner Gateway 主控台，將該檔案上傳至 Partner Gateway。可在對映配置檔中找到的值如下：

- *WTX_MAP_NAME*：此值指定與 *WTX_Map_Root* 配置值相關聯的對映執行檔名稱。
- *WTX_OUTPUT*：此值指定是否會從對映產生結果。如果呼叫的是僅執行驗證的對映，此值設定為 NONE (無)。

如果 WTX Action Exit 的實例正在處理文件，則它會執行下列步驟：

1. 從 *WTX 動作處理程式配置* 取得配置值。
 - a. *WTX_Map_Root*：此值負責指向可包含多個可執行對映檔的目錄。
 - b. *WTX_RMIServer_Hostname*：此值僅會在 WTX Action Exit 採用 RMI 模式執行時運用。
 - c. *WTX_RMIServer_Port*：此值僅會在 WTX Action Exit 採用 RMI 模式執行時運用。
2. 從 *對映配置檔* 取得所需的配置值。
 - a. *WTX_MAP_NAME*：此值用來識別特定對映。
3. 由 Transformation Extender 將執行的 WTX 對映，產生完整檔名。
4. 針對要從 *Partner Gateway 商業文件實例* 轉換的文件，取得該文件隸屬檔案的完整檔名。
5. 使用必要參數建立新的 WTX 對映。
 - a. *JNI*：使用將要呼叫之 WTX 對映的完整檔名，以建立新的 `com.ibm.websphere.dtx.dtxpi.MMap` 物件。
 - b. *RMI*：使用 Transformation Extender RMI Server 的主機名稱與連接埠建立新的 `com.ibm.websphere.dtx.dtxpi.rmi.IMFactory` 物件，然後使用將要呼叫之 Transformation Extender 對映的完整檔名來建立新的 `com.ibm.websphere.dtx.dtxpi.rmi.IMMap` 物件。
6. 修改對映的輸入卡，以使用將轉換文件所隸屬檔案的完整檔名。
 - a. 使用 `MCard inputCard = map.getInputCardObject(1)` 以取得對映的輸入卡。
 - b. 使用 `MAdapter inputAdapter = inputCard.getAdapter()` 以取得「輸入卡」的檔案配接器。

- c. 建立新的 XML 字串，包含將轉換文件所隸屬檔案的完整檔名。
 - d. 使用 `inputAdapter.setPropertiesXML(xml)`，在 XML 字串中設定檔案配接器的內容
7. 修改對映的「輸出卡」，以使用將轉換文件所隸屬檔案的完整檔名。
- a. 使用 `MCard outputCard = map.getOutputCardObject(1)` 以取得對映的輸出卡。
 - b. 使用 `MAdapter outputAdapter = outputCard.getAdapter()` 以取得輸出卡的檔案配接器。
 - c. 建立新的 XML 字串，包含將轉換文件所隸屬檔案的完整檔名。
 - d. 使用 `outputAdapter.setPropertiesXML(xml)`，在 XML 字串中設定檔案配接器的內容
8. 使用 `map.run()`；執行對映
9. 驗證對映回應訊息與結果碼，以確定轉換成功。
10. 如果轉換失敗，會傳回內含對映回應訊息與結果碼的錯誤。
11. 如果轉換成功，請將包含已轉換文件的檔案設定成要遞送給交易夥伴的文件。

範例 WTX Action Exit 功能完整，並隨範例 Transformation Extender Map 與輸入檔出貨，提供良好起點給試圖整合 Partner Gateway 與 Transformation Extender 的人士。

總結

本文簡介 Transformation Extender，並且說明 Partner Gateway 如何處理文件，從收到文件到將文件遞送至目的地。文中概述可用來整合 Partner Gateway 與 Transformation Extender 的型樣，同時深入介紹 Java JNI 與 RMI 型樣。您現在已獲得所需資訊，瞭解如何使用 Java JNI 或 Java RMI 型樣，以整合 Partner Gateway 與 Transformation Extender。

資源

- [參加討論區。](#)
- [WebSphere Partner Gateway 首頁](#)
- [WebSphere Business Integration 首頁 \(WebSphere Partner Gateway 的舊版\)](#)

- [位於資訊中心網站的 WebSphere Partner Gateway](#)
- [紅皮書：B2B Solutions using WebSphere Partner Gateway](#)
- [紅皮書：Secure Production Deployment of B2B Solutions using WebSphere Business Integration Connect](#)
- [IBM WebSphere Business Integration zone](#)
- [WebSphere Transformation Extender 產品網站](#)
- [WebSphere Transformation Extender 文件](#)

關於作者

Dave Mulley 是位 IT 專家及美國北卡羅來納州 Raleigh 當地的 IBM Software Services for WebSphere Enablement 團隊成員。Dave 專精於 WebSphere Business Integration 產品系列，他從 1999 年起開始為 IBM 效力。您可透過電子郵件聯絡 Dave：dmulley@us.ibm.com