

IBM® WebSphere™ Commerce



WebSphere Commerce Accelerator 自訂手冊

第 5.4 版

IBM® WebSphere™ Commerce



WebSphere Commerce Accelerator 自訂手冊

第 5.4 版

注意事項：

在使用本資訊及其支援的產品之前，請先閱讀第93頁的『注意事項』的一般資訊。

第一版（2002 年 1 月）。

本版適用於 IBM® WebSphere Commerce 5.4 版與其所有後續版次與修正層次，除非新版中另有提及。請確定您使用的是該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. 以電子郵件傳送至下列其中一個網路 ID。如果您希望獲得答覆，請務必註明您的完整網路位址。

Internet: torrcf@ca.ibm.com
IBMLink: [toribm\(torrcf\)](#)

2. 如果要使用傳真，請使用下列號碼：

美國以及加拿大：416-448-6161
其它國家：(+1)-416-448-6161

3. 郵件請寄到下列地址：

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

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 2001. All rights reserved.

目錄

開始之前	v
本書中使用的慣例	v
基本知識要求	v
本書的編排方式	v

第1篇 自訂 WebSphere Commerce Accelerator 1

第1章 商業關係管理 3
自訂範例 3
實務 1：新增新條款到「合約」使用者介面中 3
實務 2：移除「合約」使用者介面中的條款 9

第2章 「客戶服務代表」工具 11
自訂範例 11
實務 1：新增其它的客戶資料到商業客戶摘要對話框中 11
實務 2：在「客戶服務 - 訂單管理」介面中啓用購物車 14

第3章 共同工作區 19
自訂範例 19
實務 1：自訂工作區的外觀 19
實務 2：建立自訂佈景主題 19
實務 3：自訂電子郵件通知 21

第4章 客戶設定檔 23
範例程式碼 23
自訂範例 23
實務 1：新增屬性到客戶設定檔中 23

第5章 活動 31
直接將規則輸入到資料庫中 31
simpleCondition 元素 32
openCondition 元素 32
動作元素 33
基礎架構元素 34
規則套件 34
條件套件 34
Blaze 規則專案 35
自訂 36

第6章 型錄搜尋 39
自訂實務 39
實務 1：新增屬性到搜尋 Bean 中 39
實務 2：新增屬性到搜尋引擎中 41
實務 3：新增表格到搜尋引擎中 43
實務 4：新增多元屬性到搜尋 Bean 中 44
實務 5：藉由最佳化的摘要表格提升效能 45
型錄搜尋資料 Bean 變數 47

型錄搜尋資料庫直欄 50

第7章 優待券 51
自訂範例 51
實務 1：在建立優待券折扣時新增新資訊 51

第8章 折扣 53
自訂範例 53
實務 1：在建立折扣時新增其它資訊 53
實務 2：變更預設行爲 53

第9章 RFQ 回應 55
自訂範例 55
實務 1：複製回應 55
實務 2：將「已撤回」狀態從 RFQ 回應的循環週期中刪除 61
實務 3：在建立期間輸入回應的說明資訊 64

第10章 預期的庫存 69
自訂範例 69
實務 1：在建立預期庫存記錄時新增新資訊 69

第11章 商業智慧 71
動態環境定義 71
自訂範例 71
實務：新增新動作到現有環境定義中 71

第12章 報告組織架構的概觀 73
自訂報告組織架構 73
自訂範例 73
報告組織架構的指令 78
報告組織架構的物件模型 79
報告 JSP 檔的可重複使用元件 79
將 helper 用在報告輸入與輸出頁面上 83
撰寫一份利用可重複使用之 JSP 頁面元件的報告 85

第13章 產品顧問 87
自訂範例 87
實務 1：使用不同的運算子圖示 87
實務 2：「產品對照」虛擬狀態中的鏈結 87
實務 3：呈現「產品探索」的自訂 87

第14章 規則專案 89
如何架構以自訂規則專案爲基礎的規則服務 89
如何呼叫以自訂規則專案爲基礎的規則服務 89

第2篇 後記 91

注意事項 93
商標 94

開始之前

本書中使用的慣例

本書的使用慣例如下：

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

等寬字型表示您必須輸入完全相同的文字範例以及目錄路徑。

*斜體字型*用來強調，並代表可換成您所要之值的變數。



此圖示表示「要訣」-- 亦即提供其它資訊以協助您完成作業。

基本知識要求

如果要自訂 WebSphere Commerce Accelerator，您需要具備下列知識：

- HTML 以及 XML
- 結構化查詢語言（SQL）
- Java 程式設計

有關自訂 WebSphere Commerce 的進一步資訊，請參閱 WebSphere Commerce 程式設計手冊。本書可從下列網站中取得：

www.ibm.com/software/webservers/commerce/wcs_pro/lit-tech-general.html

本書的編排方式

下表列出本書的編排方式：

表 1.

元件	說明	位置
商業關係管理	本節詳述當您自訂「合約」工具時必須注意的部份	第3頁的『第1章 商業關係管理』
客戶服務代表	本節詳述當您自訂「客戶服務代表」工具時必須注意的部份	第11頁的『第2章 「客戶服務代表」工具』
客戶設定檔	本節詳述當您自訂「客戶設定檔」工具時必須注意的部份	第23頁的『第4章 客戶設定檔』
活動	本節詳述當您自訂「活動」工具時必須注意的部份	第31頁的『第5章 活動』
優待券	本節詳述當您自訂「優待券」工具時必須注意的部份	第51頁的『第7章 優待券』
折扣	本節詳述當您自訂「折扣」工具時必須注意的部份	第53頁的『第8章 折扣』

表 1. (繼續)

元件	說明	位置
RFQ	本節詳述當您自訂 RFQ 工具時必須注意的部份	第55頁的『第9章 RFQ 回應』
庫存	本節詳述當您自訂「庫存」工具時必須注意的部份	第69頁的『第10章 預期的庫存』
商業智慧	本節詳述當您自訂「報告」工具時必須注意的部份	第71頁的『第11章 商業智慧』
報告組織架構	本節詳述當您自訂「報告」工具時必須注意的部份	第73頁的『第12章 報告組織架構的概觀』
搜尋	本節詳述當您自訂「型錄搜尋」工具時必須注意的部份	第39頁的『第6章 型錄搜尋』
Blaze Rules Advisor	本節詳述當您自訂 Brokat Advisor 規則專案時必須注意的部份	第89頁的『第14章 規則專案』
產品顧問	本節詳述當您自訂「產品顧問」工具時必須注意的部份	第87頁的『第13章 產品顧問』

第1篇 自訂 WebSphere Commerce Accelerator

本書說明如何自訂 WebSphere Commerce Accelerator。本書會藉由提供 WebSphere Commerce Accelerator 設計決策的背景知識，讓您瞭解如何進行自訂，以及自訂的詳細必要步驟。

本書旨在教您如何自訂 WebSphere Commerce Accelerator，而不會詳盡列出所有可能的自訂方式。就本書的結構而言，會介紹構成 WebSphere Commerce Accelerator 的不同通用部份，並說明如何自訂這些部份。

WebSphere Commerce Accelerator 為各種工具的集合，這些工具有助您進行網站每日的商業運作。亦即，其主要做為商業人員的介面，以建立與修改網站運作的各種不同層面，而不必一再聯絡負責網站運作的 IT 人員。因此 WebSphere Commerce Accelerator 中提供眾多元件，而每一個各鎖定一個重要的商業層面。我們所做的每一份努力，都希望這些工具能夠通用，但在這個相同目標下，也意味著某些使用者可能會發現某些需求無法符合。如果您是在建立網站期間擴充資料庫，且這些擴充和本文件所列的任何工具有關，則 Accelerator 將無法發現這些資料，直到您自訂相關工具為止。

本書說明如何自訂 WebSphere Commerce Accelerator 中的下列元件。

第1章 商業關係管理

本章中所要討論的元素，代表 WebSphere Commerce Accelerator 中之「帳戶與合約」元件的使用者介面。這些元素有助於建立與維護業務經理或帳戶代表在執行每日所負責之相關商業工作時，所需的帳戶、合約與其它動作。「商業關係管理」元件所含的元素如下：

- 「帳戶」筆記本
- 「合約」筆記本

自訂範例

下列範例說明如何自訂 WebSphere Commerce Accelerator 中的這個部份。

實務 1：新增新條款到「合約」使用者介面中

施行概觀

爲了讓 WebSphere Commerce Accelerator 更加通用，並盡可能讓更多人覺得好用，在此省略了某些條款。因此，您或許會發現必須新增條款到網站中，使其更切合您的企業。本實務是以必要步驟引導您新增其它條款到您的網站中。

自訂步驟

1. 將條款定義在伺服器中。您必須已完成下列各項，才能自訂使用者介面：
 - 在 B2BTrading.dtd 檔中定義一項條款。有關完整詳述，請參閱 *IBM WebSphere Commerce 程式設計手冊* 第 7 章。
 - 爲新條款建立一個 Enterprise Bean 或存取 Bean。
2. 新增條款到使用者介面中。如果要新增必要元素到使用者介面中，請執行下列步驟：
 - a. 針對條款的使用者介面頁面，建立一個 JSP 檔。此頁面必須包含一個動態 JavaScript 物件，以便從頁面中的輸入欄位擷取資料。建立一個資料 Bean，其中含括了載入存取 Bean 中的資料。您可以選擇性地將存取 Bean 直接含在頁面中，以便從資料庫載入條款。contractId 參數會傳到所有頁面中，而有助於縮短資料載入效能。此 JSP 檔的範例如下：

```
<!--=====
/*-----
/* 此處所含的範例是按 "原樣" 提供給您。
/*
/* 在此，IBM 所提供的是一個簡單的範例，而未徹底在所有條件下
/* 進行測試。因此 IBM 不能保證其可靠性、可用性或功能性。
/*
/* 為了盡可能詳述概念，本範例中可能涵蓋個人、公司、
/* 品牌及產品的名稱。這些名稱全部都是虛構的，如果與
/* 真實企業或個人的名稱和地址雷同，純屬巧合。
/*-----
/*
=====>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<%@page language="JAVA"
import="com.ibm.commerce.tools.util.UIUtil,
com.ibm.commerce.beans.DataBeanManager,
```

```

com.ibm.commerce.tools.contract.beans.MemberDataBean,
com.ibm.commerce.tools.contract.beans.MyTCDataBean,
com.ibm.commerce.tools.contract.beans.PolicyDataBean,
com.ibm.commerce.tools.contract.beans.PolicyListDataBean"
%>

<%@include file="../common/common.jsp" %>
<%@include file="ContractCommon.jsp" %>

<HTML>

<HEAD>
  <%= fHeader %>
  <LINK rel="stylesheet" href="<%= UIUtil.getCSSFile(fLocale) %>"
  type="text/css">

  <TITLE><%= contractsRB.get("MyTCHeading") %></TITLE>
  <SCRIPT LANGUAGE="JavaScript" SRC="/wcs/javascript/tools/common/Util.js">
</SCRIPT>
  <SCRIPT LANGUAGE="JavaScript" SRC="/wcs/javascript/tools/contract/
ContractUtil.js"></SCRIPT>
  <SCRIPT LANGUAGE="JavaScript" SRC="/wcs/javascript/tools/contract/
Extensions.js"></SCRIPT>

<SCRIPT LANGUAGE="JavaScript">

var MyTCModel;

////////////////////////////////////
// LOAD-SAVE-VALIDATE SCRIPTS
////////////////////////////////////
function onLoad() {

    if (parent.setContentFrameLoaded) {
        parent.setContentFrameLoaded(true);
    }

    // 檢查看模型是否已經載入
    var isModelLoaded = parent.get("ContractMyTCModelLoaded", null);

    if (isModelLoaded) {
        // 回到此頁面，並從模型重新載入
        // 取得模型
        MyTCModel = parent.get("ContractMyTCModel", null);
    }
    else {
        // 先探訪此頁面，並建立模型

        // 建立模型以儲存 MyTC 資料
        MyTCModel = new ContractMyTCModel();

        // 保留模型
        parent.put("ContractMyTCModel", MyTCModel);
        parent.put("ContractMyTCModelLoaded", true);

    }

    var myTCPolicyList = new Array();
    <%
    try {
    // 載入資料庫中的所有原則
    PolicyListDataBean policyList = new PolicyListDataBean();
    PolicyDataBean policy[] = null;

    policyList.setPolicyType(policyList.TYPE_PRODUCT_SET);
    DataBeanManager.activate(policyList, request);
    policy = policyList.getPolicyList();
    </%

```

```

for (int i = 0; i < policy.length; i++) {
MemberDataBean mdb = new MemberDataBean();
mdb.setId(policy[i].getStoreMemberId());
DataBeanManager.activate(mdb, request);
%>
myTcPolicyList[myTcPolicyList.length] =
new PolicyObject('<%=UIUtil.toJavaScript(policy[i].getShortDescription())%>',
'<%= policy[i].getPolicyName() %>',
'<%= policy[i].getId() %>',
'<%= policy[i].getStoreIdentity() %>',
new Member('<%= mdb.getMemberType() %>',
'<%= mdb.getMemberDN() %>',
'<%= mdb.getMemberGroupName() %>',
'<%= mdb.getMemberGroupOwnerMemberType() %>',
'<%= mdb.getMemberGroupOwnerMemberDN() %>'
);
<%
}
} catch (Exception e) {}
%>
MyTCModel.policyList = myTcPolicyList;

// 檢視此項是否是合約的更新版
if (<%= foundContractId %> == true) {
// 載入資料 Bean 中的資料
<%
if (foundContractId) {
MyTCDataBean tc = new MyTCDataBean(new Long(contractId));
DataBeanManager.activate(tc, request);
if (tc.getHasMyTC()) {
%>
MyTCModel.attr1 = '<%= UIUtil.toJavaScript((String)tc.getAttr1()) %>';
MyTCModel.attr2 = '<%= UIUtil.toJavaScript((String)tc.getAttr2()) %>';
MyTCModel.tcReferenceNumber = '<%= tc.getReferenceNumber() %>';
MyTCModel.policyReferenceNumber = '<%= tc.getPolicyReferenceNumber() %>';
for (var i = 0; i < myTcPolicyList.length; i++) {
if (myTcPolicyList[i].policyId == '<%= tc.getPolicyReferenceNumber() %>') {
MyTCModel.selectedPolicyIndex = i;
}
}
<%
}
}
%>
}

loadPanelData();

// 回到驗證頁面處理錯誤訊息
if (parent.get("attr1Empty", false))
{
parent.remove("attr1Empty");
alertDialog("<%= UIUtil.toJavaScript((String)contractsRB.
get("attr1Empty"))%>");
}
else if (parent.get("attr2Empty", false))
{
parent.remove("attr2Empty");
alertDialog("<%= UIUtil.toJavaScript((String)contractsRB.
get("attr2Empty"))%>");
}
else if (parent.get("attr1TooLong", false))
{
parent.remove("attr1TooLong");
}
}

```

```

        alertDialog("<%= UIUtil.toJavaScript((String)contractsRB.
get("attr1TooLong"))%>");
    }
    else if (parent.get("attr2TooLong", false))
    {
        parent.remove("attr2TooLong");
        alertDialog("<%= UIUtil.toJavaScript((String)contractsRB.
get("attr2TooLong"))%>");
    }

    return;
}

function loadPanelData() {

// 設定輸入欄位
document.MyTCForm.Attr1.value = MyTCModel.attr1;
document.MyTCForm.Attr2.value = MyTCModel.attr2;

// 載入原則
for (var i = 0; i < MyTCModel.policyList.length; i++) {
    if (MyTCModel.selectedPolicyIndex == i) {
        document.MyTCForm.PolicyList.options[i] = new Option(MyTCModel.policyList[i].
displayText,
i, true, true);
    } else {
        document.MyTCForm.PolicyList.options[i] = new Option(MyTCModel.policyList[i].
displayText,
i, false, false);
    }
}
}

function savePanelData() {
    MyTCModel.attr1 = document.MyTCForm.Attr1.value;
    MyTCModel.attr2 = document.MyTCForm.Attr2.value;
    MyTCModel.selectedPolicyIndex = document.MyTCForm.PolicyList.selectedIndex;
}
</SCRIPT>

</HEAD>

<!--
////////////////////////////////////
// HTML 區段
////////////////////////////////////
-->

<BODY onLoad="onLoad()" class="content">

    <H1>
    <%= contractsRB.get("MyTCHeading") %>
    </H1>

    <FORM NAME="MyTCForm">

        <%= contractsRB.get("MyTCAttr1Label") %>
        <BR>
        <INPUT type="text" name="Attr1" value="" size="10" maxlength="10">
        <BR>

        <%= contractsRB.get("MyTCAttr2Label") %>
        <BR>
        <INPUT type="text" name="Attr2" value="" size="10" maxlength="10">
        <BR>

        <%= contractsRB.get("MyTCPolicyLabel") %>

```

```

        <BR>
        <SELECT NAME="PolicyList" SIZE="1">
        </SELECT>

</FORM>

</BODY>
</HTML>

```

- b. 針對條款的使用者介面頁面，建立一個 JavaScript 檔。建立函數以驗證與提交 JavaScript 資料。當您提交資料時，您必須為新條款建立一個符合 XML 格式的新 JavaScript 物件。此 JavaScript 檔的範例如下：

```

/*-----
/* 此處所言之範例是按 "原樣" 提供給您。
/*
/* 在此，IBM 所提供的是一個簡單的範例，而未徹底在所有條件下
/* 進行測試。因此 IBM 不能保證其可靠性、可用性或功能性。
/*
/* 為了盡可能詳述概念，本範例中可能涵蓋個人、公司、
/* 品牌及產品的名稱。這些名稱全部都是虛構的，如果與
/* 真實企業或個人的名稱和地址雷同，純屬巧合。
/*-----
/*

function ContractMyTCModel() {

    this.tcReferenceNumber = "";
    this.policyReferenceNumber = "";

    this.attr1 = "";
    this.attr2 = "";

    this.policyList = new Array();
    this.selectedPolicyIndex = "0";
}

function validateMyTCPanel() {

    var tcModel = get("ContractMyTCModel");
    if (tcModel != null) {
    // 檢查 attr1 是否為空的或太長
    if (!tcModel.attr1)
    {
        put("attr1Empty", true);
        gotoPanel("MyTCHeading");
        return false;
    }

    if (!isValidUTF8length(tcModel.attr1, 10))
    {
        put("attr1TooLong", true);
        gotoPanel("MyTCHeading");
        return false;
    }
    // 檢查 attr2 是否為空的或太長
    if (!tcModel.attr2)
    {
        put("attr2Empty", true);
        gotoPanel("MyTCHeading");
        return false;
    }

    if (!isValidUTF8length(tcModel.attr2, 10))
    {

```

```

        put("attr2TooLong", true);
        gotoPanel("MyTCHeading");
        return false;
    }
}

function submitMyTC(termsAndConditions) {

    var tcModel = get("ContractMyTCModel");

    if (tcModel != null) {

        var myTC = new Object();
        myTC.MyTC = new Object();
        myTC.MyTC.MySubTC = new Object();
        myTC.MyTC.MySubTC.attr1 = tcModel.attr1;
        myTC.MyTC.MySubTC.attr2 = tcModel.attr2;

        myTC.MyTC.PolicyReference = new Object();
        myTC.MyTC.PolicyReference.policyName = tcModel.policyList[tcModel.
        selectedPolicyIndex].policyName;
        myTC.MyTC.PolicyReference.policyType = "ProductSet";
        myTC.MyTC.PolicyReference.storeIdentity = tcModel.policyList[tcModel.
        selectedPolicyIndex].storeIdentity;
        myTC.MyTC.PolicyReference.Member = tcModel.policyList[tcModel.
        selectedPolicyIndex].member;

        if (tcModel.tcReferenceNumber != "") {
            // 變更條款
            myTC.action = "update";
            myTC.referenceNumber = tcModel.tcReferenceNumber;
        }
        else {
            // 建立新條款
            myTC.action = "new";
        }

        termsAndConditions[termsAndConditions.length] = myTC;
    }

    return true;
}

function validateContractExtensions() {
    if (this.validateMyTCTPanel) {
        if (validateMyTCTPanel() == false) {
            return false;
        }
    }
    return true;
}

function submitContractExtensions(termsAndConditions) {
    if (this.submitMyTC) {
        if (submitMyTC(termsAndConditions) == false) {
            return false;
        }
    }
}

```

- c. 修改 Contract.js 檔，以呼叫步驟 2b 中所建的新 submit 與 validate 函數。
- d. 修改資源連結，以新增您所要的新字串。
- e. 修改 ContractNotebook.xml，以新增新畫面。

實務 2：移除「合約」使用者介面中的條款

施行概觀

如同您可能會發現某些條款遺漏，某些條款對您的企業而言或許不必要。本實務是以必要步驟引導您將多餘的條款從您網站中移除。

自訂步驟

修改 `ContractNotebook.xml` 檔，以移除內含您想移除之條款的畫面元素的區段。在下列目錄中找出此檔案：

 AIX

`/usr/WebSphere/CommerceServer/xml/tools/contract/`

 400

`/QIBM/UserData/WebSphere/CommerceServer/CommerceServer/xml/tools/contract/`

 Linux `/opt/WebSphere/CommerceServer/xml/tools/contract/`

 Solaris `/opt/WebSphere/CommerceServer/xml/tools/contract/`

 Windows `drive:\WebSphere\CommerceServer\xml\tools\contract\`

第2章 「客戶服務代表」工具

本章中所要討論的元素，代表 WebSphere Commerce Accelerator 中之「客戶服務代表」工具的使用者介面。這些元素有助於建立與維護 CSR 在執行每日所負責之相關商業工作時，所需的客戶、訂單與其它動作。「CSR 工具」元件所含的元素如下：

- 「客戶」精靈
- 「客戶」筆記本
- 「訂單」精靈
- 「訂單」筆記本

自訂範例

下列範例說明如何自訂 WebSphere Commerce Accelerator 中的 CSR 工具。






實務 1：新增其它的客戶資料到商業客戶摘要對話框中

施行概觀

本範例說明如何新增商業客戶的員工編號到商業客戶摘要對話框中。

自訂步驟

延伸 ShopperSummaryB2BDialog.jsp。在下列目錄中找出此檔案：

-  /usr/WebSphere/CommerceServer/web/tools/csr/
-  /QIBM/UserData/WebSphere/CommerceServer/CommerceServer/web/tools/csr/
-  /opt/WebSphere/CommerceServer/web/tools/csr/
-  /opt/WebSphere/CommerceServer/web/tools/csr/
-  drive:\WebSphere\CommerceServer\web\tools\csr\

查看以下的程式碼範例，以瞭解如何變更 JSP 檔，使其在摘要對話框中顯示員工編號。員工編號屬於 OptoolsRegisterDataBean 中的內容。本範例並未使用內容檔來儲存 Employee Number（員工編號）標籤。以下是會讓員工編號顯示在客戶摘要對話框中的範例輸出：

```
<%--
//*****
//*-----
//* Licensed Materials - Property of IBM
//*
//* 5724-A18
//*
//* (c) Copyright IBM Corp. 2001
//*
//* US Government Users Restricted Rights - Use, duplication or
//* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
//*
//*-----
//*
--%>
```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
:
:
: 此處省略部份程式碼
:
:
<HTML>

<HEAD>
<LINK rel=stylesheet
      href="<%= UIUtil.getCSSFile(cmdContext.getLocale()) %>"
      type="text/css">
<SCRIPT SRC="/wcs/javascript/tools/common/Util.js"></SCRIPT>

<script>
<%@ include file = "SummaryDisplay.jsp" %>
:
:
: 此處省略部份程式碼
:
:
</script>

</HEAD>
<BODY CLASS=content onLoad = "initializeState();">
<p>
</p>
<FORM NAME="profile" action="" Method="POST">

      <INPUT type="hidden" name="logonId" value="">
      <INPUT type="hidden" name="XML" value="">
      <INPUT type="hidden" name="URL" value="">

