

版本 6.1.0



商業程序編排器



版本 6.1.0



商業程序編排器

**請注意**

使用此資訊之前，請務必閱讀此文件結尾處注意事項一節中的一般資訊。

**2008 年 2 月 1 日**

本版本適用於 WebSphere Process Server for Multiplatforms (產品號碼 5724-L01) 版本 6 版次 1 修正層次 0，以及所有後續版次及修訂版，除非新版本中另有指示。

若要將您對此文件的意見傳送給我們，請發送電子郵件訊息至 [doc-comments@us.ibm.com](mailto:doc-comments@us.ibm.com)。我們期待收到您的寶貴意見。

當您向 IBM 傳送資訊時，我們會授與 IBM 非專屬權，以任何 IBM 認為適當的方式使用或發佈該資訊，而無需對您承擔任何責任。

**© Copyright International Business Machines Corporation 2006, 2008. All rights reserved.**

# 目錄

## 第 1 篇 WebSphere Process Server 中的商業程序及人力作業 . . . . . 1

### 第 1 章 關於商業程序 . . . . . 3

程序範本 . . . . .	4
商業程序類型 . . . . .	5
程序版本管理 . . . . .	5
程序實例 . . . . .	6
程序生命週期 . . . . .	6
程序實例的狀態轉移圖解 . . . . .	6
子程序的生命週期管理 . . . . .	9
活動的狀態轉移圖解 . . . . .	9
商業程序的呼叫範例情況 . . . . .	14
影響商業程序互動的因素 . . . . .	15
在商業程序與服務之間傳遞參數 . . . . .	16
商業程序的交易式行爲 . . . . .	17
微流程的交易式行爲 . . . . .	17
長時間執行的程序的交易式行爲 . . . . .	18
商業程序中的錯誤處理及補償處理 . . . . .	21
錯誤處理 . . . . .	22
商業程序中的補償 . . . . .	25
從基礎架構失敗回復 . . . . .	25
程序的授權及人員指派 . . . . .	28
商業程序的授權角色 . . . . .	28
建立及啟動商業程序的授權 . . . . .	30
管理商業程序的授權 . . . . .	31

### 第 2 章 關於人力作業 . . . . . 33

作業範本 . . . . .	34
人力作業類型 . . . . .	35
人力作業的版本管理 . . . . .	35
作業實例 . . . . .	36
獨立式及行內作業 . . . . .	36
子作業 . . . . .	39
後續作業 . . . . .	41
呈報 . . . . .	42
呈報的電子郵件通知 . . . . .	44
人力作業的生命週期 . . . . .	45
呼叫作業的範例情況 . . . . .	49
影響獨立式呼叫作業及其服務元件行爲的因素 . . . . .	51
範例情況：支援非同步服務呼叫的獨立式呼叫作業 . . . . .	52
範例情況：支援非同步及同步服務呼叫的獨立式呼叫作業 . . . . .	54
授權及人員指派 . . . . .	56
人力作業的授權角色 . . . . .	57
授權及工作項目 . . . . .	58
人員指派準則 . . . . .	59
人員解析 . . . . .	69
缺席者的代理 . . . . .	74
預設人員指派 . . . . .	75

管理人員指派準則及人員解析結果 . . . . .	76
共用人員指派 . . . . .	77

## 第 2 篇 規劃及配置商業程序編排器 79

### 第 3 章 規劃配置商業程序編排器 . . . . . 81

規劃拓撲、設定及配置路徑 . . . . .	81
規劃建立「商業程序編排器」配置基本範例 . . . . .	84
規劃建立「商業程序編排器」配置範例，包括組織範例在內 . . . . .	85
規劃非正式作業部署環境配置 . . . . .	85
規劃使用管理主控台的部署環境精靈 . . . . .	86
規劃自訂商業程序編排器配置 . . . . .	88
規劃安全、使用者 ID 及權限 . . . . .	89
規劃「商業程序編排器」的資料庫 . . . . .	94
規劃商業流程管理程式及人力作業管理程式 . . . . .	104
規劃人員目錄提供者 . . . . .	105
規劃商業程序編排器瀏覽器 . . . . .	106
規劃商業程序編排器觀察程式 . . . . .	107
關於商業程序編排器 . . . . .	109
關於商業程序編排器瀏覽器 . . . . .	110
關於商業程序編排器觀察程式 . . . . .	111
「商業程序編排器」配置 . . . . .	113
資料來源 . . . . .	114
編輯 . . . . .	114
測試連線 . . . . .	114
資料庫實例 . . . . .	114
綱目名稱 . . . . .	114
建立表格 . . . . .	114
使用者名稱 . . . . .	114
密碼 . . . . .	115
伺服器 . . . . .	115
提供者 . . . . .	115
人力作業管理程式郵件階段作業 . . . . .	115
啓用電子郵件服務 . . . . .	115
郵件傳輸主機 . . . . .	115
郵件傳輸使用者 . . . . .	115
郵件傳輸密碼 . . . . .	116
商業程序編排器瀏覽器 URL . . . . .	116
安全 . . . . .	116
管理者使用者 . . . . .	116
管理者群組 . . . . .	116
監視者使用者 . . . . .	116
監視者群組 . . . . .	116
JMS 鑑別使用者 . . . . .	117
JMS 鑑別密碼及確認密碼 . . . . .	117
JMS API 鑑別使用者 . . . . .	117
JMS API 鑑別密碼及確認密碼 . . . . .	117
呈報使用者鑑別使用者 . . . . .	117
呈報使用者鑑別密碼及確認密碼 . . . . .	117
狀態觀察程式 . . . . .	118

商業流程管理程式的審核日誌記載	118
人力作業管理程式的審核日誌記載	118
商業流程管理程式的共用事件基礎架構記載	118
人力作業管理程式的共用事件基礎架構記載	118
SCA 連結	118
主機	118
商業流程管理程式的環境定義根目錄	118
人力作業管理程式的環境定義根目錄	119
相對路徑	119
匯流排	119
使用預設配置	119
匯流排成員位置	119
遠端目的地位置	119
新建	120
編輯	120
測試連線	120
資料庫實例	120
綱目名稱	120
建立表格	120
使用者名稱	120
密碼	120
伺服器	121
提供者	121
商業程序編排器瀏覽器設定	121
環境定義根目錄	121
瀏覽器搜尋結果限制	121
受管理的「商業程序編排器」儲存器	122
「商業程序編排器觀察程式」設定	122
環境定義根目錄	122
將來自這個「商業程序編排器」事件收集程式的監視資料視覺化	122
<b>第 4 章 配置商業程序編排器</b>	<b>123</b>
使用安裝程式或設定檔管理工具來配置商業程序編排器	123
使用管理主控台的部署環境精靈來配置商業程序編排器	125
使用管理主控台的商業程序編排器配置頁面	127
使用 bpeconfig.jacl Script 來配置商業程序編排器	137
bpeconfig.jacl Script 檔	143
使用已產生的 SQL Script 來建立商業程序編排器的資料庫綱目	156
使用 SQL Scripts 來建立商業程序編排器的資料庫	159
建立商業程序編排器的 Derby 資料庫	161
建立商業程序編排器的 DB2 for i5/OS 資料庫	161
建立商業程序編排器的 DB2 for Linux、UNIX 及 Windows 資料庫	162
建立商業程序編排器的 DB2 for z/OS 資料庫	163
建立商業程序編排器的 Informix Dynamic Server 資料庫	165
建立商業程序編排器的 Microsoft SQL Server 資料庫	166
建立商業程序編排器的 Oracle 資料庫	167
配置人員目錄提供者	169
配置虛擬成員管理程式人員目錄提供者	169
配置 LDAP 人員目錄提供者	170

配置人員替代	175
建立商業程序編排器的佇列管理程式及佇列	178
建立商業程序編排器的叢集佇列管理程式及佇列	179
概觀：配置商業程序編排器瀏覽器	182
配置商業程序編排器瀏覽器	183
配置商業程序編排器觀察程式	186
移除商業程序編排器觀察程式 6.0.1 版範例	186
準備商業程序編排器觀察程式的資料庫	187
在 Java 及 SQL 使用者定義函數之間進行選取	211
配置商業程序編排器事件收集程式應用程式	217
配置商業程序編排器觀察程式應用程式	223
啓用商業程序編排器的記載功能	225
變更商業程序編排器觀察程式的配置參數	226
驗證商業程序編排器觀察程式	234
啓動商業程序編排器	234
驗證商業程序編排器是否運作	235
瞭解商業程序編排器的啓動行爲	235
聯合已配置商業程序編排器的獨立式節點	236

## 第 5 章 移除商業程序編排器配置 . . . 237

使用 Script 來移除商業程序編排器配置	237
使用工具來移除商業程序編排器觀察程式及事件收集程式	239
使用管理主控台來移除商業程序編排器配置	240
使用管理主控台來移除商業程序編排器事件收集程式	244
使用管理主控台來移除商業程序編排器觀察程式	245

## 第 3 篇 管理 . . . 247

## 第 6 章 管理商業程序編排器 . . . 249

使用管理主控台來管理商業程序編排器	249
管理伺服器的補償服務	249
查詢及重新播放失敗的訊息，使用管理主控台	249
使用管理主控台來重新整理人員查詢結果	250
使用管理主控台來啓用共用基礎事件及審核追蹤	251
使用重新整理常駐程式來重新整理人員查詢結果	252
使用 Script 來管理商業程序編排器	253
使用管理指令來刪除審核日誌項目	253
刪除無效的程序範本	255
刪除無效的人力作業範本	257
刪除已完成的程序實例	260
從觀察程式資料庫刪除資料	261
查詢及重新播放失敗的訊息，使用管理指令	264
使用管理指令重新整理人員查詢結果	267
使用管理指令來移除未用的人員查詢結果	270

## 第 7 章 商業程序編排器瀏覽器入門 . . 273

啓動商業程序編排器瀏覽器	277
自訂商業程序編排器瀏覽器	278
自訂不同使用者群組的商業程序編排器瀏覽器介面	278
將「商業程序編排器瀏覽器」介面個人化	280
變更預設 Web 應用程式的外觀	281

## 第 8 章 商業程序編排器觀察程式入門 285



## 第 9 章 管理商業程序及人力作業 . . . . . 289

管理程序範本和程序實例 . . . . .	289
使用管理主控台停止及啟動程序範本 . . . . .	291
使用管理指令停止及啟動程序範本 . . . . .	292
管理程序生命週期 . . . . .	293
修復程序及活動 . . . . .	296
管理作業範本及作業實例 . . . . .	298
利用管理主控台停止及啟動作業範本 . . . . .	298
利用管理指令停止及啟動作業範本 . . . . .	298
建立及啟動作業實例 . . . . .	299
使用您的作業 . . . . .	300
暫停及回復作業實例 . . . . .	300
管理人力作業的優先順序 . . . . .	301
管理指派的工作 . . . . .	301
檢視作業呈報 . . . . .	306
在商業程序編排器瀏覽器建立及編輯自訂內容 . . . . .	308
報告商業程序及活動 . . . . .	308
使用預先定義的清單及圖表 . . . . .	312
建立使用者定義的報告 . . . . .	316
使用已儲存的使用者定義的報告定義 . . . . .	326

## 第 4 篇 開發及部署模組 . . . . . 331

### 第 10 章 開發商業程序及作業的用戶端應用程式 . . . . . 333

### 第 11 章 開發商業程序及人力作業的 EJB 用戶端應用程式 . . . . . 335

存取 EJB API . . . . .	335
存取 Session Bean 的遠端介面 . . . . .	336
存取 Session Bean 的本端介面 . . . . .	338
查詢商業程序及作業相關物件 . . . . .	340
在查詢中使用變數過濾資料 . . . . .	367
管理儲存查詢 . . . . .	368
開發商業程序的應用程式 . . . . .	370
程序實例上動作所需的角色 . . . . .	370
商業程序活動上動作所需的角色 . . . . .	371
管理商業程序的生命週期 . . . . .	372
處理人力作業活動 . . . . .	378
處理單一人員工作流程 . . . . .	380
將訊息傳送給等待中活動 . . . . .	381
處理事件 . . . . .	382
分析程序結果 . . . . .	383
修復活動 . . . . .	383
BusinessFlowManagerService 介面 . . . . .	385
開發人力作業的應用程式 . . . . .	387
啟動呼叫同步化介面的呼叫作業 . . . . .	387
啟動呼叫非同步化介面的呼叫作業 . . . . .	388
建立及啟動作業實例 . . . . .	389
處理待辦作業或協同作業 . . . . .	390
暫停及回復作業實例 . . . . .	391
分析作業結果 . . . . .	392
終止作業實例 . . . . .	392
刪除作業實例 . . . . .	392
釋放已要求作業 . . . . .	393

管理工作項目 . . . . .	393
在執行時期建立作業範本及作業實例 . . . . .	394
HumanTaskManagerService 介面 . . . . .	400
開發商業程序及人力作業的應用程式 . . . . .	403
判斷可以啟動的程序範本或活動 . . . . .	404
處理包括人力作業的單一人員工作流程 . . . . .	406
處理異常及錯誤 . . . . .	408
處理 API 異常 . . . . .	408
檢查針對活動設定的錯誤 . . . . .	408
檢查已停止呼叫活動所發生的錯誤 . . . . .	409

### 第 12 章 開發 Web 服務 API 用戶端應用程式 . . . . . 411

簡介：Web 服務 . . . . .	411
Web 服務元件及控制順序 . . . . .	411
Web 服務 API 的概觀 . . . . .	412
商業程序及人力作業的需求 . . . . .	412
開發用戶端應用程式 . . . . .	413
複製成品 . . . . .	413
從伺服器環境中發佈及匯出成品 . . . . .	413
使用用戶端 CD 上的檔案 . . . . .	418
在 Java Web 服務環境中開發用戶端應用程式 . . . . .	421
產生 Proxy 用戶端 (Java Web 服務) . . . . .	421
建立 BPEL 程序的 Helper 類別 (Java Web 服務) . . . . .	424
建立用戶端應用程式 (Java Web 服務) . . . . .	425
新增安全 (Java Web 服務) . . . . .	426
新增交易支援 (Java Web 服務) . . . . .	429
在 .NET 環境中開發用戶端應用程式 . . . . .	429
產生 Proxy 用戶端 (.NET) . . . . .	429
建立 BPEL 程序的 Helper 類別 (.NET) . . . . .	430
建立用戶端應用程式 (.NET) . . . . .	432
新增安全 (.NET) . . . . .	433
查詢商業程序及作業相關物件 . . . . .	434
商業程序及作業相關物件的查詢 . . . . .	434
商業程序及人力作業物件查詢的預先定義視圖 . . . . .	437
管理儲存查詢 . . . . .	438

### 第 13 章 開發 JMS 用戶端應用程式 439

JMS 簡介 . . . . .	439
商業程序的需求 . . . . .	439
存取 JMS 介面 . . . . .	440
商業程序編排器 JMS 訊息的結構 . . . . .	441
JMS 呈現的授權 . . . . .	443
JMS API 概觀 . . . . .	443
開發 JMS 應用程式 . . . . .	444
複製成品 . . . . .	444
檢查回應訊息是否有商業異常 . . . . .	445

### 第 14 章 使用 JSF 元件開發商業程序及人力作業的 Web 應用程式 . . . . . 447

新增 List 元件至 JSF 應用程式 . . . . .	452
新增 Details 元件至 JSF 應用程式 . . . . .	458
新增 CommandBar 元件至 JSF 應用程式 . . . . .	459
新增 Message 元件至 JSF 應用程式 . . . . .	463

<b>第 15 章 開發作業及程序訊息的 JSP 頁面</b>	<b>467</b>
使用者定義的 JSP 片段	468

<b>第 16 章 建立外掛程式以自訂人力作業功能</b>	<b>469</b>
建立 API 事件處理程式	469
API 事件處理程式	470
建立通知事件處理程式	471
建立對人員查詢結果進行後處理的外掛程式	472
安裝外掛程式	474
登錄外掛程式	474

<b>第 17 章 安裝商業程序和人力作業應用程式</b>	<b>477</b>
以互動方式安裝商業程序及人力作業應用程式	478
配置程序應用程式資料來源及設定參照設定	479
使用管理主控台來解除安裝商業程序和人力作業應用程式	480
使用管理指令來解除安裝商業程序和人力作業應用程式	481

## **第 5 篇 監視商業程序及作業** . . . . . **483**

### **第 18 章 監視商業程序及人力作業** . . . . . **485**

<b>第 19 章 監視商業程序事件</b>	<b>487</b>
商業程序的特定事件資料	487
商業程序事件的延伸名稱	491
商業程序事件	499
商業程序事件的狀況	504

### **第 20 章 監視人力作業事件** . . . . . **507**

人力作業的特定事件資料	507
人力作業事件的延伸名稱	508
人力作業事件	511
人力作業事件中的狀況	513

## **第 6 篇 調整** . . . . . **515**

### **第 21 章 調整商業程序** . . . . . **517**

調整長時間執行的程序	518
平衡硬體資源	518
指定起始 DB2 資料庫設定	519
指定起始 Oracle 資料庫設定	522
規劃傳訊引擎設定	523
調整應用程式伺服器	523
細部調整資料庫	524
細部調整傳訊提供者	528
調整微流程	528
調整包含人力作業的商業程序	529
減少人力作業的並行存取	529
減少查詢回應時間	529
避免掃描整個表格	530
最佳化作業及程序查詢	530

### **第 22 章 調整商業程序編排器瀏覽器** . . . . . **533**

### **第 23 章 調整商業程序編排器觀察程式** . . . . . **535**

## **第 7 篇 疑難排解** . . . . . **537**

### **第 24 章 商業程序編排器配置疑難排解** . . . . . **539**

商業程序編排器日誌檔	539
商業程序編排器資料庫及資料來源疑難排解	540
已啟用代理時，作業儲存器無法啟動	541
6.0.x 商業程序編排器 API 用戶端無法在 6.1 環境中執行	542
啟用商業程序編排器的追蹤功能	543

### **第 25 章 商業程序及人力作業疑難排解** . . . . . **545**

商業程序及人力作業應用程式的安裝疑難排解	545
商業程序的執行疑難排解	546
停止內含微流程的應用程式時發生	
ClassCastException	546
呼叫 processMessage 方法期間發生非預期的異常 (訊息：CNTR0020E)	546
XPath 查詢從陣列中傳回非預期的值	546
活動因未處理的錯誤而停止 (訊息：CWWBE0057I)	547
未補償微流程	547
長時間執行的程序似乎已停止	547
呼叫另一個 EAR 檔中的同步子程序失敗	548
執行期間發生非預期的異常 (訊息：CWWBA0010E)	548
事件不明 (訊息：CWWBE0037E)	549
找不到或無法建立程序實例 (訊息：CWWBA0140E)	549
程序實例的失敗狀態不容許執行所要求的 sendMessage 動作 (訊息：CWWBE0126E)	549
Java 片段中的 Uninitialized 變數或 NullPointerException	550
標準錯誤異常 "missingReply" (訊息：CWWBE0071E)	550
序列化平行路徑	550
將巢狀資料物件複製至另一個資料物件，會毀損來源物件的參照	551
CScope 無法使用	551
使用與程序或作業相關的訊息	551
商業程序及人力作業的管理疑難排解	552
呈報電子郵件疑難排解	553
人員指派疑難排解	554
疑難排解商業程序編排器瀏覽器	558
商業程序編排器觀察程式疑難排解	559
使用程序相關及作業相關審核追蹤資訊	560
商業程序的審核事件類型	560
人力作業的審核事件類型	562
商業程序的審核追蹤資料庫視圖的結構	563
人力作業的審核追蹤資料庫視圖的結構	566

## **第 8 篇 後記** . . . . . **571**



注意事項. . . . . 573



---

## 第 1 篇 WebSphere Process Server 中的商業程序及人力作業



---

## 第 1 章 關於商業程序

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

以「Web 服務商業程序執行語言 (WS-BPEL)」定義的程序包含：

- 屬於程序中個別步驟的活動。活動可以是數種不同類型的其中一種。同時，活動可以分類為基本活動或結構化活動。
  - 基本活動是沒有結構且未含其他活動的活動 (例如，指派或呼叫活動)。
  - 結構化活動是指包含其他活動的活動 (例如，順序或 While 活動)。
- 夥伴鏈結 (也稱為介面夥伴或參照夥伴)，指定使用 WSDL 介面與外部夥伴進行的互動。
- 儲存資料的變數，此資料是要與程序交換並在活動之間傳遞。
- 相互關係集，用來讓多個服務互動與相同商業程序實例產生關聯。相互關係集的基礎為應用程式資料 (包含在與程序交換的訊息中)。
- 錯誤處理程式，可處理可能在商業程序執行時發生的異常狀況。
- 事件處理程式，可平行接收及處理即將刪除的訊息，以正常執行程序。
- 補償處理程式，可指定單一活動、活動群組或範圍的補償邏輯。

如需這些建構的相關資訊，請參閱 BPEL 規格。

「商業程序編排器」還支援 BPEL 語言的 IBM® 延伸，例如：

- 人機互動的人力作業活動。這些行內待辦作業可以是商業程序中涉及某個人員的步驟，例如完成表單、核准文件等。
- 用來執行行內 Java™ 程式碼的 Script 活動。Java 程式碼可以存取所有 BPEL 變數、相互關係內容、夥伴鏈結，以及程序及活動環境定義。
- 用來直接存取 WebSphere® Information Server 或關聯式資料庫的資訊服務活動。
- 程序模型版本管理的有效起始時間的時間戳記。
- 手動設定或控制商業程序中交易界限的延伸。
- 活動逾時。

### 相關工作

第 289 頁的第 9 章, 『管理商業程序及人力作業』

商業程序和人力作業已部署和安裝成爲企業應用程式的一部分。您可以使用管理主控台或管理指令來管理程序範本和作業範本，也可以使用「商業程序編排器瀏覽器」來處理程序實例和作業實例。請使用「商業程序編排器觀察程式」來報告商業程序和人力作業。

第 289 頁的『管理程序範本和程序實例』

使用管理主控台或管理指令來管理程序範本。使用「商業程序編排器瀏覽器」來處理程序實例。

第 291 頁的『使用管理主控台停止及啓動程序範本』

您可以使用管理主控台來個別啓動和停止每一個已安裝的程序範本。

第 292 頁的『使用管理指令停止及啓動程序範本』

除了使用管理主控台之外，管理指令也可以用來停用和啓動程序範本。使用管理指令，停止企業應用程式內的所有程序範本。

第 293 頁的『管理程序生命週期』

在啓動之後，程序會經歷各種狀態，直到結束爲止。作爲程序管理者，您可以在程序的生命週期中對該程序採取各種動作。

第 293 頁的『啓動新的程序實例』

您可以從任何您有權使用的程序範本啓動新的程序實例。

第 294 頁的『監視程序實例的進度』

您可以監視程序實例的進度，以判定您是否需要採取動作，以便程序可以執行完畢。

第 294 頁的『暫停及回復程序實例』

您可以暫停長時間執行的最上層程序實例。例如，您可能想要這樣做，以便您可以配置稍後程序將使用的後端系統存取權，或修正導致程序實例失敗的問題。而在符合程序的必備項目後，就可以回復執行中的程序實例。

第 295 頁的『終止程序實例』

您可能想要終止程序實例，例如，如果不再需要它所代表的工作或文件、如果沒有人可以完成程序實例、如果您發現程序範本方面的問題，以及它需要重新設計等等。

第 295 頁的『刪除程序實例』

可以建立程序範本的模型，以便在完成程序實例時不會自動加以刪除。您可以在完成這些程序實例之後明確地刪除它們。

第 296 頁的『修復程序及活動』

如果程序發生問題，您可以分析程序，然後修復活動。

第 296 頁的『重新啓動活動』

如果已修復活動，則可以使用新的輸入資料來重新啓動它。

第 297 頁的『強制完成活動』

如果您知道將無法及時完成活動，例如，因爲再也無法使用所呼叫的服務，您可以強制完成活動，以便可以繼續執行程序流程。

第 297 頁的『管理微流程的補償服務』

執行微流程時，它可能會發生問題。對於這些狀況，可能已對程序模型中的程序定義補償。補償可讓您復原先前完成的步驟，例如，重設資料及狀態，以便您可以從這些問題回復。

### 相關資訊



Business Process Execution Language for Web Services 1.1 版



OASIS Web Services Business Process Execution Language 2.0 版

---

## 程序範本

程序範本是在執行時期環境中部署及安裝的程序定義。

除了定義程序時指定的內容之外，安裝的商業程序也可以具有下列其中一種狀態：

**已啓動** 建立及啓動程序範本時，可以啓動新的範本實例。



**已停止** 必須先停止程序範本，才可以解除安裝商業程序應用程式。程序範本處於停止狀態時，無法建立及啓動此範本的新實例。

## 商業程序類型

商業程序可以是長時間執行的程序或微流程。

### 長時間執行的程序

長時間執行的商業程序是可岔斷的，且程序的每個步驟都可以在自己的實體交易中執行。長時間執行的商業程序可以等待外部刺激。外部刺激的範例為：企業消費型商務互動中另一個商業程序傳送的事件，對非同步呼叫的回應，或者人力作業的完成。

長時間執行的程序包含下列性質：

- 在多筆交易中執行
- 以同步及非同步方式與服務互動。
- 它的狀態會儲存在執行時期資料庫中，這樣可以往前回復程序

### 微流程

微流程在一個實體執行緒中從開始執行到完成，而沒有任何岔斷。有時，微流程會視為不可岔斷的商業程序。微流程可以具有不同的交易式功能。微流程會參與是廣域交易或活動階段作業的工作單位。

微流程具有下列性質：

- 在一筆交易或一個活動階段作業中執行
- 通常執行時間較短
- 它的狀態是暫時性的，因此不會儲存在執行時期資料庫中
- 它一般會以同步方式呼叫服務
- 它只可以有不可岔斷的子項程序
- 它不可以包含：
  - 人力作業
  - 等待活動
  - 非起始接收活動或挑選活動

#### 相關概念

第 15 頁的『影響商業程序互動的因素』

許多因素都會影響商業程序在各種呼叫範例情況中的行為。這些包括互動樣式、商業程序類型、作業類型及服務端點解析。

## 程序版本管理

您可以建立新版本的商業程序，以在執行時期環境中同時存在相同程序的多個版本。

在 WebSphere Integration Developer 中定義商業程序時，可以包括版本管理化資訊 (例如有效起始日期)。程序的版本是透過它的有效起始日期所決定。這表示不同版本的程序可能會有相同的程序名稱，但是具有不同的有效起始日期。在執行時期使用的程序版本是由程序是用於早期連結範例情況或晚期連結範例情況而決定。

### 早期連結

在早期連結範例情況中，是於建立模型期間或部署程序時決定呼叫的程序版

本。呼叫者會呼叫專用的靜態連結程序。即使存在有根據不同版本的有效起始日期而有效的另一個程序版本，還是會呼叫現行靜態連接的程序，並略過其他所有版本。

SCA 連接是早期連結的一個範例。如果您將獨立式參照連接至程序元件，則每次呼叫使用此參照的程序都會將目標設為由程序元件代表的特定版本。

### 晚期連結

在晚期連結範例情況中，是於呼叫者呼叫程序時決定要呼叫的程序範本。在此情況下，會使用目前有效的程序版本。程序的目前有效版本會取代所有之前的程序版本。現有的程序實例會繼續執行程序範本，在其啟動時與該程序範本相關聯。這會產生下列種類的程序範本：

- 目前有效的程序範本用於新的程序實例
- 不再有效的程序範本，仍然會用於現有的長時間執行程序實例
- 根據程序範本有效起始日期，會在以後變為有效的程序範本。

若要在呼叫子程序時套用晚期連結，母項程序必須指定子程序範本的名稱，在參照的夥伴上會從其選擇有效的子程序。程序的有效起始時間屬性可用來決定目前有效的子程序範本。

晚期連結的範例是在「商業程序編排器瀏覽器」中呼叫新程序時。建立的實例一律會根據程序的目前有效版本，而此程序的有效起始日期不是以後的日期。

---

## 程序實例

程序實例是程序範本的有狀態表現。

「Web 服務商業程序執行語言 (WS-BPEL)」中定義的商業程序，代表有狀態的 Web 服務，因此，它們可以與其他 Web 服務進行長時間執行的互動。只要啟動 BPEL 程序，就會建立該程序的新實例，以與其他商業夥伴進行通訊。實例的最後一個活動完成、執行終止活動或實例發生程序未處理的錯誤時，實例就會完成。

---

## 程序生命週期


起始程序時，會啟動商業程序實例的處理，而且會開始與其環境進行互動。這表示特定互動只有在特定程序狀態下才可以進行，這些互動又會影響程序實例的狀態。

### 程序實例的狀態轉移圖解

只要程序實例的生命週期期間發生某些重要事件，程序就會變更狀態。例如，API 要求導致處於執行中狀態的程序進入暫停狀態。狀態轉移圖解顯示在程序生命週期期間可能發生的狀態轉移。

### 這些圖解中的使用慣例

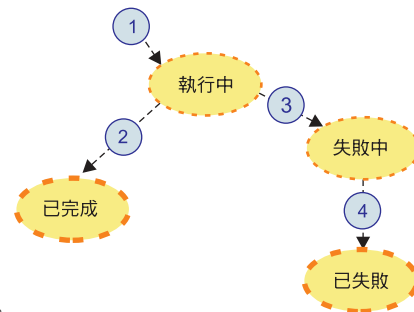
圖解中的狀態轉移是以數字表示。然後再以支援文字來說明這些數字。此外，圖解還包含下列類型的符號：

符號	說明
	暫時性狀態。這些狀態是隱藏的。

符號	說明
	持續狀態。
	暫時性結束狀態。
	持續結束狀態。
	「商業流程管理程式」自動觸發的狀態轉移。
	使用 API 進行的外部互動所造成的狀態轉移。
	「商業流程管理程式」所控制的狀態轉移，或使用 API 進行的外部互動所造成的狀態轉移。

### 微流程實例的狀態轉移圖解

因為程序一律會在交易中執行，而且不會持續保留用於導覽程序實例的實例資訊，所以微流程被視為無狀態。然而，根據程序定義及「商業流程管理程式」的配置方式，微流程的狀態可以公開在「共用基礎事件」或審核日誌中。



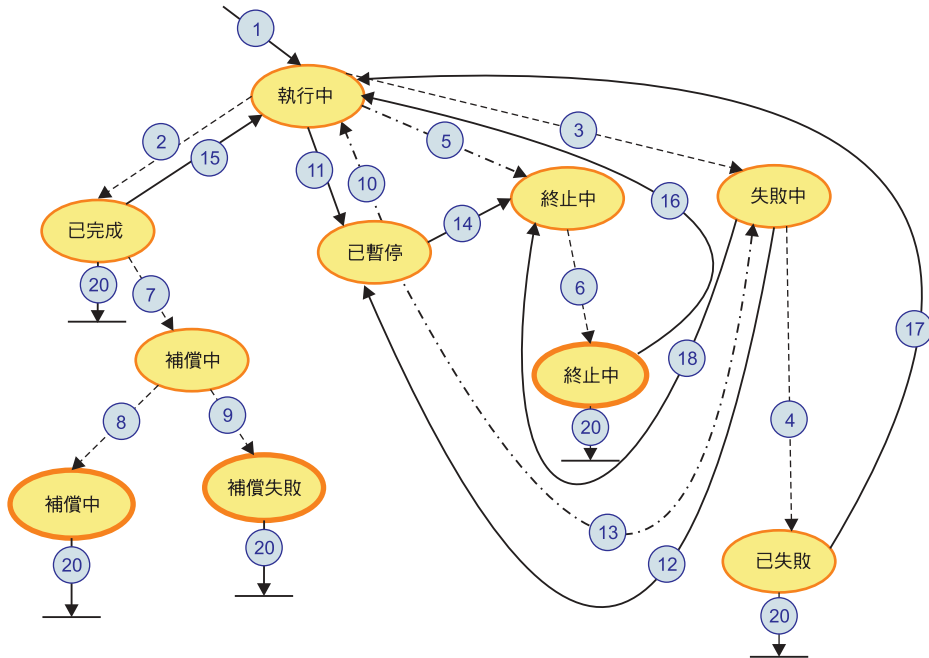
下列圖解顯示微流程實例可能具備的狀態。

正常起始程序實例之後，程序實例達到的第一個程序狀態是執行中狀態 (1)。當程序實例正常地執行到完成時，程序狀態會從執行中變更到已完成 (2)。如果錯誤達到程序界限，則程序會進入失敗中狀態 (3)。執行錯誤處理程式時，程序會停留在失敗中狀態。之後，程序實例會進入失敗狀態 (4)。

所有這些狀態轉移都是由「商業流程管理程式」所觸發。啟動微流程之後，您就無法影響這些自動步驟。

## 長時間執行程序實例的狀態轉移圖解

長時間執行程序會在數個交易中執行。長時間執行程序的狀態會持續保存，因此是可見的。下列圖解顯示長時間執行程序實例可能發生的狀態轉移。



執行中、已完成、失敗中及失敗狀態，以及它們之間的狀態轉移，都與微流程相同。

可以透過外部要求或終止活動終止程序實例。例如，程序實例的終止可以跨多個導覽步驟，因此可以跨多筆已鏈結的交易，以終止長時間執行活動或子程序。在此終止階段期間，程序實例會處於終止中狀態 (5)、(14)、(18)。終止程序的所有長時間執行部分時，程序實例的狀態也會變更為終止 (6)。

當子項程序順利結束，且母項程序在稍後失敗時，可以補償子項程序。在補償期間，子項程序會處於補償中狀態 (7)。如果補償順利結束，則子項程序會進入已補償狀態 (8)。如果補償未順利結束，則子項程序會進入補償失敗狀態 (9)。這些狀態交易是由母項程序自動起始。

如果程序實例的導覽仍然作用中 (亦即，處於執行中或失敗中狀態)，則可以使用 API 要求將它暫停。然後，就可以在指定的時間之後或透過回復要求重新將它啟動。程序的狀態使用暫停要求從執行中或失敗中變更為已暫停 (11)、(12)，並使用回復要求從已暫停變更為執行中或失敗中 (10)、(13)。還可以終止處於暫停狀態的程序 (14)。只能暫停及回復最上層的程序實例。然而，暫停或回復狀態都會延伸至子項程序。

當程序達到其中一個結束狀態 (已完成、終止或失敗) 時，可以使用重新啟動 API 要求重新啟動 (15)、(16)、(17)。只能重新啟動最上層的程序實例，同時只能補償子項程序實例。

程序實例達到結束狀態 (20) 時，就可以將它刪除。如果也設定了 **automatically delete on completion** 屬性，則會自動刪除程序，否則可以透過明確刪除要求來觸發它。

## 子程序的生命週期管理

由另一個程序啟動的程序稱為子程序。子程序生命週期的管理方式，視如何建立這些程序的模型而定。

若要模組化及重複使用，通常是將一或多個商業邏輯的步驟實作為不同的程序，並從主程序呼叫此程序。子程序也可以啟動另一個程序。這樣會產生程序實例階層。當部署這些程序時，程序至程序關係中的所有程序範本，都必須部署至相同的「商業程序編排器」資料庫。

子程序與呼叫程序的關係可以是對等式關係或上下代關係。當針對呼叫程序而管理程序生命週期的動作時，此關係會決定子程序的行為。生命週期作業包含暫停、回復、終止、刪除及補償。在上下代關係中，管理程序生命週期的作業只能在最上層程序實例上執行。

程序/子程序關係是由子程序的 **autonomy** 屬性決定。此屬性可以具有下列其中一個值：

**Peer** 同層級程序會視為最上層程序。最上層程序不一定是由另一個程序實例呼叫的程序實例，但它具有同層級 **autonomy**。如果子程序是對等式關係的一部分，則呼叫程序實例的生命週期作業不會延伸到子程序實例。

使用單向介面建立及啟動的長時間執行程序，視為同層級程序。在執行時期，會略過它的 **autonomy** 屬性。

**Child** 如果子程序是上下代關係的一部分，則母項程序實例的生命週期作業會套用于子程序實例。例如，如果暫停母項程序實例，則也會暫停 **autonomy** 為 **child** 的所有子程序實例。子項程序在傳回給它的母項程序時必須完成，亦即，子項程序的最後一個作業必須是其對呼叫母項程序的回覆。請確定程序邏輯中的所有可能路徑都以回覆活動作為路徑的最後一個作業結束。

微流程一律會作為子項程序執行，亦即，會略過它的 **autonomy** 屬性。

只有在直接互動的程序之間才可以建立上下代關係。如果另一個 **SCA** 元件截取此互動，則可能會無法建立上下代關係，例如，兩個程序元件之間連接的介面對映元件。

## 活動的狀態轉移圖解

執行活動實例的重要步驟時，活動實例狀態會變更。狀態及狀態轉移是視活動類型而定。

在基本活動的生命週期，狀態及狀態轉移十分重要。基本活動分組成下列活動類型。狀態轉移圖解會根據活動類型而不同：

- 短期活動 (例如指派、空、回覆、重新擲出、擲出及終止活動)
- 等待外部事件的活動 (例如接收及等待活動)
- 挑選 (接收選擇) 活動
- Java Snippet 活動
- 呼叫活動
- 人力作業活動

與程序實例的狀態圖解相比，未明確公開活動結束狀態。活動的生命週期是視含括程序而定。活動一律會與程序實例一起刪除。

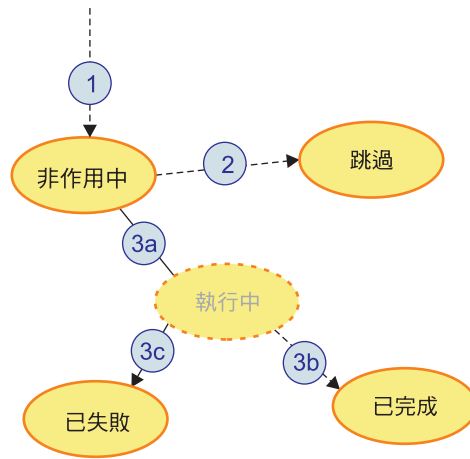
## 這些圖解中的使用慣例

圖解中的狀態轉移是以數字表示。然後再以支援文字來說明這些數字。此外，圖解還包含下列類型的符號：

符號	說明
	暫時性狀態。這些狀態是隱藏的。
	持續狀態。
	「商業流程管理程式」自動觸發的狀態轉移。
	使用者互動所造成的狀態轉移 (例如，透過 API 要求)。
	「商業流程管理程式」或使用者互動所控制的狀態轉移。

## 短期活動類型的狀態轉移圖解

下列狀態圖解顯示簡式短期活動類型 (例如指派、空、回覆、重新擲出、擲出及終止活動) 的狀態及狀態轉移。並引進下列狀態：非作用中、已略過、已完成及失敗。這些狀



態適用於所有基本活動類型。

活動在建立之後，會處於非作用中狀態 (1)。流程中含括的活動可以有多个送入鏈結及一個結合條件。啟動這類活動之前，必須導覽過所有的送入鏈結。活動的 **suppressJoinFailure** 屬性及結合條件的評估結果可判斷活動的後續行為：

- 結合條件評估為 `false`，且 **suppressJoinFailure** 屬性設為 `true`。

活動狀態會變更為已略過 (2)，且會將離開活動的鏈結導覽為停用路徑。

- 結合條件評估為 `false`，且 **suppressJoinFailure** 屬性設為 `false`。

活動未啟動因此仍然保持為非作用中狀態，且會引發 `bpws:joinFailure` 標準錯誤。

- 結合條件評估為 `true`。

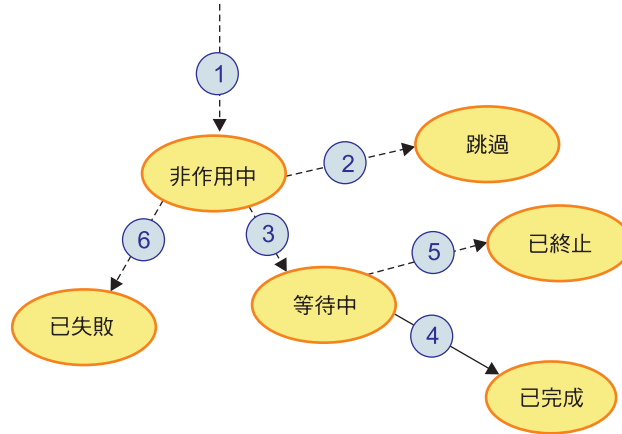
若為流程中未含括的活動，這是預期的行為。活動會啟動，且其狀態會變更為執行中 (3a)。會執行活動實作，然後活動狀態會變更為已完成 (3b)。如果實作失敗 (例如，



指派活動中的 copy 陳述式語法不正確時)，活動狀態會變更為失敗 (3c)。所有短期活動都是不可岔斷的。因此，絕對看不到執行中狀態。

### 等待外部事件的活動的狀態轉移圖解

下列圖解顯示在等待或接收活動生命週期期間，可能發生的狀態及狀態轉移。



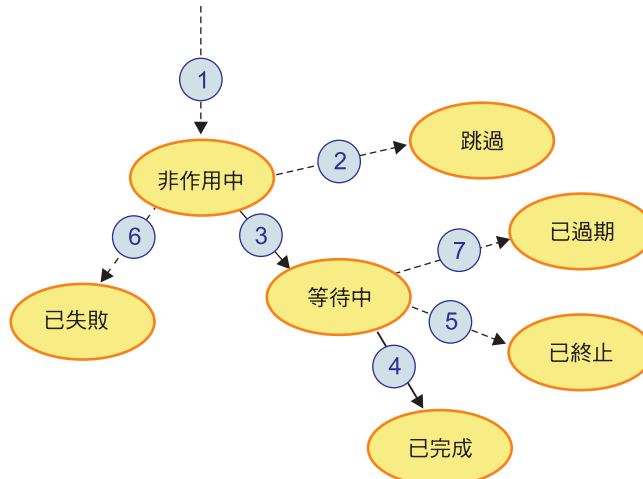
接收及等待活動的開始階段與短期活動的開始階段相同。然而，啟動接收及等待活動之後，狀態會變更為等待中，而不是執行中 (3)。接收或等待活動在完成並移至已完成狀態 (4) 之前，已可以接收外部要求，或等待指定的逾時。接收活動的轉移至已完成狀態作業，是由接收到的訊息所觸發。而等待活動的這項轉移則是在經歷指定的等待時間之後自動完成。

等待或接收活動的實作會比簡式短期活動類型的實作更為複雜。等待或接收活動可能會在完成開始活動這個作業之前就失敗，例如，評估等待活動的等待時間時，就失敗。此失敗會導致活動狀態在到達等待中狀態之前變更為失敗 (6)。

就算活動處於等待中狀態，含括程序還是可能會接收到終止要求，或是與等待或接收活動平行的分支還是會發生錯誤。如果發生上述任一事件，則會終止等待或接收活動，而且活動狀態會變更為終止 (5)。

### 挑選 (接收選擇) 活動的狀態轉移圖解

挑選活動 (也稱為接收選擇活動) 的狀態及狀態轉移會顯示在下列狀態圖解中。

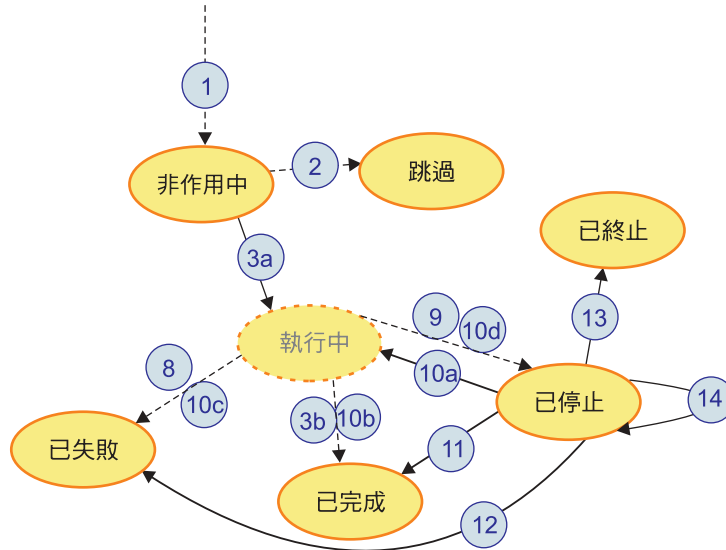


挑選活動從 (1) 到 (6) 的狀態及狀態轉移與接收活動的那些狀態及狀態轉移相同。

此外，在挑選活動的要求到達之前，啟動等待中挑選活動的警示分支時，挑選活動就會到期。然後，活動會處於過期狀態 (7)。

## Java Snippet 活動的狀態轉移圖解

下列圖解顯示在 Java Snippet 活動的生命週期期間，可能發生的狀態及狀態轉移。



Java Snippet 活動的執行時期行為與短期活動的執行時期行為類似；當啟動 (3a) Java Snippet 活動時，就會執行實作。因為 Java Snippet 的實作很短，所以看不到執行中狀態。如果 Java Snippet 順利完成，則活動的狀態會立即變更為已完成 (3b)。

如果 Java Snippet 失敗，則活動會以下列其中一種方式進行：

- 活動狀態變更為已停止 (9)。如果活動的 **continueOnError** 屬性設為 **false**，而且 Java Snippet 含括範圍上的錯誤處理常式捕捉不到 Java Snippet 所擲出的錯誤，則需要進行管理互動，以繼續進程序導覽。
- 活動狀態變更為失敗 (8)，並啟動錯誤處理。

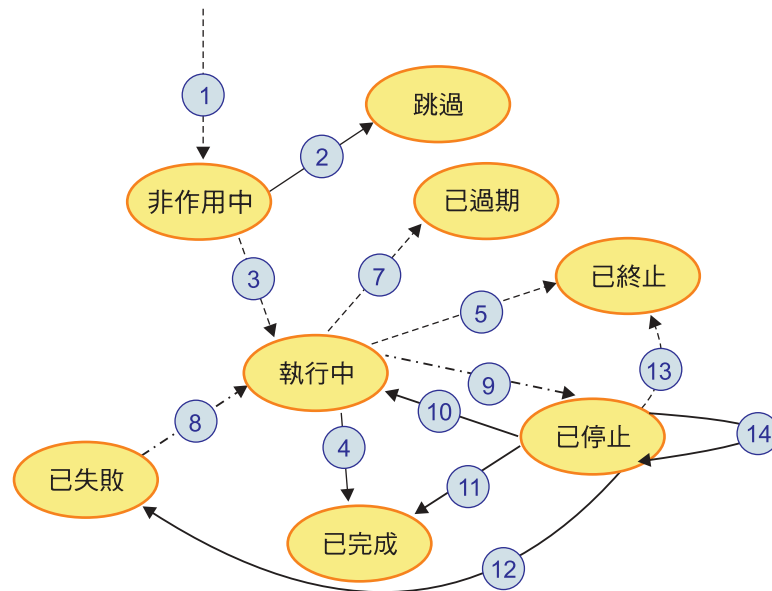
如果活動處於已停止狀態，則可以使用 API 要求將其變更為強制重試或強制完成。如果是強制重試，則活動會進入執行中狀態，並重新執行實作 (10a)。因為外部或程序內部條件已變更，所以 Java Snippet 可以順利完成，而且活動的狀態會變更為已完成 (10b)。如果活動再次失敗，則管理者可以實作強制重試 API 來停止活動 (10d)，或讓程序錯誤處理來處理此失敗 (10c)。

如果含括程序或範圍在活動處於已停止狀態時終止，則活動狀態會變更為終止 (13)。

如果將活動強制為已完成，則不會重新執行活動的實作。活動會進入已完成 (11)、失敗 (12) 或已停止狀態 (14)。

## 呼叫活動的狀態轉移圖解

下列圖解顯示在具有非同步實作的呼叫活動生命週期期間，可能發生的狀態及狀態轉移。如果在服務要求交易的後續交易中發生服務回覆，則實作會非同步。程序及服務元件的 SCA 限定元可判斷服務是以同步方式還是以非同步方式呼叫。



呼叫活動的啟動與其他所有活動類型 (1)、(2) 的啟動相同。

當呼叫活動正常地執行到完成時，則會啟動活動，且狀態會變更為執行中 (3)。如果服務呼叫順利傳回，則活動會進入已完成狀態 (4)。

只要服務未回覆，且活動處於已停止狀態，管理者就可以強制重試或強制完成活動。這適用於服務無法回覆時 (例如，系統中斷)。對應的 API 也會導致從執行中到已停止 (9)、失敗 (8) 及已完成 (4) 的狀態轉移。如果非同步服務是子項程序，則在活動處於執行中狀態時，無法對活動進行強制重試或強制完成。

與 Java Snippet 活動相同，也可以停止呼叫活動 (9)。接著可以透過管理動作予以修復，或因含括範圍或程序終止而一併終止 (13)。

如果定義活動的期限，則處於執行中狀態的活動可能會到期。活動狀態接著會變為過期 (7)，並擲出逾時錯誤。此錯誤可以由錯誤處理常式進行處理。

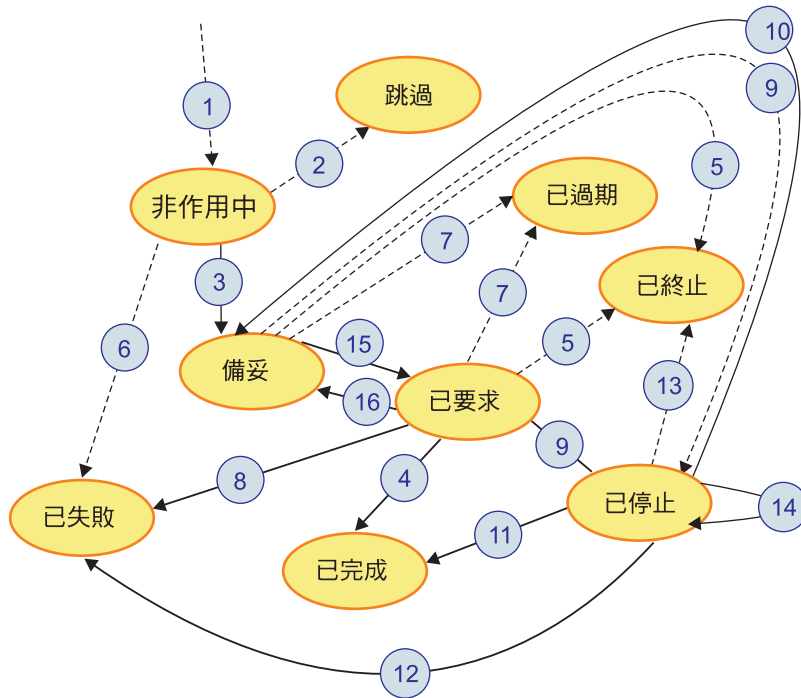
如果含括的活動範圍已終止 (例如，因為程序中的平行路徑失敗，且活動處於執行中狀態)，則活動也會終止，並進入已終止狀態 (5)。

具有同步服務呼叫的呼叫活動的狀態轉移，與 Java Snippet 的這類狀態轉移相同。同步與非同步呼叫間的狀態及狀態轉移差異如下所示：

- 絕對看不到具有同步服務呼叫的呼叫活動的執行中狀態。
- 期限不適用於具有同步呼叫的呼叫活動；絕對不會達到過期狀態。
- 絕對不會終止具有同步服務呼叫的呼叫活動。

## 人力作業活動的狀態轉移圖解

下列圖解顯示在人力作業活動的生命週期期間，可以發生的狀態及狀態轉移。



人力作業活動的執行時期行為類似於呼叫活動。呼叫活動的執行中狀態對應於人力作業活動的備妥及已要求狀態。備妥狀態指出活動可供人員使用。當某人要求使用活動，該活動會進入已要求狀態 (15)。

使用活動的人員會提供所需要的資訊，並完成活動。然後，活動會處於已完成、失敗或已停止狀態。或者，要求活動的人員可能決定無法完成活動。然後，該人員會釋放活動供其他人使用。在此情況下，活動會回復為備妥狀態 (16)。

其他狀態轉移則會與具有非同步服務呼叫的呼叫活動相同。

### 相關概念

第 21 頁的『商業程序中的錯誤處理及補償處理』

錯誤是任何可變更商業程序正常處理的異常條件。設計良好的程序應該會考量到錯誤，並盡可能地處理它們。補償是一種處理錯誤的方式。

## 商業程序的呼叫範例情況

商業程序是 SCA (「服務元件架構」) 元件實作類型。它可以將服務公開給其他夥伴，並使用其他夥伴提供的服務。商業程序可以是透過「商業程序編排器 API」變為可用的服務提供者、其他 SCA 服務元件的 SCA 服務提供者，或呼叫其他 SCA 服務元件 (包括其他商業程序) 的 SCA 用戶端。

### 商業程序作為透過「商業程序編排器」API 變為可用的服務提供者

您可以使用「商業流程管理程式」API 來實例化商業程序。「商業程序編排器」用戶端應用程式也會使用此 API，將商業程序當成服務提供者使用。這些用戶端應用程式可以建立和啟動商業程序實例，以及查詢和操作現有的程序實例。

「商業流程管理程式」API 提供為 EJB、Web 服務及 JMS 訊息介面，您可用來設計 EJB、Web 服務及 JMS 用戶端。

#### 商業程序作為其他 SCA 服務元件的 SCA 服務提供者

在此呼叫範例情況中，商業程序代表作為用戶端的其他 SCA 元件可以呼叫的 SCA 元件。因為是 SCA 元件的實作，所以可以從 SCA 用戶端及其他 SCA 元件來呼叫商業程序所提供的服務。這些機制包括：

- 連線，連接 SCA 用戶端 (參照) 及代表商業程序的元件介面
- 決定互動樣式、交易行為及互動可靠性這類部分的元件參照及介面的 SCA 限定元設定

#### 商業程序作為呼叫其他 SCA 服務元件的 SCA 用戶端

相反地，因為商業程序是 SCA 元件的實作，所以其他 SCA 元件或 SCA 匯入可以提供它所使用的服務。離埠互動中所用的 BPEL 夥伴鏈結是透過 SCA 參照代表。此參照可以連接至其他 SCA 元件或匯入，而 SCA 限定元可以用來關聯服務品質屬性與互動。

#### 商業程序作為呼叫其他商業程序的 SCA 用戶端

如果 SCA 用戶端及 SCA 服務都是由商業程序代表，則可以在 SCA 層次及商業程序層次上同時選取兩者。在 SCA 層次上，可以使用 SCA 連線，將 SCA 用戶端連接至 SCA 服務。在商業程序層次上，則可以關聯夥伴鏈結與作為服務提供者的商業程序名稱。

## 影響商業程序互動的因素

許多因素都會影響商業程序在各種呼叫範例情況中的行為。這些包括互動樣式、商業程序類型、作業類型及服務端點解析。

### 互動樣式

可以同步或非同步方式呼叫商業程序提供的作業。

**重要：** 同步互動的合理回應時間不應該超出數秒。如果商業程序所實作的要求回應作業未在短時間內傳回它的結果，請考慮使用非同步互動樣式，以改善效能。同步呼叫這類作業會導致資源封鎖。也容易發生與系統工作量相關而無法判斷的逾時狀況。

### 商業程序類型

商業程序可以是微流程或長時間執行程序。每個程序類型的性質都會影響呼叫範例情況。

### WSDL 作業類型

SCA 參照及 SCA 介面與包含一或多個作業的 WSDL 埠類型相關聯。每個作業都可以是單向或要求回應作業。單向作業表示完成呼叫用戶端不知道的服務執行。服務執行結束時，會成功呼叫相關聯的服務。要求回應作業表示完成呼叫用戶端知道的服務執行。呼叫用戶端可以使用服務執行結果時，服務執行會結束。

### 服務端點解析

在商業程序的環境定義中，使用下列方式可以建立呼叫用戶端與所呼叫服務的關聯：

- SCA 連線以靜態方式建立 SCA 參照與所呼叫服務介面的關聯。這是 SCA 層次機制，而且可以在將用戶端及 (或) 服務端實作為商業程序時套用。

- 端點參照可以設定於夥伴鏈結的環境定義中，而夥伴鏈結屬於作為 SCA 用戶端的商業程序一部分。端點參照可以唯一決定要呼叫的 Web 服務的通訊終點。一般而言，可以呼叫所有 Web 服務。端點參照可以套用至 Web 服務連結或 SCA 連結。
- 可以針對屬於作為 SCA 用戶端的商業程序一部分的夥伴鏈結，設定商業程序範本名稱。範本名稱可以唯一決定另一個部署於相同伺服器或叢集中的商業程序的名稱。

### 相關概念

第 5 頁的『商業程序類型』

商業程序可以是長時間執行的程序或微流程。

## 在商業程序與服務之間傳遞參數

商業程序可以使用服務元件架構 (SCA) 服務，或可以由其他 SCA 服務使用。資料在 SCA 服務與程序之間的交換方式，視建立程序模型的方式而定。

### 商業程序使用服務

在商業程序中使用服務，可透過使用程序模型中的「商業程序執行語言 (BPEL)」呼叫活動來實作。傳遞至 SCA 服務的資料是擷取自一個以上的 BPEL 變數。通常，會透過值來傳遞資料，其表示呼叫的服務會處理資料的副本。

在特定情況下，可以透過參照傳遞資料。透過參照傳遞資料可有助於改進商業程序的效能。

如果符合**所有**下列條件，則會透過參照將資料傳遞至商業程序：

- 服務的呼叫是同步的。
- BPEL 程序及呼叫的服務在相同的模組中。
- 資料是使用資料類型變數進行交換。

如果呼叫的服務修改資料，則這些變更會套用至對應的 BPEL 變數。然而，最佳作法是呼叫的服務不應更新資料，因為對資料進行的任何變更都不會持續存在。若為長時間執行的程序，當現行交易確定時會捨棄變更，而若為微流程，當程序結束時會捨棄變更。此外，當由呼叫的服務更新變數時，不會產生事件。

### 商業程序由服務使用

由其他服務使用的商業程序包含程序模型中的接收活動、挑選活動，或事件處理常式。傳遞至程序的資料會寫入一個以上的 BPEL 變數。通常，會透過值來傳遞資料。

然而，如果符合**所有**下列條件，則會透過參照傳遞資料：

- 商業程序的呼叫是同步的。
- 服務及呼叫的商業程序在相同的模組中。
- 資料是使用資料類型變數進行交換。

如果呼叫的程序修改 BPEL 變數，則也會修改來自呼叫服務的輸入資料。



## 商業程序的交易式行為

商業程序是執行為交易的一部分。如果是長時間執行程序，則商業程序的導覽可以跨多個交易，如果是微流程，則可以作為某個交易的一部分。外部要求、內部訊息或非同步服務的回應都可以觸發這類導覽交易。交易啟動時，會根據程序定義來執行必要活動。呼叫服務可以參與交易。

### 相關概念

第 14 頁的『商業程序的呼叫範例情況』

商業程序是 SCA（「服務元件架構」）元件實作類型。它可以將服務公開給其他夥伴，並使用其他夥伴提供的服務。商業程序可以是透過「商業程序編排器 API」變為可用的服務提供者、其他 SCA 服務元件的 SCA 服務提供者，或呼叫其他 SCA 服務元件（包括其他商業程序）的 SCA 用戶端。

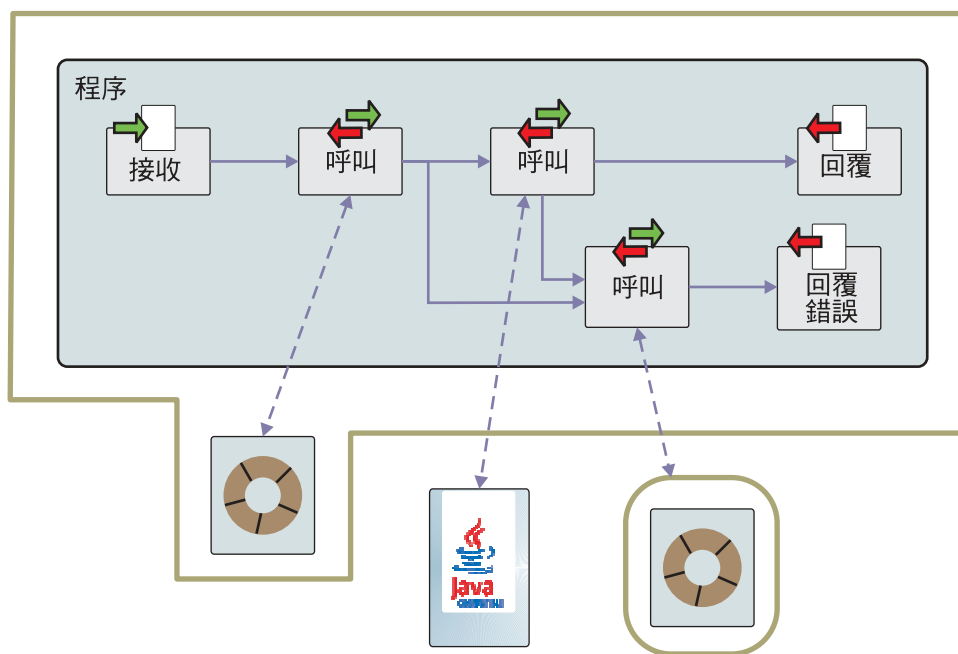
## 微流程的交易式行為

微流程是短期程序。它們可以在交易中執行，或在微流程的 SCA 元件上所指定的活動階段作業中執行。這裡會說明當成交易一部分執行的微流程。

微流程是不可岔斷的。因此，微流程不可包含等待外部事件或等待使用者互動（例如，人力作業活動）的活動。

微流程是暫時性的。微流程的程序實例狀態保留在記憶體中，而不是儲存在執行時期資料庫中。然而，微流程實例的狀態可以持續保存在審核日誌或「共用基礎事件」中。

下列圖解顯示微流程的交易以及與微流程互動的服務。交易界限內的服務會參與微流程交易；而在界限外的服務則不會參與交易。



## 呼叫服務及微流程交易

雖然微流程是在一個交易中執行，但是微流程的執行可以含括多個交易。原因是透過呼叫活動所呼叫的服務可以參與微流程交易，也可以在它自己的交易中執行。

下列設定可判斷服務是參與微流程交易，還是在它自己的交易中執行。

- 用來呼叫服務的互動樣式。

互動樣式可以是同步也可以是非同步。樣式的判斷是根據目標 SCA 元件或 SCA 匯入的偏好互動樣式，以及作業是單向作業還是要求回應作業，如下表所示：

表 1.