<h1><%= userNLS.get("customerSummaryTitle") %></h1>
<P><B><%=userNLS.get("generalHeader")%></B>
<BR>
      <%=userNLS.get("logonid")%>
      <I><%=UIUtil.toHTML(registerDataBean.getLogonId()) %></I>
<BR>
      <%=userNLS.get("custName")%>:
      <I><script>displayNameSummary()</script></I>
<BR>
      Employee Number:
      <I><%=UIUtil.toHTML(registerDataBean.getEmployeeId()) %></I>
<BR>
      <%=userNLS.get("orgAccount")%>:
<% if (accountDBean != null) { %>
      <I><%=UIUtil.toHTML(accountDBean.getAccountName()) %></I>
<% } %>
<BR>
      <%=userNLS.get("challengeQuestion")%>:
      <I><%=UIUtil.toHTML(registerDataBean.getChallengeQuestion())%></I>
<BR>
      <%=userNLS.get("challengeAnswer")%>:
      <I><%=UIUtil.toJavaScript(registerDataBean.getChallengeAnswer())%></I>
<BR>
      <%=userNLS.get("clientCertificate")%>:
      <I>
<% if (certStatus != "") { %>
      <% if (certStatus == "V") {%><%=userNLS.get("valid")%><%}%>
      <% if (certStatus == "E") {%><%=userNLS.get("expired")%><%}%>
      <% if (certStatus == "R") {%><%=userNLS.get("revoked")%><%}%>
<% } else { %>
      <%=userNLS.get("noCertificate")%>
<% } %>
      </I>

```

```

<BR>
  <%=userNLS.get("status")%>:
  <I>
  <% if (userRegistry.getStatus().equals("1")) {%>
    <%=userNLS.get("accountStatusEnabled")%>
  <% } else { %>
    <%=userNLS.get("accountStatusDisabled")%>
  <% } %>
  </I>
<BR>
<P><B><%=userNLS.get("contactHeader")%></B>
  <TABLE border="0" CELLPADDING=0 CELLSPACING=0 >
    <TR valign="top">
      <TD><%=userNLS.get("address")%>:</TD>
      <TD><I><script>displayAddrSummary(0)</script></I></TD>
    </TR>
    <TR valign="top">
      <TD></TD>
      <TD><I><script>displayAddrSummary(1)</script></I></TD>
    </TR>
    <TR valign="top">
      <TD></TD>
      <TD><I><script>displayAddrSummary(2)</script></I></TD>
    </TR>
    <TR valign="top">
      <TD></TD>
      <TD><I><script>displayAddrSummary(3)</script></I></TD>
    </TR>
    <TR valign="top">
      <TD></TD>
      <TD><I><script>displayAddrSummary(4)</script></I></TD>
    </TR>
  </TABLE>
  <%=userNLS.get("phone")%>:
  <I><%=UIUtil.toHTML(address.getPhone1())%></I>
<BR>
  <%=userNLS.get("fax")%>:
  <I><%=UIUtil.toHTML(address.getFax1())%></I>
<BR>
  <%=userNLS.get("email")%>:
  <I><%=UIUtil.toHTML(address.getEmail1())%></I>
<BR>
<P><B><%=userNLS.get("orgHeader")%></B>
<BR>
  <TABLE border="0" CELLPADDING=0 CELLSPACING=0 >
    <TR valign="top">
      <TD><%=orgEntityNLS.get("OrgEntityDeliveryDescription")%>:</TD>
      <TD>
        <% if (orgEntity != null) { %>
          <I><%=UIUtil.toHTML(orgEntity.getDescription())%></I>
        <% } %>
      </TD>
    </TR>
  </TABLE>
  <TABLE border="0" CELLPADDING=0 CELLSPACING=0 >
    <TR valign="top">
      <TD><%=orgEntityNLS.get("OrgEntityGeneralBusCat")%>:</TD>
      <TD>
        <% if (orgEntity != null) { %>
          <I><%=UIUtil.toHTML(orgEntity.getBusinessCategory())%></I>
        <% } %>
      </TD>
    </TR>
  </TABLE>
</FORM>
<%

```

```

}
catch(Exception e)
{
    e.printStackTrace();
}
}

%>

</BODY>
</HTML>

```

實務 2：在「客戶服務 - 訂單管理」介面中啟用購物車

施行概觀

如果商店設計者想使用 CSR 工具來管理處於 P 狀態下的購物者訂單，則其必須修改 **CSROrderSearchB2B** 與 **CSROrderSearchB2C** 檢視畫面。

自訂步驟

延伸 CSROrderSearchB2B.jsp 與 CSROrderSearchB2C.jsp JSP 檔。藉由在訂單狀態準則中新增 Pending（擱置）選項，可讓 CSR 搜尋客戶的擱置訂單並加以管理。擱置訂單是指位於購物車中的訂單。這些檔案可在下列目錄中找到：

▶ AIX

/usr/WebSphere/CommerceServer/web/tools/order/

▶ 400

/QIBM/UserData/WebSphere/CommerceServer/CommerceServer/web/tools/order/

▶ Linux

/opt/WebSphere/CommerceServer/web/tools/order/

▶ Solaris

/opt/WebSphere/CommerceServer/web/tools/order/

▶ Windows

drive:\WebSphere\CommerceServer\web\tools\order\

其程式碼範例如下：

```

<%--
//*****
//*-----
//* Licensed Materials - Property of IBM
//*
//* 5724-A18
//*
//* (c) Copyright IBM Corp. 2001
//*
//* US Government Users Restricted Rights - Use, duplication or
//* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
//*
//*-----
//* --%>
<%@ page language="java" %>
<%@ page import="java.util.*" %>
<%@ page import="com.ibm.commerce.tools.util.*" %>
<%@ page import="com.ibm.commerce.command.CommandContext" %>

```

```

<%@ page import="com.ibm.commerce.server.*" %>
<%@ page import="com.ibm.commerce.tools.optools.user.beans.*" %>
<%@ page import="com.ibm.commerce.tools.optools.order.commands.*" %>
<%@ page import="com.ibm.commerce.tools.contract.beans.*" %>
<%@ page import="com.ibm.commerce.beans.*" %>
<%@ page import="com.ibm.commerce.tools.util.UIUtil" %>
<%@include file="../common/common.jsp" %>

<%! public String getUserLogon(String customerId, HttpServletRequest request) {
    try {
        if (customerId != null && !customerId.equals("")) {
            OptoolsRegisterDataBean userBean = new OptoolsRegisterDataBean();
            userBean.setUserId(customerId);
            DataBeanManager.activate(userBean, request);

            if (userBean.getLogonId() != null)
                return userBean.getLogonId();
        }
    } catch (Exception ex) {
        return "";
    }
    return "";
}
%>

<HTML>
<HEAD>
<%
    // 取得要顯示的資源連結
    CommandContext cmdContextLocale =
        (CommandContext)request.getAttribute(com.ibm.commerce.server.ECConstants.
    EC_COMMANDCONTEXT);
    Locale jLocale = cmdContextLocale.getLocale();
    Hashtable orderLabels =
        (Hashtable)ResourceDirectory.lookup("order.orderLabels", jLocale);

    // 擷取 request 參數
    JSPHelper jspHelp = new JSPHelper(request);
    String customerId =
        jspHelp.getParameter(ECOptoolsConstants.EC_OPTOOL_CUSTOMER_ID);

    if (customerId == null) {
        customerId = "";
    }

    //撰寫整個帳戶名稱清單
    AccountListDataBean acctListDB = new AccountListDataBean();
    DataBeanManager.activate(acctListDB, request);
    AccountDataBean[] acctList = acctListDB.getAccountList();

%>
<link rel=stylesheet
    href="<%= UIUtil.getCSSFile(jLocale) %>" type="text/css">
<TITLE></TITLE>
<SCRIPT SRC="/wcs/javascript/tools/common/Util.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
<!-- 舊瀏覽器將看不到 Script

function initializeState()
{
    parent.setContentFrameLoaded(true);
}

function savePanelData()
{
}

```

```

function onLoad() {
    initializeState()
}

function isEmpty(id) {
    return !id.match(/[^\s]/);
}

function isNumber(word)
{
    var numbers="0123456789";
    for (var i=0; i < word.length; i++)
    {
        if (numbers.indexOf(word.charAt(i)) == -1)
            return false;
    }
    return true;
}

function formValid() {
    var invalidChars = /[.,;&#%|'"\ \\/]/
    if (document.orderFindForm.orderId.value.match(invalidChars) ||
        document.orderFindForm.userLogon.value.match(invalidChars))
    {
        // alert(",;is not valid");
        alertDialog("<%=UIUtil.toJavaScript(orderLabels.get
            ("invalidChars").toString()) %>");
        return false;
    }
    return true;
}

function validateEntries()
{
    if (isEmpty(document.orderFindForm.orderId.value)
        && isEmpty(document.orderFindForm.userLogon.value)
        && (document.orderFindForm.orderState.value == "all")
        && isEmpty(document.orderFindForm.accountId.value)) {
        alertDialog("<%=UIUtil.toJavaScript((String)orderLabels.
get("findDialogNoCriteria"))%>');
        return false;
    } else

    if (!isEmpty(document.orderFindForm.orderId.value)) {
        if (!isNumber(document.orderFindForm.orderId.value)) {
            alertDialog ('<%=UIUtil.toJavaScript((String)orderLabels.get
                ("findDialogInvalidNumber"))%>');
            return false;
        }
    } else if (!formValid()) {
        return false;
    }

    return true;
}

function findAction() {
    if (validateEntries() == true) {
        url = '/webapp/wcs/tools/servlet/NewDynamicListView';
        var urlPara = new Object();
        urlPara.listsize='22';
        urlPara.startindex='0';
        if ("<%=customerId%>" != "") {
            urlPara.ActionXMLFile='order.csadminOrderListB2B';
        } else {
            urlPara.ActionXMLFile='order.csOrderListB2B';
        }
        urlPara.cmd='OrderListViewB2B';
        urlPara.orderId=document.orderFindForm.orderId.value;
    }
}

```



```

urlPara.userLogon=document.orderFindForm.userLogon.value;
urlPara.accountId=document.orderFindForm.accountId.value;
urlPara.orderType=document.orderFindForm.orderState.value;
urlPara.orderby='orderid';

top.setContent("<%= UIUtil.toJavaScript((String)orderLabels.get
(\"findResultBCT\") %>\",url,true, urlPara);
return true;
}
return false;
}
function cancelAction() {
top.goBack();
}
// -->
</SCRIPT>
</HEAD>
<BODY CLASS=content ONLOAD="initializeState();">
<H1><%=orderLabels.get("findDialog")%></H1>
<P><%=orderLabels.get("findCSOrderInst")%>
<FORM NAME="orderFindForm">
<TABLE>
<TBODY>
<TR>
<TD><%=orderLabels.get("orderNumber")%></TD>
</TR>
<TR>
<TD><INPUT size="9" type="text" maxlength="9" name="orderId"></TD>
</TR>
<TR>
<TD></TD>
</TR>
<TR>
<TD><%=orderLabels.get("customerName")%></TD>
</TR>
<TR>
<TD><INPUT size="31" type="text" maxlength="31" name="userLogon"
value="<%=getUserLogon(customerId, request)%>"></TD>
</TR>
<TR>
<TD></TD>
</TR>
<TR>
<TD><%= orderLabels.get("orderStatus") %></TD>
</TR>
<TR>
<TD>
<SELECT name="orderState">
<OPTION value="all"></OPTION>
<OPTION value="P"><%= orderLabels.get("P") %></OPTION>
<OPTION value="I"><%= orderLabels.get("I") %></OPTION>
<OPTION value="W"><%= orderLabels.get("W") %></OPTION>
<OPTION value="N"><%= orderLabels.get("N") %></OPTION>
<OPTION value="M"><%= orderLabels.get("M") %></OPTION>
<OPTION value="B"><%= orderLabels.get("B") %></OPTION>
<OPTION value="C"><%= orderLabels.get("C") %></OPTION>
<OPTION value="E"><%= orderLabels.get("E") %></OPTION>
<OPTION value="R"><%= orderLabels.get("R") %></OPTION>
<OPTION value="S"><%= orderLabels.get("S") %></OPTION>
<OPTION value="D"><%= orderLabels.get("D") %></OPTION>
<OPTION value="L"><%= orderLabels.get("L") %></OPTION>
<OPTION value="T"><%= orderLabels.get("T") %></OPTION>
<OPTION value="A"><%= orderLabels.get("A") %></OPTION>
<OPTION value="F"><%= orderLabels.get("F") %></OPTION>
<OPTION value="G"><%= orderLabels.get("G") %></OPTION>
<OPTION value="X"><%= orderLabels.get("X") %></OPTION>
</SELECT>

```

```

        </TD>
    </TR>
    <TR>
        <TD></TD>
    </TR>
    <TR>
        <TD><%= orderLabels.get("accountName") %></TD>
    </TR>
    <TR>
        <TD>
            <SELECT name="accountId">
                <OPTION value=""></OPTION>
                <% if (acctList != null) {
                    for (int i=0; i<acctList.length; i++) { %>
                        <OPTION value="<%=acctList[i].getAccountId()%>"
                            <%=acctList[i].getAccountName()%></OPTION>
                    }
                } %>
            </SELECT>
        </TD>
    </TR>
    <TR>
        <TD></TD>
    </TR>
</TBODY>
</TABLE>
</FORM>
<SCRIPT LANGUAGE="JavaScript">
<!--
//For IE if (document.all) {
    onLoad();
}
//-->
</SCRIPT>

</BODY>
</HTML>

```

第3章 共同工作區

本章中所要討論的元素會呈現一個個案研究，其中詳述自訂 QuickPlace 以及建立 WebSphere Commerce 所附「工具屋」範本所需的步驟。此外，亦會提供必要步驟，來說明當使用者受邀加入共同工作區時，該如何自訂所要傳送的電子郵件內容。

自訂範例

下列範例說明如何自訂 WebSphere Commerce Accelerator 中的這個部份。

註：實務 1 與 2 密切相關。如果您執行實務 1，您通常還需執行實務 2。

實務 1：自訂工作區的外觀

施行概觀

您要為「共同工作區」特性自訂的第一件事可能是所建共同工作區的“外觀”。如果要自訂外觀，您必須先建立一個名為 PlaceType 的範本。有關自訂的詳述，請參閱自訂 QuickPlace 紅皮書。

自訂步驟

本例使用 WebSphere Commerce 5.4 中所附的「工具屋」範例商店。本例說明如何建立 QuickPlace：

- 執行下列步驟以建立預設 QuickPlace：
 - 確定 QuickPlace 伺服器正在執行。開啓瀏覽器並前往 `http://qp_server/quickplace`，然後登入成 QuickPlace 管理者。
 - 按一下**建立 QuickPlace**。
 - 選擇團隊的標準 **QuickPlace**。填妥適當欄位，並按一下**下一步**。
 - 以建立者登入。
- 更新 QuickPlace 的歡迎頁面。按一下**編輯**。進行所要的變更，並按一下**公佈**。這會更新歡迎頁面。
- 執行下列步驟，變更 QuickPlace 的標誌：
 - 按一下目錄中的**自訂**。
 - 按一下**基本項**。
 - 按一下**變更基本項**。
 - 使用**簡單文字**、**標誌製作器**或**上傳標誌圖檔**，以設置標誌。「工具屋」範例是採用**上傳標誌圖檔**。此時標誌已更新。

實務 2：建立自訂佈景主題

施行概觀

佈景主題用以控制 QuickPlace 的「外觀」。本例是針對 QuickPlace 的佈置與樣式建立一個超文字標記語言 (HTML) 頁面與階層樣式表 (CSS)。有關如何撰寫這些檔案的進一步資訊，請參閱自訂 QuickPlace 紅皮書。

注意事項：

由於成員管理是透過 WebSphere Commerce 來管理，請移除目錄中的 **Members** 鏈結。如果要移除此鏈結，請在插入目錄時於頁面佈置檔中使用下列程式碼：

```
<QuickPlaceSkinComponent
  name=TOC
  Format={
    <tr>
    <td class=h-toc-text><br></td>
    <td class=h-toc-text><Item class=h-toc-text></td>
    <td class=h-toc-text><br></td>
    </tr>
    <tr>
    <td colspan=3></td>
    }
  SelectedFormat={
    <tr>
    <td class=h-tocSelected-text><br></td>
    <td class=h-tocSelected-text><Item class=h-tocSelected-text></td>
    <td class=h-tocSelected-text><br></td>
    </tr>
    <tr><td colspan=3>
    </td>
    }
  EmptyFormat={}
  ReplaceString={Members=>
```

自訂步驟

1. 按一下目錄中的**自訂**，然後先後按一下**自訂佈景主題**與 **新佈景主題**。填寫標題，並上傳**樣式表**與**頁面佈置**檔案。
2. 按**下一步**。
3. 按一下目錄中的**自訂**，然後按一下**佈置**
4. 按一下**選擇佈景主題**。
5. 選取您剛才建立的佈景主題，並按一下**下一步**。應會更新 QuickPlace 的外觀。
6. 請注意，目錄中的 **Members** 鏈結應會消失。
7. 從 QuickPlace 建立一個 PlaceType。此時您可以剛自訂的 QuickPlace 為基礎來建立一個 PlaceType。按一下目錄中的**自訂**，然後按一下**PlaceType** 選項。
8. 按一下**編輯**。在容許從這個 **QuickPlace** 來建立 **PlaceTypes** 嗎？，選擇是；在**要將此 QuickPlace 中的現行成員置於日後從 PlaceType 所建的 QuickPlace 中嗎？**，選擇否。
9. 按**下一步**。
10. 開啓瀏覽器並前往 http://qp_server/quickplace，並登入成 QuickPlace 管理者。
11. 按一下 **PlaceTypes**。
12. 按一下**建立 PLACETYPE...**。輸入 PlaceType 的名稱，並選擇要做為基礎的 QuickPlace。
13. 按**下一步**。您應會看到新 PlaceType。

實務 3：自訂電子郵件通知

施行概觀

如果要變更所有商店之自訂頁面的預設電子郵件通知，您必須修改 CollabEmailContent.jsp 檔，以包含您要的內容。

自訂步驟

1. 修改 CollabEmailContent.jsp 檔。此檔案位於下列目錄中：

```
▶ AIX /usr/WebSphere/AppServer  
/installedApps/WC_Enterprise_App_demo.ear/wcstores.war
```

```
▶ 400 /QIBM/ProdData/WebAsAdv4  
/installedApps/WC_Enterprise_App_demo.ear/wcstores.war
```

```
▶ Linux /opt/WebSphere/AppServer/installedApps/  
WC_Enterprise_App_demo.ear/wcstores.war
```

```
▶ Solaris /opt/WebSphere/AppServer/installedApps/  
WC_Enterprise_App_demo.ear/wcstores.war
```

```
▶ Windows drive:\WebSphere\AppServer\installedApps\  
WC_Enterprise_App_demo.ear\wcstores.war
```

2. 執行下列步驟，以自訂特定商店的電子郵件：

- a. 開啓 DB2 指令視窗。

- b. 連接 WebSphere Commerce 資料庫。如果要連接預設資料庫，請使用下列指令：

```
db2 connect to mall
```

- c. 執行下列的 DB2 指令：

```
db2 update viewreg set properties = 'docname=CollabEmailContent.jsp'  
where viewname = 'CollabEmailContentView'
```

- d. 將自訂範本檔 CollabEmailContent.jsp 置於您的商店目錄中：

```
▶ AIX /usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/  
wcstores.war/ store_name
```

```
▶ 400 /QIBM/ProdData/WebAsAdv4/installedApps/WC_Enterprise_App_demo.ear/  
wcstores.war/store_name
```

```
▶ Linux /opt/WebSphere/AppServer\installedApps/WC_Enterprise_App_demo.ear/  
wcstores.war/store_name
```

```
▶ Solaris /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/  
wcstores.war/store_name
```

```
▶ Windows drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\  
wcstores.war\store_name
```

其中 *store_name* 為您在公佈商店時所選的商店目錄名稱。

- e. 重新啓動 WebSphere 管理伺服器。

第4章 客戶設定檔

本章中所要討論的元素，代表 WebSphere Commerce Accelerator 中之「客戶設定檔」元件的使用者介面。這些元素有助於建立與維護賣方或商家在執行每日所負責之相關商業工作時，所需的客戶設定檔與所有動作。「客戶設定檔」元件所含的元素如下：

- 「客戶設定檔」精靈
- 「客戶設定檔」筆記本






範例程式碼

本節是參照位於下列 URL 之 zip 檔中所含的程式碼範例：

`ftp://ftp.software.ibm.com/software/websphere/commerce/54/profcust.zip`

如果要完成本章中的一些章節，您必須下載此範例程式碼，並安裝在執行 WebSphere Application Server 4 的開發機器上。

當您下載後，請將這個 ZIP 檔解壓縮到下列目錄中：

	<code>/usr/WebSphere/CommerceServer</code>
	<code>/QIBM/UserData/WebSphere/CommerceServer</code>
	<code>/opt/WebSphere/CommerceServer</code>
	<code>/opt/WebSphere/CommerceServer</code>
	<code>drive:\WebSphere\CommerceServer</code>

自訂範例

下列範例說明如何自訂 WebSphere Commerce Accelerator 中的這個部份。

實務 1：新增屬性到客戶設定檔中

施行概觀

在本例中，USERS 表格中的 FIELD1 直欄儲存使用者的飲料喜好設定。有效值如下：

- 啤酒
- 葡萄酒
- 咖啡
- 茶
- 飲用水

下列範例示範每一個步驟。您可在安裝範例程式碼時所建的 profcust 目錄中找到此例。

自訂步驟

1. 建立新 JSP 檔，以便在「客戶設定檔」筆記本中新增新頁面。在本例中，此頁會顯示兩個選項：

- 忽略飲料喜好設定
- 鎖定具備下列一或多項飲料喜好設定的客戶：

喜好設定會以勾選框方式出現在第二選項下。其它明細請查看下列目錄中的隨附 JSP 頁面 BeveragePanel.jsp：

▶ AIX /usr/WebSphere/CommerceServer/profcust/web/tools/segmentation/

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/profcust/web/tools/segmentation/

▶ Linux /opt/WebSphere/CommerceServer/profcust/web/tools/segmentation/

▶ Solaris /opt/WebSphere/CommerceServer/profcust/web/tools/segmentation/

▶ Windows drive:\WebSphere\CommerceServer\profcust\web\tools\segmentation\

2. 在您建立 JSP 檔後，您必須將之新增到 VIEWREG 表格中。其它明細請查看下列目錄中的隨附 SQL Script beverage.sql：

▶ AIX /usr/WebSphere/CommerceServer/profcust/schema/

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/profcust/schema/

▶ Linux /opt/WebSphere/CommerceServer/profcust/schema/

▶ Solaris /opt/WebSphere/CommerceServer/profcust/schema/

▶ Windows drive:\WebSphere\CommerceServer\profcust\schema\

3. 新頁面必須能夠讓「客戶設定檔」筆記本辨識，其位於下列目錄中：

▶ AIX /usr/WebSphere/CommerceServer/xml/tools/segmentation/

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/xml/tools/segmentation/

▶ Linux /opt/WebSphere/CommerceServer/xml/tools/segmentation/

▶ Solaris /opt/WebSphere/CommerceServer/xml/tools/segmentation/

▶ Windows drive:\WebSphere\CommerceServer\xml\tools\segmentation\

建立一份 SegmentNotebook.xml 副本。在範例中，這個新檔案的名稱為 MySegmentNotebook.xml。下列元素已新增到文件中：