目標元件或匯入的偏好互動樣式	單向作業	要求回應作業
任何	同步呼叫	同步呼叫
同步的	同步呼叫	同步呼叫
非同步	非同步呼叫	同步呼叫

- 指定給程序及所呼叫服務的「服務元件架構 (SCA)」交易限定元：
  - 程序元件參照上的 **suspendTransaction** 限定元，指定是否要將程序的環境定義傳送給要呼叫的服務。
  - 服務介面上的 **joinTransaction** 限定元，指定在傳送交易時，服務是否參與其呼叫者的交易。

根據這些設定，下列規則適用於所呼叫服務：

#### 同步呼叫

<b>joinTransaction</b>	<b>suspendTransaction = true</b>	<b>suspendTransaction = false</b>
<b>joinTransaction = true</b>	服務未參與微流程交易	服務參與微流程交易
<b>joinTransaction = false</b>	服務未參與微流程交易	服務未參與微流程交易

如果服務參與微流程交易，則只有在確定微流程交易時，才會持續保存服務對交易式資源進行的變更。如果服務未參與微流程交易，則即使回復交易，可能也會持續保存服務對交易式資源進行的變更。您可以使用補償來復原服務所進行的變更。

#### 非同步呼叫

服務一律會在它自己的交易中執行。若要確保非同步 SCA 訊息的傳送參與現行導覽交易，則微流程的 **asynchronousInvocation** 限定元必須設為 `commit`。

#### 相關概念

第 25 頁的『商業程序中的補償』

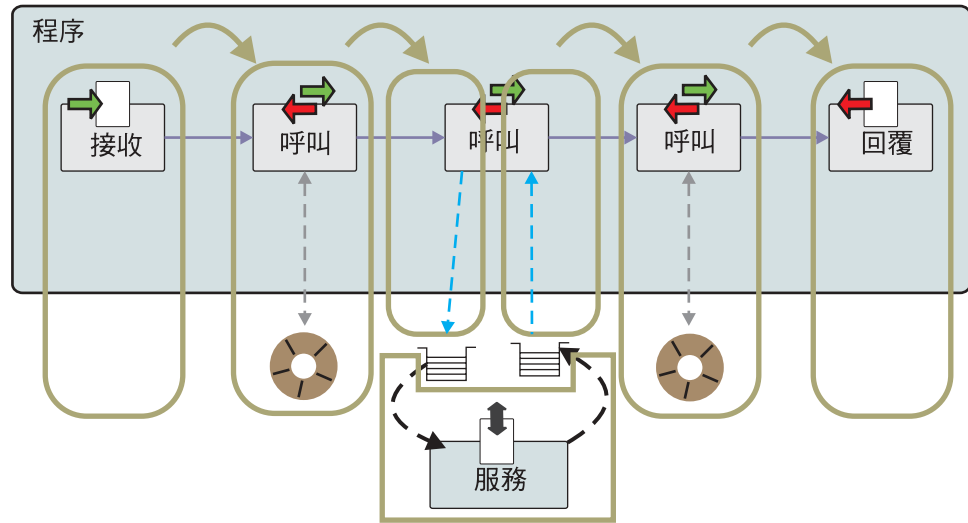
補償是可以復原已順利完成程序中作業的方法。

## 長時間執行的程序的交易式行為

長時間執行程序會跨多個交易。每個交易都是由「Java 訊息服務 (JMS)」訊息或外部事件所觸發。將新訊息放入傳訊佇列中，交易就可以觸發後續交易。

若要容許跨交易界限進行導覽，則程序實例及其活動實例的狀態都會持續保存在資料庫中。

下列圖解顯示長時間執行程序中的每個導覽步驟如何在它自己的交易中執行。導覽步驟可跨多個活動，可藉由呼叫服務的呼叫活動來顯示。而且，也可以在某一交易中執行多個活動。



下列說明長時間執行程序的交易界限。您可以使用交易式行為屬性來影響交易界限。然而，「商業流程管理程式」隨時都可以新增或移除交易界限。

一般而言，在下列狀況下需要有交易界限：

- 等待外部要求時，即到達程序導覽中尚未接收到對應要求的接收活動或挑選活動（也稱為接收選擇活動）時
- 排程等待活動的計時器時
- 使用呼叫活動以非同步方式呼叫服務時
- 呼叫人力作業活動時

此外，「商業流程管理程式」在下列狀況下會引進交易界限。然而，程序設計不得根據這些界限，因為這些界限在程序導覽期間會被置換，或者未來可能會變更。

- 接收到起始程序的接收活動或挑選活動的要求時
- 在程序導覽期間引發錯誤時
- 啟動呼叫活動之前及之後，而此呼叫活動會以同步方式呼叫服務，且此服務未參與程序的交易
- 將生命週期作業延伸至子項程序時，例如，暫停母項程序時，也會在後續交易中暫停它的子項程序
- 要在程序完成時自動刪除程序實例時
- 嘗試回復失敗時，此失敗會導致涉及一系列活動的交易回復
- 使用交易式行為屬性指定時

如果需要保證的交易界限，最好是將需要在單一交易中執行的商業邏輯，分解成微流程或子程序。微流程的邏輯一律在單一交易中執行。

## 影響交易界限

在您建立商業程序模型時，只要變更對應活動的交易式行為屬性，就可以建議呼叫、Snippet 及人力作業活動的交易界限。此屬性可以採用下列其中一個值：

### Commit before

確定現行交易，並啟動新的交易。然後，具有此屬性值的活動會成為新交易的第一個活動。

### Commit after

活動會參與現行交易。活動順利完成之後，會確定交易，並啟動新的交易。會針對每個緊接在後的活動啟動新的交易，而每個後續活動都會成為這些新交易中其中一個交易的第一個活動。

### Participates

活動會參與現行交易。在活動之前或之後，都未設定其他交易界限。

在下列狀況中，此設定容許交易根據活動的交易式行為屬性設定值，繼續導覽後面的活動。

- 如果呼叫活動以非同步方式呼叫服務，則回應訊息的到達會觸發新的交易。交易會在更新呼叫活動狀態之後立即確定，因此十分簡短。
- 在一系列人力作業活動中，每個人力作業活動都需要兩個交易，其中一個交易是啟動人力作業活動，而另一個交易是完成人力作業活動。如果將設定變更為 **Participates**，則可以將每一個人力作業活動的交易數目減少為一個。原因是前一個人力作業活動的完成及下一個活動的啟動都是在相同交易中執行。
- 啟用使用 `completeAndClaimSuccessor` API 之伺服器控制的頁面流程。

### Requires own

活動在它自己的交易中執行。這表示會在活動啟動之前確定現行交易，並會在此活動完成之後啟動新的交易。

如果呼叫活動呼叫未參與現行交易的同步服務，則會略過交易式行為屬性。在此情況下，啟動呼叫活動之前及呼叫活動完成之後，一律都會有交易界限。

## 並行導覽流程活動中的平行分支

若要並行導覽流程活動中的平行分支，則在每個分支的開頭都需要有新的交易界限，這樣會在不同的交易中處理每個平行活動。這表示每個平行分支的第一個活動的交易式行為屬性都必須設為 **Commit before** 或 **Requires own**，才能從流程開頭達成平行化。

**註：**若為 Informix®、Oracle 及 Derby 資料庫系統，則會序列化程序實例中平行分支的導覽交易，亦即，它們無法平行執行。原因是資料庫實體上的鎖定與 DB2® 資料庫上鎖定的精度不同。然而，這類平行分支以非同步方式觸發的服務，仍然會平行執行；只會序列化這些資料庫系統的程序導覽。

## 長時間執行程序中的呼叫服務及交易

使用呼叫活動在長時間執行程序內呼叫的服務，可以參與長時間執行程序的現行交易，也可以在它自己的交易中執行。

下列設定可判斷服務是參與長時間執行程序的交易，還是在它自己的交易中執行。

- 用來呼叫服務的互動樣式。

互動樣式可以是同步也可以是非同步。樣式的判斷是根據目標 SCA 元件或 SCA 匯入的偏好互動樣式，以及作業是單向作業還是要求回應作業，如下表所示：

表 2.

目標元件或匯入的偏好互動樣式	單向作業	要求回應作業
任何	非同步呼叫	非同步呼叫
同步的	同步呼叫	同步呼叫
非同步	非同步呼叫	非同步呼叫

- 指定給程序及所呼叫服務的「服務元件架構 (SCA)」交易限定元：
  - 程序元件參照上的 **suspendTransaction** 限定元，指定是否要將程序的交易環境定義傳送給要呼叫的服務。
  - 服務介面上的 **joinTransaction** 限定元，指定在傳送交易時，服務是否參與其呼叫者的交易。

根據互動樣式及 SCA 限定元的設定，下列是套用至呼叫服務的規則：

#### 同步呼叫

<b>joinTransaction</b>	<b>suspendTransaction = true</b>	<b>suspendTransaction = false</b>
<b>joinTransaction = true</b>	服務未參與長時間執行程序的交易	服務參與長時間執行程序的交易
<b>joinTransaction = false</b>	服務未參與長時間執行程序的交易	服務未參與長時間執行程序的交易

如果服務參與長時間執行程序的現行交易，則只有在確定現行交易時，才會持續保存服務對交易式資源進行的變更。

#### 非同步呼叫

服務一律會在它自己的交易中執行。若要確保非同步 SCA 訊息的傳送參與現行交易，則長時間執行程序的 **asynchronousInvocation** 限定元必須設為 `commit`。

#### 在回復交易時回復成功的服務呼叫

回復行為是視所呼叫服務是否參與現行交易而定。

呼叫活動會呼叫參與現行交易的服務。服務的執行已完成。如果在服務完成之後發生錯誤，而且交易已回復為程序在啟動交易之前的狀態，則也會回復所呼叫服務的效果。重試交易時，會再次呼叫服務。

反之，如果所呼叫服務未參與現行交易，而且所呼叫服務傳回回應，則回應會儲存在不同的交易中。如果在儲存回應之後發生錯誤，則會回復現行交易，並重試交易。在重試期間，不會重新呼叫服務，但是會還原預存回應，而且會繼續進行導覽。

---

## 商業程序中的錯誤處理及補償處理

錯誤是任何可變更商業程序正常處理的異常條件。設計良好的程序應該會考量到錯誤，並盡可能地處理它們。補償是一種處理錯誤的方式。

### 相關概念

第 9 頁的『活動的狀態轉移圖解』

執行活動實例的重要步驟時，活動實例狀態會變更。狀態及狀態轉移是視活動類型而定。

## 錯誤處理

程序中發生錯誤時，導覽會移至錯誤處理程式。您可於呼叫活動、範圍及程序上指定錯誤處理程式。

錯誤處理程式可以捕捉特定錯誤名稱及 (或) 錯誤類型。發生錯誤時，「商業流程管理程式」會嘗試為錯誤找到相符的錯誤處理程式。它會在包括範圍或發生錯誤的活動上尋找錯誤處理程式。它用來選取錯誤處理程式的規則如下：

- 如果錯誤沒有任何相關聯的錯誤資料，則「商業流程管理程式」會搭配使用錯誤處理程式與相符的錯誤名稱。否則，它會使用 `catch-all` 錯誤處理程式 (如果可用)。已定義錯誤變數的錯誤處理程式無法捕捉到不具任何資料的錯誤。
- 如果錯誤具有相關聯的錯誤資料，則「商業流程管理程式」會搭配使用錯誤處理程式與相符的錯誤名稱，以及搭配使用錯誤變數與符合錯誤資料類型的類型。如果找不到與名稱及錯誤資料類型相符的錯誤處理程式，則使用的錯誤處理程式不會有錯誤名稱及類型符合錯誤資料類型的錯誤變數。如果找不到適合的錯誤處理程式，則會使用 `catch-all` 錯誤處理程式 (如果可用)。未定義錯誤變數的錯誤處理程式無法捕捉到具有資料的錯誤。

如果引發的錯誤不符合上述任一種錯誤處理程式定義，則會啟動預設錯誤處理程式。預設錯誤處理程式未明確指定。預設錯誤處理程式會以完成對應範圍的反向順序，來執行立即包括範圍的所有可用補償處理程式，並將錯誤重新擲出給下一個層次，亦即，包括範圍或程序。在此下一個層次上，「商業流程管理程式」會再次嘗試比對錯誤與可用的錯誤處理程式。

如果特定錯誤處理程式及錯誤處理程式鏈結中任意位置的 `catch-all` 錯誤處理程式都捕捉不到錯誤，則錯誤會到達程序範圍，而程序結束時為失敗狀態。即使錯誤處理程式在程序範圍上捕捉到錯誤並進行處理，程序結束時仍為失敗狀態。

## 設計錯誤處理程式

設計錯誤處理程式時，請考慮使用下列選項：

- 捕捉錯誤，並嘗試更正問題，讓商業程序繼續進行以正常完成。
- 捕捉錯誤，並發現無法在此範圍中解析該錯誤。在此情況下，您可以有下列其他選項：
  - 擲出新的錯誤。
  - 重新擲出原始錯誤，讓另一個範圍可以處理該錯誤。
  - 如果這是要求回應作業，則會以錯誤回覆。
  - 呼叫人力作業，以更正問題。如果錯誤處理程式無法解決問題，則可能需要回復程序，並對其進行補償。
  - 若為長時間執行程序，也請考慮在程序上使用 `continueOnError` 參數，以管理方式來處理錯誤。



## 引發錯誤

您可以使用擲出及重新擲出活動來引發錯誤，或使用 Java Snippet 活動以程式設計方式引發錯誤。

若要將錯誤傳送給程序的呼叫者，則可以搭配使用回覆活動與錯誤規格。

### 使用擲出及重新擲出活動引發錯誤

商業程序中的擲出活動可以擲出任何類型的錯誤 (包括標準錯誤)，但是預期的使用型樣是擲出商業錯誤。擲出活動所擲出的異常狀況必須在商業程序內予以捕捉及處理。如果具有要求回應介面的程序未處理程序中的錯誤，則程序會結束，並發生 `bpws:missingReply` 標準錯誤。若為用戶端應用程式，則會以 `StandardFaultException` 物件傳回此錯誤。

您不可以使用擲出活動傳回商業或標準錯誤。您必須使用回覆活動，將商業錯誤傳回給程序用戶端。回覆活動只可以傳回程序所實作介面上定義的商業錯誤。

重新擲出活動可以用於錯誤處理程式，以將錯誤重新擲出給下一個含括範圍。如果想要在現行範圍上執行某些錯誤處理 (例如觸發特定補償處理常式)，且仍然想要讓含括範圍察覺此問題，則這會十分有用。如果現行錯誤處理程式無法處理錯誤，而您想要將錯誤傳送至其中一個含括範圍或程序上定義的錯誤處理程式，也可以使用重新擲出活動。

因為只可以從錯誤處理程式重新擲出現有的錯誤，所以重新擲出活動只可以在錯誤處理程式內使用。

### 以程式設計方式引發錯誤

您可以使用 `raiseFault` 方法，在商業程序的 Java Snippet 中以程式設計方式引發錯誤。您可以利用下列其中一種方式來引發商業錯誤：

- `raiseFault(QName fault, String variableName);`
- `raiseFault(QName fault);`

下列範例會在 `http://process/UpdateCustomerRecordProcess/Interface0/` 名稱空間中建立稱為 `IncompleteData` 的錯誤，然後從 Java Snippet 擲出此錯誤。

```
javax.xml.namespace.QName fault = new javax.xml.namespace.QName  
("http://process/UpdateCustomerRecordProcess/Interface0/", "IncompleteData");  
raiseFault(fault);
```

如果擲出的錯誤未在任何 WSDL 介面上進行宣告，則請將程序的目標名稱空間指定為錯誤的名稱空間。接著，您可以使用捕捉活動，在商業程序中捕捉此錯誤。

請不要直接擲出 `ServiceBusinessException` 物件，而是使用 `raiseFault` 訊息進行。

### 使用回覆活動，將錯誤提供給呼叫者

具有錯誤規格的回覆活動會將指定的錯誤傳送給要求回應作業的呼叫者。回覆活動只可以傳回程序所實作介面上定義的錯誤。在商業程序無法適當地回應所捕捉錯誤，但程序起始者可以回應所捕捉錯誤時，這十分有用。例如，如果呼叫者傳遞商業程序找不到的帳號，則程序應該使用 `AccountNotFound` 錯誤回覆此服務呼叫。



具有錯誤規格的回覆活動不會完成程序，而且會立即傳回給呼叫者。在程序達到結束狀態之前，會繼續進程序的導覽。

## 擷取錯誤的相關資訊

您的程序必須可以處理系統錯誤。您可以使用定義成捕捉 `runtimeFailure` 標準錯誤的錯誤處理程式或 `catch-all` 錯誤處理程式，來捕捉系統錯誤。在部分狀況下，您可能會需要錯誤所提供的資訊。

您可以使用下列其中一種建構來擷取此資訊：

- 錯誤變數，可在標準或系統錯誤事件中用來儲存資料。若要使用具有類型的變數來處理錯誤，您必須手動建立 `StandardFaultType` 複式類型。
- `catch-all` 錯誤處理程式。`catch-all` 錯誤處理程式沒有相關聯的變數。您可以在 `Java Snippet` 活動中使用 `getCurrentFaultAsException` 方法，以從 `catch-all` 錯誤處理程式擷取錯誤資料。您必須在 `catch-all` 錯誤處理程式中併入此 `Java Snippet` 活動。您可以使用 `getCurrentFaultAsException` 方法，來擷取任何錯誤類型的資料，而不只是系統錯誤。

`getCurrentFaultAsException` 方法會將錯誤傳回為 `com.ibm.bpe.api.BpelException` 類型的異常狀況物件。`BpelException` 物件提供數種作業來取得錯誤的相關資訊 (例如錯誤名稱)。`BpelException` 物件會包裝異常狀況實例。因此，您可以存取錯誤訊息及根異常狀況 (如下列範例所示)：

```
com.ibm.bpe.api.BpelException bpeException =
getCurrentFaultAsException();
System.out.println("Fault Name" +
bpeException.getFaultName())
bpeException.printStackTrace( System.out);
Throwable rootCause = bpeException.getRootCause()
```

## 錯誤時繼續行為

在 `WebSphere Integration Developer` 中定義程序時，此參數適用於某些活動。如果錯誤是由那些活動所引發，而且未定義該錯誤的錯誤處理程式，則它可決定程序的處理方式。

如果在活動中偵測到錯誤，則會啟動程序的錯誤處理。如果立即含括範圍未處理錯誤，且未設定 `continueOnError` 參數，則活動會停止，而且會使用管理動作以繼續進行程序。您可以使用「商業程序編排器」API 或「商業程序編排器瀏覽器」，尋找具有已停止活動的執行中程序實例。您可以重新啟動活動，也可以強制完成活動。如果設定 `continueOnError` 參數，則會套用標準錯誤處理。

例如，您可以使用 `forceRetry` 方法以不同的輸入資料來重新啟動活動，或使用 `forceComplete` 方法以輸出或錯誤資料來完成活動。

```
public interface BusinessFlowManagerService {
    public void forceRetry(String aaid, ClientObjectWrapper inputMessage,
        boolean continueOnError);
    ...
}
```

如果想要在再次發生錯誤時停止程序，則必須停用 `continueOnError` 參數。

如果使用 `forceComplete` 方法強制完成活動，就不會再執行活動。它的輸出訊息會用來繼續進程序導覽。

## 商業程序中的補償

補償是可以復原已順利完成程序中作業的方法。

補償處理這種方法可以用來處理其補償已定義於程序模型中的執行中程序實例內的錯誤。補償會回復錯誤發生時確定的作業結果，以回到一致狀態。

您可以定義程序模型中長時間執行的程序及微流程的補償。

### 長時間執行的程序的補償

長時間執行的程序的補償也稱為*商業層次補償*。這類型的補償可以定義於範圍或程序層次。這表示可以補償部分程序或整個程序。

補償可以由範圍或程序的錯誤處理程式或補償處理程式觸發；補償是程序的另一個導覽路徑。

長時間執行的程序會自動補償子項程序，而此程序會在已補償所包括的母項範圍時順利完成。在程序中，只會補償已順利完成的呼叫及範圍活動。

### 微流程的補償

微流程的補償也稱為*技術補償*。這類型的補償會在回復包含微流程的交易或活動階段作業時觸發。因此，復原動作通常指定給無法透過回復交易來回復的活動。執行程序實例時，可補償活動的復原動作會在包括的工作單位中進行登錄。根據回復或確定的結果，啟動補償。

如果微流程是可補償且長時間執行的程序的子項，則微流程的復原動作可在微流程完成時用於母項程序。因此，它可以潛在參與母項程序的補償。對於這些類型的微流程，最好是在定義程序模型時指定程序中所有活動的復原動作。

如果在補償處理期間發生錯誤，則補償動作需要手動解析，才能處理錯誤。您可以使用「商業程序編排器瀏覽器」修復這些補償動作。

#### 相關概念

第 17 頁的『微流程的交易式行爲』

微流程是短期程序。它們可以在交易中執行，或在微流程的 SCA 元件上所指定的活動階段作業中執行。這裡會說明當成交易一部分執行的微流程。

#### 相關資訊



搭配使用程序中的補償與商業程序編排器

## 從基礎架構失敗回復

「商業流程管理程式」提供處理暫時基礎架構失敗的機能。

長時間執行程序會跨多個交易。交易由「Java 訊息服務 (JMS)」訊息區隔，伺服器會將其傳送至訊息驅動 Bean。此 Bean 會將送入訊息傳遞至程序伺服器，以進行處理。每個交易都包含下列動作：

- 接收要求
- 根據要求進行導覽。
- 儲存狀態

- 傳送觸發後續交易的訊息。

伺服器可能無法處理由訊息驅動 Bean 接收的訊息，原因為下列其中一項：

- 無法處理指定數目的連續訊息。因此，會將基礎架構視為無法使用。
- 只能處理部分訊息。任何無法處理的訊息都將視為已損壞。

這些原因的回應如下所示：

原因	回應
無法使用的基礎架構	在一般處理模式中，訊息驅動 Bean 會嘗試一段指定的時間，以回復該狀況。並會嘗試維持所有訊息的可用狀態，直到基礎架構再次變為可作業狀態。例如，此問題可能是由資料庫失敗所導致。
已損壞的訊息	重試指定的次數之後，會將訊息放入存放佇列中。它也可以從存放佇列中移回至輸入佇列，來重試交易。

商業程序的訊息實作如下所示：

- 如果無法處理訊息，則訊息會儲存在保留佇列中，而在可以順利處理後續訊息時會自動從該處重新播放此訊息。
- 當訊息在保留佇列中時，選項如下所示：
  - 當可以順利處理後續的訊息時，所有訊息都會從保留佇列移回至訊息驅動 Bean 的輸入佇列。對於每個訊息，會維護訊息傳送至保留佇列的頻率計數。如果此計數超過給定訊息的重試限制，則會將訊息放置於存放佇列中。
  - 如果無法處理下一則訊息，它也會放置到保留佇列中。此程序會一直持續執行，直到已達到保留佇列中的訊息數上限臨界值為止。當達到此臨界值時，訊息驅動 Bean 會將所有訊息從保留佇列移至輸入佇列，並切換至靜止模式。

當訊息驅動 Bean 以靜止模式運作時，它會定期嘗試處理訊息。無法處理的訊息會放回至存放佇列中，而不會增加遞送計數或保留佇列遍訪計數。一旦可以順利處理訊息後，訊息驅動 Bean 就會切回到一般處理模式。

## 重試限制

重試限制可定義在將訊息放置於存放佇列之前，可以將訊息傳送至保留佇列的次數上限。

若要放置於保留佇列，訊息處理必須失敗三次。

例如，如果重試限制是 5，則訊息必須進入保留佇列五次 (它必須失敗  $3 * 5 = 15$  次)，才能啟動最後的重試迴圈。如果最後的重試迴圈再失敗兩次，則會將訊息放置於存放佇列。這表示訊息必須失敗  $(3 * \text{RetryLimit}) + 2$  次，才能放置於存放佇列。

在可靠基礎架構中執行的效能重要應用程式中，重試限制應該較小：例如一次或兩次。如果重試限制設為零，則會反覆地重試三次失敗訊息，然後該訊息會立即進入存放佇列中。

此「商業流程管理程式」內容是在管理主控台中進行指定，請按一下伺服器 → 應用程式伺服器 → **server\_name**。然後，在儲存器設定區段中，按一下商業程序編排器儲存器設定 → 商業程序編排器儲存器。並在其他內容區段中，按一下商業流程管理程式。

## 保留佇列訊息限制

保留佇列訊息限制會定義可以位於保留佇列上的訊息數目上限。如果保留佇列溢位，則系統會變為靜止模式。若要讓系統在一個訊息失敗時立即進入靜止模式，請將該值設為零。若要讓「商業流程管理程式」可容忍更多基礎架構失敗，請增加該值。

此「商業流程管理程式」內容是在管理主控台中進行指定，請按一下**伺服器** → **應用程式伺服器** → **server\_name**。然後，在**儲存器設定**區段中，按一下**商業程序編排器儲存器設定** → **商業程序編排器儲存器**。並在**其他內容**區段中，按一下**商業流程管理程式**。

## 保留佇列

保留佇列會保留失敗訊息，可透過將其移回至「商業流程管理程式」的內部工作佇列來重新播放訊息。如果訊息失敗三次，則會將其放置於保留佇列上。如果訊息失敗 ( $3 * RetryLimit$ ) + 2 次，會將其放置於存放佇列。如果保留佇列已滿，達到由保留佇列訊息限制所定義的限制，且另一則訊息失敗，則佇列會溢位，且系統會變為靜止模式。管理者可以查詢及重新播放失敗訊息，以將此佇列中的訊息移回內部佇列。

## 存放佇列

存放佇列包含失敗 ( $3 * RetryLimit$ ) + 2 次的訊息。管理者可以查詢及重新播放失敗訊息，以將此佇列中的訊息移回內部佇列。

## 重新播放訊息

管理者可以將訊息從存放或保留佇列移回至內部佇列。此動作可以使用管理主控台或使用管理指令來完成。

## 靜止模式

當保留佇列溢位時，會進入靜止模式。當發生此情況時，會假設發生嚴重 (儘管可能是暫時性的) 基礎架構失敗。靜止模式的目的是要防止系統使用太多資源，而基礎架構失敗表示大部分訊息都可能失敗。在靜止模式中，系統會休眠兩秒，然後嘗試處理下一則訊息。一旦順利處理訊息，系統會回復一般訊息處理。

## 人力作業的失敗訊息處理

「人力作業管理程式」不需要保留佇列，也沒有重試限制。它只有可在其上放置失敗訊息的存放佇列，並可以從該佇列重新播放訊息。

### 相關工作

第 249 頁的『查詢及重新播放失敗的訊息，使用管理主控台』

本文說明如何檢查和重新播放無法處理的商業程序或人力作業的任何訊息。

第 264 頁的『查詢及重新播放失敗的訊息，使用管理指令』

使用管理指令來判斷商業程序或人力作業是否有任何失敗訊息，如果有的話，則嘗試重新處理。

第 250 頁的『重新整理失敗訊息計數』

使用管理主控台來重新整理商業程序或人力作業的失敗訊息計數。

## 程序的授權及人員指派

授權是用來將特定專用權指派給特定使用者或特定使用者群組，並決定使用者可對程序及活動採取的動作。

商業程序的授權是使用人力作業予以實現。授權角色是用來定義可用於特定角色的動作集。指定給人力作業的角色是由相關聯的商業程序及活動繼承。因此，例如，如果在商業程序中建立行內人力作業的模型，則作業的擁有者會自動成為活動擁有者。每個活動角色都只會符合一個人力作業角色。「商業流程管理程式」會使用活動角色來進行導覽及授權。

### 商業程序的授權角色

角色是指具有相同授權層次的一組人員。您可以針對商業程序執行的動作，視您的授權角色而定。此角色可以是 J2EE 角色或實例型角色。

#### 相關概念

第 69 頁的『人員解析』

人員解析會根據一組參數化查詢表示式 (稱為人員指派準則)，以從人員目錄擷取使用者資訊。

第 57 頁的『人力作業的授權角色』

您可以針對人力作業執行的動作，視授權角色而定。此角色可以是系統層次 J2EE 角色或實例型角色。

### 商業程序的 J2EE 角色

配置「商業程序編排器」時，會設定 J2EE 角色。針對 J2EE 角色型授權，您必須要配置使用者登錄並啟用廣域安全。

下列是支援的程序 Java 2 Platform Enterprise Edition (J2EE) 角色：

- **BPESystemAdministrator**。指派給此角色的使用者會擁有所有專用權。此角色也稱為商業程序的系統管理者。
- **BPESystemMonitor**。指派給此角色的使用者可以檢視所有商業程序物件的內容。此角色也稱為商業程序的系統監視者。
- **JMSAPIUser**。「商業流程管理程式」JMS API 要求會代表此角色所對映的使用者 ID 來執行，而不管呼叫者是誰。

您可以使用管理主控台來變更這些角色的使用者及群組指派。

### 商業程序及活動的實例型角色

針對程序及活動都提供有預先定義的授權角色集。在您建立程序模型時，可以將這些角色指派給程序及活動。使用者與實例型角色的關聯，是在執行時期使用人員解析所決定的。

### 程序上動作的授權角色

指派給程序角色的人員已獲授權可以執行下列動作：

角色	授權的動作
程序起始者	檢視關聯程序實例的內容，以及其輸入及輸出訊息。



角色	授權的動作
程序讀者	檢視關聯程序實例的內容，以及其輸入及輸出訊息。此角色成員也會自動成為活動的讀者，以及與人力作業活動相關聯的行內待辦作業 (包括子作業、後續作業及呈報)。
程序管理者	管理程序實例、介入已起始的程序，以及建立、刪除及轉移工作項目。此角色成員也會自動成為活動的管理者，以及與人力作業活動相關聯的行內待辦作業 (包括子作業、後續作業及呈報)。
程序活動管理者	管理程序的活動。

程序起始者是「商業流程管理程式」用於程序導覽及外部服務呼叫的角色。如果程序實例仍然存在於資料庫中，請不要從使用者登錄中刪除程序起始者的使用者 ID，以便繼續進行程序的導覽。

使用人力作業可以將使用者指派給這些角色。

角色	人員指派
程序起始者	將行內人力作業指派給程序的起始接收或挑選 (接收選擇) 活動，就可以指定程序起始者。
程序讀者	在與程序相關聯的管理作業上設定讀者角色，就可以指定程序讀者。程序中的所有活動都會繼承此角色。
程序管理者	程序管理者是透過指派給程序的管理作業所定義的。程序中的所有活動都會繼承此角色。
程序活動管理者	程序活動管理者是透過與程序相關聯的管理作業所定義的。在此作業上定義的管理者角色也會作為程序活動管理者。 <b>註:</b> 此管理作業不同於用來決定程序管理者的管理作業。  程序層次上定義的活動管理作業是未定義管理作業的活動的預設管理作業。

## 活動上動作的授權角色

如果建立人力作業的模型，並將該人力作業併入為商業程序中的人力作業活動，則作業的擁有者會自動成為活動擁有者。針對人力作業定義的角色成員，會繼承對應人力作業活動上的相同角色。「商業流程管理程式」會使用活動角色來進行導覽及授權。行內呼叫作業的可能起始者會是相關聯接收或挑選 (接收選擇) 活動的可能起始者，或事件處理常式的可能起始者。

活動的實例型角色已獲授權可以執行下列動作：

角色	授權的動作
活動讀者	檢視關聯活動實例的內容，以及其輸入及輸出訊息。
活動編輯者	授權給活動讀者的動作，以及與活動相關聯的訊息及其他資料的寫入權。
可能的活動起始者	授權給活動讀者的動作。此角色的成員可以傳送訊息來接收或挑選活動。
可能的活動擁有者	授權給活動讀者的動作。此角色的成員可以要求活動。
活動擁有者	處理及完成活動。此角色的成員可以將其工作項目轉移給管理者或可能的擁有者。

角色	授權的動作
活動管理者	修復因非預期錯誤而停止的活動，並強制完成長時間執行的活動。

## 程序角色的預設人員指派

如果未定義某些角色的人員指派準則，或者人員解析失敗或未傳回結果，則會指派預設人員。下表顯示會套用哪些預設值。

商業程序的角色	如果程序模型中未定義角色 ...
程序管理者	程序起始者變成程序管理者
程序讀者	沒有讀者

此外，如果未定義用來建立及啟動商業程序的呼叫作業，則會將預設人員指派準則 **Everybody** 用於程序的可能起始者。

## 建立及啟動商業程序的授權

容許建立及啟動程序的使用者集是透過下列兩種作業決定：與接收或挑選 (接收選擇) 活動 (用來建立及啟動新的程序實例) 相關聯的呼叫作業，以及與程序相關聯的管理作業。商業程序會繼承您指派給這些作業的角色。

您可以利用下列方式，使用人力作業來建立及啟動商業程序：

- 將行內呼叫作業指派給程序的起始接收或挑選 (接收選擇) 活動

部分商業程序可能會變更機密商業資料，因此只有授權的人員才有權可以建立及啟動這些程序。若為這類型的商業程序，則指定程序範本的行內呼叫作業，就可以將人力作業指派給程序的起始接收活動。定義給行內呼叫作業的可能起始者會成為程序的可能起始者。

使用「人力作業管理程式」API 來建立及啟動呼叫作業，或使用「商業流程管理程式」API 來起始程序，都可以啟動程序。這兩種方式都會導致相同的授權檢查。如果未指定行內作業，則每個人都可以啟動程序。

- 將獨立式呼叫作業指派給程序的起始接收或挑選 (接收選擇) 活動

您也可以使用連接至商業程序的獨立式呼叫作業，在啟動程序時執行授權檢查。然而，如果使用獨立式呼叫作業，請考慮下列幾點：

- 只有在呼叫作業啟動程序時，才會進行授權檢查，亦即，使用直接連接至程序元件的「商業流程管理程式」API 或 SCA 用戶端來啟動程序時，可以略過此檢查。
- 在行內作業使用內部介面時，它會使用 SCA 基礎架構來呼叫程序。因此，行內呼叫作業的執行效果優於獨立式作業。
- 您無法存取人員指派準則定義中的程序環境定義。這表示獨立式作業不支援根據程序環境定義的動態人員指派。

- 將管理作業指派給程序。

管理作業的管理者角色是透過程序繼承。程序管理者可以建立及啟動程序。



## 管理商業程序的授權

您可以使用管理作業來授權使用者或使用者群組，以在商業程序與其相關聯活動上執行管理動作

### 程序管理

若要定義容許哪些使用者執行管理動作及讀取程序資料，您可以將管理作業指定為長時間執行商業程序的一部分。管理作業的管理者及讀者角色可決定誰是程序管理者及程序讀者。例如，程序管理者可以強制終止程序實例。管理作業與每個商業程序相關聯。如果未在 **WebSphere Integration Developer** 中建立程序的管理作業模型，則會在執行時期建立預設管理作業。此預設作業會將程序起始者定義為程序管理者，而且不會指派任何程序讀者。

### 活動管理

您可以為每個呼叫或 **Snippet** 活動建立管理作業模型。此作業可決定除了程序管理者之外還有誰可以管理活動。您也可以從程序層次上建立活動的預設管理作業模型，而此作業適用於每個未指派管理作業的呼叫或 **Snippet** 活動。活動管理作業的管理者角色可決定誰可以管理對應的活動。例如，活動管理者及程序管理者可以強制重試活動。

呼叫活動具有相關聯的管理作業。若為同步呼叫活動，則只有在活動於呼叫失敗之後停止時才會建立此作業。接著，會使用管理作業來處理修復要求，例如強制完成及強制重試。若為非同步呼叫活動，則一律會建立管理作業。因此，管理者可以在活動等待非同步回應時強制重試或強制完成活動。



---

## 第 2 章 關於人力作業

人力作業是容許人員及服務互動的元件。

部分人力作業代表人員的待辦作業。這些作業可以由人員起始，也可以由自動化服務起始。人力作業可用來實作需要人機互動的商業程序中的活動，例如，手動異常狀況處理及核准。其他人力作業則可以用來呼叫服務，或協調人員之間的協同作業。然而，不論作業的起始方式為何，都是由人員群組中指派進行作業的人員來執行與作業相關聯的工作。

以靜態方式，或指定在執行時期使用人員目錄解析的準則 (例如角色或群組)，都可以將人員指派給人力作業。也可以使用人力作業的輸入資料或商業程序的資料來尋找處理作業的正確人員。

### 相關工作

第 289 頁的第 9 章, 『管理商業程序及人力作業』

商業程序和人力作業已部署和安裝成爲企業應用程式的一部分。您可以使用管理主控台或管理指令來管理程序範本和作業範本，也可以使用「商業程序編排器瀏覽器」來處理程序實例和作業實例。請使用「商業程序編排器觀察程式」來報告商業程序和人力作業。

第 298 頁的『管理作業範本及作業實例』

使用管理主控台或管理指令，可以管理作業範本。使用「商業程序編排器瀏覽器」，可以使用作業實例。

第 298 頁的『利用管理主控台停止及啓動作業範本』

使用管理主控台來個別啓動及停止每一個已安裝的作業範本。

第 298 頁的『利用管理指令停止及啓動作業範本』

除了使用管理主控台之外，管理指令也可以用來停用和啓動作業範本。使用管理指令，停止企業應用程式內的所有作業範本。

第 299 頁的『建立及啓動作業實例』

您可以從任何您有權使用的作業範本建立及啓動作業實例。

第 300 頁的『使用您的作業』

若要使用作業，您必須要求作業，然後執行完成它所需的動作。

第 300 頁的『暫停及回復作業實例』

您可以暫停作業實例。例如，若要修正導致作業實例失敗的問題，您可能想要執行此作業。而在符合作業的必備項目後，就可以回復執行中的作業實例。

第 301 頁的『管理人力作業的優先順序』

您可以使用人力作業的優先順序，來過濾作業及排序您的作業清單。

第 301 頁的『管理指派的工作』

在啓動作業之後，您可能需要管理作業所指派的工作，例如，更適當地分配工作群組成員的工作負荷。

第 303 頁的『指定缺席設定』

如果您想要離開辦公室一段時間，請指定作業的代理人。

第 304 頁的『指定使用者的缺席設定』

如果使用者無法處理作業，例如，如果他們因病休假，請指定使用者作業的代理人。

第 302 頁的『轉移您擁有的作業』

如果您是作業的擁有者，則您可能需要將作業轉給另一位使用者，例如，如果其他人需要提供資訊，才能完成作業。

第 302 頁的『轉移工作項目，您為這些項目的作業啟動者、發送者或管理者』

您可能需要在開始作業之後變更工作指派。例如，如果作業擁有者休假中，而且作業必須在這個人回來之前完成，則您可能想要將工作項目轉移給另一位使用者。轉移工作項目的方法視您具有的角色及作業狀態而定。

第 305 頁的『建立工作項目』

例如，當沒有現行可能的擁有者可以接受任何其他工作時，您可能想要為新的可能擁有者建立工作項目。如果對人員目錄的查詢未傳回任何可能的擁有者，則您也可能想要建立工作項目。例如，如果自從啟動程序後組織已變更，則這種情況可能發生在長時間執行的程序中。

第 306 頁的『刪除工作項目』

例如，如果您建立了錯誤的工作項目，或者如果對不再替公司工作的某人產生工作項目，則您可能想要刪除工作項目。

第 306 頁的『檢視作業呈報』

呈報會通知呈報接收者，指出使用者可能發生問題，無法及時完成他們的指派作業。

第 307 頁的『傳送電子郵件以便呈報』

當作業逾期時，它可能導致呈報。您可以設定系統，傳送電子郵件至委派的人員，以通知他們關於呈報的資訊。

---

## 作業範本

人力作業範本包含已部署作業模型的定義，此定義是使用 WebSphere Integration Developer 所建立，或是使用「商業程序編排器」API 於執行時期建立的。

範本包含內容 (例如，作業名稱及優先順序)，並聚集成品 (例如，呈報範本、自訂內容及人員查詢範本)。除了建立作業範本模型時指定的內容之外，安裝的作業範本也可以具有下列其中一種狀態：

**已啟動** 啓動作業範本時，可以啓動新的範本實例。

**已停止** 必須先停止作業範本，才可以解除安裝人力作業應用程式。作業範本處於停止狀態時，無法啓動此範本的新實例。

您可以建立 `com.ibm.task.api.TaskModel` 類別的實例，以在執行時期建立待辦或協同作業的模型。然後，可以使用這些實例來建立可重複使用的作業範本，或直接建立執行一次的作業實例。在執行時期建立人力作業的模型是以 Eclipse Modeling Framework (EMF) 為基礎。

### 相關工作

第 394 頁的『在執行時期建立作業範本及作業實例』

您通常會使用塑型工具 (如 WebSphere Integration Developer) 來建置作業範本。然

後在 WebSphere Process Server 中安裝作業範本，並從這些範本建立實例 (例如，使用「商業程序編排器瀏覽器」)。然而，您也可以執行時期建立人工或參與作業實例或範本。

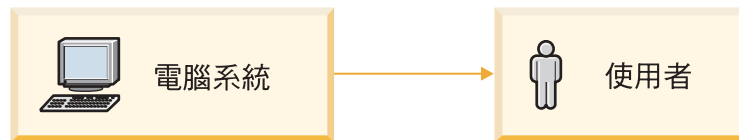
## 人力作業類型

作業類型是衍生自在建立模型期間指派的作業範本類型。

人力作業類型如下所示：

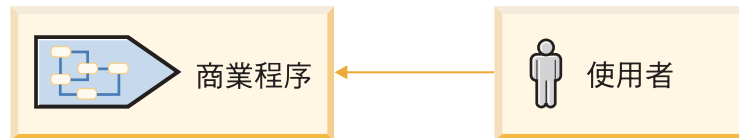
### 待辦 (參與) 作業

支援 Web 服務至人員的互動，這可讓人員實作服務。例如，待辦作業可以是商業程序中的人力作業活動。



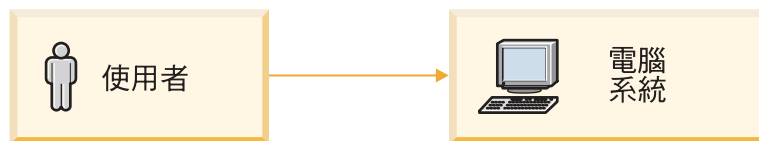
### 管理作業

管理者使用管理作業來解決程序中發生的技術問題。目前，您只可以使用商業程序的管理作業。



### 呼叫 (起始) 作業

支援人員與 Web 服務互動，這可讓人員建立及啟動服務。例如，使用者可以啟動商業程序，或者使用呼叫作業來傳送事件。



### 協同作業 (純人力) 作業

支援人員至人員的互動，這可讓人員與其他人員以結構化及受控制的方法共用。協同作業不會與商業程序或其他 Web 服務直接互動。



## 人力作業的版本管理

想要相同人力作業的替代實例並存於執行時期環境時，請使用版本管理。

在 WebSphere Integration Developer 中建立獨立式人力作業模型時，可以併入版本管理資訊。作業範本的版本是透過它的有效起始日期所決定。這表示不同版本的作業範本可以有相同的作業範本名稱，但是具有不同的有效起始日期。在執行時期使用的作業範本版本是由作業是用於早期連結範例情況或晚期連結範例情況而決定。

### 早期連結

在早期連結範例情況中，是於建立模型期間或部署作業模型時決定使用的作業範本版本。呼叫元件會根據「服務元件架構 (SCA)」連線，來呼叫專用的靜態連結作業範本。即使存在有根據其有效起始日期而有效的另一個作業範本版本，還是會使用現行靜態連接的作業範本，並略過其他所有版本。

SCA 連接是早期連結的一個範例。如果您將獨立式參照連接至人力作業元件，則每次使用此參照呼叫作業範本時，都會將目標設為由人力作業元件代表的特定版本。

### 晚期連結

在晚期連結範例情況中，是於建立作業實例時決定使用的人力作業範本。在此情況下，會使用目前有效的作業範本版本。作業範本的較新版本會取代所有舊的範本版本。現有的作業實例會繼續執行作業範本，作業實例是在啟動時與該作業範本相關聯。這會產生下列種類的作業範本：

- 目前有效的作業範本用於新的作業實例
- 不再有效的作業範本，對於執行中作業實例而言可能仍然有效
- 作業範本會根據其有效起始日期，在以後變為有效的作業範本

晚期連結的範例是在「商業程序編排器瀏覽器」中呼叫新作業時。建立的實例一律會根據作業範本的最新版本，而此作業範本的有效起始日期不是以後的日期。後續作業及子作業一律會使用晚期連結進行呼叫。

---

## 作業實例

作業實例是作業範本的執行時期版本。

---

## 獨立式及行內作業

「服務導向架構 (SOA)」型樣建議具有一組鬆散耦合元件的軟體解決方案的實現。遵循 SOA 型樣的人力作業稱為獨立式作業，而定義為商業程序一部分的人力作業稱為行內作業。

下表顯示可用於獨立式及行內作業的作業類型：

表 3.

實作	呼叫作業	待辦作業	協同作業	管理作業
獨立式	是	是	是	否
行內	是	是	否	是

### 獨立式作業

獨立式作業遵循 SOA 型樣，而且它們與呼叫它們的元件 (待辦作業) 或它們所呼叫的元件 (呼叫作業) 鬆散連結。它們可以使用「服務元件架構 (SCA)」基礎架構連接至另一個元件。

它們會與其友機元件 (與 SCA 方法互斥) 進行通訊。亦即，待辦作業會接收輸入訊息並傳回輸出或錯誤訊息，而呼叫作業會傳送輸入訊息並接收輸出或錯誤訊息。且不會進行進一步資訊交換或生命週期控制。

因為獨立式作業是分開建立模型，所以可以重複使用。獨立式作業一律會發出它們的「共用事件基礎架構 (CEI)」及審核日誌事件作為「人力作業管理程式」事件。

使用下列方式，可以將獨立式作業製作為 SCA 元件：

- 待辦作業具有的介面可以連接至用戶端元件。
- 呼叫作業具有的參照可以連接至要呼叫的服務。
- 協同作業是自行包含的 SCA 元件。雖然協同作業是獨立式作業，但是它們具有兩個人力介面，因此無法連接至服務元件。

管理作業不可以作為獨立式作業，也不可以作為 SCA 元件。

下列規則適用於與商業程序搭配使用的獨立式作業：

- 它們的生命週期與程序無關。
  - 待辦作業預設是由程序所建立，而且會在刪除程序時一併刪除。
  - 呼叫作業可以建立程序。然而，在刪除程序時不會刪除它們，因此可以看到作業的結果。
- 在商業程序中，待辦作業是作為呼叫活動。
- 呼叫作業會連接至接收或挑選活動 (也稱為接收選擇活動)，或連接至 on-event 事件處理常式。
- 作業說明、顯示名稱及文件平行支援多種語言。
- 獨立式作業無法存取程序環境定義。它們無法存取其他程序活動的程序變數、自訂內容或資料。

如果協同作業是最上層作業，則會獨立管理它的生命週期，而且可以手動刪除，或在指定的時間長度之後自動刪除。如果協同作業是子作業或後續作業，則它的生命週期是由其母項或最上層作業所管理。

## 行內作業

行內作業是商業程序的必要部分。行內作業可以是待辦作業、呼叫作業及管理作業。因為協同作業利用人員之間的互動，而不是直接與程序互動，則它們不可以是行內作業。行內作業不會顯示為 SCA 元件 (無法連接)，也無法重複使用於其他程序或活動。

行內作業可以存取程序環境定義及程序資料 (例如，程序變數、自訂內容及活動資料)。這可以用於牽涉區分責任的作業。行內待辦作業會發出它們的 CEI 及審核日誌事件作為「商業流程管理程式」人力作業活動事件。它們的子作業及後續作業會發出事件作為「人力作業管理程式」事件。

下列是適用於行內作業的規則：

- 待辦作業是程序中的人力作業活動。它們會共用相同狀態，但是人力作業活動不會反映作業子狀態。
- 呼叫作業與接收或挑選 (接收選擇) 活動，或與 on-event 事件處理常式相關聯。
- 管理作業會附加至程序或程序中的活動。
- 生命週期通常是由程序所決定。



- 待辦作業及管理作業是由商業程序所建立，而且會隨著程序一起刪除。
- 如果呼叫作業是由商業程序所建立及啟動，則它們的生命週期是由程序所決定，而且會隨著程序一起刪除。如果它們是使用「人力作業管理程式」API 所建立及啟動，則它們的生命週期會與程序無關，而且甚至在刪除程序之後，還可以顯示它們的結果。
- 待辦及呼叫作業說明、顯示名稱及文件只支援一種語言。
- 行內作業沒有期限前持續時間。然而，對應於待辦作業的人力作業活動可以定義期限。
- 只有行內呼叫作業具有刪除前持續時間，但是只適用於使用「人力作業管理程式」API 啟動作業時。
- 行內作業上的更新動作只支援一小部分的作業內容。只有未呈現於程序或活動中的作業內容才可以進行更新。如需 `update` 方法的相關資訊，請參閱 `com.ibm.task.api` 套件中 `HumanTaskManager` 介面的 Javadoc。

行內作業是用於程序授權：

- 待辦作業的角色讀者、管理者、可能的擁有者、擁有者及編輯者，會與程序中人力作業活動的對應角色相同。
- 行內呼叫作業的可能起始者角色，可決定誰可以呼叫及傳送訊息給對應的接收或挑選 (接收選擇) 活動，或 `on-event` 事件處理常式。請注意，可能的起始者及可能的實例建立者角色具有相同的人員指派。如果未定義行內呼叫作業，則每個人都有權可以啟動活動或事件處理程式。
- 程序管理作業的管理者及讀者角色可決定誰是程序管理者或程序讀者。程序管理者可以進行強制終止程序實例這類作業。
- 活動管理作業的管理者角色可決定誰可以管理對應的活動。活動管理者及程序管理者可以進行強制重試活動這類的作業。
- 程序讀者及程序管理者授權是透過每個程序活動或行內人力作業予以繼承。

## 人力作業與商業程序的關係

呼叫作業可以與接收或挑選 (接收選擇) 活動，或與 `on-event` 事件處理常式相關聯。這些作業可以是行內或獨立式作業。如果使用「商業流程管理程式」API，則只有行內呼叫作業可以影響呼叫接收活動的授權。每個人預設都可以將訊息傳送至接收或挑選活動或 `on-event` 事件處理程式。這包括在起始接收活動時呼叫商業程序。

管理作業是與每個商業程序相關聯。管理作業可決定誰有權管理及讀取程序。如果未在 `WebSphere Integration Developer` 中建立程序的管理作業模型，則會在執行時期建立預設特定作業。此預設作業會將程序起始者定義為程序管理者，而且不會指派程序的讀者。

您可以為每個呼叫或 `Snippet` 活動建立管理作業模型。此作業可決定除了程序管理者之外還有誰可以管理活動。您也可以建立預設活動管理作業的模型，而此作業適用於每個未指派明確管理作業的呼叫或 `Snippet` 活動。

呼叫活動具有相關聯的管理作業。若為 `Snippet` 活動及同步呼叫活動，則只有在活動於呼叫失敗之後停止時才會建立此作業。接著，會使用管理作業來處理修復要求 (例如強制完成及強制重試)。若為非同步呼叫活動，則一律會建立管理作業。因此，管理者可以在活動等待非同步回應時強制重試或強制完成活動。

獨立式待辦作業可以實作非同步呼叫活動。這些活動也具有相關聯的管理作業。行內待辦作業會實作人力作業活動。而在執行時期時，會建立這些活動的管理作業。

### 相關概念

第 45 頁的『人力作業的生命週期』

人員與 Web 服務或商業程序互動時，人力作業可以支援人員。作業生命期限期間進行的互動，視作業是待辦作業、協同作業、呼叫作業或管理作業而定。特定互動只有在特定作業狀態下才可以進行，因而，這些互動會對應地影響人力作業的狀態。

第 28 頁的『商業程序及活動的實例型角色』

針對程序及活動都提供有預先定義的授權角色集。在您建立程序模型時，可以將這些角色指派給程序及活動。使用者與實例型角色的關聯，是在執行時期使用人員解析所決定的。

---

## 子作業

如果人員需要將他們的一部分指派工作委任給其他人員，但想要控制整體結果，則子作業可以支援人員。它們也可以用來呼叫支援服務，以協助人員完成所使用的作業。

透過儲存在「商業程序編排器」資料庫中的獨立式作業範本、執行時期建立的作業範本，或在執行時期提供新的作業模型，就可以建立子作業。母項作業可以是待辦作業或協同作業，而且它的 **supportsSubtask** 屬性必須設為 **true**。建立的子作業可以是協同作業或呼叫作業。這些子作業因而可具有子作業或後續作業。

輸入訊息類型或輸出訊息類型並沒有任何限制。然而，子作業起始者必須提供輸入訊息。完成子作業時，母項作業的擁有者可以將子作業輸出資料對映至母項作業的輸出訊息。

### 授權考量

除了針對啟動時為子作業指定的內容之外，子作業也會繼承其母項作業的授權角色：

- 母項作業的讀者、編輯者、發送者及擁有者會成為子作業及其呈報的讀者
- 母項作業的管理者會成為子作業的管理者
- 母項作業的呈報接收者會成為子作業的讀者

### 生命週期考量

啟動第一個子作業時，母項作業會進入等待子作業子狀態。最後一個子作業達到其中一個結束狀態 (已完成、失敗、過期或終止) 之前，母項作業都會維持為這個子狀態。母項作業的某些生命週期作業 (狀態變更) 會延伸至其子作業。因此，母項作業暫停、回復、終止、刪除或到期時，它的所有子作業也會暫停、回復、終止、刪除或到期。但不會延伸母項作業的已呈報子狀態；呈報母項作業時也不會呈報子作業。子作業具有它們自己的呈報，而且只有在觸發它們自己的其中一個呈報時，才會設定它們的已呈報子狀態。