```
<panel name="segmentNotebookBeveragePanel"
      url="SegmentNotebookBeveragePanelView"
      group="segmentNotebookMiscPanelGroup" />
```

您必須在分段內容檔中包含新畫面名稱，此檔案位於下列目錄中：

▶ AIX /usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/Properties/com/ibm/commerce/tools/segmentation

▶ 400 /QIBM/ProdData/WebAsAdv4/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation

▶ Linux /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation/

▶ Solaris /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/
properties/com/ibm/commerce/tools/segmentation/

▶ Windows

drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\
properties\com\ibm\commerce\tools\segmentation\

爲了能夠生效，您必須在 Resources.properties 中加入下行：

segmentNotebookBeveragePanel=Preferred beverage

您可以選擇性地將下列目錄中的 MySegmentNotebook.xml：

▶ AIX /usr/WebSphere/CommerceServer/profcust/xml/tools/segmentation/
MySegmentNotebook.xml

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/profcust/xml/tools/
segmentation/MySegmentNotebook.xml

▶ Linux /opt/WebSphere/CommerceServer/profcust/xml/tools/
segmentation/MySegmentNotebook.xml

▶ Solaris /opt/WebSphere/CommerceServer/profcust/xml/tools/
segmentation/MySegmentNotebook.xml

▶ Windows drive:\WebSphere\CommerceServer\profcust\xml\tools\segmentation\
MySegmentNotebook.xml

複製到：

▶ AIX /usr/WebSphere/CommerceServer/xml/tools/segmentation/
MySegmentNotebook.xml

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/xml/tools/segmentation/
MySegmentNotebook.xml

▶ Linux /opt/WebSphere/CommerceServer/xml/tools/segmentation/
MySegmentNotebook.xml

▶ Solaris /opt/WebSphere/CommerceServer/xml/tools/segmentation/
MySegmentNotebook.xml

▶ Windows drive:\WebSphere\CommerceServer\xml\tools\segmentation\
MySegmentNotebook.xml

4. 此時，您必須確定新 XML 文件可供辨識。其做法是修改下列目錄中的 Resources.xml 檔：

▶ AIX /usr/WebSphere/CommerceServer/xml/tools/segmentation/

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/xml/tools/segmentation/

▶ Linux /opt/WebSphere/CommerceServer/xml/tools/segmentation/

▶ Solaris /opt/WebSphere/CommerceServer/xml/tools/segmentation/

▶ Windows drive:\WebSphere\CommerceServer\xml\tools\segmentation\
修改下列元素：

```
<XML name="SegmentNotebook"
      file="segmentation/SegmentNotebook.xml" />
```

將元素改為：

```
<XML name="SegmentNotebook"
      file="segmentation/MySegmentNotebook.xml" />
```

將檔案儲存成 MyResources.xml

您可以選擇性地將下列目錄中的 MyResources.xml：

▶ **AIX** /usr/WebSphere/CommerceServer/profcust/xml/tools/segmentation/
MyResources.xml

▶ **400** /QIBM/UserData/WebSphere/CommerceServer/profcust/xml/tools/
segmentation/MyResources.xml

▶ **Linux** /opt/WebSphere/CommerceServer/profcust/xml/tools/
segmentation/ MyResources.xml

▶ **Solaris** /opt/WebSphere/CommerceServer/profcust/xml/tools/
segmentation/MyResources.xml

▶ **Windows** drive:\WebSphere\CommerceServer\profcust\xml\tools\segmentation\
MyResources.xml

複製到：

▶ **AIX** /usr/WebSphere/CommerceServer/xml/tools/segmentation

▶ **400** /QIBM/UserData/WebSphere/CommerceServer/xml/tools/ segmentation

▶ **Linux** /opt/WebSphere/CommerceServer/xml/tools/segmentation

▶ **Solaris** /opt/WebSphere/CommerceServer/xml/tools/segmentation

▶ **Windows** drive:\WebSphere\CommerceServer\xml\tools\segmentation

5. 將應用程式指向 MyResources.xml。修改下列目錄中的 demo.xml 檔：

▶ **AIX** /usr/WebSphere/CommerceServer/instances/demo/xml/

▶ **400** /QIBM/UserData/WebSphere/CommerceServer/instances/demo/xml/

▶ **Linux** /opt/WebSphere/CommerceServer/instances/demo/xml/

▶ **Solaris** /opt/WebSphere/CommerceServer/instances/demo/xml/

▶ **Windows** drive:\WebSphere\CommerceServer\instances\demo\xml\
尋找：

```
<resourceConfig file="segmentation/Resources.xml" />
```

將元素改為：

```
<resourceConfig file="segmentation/MyResources.xml" />
```

6. 延伸說明客戶設定檔的資料 Bean，
com.ibm.commerce.tools.segmentation.SegmentNotebookDataBean。
在範例中，com.mycompany.tools.segmentation.MySegmentNotebookDataBean 延伸

了資料 Bean，以提供飲料屬性的內容。請在下列目錄中找出這個範例 MySegmentNotebookDataBean.java：

```
> AIX /usr/WebSphere/CommerceServer/profcust/lib/com/
mycompany/tools/segmentation/
```

```
> 400 /QIBM/UserData/WebSphere/CommerceServer/profcust/
lib/com/mycompany/tools/segmentation/
```

```
> Linux /opt/WebSphere/CommerceServer/profcust/lib/
com/mycompany/tools/segmentation/
```

```
> Solaris /opt/WebSphere/CommerceServer/profcust/lib/
com/mycompany/tools/segmentation/
```

```
> Windows drive:\WebSphere\CommerceServer\profcust\lib\
com\mycompany\tools\segmentation\
```

7. 爲了能讓區段筆記本能辨識新資料 Bean，您必須修改說明「客戶設定檔筆記本」的 XML。回到 MySegmentNotebook.xml，並注意如下的變更：

```
<databean name="segmentDetails"
  class="com.mycompany.tools.segmentation.SegmentNotebookDataBean" />
```

已改爲：

```
<databean name="segmentDetails"
  class="com.mycompany.tools.segmentation.MySegmentNotebookDataBean"
  stoplevel="2" />
```

8. 延伸儲存客戶設定檔的控制程式指令。此指令爲 com.ibm.commerce.tools.segmentation.SegmentSaveControllerCmd。請查看位於下列目錄中之 MySegmentSaveControllerCmdImpl 中的範例指令施行：

```
> AIX /usr/WebSphere/CommerceServer/profcust/lib/
com/mycompany/tools/segmentation/
```

```
> 400 /QIBM/UserData/WebSphere/CommerceServer/profcust/lib/
com/mycompany/tools/segmentation/
```

```
> Linux /opt/WebSphere/CommerceServer/profcust/lib/
com/mycompany/tools/segmentation/
```

```
> Solaris /opt/WebSphere/CommerceServer/profcust/lib/
com/mycompany/tools/segmentation/
```

```
> Windows drive:\WebSphere\CommerceServer\profcust\lib\
com\mycompany\tools\segmentation\
```

本例將建構飲料條件的簡單條件。

爲了能讓應用程式能辨識這個新指令，您必須將之新增到 CMDREG 表格中。相關明細請參閱上述步驟 1a 中的 beverage.sql。

9. 延伸會評估客戶設定檔的作業指令，亦即 com.ibm.commerce.membergroup.commands.CheckUserInMemberGroupCmd。有關您要如何評估新條件的詳述，請查看下列目錄中的範例指令 MyCheckUserInMemberGroupCmdImpl.java：

```
> AIX /usr/WebSphere/CommerceServer/profcust/lib/com/mycompany/
```

membergroup/commands/

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/profcust/lib/com/mycompany/membergroup/commands/

▶ Linux /opt/WebSphere/CommerceServer/profcust/lib/com/mycompany/membergroup/commands/

▶ Solaris /opt/WebSphere/CommerceServer/profcust/lib/com/mycompany/membergroup/commands/

▶ Windows *drive:*\WebSphere\CommerceServer\profcust\lib\com\mycompany\membergroup\commands\

您也必須將指令登錄在 CMDREG 表格中。相關明細請參閱 `beverage.sql`。

10. 延伸會列出客戶設定檔限制的作業指令。此指令為 `com.ibm.commerce.tools.segmentation.SegmentConstraintListCmd`。有關您必須進行哪些變更的詳述，請查看下列目錄中的範例指令 `MySegmentConstraintListCmdImpl`：

▶ AIX /usr/WebSphere/CommerceServer/profcust/lib/com/mycompany/tools/segmentation/

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/profcust/lib/com/mycompany/tools/segmentation/

▶ Linux /opt/WebSphere/CommerceServer/profcust/lib/com/mycompany/tools/segmentation/

▶ Solaris /opt/WebSphere/CommerceServer/profcust/lib/com/mycompany/tools/segmentation/

▶ Windows *drive:*\WebSphere\CommerceServer\profcust\lib\com\mycompany\tools\segmentation\

您也必須將指令登錄在 CMDREG 表格中。相關明細請參閱 `bevarage.sql`。

11. 新增下列類別
- `com.mycompany.membergroup.commands.MyCheckUserInMemberGroupCmdImpl`
 - `com.mycompany.tools.segmentation.MySegmentConstraintListCmdImpl`
 - `com.mycompany.tools.segmentation.MySegmentNotebookDataBean`
 - `com.mycompany.tools.segmentation.MySegmentSaveControllerCmdImpl`

位於下列目錄中：

▶ AIX /usr/WebSphere/CommerceServer/profcust/lib

▶ 400 /QIBM/UserData/WebSphere/CommerceServer/profcust/lib

▶ Linux /opt/WebSphere/CommerceServer/profcust/lib

▶ Solaris /opt/WebSphere/CommerceServer/profcust/lib

▶ Windows *drive:*\WebSphere\CommerceServer\profcust\lib

到下列目錄中的 `wcsmcruntime.jar` 檔內：

▶ AIX /usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_

demo.ear/properties/com/ibm/commerce/tools/segmentation

▶ 400 /QIBM/ProdData/WebAsAdv4/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation

▶ Linux /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation/

▶ Solaris /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation/

▶ Windows drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\properties\com\ibm\commerce\tools\segmentation\

12. 將步驟 1 中的 BeveragePanel.jsp 檔複製到下列目錄中：

▶ AIX /usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation

▶ 400 /QIBM/ProdData/WebAsAdv4/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation

▶ Linux /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation/

▶ Solaris /opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/properties/com/ibm/commerce/tools/segmentation/

▶ Windows drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\properties\com\ibm\commerce\tools\segmentation\

13. 這些變更會要求您變更您的存取控制設定，而這些設定並不在本手冊的討論範圍內。請參閱 *WebSphere Commerce 存取控制手冊*；您可從下列 URL 中取得：

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

第5章 活動

直接將規則輸入到資料庫中

雖說使用者介面是將規則資料輸入到資料庫的最簡便方式，但在某些情況下可能會要求您必須手動在 INITIATIVE 表格的 RULE 直欄中輸入規則。規則是使用 XML 來定義，在此情況下，您必須遵守下列所示的文件類型定義 (DTD)：

INITIATIVE 表格之 RULE 直欄的 DTD 如下：

```
<!DOCTYPE rule [  
  <!ELEMENT rule (comment?, (orListCondition |andListCondition |simpleCondition  
|trueCondition |openCondition), action)>  
  
  <!ELEMENT comment EMPTY>  
  <!ATTLIST comment text CDATA #REQUIRED>  
  
  <!ELEMENT action (parameter*)>  
  <!ATTLIST action name CDATA #REQUIRED>  
  
  <!ELEMENT orListCondition (not?, (orListCondition |andListCondition |  
simpleCondition | trueCondition |openCondition)+)>  
  
  <!ELEMENT andListCondition (not?, (orListCondition |andListCondition |  
simpleCondition | trueCondition |openCondition)+)>  
  
  <!ELEMENT simpleCondition (not?, variable, operator, value, qualifier*)>  
  
  <!ELEMENT openCondition (not?, parameter*)>  
  <!ATTLIST openCondition name CDATA #REQUIRED>  
  
  <!ELEMENT trueCondition (not?)>  
  
  <!ELEMENT not EMPTY>  
  
  <!ELEMENT variable EMPTY>  
  <!ATTLIST variable name CDATA #REQUIRED>  
  
  <!ELEMENT operator EMPTY>  
  <!ATTLIST operator name CDATA #REQUIRED>  
  
  <!ELEMENT value EMPTY>  
  <!ATTLIST value data CDATA #REQUIRED>  
  
  <!ELEMENT qualifier EMPTY>  
  <!ATTLIST qualifier name CDATA #REQUIRED>  
  <!ATTLIST qualifier data CDATA #REQUIRED>  
  
  <!ELEMENT parameter (parameter*)>  
  <!ATTLIST parameter name CDATA #REQUIRED>  
  <!ATTLIST parameter value CDATA #REQUIRED>  

```

simpleCondition 元素

預設的行動施行支援下列的 simpleCondition 元素。您可將這些合併到任何 orListCondition、andListCondition 與 openCondition 元素的組合中，以建立規則的條件部份。

表 2.

變數名稱	支援的運算子	支援的值	限定元
segment	= !=	客戶設定檔的名稱。	無
shoppingCartSku	= !=	產品 SKU。	無
shoppingCartCategory	= !=	產品種類名稱。	語言
purchaseHistorySku	= !=	產品 SKU。	無
purchaseHistoryCategory	= !=	產品種類名稱。	語言
shoppingCartTotal	= != > < >= <=	小數值。	貨幣
dayOfWeek	= !=	週日、週一、週二、週三、週四、週五、週六	無

openCondition 元素

預設施行支援下列的 openCondition 元素。您可將這些合併到任何 orListCondition、andListCondition 與 simpleCondition 元素的組合中，以建立規則的條件部份。

表 3.

開放條件名稱	支援的參數
shoppingCartTotal	comparisonType - 可為 ">"、"<" 或 "=" 之一； 值 - 小數值貨幣
shoppingCartCategory	comparisonType - 可為 "=" 或 "!=" 之一；值 - 種類語言的名稱
shoppingCartSku	comparisonType - 可為 ">"、"<" 或 "=" 之一； 值 - 小數值貨幣
purchaseHistoryCategory	comparisonType - 可為 "=" 或 "!=" 之一；值 - 種類貨幣的名稱
purchaseHistorySku	comparisonType - 可為 ">"、"<" 或 "=" 之一； 值 - 小數值貨幣

動作元素

預設施行支援下列的動作元素。這些可搭配任何條件元素而建構成一項規則。

表 4.

動作名稱	支援的參數	支援的值	支援的子參數
suggestiveSell	queryOperator	or and	無
	queryType	skuQuery genericQuery	無
	sku	產品 SKU	無
	skuPrefix	產品 SKU 字首	無
	category	產品種類名稱	語言
	productDescription	產品種類說明	語言
	inventoryQuantity	整數	comparisonOperator
	listPrice	小數值	comparisonOperator currency
availabilityDate	yyyy-mm-dd-00.00.00 日期格式	comparisonOperator	
collaborativeFiltering	無		
awarenessAd	collateral	廣告名稱	
discountAd	discountCollateral	廣告名稱	折扣識別碼
couponAd	couponCollateral	廣告名稱	
categoryRecommendation	category	種類的名稱	

下列範例行動顯示男性春季廣告以及女性秋季廣告：

```
<ruleSet>
  <rule>
    <comment text="顯示男性春季廣告"/>
    <simpleCondition>
      <variable name="segment"/>
      <operator name="="/>
      <value data="men"/>
    </simpleCondition>
    <action name="awarenessAd">
      <parameter name="collateral" value="Spring">
      </parameter>
    </action>
  </rule>
  <rule>
    <comment text="顯示女性秋季廣告"/>
    <simpleCondition>
      <variable name="segment"/>
      <operator name="="/>
      <value data="women"/>
    </simpleCondition>
    <action name="awarenessAd">
      <parameter name="collateral" value="Fall">
      </parameter>
    </action>
  </rule>
</ruleSet>
```

基礎架構元素

規則套件

「活動」元件採用 `com.ibm.commerce.rule` 套件。此套件提供的類別有助您將一組規則轉換成 XML 格式以及從 XML 格式轉換。此外亦支援呼叫規則集中的規則。

表 5.

類別/介面名稱	說明
Action	Action 類別是提供規則中動作部份所用之名稱與參數的內容。這些參數具有名稱與值，也可以具備選用的子參數。
ActionHandler	必須施行 ActionHandler 介面，才能提供規則中之動作部份的施行。當規則中的條件部份評估為 True 時，即會呼叫 "performAction" 方法。當施行 "performAction" 方法時，會傳入適當的 Action 類別案例。
Rule	Rule 類別會提供一些方法，以協助您將 Rule 案例轉換成 XML 格式（或從 XML 格式轉換）。Rule 類別有三項內容。一項是針對規則的條件部份。一項是針對規則的動作部份。另一項則是備註方面。條件內容必須是 Condition 類別的一個案例，且可在 <code>com.ibm.commerce.condition</code> 套件中找到。Rule 類別中也有一個 "invoke" 方法，可接受 Evaluator 介面與 ActionHandler 介面的施行。如果條件評估為 True，則會呼叫 ActionHandler 施行中的 "performAction" 方法。
RuleConstants	RuleConstants 介面含有規則套件所用的一些公用常數。
RuleSet	RuleSet 類別會提供一些方法，以協助您將 RuleSet 案例轉換成 XML 格式（或從 XML 格式轉換）。它有一個內容為規則物件陣列。此外，也有一個 invoke 方法，可接受 Evaluator 與 ActionHandler 介面的施行。當呼叫 invoke 方法時，它會循環執行規則物件陣列，且一個個呼叫。

活動會使用規則套件將活動行動中的規則部份儲存與載入成 XML 格式（或從 XML 格式轉換）。其亦有助於呼叫規則。CampaignInitiativeEvaluateCmd 作業指令可提供用以處理活動行動動作之 ActionHandler 介面的施行。

條件套件

規則套件採用 `com.ibm.commerce.condition` 套件。此套件提供的類別有助您將布林條件成 XML 格式（以及從 XML 格式轉換）。此外亦可支援評估條件。

表 6.




類別/介面名稱	說明
AndListCondition	AndListCondition 類別是從 ConditionList 類別延伸而來，可提供一項條件，而該條件內含 AND 布林運算子所結合的條件清單。
Condition	Condition 類別為一種抽象類別，可提供方法協助您轉換成 XML 格式（或從 XML 格式轉換）。此外亦會提供一種抽象方法，可用來評估條件。
ConditionConstants	ConditionConstants 介面提供條件套件所用的常數值。
ConditionList	ConditionList 類別為一種抽象類別，它是由 Condition 類別延伸而來。它可支援內含條件清單的條件。
ConditionUtil	ConditionUtil 類別提供一些可在評估條件期間使用的靜態方法。
Evaluator	必須施行 Evaluator 介面，才能評估簡單條件或開放條件。
OpenCondition	OpenCondition 類別是從 Condition 類別延伸而來，可提供一個通用條件。此條件是由「名稱-值」配對構成。
OrListCondition	OrListCondition 類別是從 ConditionList 類別延伸而來，可提供一項條件，而該條件內含 OR 布林運算子所結合的條件清單。
SimpleCondition	SimpleCondition 類別是從 Condition 類別延伸而來，可提供一個由變數、運算子、值構成的條件。
TrueCondition	TrueCondition 類別是從 Condition 類別延伸而來，可提供一個恆評估為 True 的條件。

規則套件會使用條件套件將規則中的條件部份儲存與載入成 XML 格式（或從 XML 格式轉換）。其亦有助於評估條件。

抽象 Condition 類別有五項具體的延伸。其中的三項（AndListCondition、OrListCondition 與 TrueCondition）知道如何評估它們。其餘的兩項（OpenCondition 與 SimpleCondition）必須施行 Evaluator 介面才能加以評估。CampaignInitiativeEvaluateCmd 作業指令可提供用以評估活動行動條件之 Evaluator 介面的施行。

Blaze 規則專案

活動行動的預設施行採用有助於評估活動行動規則的 Blaze 規則專案。依預設，所有開放條件會導致由 CampaignInitiativeEvaluateCmd 作業指令來呼叫 Blaze 規則專案。請在下列目錄中找出 CampaignInitiativeEvaluator 專案：

-  /usr/WebSphere/CommerceServer/rules/campaigns
-  /QIBM/UserData/WebSphere/CommerceServer/rules/campaigns
-  /opt/WebSphere/CommerceServer/rules/campaigns
-  /opt/WebSphere/CommerceServer/rules/campaigns

► Windows `drive:\WebSphere\CommerceServer\rules\campaigns`

此專案的預設施行能夠評估活動行動使用者介面所建的所有開放條件。您可以新增其它開放條件的支援到此專案中。

自訂

在此預期我們的客戶希望延伸活動行動的預設施行，而新增更多的支援條件與可能的其它動作與顯示類型。

針對新行動條件新增執行時期支援

用以評估活動行動條件的 **Blaze** 專案可接受 **CampaignInitiativeContext** 案例。此環境定義可提供指令環境定義的存取，以及提供開放條件的名稱與參數。如果要新增新開放條件的支援，您必須新增新規則到 `conditionEvaluator` 規則中。此規則需檢查新條件的名稱並評估條件。在評估條件期間，您可透過環境定義物件依名稱來存取開放條件參數。

針對新行動條件新增建置時期支援

如果要讓行銷經理能善用新條件，必須延伸活動行動使用者介面，以提供新條件的支援。其做法是新增新條件到「活動行動」精靈或筆記本的 `Which` 畫面中 (`WhenAddPanel.jsp`)。您也必須延伸

CampaignInitiativeSaveControllerCmdImpl 類別，以建構新條件物件。此檔案位於下列目錄中：

► AIX `/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wctools.war/tools/campaigns/`

► 400 `/QIBM/ProdData/WebAsAdv4/installedApps/WC_Enterprise_App_demo.ear/wctools.war/tools/campaigns/`

► Linux `/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wctools.war/tools/campaigns/`

► Solaris `/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wctools.war/tools/campaigns/`

► Windows `drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wctools.war\tools\campaigns\`

針對新活動行動動作新增執行時期支援

如果要提供新活動行動動作的支援，您必須延伸

CampaignInitiativeEvaluateCmdImpl 類別，並改寫 `performAction` 方法。如果動作名稱符合您的新動作名稱，您應執行動作；否則您應呼叫超級方法，以執行預設施行。

針對新活動行動動作新增建置時期支援

如果要讓行銷經理能善用新活動行動動作，您必須更新「活動行動」精靈或筆記本的 `What` 畫面 (`InitiativeWhatPanel.jsp`)。您也必須延伸

CampaignInitiativeSaveControllerCmdImpl 類別，以建構新動作物件。此檔案位於下列目錄中：

► AIX `/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear/wctools.war/tools/campaigns/`

► 400 `/QIBM/ProdData/WebAsAdv4/installedApps/WC_Enterprise`

_App_demo.ear/wctools.war/tools/campaigns/

Linux /opt/WebSphere/AppServer/installedApps/WC_Enterprise_
App_demo.ear/wctools.war/tools/campaigns/

Solaris /opt/WebSphere/AppServer/installedApps/WC_Enterprise_
App_demo.ear/wctools.war/tools/campaigns/

Windows drive:\WebSphere\AppServer\installedApps\WC_Enterprise_
App_demo.ear\wctools.war\tools\campaigns\

新增新 e-行銷點類型的支援

如果您發現您必須提供其它的輸出類型，您可以延伸 EMarketingSpot 類別，以提供可供頁面設計者使用的新內容。您也必須延伸

CampaignInitiativeEvaluateCmdImpl 類別，以傳回新輸出類型。

第6章 型錄搜尋

本章中所要討論的元素是說明必須自現有搜尋功能延伸而來的活動。其它功能包括介紹搜尋元件所能存取之新屬性與表格的相關指引。將綱目最佳化除了有助於降低執行時期的成本，也可提升搜尋元件的效能。此元件所含的可自訂元素如下：

- 搜尋 Bean
- 搜尋引擎
- 摘要表格定義 Script

表格 8 彙整可透過型錄搜尋 Bean 搜尋的資料庫直欄清單。由於網站需求的多樣化，您可能會發現必須提供此表格未說明的直欄搜尋。

自訂實務

如果您要在可從搜尋引擎取得這些屬性的搜尋 Bean 中新增搜尋屬性，請施行實務 1，以便新增屬性到搜尋 Bean 中。如果搜尋引擎不支援搜尋此屬性，但此屬性位於可搜尋的表格之一中，請施行實務 2，以新增屬性到引擎中。如果此屬性不在搜尋引擎要搜尋的表格之一中，請施行實務 3，以新增表格到搜尋引擎中。新增多元屬性到搜尋 Bean 的程序，和新增其它任何屬性到搜尋 Bean 的程序大致一樣（施行實務 1），只有在施行上略有不同。請參閱實務 4。如果要提升執行時期效能，請施行實務 5，以藉由為每個種類各建一個表格（以降低所要搜尋的列數），來修改與最佳化摘要表格定義 Script。下列實務描述自訂此元件的時機：

實務 1： 新增屬性到搜尋 Bean 中

將新屬性引入已可從搜尋引擎中取得的 Bean 中。

實務 2： 新增屬性到搜尋引擎中

將新屬性引入已使用該表格進行搜尋的搜尋引擎中。

實務 3： 新增表格到搜尋引擎中

將新表格引入搜尋引擎中。

實務 4： 新增多元屬性到搜尋 Bean 中

將多元屬性引入 Bean 中。

實務 5： 藉由最佳化的摘要表格提升效能

運用資料集知識來最佳化摘要表格定義。

註：「搜尋引擎」、「搜尋介面」與「制式搜尋架構」三者同義，而在本文件中將交替使用。

實務 1： 新增屬性到搜尋 Bean 中

施行概觀

您最想自訂的第一件事或許是新增另一個可搜尋的屬性。此活動需要變更搜尋 Bean，而如果此屬性無法使用，您也可以選擇性地變更搜尋引擎（請見實務 2）。若要自訂搜尋 Bean，您必須為其建立子類別，並延伸現有方法。有關其做法的相關範例程式碼說明，可在 `samples/search` 目錄中找到。

自訂步驟

本例說明如何新增其它搜尋屬性到型錄搜尋 Bean 中，其中 meta 資料類別存在於 com.ibm.search.catalog 套件中。型錄搜尋資料 Bean 是一種類別。若要自訂，請執行下列步驟：

1. 建立 CatEntrySearchListDataBean 類別的子類別。
2. 宣告您要新增之新可搜尋屬性的變數。您可宣告兩種變數類型，以儲存可搜尋屬性的相關資訊。第一種類型可用來儲存屬性的值。第二種類型可內含可在搜尋中使用的相關資訊，像是：搜尋運算子、布林搜尋、區分大小寫等。儲存在這些變數中的資訊可在產生查詢型錄資訊的 SQL 查詢中使用。

註：產生 SQL 查詢時所用的資料庫直欄必須已定義在搜尋介面 RuleQuery 類別中或其子類別之一中（請參閱 '自訂搜尋介面'），才能在產生的查詢中當成限制項使用。

3. 為所宣告的變數建立存取元（get 與 set 方法）。
4. 建立 populate() 方法以執行下列事項：
 - a. 案例化 RuleQuery 或 RuleQuery 的子類別。
 - b. 使用 setRuleQuery(ruleQueryInstance) 方法，讓此 RuleQuery 案例參照上層 Bean 類別中的案例。
 - c. 呼叫 super.setPredefinedAttributes() 方法。
 - d. 針對使用搜尋介面的新可搜尋限制項，將 <Predicate> 公式化。
 - e. 使用 super.setIsAllNull(false)，通知上層已新增一個新搜尋屬性。
 - f. 呼叫 super.execute() 方法。

範例

本例說明如何新增新可搜尋屬性 onSpecial 到型錄搜尋 Bean 中，並假設 meta 資料類別存在於 com.ibm.search.catalog 套件中：

1. 建立一個新子類別。
2. 為可搜尋屬性建立新變數。