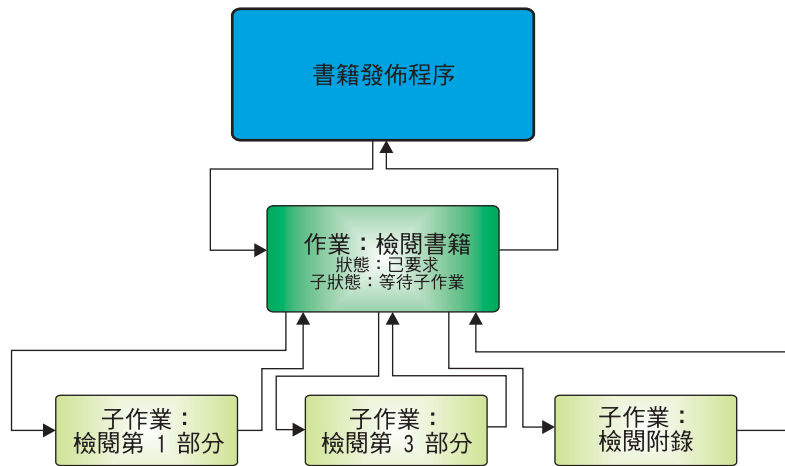
子作業的部分生命週期作業會與母項作業的生命週期作業發生衝突，因此不容許。這些是影響子作業生命週期結束的主要作業，而且需要與母項進行協調。下列是可以在子作業上執行的作業：

- 一律支援未與母項作業發生衝突的生命週期作業。這些是子作業或進一步後續作業的要求、取消要求、完成、建立及啟動等這類作業。
- 子作業可以到期。

- 因為可能需要停止處理子作業，但是母項作業會繼續進行處理，所以可以暫停及回復子作業。
- 可以終止子作業。
- 子作業可以具有它們自己的呈報，因此母項作業擁有者及子作業發送者較能控制子作業的進度。
- 若為啟動為子作業的作業，則會略過自動刪除設定。子作業會在刪除其母項作業時一併刪除。不支援使用「商業程序編排器」API 刪除個別子作業。

### 範例：母項作業與協同作業之間的互動

下圖顯示具有人力作業活動之子作業的書籍發佈程序。



在書籍發佈程序中，「檢閱書籍」作業由 Linda 要求。她發現書籍太大，無法一人獨自檢閱完成，而且書籍的某些部分還牽涉到特殊知識。因此她決定不採用標準發佈程序，並將她的一部分作業指派給她的一些同事。她透過「檢閱書籍章節」範本建立了三個額外作業：「檢閱第 1 部分」、「檢閱第 3 部分」及「檢閱附錄」。她自己會檢閱書籍的第 2 部分。

她將整本書作為子作業的輸入，這樣她的同事可以有足夠的環境定義資訊，但是要新增附註至作業說明，以告知她的同事只檢閱書籍中已指派給他們的部分。她將作業指派給她的同事：John 檢閱第 1 部分、Cindy 檢閱第 3 部分，Mary 則檢閱附錄。然後，她將這三個作業啟動為她自己的「檢閱書籍」作業的子作業。在這三個子作業完成之前，她處於已要求狀態的作業會進入等待子作業子狀態。

Cindy、John 及 Mary 要求他們的子作業，並開始檢閱該書所分到的部分。同時，Linda 會檢閱書籍的第 2 部分。在她完成負責的檢閱部分時，會檢查她同事的進度。Cindy 及 John 已完成檢閱，但是 Mary 仍在檢閱較大的附錄。Linda 的作業仍然處於等待子作業子狀態。雖然 Linda 還無法完成她的作業，但是她可以根據 Cindy 及 John 的子作業輸出開始合併檢閱意見。

同時，Mary 也完成她的子作業，因此 Linda 的「檢閱書籍」作業會離開等待子作業子狀態。現在，Linda 會合併 Mary 的檢閱意見與這本書的其餘部分，並完成她的作業。書籍發佈程序會繼續進行。因為「檢閱書籍」作業是行內人力作業，所以刪除商業程序實例時也會一併刪除它與它的子作業。

## 範例：母項作業與呼叫作業之間的互動

母項作業和呼叫作業之間的互動，與母項作業和協同作業之間的互動類似。作業擁有者會透過現有的呼叫作業範本建立作業，並將它啟動為專屬作業的子作業。母項作業會進入等待子作業子狀態，並等待呼叫子作業返回。所有子作業完成時，母項作業會離開等待子作業子狀態，而且母項作業會完成。

### 相關概念

第 57 頁的『人力作業的授權角色』

您可以針對人力作業執行的動作，視授權角色而定。此角色可以是系統層次 J2EE 角色或實例型角色。

第 45 頁的『人力作業的生命週期』

人員與 Web 服務或商業程序互動時，人力作業可以支援人員。作業生命期期間進行的互動，視作業是待辦作業、協同作業、呼叫作業或管理作業而定。特定互動只有在特定作業狀態下才可以進行，因而，這些互動會對應地影響人力作業的狀態。

---

## 後續作業

如果人員想要將他們的一部分指派工作委任給其他人員，並控制工作的完成，則後續作業可以支援人員。

透過儲存在「商業程序編排器」資料庫中的獨立式作業範本、執行時期建立的作業範本，或在執行時期提供新的作業模型，就可以建立後續作業。後續作業可以具有它自己導致作業鏈結的後續作業。

後續作業只可以是協同作業。您可以從待辦作業或協同作業中啟動 **supportsFollowOnTask** 屬性設為 **true** 的協同作業。

後續作業的輸入訊息類型可以與它的上一個作業不同。如果後續作業的輸入訊息類型與上一個作業的輸入訊息類型相同，則會自動將上一個作業的輸入訊息內容傳遞給後續作業。建立或啟動後續作業時，可以改寫訊息內容。

若為後續作業的鏈結，則每個後續作業的輸出及錯誤訊息類型必須與鏈結中最上層作業的輸出及錯誤訊息類型相同，因為鏈結中的最後一個後續作業會將訊息傳回給呼叫元件或人員 (發送者)。母項作業的輸出或錯誤訊息內容一律會複製至後續作業的輸出或錯誤訊息。您可以在後續作業中修改這些訊息。

### 授權考量

後續作業會繼承上一個作業的授權角色：

- 上一個作業的讀者、編輯者、發送者及擁有者會成為後續作業及其呈報的讀者
- 上一個作業的管理者會成為後續作業的管理者
- 上一個作業的呈報接收者會成為後續作業的讀者

### 生命週期考量

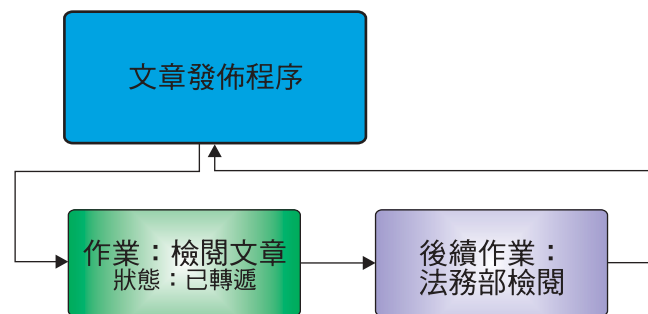
啟動後續作業時，上一個作業會進入已轉遞狀態。後續作業是其上一個作業的子項，因此上一個作業的某些生命週期作業 (狀態變更) 會延伸至其後續作業。上一個作業暫停、回復、呈報、終止、刪除或到期時，它的所有後續作業也會暫停、回復、呈報、終止、刪除或到期。

後續作業的部分生命週期作業會與上一個作業的生命週期作業發生衝突，因此不容許。這些是影響後續作業生命週期結束的主要作業，而且需要與上一個作業進行協調。下列是可以在後續作業上執行的作業：

- 一律支援未與母項作業發生衝突的生命週期作業。這些是子作業或後續作業的要求、取消要求、完成、建立及啟動等這類作業。
- 因為後續作業鏈結的行為與呼叫元件或人員（發送者）的單一作業相同，所以後續作業不支援期限前持續時間，但是會在鏈結中最上層作業的期限計時器結束時到期。
- 因為可能需要停止處理後續作業，但是母項作業會繼續進行處理，所以可以暫停及回復後續作業。
- 可以終止後續作業。
- 後續作業可以具有它們自己的呈報，因此上一個作業的擁有者及後續作業的發送者較能控制後續作業的進度。
- 不支援使用「商業程序編排器」API 刪除個別後續作業。

### 範例：後續作業

下圖顯示具有人力作業活動之後續作業的發佈程序。



在文章發佈程序中，「檢閱文章」作業由 John 要求。程序也授權他檢閱及核准文章的法律部分。然而，此文章說明與競爭者產品的協同作業，因此從法律觀點來看十分敏感。他檢閱過文章的資訊部分，並決定將文章轉給法務部門的 Sarah 進行額外檢閱。他建立「法務部檢閱」作業，並加上說明，強調有關法律部分的疑慮。他將文章併入為作業的輸入，然後將它指派給 Sarah。接著，將新的作業啟動為他自己的「檢閱文章」作業的後續作業。他的作業會進入已轉遞狀態，並結束處理作業。此程序會等待已呼叫「檢閱文章」作業的回應。

Sarah 要求她的「法務部檢閱」後續作業，並開始檢閱法律部分。她寫下一些意見，並完成她的作業。後續作業的輸出訊息會傳遞至商業程序。文章發佈程序會繼續處理與「檢閱文章」作業相關聯的輸出，但是它實際是來自「法務部檢閱」後續作業。因為「檢閱文章」作業是行內人力作業，所以刪除商業程序實例時也會一併刪除它與「法務部檢閱」作業。

---

## 呈報

呈報是人力作業未在指定的時間量動作時自動引發的警示。例如，如果作業未要求，或者未在定義的時間限制內完成。您可以為作業指定一個以上的呈報。這些呈報可以平行啟動，或者作為呈報鏈結啟動。



定義作業範本的呈報範本，或在執行時期使用特定作業來定義它們，就可以定義任何作業的呈報。

呈報會在特定作業狀態時啟動，而且只有在呈報的時間限制到期但尚未達到預期的作業狀態 (監督狀態) 時才會呈報。呈報逾時的時間限制是指定為字串，而且是由指定給作業的行事曆進行解譯。您可以指定多個啟動狀態相同的呈報 (或呈報鏈結)。

您可以定義在作業達到下列作業狀態時啟動的呈報：

**備妥** 若為處於備妥狀態的作業，則可以定義下列狀況的呈報：

- 使用預期的作業狀態 (已要求) 未及時要求作業時呈報。
- 使用預期的作業狀態 (結束) 未及時完成作業時呈報。

**已要求** 若為處於已要求狀態的待辦作業或協同作業，則可以定義下列狀況的呈報：

- 使用預期的作業狀態 (結束) 未及時完成作業時呈報。
- 使用預期的作業狀態 (子作業已完成) 未及時完成此作業的子作業時呈報。或者，您可以使用等待子作業啟動狀態作為預期狀態，以便您可以追蹤子作業的進度。

**等待子作業**

在待辦或協同作業的等待子作業子狀態中，您可以在使用預期作業狀態 (子作業已完成) 未及時完成其子作業時呈報作業。

**執行中** 在呼叫作業的執行中狀態中，可以在使用預期的作業狀態 (已結束) 未及時傳回所呼叫服務時呈報。

您可以定義重複呈報。這些呈報會在每個逾時時檢查相同的預期作業狀態，並在達到預期的作業狀態之前執行定義的呈報動作。

引發呈報時，受呈報影響的人員 (呈報接收者) 會接收到工作項目。而根據呈報的定義，呈報接收者可能也會接收到電子郵件，通知他們已呈報作業。要通知的使用者清單是透過人員查詢進行定義。此查詢必須解析為一組使用者 ID，以建立工作項目。

您可以定義呈報，以使用 `increasePriority` 內容來增加所呈報作業的優先順序。只有針對第一個疊代或呈報的每個疊代，才可以自動增加此優先順序。

## 呈報生命週期

呈報在其生命週期會經過下列狀態：

- 建立之後，在作業達到啟動狀態之前，呈報都保持為非作用中。
- 作業達到呈報的啟動狀態時，呈報會進入等待中狀態。計時器會啟動，而呈報會等到計時器逾時。
- 發生逾時時，會檢查受監督作業的 `atLeastExpectedState` 內容。如果作業已達到或已通過此狀態，則呈報狀態會進入不必要的狀態。如果尚未達到預期的狀態，則呈報會進入已呈報狀態，並呼叫已建立模型的呈報動作。

建立呈報之後，就無法再修改呈報。呈報動作可以反覆地執行。重複間隔是透過呈報的 `autoRepeatDuration` 內容所定義。

## 鏈結的呈報

作業達到鏈結中第一個呈報的啟動狀態時，會啟動呈報鏈結。鏈結中所有呈報的啟動狀態都必須相同。在鏈結中，一次只會啟動一個呈報，但重複呈報保持為作用中，因此不受此限。定義為順序的呈報會循序進行處理：提出第一個呈報時，會啟動鏈結中的下一個呈報，以此類推。

所鏈結呈報的等待持續時間是根據前一個呈報的逾時計算而來，而不是作業達到呈報啟動狀態的時間。因此，如果鏈結中第一個呈報的等待持續時間是兩個小時，而鏈結中第二個呈報的等待持續時間是三個小時，則第一個逾時是發生在作業達到啟動狀態之後的兩個小時，接著第二個逾時是發生在三個小時之後，也就是作業達到啟動狀態之後的五個小時。此行為確保鏈結中較後面的呈報不會在它的上一個呈現之前逾時。

## 呈報的動態持續時間

針對部分呈報，您可能會想要在執行時期動態設定呈報期間。作法是在定義呈報時指定取代表示式，而不要指定固定值。持續時間變數必須用百分比符號 (%) 括住。

此變數可以是下列任一項：

- 作業變數 (例如 `%htm:input.myEscalationDurationValue%`)
- 自訂內容 (例如 `%htm:task.property.myEscalationDurationValue%`)
- 程序變數 (例如 `%wf:variable.myVariable\myPart\myEscalationDurationValue%`)

您必須確定在評估呈報時，可以取得所存取的環境定義資料。

下表顯示呈報持續時間的評估時機：

下列項目的持續時間	評估時機	必須在作業達到下列狀態之前設定環境定義日期：
呈報	作業達到呈報的啟動狀態	呈報的作業啟動狀態
呈報重複	引發呈報	已呈報

### 相關概念

第 69 頁的『人員目錄』

人員目錄會儲存用於人員解析的使用者資訊。

### 相關工作

第 471 頁的『建立通知事件處理程式』

通知事件是在呈報人力作業時產生。「商業程序編排器」提供用來處理呈報的功能 (例如建立呈報工作項目或傳送電子郵件)。您可以建立通知事件處理程式，以自訂呈報的處理方式。

## 呈報的電子郵件通知

提出呈報時，受呈報影響的人員會接收到工作項目。他們也會接收到一封有關作業已呈報的電子郵件。這封電子郵件會套用特定規則。

每個呈報都可以有不同的電子郵件。例如，您可以個人化標準呈報電子郵件，讓它符合組織的標準。若要個人化電子郵件，請在人力作業編輯器中編輯呈報明細。



電子郵件主旨行及內文文字都可以包含取代表示式，以提高電子郵件與它所參照的作業兩者之間的關聯性 (例如，包括作業名稱)。這些表示式必須在傳送電子郵件之前設定，否則，電子郵件收件者會在他們的電子郵件中看到 *%variable name%*。您可以使用任何作業及呈報表示式。

下列 HTML Snippet 顯示包含取代表示式的範例電子郵件：

```
<html>
<head>
</head>
<body lang="EN-US"><div>
<p>The task '<span style="font-size:14.0pt">%htm:task.displayName%/span></b>'
with id '<span style="font-size:14.0pt">%htm:task.instanceID%/span></b>
'&nbsp;has been escalated because the </p>
<p>expected state '<span style="font-size:14.0pt">%htm:escalation.expectedTaskState%/span>
</b>'
&nbsp;has not been reached in time.
</p>
<p>&nbsp;</p>
<p>The task has the following description: </p>
<p><span style="font-size:14.0pt;color:red">%htm:task.description%/span></p>
<p>&nbsp;</p>
<p><span style="font-size:14.0pt;color:green">The name of the Escalation is: %htm:escalation.displayName%
and the escalation description reads: %htm:escalation.description%/span></p>
<p>&nbsp;</p>
<p><a href="%htm:task.URLPrefix?id=%htm:task.instanceID%">Task details</a></p>
<p><a href="%htm:escalation.URLPrefix?id=%htm:escalation.instanceID%">Escalation details</a></p>
</div>
</body>
</html>
```

---

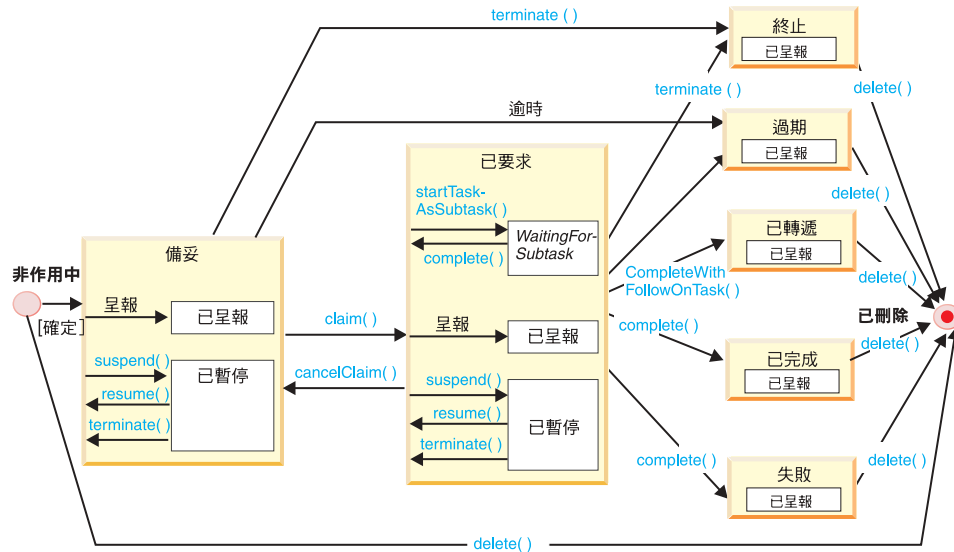
## 人力作業的生命週期

人員與 Web 服務或商業程序互動時，人力作業可以支援人員。作業生命期限期間進行的互動，視作業是待辦作業、協同作業、呼叫作業或管理作業而定。特定互動只有在特定作業狀態下才可以進行，因而，這些互動會對應地影響人力作業的狀態。

### 待辦及協同作業

如果人員將工作當成商業程序的一部分來執行 (行內作業)，或實作所有人都可以使用的 Web 服務 (獨立式作業)，則待辦作業及協同作業可以支援人員。待辦作業及協同作業的差異在於它們的啟動方式。待辦作業是由用戶端應用程式或呼叫元件自動建立。協同作業則是由人員建立及啟動。

下列圖解顯示在待辦作業及協同作業的生命週期期間，可以發生的狀態轉移：



作業在建立之後會進入非作用中狀態。在此狀態中，您可以更新作業內容或設定自訂內容，但是不可以要求作業。若要處理待辦作業或協同作業，則必須啟動該作業。

作業在啟動之後會進入備妥狀態。在此狀態中，作業會等待其中一位可能的擁有者來要求它，並執行與此作業相關聯的工作。在此狀態中，會發生下列異常事件：

- 因為未及時要求作業，所以可以呈報作業。該作業會進入已呈報子狀態，而且在作業生命週期的剩餘時間都會停留在此子狀態。
- 可以手動暫停作業。作業會進入已暫停子狀態。在此狀態中，會封鎖作業上的大部分動作。該作業可以手動回復，也可以透過設定具有暫停動作的計時器自動回復。
- 作業可以到期。此狀態變更會結束作業。
- 可以使用終止動作手動終止作業。此狀態變更會結束作業。

在一般作業流程中，其中一位可能的擁有者會要求作業，進而成為擁有者。作業會進入已要求狀態，而擁有者及編輯者可以處理它。作業處於已要求狀態時，作業擁有者可以採取下列動作：

- 如果它們需要工作的支援，則可以使用子作業來委任一部分的工作。這些子作業可以是協同作業或呼叫作業。母項作業接著會進入等待子作業子狀態，而且在它的所有子作業在達到結束狀態之前都會維持此狀態。您可以在母項作業等待子作業時暫停母項作業，但是母項作業會無法完成，而且無法取消要求。
- 如果想要委任某個人完成工作，則可以建立協同作業，並完成具有後續作業的作業。母項作業會進入已轉遞結束狀態。
- 如果想要委任整個作業責任，則可以將擁有者工作項目轉移給另一個可能的擁有者或管理者。
- 如果想要放棄作業的所有權，則可以取消作業的要求。作業會再次進入備妥狀態，而其中一位可能的擁有者可以要求該作業。

在已要求狀態中，會發生下列異常事件：

- 因為作業未及時完成，或等待子作業的時間太長，所以可以呈報作業。該作業會進入已呈報子狀態，而且在作業生命週期的剩餘時間都會停留在此狀態。

- 可以手動暫停作業。作業會進入已暫停子狀態。在此狀態中，會封鎖作業上的大部分動作。該作業可以手動回復，也可以透過設定具有暫停動作的計時器自動回復。或者，在計時器到期時，會取消作業上的要求，而且會再次進入備妥狀態。
- 作業可以到期。此為結束作業的狀態變更。
- 可以使用終止動作手動終止作業。此為結束作業的狀態變更。

擁有者完成作業的處理時，即完成作業。然後，如果作業順利完成，則作業會進入已完成狀態，如果發生錯誤，則會進入失敗狀態。

失敗、終止、已完成及過期狀態都是無法執行工作的結束狀態。如果作業範本指定自動刪除，則會立即刪除作業，或是在刪除計時器到期之後才會刪除作業。如果未使用自動刪除，則作業在明確刪除之前都會維持它的結束狀態。刪除母項作業時，也會一併刪除它的子作業。如果作業處於已轉遞結束狀態，則不會套用自動刪除。在此情況下，會刪除母項作業及其後續作業。後續作業達到結束狀態時，就會啟動刪除計時器。

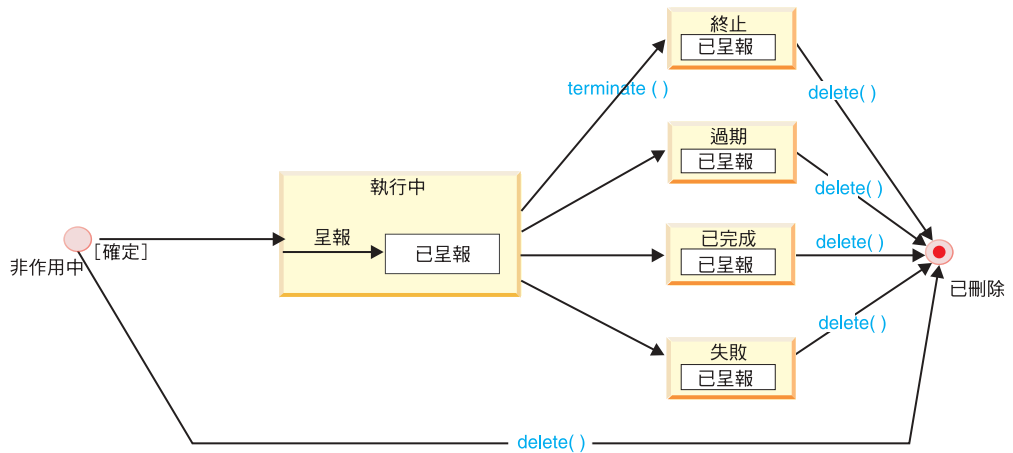
部分其他規則會套用至行內待辦作業。行內作業是商業程序不可缺的部分，因此它們的生命週期是由程序生命週期所控制：

- 商業程序會隱含地建立及啓動作業。
- 作業在商業程序中是以人力作業活動代表。作業及活動都具有相同的狀態，例如，該作業處於備妥狀態時，人力作業活動也會處於備妥狀態。人力作業活動不會反映已轉遞狀態或作業子狀態。
- 如果行內作業具有子作業，則人力作業活動會無法察覺到它們，因此在母項作業完成之前，它都會以已要求狀態進行等待。
- 如果行內作業具有後續作業，則人力作業活動會無法察覺到它們，因此在後續作業完成之前，它都會以已要求狀態進行等待。
- 行內待辦作業沒有期限前的持續時間，因此無法手動終止。期限及終止都是由人力作業活動或商業程序所控制。
- 作業會與商業程序一起刪除。它們不可以手動刪除，在刪除之前也不會有持續時間。

## 呼叫 (起始) 作業

人員呼叫 Web 服務時，呼叫作業可以支援人員。建立及啓動呼叫作業的人員會成為作業發送者。作業在啓動時會自動呼叫服務，並等待其結果。取得服務結果時，呼叫作業會儲存該結果，而且只要作業存在，發送者就可以擷取該結果。

下列圖解顯示在呼叫作業的生命週期期間，可以發生的狀態轉移：



建立之後，作業會達到非作用中狀態。在此狀態中，您可以更新作業內容，或設定自訂內容。若要呼叫服務，則必須啓動作業。發送者或其中一位可能的起始者都可以啓動該作業。

作業在啓動之後會進入執行中狀態。在此狀態中，作業會等待所呼叫服務返回。在此狀態中，會發生下列異常事件：

- 如果服務未及時返回，則可以呈報作業。該作業會進入已呈報子狀態，而且在作業生命週期的剩餘時間都會停留在此狀態。
- 作業可以到期。此為結束作業的狀態變更。
- 可以使用終止動作手動終止作業。此為結束作業的狀態變更。

一般作業流程是指服務返回，並傳回輸出或錯誤訊息。然後，如果傳回輸出訊息，則作業會進入已完成狀態，如果傳回錯誤訊息，則會進入失敗狀態。在這兩種情況下，作業發送者及起始者都可取得訊息。

失敗、終止、已完成及過期狀態都是結束狀態。如果作業範本指定自動刪除，則會在刪除計時器到期之後刪除作業，或是手動刪除作業。預設並不會自動刪除呼叫作業，因此可以存取所呼叫服務的結果。

部分其他規則會套用至行內呼叫作業。這些作業是商業程序不可缺的部分，因此程序可以控制它們的生命週期：

- 如果使用「商業流程管理程式」API 或 SCA 用戶端啓動商業程序，則商業程序會針對建立程序實例的活動隱含地建立及啓動作業。執行中程序實例也可以使用呼叫作業。在此情況下，它們是由程序所建立，而且與接收或挑選 (接收選擇) 活動或者 on-event 事件處理程式相關聯。
- 作業在商業程序中是以接收或挑選 (接收選擇) 活動，或者 on-event 事件處理常式代表。如果針對活動定義了行內呼叫作業，則也會定義此活動的授權。
- 行內呼叫作業沒有期限前的持續時間，因此無法手動終止。
- 如果作業是由商業程序隱含地啓動，則也會與商業程序一起隱含地刪除。
- 如果作業是由「人力作業管理程式」API 啓動，則不會與程序一起刪除。如果建立的作業模型具有自動化刪除，則會在刪除計時器到期之後刪除作業。也可以手動將它刪除。

## 管理作業

管理作業支援人員管理商業程序及其活動。如果沒有可用的管理作業範本，則只要商業程序需要管理作業，就會在執行時期建立預設管理作業。

下列圖解顯示管理作業可以發生的狀態轉移：



「商業流程管理程式」會隱含地在單一交易中建立及啟動管理作業。因此無法在外部看到非作用中狀態，而作業會直接達到備妥狀態。

已完成狀態是結束狀態。然而，它不會禁止進一步管理動作。

管理作業一律是行內作業，因此它們的生命週期是由商業程序所控制。它們一律會與商業程序一起刪除。

### 相關概念

第 39 頁的『子作業』

如果人員需要將他們的一部分指派工作委任給其他人員，但想要控制整體結果，則子作業可以支援人員。它們也可以用來呼叫支援服務，以協助人員完成所使用的作業。

第 36 頁的『獨立式及行內作業』

「服務導向架構 (SOA)」型樣建議具有一組鬆散耦合元件的軟體解決方案的實現。遵循 SOA 型樣的人力作業稱為獨立式作業，而定義為商業程序一部分的人力作業稱為行內作業。

---

## 呼叫作業的範例情況

此處說明可以用來呼叫作業的各種方式。

### 使用人力作業管理程式 API 呼叫作業元件

使用「人力作業管理程式」API 可以實例化作業。「人力作業管理程式」API 用戶端使用 API 來建立、啟動、查詢及操作作業實例。若為作業呼叫，API 會提供方法來建立及啟動下列作業類型：

- 獨立式及行內呼叫作業
- 獨立式待辦作業
- 協同作業

因為管理作業是在商業程序環境定義內呼叫，所以不可以使用此 API 呼叫管理作業。

API 支援作業的下列互動樣式：

- 同步呼叫作業及關聯服務

此互動樣式使用 `callTask` 方法。若為單向作業，則呼叫會在觸發作業及服務元件的執行之後返回。若為要求回應作業，則呼叫會等到服務及作業完成，並傳回呼叫結果。

此互動樣式只可以套用至呼叫作業。

- 非同步呼叫作業及關聯服務

此互動樣式使用 `startTask` 方法。若為單向及要求回應作業，則呼叫會在觸發作業及服務元件的執行之後返回。此外，若為要求回應作業，則呼叫會以非同步方式傳回結果，而結果會儲存為呼叫作業環境定義中的輸出或錯誤訊息。呼叫 API 用戶端必須使用 API 方法，以程式設計方式來擷取結果。您也可以使用回覆處理程式，確保在回應變成可用時，立即將非同步回應傳回至用戶端。

此互動樣式可以套用至待辦、協同作業及呼叫作業。

提供「人力作業管理程式」API 作為 Enterprise JavaBeans (EJB) 實作及 Web 服務實作。所有實作的 API 方法都類似，差別在於它們的功能範圍。

如需這些 API 方法的相關資訊，請參閱 `com.ibm.task.api` 套件中 `HumanTaskManager` 介面的 Javadoc。

## 呼叫待辦作業作為 SCA 服務元件

獨立式待辦作業代表 SCA 用戶端可以非同步呼叫的 SCA 服務元件。SCA 提供的機制可用來連接 SCA 用戶端及獨立式待辦作業。這包括定義下列各項的 SCA 方法：

- 連線，連接 SCA 用戶端 (參照) 及代表待辦作業的元件介面
- 控制互動樣式、交易行為及互動可靠性這類部分的元件參照及介面的 SCA 限定元設定。

此外，實作為商業程序的 SCA 用戶端可以呼叫獨立式待辦作業。在此情況下，必須考量 SCA 及程序層次上的連線。在 SCA 層次上進行檢視時，SCA 用戶端參照會連接至 SCA 服務的介面。而在程序層次上進行檢視時，呼叫活動的夥伴鏈結會連接至待辦作業。

## 呼叫行內待辦作業

待辦作業可以指定於長時間執行商業程序的人力作業活動環境定義內。在此情況下，作業不會呈現在 SCA 層次上，而是屬於代表商業程序的 SCA 元件的一部分。此作業可作為人力作業活動的服務提供者。在程序導覽期間只要到達此活動，就會以非同步方式呼叫待辦作業。

## 透過呼叫作業呼叫 SCA 服務

獨立式呼叫作業可作為關聯 SCA 服務的存取元件。與服務的關聯定義於 SCA 層次上：作業代表連接至 SCA 服務元件的 SCA 用戶端。呼叫作業的呼叫同時包含「人力作業管理程式」及 SCA 層次。呼叫作業本身可由「人力作業管理程式」API 以同步或非同步方式呼叫。然後，作業 (SCA 用戶端) 會以同步方式 (如果以同步方式呼叫作業) 或非同步方式 (如果以非同步方式呼叫作業) 呼叫相關聯的 SCA 服務元件。

建立作業與服務之間的關聯模型是在 SCA 層次上完成。因此，SCA 提供的概念及機制可用於連接獨立式呼叫作業及 SCA 服務元件。這包括定義下列各項的 SCA 方法：



- 連線，連接 SCA 用戶端參照及服務元件介面
- 控制互動樣式、交易行為及互動可靠性這類部分的元件參照及介面的 SCA 限定元設定

此外，獨立式呼叫作業可以連接至商業程序所實作的 SCA 元件。

## 透過行內呼叫作業呼叫商業程序

行內呼叫作業可以指定於商業程序的接收或挑選活動環境定義或事件處理程式中。作業不會呈現在 SCA 層次上，而是屬於代表商業程序的 SCA 元件的一部分。但是，作業會作為商業程序的用戶端。當「人力作業管理程式」API 呼叫作業時，作業就會對應地以「人力作業管理程式」API 呼叫它的相同方式呼叫商業程序。

## 針對管理商業程序及活動建立的管理作業

管理作業支援人員管理商業程序及其活動。「商業流程管理程式」會為允許管理的每個商業程序或活動類型建立及啟動管理作業。如果針對程序或活動指定了管理作業範本，則會使用此範本。如果無法使用範本，則只要商業程序需要管理作業，就會建立預設管理作業。

### 相關概念

『影響獨立式呼叫作業及其服務元件行為的因素』

您可以使用獨立式呼叫作業，來執行與作業的 SCA 元件相關聯的「服務元件架構 (SCA)」服務元件。將作業元件的參照連接至關聯服務元件的介面，就可以在 SCA 層次建立呼叫作業與服務元件的關聯模型。許多因素都會影響呼叫作業及其關聯服務元件的行為。

第 52 頁的『範例情況：支援非同步服務呼叫的獨立式呼叫作業』

此範例情況只考量作業及服務的非同步呼叫，並說明「服務元件架構 (SCA)」設定，以及此呼叫類型的預期交易式和錯誤行為。

第 54 頁的『範例情況：支援非同步及同步服務呼叫的獨立式呼叫作業』

此範例情況考量作業及其相關聯服務的非同步及同步呼叫，並說明「服務元件架構 (SCA)」設定，以及這些呼叫類型的預期交易式和錯誤行為。

### 相關工作

第 469 頁的『建立 API 事件處理程式』

在 API 方法操作人力作業時會發生 API 事件。請使用 API 事件處理程式外掛程式服務提供者介面 (SPI) 來建立外掛程式，以處理 API 或具有相等 API 事件的內部事件所傳送的作業事件。

## 影響獨立式呼叫作業及其服務元件行為的因素

您可以使用獨立式呼叫作業，來執行與作業的 SCA 元件相關聯的「服務元件架構 (SCA)」服務元件。將作業元件的參照連接至關聯服務元件的介面，就可以在 SCA 層次建立呼叫作業與服務元件的關聯模型。許多因素都會影響呼叫作業及其關聯服務元件的行為。

### WSDL 作業類型

SCA 參照及 SCA 介面與包含一或多個作業的 WSDL 埠類型相關聯。每個作業都可以是單向或要求回應作業：

- 單向作業表示完成呼叫作業不知道的服務執行。作業服務執行結束時，會成功呼叫相關聯的服務。

- 要求回應作業表示完成呼叫作業知道的服務執行。呼叫作業可以使用服務執行結果時，作業執行會結束。

### API 呼叫方法

「人力作業管理程式」API 支援作業的下列互動樣式：

- 使用 `callTask` 方法同步呼叫作業及關聯服務
- 使用 `startTask` 方法非同步呼叫作業及關聯服務

### 服務元件的執行持續時間

設定的執行持續時間值必須考量預期因系統上的其他工作量而產生的額外負荷。執行持續時間在考量時也必須參考針對管理「商業程序編排器」的伺服器所設定的交易逾時值。請在決定讓含有要求回應介面的服務元件可用於同步呼叫之前，比較這些值。在這類情況下，服務元件的執行時間必須低於設定給伺服器的交易逾時值。

### SCA 限定元設定

只有特定的 SCA 限定元組合適用於作業元件參照及服務元件介面。

#### 相關概念

第 49 頁的『呼叫作業的範例情況』

此處說明可以用來呼叫作業的各種方式。

## 範例情況：支援非同步服務呼叫的獨立式呼叫作業

此範例情況只考量作業及服務的非同步呼叫，並說明「服務元件架構 (SCA)」設定，以及此呼叫類型的預期交易式和錯誤行為。

此範例情況適用於「人力作業管理程式」API 用戶端 (例如，「商業程序編排器瀏覽器」)，這些用戶端只使用非同步呼叫。它不需要在建立作業模型時，評量與作業相關聯的服務的執行持續時間。

### 作業元件設定

作業元件可以採用下列設定。如果使用 WebSphere Integration Developer 定義作業元件，則會自動產生有效的屬性類型值。

限定元類型：屬性類型	值
參照屬性：Multiplicity	1:1 (必要)
參照限定元：DeliverAsyncAt	commit (必要)
實作限定元*：Transaction	global (必要)
參照限定元**：SuspendTransaction	不適用
實作限定元***：ActivitySession	true (必要)
參照限定元***：SuspendActivitySession	false (預設值)
參照限定元：Reliability	assured (必要)
參照限定元：RequestExpiration	any
參照限定元：ResponseExpiration	any
<b>註：</b> <ul style="list-style-type: none"> <li>• *：如果使用交易設定，則使用 <code>global</code>，如果使用活動階段作業設定，則使用 <code>local</code>。</li> <li>• **：如果交易設為 <code>global</code>，則只會使用交易設定</li> <li>• ***：如果交易設為 <code>local</code>，則只會使用活動階段作業的設定</li> </ul>	

## 服務元件設定

服務元件可以採用下列設定。如果使用 WebSphere Integration Developer 定義作業元件，則會自動產生有效的屬性類型值。

限定元類型：屬性類型	值
介面屬性：PreferredInteractionStyle	Ignored
實作限定元*：Transaction	local (預設值) global
介面限定元**：JoinTransaction	false (預設值)true
實作限定元***：ActivitySession	any (預設值)
介面限定元***：JoinActivitySession	false (預設值)
<b>註：</b> <ul style="list-style-type: none"> <li>*：如果使用交易設定，則使用 global，如果使用活動階段作業設定，則使用 local。</li> <li>**：如果交易設為 global，則只會使用交易設定</li> <li>***：如果交易設為 local，則只會使用活動階段作業的設定</li> </ul>	

下列清單提供服務 **Transaction** 及 **JoinTransaction** 限定元的有效設定組合：

- **Transaction** 限定元設為 local，而 **JoinTransaction** 設為 false。使用這些設定，作業及服務呼叫會在不同交易中執行。
- **Transaction** 限定元設為 global，而 **JoinTransaction** 設為 false。使用這些設定，作業及服務呼叫會在不同交易中執行。
- **Transaction** 限定元設為 global，而 **JoinTransaction** 設為 true。使用這些設定，作業及服務呼叫會在相同交易中執行。

## 交易式及錯誤行為

在此非同步呼叫範例情況中，startTask 方法只適用於 API 呼叫。作業及服務呼叫可在不同交易中進行。發生服務實作未處理的執行時期異常狀況時，適用下列各項。此範例情況具有下列交易式行為及異常狀況處理。

作業類型	發生 SCA 執行時期異常狀況時	作業及服務的行為
單向作業	在服務呼叫期間，但在啟動服務執行之前	作業接收到 SCA 執行時期異常狀況。「人力作業管理程式」API 方法擲出 CoreOTaskServiceRuntimeExceptionReceivedException 異常狀況。會回復作業交易，而作業會停留在非作用中狀態。
單向作業	在服務執行期間	未通知呼叫作業。作業會移至已完成狀態，並產生可以使用失敗事件管理程式處理的失敗事件。
要求回應作業	在服務呼叫期間，但在啟動服務執行之前	作業接收到 SCA 執行時期異常狀況。「人力作業管理程式」API 方法擲出 CoreOTaskServiceRuntimeExceptionReceivedException 異常狀況。會回復作業交易，而作業會停留在非作用中狀態。

作業類型	發生 SCA 執行時期異常狀況時	作業及服務的行為
要求回應作業	在服務執行期間	通知作業發生 SCA 執行時期異常狀況，並將該異常狀況儲存在資料庫的作業環境定義中。如果回覆處理程式可用，則會使用它來通知用戶端。作業會進入已失敗狀態。

作業定義可以包括一個以上在執行期間由服務元件所擲出的錯誤訊息。

作業元件會收到錯誤訊息通知，如下所示：

- 將錯誤訊息儲存在作業環境定義的資料庫中
- 作業會進入失敗狀態
- 如果作業是以同步方式呼叫，且指定了回覆處理程式，則會呼叫回覆處理程式，將發生的錯誤傳回給用戶端
- 如果作業是以非同步方式呼叫，則會將錯誤訊息傳回給用戶端作為 `FaultReplyException` 異常狀況

錯誤處理並不會影響交易式行為，也不會回復交易。

#### 相關概念

第 49 頁的『呼叫作業的範例情況』

此處說明可以用來呼叫作業的各種方式。

### 範例情況：支援非同步及同步服務呼叫的獨立式呼叫作業

此範例情況考量作業及其相關聯服務的非同步及同步呼叫，並說明「服務元件架構 (SCA)」設定，以及這些呼叫類型的預期交易式和錯誤行為。

在此範例情況中，「人力作業管理程式」用戶端同時使用非同步及同步呼叫。這表示您已評量過服務執行時間是否少於預期的伺服器交易逾時值。一般而言，執行持續時間必須遠少於伺服器交易逾時值。

### 作業元件設定

作業元件可以採用下列設定。如果使用 WebSphere Integration Developer 定義作業元件，則會自動產生有效的屬性類型值。

限定元類型：屬性類型	值
參照屬性：Multiplicity	1:1 (必要)
參照限定元：DeliverAsyncAt	commit (必要)
實作限定元 <sup>*</sup> ：Transaction	global (必要)
參照限定元 <sup>**</sup> ：SuspendTransaction	不適用
實作限定元 <sup>***</sup> ：ActivitySession	true (必要)
參照限定元 <sup>***</sup> ：SuspendActivitySession	false (預設值)
參照限定元：Reliability	assured (必要)
參照限定元：RequestExpiration	any
參照限定元：ResponseExpiration	any

限定元類型：屬性類型	值
<b>註:</b> <ul style="list-style-type: none"> <li>*：如果使用交易設定，則使用 global，如果使用活動階段作業設定，則使用 local。</li> <li>**：如果交易設為 global，則只會使用交易設定</li> <li>***：如果交易設為 local，則只會使用活動階段作業的設定</li> </ul>	

## 服務元件設定

服務元件可以採用下列設定。如果使用 WebSphere Integration Developer 定義作業元件，則會自動產生有效的屬性類型值。

限定元類型：屬性類型	值
介面屬性：PreferredInteractionStyle	Ignored
實作限定元*：Transaction	local (預設值) global
介面限定元**：JoinTransaction	false (預設值)true
實作限定元***：ActivitySession	any (預設值)
介面限定元***：JoinActivitySession	false (預設值)
<b>註:</b> <ul style="list-style-type: none"> <li>*：如果使用交易設定，則使用 global，如果使用活動階段作業設定，則使用 local。</li> <li>**：如果交易設為 global，則只會使用交易設定</li> <li>***：如果交易設為 local，則只會使用活動階段作業的設定</li> </ul>	

下列清單提供服務 **Transaction** 及 **JoinTransaction** 限定元的有效設定組合：

- **Transaction** 限定元設為 local，而 **JoinTransaction** 設為 false。使用這些設定，作業及服務呼叫會在不同交易中執行。
- **Transaction** 限定元設為 global，而 **JoinTransaction** 設為 false。使用這些設定，作業及服務呼叫會在不同交易中執行。
- **Transaction** 限定元設為 global，而 **JoinTransaction** 設為 true。使用這些設定，作業及服務呼叫會在相同交易中執行。

## 交易式及錯誤行為

此範例情況具有下列交易式行為及異常狀況處理。

API 呼叫樣式	作業類型	發生 SCA 執行時期異常狀況時	作業及服務的行為
callTask	單向作業	在服務呼叫期間，但在啟動服務執行之前	作業接收到 SCA 執行時期異常狀況。「人力作業管理程式」API 方法擲出 CoreOTaskServiceRuntimeExceptionReceivedException 異常狀況。會回復作業交易，而作業會停留在非作用中狀態。
callTask	單向作業	在服務執行期間	作業接收到 SCA 執行時期異常狀況。「人力作業管理程式」API 方法擲出 CoreOTaskServiceRuntimeExceptionReceivedException 異常狀況。會回復作業交易，而作業會停留在非作用中狀態。

API 呼叫樣式	作業類型	發生 SCA 執行時期異常狀況時	作業及服務的行為
callTask	要求回應作業	在服務呼叫期間，但在啟動服務執行之前	作業接收到 SCA 執行時期異常狀況。「人力作業管理程式」API 方法擲出 CoreOTaskServiceRuntimeExceptionReceivedException 異常狀況。會回復作業交易，而作業會停留在非作用中狀態。
callTask	要求回應作業	在服務執行期間	作業接收到 SCA 執行時期異常狀況。「人力作業管理程式」API 方法擲出 CoreOTaskServiceRuntimeExceptionReceivedException 異常狀況。會回復作業交易，而作業會停留在非作用中狀態。
startTask	單向作業	在服務呼叫期間，但在啟動服務執行之前	作業接收到 SCA 執行時期異常狀況。「人力作業管理程式」API 方法擲出 CoreOTaskServiceRuntimeExceptionReceivedException 異常狀況。會回復作業交易，而作業會停留在非作用中狀態。
startTask	單向作業	在服務執行期間	未通知呼叫作業。作業會移至已完成狀態，並產生可以使用失敗事件管理程式處理的失敗事件。
startTask	要求回應作業	在服務呼叫期間，但在啟動服務執行之前	作業接收到 SCA 執行時期異常狀況。「人力作業管理程式」API 方法擲出 CoreOTaskServiceRuntimeExceptionReceivedException 異常狀況。會回復作業交易，而作業會停留在非作用中狀態。
startTask	要求回應作業	在服務執行期間	通知作業發生 SCA 執行時期異常狀況，並將該異常狀況儲存在資料庫的作業環境定義中。如果回覆處理程式可用，則會使用它來通知用戶端。作業會移至失敗狀態。

作業定義可以包括一或多個在執行期間由服務元件所擲出的錯誤訊息。

作業元件會收到錯誤訊息通知，如下所示：

- 將錯誤訊息儲存在作業環境定義的資料庫中
- 作業會進入失敗狀態
- 如果作業是以非同步方式呼叫，且指定了回覆處理程式，則會呼叫回覆處理程式，將發生的錯誤傳回給用戶端
- 如果作業是以同步方式呼叫，則會將錯誤訊息傳回給用戶端作為 FaultReplyException 異常狀況

錯誤處理並不會影響交易式行為，也不會回復交易。

#### 相關概念

第 49 頁的『呼叫作業的範例情況』

此處說明可以用來呼叫作業的各種方式。

## 授權及人員指派

授權是一種機制，特定人員啟用這種機制就可以對作業範本、作業實例及呈報執行選取的動作。授權角色是用來定義可用於特定角色的動作集。使用 J2EE 機制可以將人員指派給系統層次角色，而使用人員指派準則可以將人員指派給作業實例角色。



## 人力作業的授權角色

您可以針對人力作業執行的動作，視授權角色而定。此角色可以是系統層次 J2EE 角色或實例型角色。

系統層次 Java 2 Platform Enterprise Edition (J2EE) 角色是在配置「人力作業管理程式」時設定。這些角色隱含的權限層次適用於所有作業及呈報。實例型角色則適用於個別作業及呈報實例，或用來建立作業或呈報實例的範本。角色型授權則需要啓用應用程式伺服器及管理應用程式安全。

### J2EE 角色

支援的 J2EE 角色如下：

- **TaskSystemAdministrator**。指派給此角色的使用者具有所有專用權。此角色也稱為人力作業的系統管理者。
- **TaskSystemMonitor**。指派給此角色的使用者可以檢視所有作業物件的內容。此角色也稱為人力作業的系統監視者。

您可以使用管理主控台來變更這些角色的使用者及群組指派。

### 實例型角色

作業實例或呈報實例不會直接指派給人員，而是與指派給人員的預先定義角色相關聯。指派給實例型角色的所有人都可以執行該角色的動作。使用者與實例型角色的關聯是根據人員指派所決定，或是由作業動作所產生。

人員指派會根據人員目錄中儲存的使用者及使用者群組資訊，於執行時期將人員指派給下列角色：可能的建立者、可能的起始者、可能的擁有者、讀者、編輯者、管理者及呈報接收者。下列角色只會與一位使用者相關聯，而且是作為作業動作結果進行指派：發送者、起始者、擁有者。

這些角色已授權執行下列動作：

角色	授權的動作
可能的建立者	此角色的成員可以建立作業的實例。如果未定義作業範本的可能實例建立者，則會將所有使用者都視為此角色的成員。
發送者	在啓動作業之前，具有此角色的人員會擁有管理權限。當作業啓動時，發送者具有讀者的權限，可以執行部分管理動作，例如暫停及回復作業，以及轉移工作項目。
可能的起始者	此角色的成員可以啓動現有的作業實例。如果沒有指定可能的起始者，則發送者會變成可能的起始者。若為沒有可能的起始者的行內作業，則預設值是每個人。
起始者	具有此角色的人員擁有讀者權限，而且可以執行一些管理動作 (例如轉移工作項目)。
可能的擁有者	此角色的成員可以要求作業。如果未定義作業範本的可能擁有者，則會將所有使用者都視為此角色的成員。如果此角色的人員解析失敗，則管理者會指派為可能的擁有者。
擁有者	具有此角色的人員可以使用及完成作業。
讀者	此角色的成員可以檢視所有作業物件的內容，但無法處理它們。
編輯者	此角色的成員可以處理作業的內容，但無法要求或完成它。

角色	授權的動作
管理者	此角色的成員可以管理作業、作業範本及呈報。
呈報接收者	此角色的成員具有呈報及已呈報作業的讀者權限。

### 相關概念

第 28 頁的『商業程序的授權角色』

角色是指具有相同授權層次的一組人員。您可以針對商業程序執行的動作，視您的授權角色而定。此角色可以是 J2EE 角色或實例型角色。

第 39 頁的『子作業』

如果人員需要將他們的一部分指派工作委任給其他人員，但想要控制整體結果，則子作業可以支援人員。它們也可以用來呼叫支援服務，以協助人員完成所使用的作業。

第 75 頁的『預設人員指派』

如果未定義某些作業角色的人員指派準則，或者人員解析失敗或未傳回結果，則會執行預設人員指派。行內作業及獨立式作業的預設指派會不同。

### 作業類型的實例型授權角色

建立作業模型時，實例型授權角色會與人力作業及呈報相關聯。作業類型決定特定的授權角色是否可用。

角色	待辦作業	呼叫作業	協同作業	管理作業	備註
可能的實例建立者	X	X	X		受允許建立作業實例的人員
發送者	X	X	X		建立作業的人員
可能的擁有者	X		X		可以要求及使用作業的人員
擁有者	X		X		要求作業的人員
可能的起始者		X			受允許啓動作業的人員
起始者		X			啓動作業的人員
管理者	X	X	X	X <sup>1</sup>	受允許管理作業的人員
編輯者	X		X		受允許編輯作業資料的人員
讀者	X	X	X	X <sup>2</sup>	受允許查看作業資料的人員
呈報接收者	X <sup>3</sup>	X <sup>3</sup>	X <sup>3</sup>	X <sup>3</sup>	接收呈報的人員

#### 附註:

1. 此角色還具有對受管理程序或活動執行管理動作的權限。
2. 此角色還具有對受管理程序或活動執行讀取作業的權限。
3. 此角色獲得對產生自這些作業類型的呈報 (而非作業本身) 執行動作的權限。

### 授權及工作項目

每個作業角色都可以讓使用者對相關聯的作業執行一組確切的動作。而人員的授權是使用工作項目進行管理。工作項目代表所指派人員與作業角色所隱含作業動作的關係。

工作項目具有下列部分：

- 使用者或使用者群組的身分

- 可以對其執行動作的作業的身分
- 與使用者相關聯的作業角色

您可以使用下列其中一種方式來指定與工作項目相關聯的人員：

- 只有一個使用者 ID。這會導致使用者工作項目。
- 只有一個使用者群組 ID。這會導致群組工作項目。
- 針對每位使用者，使用 **Everybody** 人員指派準則。這會導致 **Everybody** 工作項目。

「商業程序編排器」的授權機制確保在具有下列其中一個條件時，使用者可以執行與工作項目相關聯的動作：

- 使用者使用與使用者工作項目的指定使用者 ID 相符的使用者 ID 進行登入
- 登入使用者是群組成員，該組對應於群組工作項目的指定群組 ID
- 工作項目是指派給每個人的工作項目

「人力作業管理程式 API」提供方法來查詢人力作業、呈報及其他物件。當查詢執行時，使用者查看所查詢資料的授權，是透過只傳回使用者擁有其工作項目的資料來確定。您也可以使用 API 來管理實例型授權。建立及刪除工作項目，以及在人員之間轉移工作項目，就可以完成此作業。如需這些 API 方法的相關資訊，請參閱 `com.ibm.task.api` 套件中 `HumanTaskManager` 介面的 Javadoc。

## 人員指派準則

人員指派準則用於作業模型中的建構，用以識別可以指派給實例型授權角色的人員集。在執行時期，人員解析會使用人員指派準則來擷取使用者 ID 及其他使用者資訊 (例如，用於撰寫電子郵件)。以程式設計方式建立作業模型或特定作業時，也可以在執行時期使用人員指派準則。

您可以在 **WebSphere Integration Developer** 中使用人員指派準則定義 (之前稱為人員動詞)，建立作業角色的人員指派模型。定義包含查詢名稱及一組查詢參數。部署作業時，會將指派準則轉換為人員目錄特有的查詢 (例如，虛擬成員管理程式)。作業執行時，這些查詢會擷取指派給角色的那組人員，例如可能的擁有者。

下列範例說明與實作作業角色的人員指派準則定義有關的步驟：

1. 在 **WebSphere Integration Developer** 中，模型產生器會關聯新的作業與人員目錄配置，例如，若為虛擬成員管理程式，則為 `bpe/staff/samplevmmconfiguration`。

此步驟會決定可用於人員指派的人員指派準則。

2. 在 **WebSphere Integration Developer** 中，模型產生器會關聯作業角色與人員指派準則定義。

例如，可能的擁有者角色與人員指派準則**群組成員**相關聯，包括下列參數：

- **GroupName** 設為值 `cn=group1, dc=mycomp, dc=com`
  - **IncludeSubgroups** 設為值 `true`
3. 部署作業時，人員指派服務會建立要使用的人員目錄提供者。它會將人員指派準則轉換為人員目錄提供者的查詢，其會儲存在內部。

根據使用的人員目錄，在建立作業模型時，可使用不同的預先定義人員指派準則子集：

- LDAP 及虛擬成員管理程式人員目錄提供者可支援所有預先定義的定義
- 使用者登錄人員目錄提供者只支援那些根據使用者及群組名稱的定義。而不會為根據管理者或電子郵件屬性的定義提供支援。
- 系統人員目錄提供者僅供測試使用。支援只限於指定寫在程式中的使用者 ID 集，因此不需要存取人員目錄。

## 預先定義的人員指派準則

提供預先定義人員指派準則，用來擷取人員目錄中的不同使用者組。

您可以在 WebSphere Integration Developer 中使用人員指派準則 (之前稱為人員動詞)，以指定人力作業中的人員指派。在建立模型及部署期間，這些準則會轉換為可以在人員目錄上執行的查詢集。這裡列出下列預先定義人員指派準則的參數：

- Department Members
- Everybody
- Group
- Group Members
- Group Members without Named Users
- Group Members without Filtered Users
- Group Search
- Manager of Employee
- Manager of Employee by user ID
- Native Query
- Nobody
- Person Search
- Role Members
- User Records by user ID
- User Records by user ID without Named Users
- Users
- Users by user ID
- Users by user ID without Named Users

指定人員指派準則時，請考慮下列各項：

- 如果要處理大群人員，因為群組人員指派準則可將群組成員視為整體進行處理，所以十分適用。這可讓您輕鬆地將人力作業從某個群組轉移至另一個群組。在人員登入並存取人力作業時，會解析人員的群組成員資格。
- 若要將屬於某個群組的人員個別指派給人力作業，群組成員人員指派準則提供另一種群組指派方式。群組成員人員指派準則會個別建立每個人員的指派。接著，可以將此指派轉移給另一位人員。這可能會進行代理，亦即，另一位人員可以取代缺席的人員。此人員指派準則的變式 (沒有指定使用者的群組成員) 支援區分責任指派型樣。

**註：** 將人員個別指派給群組會在執行時期影響效能，特別是在將多位人員指派給群組時。

- 若要將不是全部屬於相同群組的部分人員指派給人力作業，請考慮使用依使用者 ID 的使用者記錄人員指派準則定義。如果人員指派不是在建立模型期間以靜態方式定義，但包括取代表示式，您也可以使用此定義。取代表示式可以參照自訂內容或人力作業的輸入訊息。依使用者 ID 的使用者人員指派準則定義，與依使用者 ID 的使

用者記錄定義類似。在執行時期，雖然依使用者 ID 的使用者定義的執行效能優於依使用者 ID 的使用者記錄定義，但是提供的功能較少：

- 不會檢查輸入的使用者 ID 是否正確
- 例如，不會擷取所指定使用者 ID 的電子郵件位址，這較不適用於將人員指派給電子郵件呈報。
- 也可以考量 Everybody 人員指派準則定義。這表示會將所有已鑑別的使用者指派給人力作業。因為可能發生組織中所有人員都可以執行某個特定工作的情況，所以此定義特別適用於開發期間，或快速建立應用程式原型時。

## 部門成員

使用此準則，可擷取部門的成員。「輕量型目錄存取通訊協定 (LDAP)」及虛擬成員管理程式人員目錄提供者都支援此準則。

參數	使用	類型	說明
DepartmentName	必要	字串	要擷取的使用者的部門名稱。部門名稱必須對應於下列其中一個值： <ul style="list-style-type: none"> <li>• 若為虛擬成員管理程式，是虛擬成員管理程式群組的唯一名稱</li> <li>• 若為 LDAP，是 LDAP 群組的識別名稱 (DN)</li> </ul>
IncludeNestedDepartments	必要	布林	指定在查詢中是否考量巢狀部門。
AlternativeDepartmentName1	選用	字串	使用者可以屬於的其他部門。
AlternativeDepartmentName2	選用	字串	使用者可以屬於的其他部門。