```
public class CustomCatEntrySearchListDataBean
    extends CatEntrySearchListDataBean {
    private static final String COPYRIGHT =
        com.ibm.commerce.copyright.IBMCopyright.SHORT_COPYRIGHT;
    protected java.lang.String onSpecial;
    protected java.lang.String onSpecialOperator;
}
```

onSpecial 變數用來儲存可搜尋屬性的值。onSpecialOperator 變數用來儲存搜尋運算子（例如 like、=、<、>、<=、>=）。

3. 建立所有變數的存取元

```
public java.lang.String getOnSpecial(){
    return onSpecial;
}

public void setOnSpecial(java.lang.String newOnSpecial) {
    onSpecial = newOnSpecial;
}
```

4. 建立 populate() 方法


```

public void populate() throws Exception {
    String langId = commandContext.getLanguageId().toString();
    RuleQuery ruleQueryInstance = new RuleQuery();
    setRuleQuery(ruleQueryInstance);
    super.setPredefinedAttributes();
    if(onSpecial != null){
        ruleQueryInstance.addSelectAttribute(ruleQueryInstance.
CATENTRY_ONSPECIAL_Attr,
getNumeric(onSpecialOperator), onSpecial);
        ruleQueryInstance.addSelectOperator(ruleQueryInstance.AND_Operator);
    }

    super.setIsAllNull(false);
}
q.addSelectOperator(q.AND_Operator);
}
super.execute(); //步驟 4f
}

```

實務 2：新增屬性到搜尋引擎中

施行概觀

搜尋引擎需要屬性與表格的相關資訊，才能適當將查詢公式化，以便從型錄中擷取結果。此資訊是以 meta 資料方式保留。如果要延伸搜尋引擎，以新增新可搜尋屬性，則必須提供額外的屬性 meta 資料，以及選擇性提供表格 meta 資料定義。只有在將新表格引入搜尋引擎中時，才需要表格 meta 資料，其進一步詳述，請見「實務 3」。一旦您完成本實務，您必須施行「實務 1：新增屬性到搜尋 Bean 中」，以便讓您的 JSP 頁面設計者可以使用到。有關其做法的範例程式碼，可在 samples/search 目錄中找到。

自訂步驟

搜尋引擎為制式搜尋架構中的一部份。它以名為 RuleQuery 的類別方式呈現，而其它的 meta 資料（像是 AttributeInfo 與 TableInfo）則是分別說明直欄與表格的名稱。本例說明如何針對不存在於 com.ibm.search.catalog 套件中的資料庫直欄（但此套件中含有該直欄所屬表格的 meta 資料），建立 meta 資料類別。如果要自訂搜尋引擎，請執行下列步驟：

1. 針對您要變成可搜尋的每一個直欄與表格，定義搜尋介面 meta 資料。這會要求：
 - a. 每個可搜尋表格必須有一個對應的類別，且該類別是 TableInfo 類別的子類別。這個子類別必須指定表格名稱。
 - b. 每個可搜尋直欄必須有一個對應的類別，且該類別是 AttributeInfo 類別的子類別。此子類別必須指定直欄名稱、該直欄所屬的 TableInfo 子類別以及直欄的 SQL 資料類型。
2. 建立 RuleQuery 類別的子類別，並針對每個新資料庫直欄定義靜態整數參照。請注意，整數值 0 到 1000 保留供系統使用。
3. 建立 findAttributeInfoName() 方法。為了避免此方法改寫上層 findAttributeInfoName() 方法，建議您呼叫 super.findAttributeInfoName() 方法。如果上層 findAttributeInfoName() 被改寫，將無法使用定義在上層類別中的資料庫直欄。
4. 新增 factory 類別建立邏輯到此方法中，以便針對每一個資料庫直欄各建一個 AttributeInfo meta 資料。
5. 修改 search.xml 檔，以便針對任何新的可搜尋表格，新增預先定義的表格結合關係。所有表格組合皆須具備結合關係。如果要修改 search.xml，請執行下列步驟：

- a. 新增 COMMENT 區段，以說明項目所公式化的表格結合關係。
- b. 新增 KEY 區段，以唯一識別該項目。所要結合的表格會當成唯一鍵值使用。
- c. 新增 VALUE 區段，以指定鍵值的相關值。 value 區段中的項目是定義結合關係。 value 區段中的直欄名稱是使用該直欄的 AttributeInfo 子類別定義而成。

範例

下列範例將建立新可搜尋資料庫直欄 CATENTRY.ONSPECIAL：

1.

- a. 建立 TableInfo 類別的子類別，以指定表格名稱 CATENTRY（如果此表格的類別不存在的話）：

```
public final class CatEntryTableInfo extends TableInfo
{
    private static final String COPYRIGHT =
        com.ibm.commerce.copyright.IBMCopyright.SHORT_COPYRIGHT;

    // Java 中的獨立慣用句
    private static CatEntryTableInfo info = new CatEntryTableInfo ();

    /**
     * 建構子集的表格名稱。
     */
    public CatEntryTableInfo() {
        super("CATENTRY");
    }

    /**
     * 方法會傳回獨立慣用句。
     * @return com.ibm.commerce.search.catalog.CatEntryTableInfo
     */
    public static CatEntryTableInfo getSingleton()
    {
        return info;
    }
}
```

- b. 建立 AttributeInfo 類別的子類別，以指定直欄名稱 ONSPECIAL：

```
public final class CatEntryOnSpecialAttributeInfo extends AttributeInfo
{
    private static final String COPYRIGHT =
        com.ibm.commerce.copyright.IBMCopyright.SHORT_COPYRIGHT;

    // Java 中的獨立慣用句
    private static CatEntryOnSpecialAttributeInfo info = new
        CatEntryOnSpecialAttributeInfo(CatEntryTableInfo.getSingleton());

    /**
     * 建構子集的直欄名稱、直欄的表格名稱，以及資料類型。
     */
    public CatEntryOnSpecialAttributeInfo(TableInfo info) {
        super(info);
        columnName = "ONSPECIAL";
        sqltype = SQLTYPE_NUM;
    }

    /**
     * 方法會傳回獨立慣用句。
     * @return com.ibm.commerce.search.catalog.CatEntryOnSpecialAttributeInfo
     */
    public static CatEntryOnSpecialAttributeInfo getSingleton()
    {
        return info;
    }
}
```

2. 建立搜尋介面 RuleQuery 的子類別：

```
public class CustomQuery extends RuleQuery {
    private static final String COPYRIGHT =
        com.ibm.commerce.copyright.IBMCopyright.SHORT_COPYRIGHT;

    public final static int CATENTRY_ONSPECIAL_Attr = 1001; //Static reference
}
```

3. 建立 findAttributeInfoName() 方法：

```
protected String findAttributeInfoName(int attrId) {
    String className;
    className = super.findAttributeInfoName(attrId);
    switch (attrId) {
        case CATENTRY_ONSPECIAL_Attr:
            className = new String("CatEntryOnSpecialAttributeInfo");
            break;
    }
    return className;
}
```

4. 必要時可修改 Search.xml（在本例中，不必修改 search.xml 檔）

註：如果要新增這個新 meta 資料到搜尋 Bean 中，請遵循實務 1 中的步驟進行。不過，您必須使用具備 meta 資料資訊的 RuleQuery 子查詢的案例（請見上例中的 CustomQuery），而非使用 RuleQuery 類別的案例。

實務 3：新增表格到搜尋引擎中

施行概觀

如果要從目前搜尋引擎不支援的新表格新增屬性，您必須定義表格的相關資訊，並說明此表格和引擎所用之其它表格間的關係。此資訊為結合關係，用以衍生述語以指出各表格和其它表格間的關係，而此資訊將保留做為 meta 資料用。如果要延伸搜尋引擎，以新增新可搜尋屬性，您必須定義每一個表格以及其和其它表格間的關係。

您必須施行實務 2：“新增屬性到搜尋引擎中”，以便讓和此表格有關的屬性可供搜尋 Bean 建立者使用。有關其做法的範例程式碼，可在 samples/search 目錄中找到。

自訂步驟

這些範例顯示如何建立資料庫表格的結合關係 meta 資料類別。一般而言，您應為每個邏輯表格關係組合，分別定義一個結合關係。

範例

本例是變更現有的 search.xml，以包含 CATENTRY 表格與 OFFERPRICE 表格間的新結合關係項目：

1. 新增 COMMENT 區段：

```
<!-- ***** WWW *****
    setW.add("OFFER.CATENTRY_ID = CATENTRY.CATENTRY_ID");
    setW.add("OFFER.OFFER_ID = OFFERPRICE.OFFER_ID");
-->
```

2. 新增 KEY 與 VALUE 區段：

```
<entry>
  <key1>OFFERPRICE</key1>
  <key2>CATENTRY</key2>
  <value>
    <attrinfo1>OfferOfferIdAttributeInfo</attrinfo1>
    <attrinfo2>OfferPriceOfferIdAttributeInfo</attrinfo2>
```

```

    </value>
    <value>
      <attrinfo1>CatEntryIdentifierAttributeInfo</attrinfo1>
      <attrinfo2>OfferCatentryIdAttributeInfo</attrinfo2>
    </value>
  </entry>

```

實務 4：新增多元屬性到搜尋 Bean 中

施行概觀

新增多元屬性的搜尋到 Bean 中，和新增其它任何可搜尋屬性到 Bean 中的程序一樣。只要任何多元屬性可認識與識別資料集，請遵循實務 1，以便在搜尋 Bean 中提供這些屬性。此實務和實務 1 間唯一的差異是在用來建構多元屬性述語的方法參數中（相關明細請見下列範例中的粗體文字）。搜尋引擎早已能搜尋定義在 ATTRIBUTE 與 ATTRVALUE 表格中的多元屬性。

自訂步驟

和「自訂實務 1：新增屬性到搜尋 Bean 中」相同。

範例

本例說明如何新增新可搜尋多元屬性 color 到型錄搜尋 Bean 中。

1. 建立一個新子類別。
2. 為可搜尋屬性建立新變數。

```

public class ExtendedCatEntrySearchListDataBean
    extends CatEntrySearchListDataBean implements
        ExtendedCatEntrySearchListInputDataBean,
        ExtendedCatEntrySearchListSmartDataBean {
    private static final String COPYRIGHT =
        com.ibm.commerce.copyright.IBMCopyright.SHORT_COPYRIGHT;
    protected java.lang.String colorValue;
    protected java.lang.String colorValueCaseSensitive;
    protected java.lang.String colorValueOperator;
}

```

colorValue 變數中儲存可搜尋屬性的值。colorValueOperator 變數儲存搜尋運算子（例如，like、=、<、>、<=、>=）。

3. 建立所有變數的存取元

```

public java.lang.String getColorValue(){
    return colorValue;
}

public void setColorValue(java.lang.String newColorValue) {
    colorValue = newColorValue;
}

```

4. 建立 populate() 方法

```

public void populate() throws Exception {
    String langId = commandContext.getLanguageId().toString();
    RuleQuery ruleQueryInstance = new RuleQuery();
    setRuleQuery(ruleQueryInstance);
    super.setPredefinedAttributes();
    if(isEmpty(colorValue)){
        if(colorValueCaseSensitive ==null || !colorValueCaseSensitive.equals
(CASE_SENSITIVE)){
            ruleQueryInstance.addSelectAttribute("Color",
getStringOperator(colorValueOperator),
colorValue.toUpperCase(),

```

```

        ruleQueryInstance.ATTRVALUE_STRINGVALUE_Attr,langId, null,
ruleQueryInstance.UPPER_Function);
        super.setIsAllNull=false;
    }else{
        ruleQueryInstance.addSelectAttribute("Color",
getStringOperator(colorValueOperator),
        colorValue,
        ruleQueryInstance.ATTRVALUE_STRINGVALUE_Attr,langId, null);
        super.setIsAllNull=false;
    }
    ruleQueryInstance.addSelectOperator(q.AND_Operator);
}
super.setIsAllNull(false);

q.addSelectOperator(q.AND_Operator);
}
super.execute(); //步驟 4f
}

```

實務 5：藉由最佳化的摘要表格提升效能

施行概觀

當定義資料集時，此知識可用來建立更為最佳化的摘要表格，而明顯提升搜尋效能。其目標是建立和將會遇到之預期查詢類型類似的表格。在大部份情況下，這些摘要表格通常很小，因此套用於這些表格上的查詢，會比使用較大的基礎原生表格或其它最佳化較少的摘要表格快。

自訂步驟

本例說明如何為每個種類群組建立一個多元屬性種類群組摘要表格。如果有許多種類群組 (CATGROUPS) 的項目具有多元屬性，請為每個群組分別定義一個摘要表格定義。

建立此摘要表格的步驟如下：

1. 定義一個表格建立陳述式，且陳述式中內含一個要做為摘要表格定義的 select 查詢。此 select 陳述式的 "group by" 子句中，必須指定和結果集中所列相同的直欄。有關進一步詳述，請參閱 DB2 文件中有關 "自動化摘要表格" (AST) 或 Oracle 文件中有關 "將概略表具體化" 的說明。
2. 在 select 陳述式中，為種類指定一項種類群組述語。
3. 建立內含相同結果集直欄的索引，且其順序和上述查詢的指定相同。
4. 針對每一個種類群組，重複上述步驟。

範例

本例顯示在三個種類群組 (10000、10001 與 10002) 下，其多元屬性的摘要表格建立 Script。粗體部份則是和 wcs.summary.richAttributeCatGroup.create.sql 檔 (本例的基礎) 中的一般定義相對照。

```

// 種類群組 10000 的摘要表格定義
CREATE TABLE RICHATTRCG0 AS
(
    SELECT ATTRIBUTE.NAME,
        ATTRIBUTE.LANGUAGE_ID,
        ATTRVALUE.CAENTRY_ID,
        ATTRVALUE.FLOATVALUE,
        ATTRVALUE.INTEGERVALUE,
        ATTRVALUE.STRINGVALUE,
        CATGPENREL.CATGROUP_ID,
        COUNT(*) AS C5
    FROM ATTRIBUTE, ATTRVALUE, CATGPENREL

```

```

WHERE ATTRIBUTE.ATTRIBUTE_ID=ATTRVALUE.ATTRIBUTE_ID
      AND ATTRIBUTE.CAENTRY_ID=CATGPENREL.CAENTRY_ID
      AND CATGPENREL.CATGROUP_ID = 10000
GROUP BY ATTRIBUTE.NAME,
      ATTRIBUTE.LANGUAGE_ID,
      ATTRVALUE.CAENTRY_ID,
      ATTRVALUE.FLOATVALUE,
      ATTRVALUE.INTEGERVALUE,
      ATTRVALUE.STRINGVALUE,
      CATGPENREL.CATGROUP_ID
)DATA INITIALLY DEFERRED REFRESH IMMEDIATE NOT LOGGED INITIALLY;
ALTER TABLE RICHATTRCG0 ACTIVATE NOT LOGGED INITIALLY;
REFRESH TABLE RICHATTRCG0;

commit;

SET CURRENT REFRESH AGE = ANY;

CREATE INDEX I_RICHATTRCG0 ON RICHATTRCG0
(
  NAME          ASC,
  LANGUAGE_ID   ASC,
  CAENTRY_ID    ASC,
  FLOATVALUE    ASC,
  INTEGERVALUE  ASC,
  STRINGVALUE   ASC,
  CATGROUP_ID  ASC
);

commit;

// 種類群組 10001 的摘要表格定義
CREATE TABLE RICHATTRCG1 AS
(
  SELECT ATTRIBUTE.NAME,
         ATTRIBUTE.LANGUAGE_ID,
         ATTRVALUE.CAENTRY_ID,
         ATTRVALUE.FLOATVALUE,
         ATTRVALUE.INTEGERVALUE,
         ATTRVALUE.STRINGVALUE,
         CATGPENREL.CATGROUP_ID,
         COUNT(*) AS C5
  FROM ATTRIBUTE, ATTRVALUE, CATGPENREL
  WHERE ATTRIBUTE.ATTRIBUTE_ID=ATTRVALUE.ATTRIBUTE_ID
        AND ATTRIBUTE.CAENTRY_ID=CATGPENREL.CAENTRY_ID
        AND CATGPENREL.CATGROUP_ID = 10001
  GROUP BY ATTRIBUTE.NAME,
         ATTRIBUTE.LANGUAGE_ID,
         ATTRVALUE.CAENTRY_ID,
         ATTRVALUE.FLOATVALUE,
         ATTRVALUE.INTEGERVALUE,
         ATTRVALUE.STRINGVALUE,
         CATGPENREL.CATGROUP_ID
)DATA INITIALLY DEFERRED REFRESH IMMEDIATE NOT LOGGED INITIALLY;
ALTER TABLE RICHATTRCG1 ACTIVATE NOT LOGGED INITIALLY;
REFRESH TABLE RICHATTRCG1;

commit;

SET CURRENT REFRESH AGE = ANY;

CREATE INDEX I_RICHATTRCG1 ON RICHATTRCG1
(
  NAME          ASC,
  LANGUAGE_ID   ASC,
  CAENTRY_ID    ASC,
  FLOATVALUE    ASC,

```

```

        INTEGERVALUE    ASC,
        STRINGVALUE    ASC,
        CATGROUP_ID    ASC
    );

commit;

// 種類群組 10002 的摘要表格定義
CREATE TABLE RICHATTRCG2 AS
(
    SELECT ATTRIBUTE.NAME,
           ATTRIBUTE.LANGUAGE_ID,
           ATTRVALUE.CAENTRY_ID,
           ATTRVALUE.FLOATVALUE,
           ATTRVALUE.INTEGERVALUE,
           ATTRVALUE.STRINGVALUE,
           CATGPENREL.CATGROUP_ID,
           COUNT(*) AS C5
    FROM ATTRIBUTE, ATTRVALUE, CATGPENREL
    WHERE ATTRIBUTE.ATTRIBUTE_ID=ATTRVALUE.ATTRIBUTE_ID
          AND ATTRIBUTE.CAENTRY_ID=CATGPENREL.CAENTRY_ID
          AND CATGPENREL.CATGROUP_ID = 10002
    GROUP BY ATTRIBUTE.NAME,
           ATTRIBUTE.LANGUAGE_ID,
           ATTRVALUE.CAENTRY_ID,
           ATTRVALUE.FLOATVALUE,
           ATTRVALUE.INTEGERVALUE,
           ATTRVALUE.STRINGVALUE,
           CATGPENREL.CATGROUP_ID
)
DATA INITIALLY DEFERRED REFRESH IMMEDIATE NOT LOGGED INITIALLY;
ALTER TABLE RICHATTRCG2 ACTIVATE NOT LOGGED INITIALLY;
REFRESH TABLE RICHATTRCG2;

commit;

SET CURRENT REFRESH AGE = ANY;

CREATE INDEX I_RICHATTRCG2 ON RICHATTRCG2
(
    NAME                ASC,
    LANGUAGE_ID         ASC,
    CAENTRY_ID          ASC,
    FLOATVALUE          ASC,
    INTEGERVALUE        ASC,
    STRINGVALUE         ASC,
    CATGROUP_ID        ASC
);

commit;

```

型錄搜尋資料 Bean 變數

必須透過 URL (JSP) 傳給型錄搜尋 Bean 的變數清單

表 7.

名稱	資料類型	說明
beginIndex	字串	此變數用來分頁結果集。其值必須是頁面中第一結果列的索引。
catGroupId	字串	此變數的值用於種類 ID 的搜尋上。
categoryTerm	字串	此變數的值用於種類名稱或說明（或兩者）的搜尋上。
categoryTermCaseSensitive	字串	使用者可選擇搜尋時是否區分大小寫。此變數中的值用來識別搜尋有否區分大小寫。值必須是 yes（搜尋時區分大小寫）或 no（搜尋時不區分大小寫）。

表 7. (繼續)

categoryTermOperator	字串	使用者可選擇要以 like 或 equal 做為搜尋運算子。此變數中的值用來儲存使用者的選擇。值必須是 LIKE (使用 like 運算子) 或 EQUAL (使用 equal 運算子)。
categoryTermScope	整數	使用者可將搜尋範圍限定在：「名稱之一」、「名稱與簡短說明」或「名稱、簡短說明與詳細說明」上。此變數中的值用來儲存使用者的選擇。值必須是 1 (名稱與簡短說明)、2 (僅名稱) 或 3 (名稱、簡短說明與詳細說明)。
categoryType	字串	使用者可指定三種搜尋準則類型：All、Any 或 Exact Phrase。此變數中的值用來儲存使用者的搜尋準則。值必須是 ALL (所有搜尋準則)、ANY (任何搜尋準則) 或 EXACT (精確詞組準則)。
currency	字串	此變數的值用於貨幣搜尋上。
currencyCaseSensitive	字串	使用者可選擇搜尋時是否區分大小寫。此變數中的值用來識別搜尋有否區分大小寫。值必須是 yes (搜尋時區分大小寫) 或 no (搜尋時不區分大小寫)。
currencyOperator	字串	使用者可選擇要以 like 或 equal 做為搜尋運算子。此變數中的值用來儲存使用者的選擇。值必須是 LIKE (使用 like 運算子) 或 EQUAL (使用 equal 運算子)。
filterTerm	字串	此變數中的值用來過濾指定值的搜尋。
filterTermCaseSensitive	字串	使用者可選擇搜尋時是否區分大小寫。此變數中的值用來識別搜尋有否區分大小寫。值必須是 yes (搜尋時區分大小寫) 或 no (搜尋時不區分大小寫)。
filterTermOperator	字串	使用者可選擇要以 like 或 equal 做為搜尋運算子。此變數中的值用來儲存使用者的選擇。值必須是 LIKE (使用 like 運算子) 或 EQUAL (使用 equal 運算子)。
filterType	字串	使用者可指定三種搜尋準則類型：All、Any 或 Exact Phrase。此變數中的值用來儲存使用者的搜尋準則。值必須是 ALL (所有搜尋準則)、ANY (任何搜尋準則) 或 EXACT (精確詞組準則)。
isListPriceOn	字串	型錄搜尋 Bean 使用者可選擇要顯露標價或標準價。如果要顯露標價，請將此變數設為 yes；如果要顯露標準價，請將此變數設為 no。依預設，搜尋 Bean 是顯露標準價。
manufacturer	字串	此變數的值用於製造商名稱的搜尋上。
manufacturerCaseSensitive	字串	使用者可選擇搜尋時是否區分大小寫。此變數中的值用來識別搜尋有否區分大小寫。值必須是 yes (搜尋時區分大小寫) 或 no (搜尋時不區分大小寫)。
manufacturerOperator	字串	使用者可選擇要以 like 或 equal 做為搜尋運算子。此變數中的值用來儲存使用者的選擇。值必須是 LIKE (使用 like 運算子) 或 EQUAL (使用 equal 運算子)。
manufacturerPartNum	字串	此變數的值用於製造商之產品編號的搜尋上。

表 7. (繼續)

manufacturerPartNumCaseSensitive	字串	使用者可選擇搜尋時是否區分大小寫。此變數中的值用來識別搜尋有否區分大小寫。值必須是 yes (搜尋時區分大小寫) 或 no (搜尋時不區分大小寫)。
manufacturerPartNumOperator	字串	使用者可選擇要以 like 或 equal 做為搜尋運算子。此變數中的值用來儲存使用者的選擇。值必須是 LIKE (使用 like 運算子) 或 EQUAL (使用 equal 運算子)。
maxPrice/minPrice	字串	這些變數的值用於價格範圍的搜尋上。
pageSize	字串	此變數的值指出每頁要顯示多少搜尋結果列。
price	字串	此變數的值用於價格的搜尋上。
priceOperator	字串	使用者可選擇下列一個運算子做為搜尋運算子：=、<、>、!=、<=、>=。此變數中的值用來儲存使用者的選擇。其必須是下列之一：EQUAL、NOTEQUAL、GREATER、LESS、GREATER_EQUAL、LESS_EQUAL。
qtyAvailable	字串	此變數的值用於產品或項目庫存的搜尋上。
qtyAvailableOperator	字串	使用者可選擇下列一個運算子做為搜尋運算子：=、<、>、!=、<=、>=。此變數中的值用來儲存使用者的選擇。其必須是下列之一：EQUAL、NOTEQUAL、GREATER、LESS、GREATER_EQUAL、LESS_EQUAL。
qtyMeasure	字串	此變數的值用於數量尺度的搜尋上。
qtyMeasureCaseSensitive	字串	使用者可選擇搜尋時是否區分大小寫。此變數中的值用來識別搜尋有否區分大小寫。值必須是 yes (搜尋時區分大小寫) 或 no (搜尋時不區分大小寫)。
qtyMeasureOperator	字串	使用者可選擇要以 like 或 equal 做為搜尋運算子。此變數中的值用來儲存使用者的選擇。值必須是 LIKE (使用 like 運算子) 或 EQUAL (使用 equal 運算子)。
resultCount	字串	此變數含有所傳回的搜尋結果總數。
resultType	字串	商家可指出要在搜尋結果中顯示「產品」或「項目」或「產品與項目兩者」。此變數中的值用來儲存此值。值必須是 1 (僅產品)、2 (僅項目) 或 3 (產品與項目兩者)。
searchTerm	字串	此變數的值用於字組的搜尋上。
searchTermCaseSensitive	字串	使用者可選擇搜尋時是否區分大小寫。此變數中的值用來識別搜尋有否區分大小寫。值必須是 yes (搜尋時區分大小寫) 或 no (搜尋時不區分大小寫)。
searchTermOperator	字串	使用者可選擇要以 like 或 equal 做為搜尋運算子。此變數中的值用來儲存使用者的選擇。值必須是 LIKE (使用 like 運算子) 或 EQUAL (使用 equal 運算子)。
searchTermScope	整數	使用者可將搜尋範圍限定在：「名稱之一」、「名稱與簡短說明」、「名稱、簡短說明與詳細說明」或「關鍵字」上。此變數中的值用來儲存使用者的選擇。值必須是 1 (名稱與簡短說明)、2 (僅名稱)、3 (名稱、簡短說明與詳細說明)、4 (關鍵字)。

表 7. (繼續)

searchType	字串	使用者可指定三種搜尋準則類型：All、Any 或 Exact Phrase。此變數中的值用來儲存使用者的搜尋準則。值必須是 ALL (所有搜尋準則)、ANY (任何搜尋準則) 或 EXACT (精確詞組準則)。
sku	字串	此變數的值用於 SKU 的搜尋上。
skuCaseSensitive	字串	使用者可選擇搜尋時是否區分大小寫。此變數中的值用來識別搜尋有否區分大小寫。值必須是 yes (搜尋時區分大小寫) 或 no (搜尋時不區分大小寫)。
skuOperator	字串	使用者可選擇要以 like 或 equal 做為搜尋運算子。此變數中的值用來儲存使用者的選擇。值必須是 LIKE (使用 like 運算子) 或 EQUAL (使用 equal 運算子)。

型錄搜尋資料庫直欄

型錄搜尋 Bean 所能搜尋的資料庫直欄清單：

表 8.

定義於 Bean 介面中的可搜尋屬性	表格	直欄	支援布林表示式
種類群組 - 所有名稱和說明	CatGrpDesc	NAME LONGDESCRIPTION SHORTDESCRIPTION	是
種類群組 - 僅標題	CatGrpDesc	NAME	是
種類群組 - 標題與說明 (建議的預設值)	CatGrpDesc	NAME SHORTDESCRIPTION	是
種類語言 ID	CatGrpDesc	LANGUAGE_ID	否
種類識別碼	CatGpEnRel	CATGROUP_ID	否
型錄項目說明 - 所有名稱和說明	CatEntDesc	NAME LONGDESCRIPTION SHORTDESCRIPTION	是
型錄項目 - 僅標題	CatEntDesc	NAME	是
型錄項目 - 標題與說明 (建議的預設值)	CatEntDesc	NAME SHORTDESCRIPTION	是
型錄項目語言 ID	CatEntDesc	LANGUAGE_ID	否
型錄項目製造商名稱	CatEntry	MFNAME	否
型錄項目製造商產品編號	CatEntry	MFPARTNUMBER	否
型錄項目 SKU	CatEntDesc	PARTNUMBER	否
價格	標價/報價	PRICE	否
價格範圍	標價/報價	PRICE	否
貨幣	標價/報價	CURRENCY	否
可供應量	InvStVw/ StoreInv	QTYAVAILABLE	否
數量尺度	InvStVw/ StoreInv	QUANTITYMEASURE	否

Bean 的使用者可自訂 Bean，以顯露像是多元屬性等更多的直欄。有關自訂的進一步資訊，請參閱線上說明中之 ExtendedCatEntrySearchListDataBean 類別的 JavaDoc。

第7章 優待券

本章中所要討論的元素，代表 WebSphere Commerce Accelerator 中之「優待券」元件的使用者介面。這些元素有助於建立與維護賣方或商家在執行每日所負責之相關商業工作時，所需的優待券、優待券優惠以及其它作業。「優待券」元件所含的元素如下：

- 「優待券」精靈
- 「優待券」筆記本

自訂範例

下列範例說明如何自訂 WebSphere Commerce Accelerator 中的這個部份。

實務 1：在建立優待券折扣時新增新資訊

施行概觀

本實務會以必要步驟引導您自訂「優待券」工具，以記錄優待券中的其它資訊，像是：何人建立優待券折扣等。

自訂步驟

1. 施行新 **CustomCreateCouponDiscountCmdImpl**（自作業指令施行 **CreateCouponDiscountCmdImpl** 延伸而來），並施行 **CreateCouponDiscountCmd**。
2. 使用 CALCODE 表格的 FIELD1 直欄，儲存建立此優待券折扣之使用者的 `userId`。
3. 將新作業指令登錄在 CMDREG 表格中。假設客戶商店 ID 為 1，請執行下列的 SQL 陳述式。

```
insert into cmdreg( storeent_id,interfacename, classname)
  values(1,'com.ibm.commerce.tools.ecoupon.CreateCouponDiscountCmd',
        'com.ibm.commerce.tools.ecoupon.CustomCreateCouponDiscountCmdImpl')
```

4. 更新 **CustomCreateCouponDiscountCmdImpl**。在指令中施行 `performExecute()` 方法：

```
public void performExecute() throws ECSystemException, ECException {
    super.performExecute();
    /** 從指令環境定義取得 userId
    將 userId 儲存在 calcode 表格中
    **/
}
```

第8章 折扣

本章中所要討論的元素，代表 WebSphere Commerce Accelerator 中之「折扣」元件的使用者介面。這些元素有助於建立與維護賣方或商家每日執行的折扣與其它動作。「折扣」元件所含的元素如下：

- 「折扣」精靈
- 「折扣」筆記本

自訂範例

下列範例說明如何自訂 WebSphere Commerce Accelerator 中的「折扣」元件。

實務 1：在建立折扣時新增其它資訊

施行概觀

本實務是以必要步驟引導您在建立折扣時新增其它資訊。本例會假設您想建立一項何人建立折扣的審核追蹤。本實務不會要求您提交新參數。

自訂步驟

1. 施行新 **CustomCreateDiscountCmdImpl**（自作業指令施行 **CreateDiscountCmdImpl** 延伸而來），並施行 **CreateDiscountCmd**。
2. 使用 **CALCODE** 表格的 **field1** 直欄，儲存建立此折扣之使用者的 **userId**。
3. 將新作業指令登錄在 **CMDREG** 表格中。假設客戶商店 ID 為 1，請執行下列的 SQL 陳述式：

```
insert into cmdreg( storeent_id,interfacename, classname)
  values(1,'com.ibm.commerce.tools.promotions.CreateDiscountCmd',
        'com.ibm.commerce.tools.promotions.CustomCreateDiscountCmdImpl')
```

4. 在 **CustomCreateDiscountCmdImpl** 中，施行 **performExecute()** 方法：

```
public void performExecute() throws ESystemException, ECEException {
    super.performExecute();
    /** 從指令環境定義取得使用者 ID
        將使用者 ID 儲存在 calcode 表格中
    **/
}
```

實務 2：變更預設行為

施行概觀

由於本實務改變了「折扣」精靈的預設行為，因此您可以指定套用在訂單上的折扣順序。本實務亦會要求您透過 URL 傳遞新參數。這會要求您在「折扣」精靈的第一個畫面中新增一個欄位，以取得額外的資料。

自訂步驟

1. 執行下列步驟，以更新因應「折扣」精靈的 JSP 檔：

- a. 變更至下列目錄：

▶ **AIX** /usr/WebSphere/CommerceServer/wcstool.war/tools/promotions/

▶ **400**

/QIBM/UserData/WebSphere/CommerceServer/wcstool.war/tools/promotions/

▶ **Linux** /opt/WebSphere/CommerceServer/wcstool.war/tools/promotions/

▶ **Solaris** /opt/WebSphere/CommerceServer/wcstool.war/tools/promotions/

▶ **Windows** drive:\WebSphere\CommerceServer\wcstool.war\tools\promotions\

- b. 將 DiscountWizardWelcome.jsp 複製到 MyDiscountWizardWelcome.jsp
- c. 更新 MyDiscountWizardWelcome.jsp。在套表區塊中，新增下列程式碼：
- ```
<input name="sequence" type="TEXT" size=3 MAXLENGTH=3> Sequence
```
- d. 將項目儲存到組織架構中。在您的 JSP 中會有一個 JavaScript 函數 savePanelData，請在此函數中加入下行：
- ```
parent.put("sequence", document.welcomeForm.sequence.value);
```
- e. 將新項目登錄在 VIEWREG 表格中。舉例來說，如果客戶商店 ID 為 1，請執行下列的 SQL 陳述式：

```
insert into viewreg (viewname, devicefmt_id, storent_id, interfacename,
                    classname, properties, https, internal)
values ('CusDiscountWizWelcomeView', -1, 0,
       'com.ibm.commerce.tools.command.ToolsForwardViewCommand',
       'com.ibm.commerce.tools.command.ToolsForwardViewCommandImpl',
       'docname=tools/promotions/extend/CusDiscountWizardWelcome.jsp', 0, 1)
```

2. 施行新 **DiscountSaveCmdCustomImpl**（自控制程式指令施行 **DiscountSaveCmdImpl** 延伸而來），並施行 **DiscountSaveCmd**。

3. 將此施行登錄在 CMDREG 表格中。假設客戶商店 ID 為 1，請執行下列的 SQL 陳述式。

```
insert into cmdreg(storent_id,interfacename,classname,target)
values (1,'com.ibm.commerce.tools.promotions.DiscountSaveCmd',
       'com.ibm.commerce.tools.promotions.DiscountSaveCmdCustomImpl')
```

4. 在 **DiscountSaveCmdCustomImpl** 指令中，使用下列的範例程式碼，以施行兩個方法：

```
Public void validateParameters() {
    super.validateParameters();
    /*
    從 requestProperty 取得新參數
    */
}
Public void customMethod() {
    super.customMethod();
    /*
    此處為您的動作。
    */
}
```

第9章 RFQ 回應

自訂範例

在本文件中，我們將提供三個自訂範例。分別是：

- 複製回應
- 將「已撤回」狀態從 RFQ 回應的循環週期中刪除
- 在建立期間輸入回應的說明資訊

實務 1：複製回應

本範例自訂是在「RFQ 回應」工具中新增功能，以便讓使用者可根據所選的 RFQ 回應建立一項複製的 RFQ 回應。此自訂工作包括新增**複製**按鈕，以便在使用者點選時，即會提示使用者為複製的回應輸入一個唯一名稱。當建立好後，新回應會處在初稿狀態。

施行概觀

擷取此回應的所有詳細資訊，並使用所擷取的資訊來建立新回應。由於回應名稱不能和現行綱目中的相同，因此必須提供一個頁面，讓使用者可提供另一名稱。如此有助於在此頁面中準備所有資訊，並將之傳給 **RFQResponseCopyCmd** 指令，以產生一個「建立回應」事件。接著 UBF 呼叫 **RFQResponseCreateCmd** 指令，以建立回應。

自訂步驟

1. 執行下列步驟，以新增 **RFQResponseCopyCmd** 介面與其施行 **RFQResponseCopyCmdImpl**：

- a. 新增新介面 **RFQResponseCopyCmd**。以下是定義介面的程式碼：

```
public interface RFQResponseCopyCmd extends ControllerCommand{
    public final static String NAME =
        "com.ibm.commerce.rfq.commands.RFQResponseCopyCmd";
    public final static String defaultCommandClassName =
        "com.ibm.commerce.rfq.commands.RFQResponseCopyCmdImpl";
}
```

- b. 延伸 `com.ibm.commerce.ubf.commands.ToolsBusinessFlowEventCmdImpl` 並施行 **RFQResponseCopyCmd**，以新增指令施行 **RFQResponseCopyCmdImpl**。此指令會使用經由對話框提交的資料，以觸發「建立回應」事件。此事件會讓 UBF 呼叫 **RFQResponseCreateCmdImpl**，以建立新回應。以下是定義指令的程式碼：

```
public void performExecute() throws ECEException
{
    TypedProperty parms = null;
    BusinessFlowEventData data = null;
    try {
        parms = getRequestProperties();
        parms.put(BusinessFlowConstants.EC_FLOWID, RFQConstants.EC_FLOW_RESPONSE_ID);
        parms.put(BusinessFlowConstants.EC_BUSINESS_FLOW_EVENT_IDENTIFIER,
            "createRFQResponse");

        data = new BusinessFlowEventData(getCommandContext(), parms);
        BusinessFlowEvent event = new BusinessFlowEvent(data, true);
```

```

parms = data.getResponseProperties();
responseProperties = new TypedProperty();

for (Enumeration pns = parms.keys(); pns.hasMoreElements();) {
String paramName = (String) pns.nextElement();
if (paramName.equals(EConstants.EC_VIEWTASKNAME))
responseProperties.put(EConstants.EC_VIEWTASKNAME, "DialogNavigation");
else if (paramName.equals(UIProperties.SUBMIT_FINISH_MESSAGE))
responseProperties.put(UIProperties.SUBMIT_FINISH_MESSAGE,
"The response was copied successfully!");
else {
Object newVal = parms.get(paramName);
responseProperties.put(paramName, newVal);
}
}
} catch (EException e) {
parms = e.getErrorProperties();
responseProperties = new TypedProperty();
responseProperties.put(EConstants.EC_VIEWTASKNAME, "DialogNavigation");
responseProperties.put(UIProperties.SUBMIT_ERROR_STATUS, "ERROR");
responseProperties.put(UIProperties.SUBMIT_ERROR_MESSAGE,
"Copying response failed, please input another name.");
throw new EApplicationException(new EMessage(EMessageSeverity.INFO,
EMessageType.USER, "_ERR_RESPONSE_COPY",
"com.ibm.commerce.tools.rfq.properties.RFQMessages"),
this.getClass().getName(), "",
"DialogNavigation",responseProperties);
}
}
}

```

- c. 將 **RFQResponseCopy** 登錄在 URLREG 表格中。請執行下列 SQL，以登錄指令：

```

insert into urlreg values('RFQResponseCopy',0,'com.ibm.commerce.
rfq.commands.RFQResponseCopyCmd',0,null,null,1);

```

2. 新增 JSP 讓使用者可輸入回應的名稱，以便擷取和回應有關的所有資訊：
- a. 此 JSP 檔會提供一個文字欄位供使用者提供新回應名稱。以下是文字欄位的原始程式碼：

```

<FORM name="responseCopyForm">
<table>
<tr><td>
<label><%= rfqNLS.get("name") %> <%= rfqNLS.get("required") %></label>
</td></tr>
<tr><td>
<input type="Text" name="response_name" size="30" maxlength="200">
</td></tr>
</table>
</FORM>

```

當使用者按一下**確定**時，下列程式碼即會儲存輸入欄位中的回應名稱：

```

function savePanelData() {
var form = document.responseCopyForm;
parent.put("<%= RFQConstants.EC_RFQ_RESPONSE_NAME %>",form.response_name.value);
return true;
}

```

當起始設定這個 JSP 檔時，下列程式碼會擷取回應的所有相關資訊，並加以儲存。資訊將提供給 **RFQResponseCopyCmdImpl**。

```

function initializeState(){
parent.setContentFrameLoaded(false);
parent.put("<%= RFQConstants.EC_RFQ_REQUEST_ID %>" ,<%= RequestId %>);
parent.put("<%= RFQConstants.EC_RFQ_RESPONSE_REMARK%>",

```



```

        "<%= UIUtil.toJavaScript((String)RFQres.getRemarks())%>");

var rfqCommentsArray = new Array() ;
<%
RFQResCommentsPair[] commentsPair =
    RFQResProdHelper.getRFQLevelCommentsPair(RequestId,ResponseId,null);
for (int index=0; commentsPair != null && index <commentsPair.length;
    index++){
    %>

rfqCommentsArray[<%=index%>] = new Object();
    rfqCommentsArray[<%=index%>].<%=RFQConstants.EC_REQUEST_TC_ID%>="<%=
        commentsPair[index].getRFQ_TC_ID() %>";
    rfqCommentsArray[<%=index%>].<%=RFQConstants.
EC_TC_RFQ_LEVEL_COMMENTS%>="
        <%= UIUtil.toJavaScript((String)commentsPair
[index].getRFQ_value())%>";
    rfqCommentsArray[<%=index%>].<%=RFQConstants.
EC_ATTR_MANDATORY%>= "
        <%= commentsPair[index].getMandatory() %>";
    rfqCommentsArray[<%=index%>].<%=RFQConstants.
EC_ATTR_CHANGEABLE%>= "
        <%= commentsPair[index].getChangeable()%>";
    rfqCommentsArray[<%=index%>].<%=RFQConstants.
EC_ATTR_RES_CMMENTS_VALUE%>="
        <%= UIUtil.toJavaScript((String)commentsPair[index]
.getRes_value())%>";
    %>
parent.put("<%= RFQConstants.EC_TC_RFQ_LEVEL_COMMENTS %>",rfqCommentsArray);

var ProductsArray = new Array();
<%
RFQResNewProd[] ResPros = RFQResProdHelper.getResAllProds(RequestId,
ResponseId,langId);
int i=0;
for(;ResPros != null && i < ResPros.length;i++){
    %>
    ProductsArray[<%=i%>] = new Object();
    ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_CATENTRYID%>="
        <%=ResPros[i].getCatentry_id() %>";
    ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRICE%>= "
        <%=ResPros[i].getPrice() %>";
    ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_QUANTITY%>="
        <%=ResPros[i].getQuantity() %>";
    ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_CURRENCY%>= "
        <%=ResPros[i].getCurrency() %>";
    ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_UNIT%>="
        <%=ResPros[i].getUnit() %>";
    ProductsArray[<%=i%>].<%=RFQConstants.EC_ATTR_PRODUCT_COMMENTS%>=
new Array();
    ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>=
new Array();
    %>
RFQResProdAttributes[] resAttrs=RFQResProdHelper.getResAllAttributes(RequestId,
ResponseId, ResPros[i].getCatentry_id(), langId,
    rfqNLS.get("valuedelim").toString());

int m=0,n=0;
for (int j=0;resAttrs != null && j<resAttrs.length>

ProductsArray[<%=i%>].<%=RFQConstants.EC_ATTR_PRODUCT_COMMENTS%>[<%=m++%>]
    = new Object;
ProductsArray[<%=i%>].<%=RFQConstants.EC_ATTR_PRODUCT_COMMENTS%>[<%=m-1%>].
    <%=RFQConstants.EC_ATTR_PATTRID%> = "<%= resAttrs[j].
getPAttribute_id()%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_ATTR_PRODUCT_COMMENTS%>[<%=m-1%>].
    <%=RFQConstants.EC_ATTR_NAME%> = "<%= UIUtil.toJavaScript
((String)resAttrs[j].getName())%>";

```

```

ProductsArray[<%=i%>].<%=RFQConstants.EC_ATTR_PRODUCT_COMMENTS%>[<%=m-1%>].
    <%=RFQConstants.EC_ATTR_VALUE%> = "<%= UIUtil.toJavaScript
((String)resAttrs[j].getRes_value())%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_ATTR_PRODUCT_COMMENTS%>[<%=m-1%>].
    <%=RFQConstants.EC_ATTR_MANDATORY%> = "<%= resAttrs[j].
getMandatory()%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_ATTR_PRODUCT_COMMENTS%>[<%=m-1%>].
    <%=RFQConstants.EC_ATTR_CHANGEABLE%> = "<%= resAttrs[j].
getChangeable()%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_ATTR_PRODUCT_COMMENTS%>[<%=m-1%>].
    <%=RFQConstants.EC_REQUEST_TC_ID%> = "<%= resAttrs[j].
getReq_tc_id()%>";