## 每個人

使用此準則，可將 WebSphere Process Server 鑑別的每位使用者指派給作業角色。此準則沒有參數。

所有人員目錄提供者都支援此準則。

## 群組

使用此準則，可將群組指派給作業角色。此指派會建立每個已指派使用者的群組工作項目，而不是建立使用者工作項目。

所有人員目錄提供者都支援此準則。

參數	使用	類型	說明
GroupId	必要	字串	<p>要擷取的使用者群組名稱。此參數支援取代表示式。群組 ID 必須對應於下列其中一個值：</p> <ul style="list-style-type: none"> <li>• 若為虛擬成員管理程式，是群組項目的唯一名稱</li> <li>• 若為 LDAP，是群組項目的 DN</li> <li>• 若為使用者登錄提供者，則使用的名稱格式是視設定給部署作業的應用程式伺服器的使用者儲存庫而定： <ul style="list-style-type: none"> <li>– 若為本端作業系統，則使用本端作業系統所支援的群組名稱</li> <li>– 若為獨立式自訂登錄，則使用自訂實作所支援的群組名稱</li> <li>– 若為獨立式 LDAP 登錄，則使用群組項目的 DN</li> </ul> </li> </ul>

## 群組成員

使用此準則，可擷取群組的成員。LDAP、虛擬成員管理程式及使用者登錄人員目錄提供者都支援此準則。

參數	使用	類型	說明
GroupName	必要	字串	<p>要擷取的使用者群組名稱。此參數支援取代表示式。群組 ID 必須對應於下列其中一個值：</p> <ul style="list-style-type: none"> <li>• 若為虛擬成員管理程式，是群組項目的唯一名稱</li> <li>• 若為 LDAP，是群組項目的 DN</li> <li>• 若為使用者登錄提供者，則使用的名稱格式是視設定給部署作業的應用程式伺服器的使用者儲存庫而定： <ul style="list-style-type: none"> <li>– 若為本端作業系統，則使用本端作業系統所支援的群組名稱</li> <li>– 若為獨立式自訂登錄，則使用自訂實作所支援的群組名稱</li> <li>– 若為獨立式 LDAP 登錄，則使用群組項目的 DN</li> </ul> </li> </ul>
IncludeSubgroups	必要	布林	指定在查詢中是否考量巢狀子群組。
AlternativeGroupName1	選用	字串	使用者可以屬於的其他群組。
AlternativeGroupName2	選用	字串	使用者可以屬於的其他群組。

## 沒有指定使用者的群組成員

使用此準則，可擷取群組的所有成員，但不含明確指定的使用者。LDAP、虛擬成員管理程式及使用者登錄人員目錄提供者都支援此準則。



參數	使用	類型	說明
GroupName	必要	字串	要擷取的使用者群組名稱。此參數支援取代表示式。群組 ID 必須對應於下列其中一個值： <ul style="list-style-type: none"> <li>• 若為虛擬成員管理程式，是群組項目的唯一名稱</li> <li>• 若為 LDAP，是群組項目的 DN</li> <li>• 若為使用者登錄提供者，則使用的名稱格式是視設定給部署作業的應用程式伺服器的使用者儲存庫而定： <ul style="list-style-type: none"> <li>– 若為本端作業系統，則使用本端作業系統所支援的群組名稱</li> <li>– 若為獨立式自訂登錄，則使用自訂實作所支援的群組名稱</li> <li>– 若為獨立式 LDAP 登錄，則使用群組項目的 DN</li> </ul> </li> </ul>
IncludeSubgroups	必要	布林	指定在查詢中是否考量巢狀子群組。
NamedUsers	必要	字串	要從擷取的群組成員清單中排除的使用者的使用者 ID。此參數支援取代表示式。

### 沒有已過濾使用者的群組成員

使用此準則，可擷取群組的所有成員，但不含搜尋過濾器所定義的一組使用者。LDAP 及虛擬成員管理程式人員目錄提供者都支援此準則。

參數	使用	類型	說明
GroupName	必要	字串	要擷取的使用者群組名稱。此參數支援取代表示式。群組 ID 必須對應於下列其中一個值： <ul style="list-style-type: none"> <li>• 若為虛擬成員管理程式，是群組項目的唯一名稱</li> <li>• 若為 LDAP，是群組項目的 DN</li> </ul>
IncludeSubgroups	必要	布林	指定在查詢中是否考量巢狀子群組。
FilterAttribute	必要	字串	要在搜尋過濾器中使用的屬性名稱。
FilterValue	必要	字串	要在搜尋過濾器中使用的過濾器值。您可以在過濾器中使用萬用字元 (星號 (*))。

### 群組搜尋

使用此準則，可根據屬性相符項來搜尋群組，以及指派群組的成員。LDAP 及虛擬成員管理程式人員目錄提供者都支援此準則。

參數	使用	類型	說明
GroupID	選用	字串	要擷取的使用者群組 ID。
Type	選用	字串	要擷取的使用者的群組類型。
IndustryType	選用	字串	使用者所屬群組的產業類型。
BusinessType	選用	字串	使用者所屬群組的商業類型。
GeographicLocation	選用	字串	使用者所在位置的指示。

參數	使用	類型	說明
Affiliates	選用	字串	使用者的成員組織。
DisplayName	選用	字串	群組的顯示名稱。
Secretary	選用	字串	使用者的秘書。
Assistant	選用	字串	使用者的助理。
Manager	選用	字串	使用者的經理。
BusinessCategory	選用	字串	使用者所屬群組的商業種類。
ParentCompany	選用	字串	使用者的母公司。

若為虛擬成員管理程式，則群組實體的內容會與下列群組搜尋準則參數相等：

- GS\_GroupID : cn
- GS\_DisplayName : displayName
- GS\_BusinessCategory : businessCategory

### 員工經理

使用此準則，可擷取使用該人員姓名的員工經理。LDAP 及虛擬成員管理程式人員目錄提供者都支援此準則。

參數	使用	類型	說明
EmployeeName	必要	字串	已擷取其經理的員工名稱。員工姓名必須對應於下列其中一個值： <ul style="list-style-type: none"> <li>• 若為虛擬成員管理程式，是人員項目的唯一名稱</li> <li>• 若為 LDAP，是人員項目的 DN</li> </ul>

### 依使用者 ID 的員工經理

使用此準則，可擷取使用該人員使用者 ID 的人員經理。LDAP 及虛擬成員管理程式人員目錄提供者都支援此準則。

參數	使用	類型	說明
EmployeeUserID	必要	字串	已擷取其經理的員工的登入使用者 ID。此參數支援取代表示式。

### 原生查詢

使用此準則，可根據目錄特定的參數來定義原生查詢。LDAP 及虛擬成員管理程式人員目錄提供者都支援此準則。

參數	使用	類型	說明
QueryTemplate	必要	字串	要使用的查詢範本。這必須是下列其中一個值：search、user 及 usersOfGroup。

參數	使用	類型	說明
Query	必要	字串	指定查詢。此參數支援取代表示式。查詢的類型視查詢範本而定。 <ul style="list-style-type: none"> <li>搜尋範本：搜尋過濾器 <ul style="list-style-type: none"> <li>若為虛擬成員管理程式，是有效的搜尋表示式</li> <li>若為 LDAP，是有效的 LDAP 過濾器</li> </ul> </li> <li>使用者範本：使用者 DN <ul style="list-style-type: none"> <li>若為虛擬成員管理程式，是使用者項目的唯一名稱</li> <li>若為 LDAP，是使用者項目的 DN</li> </ul> </li> <li>usersOfGroup：群組 DN <ul style="list-style-type: none"> <li>若為虛擬成員管理程式，是群組的唯一名稱</li> <li>若為 LDAP，是群組的 DN</li> </ul> </li> </ul>
AdditionalParameter1	選用	字串	指定查詢。此參數支援取代表示式。參數類型會視查詢範本而定。 <ul style="list-style-type: none"> <li>搜尋範本。用來指定遞迴搜尋是否已完成。支援的值：yes 及 no</li> <li>使用者範本。不支援</li> <li>usersOfGroup。用來指定遞迴搜尋是否已完成。支援的值：yes 及 no</li> </ul>
AdditionalParameter2	選用	字串	使用此準則，可指定基本項目以進行搜尋。 <ul style="list-style-type: none"> <li>若為虛擬成員管理程式，是基本項目的唯一名稱 (例如，dc=mycomp, dc=com)</li> <li>若為 LDAP，是基本項目的 DN</li> </ul>
AdditionalParameter3	選用	字串	使用此準則來指定其他參數。  如果您使用預設對映 XSLT 檔案，則不支援這個參數。
AdditionalParameter4	選用	字串	使用此準則來指定其他參數。  如果您使用預設對映 XSLT 檔案，則不支援這個參數。
AdditionalParameter5	選用	字串	使用此準則來指定其他參數。  如果您使用預設對映 XSLT 檔案，則不支援這個參數。

## 無人

使用此準則，可以拒絕作業角色的使用者存取。此準則只會套用授權繼承及人員解析預設值。此準則沒有參數。

## 人員搜尋

使用此準則，可根據屬性相符項來搜尋人員。LDAP、虛擬成員管理程式及使用者登錄人員目錄提供者都支援此準則。

參數	使用	類型	說明
UserID	選用	字串	要擷取的使用者的使用者 ID。
Profile	選用	字串	要擷取的使用者設定檔。

參數	使用	類型	說明
LastName	選用	字串	要擷取的使用者的姓氏。
FirstName	選用	字串	要擷取的使用者的名字。
MiddleName	選用	字串	要擷取的使用者的中間名。
Email	選用	字串	使用者的電子郵件位址。
Company	選用	字串	使用者所屬的公司。
DisplayName	選用	字串	使用者的顯示名稱。
Secretary	選用	字串	使用者的秘書。
Assistant	選用	字串	使用者的助理。
Manager	選用	字串	使用者的經理。
Department	選用	字串	使用者所屬的部門。
Phone	選用	字串	使用者的電話號碼。
Fax	選用	字串	使用者的傳真號碼。
Gender	選用	字串	使用者的性別。
Timezone	選用	字串	使用者所在位置的時區。
PreferredLanguage	選用	字串	使用者偏好的語言。

若為虛擬成員管理程式，則 PersonAccount 實體的內容會與下列人員搜尋準則參數相等：

- PS\_UserID : uid
- PS\_LastName : sn
- PS\_FirstName : givenName
- PS\_MiddleName : initials
- PS\_Email : mail
- PS\_DisplayName : displayName
- PS\_Secretary : secretary
- PS\_Manager : manager
- PS\_Department : departmentNumber
- PS\_Phone : telephoneNumber
- PS\_PREFERREDLANGUAGE : preferredLanguage

### Role Members

使用此準則，可擷取與角色相關聯的使用者。LDAP 及虛擬成員管理程式人員目錄提供者都支援此準則。

參數	使用	類型	說明
RoleName	必要	字串	要擷取的使用者的角色名稱。
IncludeNestedRoles	必要	布林	指定在查詢中是否考量巢狀角色。
AlternativeRoleName1	選用	字串	使用者的其他角色名稱。
AlternativeRoleName2	選用	字串	使用者的其他角色名稱。

## 依使用者 ID 的使用者記錄

使用此準則，可定義已知其使用者 ID 的使用者查詢。LDAP 及虛擬成員管理程式人員目錄提供者都支援此準則。此準則會傳回這些使用者的使用者 ID、電子郵件資訊及偏好語言環境 (如果已設定)。

參數	使用	類型	說明
UserID	必要	字串	要擷取的使用者的使用者 ID。此參數支援取代表示式。
AlternativeID1	選用	字串	其他使用者 ID。使用這個參數來擷取多位使用者。
AlternativeID2	選用	字串	其他使用者 ID。使用這個參數來擷取多位使用者。

## 排除指定使用者的使用者記錄，依使用者 ID 排列

使用此準則，可定義已知其使用者 ID 的使用者的查詢，但排除明確指定的使用者 ID。LDAP、虛擬成員管理程式及使用者登錄人員目錄提供者都支援此準則。此準則會傳回這些使用者的使用者 ID 及電子郵件資訊。

參數	使用	類型	說明
UserID	必要	字串	要擷取的使用者的使用者 ID。此參數支援取代表示式。
AlternativeID1	選用	字串	其他使用者 ID。使用這個參數來擷取多位使用者。
AlternativeID2	選用	字串	其他使用者 ID。使用這個參數來擷取多位使用者。
NamedUsers	必要	字串	要從使用者 ID 清單排除的使用者 ID。此參數支援取代表示式。

## 使用者

使用此準則，可定義已知其名稱的使用者查詢。所有人員目錄提供者都支援此準則。

參數	使用	類型	說明
Name	必要	字串	要擷取的使用者名稱。 <ul style="list-style-type: none"><li>• 若為虛擬成員管理程式，是人員項目的唯一名稱</li><li>• 若為 LDAP，是人員項目的 DN</li><li>• 若為使用者登錄提供者，則使用的名稱格式是視設定給部署作業的應用程式伺服器的使用者儲存庫而定：<ul style="list-style-type: none"><li>– 若為本端作業系統，則使用要指派的使用者的使用者 ID</li><li>– 若為獨立式自訂登錄，則使用自訂實作所支援的人員名稱</li><li>– 若為獨立式 LDAP 登錄，則使用人員項目的 DN</li></ul></li></ul>

參數	使用	類型	說明
AlternativeName1	選用	字串	其他使用者名稱。使用這個參數來擷取多位使用者。
AlternativeName2	選用	字串	其他使用者名稱。使用這個參數來擷取多位使用者。

### 依使用者 ID 的使用者

使用此準則，可定義已知其使用者 ID 的使用者查詢。使用簡短名稱來指定值，例如 wpsadmin。此準則不需要人員目錄的存取權。所有人員目錄提供者都支援此準則。

參數	使用	類型	說明
UserID	必要	字串	要擷取的使用者的使用者 ID。此參數支援取代表式。
AlternativeID1	選用	字串	其他使用者 ID。使用這個參數來擷取多位使用者。
AlternativeID2	選用	字串	其他使用者 ID。使用這個參數來擷取多位使用者。

### 沒有指定使用者的依使用者 ID 的使用者

使用此準則，可定義已知其使用者 ID 的使用者的查詢，但排除明確指定的使用者 ID。使用簡短名稱來指定值，例如 wpsadmin。此準則不需要存取人員儲存庫。所有人員目錄提供者都支援此準則。

參數	使用	類型	說明
UserID	必要	字串	要擷取的使用者的使用者 ID。此參數支援取代表式。
AlternativeID1	選用	字串	其他使用者 ID。使用這個參數來擷取多位使用者。
AlternativeID2	選用	字串	其他使用者 ID。使用這個參數來擷取多位使用者。
NamedUsers	必要	字串	要從使用者 ID 清單排除的使用者 ID。此參數支援取代表式。

### 相關概念

『人員指派準則定義中的取代表式』

您可以使用取代表式作為部分人員指派準則定義中的參數值。人員解析可以根據環境定義提供的資訊，在執行時期解析指派準則。

### 人員指派準則定義中的取代表式

您可以使用取代表式作為部分人員指派準則定義中的參數值。人員解析可以根據環境定義提供的資訊，在執行時期解析指派準則。

例如，下列人員指派準則定義指定 `htm:input.\name` 取代表式作為參數：



```
<verb>
<name>Users by user ID</name>
<parameter id="UserID">%htm:input.\name%</parameter>
</verb>
```

此變數表示作業在起始時接收到的作業輸入訊息值的 "name" 元素。人員解析會使用實際的作業輸入訊息值來動態取代表示式。

#### 相關資訊

第 60 頁的『預先定義的人員指派準則』

提供預先定義人員指派準則，用來擷取人員目錄中的不同使用者組。

## 人員解析

人員解析會根據一組參數化查詢表示式 (稱為人員指派準則)，以從人員目錄擷取使用者資訊。

#### 相關概念

第 28 頁的『商業程序的授權角色』

角色是指具有相同授權層次的一組人員。您可以針對商業程序執行的動作，視您的授權角色而定。此角色可以是 J2EE 角色或實例型角色。

## 人員目錄

人員目錄會儲存用於人員解析的使用者資訊。

若要支援人員解析，人員目錄必須支援下列屬性：

- 識別使用者設定檔的名稱及使用者的登入 ID
- 若要利用與使用者的管理者相關的資訊，人員目錄應該提供對應屬性 (預設是管理者屬性)
- 若要利用電子郵件通知功能進行呈報，人員目錄應該提供使用者電子郵件位址

「商業程序編排器」支援下列人員目錄以進行人員解析。如果您想要利用「商業程序編排器」提供的完整功能集進行人員指派，請使用虛擬成員管理程式作為人員目錄。

- 聯合儲存庫 (也稱為虛擬成員管理程式)

這是 WebSphere Application Server 支援的預設人員目錄。它提供對各種目錄類型的存取，包括「輕量型目錄存取通訊協定 (LDAP)」目錄、資料庫和檔案型儲存庫及自訂儲存庫。也支援儲存庫聯合。

您可以擷取人員及群組資訊。支援的人員綱目 (PersonAccount 實體類型) 包括使用者的名稱、登入身分、管理者身分及電子郵件位址的內容。若要進行人員解析，則必須在 WebSphere Application Server 中將聯合儲存庫配置為作用中安全範圍定義。

- LDAP 目錄

「商業程序編排器」可以直接存取人員解析的 LDAP 目錄，而無需使用 WebSphere Application Server 安全。若要確保人員解析 (透過「商業程序編排器」實作) 與使用者鑑別 (透過 WebSphere Application Server 安全實作) 的一致性，則必須配置 WebSphere Application Server 安全，以存取「商業程序編排器」中指定要進行人員解析的相同 LDAP 目錄伺服器。

根據使用的 LDAP 人員綱目，人員相關資訊包括使用者名稱、身分、管理者名稱及電子郵件位址。若要進行人員解析，則需要「商業程序編排器」人員目錄提供者配置。

- **WebSphere Application Server 使用者登錄**

使用者登錄是用來擷取使用者資訊的應用程式伺服器子系統。「商業程序編排器」可以使用此使用者登錄作為人員目錄。「商業程序編排器」使用它專屬的使用者登錄人員目錄提供者來存取 WebSphere Application Server 使用者登錄。

#### **相關概念**

第 42 頁的『呈報』

呈報是人力作業未在指定的時間量動作時自動引發的警示。例如，如果作業未要求，或者未在定義的時間限制內完成。您可以為作業指定一個以上的呈報。這些呈報可以平行啟動，或者作為呈報鏈結啟動。

### **人員目錄提供者及配置**

「商業程序編排器」使用人員目錄提供者作為存取人員目錄的配接器。您可以配置虛擬成員管理程式、LDAP、使用者登錄及系統人員目錄提供者，以擷取使用者資訊。

決定要使用的人員目錄提供者，是視您需要的人員解析支援而定。若要利用「商業程序編排器」提供的所有人員指派功能，請使用虛擬成員管理程式。

在節點層次可以使用所有人員目錄提供者。

#### **虛擬成員管理程式人員目錄提供者**

虛擬成員管理程式人員目錄提供者可用來存取 WebSphere Application Server 聯合儲存庫。您可以使用此提供者來利用人員解析的下列部分：

- 聯合儲存庫功能，包括使用各種儲存庫 (例如，檔案及資料儲存庫、LDAP 目錄、內容延伸儲存庫及儲存庫聯合)
- 呈報的電子郵件通知
- 缺席者的代理
- 所有預先定義人員指派準則

#### **輕量型目錄存取通訊協定 (LDAP) 人員目錄提供者**

在不使用 WebSphere Application Server 的情況下，可以使用 LDAP 人員目錄提供者直接存取 LDAP 目錄。在大部分情況下，WebSphere Application Server 安全範圍是設為獨立式 LDAP 登錄，且配置成指向 LDAP 人員目錄提供者所參照的相同 LDAP 目錄。您可以使用此提供者來利用人員解析的下列部分：

- 呈報的電子郵件通知
- 所有預先定義人員指派準則

#### **使用者登錄人員目錄提供者**

您可以利用 WebSphere Application Server，使用使用者登錄人員目錄提供者來存取下列人員目錄：本端作業系統、獨立式 LDAP 登錄或獨立式自訂登錄。使用的人員目錄視應用程式伺服器安全範圍的配置而定。您可以使用此提供者來利用人員解析的下列部分：

- 「商業程序編排器」的人員目錄提供者的最小配置 (因為儲存庫是透過應用程式伺服器的安全範圍所決定)

- 有限的一組預先定義人員指派準則。使用者登錄人員目錄提供者可以解析使用者及群組，而不是員工與管理者關係、使用者內容或電子郵件位址。

### 系統人員目錄提供者

系統人員目錄提供者具備有限的人員解析支援。因為系統提供者只支援寫在程式中的查詢，所以只適用於測試。

所有人員目錄配置都需要啟用 WebSphere Application Server 管理及應用程式安全。

每個人員目錄提供者都可以與一個以上人員目錄提供者配置相關聯。您可以使用所有配置 (除了 LDAP 人員目錄提供者之外)。若為虛擬成員管理程式人員目錄提供者，則必須在 WebSphere Application Server 中配置聯合儲存庫功能。若為 LDAP 提供者配置，則必須設定必要的連線參數。此外，必須自訂 LDAP 提供者配置的轉換檔。

每個配置都是透過其「Java 命名目錄 (JNDI)」名稱予以唯一識別。JNDI 名稱是作業範本定義與人員目錄配置之間的鏈結，用來將人員指派解析為作業角色。使用 WebSphere Integration Developer，可以指定作業範本的配置名稱。如果是在執行時期使用作業建立 API 來定義作業，您可以直接在 API 中指定配置名稱。不同的作業範本可以參照不同的人員目錄配置。

部署作業範本之後，在所部署範本的生命期限內，人員目錄配置名稱會固定。如果需要變更與範本相關聯的人員目錄，請使用 WebSphere Integration Developer 來變更定義給作業範本定義的人員目錄配置的 JNDI 名稱，並重新部署範本。

#### 相關工作

第 170 頁的『配置 LDAP 人員目錄提供者』

您可以使用這項作業來配置「輕量型目錄存取通訊協定 (LDAP)」人員目錄提供者，讓「商業程序編排器」執行人員指派，決定誰可以啟動程序，或誰可以要求活動或作業。

第 169 頁的『配置虛擬成員管理程式人員目錄提供者』

配置「虛擬成員管理程式 (VMM)」人員目錄提供者，讓「商業程序編排器」執行人員指派，決定誰可以啟動程序，或誰可以要求活動或作業。預設人員目錄提供者已備妥可供使用，而且只在您引進自訂的人員指派準則時才需要配置。

### 對映人員指派準則與人員查詢

部署應用程式時，會將人員指派準則定義轉換為人員目錄配置特有的查詢集。產生的人員查詢會與作業範本一起儲存在「商業程序編排器」資料庫中。

如果使用虛擬成員管理程式作為人員目錄，則只有在定義自訂人員指派準則時，才需要變更轉換 XSL 檔中的預先定義對映。

轉換 (XSLT) 檔包含用來轉換人員指派準則的指示。每個人員目錄配置都與轉換檔相關聯。下列是提供給預設人員目錄配置的轉換檔：

- LDAPTransformation.xml，用於 LDAP 人員目錄提供者
- VMMTransformation.xml，用於虛擬成員管理程式人員目錄提供者
- UserRegistryTransformation.xml，用於使用者登錄人員目錄提供者
- SystemTransformation.xml 及 EverybodyTransformation.xml，用於系統人員目錄提供者

在 Windows® 平台上，這些檔案位於 *install\_root*\ProcessChoreographer\Staff 目錄中。在 Linux®、UNIX® 及 i5/OS® 平台上，這些檔案位於 *install\_root*/ProcessChoreographer/Staff 目錄中。

## 特定人員目錄提供者的人員查詢

與人員目錄配置相關聯的 XSL 轉換檔可用來產生特定儲存庫特有的人員查詢。每個查詢都可以由個別人員目錄提供者執行，以取得使用者 ID 的清單。可用於人員目錄提供者的預先定義查詢，會對應於可以由提供者執行的呼叫，並因此進行修正。

人員目錄提供者所提供的儲存庫特有查詢集，會對應於可用來從對應人員目錄擷取使用者資訊的方法。您可以使用這個設定的查詢來形成較複雜的查詢，如下列範例所示：

- 結合查詢結果，以將個別查詢傳回的使用者 ID 新增至使用者 ID 的現行結果清單。例如，LDAP 人員目錄提供者容許下列預先定義的查詢：

- 指定群組的群組成員的使用者 ID 清單：

```
<sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
...
</sldap:usersOfGroup>
```

- 指定使用者的識別名稱 (DN)：

```
<sldap:user dn="uid=user1,dc=mycomp" .../>
```

- 針對指定群組的成員使用者 ID 清單及指定使用者的 DN，可以建構複式查詢：

```
<sldap:staffQueries>
  <sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
    ...
  </sldap:usersOfGroup>
  <sldap:user dn="uid=user1,dc=mycomp" .../>
</sldap:staffQueries>
```

- 從現行結果清單中移除查詢結果。例如，下列 Snippet 顯示如何從針對指定群組成員擷取的 ID 清單中移除 "user1"：

```
<sldap:staffQueries>
  <sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
    ...
  </sldap:usersOfGroup>
  <sldap:remove value="user1"/>
</sldap:staffQueries>
```

- 使用從某個查詢取得的查詢結果，會影響後續查詢的行為。例如，在下列片段中，會執行兩個查詢。首先，會擷取使用者 "uid=user1,..." 的 LDAP 登錄中 "manager" 屬性的值，並將該值儲存在中間變數 "supervisor" 中。然後，會使用此變數來查閱管理者的 LDAP 項目，並擷取相關聯的使用者 ID。

```
<sldap:staffQueries>
  <sldap:intermediateResult name="supervisor">
    <sldap:user dn="uid=user1,dc=mycomp" attribute="manager" ... />
  </sldap:intermediateResult>
  <sldap:user dn="%supervisor%" .../>
</sldap:staffQueries>
```

根據這些組合規則建構的人員查詢，可由人員目錄提供者執行。

## 自訂人員指派準則的轉換

在下列狀況下，您可能需要自訂人員指派準則的轉換：使轉換適用於 LDAP 人員及群組綱目，或定義自訂人員指派準則。

若要使用自訂的轉換，您必須建立轉換檔。請勿變更預設的轉換檔，也不要將這些檔案名稱重複用於轉換檔。若為 LDAP 或虛擬成員管理程式人員目錄提供者的新轉換檔，

請一律在檔案中併入 **User Records by User ID** 人員指派準則定義。即使未針對人員指派明確指定此定義，「商業程序編排器」還是需要此定義。

若要使用新的轉換檔，請定義指向該檔案的新人員目錄配置。

### 相關工作

第 170 頁的『配置 LDAP 人員目錄提供者』

您可以使用這項作業來配置「輕量型目錄存取通訊協定 (LDAP)」人員目錄提供者，讓「商業程序編排器」執行人員指派，決定誰可以啟動程序，或誰可以要求活動或作業。

第 169 頁的『配置虛擬成員管理程式人員目錄提供者』

配置「虛擬成員管理程式 (VMM)」人員目錄提供者，讓「商業程序編排器」執行人員指派，決定誰可以啟動程序，或誰可以要求活動或作業。預設人員目錄提供者已備妥可供使用，而且只在您引進自訂的人員指派準則時才需要配置。

### 定義自訂人員指派準則：

您可能需要使用您專屬的準則來延伸這組預先定義人員指派準則。

當您建立自訂人員指派準則時，需要將它併入下列檔案中：

- VerbSet.xml 檔
- 轉換檔

若要使用自訂的轉換，您必須建立轉換檔。請勿變更預設的轉換檔，也不要將這些檔案名稱重複用於轉換檔。

例如，您建立了新的準則 **Mentor of Employee**。此準則與 **Manager of Employee** 準則類似，但它會擷取員工指導者的使用者 ID，而不是員工管理者的使用者 ID。

1. 將下列 XML Snippet 新增至 VerbSet.xml 檔：

```
<vs:DefineVerb name='Mentor of Employee'>
  <vs:Description>Assigns the mentor of an employee.
  Supported by sample XSLT files for:
  - LDAP
  </vs:Description>
  <vs:Mandatory>
    <vs:Parameter>
      <vs:Name>EmployeeName</vs:Name>
      <vs:Type>xsd:string</vs:Type>
    </vs:Parameter>
  </vs:Mandatory>
  <vs:Optional>
    <vs:Parameter>
      <vs:Name>Domain</vs:Name>
      <vs:Type>xsd:string</vs:Type>
    </vs:Parameter>
  </vs:Optional>
</vs:DefineVerb>
```

2. 在轉換檔中，定義新的 XSL 變數，以包含人員指導者的 LDAP 屬性：

```
<xsl:variable name="DefaultMentorAttribute">mentor</xsl:variable>
```