<%}else{%>
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n++%>]
    = new Object;
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_ATTR_PATRID%> = "<%= resAttrs[j].
getPAttribute_id()%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_ATTR_NAME%> = "<%= UIUtil.toJavaScript
((String)resAttrs[j].getName())%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_ATTR_VALUE%> = "<%= UIUtil.toJavaScript
((String)resAttrs[j].getRes_value())%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_ATTR_VALUEDELIM%> = "<%=rfqNLS.get
("valuedelim")%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_ATTR_MANDATORY%> = "<%= resAttrs[j].
getMandatory()%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_ATTR_CHANGEABLE%> = "<%= resAttrs[j].
getChangeable()%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_REQUEST_TC_ID%> = "<%= resAttrs[j].
getReq_tc_id()%>";
    ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
        <%=RFQConstants.EC_ATTR_OPERATOR%> = "<%= resAttrs[j].
getOperator_id()%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_ATTR_UNIT%> = "<%= resAttrs[j].getUnit()%>";
ProductsArray[<%=i%>].<%=RFQConstants.EC_OFFERING_PRODATTRLIST%>[<%=n-1%>].
    <%=RFQConstants.EC_ATTR_TYPE%> = "<%= resAttrs[j].getType()%>";

<% }

}>
<%
}
%>

parent.put("<%= RFQConstants.EC_OFFERING_PRODITEM %>",ProductsArray);

parent.setContentFrameLoaded(true);
}
.
.
.
<BODY class="content" onLoad="initializeState()">

```

- b. 執行下列 SQL 陳述式，將檢視畫面指令登錄在會映射新 JSP 檔的 VIEWREG 表格中。

```
insert into viewreg values('RFQRspDuplicateDialog', -1, 0,
    'com.ibm.commerce.tools.command.ToolsForwardViewCommand',
    'com.ibm.commerce.tools.command.ToolsForwardViewCommandImpl',
    'docname=tools/rfq/rfq_response_duplicate_dialog.jsp', null, 0, null, 0);
```

- c. 在下列目錄中新增新 XML 檔 rfq_response_duplicate_dialog.xml :

▶ **AIX** /usr/WebSphere/CommerceServer/xml/tools/rfq
▶ **400** /QIBM/UserData/WebSphere/CommerceServer/xml/tools/rfq
▶ **Linux** /opt/WebSphere/CommerceServer/xml/tools/rfq
▶ **Solaris** /opt/WebSphere/CommerceServer/xml/tools/rfq
▶ **Windows** drive:\WebSphere\CommerceServer\xml\tools\rfq

finishURL 為 **RFQRResponseCopy**，其會映射所登錄的 URL。

此對話框的畫面為 **RFQRspDuplicateDialog**，其會映射您所登錄的檢視畫面指令。XML 檔的程式碼如下：

```
<?xml version="1.0" ?>

<dialog resourceBundle="utf.utfNLS"
    windowTitle="dupliateDialog_Title"
    finishURL="RFQRResponseCopy">

    <panel name="general"
        url="/webapp/wcs/tools/servlet/RFQRspDuplicateDialog"
        parameters="responseId"
        hasFinish="YES"/>
    <jsFile src="/wcs/javascript/tools/rfq/rfq_response_duplicate_dialog.js"/>
</dialog>
```

- d. 將此 XML 登錄在 resource.xml 中。

- 1) 備份下列目錄中的 resource.xml :

▶ **AIX** /usr/WebSphere/CommerceServer/xml/tools/rfq
▶ **400** /QIBM/UserData/WebSphere/CommerceServer/xml/tools/rfq
▶ **Linux** /opt/WebSphere/CommerceServer/xml/tools/rfq
▶ **Solaris** /opt/WebSphere/CommerceServer/xml/tools/rfq
▶ **Windows** drive:\WebSphere\CommerceServer\xml\tools\rfq

- 2) 在 resource.xml 檔中加入下列兩行：

```
<XML name="rfqRspDuplicateDialog"
    file="rfq/rfq_response_duplicate_dialog.xml"/>
```

- e. 在下列目錄中新增新 JavaScript 檔 rfq_response_duplicate_dialog.js :

▶ **AIX** /usr/WebSphere/CommerceServer/web/javascript/tools/rfq
▶ **400** /QIBM/UserData/WebSphere/CommerceServer/web/javascript/tools/rfq
▶ **Linux** /opt/WebSphere/CommerceServer/web/javascript/tools/rfq
▶ **Solaris** /opt/WebSphere/CommerceServer/web/javascript/tools/rfq
▶ **Windows** drive:\WebSphere\CommerceServer\web\javascript\tools\rfq

JavaScript 檔含有一些供對話框使用的 JavaScript 函數。以下是 JavaScript 檔的原始程式碼：

```

function submitErrorHandler (errMessage){
    self.CONTENTS.alertDialog(errMessage);
}

function submitFinishHandler (finishMessage){
    alertDialog(finishMessage);
    top.goBack();
}






function submitCancelHandler(){
    top.goBack();
}

```

3. 將此函數整合到 RFQ 回應清單頁面中。

a. 在回應清單頁面中新增複製按鈕。亦即，按如下所示變更 XML 檔：

註：備份 XML 檔 rfq_response_list.xml；此檔案可在下列目錄中找到：

	/usr/WebSphere/CommerceServer/xml/tools/rfq
	/QIBM/UserData/WebSphere/CommerceServer/xml/tools/rfq
	/opt/WebSphere/CommerceServer/xml/tools/rfq
	/opt/WebSphere/CommerceServer/xml/tools/rfq
	drive:\WebSphere\CommerceServer\xml\tools\rfq

在 rfq_response_list.xml 中之 <button>... </button> 節點的尾端加入下行：

```

<menu name="duplicate"
    action="basefrm.duplicateRes()"
    selection="single">
</menu>

```

b. 在 rfq_response_list.jsp 中新增新 JavaScript 函數 getDuplicateBCT() 與 duplicateRes()：

```

function getDuplicateBCT() {
    return "<%= rfqNLS.get("duplicate") %>";
}

function duplicateRes(){
    if(isButtonDisabled(parent.buttons.buttonForm.duplicateButton))
        return;
    var checkedEntries = parent.getChecked().toString();
    var parms = checkedEntries.split(';');
    var resId = parms[0];
    top.setContent(getDuplicateBCT(), '/webapp/wcs/tools/servlet/DialogView?
        XMLFile=rfq.rfqRspDuplicateDialog&responseId='+resId,true)
}

```

c. 在 rfq_response_list.jsp 中新增一個 JavaScript 函數，並在對應 RFQ 的狀態並非 ACTIVE（作用中）時，呼叫它以停用複製按鈕。在 rfq_response_list.jsp 中新增下列函數，並在呼叫 DisableNewButton() 後呼叫它：

```

function DisableDuplicateButton()
{
    var reqState;
    var active=<%= com.ibm.commerce.utf.helper.UTFOtherConstants.
    EC_STATE_ACTIVE %>;
    reqState="<%=rfqState%>";
}

```

```

        if (reqState !=active){
            parent.hideButton('duplicate');
        }
    }
}

```

4. 在 RFQMessages_en_US.properties 檔中新增新訊息；此檔案位於下列目錄中：

▶ **AIX** /usr/WebSphere/CommerceServer/properties/com/ibm/commerce/
tools/rfq/properties

▶ **400**

/QIBM/UserData/WebSphere/CommerceServer/properties/com/ibm/commerce/
tools/rfq/properties

▶ **Linux** /opt/WebSphere/CommerceServer/properties/com/ibm/commerce/
tools/rfq/properties

▶ **Solaris** /opt/WebSphere/CommerceServer/properties/com/ibm/commerce/
tools/rfq/properties

▶ **Windows** drive:\WebSphere\CommerceServer\properties\com\ibm\commerce\
tools\rfq\properties

在檔案中加入下行。當複製失敗時將會顯示此訊息：

_ERR_RESPONSE_COPY = 複製回應失敗。

實務 2：將「已撤回」狀態從 RFQ 回應的循環週期中刪除

當使用者撤回回應時，回應會從作用中狀態變成已撤回狀態。如果使用者希望在撤回回應後，該回應會直接設成初稿狀態，您必須自訂函數，亦即，將已撤回狀態從回應的循環週期中刪除。

施行概觀

在此項自訂中，首先必須將已撤回狀態從回應狀態機中刪除。一旦已無已撤回狀態，也應移除回應清單頁面之檢視畫面清單內的已撤回狀態。

自訂步驟

1. 變更 UBFStateMachine.xml 與 UBFStateMachine_en_US.xml，然後再重新載入。備份下列目錄中的 UBFStateMachine.xml 與 UBFStateMachine_en_US.xml：

▶ **AIX** /usr/WebSphere/CommerceServer/xmlloadutility/
businessfollows/xml

▶ **400** /QIBM/UserData/WebSphere/CommerceServer/xmlloadutility/
businessfollows/xml

▶ **Linux** /opt/WebSphere/CommerceServer/xmlloadutility/
businessfollows/xml

▶ **Solaris** /opt/WebSphere/CommerceServer/xmlloadutility/
businessfollows/xml

▶ **Windows** drive:\WebSphere\CommerceServer\xmlloadutility\businessfollows\xml

在這兩個檔案中進行下列變更：

- 將 retractRFQResponse 轉移的目標狀態從 RETRACTED（已撤回）改為 DRAFT（初稿）。

- 刪除 RETRACTED 狀態的相關資訊以及此狀態的所有轉移。

下列程式碼範例說明 UBFStateMachine.xml 檔中所需進行的變更：

變更之前

```

<Flow identifier="RFQ response process totally" priority="2">
  <State identifier="ACTIVE">
    <Transition eventidentifier="retractRFQResponse" approval="0"
priority="2">
      <Action
        interface="com.ibm.commerce.rfq.commands.RFQResponse
ChangeStateAdvCmd">
        <AccessControlGuard actiongroup="RFQResponseManage">
          <TargetState identifier="RETRACTED"/>
        </Transition>
      </State>
    <State identifier="RETRACTED">
      <Transition eventidentifier="changeRFQResponseToDraft"
approval="0" priority="1">
        <Action
          interface="com.ibm.commerce.rfq.commands.RFQResponse
ChangeStateAdvCmd">
          <AccessControlGuard actiongroup="RFQResponseManage">
            <TargetState identifier="DRAFT"/>
          </Transition>
          <Transition eventidentifier="cancelRFQResponse"
approval="0" priority="2">
            <Action
              interface="com.ibm.commerce.rfq.commands.RFQResponse
ChangeStateAdvCmd">
              <AccessControlGuard actiongroup="RFQResponseManage">
                <TargetState identifier="CANCELLED"/>
              </Transition>
              <Transition eventidentifier="closeRFQRequest" approval="0"
priority="3">
                <Action
                  interface="com.ibm.commerce.rfq.commands.RFQResponse
ChangeStateBaseCmd">
                  <AccessControlGuard actiongroup="RFQManage">
                    <TargetState identifier="CANCELLED"/>
                  </Transition>
                  <Transition eventidentifier="cancelRFQRequest" approval="0"
priority="4">
                    <Action
                      interface="com.ibm.commerce.rfq.commands.RFQResponse
ChangeStateBaseCmd">
                      <AccessControlGuard actiongroup="RFQManage"/>
                      <TargetState identifier="CANCELLED"/>
                    </Transition>
                  </State>
                </Flow>

```

變更之後

```

<Flow identifier="RFQ response process totally" priority="2">
  <State identifier="ACTIVE">
    <Transition eventidentifier="retractRFQResponse" approval="0"
priority="2">
      <Action
        interface="com.ibm.commerce.rfq.commands.RFQResponse
ChangeStateAdvCmd">
        <AccessControlGuard actiongroup="RFQResponseManage"/>
        <TargetState identifier="DRAFT"/>
      </Transition>
    </State>
  </Flow>

```

按如下所示變更 UBFStateMachine.xml :

變更之前

```
<State identifier="ACTIVE">
  <TransitionDesc eventidentifier="closeRFQRequest"
    transitiondescription=
      "當關閉要求時，將回應狀態改為「評估中」"
    eventdescription="closeRFQRequest">
    <TargetState identifier="IN-EVALUATION"/>
  </TransitionDesc>
  <TransitionDesc eventidentifier="retractRFQResponse"
    transitiondescription="撤回回應"
    eventdescription="retractRFQResponse">
    <TargetState identifier="RETRACTED"/>
  </TransitionDesc>
  <TransitionDesc eventidentifier="cancelRFQRequest"
    transitiondescription="當取消 RFQ 時取消回應"
    eventdescription="cancelRFQRequest">
    <TargetState identifier="CANCELLED"/>
  </TransitionDesc>
</State>
<State identifier="RETRACTED">
  <TransitionDesc eventidentifier="changeRFQResponseToDraft"
    transitiondescription="將被撤回的回應改為初稿"
    eventdescription="changeRFQResponseToDraft">
    <TargetState identifier="DRAFT"/>
  </TransitionDesc>
  <TransitionDesc eventidentifier="cancelRFQResponse"
    transitiondescription="取消回應"
    eventdescription="cancelRFQResponse">
    <TargetState identifier="CANCELLED"/>
  </TransitionDesc>
  <TransitionDesc eventidentifier="closeRFQRequest"
    transitiondescription="當關閉 RFQ 時取消回應"
    eventdescription="closeRFQRequest">
    <TargetState identifier="CANCELLED"/>
  </TransitionDesc>
  <TransitionDesc eventidentifier="cancelRFQRequest"
    transitiondescription="當取消 RFQ 時取消回應"
    eventdescription="cancelRFQRequest">
    <TargetState identifier="CANCELLED"/>
  </TransitionDesc>
</State>
```

變更之後

```
<State identifier="ACTIVE">
  <TransitionDesc eventidentifier="closeRFQRequest"
    transitiondescription=
      "當關閉要求時，將回應狀態改為「評估中」"
    eventdescription="closeRFQRequest">
    <TargetState identifier="IN-EVALUATION"/>
  </TransitionDesc>
  <TransitionDesc eventidentifier="retractRFQResponse"
    transitiondescription="撤回回應"
    eventdescription="retractRFQResponse">
    <TargetState identifier="DRAFT"/>
  </TransitionDesc>
  <TransitionDesc eventidentifier="cancelRFQRequest"
    transitiondescription="當取消 RFQ 時取消回應"
    eventdescription="cancelRFQRequest">
    <TargetState identifier="CANCELLED"/>
  </TransitionDesc>
</State>
```

- 由於回應的循環週期中已無已撤回狀態，請在 `rfq_response_list.xml` 與 `rfq_response_state_list.xml` 中刪除下列各行。此處的狀態包括：`draft`、`cancelled`、`pendingapproval`、`rejected`、`active`、`inevaluation`、`win`、`lost`、`wincomplete` 與 `lostcomplete`。

註：此處的折行僅爲了方便閱讀。

```
<view name="resretracted"
      action="top.setContent(basefrm.getPageTitle(),
        '/webapp/wcs/tools/servlet/NewDynamicListView?
        ActionXMLFile=rfq.rfqresponseretractedlist&
        cmd=RFQResponseList&rfqId='+basefrm.getRfqId(),false)"/>
```

註：

- 本範例說明如何刪除狀態。如果您想新增狀態，請以相反動作執行類似步驟：
 - 在狀態機 XML 檔中新增新狀態轉移，然後重新載入。
 - 將新檢視畫面新增到回應清單頁面的檢視畫面中。
 有關重新載入狀態機的資訊，請參閱 UBF 線上說明。

實務 3：在建立期間輸入回應的說明資訊

如果使用者想在建立 RFQ 的回應期間輸入一些說明文字，您必須在「RFQ 建立」精靈的第一個畫面中新增一個文字欄位。

施行概觀

在「RFQ 回應建立」精靈的第一個畫面中新增一個新文字欄位。由於 `RFQResponseCreateCmdImpl` 不會處理多出的輸入，請延伸 `RFQResponseCreateCmd` 與 `RFQResponseCreateCmdImpl`，以建立新介面 `MyRFQResponseCreateCmd` 以及其指令施行 `MyRFQResponseCmdImpl`。

自訂步驟

- 新增介面 `MyRFQResponseCreateCmd` 與其施行 `MyRFQResponseCreateCmdImpl`。

- 新增新介面 `MyRFQResponseCreateCmd`。以下是介面的程式碼：

```
public interface MyRFQResponseCreateCmd extends RFQResponseCreateCmd {
    public final static String NAME =
        "com.ibm.commerce.rfq.commands.MyRFQResponseCreateCmd";
    public static final String COPYRIGHT =
        com.ibm.commerce.copyright.IBMCopyright.SHORT_COPYRIGHT;
    public static String defaultCommandClassName =
        "com.ibm.commerce.rfq.commands.MyRFQResponseCreateCmdImpl";
}
```

- 延伸 `RFQResponseCreateCmdImpl` 並施行 `MyRFQResponseCreateCmd`，以新增指令施行 `MyRFQResponseCreateCmdImpl`。新增新欄位 `responseDescription`，以儲存從使用者介面轉送過來的回應說明。此外，請新增此欄位的 `getter` 方法與 `setter` 方法：

```
protected String responseDescription = "";
public java.lang.String getResponseDescription() {
    return responseDescription;
}
public void setResponseDescription(String name, boolean isReq) throws
EApplicationException {
    try {
        responseDescription = (String)getToolXMLObject().get(name);
    }
```



```

} catch (Exception e) {

    if(isReq) {
        if (getResponseDescription()==null || getResponseDescription().
length()==0) {
            setErrorFlag(true);

            getErrorContent().put(ECRFQMessageKey._ERR_RFQ_MISSING_
RESPONSEREMARKS,"");
        }
    }
}
}

```

改寫 `initParameters()` 方法。此方法將由 **RFQResponseCreateCmdImp** 中的 `checkParameters()` 所呼叫，以起始設定參數。此方法和對應超類別中方法的差異在於，藉由呼叫 **setResponseDescription (MyRFQConstants.EC_RFQ_RESPONSE_DESCRIPTION, false)** 來取得從使用者介面轉送過來的回應說明。以下是方法的原始程式碼：

```

protected void initParameters()
throws com.ibm.commerce.exception.ECApplicationException
{
    setRequestId(RFQConstants.EC_RFQ_REQUEST_ID, true);
    setResponseName(RFQConstants.EC_RFQ_RESPONSE_NAME, true);
    setResponseRemarks(RFQConstants.EC_RFQ_RESPONSE_REMARK, false);
    setCommentsRFQLevelList(RFQConstants.EC_TC_RFQ_LEVEL_COMMENTS,false);
    setResProductsList(RFQConstants.EC_OFFERING_PRODITEM, false);

    //從使用者介面取得回應說明
    setResponseDescription(MyRFQConstants.EC_RFQ_RESPONSE_DESCRIPTION, false);
}

```

改寫 `createResponse()` 方法。除了您必須在 **RFQResponseAccessBean** 與 **TradingDescriptionAccessBean** 中設定說明欄位外，此方法大致相同。以下是方法的原始程式碼：

```

protected RFQResponseAccessBean createResponse ()
throws ECApplicationException,ECEException {
    RFQResponseDataBean responseDB = null;
    try{
        CreateResponseBasicInfoCmd createResCmd = null;
        createResCmd = (CreateResponseBasicInfoCmd)
            CommandFactory.createCommand(CreateResponseBasicInfoCmd.NAME,
getStoreId());
        createResCmd.setRequestId(getRequestId());
        createResCmd.setOwnerId(getOwnerId());
        createResCmd.setResponseName(getResponseName());
        createResCmd.setResponseRemarks(getResponseRemarks());
        createResCmd.setStateIdentifier(getStateIdentifier());
        createResCmd.setCommandContext(getCommandContext());

        createResCmd.execute();

        responseDB=createResCmd.getResponseDataBean();
        //BusinessFlowEventListener 會擷取此 ID
        //以便在 FLINSTANCE 表格中新增記錄

        this.setEntityObject(responseDB);

        //設定此回應參考號碼
        setResponseId(responseDB.getRfqResponseIdInEJBType());

        responseDB.setDescription(responseDescription);
        responseDB.commitCopyHelper();
        TradingDescriptionAccessBean trdDesc = new TradingDescription

```

```

AccessBean();
    trdDesc.setInitKey_languageId(getCommandContext().getLanguageId().
toString());
    trdDesc.setInitKey_tradingId(responseDB.getRfqResponseId());
    trdDesc.refreshCopyHelper();
    if (getResponseDescription()==null || getResponseDescription().
length()==0) {
        trdDesc.setShortDescription(responseDB.getName());
    }
    trdDesc.setShortDescription(getResponseDescription());
    trdDesc.setTimeCreated(TimestampHelper.systemCurrentTimestamp());
    trdDesc.commitCopyHelper();
} catch (javax.ejb.CreateException e) {
    throw new ECAApplicationException(ECRFQMessage._ERR_RESPONSE_BASICINFO
_SAVE,
        this.getClass().getName(), "performExecute");
} catch (javax.naming.NamingException e) {
    throw new ECSYSTEMException(ECMessage._ERR_NAMING_EXCEPTION,
this.getClass().getName(), "createResponse");
} catch (java.rmi.RemoteException e) {
    throw new ECSYSTEMException(ECMessage._ERR_REMOTE_EXCEPTION,
this.getClass().getName(), "createResponse");
} catch (javax.ejb.FinderException e) {
    throw new ECSYSTEMException(ECMessage._ERR_FINDER_EXCEPTION,
this.getClass().getName(), "createResponse");
}
}
return responseDB;
}

```

- c. 執行下列 SQL 陳述式，以便將 **MyRFQReponseCreate** 指令登錄在 URLREG 表格中：

```

insert into urlreg values('MyRFQResponseCreate', 0,
    'com.ibm.commerce.ubf.commands.ToolsBusinessFlowEventCmd', 0,
    null, null, 1);

```

2. 延伸 RFQConstants，以建立新常數類別 MyRFQConstants。只有一個新常數定義：

```
public final static String EC_RFQ_RESPONSE_DESCRIPTION = "response_description";
```
3. 變更 rfq_w_response_general.jsp 檔，以新增新文字欄位與相關處理程序。此檔案位於下列目錄中：

 /usr/WebSphere/CommerceServer/web/tools/rfq

 /QIBM/UserData/WebSphere/CommerceServer/web/tools/rfq

 /opt/WebSphere/CommerceServer/web/tools/rfq

 /opt/WebSphere/CommerceServer/web/tools/rfq

 drive:\WebSphere\CommerceServer\web/tools/rfq

按如下所示變更 Form 區段：

```

<FORM name="rfqCreateForm">
<table COLS=3 WIDTH="60%">
<tr>
<td><%= rfqNLS.get("name") %></td>
</tr>
<tr>
<td><INPUT name="response_name" maxLength=100></td>
</tr>
<tr>
<td><%= rfqNLS.get("remark") %></td>
</tr>
<tr>
<td><TEXTAREA rows="4" cols="40" name="response_remark"></TEXTAREA></TD>

```

```

</tr>
<tr>
<td><%= rfqNLS.get("desc") %></td>
</tr>
<tr>
<td><TEXTAREA rows="4" cols="40" name="response_description"></TEXTAREA></TD>
</TR>
</table>
</FORM>

```

在 savePanelData() 函數中，新增程式碼以儲存回應說明。變更後的函數如下：

```

function savePanelData(){
  VPDResult = validatePanelData0();
  if(!VPDResult)
    return;
  if (isFirstTimeLogonWizard== "1")
    parent.put("<%=RFQConstants.EC_RFQ_REQUEST_ID%>", getRequestId());
    parent.put("<%=RFQConstants.EC_RFQ_RESPONSE_NAME%>",
      document.rfqCreateForm.response_name.value);
    parent.put("<%=RFQConstants.EC_RFQ_RESPONSE_REMARK%>",
      document.rfqCreateForm.response_remark.value);
    parent.put("<%=BusinessFlowConstants.EC_FLOWID%>",
      "<%=RFQConstants.EC_FLOW_RESPONSE_ID%>");
    parent.put("<%=BusinessFlowConstants.EC_BUSINESS_FLOW_EVENT_IDENTIFIER%>",
      "createRFQResponse");
    parent.put("<%=MyRFQConstants.EC_RFQ_RESPONSE_DESCRIPTION%>",
      document.rfqCreateForm.response_description.value);
}

```






在 retrievePanelData() 函數中，新增程式碼以擷取回應說明。變更後的程式碼如下：

```

function retrievePanelData(){
  var form = document.rfqCreateForm;
  form.response_name.value = parent.get("<%=RFQConstants.EC_RFQ_
RESPONSE_NAME%>", "");
  form.response_remark.value = parent.get("<%=RFQConstants.EC_RFQ
_RESPONSE_REMARK%>", "");
  form.response_description.value = parent.get("<%=MyRFQConstants.EC_RFQ
_RESPONSE_DESCRIPTION%>", "");\
}

```

4. 在 rfq_response_wizard.xml 檔中將目標 finishURL 從 RFQResponseCreate 改為 MyRFQResponseCreate。此檔案位於下列目錄中：

-  /usr/WebSphere/CommerceServer/xml/tools/rfq
-  /QIBM/UserData/WebSphere/CommerceServer/xml/tools/rfq
-  /opt/WebSphere/CommerceServer/xml/tools/rfq
-  /opt/WebSphere/CommerceServer/xml/tools/rfq
-  drive:\WebSphere\CommerceServer/xml/tools/rfq

依照下列的程式碼區段進行變更。

```

<wizard resourceBundle="utf.utfNLS"
  windowTitle="matchlisttitle"
  finishConfirmation="ex_finish"
  cancelConfirmation="ex_cancel"
  finishURL="MyRFQResponseCreate"
  tocBackgroundImage="/wcs/images/tools/toc/W_merchand.jpg">

```

5. 變更 UBFStateMachine.xml 檔，並重新載入。此檔案位於下列目錄中：

▶ AIX /usr/WebSphere/CommerceServer/xmlloadutility/businessfollows/xml

▶ 400

/QIBM/UserData/WebSphere/CommerceServer/xmlloadutility/businessfollows/xml

▶ Linux /opt/WebSphere/CommerceServer/xmlloadutility/businessfollows/xml

▶ Solaris /opt/WebSphere/CommerceServer/xmlloadutility/businessfollows/xml

▶ Windows drive:\WebSphere\CommerceServer\xmlloadutility\businessfollows\xml

在這兩個檔案中將動作介面從

com.ibm.commerce.rfq.commands.RFQResponseCreateCmd 改為

com.ibm.commerce.rfq.commands.RFQResponseCreateCmd。下列的範例程式碼說明必要的變更：

變更之前

```
<Flow identifier="RFQ response process totally" priority="2">
  <StartState identifier="START">
    <Transition eventidentifier="createRFQResponse" approval="0"
      priority="1">
      <Action interface="com.ibm.commerce.rfq.commands.RFQResponse
        CreateCmd"/>
      <AccessControlGuard actiongroup="RFQResponseCreate"/>
      <TargetState identifier="DRAFT"/>
    </Transition>
  </StartState>
```

變更之後

```
<Flow identifier="RFQ response process totally" priority="2">
  <StartState identifier="START">
    <Transition eventidentifier="createRFQResponse" approval="0"
      priority="1">
      <Action interface="com.ibm.commerce.rfq.commands.RFQResponse
        CreateCmd"/>
      <AccessControlGuard actiongroup="RFQResponseCreate"/>
      <TargetState identifier="DRAFT">
    </Transition>
  </StartState>
```

註：有關重新載入狀態機的資訊，請參閱 UBF 線上說明。

第10章 預期的庫存

自訂範例

下列範例說明如何自訂 WebSphere Commerce Accelerator 中的這個部份。

實務 1：在建立預期庫存記錄時新增新資訊

本實務會以必要步驟引導您自訂「預期庫存」工具，以記錄預期庫存中的其它資訊，像是：何人建立預期庫存記錄。

1. 在 RA 表格中新增適當 userID 的直欄。建立新 ExpectedInventoryRecords Enterprise Bean 與存取 Bean。
2. 施行新 **CustomExpectedInventoryRecordCreateCmdImpl**（自控制程式指令施行 **ExpectedInventoryRecordCreateCmdImp** 延伸而來），並施行 **ExpectedInventoryRecordCreateCmd**。
3. 將新控制程式指令登錄在 CMDREG 中。假設客戶商店 ID 為 1，請執行下列的 SQL 陳述式：

```
insert into cmdreg(storeent_id,interfacename,classname)
  values(1,'com.ibm.commerce.tools.inventory.ExpectedInventoryRecordCreateCmd ',
        'com.ibm.commerce.tools.inventory.CustomExpectedInventoryRecordCreateCmdImpl ')
```

4. 更新 **CustomExpectedInventoryRecordCreateCmdImpl**。在指令中施行 `performExecute()` 方法：

```
public void performExecute()throws ECSystemException,ECException {
    super.performExecute();
    /**從指令環境定義取得使用者 ID
    將使用者 ID 儲存到 RA 表格中
    **/
}
```

第11章 商業智慧

動態環境定義

動態環境定義中列有可供使用者選擇執行的動作群。清單中含有動作的名稱與簡短說明。清單會隨著所啓用的使用者職務與元件，而動態變更。

自訂範例

實務：新增新動作到現有環境定義中

如果要新增新動作到現有環境定義中，請執行下列步驟：

1. 在環境定義 XML 檔（例如 xml/tools/bi 目錄中的 biContext.xml 檔）中新增動作。您必須找出所要的環境定義，然後才在其中新增項目。以下是「活動」環境定義的「活動」動作：

```
<entry nameKey="campaign" descriptionKey="campaignDescription"
      breadcrumbTrailTextKey="Report" toolsComponent="CommerceAnalyzer">
  <roles>
    <role>siteOwner</role>
    <role>siteAdmin</role>
    <role>seller</role>
    <role>merchant</role>
    <role>makMgr</role>
    <role>podMgr</role>
  </roles>
  <command name = "BIShowReport">
    <parameter name="reportId" value="BICampaignsIndex.html" />
  </command>
</entry>
```

如果要檢視「活動」環境定義，請從 WebSphere Commerce Accelerator 的行銷功能表中，選取**活動**，然後按一下**報告**。

在上述程式碼片段中，toolsComponent 為選用屬性。若有指定，則只有在從架構管理程式啓用指定的元件時，才會列出該動作。

如果使用者具備 roles 元素中定義的職務之一，則會列出該動作給該使用者。

command 元素的 name 屬性是指當從清單選出動作時所要執行的指令名稱。parameter 與 value 屬性是指指令參數與參數的值。

在上述的程式碼片段中未顯示 entry 元素的 appendQueryString 屬性。如果有提供此屬性並設為 true，則會將現行要求的要求內容附加到「名稱-值」配對格式中的動作指令中。舉例來說，下列的程式碼範例說明帳戶報告環境定義中的「帳戶的訂單報告」動作：

```
entry nameKey="ordersByAccount" descriptionKey="ordersByAccountDescription"
      breadcrumbTrailTextKey="reportCriteria" appendQueryString="true">
  <roles>
    <role>siteOwner</role>
    <role>siteAdmin</role>
    <role>seller</role>
    <role>actRep</role>
  </roles>
```

```

        <role>salesMgr</role>
    </roles>
    <command name = "OrdersByAccountDialogView">
        <parameter name="XMLFile" value="reporting.OrdersByAccountReportDialog"/>
    </command>
</entry>

```

ContextEntry 類別會根據上述定義產生下列指令（指令為單行，在此折行僅爲了閱讀方便）：

```

/webapp/wcs/tools/servlet/OrdersByAccountDialogView
?contextConfigXML=contract.brmReportContext
&startIndex=0&ActionXMLFile=contract.rptAccountContextList
&accountId=10001&docname=tools/bi/ContextList.jsp&resultsSize=0
&storeId=135&context=account&langId=-1
&XMLFile=reporting.OrdersByAccountReportDialog&listSize=15

```

在上述指令中，以下的「名稱-值」配對取自要求內容，並附加到指令中：
**contextConfigXML=contract.brmReportContext &startIndex=0
 &ActionXMLFile=contract.rptAccountContextList
 &accountId=10001&docname=tools/bi/ContextList.jsp &resultsSize=0&storeId=135
 &context=account&langId=-1&listSize=15**。

2. 在內容檔中新增內容鍵值。

爲了能啓用國家語言，內容檔中含有 `nameKey`、`descriptionKey` 與 `breadCrumbTrailTextKey` 鍵值。上例的內容檔爲 `properties/com/ibm/commerce/tools/bi/properties/BINLS_en_US.properties`。
`nameKey` 映射至動作名稱。`descriptionKey` 映射至動作的說明。
`breadCrumbTrailTextKey` 則會在選取動作時映射至附加於瀏覽記錄後的文字。

第12章 報告組織架構的概觀

報告組織架構幾乎可為您網站的各個層次，提供一種通用、可自訂的報告功能。報告可供使用 Commerce Accelerator 的任何職務存取。您可定義特定報告的存取權，並限定在該報告的範圍內。WebSphere Commerce Accelerator 使用者隨時可要求取得報告，且組織架構會使用正式作業資料庫中的資料來產生報告，並即時顯示報告。

組織架構內含一個通用的控制程式指令、一個資料 Bean、以及一個用以顯示結果的通用檢視畫面。您可以自訂組織架構，例如：新增有效的 SQL 查詢、定義用以要求與顯示所產生之報告的 JSP 檔等。

自訂報告組織架構

您可從 Commerce Accelerator 存取報告。因此，每一份報告皆須要一些相關的資產。報告本身是由以列表格式出現的資料組成，而基礎資產則是由報告識別碼、SQL 查詢、存取控制元素等組成。報告要求會在伺服器中啟動一個控制程式指令。除非報告另有指定特定的檢視畫面，否則控制程式指令會呼叫作業以設定通用的檢視畫面。指令亦會設定一些必要變數，並傳回此資料，以便將資料移入目標 JSP 檔的 ReportDataBean 中。報告的存取控制設定於會要求（輸入）與顯示報告的檢視畫面上。從資料庫傳回的結果會當成雜湊表向量儲存在資料 Bean 中。最後，JSP 檔會顯示報告。如果報告為空的，JSP 檔將改為顯示一般字串。

報告的存取控制設定於會要求（輸入）與輸出報告的檢視畫面上。

新增新報告的必要步驟如下：

1. 在 XML 檔中定義報告。
2. 必要時請建立 JSP 檔，以便從這裡要求報告。
3. 建立 JSP 檔以顯示報告（除非您採用通用 JSP）。

自訂範例

在 XML 檔中定義報告

個別報告是以 XML 檔來定義。每一份報告各有一個對應的 *reportName.xml* 檔。此檔案含有產生報告所需的所有資訊。*reportname.xml* 檔類似下列的 MyStoreOverviewReport 範例：

```
<?xml version="1.0" standalone="yes" ?>
<Reporting>
<!-- owner="ownerName" location="path_to_this_XML_file " -->
<!-- 集合是由 SQL for WCS 報告組成 -->

<Report reportName="MyStoreOverviewReport" online="true">
<comment>store_overview, yesterday, all measurements</comment>
<SQLvalue>
</SQLvalue>
<mergeOperation>1000000,1000001</mergeOperation>
<display>
<standardInfo>
<resource>reporting.ReportingString</resource>
<title></title>
<message>messageMyReport</message>
```

```

<columnTitles>CRITERIA,KEY,VALUE,CURRENCY,DATESTMP</columnTitles>
</standardInfo>
<userDefinedParameters>
</userDefinedParameters>
</display>
</Report>
<Report reportName="1000000" online="true">
<comment>store_overview, yesterday, revenue</comment>
<SQLvalue>
    select {revenue} as criteria, storeent_id as key,
           sum(totalproduct+totalshipping+totaltax+totaltaxshipping) as value,
           currency as currency, 0 as datestamp
    from orders
    where $DB_DATE_GREATER_EQUAL_FUNC(lastupdate,{beginDate})$ and
           $DB_DATE_LESS_EQUAL_FUNC(lastupdate,{endDate})$
           and status in ('C','M','S') and storeent_id={storeent_id}
    group by storeent_id, currency
</SQLvalue>
<display>
<standardInfo>
<resource>reporting.ReportingString</resource>
<title></title>
<message>message1000000</message>
<columnTitles>CRITERIA,KEY,VALUE,CURRENCY,DATESTMP</columnTitles>
</standardInfo>
<userDefinedParameters>
</userDefinedParameters>
</display>
</Report>
<Report reportName="1000001" online="true">
    <comment>store_overview, yesterday, number of orders</comment>
    <SQLvalue>
        select {orders} as criteria, storeent_id as key, count(*) as value,
               '-' as currency,
               0 as datestamp
        from orders
        where $DB_DATE_GREATER_EQUAL_FUNC(lastupdate,{beginDate})$ and
               $DB_DATE_LESS_EQUAL_FUNC(lastupdate,{endDate})$
               and status in ('C','M','S') and storeent_id={storeent_id}
        group by storeent_id
    </SQLvalue>
<display>
    <standardInfo>
<resource>reporting.ReportingString</resource>
<title></title>
<message>message1000000</message>
<columnTitles>CRITERIA,KEY,VALUE,CURRENCY,DATESTMP</columnTitles>
</standardInfo>
<userDefinedParameters>
</userDefinedParameters>
</display>
</Report>
</Reporting>

```

如果要建立新報告，您必須建立一個類似上例的 XML 檔。有關各 XML 元素的詳細說明，請參閱『有效的 XML 元素』。

有效的 XML 元素: 請使用下列的 XML 元素來定義報告：

<Reporting></Reporting>

此為根元素。

<Report></Report>

此必要元素用來定義特定報告。您可在單一 XML 檔中使用多個 <Report> 元素，以定義多份報告。此元素含有下列兩個必要屬性：

ReportName

此字串用以定義報告的唯一名稱。

online 此布林值用來指出報告能否即時提供。目前僅支援 `true` 值。

<comment></comment>

此選用元素可用來說明報告。此字串不需翻譯。此元素只能定義在現有的 `<Report>` 元素中。

<SQLvalue></SQLvalue>

此必要元素是定義用來產生報告的 SQL 查詢。雖然此為必要元素，但可以是空的。此元素只能定義在現有的 `<Report>` 元素中。

<mergeOperation></mergeOperation>

此選用元素可讓您將多項 SQL 查詢合併到一份報告中。其中列出以逗點隔開的 `reportNames`，而這些 `reportNames` 指向其它 `<Report>` 元素中的 `reportName` 屬性。所參照之報告中的所有 SQL 查詢所傳回的直欄數目必須相同，且用來識別雜湊表鍵值的直欄名稱必須相同。每一項 SQL 查詢和其它查詢間互相獨立。每一項 SQL 查詢各會產生自己的雜湊表向量，且在呈現前，這些向量會附加到其它各 SQL 查詢中，以構成一份單一報告。

上節的「商店概觀」報告範例中即顯示 `<mergeOperation>` 元素的用法。最後報告的第一列取自一個 SQL 查詢，第二列取自後續另一項查詢。此元素只能定義在現有的 `<Report>` 元素中。

<extended_object_class></extended_object_class>

此選用元素含有一個 Java 類別名稱，以藉由使用延伸的 Java 類別來建立 SQL 陳述式。此類別會產生報告，並將資料傳回給報告控制中心。此元素用來建立遞迴報告。此元素只能定義在現有的 `<Report>` 元素中。

<display></display>

此選用元素用來定義顯示結果時所用的參數。此元素只能定義在現有的 `<Report>` 元素中。

<standardInfo></standardInfo>

此必要元素用來集結可透過 `ReportDataBean` 中的 `getter` 存取的元素。這些元素對大部份的報告而言屬於基本元素。此元素只能定義在現有的 `<display>` 元素中。

<resourceBundle></resourceBundle>

此必要元素用來指定要在報告中使用的內容檔。其值亦必須參照於 `reports/resources.xml` 檔中。此元素只能定義在現有的 `<standardInfo>` 元素中。

<title></title>

此必要元素用來指定報告的標題。其值必須是內容檔中的一個鍵值。此元素只能定義在現有的 `<standardInfo>` 元素中。

<message></message>

此必要元素用來顯示報告的相關訊息。例如，用來提供報告的說明。其值必須是內容檔中的一個鍵值。此元素只能定義在現有的 `<standardInfo>` 元素中。

<columnTitles></columnTitles>

此必要元素用來定義直欄的標題。其中列出以逗點隔開的名稱。名稱必須是內容檔中的鍵值。如果在內容檔中找不到鍵值，則會以元素中提供的鍵值做為直欄標題。此元素只能定義在現有的 `<standardInfo>` 元素中。

<userDefinedParameters></userDefinedParameters>

此選用元素用來定義報告組織架構中的自訂元素。報告組織架構所預期的元素格式如下：

```
<element1>value1</element1>
<element2>value2</element2>
```

ReportDataBean 會提供 `getter` 方法，以傳回上述元素的雜湊表，以用於自訂顯示 JSP 中。此元素只能定義在現有的 `<display>` 元素中。

註：雖然 `<display>` 元素中所含的元素被列為必要元素，不過，只有在有定義選用的 `display` 元素時，此項才為真。

SQL 查詢中的變數：當您在 `reportName.xml` 檔中定義變數時，變數必須括上大括弧 (`{variableName}`)。這是告訴報告組織架構該值為一個從屬站變數，而必須自從屬站雜湊表中取得。在上述範例 XML 檔中，`{revenue}`、`{beginDate}`、`{endDate}`、`{storeent_id}` 與 `{orders}` 皆為從屬站變數。

建立 JSP 檔以便由此要求報告

視產生報告時所需的資訊量而定，您必須決定要使用對話框或精靈來收集必要資料。不論哪個元素較適合，JSP 皆必須包含 `savePanelData` JavaScript 函數：

```
function savePanelData()
{
    var reportInputData = new Object();

    reportInputData.SQLid = "the requested report name" ;
    reportInputData.reportXML = "some file";
    reportInputData.variable1 = "some value 1";
    reportInputData.variable2 = "some value 2" ;
    .
    :
    reportInputData.variableN = "some value N";
    reportInputData.varProperties = "a list of variable separated by a comma";
    parent.put("reportInputData", reportInputData);

    // 下列區段可用來指出所要使用的不同檢視畫面
    // var reportResultPage = new Object();
    // reportResultPage.cmd = "ASpecificDisplayReportView";
    // parent.put("reportResultPage",reportResultPage);

    return true;
}
```

必須參照 `reportInputData` 與 `reportResultPage`，以傳遞參數給控制程式指令。此外亦必須指定 `SQLid` 與 `reportXML` 變數。`variable1` 到 `variableN` 以及 `varProperties` 皆為選用的。在範例 `variable1` 到 `variableN` 中，代表將在 SQL 查詢中使用的變數。舉例來說，`variable1` 與 `variable2` 可代入 `beginDate` 與 `endDate`。因此，下列程式碼將出現在 `savePanelData()` 函數中：

```
reportInputData.begindate = " some value";
reportInputData.enddate   = " some value";
```

`varProperties` 變數列出將從內容檔取得其值的變數。舉例來說，其類似如下：

```
reportInputData.varProperties = "revenue,orders,pages,customers,visits";
```

如果 JSP 中未參照 reportResultPage 物件，則控制程式指令會將之設為使用報告組織架構所提供的通用檢視畫面來顯示報告。藉由設定 reportResultPage.cmd，您將能夠指定所要使用的檢視畫面。

下列的程式碼範例顯示用來收集報告之輸入資料的 JSP 檔範例：

```
<!-- =====
Licensed Materials - Property of IBM

5724-A18

(c) Copyright IBM Corp. 2001

US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-----
OrderSummaryReportInputView.jsp
----->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<%@ page import="java.util.*" %>
<%@page import="com.ibm.commerce.tools.util.*" %>
<%@page import="com.ibm.commerce.tools.xml.*" %>

<%@include file="common.jsp" %>
<%@include file="ReportStartDateEndDateHelper.jsp" %>
<%@include file="ReportFrameworkHelper.jsp" %>

<HTML>
<HEAD>
  <%=fHeader%>

  <TITLE><<%=reportsRB.get("OrderSummaryReportInputViewTitle")%></TITLE>

  <SCRIPT SRC="/wcs/javascript/tools/common/Util.js"></SCRIPT>
  <SCRIPT SRC="/wcs/javascript/tools/common/DateUtil.js"></SCRIPT>
  <SCRIPT SRC="/wcs/javascript/tools/common/SwapList.js"></SCRIPT>
  <SCRIPT SRC="/wcs/javascript/tools/reporting/ReportHelpers.js"></SCRIPT>

  <SCRIPT>

    ////////////////////////////////////////////////////////////////////
    // 針對頁面中的各種元素呼叫 initialize 常式
    ////////////////////////////////////////////////////////////////////
    function initializeValues()
    {
      onLoadStartDateEndDate("enquiryPeriod");
      if (parent.setContentFrameLoaded) parent.setContentFrameLoaded(true);
    }

    ////////////////////////////////////////////////////////////////////
    // 針對頁面中的各種元素呼叫 sava 常式
    ////////////////////////////////////////////////////////////////////
    function savePanelData()
    {
      saveStartDateEndDate("enquiryPeriod");

      ////////////////////////////////////////////////////////////////////
      // 指定報告組織架構特定項
      ////////////////////////////////////////////////////////////////////
      setReportFrameworkOutputView("DialogView");
      setReportFrameworkParameter("XMLFile", "reporting.OrderSummaryReport
OutputDialog");
      setReportFrameworkReportXML("reporting.OrderSummaryReport");
      setReportFrameworkReportName("OrderSummaryReport");

      ////////////////////////////////////////////////////////////////////
```

```

// 指定報告特定參數並儲存
////////////////////////////////////
setReportFrameworkParameter("StartDate", returnStartDateAsJavaTimestamp
("enquiryPeriod"));
setReportFrameworkParameter("EndDate", returnEndDateAsJavaTimestamp
("enquiryPeriod"));
saveReportFramework();
return true;
}

////////////////////////////////////
// 針對頁面中的各種元素呼叫 validate 常式
////////////////////////////////////
function validatePanelData()
{
    if (validateStartDateEndDate("enquiryPeriod") == false) return false;
    return true;
}

</SCRIPT>
</HEAD>

<BODY ONLOAD="initializeValues()" CLASS=content>

    <H1><%=reportsRB.get("OrderSummaryReportInputViewTitle") %></H1>
    <i><%=reportsRB.get("OrderSummaryReportDescription")%></i>
    <p>

    <DIV ID=pageBody STYLE="display: block; margin-left: 20">
        <%=generateStartDateEndDate("enquiryPeriod", reportsRB, null)%>
    </DIV>

</BODY>
</HTML>

```

報告組織架構的指令

報告組織架構採用 WebSphere Commerce 所附的下列指令：

檢視畫面指令

表 9. 報告組織架構所用的檢視畫面指令

檢視畫面名稱	JSP 檔
ReportRedirectView	\tools\reporting\ReportRedirect.jsp
ReportGenericView	\tools\reporting\ReportGenericView.jsp

控制程式指令

表 10. 報告組織架構所用的控制程式指令

URL	介面
GenericReportController	com.ibm.commerce.tools.reporting.command.GenericReportControllerCmd

進一步資訊請參閱 WebSphere Commerce 所附之下列套件的相關 JavaDoc 說明：

- com.ibm.commerce.tools.reporting.commands
- com.ibm.commerce.tools.reporting.framework
- com.ibm.commerce.tools.reporting.reports
- com.ibm.commerce.tools.reporting.util

報告組織架構的物件模型

在您產生 JSP 檔時必須使用 ReportDataBean。populate() 方法含有支援即時報告的邏輯。下列方法可提供於資料 Bean 中。

表 11.

方法名稱	傳回類型	說明
populate()	void	執行 SQL 查詢，以產生報告
getErrorCode()	int	用於錯誤碼的 getter 方法
getNumberOfColumns()	int	用於報告中之直欄數的 getter 方法
getNumberOfRows()	int	用於報告中之列數的 getter 方法
getColumnTitlesName(int)	字串	用於 (i+1) 直欄名稱的 getter 方法
getRow(i)	雜湊表	用於 (i+1) 列的 getter 方法
getValue(i,j)	字串	用於報告中之 (i+1,j+1) 值的 getter 方法
getValue(i,keyname)	字串	用於鍵值名稱之相關直欄中之 (i+1) 列的 getter 方法
getUserDefinedParameters()	雜湊表	用於雜湊表的 getter 方法；此雜湊表是從 reportName.xml 中的使用者定義參數建立而來
getEnv()	雜湊表	用於雜湊表的 getter 方法；此雜湊表內含定義在輸入 JSP 檔中的輸入參數

進一步資訊請參閱 WebSphere Commerce 所附的 JavaDoc 說明：

報告 JSP 檔的可重複使用元件

報告 JSP 當搭配報告組織架構時，可提供報告的輸入與輸出檢視畫面。

每一份報告各有一個輸入檢視畫面與輸出檢視畫面。所有輸入 JSP 的作業或用途皆相同。它們會共用相同的外觀；但會要求使用有別於共用輸入小組件儲存池的輸入準則。另一方面，輸出檢視畫面則有共通的結構與格式。根據報告的性質，報告 JSP 中所有可重複使用的部份皆建置成可共用的元件。這些元件是以下列的檔案結構與命名慣例建置而成。

其中 ReportName 應換成新報告的名稱，InputComponent 應換成輸入元件的名稱。所有輸入以及輸出報告 JSP 檔會利用 Reports.properties 來解決多種語言的問題。因此，內容檔會映射 XML 與 JSP 檔中所用的所有標題與鍵值。

以下是最初針對 CSA 營運報告所建置的元件清單：

ReportDaysWaitedHelper.jsp

「等候天數」可編輯框元件

ReportFulfillmentHelper.jsp

供貨選擇元件

ReportInventoryAdjustmentCodeHelper.jsp

庫存調整代碼選擇

ReportProductHelper.jsp

選取產品元件

ReportStartDateEndDateHelper.jsp

「日期對組」輸入元件

ReportVendorHelper.jsp

供應商選擇元件

ReportFrameworkHelper.jsp

輸入 JSP 檔的共通函數

ReportOutputHelper.jsp

輸出頁面格式製作器元件

ReportProductFindDialogView.jsp

搜尋準則輸入頁面

ProductSearch.jsp

搜尋結果選擇頁面

ReportProductFindView 與 ReportProductSearchView 為 ReportProductHelper 所用的獨立頁面。

Common.jsp 同時供報告輸入與輸出頁面共用。ReportFrameworkHelper 供所有報告輸入 JSP 檔共用，ReportOutputHelper 則適用於所有報告輸出頁面。

其它所有 JSP 檔為輸入元件，且可拖曳到需要輸入的報告輸入頁面中。

每一份報告皆須有三個 XML 檔。*ReportNameReportDefinition.xml* 是定義用於報告的 SQL 陳述式，以及報告中每一個直欄的格式。有關如何撰寫 *ReportNameReportDefinition.xml* 的說明，請參閱報告組織架構設計文件。

ReportNameReportInputDialog.xml 為對話框定義。其語法如下：

```
<?xml version="1.0" ?>

<dialog resourceBundle="reporting.reportStrings"
        windowTitle="ReportNameReportWindowTitle"
        finishURL="GenericReportController" >

    <panel name="report"
          url="ReportNameReportInputView"
          hasFinish="YES"
          helpKey="CM.reports.ReportNameReportInputView.Help" />

</dialog>
```

其中 *ReportName* 應換成報告名稱。視窗標題應映射於 *Reports.properties* 檔中。因此 *reporting.reportStrings* 定義在 *xml/tools/reporting/resources.xml* 檔中，並指向 *properties/com/ibm/commerce/tools/reporting/properties/Reports.properties*。說明鍵則映射於 *CMHelpMap.xml* 中。最後，URL 為報告輸入檢視畫面 JSP 的檢視畫面指令名稱。

ReportNameReportOutputDialog.xml 為另一個對話框定義，其中指定報告輸出內容。其格式如下：

```
<?xml version="1.0" ?>

<dialog resourceBundle="reporting.reportStrings"
        windowTitle="ReportNameReportOutputViewTitle"
```



```

        finishURL="" >

<panel name="report"
        url="ReportNameReportOutputView"
        passAllParameters="true"
        hasFinish="NO"
        hasCancel="NO"
        helpKey="CM.reporting.ReportNameReportOutputView.Help" />

<button name="ReportOutputViewPrintTitle"
        action="CONTENTS.printButton()" />

<button name="ReportOutputViewOkTitle"
        action="CONTENTS.okButton()" />

</dialog>

```

上述的範例輸出檢視畫面有兩個自訂按鈕：**列印**與**確定**。其處理程序函數定義在輸出檢視畫面 JSP 中。再者，視窗標題映射於內容檔中，而說明鍵映射於 XML 檔中。

本設計是以可重複使用的元件概念為基礎。輸入檢視畫面 JSP 檔是由一組準則項目組成（視報告規格而定）。由於準則項目可能出現在不同報告的輸入頁面中，因而會將每一項準則項目建置成可重複使用的元件。將此策略運用在輸入檢視畫面 JSP 頁面設計上，具有下列優點：

1. 更容易建立新報告輸入檢視畫面。
2. 所有報告輸入檢視畫面的外觀一致。
3. 可簡化工具組織架構所需的驗證、載入與儲存函數，因為這些函數已內建在每個元件中。

輸出檢視畫面 JSP 檔亦會建置在可重複使用的 helper 上。Helper 會根據資料類型與語言喜好設定，來提供所有必要的格式設定與轉換。每一個直欄的資料類型是指定於報告定義 XML 的使用者定義區段中。在支援更準確的輸出格式情況下，helper 配上 XML 定義，可讓輸出檢視畫面的建立更加容易。

下一節說明如何使用可重複使用的元件來建立報告輸入檢視畫面與輸出檢視畫面 JSP 檔。

如果要建立報告輸入檢視畫面 JSP 檔，您必須匯入您希望 JSP 檔中具備的必要元件。所提供的元件如下：

1. DaysWaited - 指出到今天為止已等候幾天。
2. FulfillmentCenter - 有助於供貨中心的選擇。
3. InventoryAdjustment - 有助於庫存調整代碼的選擇。
4. StartDateEndDate - 指定有效期。
5. Vendor - 有助於供應商的選擇。

只要符合設計策略，您也可建立其它元件。我們將在稍後討論如何建立 helper。

這些元件皆提供共通介面給 JSP 頁面開發人員使用：

1. JSP 函數：

```

String generateInputComponent(String containerName,
        Hashtable reportsRB, String label1 [, String label2])

```

此函數會在輸入檢視畫面中建立元件。 *InputComponent* 為元件的名稱。每一個元件各有一個 *containerName*，這是一個指出此元件之 JavaScript 物件的唯一名稱。所有 JavaScript 函數皆會參照 *containerName*。所有元件皆須有一個報告資源連結 *reportsRB*，以反映現行的語言喜好設定。每一個元件至少有一個映射自標籤的標題。

2. JavaScript 函數：

function onLoadInputComponent(containerName)

當載入頁面時會呼叫此函數。如果是第一次在交易中載入頁面，則會起始設定 *InputComponent*（從資料 Bean）。如果是在交易中重新載入頁面，則會擷取所儲存的資料。

function validateInputComponent(containerName)

此函數應會在您提交要求前呼叫。它會驗證此元件的輸入資料。如果資料無效，則會蹦現一個內含適當訊息的對話視窗，以提醒使用者。如果資料有效，則傳回 *true*，只要資料中有任一部份無效，則會傳回 *false*。

function saveInputComponent(containerName)

只要您導覽至有別於現行頁面的其它頁，應會呼叫此函數。它會儲存元件的現行輸入資料，以便稍後在導覽回現行頁面時再擷取回來。

function visibleList(state)

callback 函數。當組織架構想在頁面中顯示或隱藏選取框時，即會呼叫此函數。此函數應會呼叫：

setSelectComponentVisible(container, state)

定義在每一個有用到選取框的輸入元件中。所有輸入元件的施行相同但名稱互異：

```
function setSelect<Component>Visible(container, state) {  
    document.forms[container].ProductHelperSelectBox.style.visibility = state;  
}
```

報告輸入 JSP 頁面應命名為 *ReportName ReportInputView.jsp*，且應位於下列目錄中：

▶ **AIX** /usr/WebSphere/AppServer/installedApps/ear_directory/
wctools.war/tools/ reporting

▶ **400** /QIBM/ProdData/WebAsAdv4/installedApps/ear_directory/
/wctools.war/tools/ reporting

▶ **Linux** /opt/WebSphere/AppServer/installedApps/ear_directory/
/wctools.war/tools/ reporting

▶ **Solaris** /opt/WebSphere/AppServer/tools/reporting/

▶ **Windows** drive:\WebSphere\AppServer\installedApps\ear_directory
\wctools.war\tools\reporting

它是一個對話框畫面。如果要施行對話框畫面，請參閱「工具組織架構使用者手冊」。對話框畫面含有下列兩個區段：*HTML* 內容產生區段、*JavaScript* 函數區段。在 *HTML* 區段中，您可針對您想顯示在輸入畫面上的每一個元件，分別呼叫 *generateInputComponent*。在 *JavaScript* 區段中，*initializeValue*、*savePanelData* 與 *validatePanelData* 應呼叫定義在各輸入元件中的對應 *JavaScript* 函數。當載入頁面時，會呼叫 *initializedValue*。工具組織架構將會呼叫 *savePanelData* 與 *validatePanelData*。*SavePanelData* 應會呼叫下列報告組織架構所需的 *JavaScript* 函數：

1. `setReportFrameworkOutputView("DialogView");`
2. `setReportFrameworkParameter("XMLFile","reporting.OutputPanelName")`
3. `setReportFrameworkReportXML("reporting.ReportDefinitionXML");`
4. `setReportFrameworkReportName("SQLName");`, 指定於 *ReportXML* 中

您也可以呼叫 `setReportFrameworkParameter("name", value)`，以設定報告輸出 JSP 檔所需的參數。其採用「名稱-值」配對。會傳給 `generator` 的所有標籤以及 `setReportFrameworkParameter` 函數呼叫中的值皆為鍵值，而會根據語言環境與語言喜好設定，映射至正確的字串。映射會定義在下列目錄中的 `Reports_en_US.properties` 內容檔內：

```

▶ AIX /usr/WebSphere/AppServer/installedApps/ear_directory
/properties/com/ibm/commerce/tools/reporting/properties

▶ 400 /QIBM/ProdData/WebAsAdv4/installedApps/ear_directory
/properties/com/ibm/commerce/tools/reporting/properties

▶ Linux /opt/WebSphere/AppServer/installedApps/ear_directory
/properties/com/ibm/commerce/tools/reporting/properties

▶ Solaris /opt/WebSphere/AppServer/installedApps/ear_directory
/properties/com/ibm/commerce/tools/reporting/properties

▶ Windows drive:\WebSphere\AppServer\installedApps\ear_directory
\properties\com\ibm\commerce\tools\reporting\properties

```

將 helper 用在報告輸入與輸出頁面上

報告輸入頁面（為對話框畫面）在輸入準則方面必須有一組元件，並且會施行下列四個 Java Script 函數：

initializeValues()

在每次載入頁面時呼叫。

savePanelData()

每當使用者從這個頁面跳出時呼叫。

validatePanelData()

在傳送準則給報告組織架構前呼叫。

visibleList()

當報告組織架構要求變更此頁面中之元件的可見度設定時呼叫。

如果要新增元件到報告輸入頁面中，請匯入元件 JSP，並在輸入頁面中新增一則 JSP 表示式。依照命名慣例，會命名為 `generateInputComponent`，並採用一個儲存器名稱（對此頁面而言為唯一的）、一個資源連結以及一或多個標題做為參數。舉例來說，您的輸入頁面中將含有類似下列的內容：

```

<%page "ReportStartDateEndDateHelper.jsp" %>
...
<body>
...
<%=generateStartDateEndDate("RequestPeriod", reportRB, "RequestPeriodTitleKey") %>
...
</body>

```

表示式會傳回一個字串，以便在頁面中產生可見元件。如果要搭配上上述三個函數使用，則採用如下命名慣例來定義 `callback` 函數：`onLoadInputComponent`、`saveInputComponent` 與 `validateInputComponent`。這些應分別在 `initializeValue`、`savePanelData` 與 `validatePanelDate` 中呼叫。

`SavePanelData` 函數亦會儲存報告組織架構所需的資訊。每一個輸入元件各會提供傳回函數，以擷取回該元件中的輸入 ID、名稱或其它欄位。有關這些傳回函數的詳述，請參閱任何輸入元件 JSP。

輸出頁面負責處理格式設定。所有格式設定方法皆置於 `ReportOutputHelper` 中，而搭配報告定義 XML 檔使用。只有報告名稱才需指定在報告輸出 JSP 檔中。您可以複製任何報告輸出 JSP 檔，並修改 `reportPrefix` 值，以反映您的報告名稱。

不過，報告定義 XML 檔會根據直欄類型來指定所有直欄與其格式設定。直欄類型如下：

表 12.

直欄類型	預設值	可自訂的內容	預設對齊方式
字串	是	<code>maxEntryLength</code>	右
整數	否	<code>setMinimumIntegerDigits</code> <code>setMaximumIntegerDigits</code>	左
小數	否	<code>setMinimumIntegerDigits</code> <code>setMaximumIntegerDigits</code> <code>setMinimumFractionDigits</code> <code>setMaximumFractionDigits</code>	左
貨幣	否	<code>currencySymbolColumn</code>	左
列舉	否		右
日期	否		左
時間	否		左
月	否		左

預設直欄類型為使用 HTML 直欄選項 `align=left height=20 nowrap` 的字串。所有直欄類型皆可改寫預設直欄選項；其做法是在直欄中指定 `<columnOptions>` 標籤。

您可以選擇性地將所有直欄的 `displayInReport` 值設為 `true` 或 `false`。預設值為 `true`，表示要在報告中顯示該直欄。如果將值設為 `false`，則會隱藏該直欄。此特性可用來自訂報告輸出檢視畫面，而不需變更 SQL 查詢。此外，在您設定貨幣格式，而需要一個參照直欄以告訴格式製作器該貨幣的格式時，則此特性亦很好用。

整數、小數、日期與時間直欄會根據指令環境定義中指定的語言與貨幣值來設定其格式。整數與小數直欄亦可指出整數與分數值的最小與最大位數。

依預設，貨幣直欄會根據指令環境定義中指定的語言與貨幣值來設定其格式。如果您在貨幣直欄中指定 `currencySymbolColumn`，則會從資料庫擷取三個字元的貨幣符號，並用來設定貨幣的格式。如果報告建立者不想顯示貨幣符號字串，則可將所參照的貨幣符號直欄設為隱藏。

列舉是一種特殊的直欄類型，它會將取自資料庫的值映射至某鍵值指定的字串上。舉例來說，Y 或 N 可擷取自表格。這些值可映射至較有意義的字串上，像是不同報告或不同語言中的 Yes 或 No、Approved 或 Denied。如果要如此做，可按如下所示來定義直欄：

```
<columns>
  <columnKey>C2</columnKey>
  <columnName>yyyColumnTitle</columnName>
```

```
<columnType>enumeration</columnType>
  <Y>Yes</Y>
  <N>No</N>
</columns>
```

或

```
<columns>
  <columnKey>C2</columnKey>
  <columnName>yyyColumnTitle</columnName>
  <columnType>enumeration</columnType>
  <Y>Approved</Y>
  <N>Denied</N>
</columns>
```

其中 Yes、No、Approved 與 Denied 字串定義在適當的內容檔中，以適用於多國語言。如果查詢傳回 0、1、2 等數值，而您想將之映射至特定值，您必須以 `<X_n></X_n>` 做為元素。例如：

```
<columnType>enumeration</columnType>
  <X_0>ValueFor0</X_0>
  <X_1>ValueFor1</X_1>
```

撰寫一份利用可重複使用之 JSP 頁面元件的報告

如果要利用可重複使用的 JSP 頁面元件來撰寫一份報告，您必須建立下列：

- JSP 檔：XXXReportInputView.jspXXXReportOutputView.jsp
- XML 檔：
XXXReportInputDialog.xmlXXXReportDefinition.xmlXXXReportOutputDialog.xml
- 更新內容檔：Reports_en_US.properties，您必須在其中新增一個內含所有必要映射的區段。
- 新增檢視畫面指令到資料庫中XXXReportInputView 與 XXXReportOutputView 指令必須新增到資料庫中。
- 在您於上個步驟中所新增的兩個檢視畫面（XXXReportInputView 與 XXXReportOutputView）中設定存取控制。

第13章 產品顧問

自訂範例

在本文件中，我們將提供三個自訂範例。分別是：

- 變更「產品探索」範例 JSP 檔，以使用不同的運算子圖示。
- 變更「產品對照」中所用的預設鏈結。
- 呈現「產品探索」值而不使用所提供的小組件。

實務 1：使用不同的運算子圖示

在「產品探索」虛擬狀態中，會使用運算子圖示（一個會呈現所有運算子的影像）中的選擇項 X 座標來判斷已選取哪個運算子。當您更換影像時，請確定座標不變，如此當點選影像時，才會識別正確的運算子。影像檔位於 `CommerceDir\web\tools\pa\icons` 中。`equalone.gif` 代表 `=`、`<>` 運算子；`equaltoo.gif` 代表用於數值屬性值的 `=`、`<>`、`<=`、`>=` 運算子。預期的 X 座標為：

- `=` 0-22
- `<>` 23-44
- `<=` 45-66
- `>=` 67-88

在值選擇項傳遞 X 座標參數時，會將適當的運算子套用在該選擇項上。

實務 2：「產品對照」虛擬狀態中的鏈結

「產品對照」虛擬狀態支援從個別項目鏈結至另一頁面。目前僅支援一個其它頁面（除非您為值建立一個使用 URL 的屬性）。範例 JSP 會鏈結至 `ProductDisplay` 頁面。請透過 `ProductCompareDataBean` 中的 `productLinkName` 內容來指定此頁面。在範例中，將透過 `ClickInfo` 指令（會收集「產品對照」虛擬狀態的使用統計值）來重新導向。當鏈結至另一頁時，可能需從「產品對照」頁面傳遞一些參數。請在 `ProductCompareDataBean` 的 `productLinkParameters` 內容中識別參數，以傳遞這些參數。只要參數存在，任何在此識別的參數名稱，都會讓參數傳遞到鏈結頁面中。此外，`productId` 參數（例如範例中的 `ECConstants.EC_PRODUCT_ID`）會配有鏈結相關聯之項目的型錄項目 ID (`catentry_id`) 值。`productId` 可讓鏈結頁面識別出使用者所選的項目。目前不支援「產品對照」表格中具備個別項目唯一值的其它參數。

實務 3：呈現「產品探索」的自訂

`ProductExploreDataBean` 是透過 `DynamicForm` 小組件來呈現。如果要呈現您自己的，您可以建立 `DynamicForm` 類別的子類別（有關方法簽章，請參閱 JavaDoc），並改寫 `render` 方法，或者直接從 `ProductExploreDataBean` 取得資料。在 `render` 方法內，請使用 `getDataBean()` 方法來取得 `ProductExploreDataBean` 物件。請使用 `ProductExploreDataBean.getFormElements()` 方法，來取得代表每個屬性直欄的 `ColumnDataBeans` 集合。在每個直欄中，`ColumnDataBean` 物件含有該直欄的值。請使用 `ColumnDataBean.getColumn()` 方法來取得內含個別屬性值的 `DsData` 物件集合。有關方法資訊，請參閱 `ColumnDataBean` 與 `DsData` 中的 JavaDoc。請使用 `DsData` 的 `getPresentationString()` 方法來擷取屬性值的已格式化表示法，以及使用

`getUnformattedData()` 方法來擷取原始資料。如果要為您的 HTML 套表建立適當參數，請使用 `ColumnDataBean.getFormElementName()` 方法來擷取正確的參數名稱，以及使用 `DsData.getUnformattedData()` 方法來取得每一個值。有關適當的運算子選擇，請參閱運算子圖示資訊。

第14章 規則專案

規則專案的循環週期包括下列階段：

1. 建立規則專案
2. 架構以新規則專案為基礎的規則服務
3. 呼叫規則服務
4. 移除以規則專案為基礎的規則服務

假設：此處的資訊僅涵蓋 2、3 與 4。我們假設您已建立規則專案。

有關如何建立規則專案的資訊，請向 **Blaze** 專業服務人員查詢。

如何架構以自訂規則專案為基礎的規則服務

規則系統不會區分 WebSphere Commerce 所附的規則專案或自訂的規則專案。您永遠可以使用管理主控台在規則專案中建立、修改或移除規則服務。不過，自訂的規則專案必須符合某些需求。這些需求為：

1. 張貼到規則專案中的外部事件必須能觸發規則專案的執行
2. 外部事件必須施行 `com.ibm.commerce.rules.InvocationContext` 介面。此介面純為單一標籤介面。

如何呼叫以自訂規則專案為基礎的規則服務

您可以採用和從 WebSphere Commerce 所附規則專案所建之規則服務相同的形式，來呼叫規則服務（從符合上述兩項準則的規則專案建立而來）。亦即，呼叫下列的作業指令：

```
com.ibm.commerce.rules.commands.InvokePersonalizationRuleServiceCommand。
```

第2篇 後記

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。

本書在提及任何 IBM 授權程式時，不表示或暗示只可以使用 IBM 授權程式。只要未侵犯 IBM 的智慧財產權，任何功能相等的產品、程式或服務，都可以取代 IBM 的產品、程式或服務。其它產品、程式或服務在運作上的評價與驗證，除非 IBM 特別指示，其責任屬於使用者。

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。本書使用者並不享有前述專利之任何授權。您可以用書面方式來查詢授權，來函請寄到：

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

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, Japan

下列段落若與該國之法律條款抵觸，即視為不適用：

IBM 就本書僅提供「交附時之現況」保證，而並不提供任何明示或默示之保證，如默示保證書籍之適售性或符合客戶之特殊使用目的；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) 變動本出版品中所提及的產品及 (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。這些網站中的教材不屬於此 IBM 產品的相關教材，若使用這些網站則風險自行負責。

IBM 對您以任何方式提供的資訊隨時享有使用或公開權，且不須負法律責任。

本程式的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」、「IBM 國際程式授權合約」（或任何同等合約）條款，提供本資訊中所述的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。

此資訊僅供規劃用。因此在產品尚未上市前此資訊仍有變更的可能。

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。這些名稱全部都是虛構的，如果與真實企業或個人的名稱和地址雷同，純屬巧合。

本產品中提到的信用卡影像、商標和商品名稱只限於已取得信用卡商標的擁有者授權可以接受以該信用卡付款的商家使用。

商標

下列詞彙為 IBM 公司在美國及（或）其他國家的商標或註冊商標：

Blaze Advisor 為 HNC Software Inc. 在美國與（或）其它國家中的商標。

Microsoft、Windows、Windows NT 與 Windows 標誌為 Microsoft Corporation 在美國與（或）其他國家中的商標或註冊商標。

Netscape 是 Netscape Communications Corporation 在美國與（或）其它國家中的註冊商標。

Oracle 是 Oracle Corporation 在美國與（或）其他國家的商標或註冊商標。

Java、JavaBeans 與所有包含 Java 的商標與標誌是 Sun Microsystems, Inc. 的商標或註冊商標。

其它公司、產品或服務名稱可能是其他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
WebSphere Commerce Accelerator
自訂手冊
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