3. 將下列 XML Snippet 新增至轉換檔：

```
<!-- Begin template ManagerOfEmployee -->
<xsl:template name="ManagerOfEmployee">
  <slldap:staffQueries>
    <xsl:attribute name="threshold">
      <xsl:value-of select="{$Threshold}"/>
  </slldap:staffQueries>
</xsl:template>
```

```

        </xsl:attribute>

        <slldap:intermediateResult>
          <xsl:attribute name="name">mentorvar</xsl:attribute>
          <slldap:user>
            <xsl:attribute name="dn">
              <xsl:value-of select="staff:parameter[@id='EmployeeName']"/>
            </xsl:attribute>

            <slldap:resultObject>
              <xsl:attribute name="objectclass">
                <xsl:value-of select="$DefaultPersonClass"/>
              </xsl:attribute>
            <slldap:resultObject>
              <xsl:attribute name="usage">simple</xsl:attribute>

              <slldap:resultAttribute>
                <xsl:attribute name="name">
                  <xsl:value-of select="$DefaultMentorAttribute"/>
                </xsl:attribute>
                <xsl:attribute name="destination">intermediate</xsl:attribute>
              </slldap:resultAttribute>
            </slldap:resultObject>

            <slldap:user>
              <xsl:attribute name="dn">%mentorvar%</xsl:attribute>
              <xsl:call-template name="ResultObjectSpecForUserData"/>
            </slldap:user>
          </slldap:staffQueries>
        </xsl:template>
      <!-- End template ManagerOfEmployee -->

```

## 缺席者的代理

代理功能可讓您指定專屬的缺席設定，或您所管理的群組成員的缺席設定。代理原則可定義如何處理指派給缺席使用者的作業及呈報。

建立作業範本模型時，會定義代理原則。相同的原則適用於與作業範本相關聯的所有作業角色。部署作業範本之後，就無法變更原則。

如果使用者缺席，則會將代理原則套用至人員解析結果，以決定由誰接收工作項目，而不是缺席使用者。這只適用於具有人員指派準則的作業角色。這表示作業發送者、起始者或擁有者無法進行代理。同樣地，如果重新整理人員指派準則，就會重新整理代理。

根據特定的代理原則，會套用下列動作：

### 不進行代理 (預設值)

使用者集維持不變

### 使用者缺席時代理

- 針對出席的每位使用者，會使用使用者本身。
- 針對缺席的每位使用者，會使用出席的第一個代理人。
- 如果使用者及他們的代理人都未出席，則會套用預設人員指派規則。

### 出席時選取使用者

- 針對出席的每位使用者，會使用使用者。
- 不考慮代理人。



- 如果使用者未出席，則會使用原始的使用者集，亦即，不處理缺席狀況。

代理功能要求虛擬成員管理程式作為人員目錄。若要讓虛擬成員管理程式可用於進行代理，則必須在 WebSphere Application Server 中將聯合儲存庫配置為作用中安全範圍。請確定在管理主控台中啟用「人力作業管理程式」的代理。如果將具有非預設代理原則的作業範本部署給人員目錄提供者而非虛擬成員管理程式，則部署會失敗。

#### 相關工作

第 175 頁的『配置人員替代』

建立並啟動「虛擬成員管理程式 (VMM)」內容延伸儲存庫，讓「商業程序編排器」支援使用者替代。

第 303 頁的『指定缺席設定』

如果您想要離開辦公室一段時間，請指定作業的代理人。

第 304 頁的『指定使用者的缺席設定』

如果使用者無法處理作業，例如，如果他們因病休假，請指定使用者作業的代理人。

## 預設人員指派

如果未定義某些作業角色的人員指派準則，或者人員解析失敗或未傳回結果，則會執行預設人員指派。行內作業及獨立式作業的預設指派會不同。

### 行內作業

下表顯示行內作業的預設人員指派。

行內人力作業及其呈報的角色	如果作業模型中未定義角色 ...	如果人員指派失敗...
作業管理者	只套用繼承關係	只套用繼承關係
作業可能的實例建立者	每個人都變成可能的實例建立者	每個人都變成可能的實例建立者
作業可能的起始者	每個人都變成可能的起始者	每個人都變成可能的起始者
作業可能的擁有者	每個人都變成可能的擁有者	管理者變成可能的擁有者
作業編輯者	沒有編輯者	沒有編輯者
作業讀者	只套用繼承關係	只套用繼承關係
呈報接收者	管理者變成呈報接收者	管理者變成呈報接收者

下列繼承規則適用於行內作業：

- 程序管理者變成所有行內作業、其子作業、後續作業及呈報的管理者。
- 程序讀者變成所有行內作業、其子作業、後續作業及呈報的讀者。
- 作業管理者變成呈報所有子作業、後續作業及所有這些作業的管理者。
- 作業讀者變成呈報所有子作業、後續作業及所有這些作業的讀者。
- 任何作業角色的成員變成呈報此作業、子作業及後續作業的讀者。
- 呈報接收者變成已呈報作業的讀者。

## 獨立式作業

下表顯示獨立式作業的預設人員指派。

獨立式人力作業及其呈報的角色	如果作業模型中未定義角色 ...	如果人員指派失敗...
作業管理者	發送者變成管理者	未啓動作業
作業可能的實例建立者	每個人都變成可能的實例建立者	每個人都變成可能的實例建立者
作業可能的起始者	發送者變成可能的起始者	未啓動作業
可能的擁有者	每個人都變成可能的擁有者	管理者變成可能的擁有者
編輯者	沒有編輯者	沒有編輯者
讀者	只套用繼承關係	只套用繼承關係
呈報接收者	管理者變成呈報接收者	管理者變成呈報接收者

下列繼承規則適用於獨立式作業：

- 作業管理者變成呈報所有子作業、後續作業及所有這些作業的管理者。
- 作業讀者變成呈報所有子作業、後續作業及所有這些作業的讀者。
- 任何作業角色的成員變成呈報此作業、子作業及後續作業的讀者。
- 呈報接收者變成已呈報作業的讀者。

使用「商業流程管理程式」API 呼叫方法時，BPESystemAdministrator 角色的成員具有管理者權限，而 BPESystemMonitor 角色的成員具有讀者權限。透過「人力作業管理程式」API 呼叫方法時，TaskSystemAdministrator 角色的成員具有管理者權限，而 TaskSystemMonitor 角色的成員具有讀者權限。

### 相關概念

第 57 頁的『人力作業的授權角色』

您可以針對人力作業執行的動作，視授權角色而定。此角色可以是系統層次 J2EE 角色或實例型角色。

## 管理人員指派準則及人員解析結果

與作業授權角色相關聯的人員指派準則，適用於已部署作業範本或作業實例的生命期限。

如果需要變更人員指派準則，您必須在 WebSphere Integration Developer 中變更作業定義，並重新部署作業範本。

衍生自人員指派準則的人員查詢，會儲存為已部署作業範本或作業實例的一部分。在作業執行期間，授權角色需要解析相關聯的人員查詢。

人員查詢的結果視人員目錄的內容而定，而此內容可能會隨時間而變更。例如，可能會在人員群組中新增成員。若要反映人員目錄中的變更，則必須使用下列其中一種方式來重新整理人員查詢：

- 由管理者明確進行

管理者可以使用管理主控台或管理指令來重新整理人員查詢結果。下列動作具有指令：

- 立刻重新整理所有人員查詢結果
- 重新整理與作業範本相關聯的所有人員查詢結果
- 重新整理現行結果中包含特定使用者 ID 的人員查詢結果。
- 透過排程重新整理過期的人員查詢所觸發

此方式根據下列參數：

- 人員查詢結果的逾時值 ( $T_{out}$ )。
- 人員查詢的重新整理排程。使用 WebSphere Application Server CRON 語法來定義排程，例如，每個星期一的下午一點，或每個工作日的午夜。

下列參數決定如何自動重新整理人員查詢：

- 第一次執行查詢或重新整理查詢時，查詢結果會取得期限時間戳記 ( $t_{exp} = t_{current} + T_{out}$ )
- 呼叫查詢重新整理常駐程式時，會重新執行結果已過期的所有人員查詢

您可以將逾時值設為高於排程更新間隔。例如，您可以將逾時值設為 24 小時，而將更新間隔設為 1 小時。這樣就可以將一天的更新展開至人員查詢，並避免立即重新整理所有人員查詢結果的額外負荷。

#### 相關工作

第 250 頁的『使用管理主控台來重新整理人員查詢結果』  
人員查詢的結果是固定的。請使用管理主控台來重新整理人員查詢。

第 267 頁的『使用管理指令重新整理人員查詢結果』  
人員查詢的結果是固定的。使用管理指令來重新整理人員查詢。

第 252 頁的『使用重新整理常駐程式來重新整理人員查詢結果』  
如果想要設定定期及自動重新整理所有過期人員查詢結果，請使用這個方法。

## 共用人員指派

若為特定的作業角色，則作業範本的所有實例都會使用相同的人員指派準則。這是因為所有作業實例都是從相同的作業範本進行實例化。為了避免重新執行人員查詢，作業範本的作業實例會共用查詢的結果。

只有在人員指派準則定義包含固定的參數值時，才會共用結果。不論在其中解析人員查詢的作業實例環境定義為何，這類值 (例如，群組名稱：cn=group1, cn=groups) 表示對應的人員查詢結果會相同。

如果人員指派準則定義包含取代變數，則會將共用範圍縮小為具有相同取代變數值的人員指派。例如，參數值可以視作業輸入訊息的一部分而定。因為不同的作業實例可能會有不同的輸入訊息，所以人員查詢的參數值也會不同。

如果公佈程序人員查詢結果，則預設不會共用這些結果。若要共用後置處理結果，請在管理主控台中完成下列步驟：

1. 如果在伺服器上配置「商業程序編排器」，請按一下伺服器 → 應用程式伺服器 → **server\_name**。
2. 如果在叢集上配置「商業程序編排器」，請按一下伺服器 → 叢集 → **cluster\_name**。
3. 在商業整合之下，按一下商業程序編排器 → 人力作業管理程式 → [其他內容] 自訂內容。

4. 將 **Staff.PostProcessorPlugin.EnableResultSharing** 自訂內容的值變更為 `true`，並儲存變更
5. 重新啟動伺服器或叢集，讓變更生效。

#### 相關工作

第 472 頁的『建立對人員查詢結果進行後處理的外掛程式』

人員解析會傳回指派給特定角色的使用者清單 (例如，作業的可能擁有者)。您可以建立外掛程式，以變更人員解析所傳回的人員查詢結果。例如，若要改善工作量平衡，則您可能會有外掛程式，從查詢結果中移除已有高工作量的使用者。

---

## 第 2 篇 規劃及配置商業程序編排器





---

## 第 3 章 規劃配置商業程序編排器

計劃「商業程序編排器」設定與配置參數。

1. 執行『規劃拓撲、設定及配置路徑』。
2. 視您選擇的配置路徑而定，請執行下列其中一個動作：
  - 若為『基本範例』，請執行第 84 頁的『規劃建立「商業程序編排器」配置基本範例』。
  - 若為『含有組織的範例』，請執行第 85 頁的『規劃建立「商業程序編排器」配置範例，包括組織範例在內』。
  - 若為『非正式作業部署環境』，請執行第 85 頁的『規劃非正式作業部署環境配置』。
  - 若為『正式作業部署環境』，請執行第 86 頁的『規劃使用管理主控台的部署環境精靈』。
  - 若為『彈性自訂配置』，請執行第 88 頁的『規劃自訂商業程序編排器配置』。

您已計劃好所需要的一切，現在可以執行第 123 頁的第 4 章，『配置商業程序編排器』。

### 相關概念

第 109 頁的『關於商業程序編排器』

說明「商業流程管理程式」及「人力作業管理程式」提供的功能。

---

## 規劃拓撲、設定及配置路徑

您選擇的拓撲及設定會影響您可使用的「商業程序編排器」配置路徑。

不同的配置路徑在複雜性、彈性及其對不同拓撲和資料庫的支援各不相同。

1. 請注意，您必須在五個不同的配置路徑之間選擇。
  - 『基本範例』
  - 『含有組織的範例』
  - 『非正式作業部署環境』
  - 『正式作業部署環境』
  - 『彈性自訂配置』

對於大部分配置路徑，您可以選擇配置工具。

2. 請注意，有許多不同的配置工具可讓您用來配置「商業程序編排器」。

### 安裝程式或設定檔管理工具

提供最簡單的方式來建立非正式作業系統，且它們需要最少規劃。

- 『基本範例』配置包括下列「商業程序編排器」元件：
  - 商業程序編排器
  - 瀏覽器
  - 觀察程式及事件收集程式

- 『含有組織的範例』配置也包括人員目錄，它已預先配置為在一個組織範例中有 15 個使用者，並且已啓用代理和群組工作項目。
- 『非正式作業部署環境』配置提供簡單方式，在叢集上配置「商業程序編排器」，但「商業程序編排器」無法有它自己的資料庫，相反地，它使用一般 WPRCSDB 資料庫。

#### 管理主控台的部署環境精靈

可用來建立『正式作業部署環境』「商業程序編排器」配置 (根據部署環境型樣)。

#### 管理主控台的「商業程序編排器」配置頁面

您可以使用此管理主控台頁面，在伺服器或叢集上配置『彈性自訂配置』「商業程序編排器」正式作業系統。它提供設定許多配置參數的機會，而這需要詳細規劃。這個頁面不配置「商業程序編排器瀏覽器」或「商業程序編排器觀察程式」；它們有專屬的配置頁面，或可以藉由執行 Script 加以配置。這個配置路徑最適用於建立正式作業系統。

#### bpeconfig.jacl 配置 Script

您可以使用這個 Script，在給定的伺服器或叢集上配置『彈性自訂配置』「商業程序編排器」正式作業系統及所有必要資源。您可以互動方式執行這個 Script，如果您提供所有必要參數，則可以在批次模式中執行它，以達到可重複自動化的目的。它可以建立本端資料庫、必要的傳訊資源，並可以選擇性地配置「商業程序編排器瀏覽器」及「商業程序編排器觀察程式」。對於某些資料庫系統而言，它也可以建立遠端資料庫。這個配置路徑最適用於建立正式作業系統。

3. 請注意，有些配置路徑有限制，限制其在正式作業系統上的適用性：例如：
  - 在體驗其中一個配置範例之後，必須將它移除，然後您才可以建立適用於正式作業系統的配置。
  - 如果您建立的配置使用 Derby Embedded 資料庫或一般 WPRCSDB 資料庫，它將不適合高效能系統。在建立一個使用另一個高效能資料庫的新配置之前，您必須先移除此配置。
  - 若您使用 FILESTORE 或 Derby Embedded 訊息儲存庫，則無法將設定檔聯合到網路部署環境中。若要聯合設定檔，您必須完全移除「商業程序編排器」配置，並為訊息儲存庫建立一個使用可遠端存取的資料庫的新配置。
4. 識別決定要使用的配置途徑的主要準則。使用下表來確認選擇和限制：

表 4. 選取配置路徑的準則

適用於正式作業系統嗎？	部署目標	「商業程序編排器」配置	可以有不同的 BPEDB 資料庫嗎？	針對傳訊引擎支援的訊息儲存庫	適合的配置名稱、工具和選項
否	獨立式伺服器	基本範例  (不含組織範例、人員指派及代理)	是的，但僅限 Derby Embedded	僅限 Derby Embedded	『基本範例』，使用下列其中一項： • 安裝程式 • 設定檔管理工具  選取選項： • 獨立式伺服器設定檔 • 一般 • 啟用管理安全
		包括 15 人的組織範例，並已啓用人員指派及代理。  此範例與 WebSphere Integration Developer 中可用的範例相同。		Derby Embedded、檔案儲存庫或 WPRCSDB	『含有組織的範例』，它使用： • 設定檔管理工具  選取選項： • 獨立式伺服器設定檔 • 進階 • 從開發範本建立伺服器 • 啟用管理安全
	叢集	選擇部署環境型樣： • 遠端傳訊及遠端支援 • 遠端傳訊 • 單一叢集	不，它共用 WPRCSDB，這可以是 Derby Embedded 以外的任何資料庫	共用 WPRCSDB，它可以是「檔案儲存庫」及 Derby Embedded 以外的任何支援的資料庫	『非正式作業部署環境』，使用下列其中一項： • 安裝程式 • 設定檔管理工具  選取：部署環境
是	叢集	選擇部署環境型樣： • 遠端傳訊及遠端支援 • 遠端傳訊 • 單一叢集 • 自訂	是的，除 Derby Embedded 以外的任何支援資料庫	除「檔案儲存庫」及 Derby Embedded 以外的任何支援資料庫	『正式作業部署環境』，它使用： • 管理主控台  選取：部署環境
		彈性自訂配置	是的，任何支援的資料庫	除「檔案儲存庫」及 Derby Embedded 以外的任何支援資料庫  任何支援的資料庫或「檔案儲存庫」	『彈性自訂配置』，使用下列其中一項： • bpeconfig.jacl Script • 管理主控台「商業程序編排器」配置頁面
	獨立式伺服器				

**註：** 也可以使用建議用來建立正式作業系統的任何配置路徑，建立不適用於正式作業系統的配置。

請考慮下列選項：

- 決定您是否配置正式作業系統。一般而言，正式作業系統需要高效能、可調整性和安全。若為「商業程序編排器」，正式作業系統應該擁有它專屬的非 Derby BPEDB 資料庫。
- 決定「商業程序編排器」的部署目標是獨立式伺服器或叢集。

- c. 如果您不想建立正式作業系統，請決定獨立式伺服器上的配置範例是否符合您的需求。如果是，請決定您是否要該範例包含已啓用人員指派及代理的人員目錄範例。

**註：**人員目錄範例對聯合儲存庫使用檔案登錄，並以相同密碼『wid』包含所有人員範例。WebSphere 管理使用者 ID 也使用在設定檔建立期間指定的密碼來新增至該目錄。

- d. 若您要在叢集上配置「商業程序編排器」，視您的效能需求而定，請決定傳訊引擎及支援的應用程式 (例如「商業程序編排器瀏覽器」、「觀察程式」及「共用事件基礎架構」) 是否有它們專屬的叢集，或共用同一個叢集。標準部署環境型樣為：

#### 遠端傳訊及遠端支援

使用三個叢集。各用於應用程式、傳訊引擎及支援應用程式。

#### 遠端傳訊

一個叢集用於應用程式及支援函數。第二個叢集用於傳訊引擎。

#### 單一叢集

只有一個叢集用於應用程式、傳訊引擎及支援應用程式。

**自訂** 更有彈性的設定。

- e. 決定「商業程序編排器」是否有專用的 BPEDB 資料庫。
  - f. 決定「商業程序編排器」傳訊引擎要共用 WPRCSDB 資料庫，或有它專屬的訊息儲存庫，以及它將使用「檔案儲存庫」或資料庫系統。
  - g. 若您要使用「商業程序編排器觀察程式」，可在您建立「商業程序編排器」配置的同時配置它，或者，您也可以稍後建立。決定「商業程序編排器觀察程式」是否也要使用 BPEDB 資料庫，或它是否有專屬的 OBSRVADB 資料庫。請同時規劃「商業程序編排器觀察程式」元件的拓撲。如需相關資訊，請參閱第 107 頁的『規劃商業程序編排器觀察程式』。
5. 若您要「WebSphere 入口網站」存取「商業程序編排器」，請規劃在入口網站伺服器上配置「商業程序編排器」用戶端。同樣地，您可以配置「商業程序編排器」用戶端來啓用任何自訂 WebSphere Process Server 用戶端，以存取「商業程序編排器」。
  6. 如果您已啓用應用程式安全且擁有呼叫遠端 EJB 方法的長時間執行程序，則當您配置 CSIv2 入埠鑑別時，必須啓用 Common Secure Interoperability 第 2 版 (CSIv2) 身分主張。

您已規劃拓撲並且知道要使用哪一個配置路徑及配置工具。

---

## 規劃建立「商業程序編排器」配置基本範例

獨立式伺服器的基本範例，不包括人員指派。

您已執行第 81 頁的『規劃拓撲、設定及配置路徑』，並已選取『基本範例』配置路徑。

1. 決定是否要使用建立此範例安裝程式或設定檔管理工具。在這兩種情況下，此範例都相同，唯一的差異是您使用的工具。
2. 如果您想要「人力作業管理程式」能夠傳送「呈報電子郵件」，請規劃下列項目：
  - 如果沒有本端「簡易郵件傳送通訊協定 (SMTP)」郵件伺服器可用，請規劃於稍後變更郵件階段作業，以指向適當的郵件伺服器。

- 規劃變更電子郵件的寄件者地址。否則，它將使用虛擬的寄件者地址。
3. 請注意，此配置範例對不同的「商業程序編排器」使用者 ID 使用 WebSphere 管理者使用者 ID 及密碼。

---

## 規劃建立「商業程序編排器」配置範例，包括組織範例在內

此範例包括 15 人組成的組織範例，適合體驗在獨立式伺服器上進行人員指派及代理。此範例與 WebSphere Integration Developer 中可用的範例相同。

您已執行第 81 頁的『規劃拓撲、設定及配置路徑』，並已選取『含有組織的範例』配置路徑。

此「商業程序編排器」配置範例不太需要規劃。

1. 請注意，此範例只能使用設定檔管理工具建立。若要取得此範例，您必須選取下列選項：

- 獨立式伺服器設定檔
- 進階
- 從開發範本建立伺服器
- 啟用管理安全

例如，若您未啟用管理安全，則不會建立「商業編排器」配置範例。

2. 決定「商業程序編排器」傳訊引擎要使用檔案儲存庫、內嵌式 Derby 資料庫或一般 WPRCSDB 資料庫。
3. 如果您想要「人力作業管理程式」能夠傳送「呈報電子郵件」，請規劃下列項目：
  - 如果沒有本端「簡易郵件傳送通訊協定 (SMTP)」郵件伺服器可用，請規劃於稍後變更郵件階段作業，以指向適當的郵件伺服器。
  - 規劃變更電子郵件的寄件者地址。否則，它將使用虛擬的寄件者地址。
4. 請注意，此配置範例對不同的「商業程序編排器」使用者 ID 使用 WebSphere 管理者使用者 ID 及密碼。

---

## 規劃非正式作業部署環境配置

規劃使用「安裝程式」或「設定檔管理工具」，來建立一個根據部署環境型樣的「商業程序編排器」配置。

您已執行第 81 頁的『規劃拓撲、設定及配置路徑』，並已選取『非正式作業部署環境』配置路徑。

使用部署環境精靈時，您必須選取部署環境型樣，然後才有機會變更 WBI\_BPC 元件的預設資料庫參數及鑑別別名，並輸入「商業程序編排器」的其他參數。

1. 決定您要使用的部署環境型樣：
  - 遠端傳訊及遠端支援
  - 遠端傳訊
  - 單一叢集
2. 規劃您在「安全」步驟期間要輸入的「商業程序編排器」JMS 鑑別別名的使用者名稱。
3. 規劃「商業程序編排器」步驟的環境定義根目錄：

### 「商業程序編排器瀏覽器」環境定義根目錄

這定義瀏覽器要到達「商業程序編排器瀏覽器」所必須使用之 URL 的一部分。

### 「商業程序編排器觀察程式」環境定義根目錄

這會定義瀏覽器要到達「商業程序編排器觀察程式」必須使用之 URL 的一部分。

4. 規劃「商業程序編排器」步驟的安全參數。這些使用者 ID 及群組將用於「商業流程管理程式」及「人力作業管理程式」：

#### 管理者使用者及群組

規劃商業管理者角色對映到的使用者 ID 清單、群組清單或兩者。

#### 監視器使用者及群組

規劃商業監督者角色對映到的使用者 ID 清單、群組清單或兩者。

#### JMS API 鑑別使用者及密碼

「商業流程管理程式」訊息驅動 Bean 的執行身分使用者 ID。

#### 呈報使用者鑑別使用者及密碼

「人力作業管理程式」訊息驅動 Bean 的執行身分使用者 ID。

5. 如果您要使用人員指派，請執行第 105 頁的『規劃人員目錄提供者』。

---

## 規劃使用管理主控台的部署環境精靈

若為正式作業系統，請規劃「商業程序編排器」的所有配置參數，包括不同的資料庫在內。若為非正式作業系統，您可以使用共用資料庫。

您已執行第 81 頁的『規劃拓撲、設定及配置路徑』，並已選取『正式作業部署環境』配置路徑。

使用部署環境精靈時，您必須選取部署環境型樣，然後才有機會變更 WBI\_BPC 元件的預設資料庫參數及鑑別別名，並輸入「商業程序編排器」的其他參數。

1. 如果您沒有足夠資訊或權限可獨自建立整個配置，請聯絡系統其他部分的負責人員來規劃。例如：
  - 如果組織的 LDAP 伺服器使用的鑑別方法需要您要求使用者 ID 及授權，則您必須要求其相關資訊。
  - 如果您未獲授權建立資料庫，則必須在規劃資料庫中包含資料庫管理者，且資料庫管理者需要資料庫 Script 的副本來自訂及執行。
2. 執行第 89 頁的『規劃安全、使用者 ID 及權限』。
3. 決定您要使用的部署環境型樣：
  - 遠端傳訊及遠端支援
  - 遠端傳訊
  - 單一叢集
  - 自訂
4. 如果您選擇自訂部署環境型樣：
  - a. 決定是否要安裝「商業程序編排器瀏覽器」。如果要，請規劃您要在何處部署它。



- b. 決定是否要安裝「商業程序編排器」事件收集程式。如果要，請規劃您要在何處部署它。
  - c. 決定是否要安裝「商業程序編排器觀察程式」。如果要，請規劃您要在何處部署它。
  - d. 規劃 SCA 連結的環境定義根目錄。
  - e. 規劃您要啟用或停用狀態觀察程式及審核日誌記載。
5. 如果您規劃下列各項有專用的資料庫：
- 「商業程序編排器」的 BPEDB 資料庫 - 元件 WBI\_BPC。
  - 「商業程序編排器觀察程式」的 OBSRV RDB 資料庫 - 元件 WBI\_BPCEventCollector。
  - 「商業程序編排器」傳訊引擎的 BPEMEDB 資料庫 - 元件 WBI\_BPC\_ME。

為每一個資料庫規劃下列參數，在精靈的資料庫頁面上輸入：

#### 資料庫實例

資料來源的名稱，例如 BPEDB、OBSRV RDB 或 BPEMEDB，而非預設值 WPRCSDB，預設值會導致共用一般資料庫。預設值只適用於較低效能的設定。

**綱目** 要使用的資料庫綱目限定元。

#### 建立表格

如果已選取，則在第一次存取資料庫時就會自動建立表格。若要此選項生效，資料庫必須已經存在，且為了建立資料來源而提供的使用者名稱，必須具有在資料庫中建立表格及索引的權限。如果未選取，則不會自動建立表格，您必須執行 Script 來手動建立表格。若為正式作業系統，請清除此選項，並規劃使用所提供的 SQL Script 來設定資料庫。

#### 使用者名稱及密碼

具有權限連接到資料庫及修改資料的使用者 ID。如果使用者 ID 具有在資料庫中建立表格及索引的權限，則可以使用自動建立表格的選項，並且在套用服務程式或修正套件後，視需要自動更新資料庫綱目。

#### Server

資料庫伺服器的位址。指定主機名稱或 IP 位址。

**提供者** JDBC 提供者。

如需規劃資料庫的詳細資訊，請參閱第 94 頁的『規劃「商業程序編排器」的資料庫』。

6. 規劃您在「安全」步驟期間要輸入的「商業程序編排器」JMS 鑑別別名的使用者名稱。
7. 規劃「商業程序編排器」步驟的環境定義根目錄：

#### 「商業程序編排器瀏覽器」環境定義根目錄

這定義瀏覽器要到達「商業程序編排器瀏覽器」所必須使用之 URL 的一部分。

#### 「商業程序編排器觀察程式」環境定義根目錄

這會定義瀏覽器要到達「商業程序編排器觀察程式」必須使用之 URL 的一部分。

8. 規劃「商業程序編排器」步驟的安全參數。這些使用者 ID 及群組將用於「商業流程管理程式」及「人力作業管理程式」：

#### 管理者使用者及群組

規劃商業管理者角色對映到的使用者 ID 清單、群組清單或兩者。

#### 監視器使用者及群組

規劃商業監督者角色對映到的使用者 ID 清單、群組清單或兩者。

#### JMS API 鑑別使用者及密碼

「商業流程管理程式」訊息驅動 Bean 的執行身分使用者 ID。

#### 呈報使用者鑑別使用者及密碼

「人力作業管理程式」訊息驅動 Bean 的執行身分使用者 ID。

9. 若您想要配置電子郵件階段作業供「人力作業管理程式」呈報，請為「商業程序編排器」步驟規劃下列參數：

#### 郵件傳輸主機

「簡易郵件傳送通訊協定 (SMTP)」電子郵件服務所在的主機名稱或 IP 位址。

#### 郵件傳輸使用者及郵件傳輸密碼

如果郵件伺服器不需要鑑別，您可以將這些欄位保留空白。

#### 商業程序編排器瀏覽器 URL

這個 URL 是用來提供所產生的電子郵件的鏈結，因此接收到電子郵件通知的商業管理者可以按一下鏈結，並在他們的 Web 瀏覽器中檢視相關的商業程序或人力作業。

10. 如果您要使用人員指派，請執行第 105 頁的『規劃人員目錄提供者』。

---

## 規劃自訂商業程序編排器配置

使用管理主控台的「商業程序編排器」配置頁面或 `bpeconfig.jacl` 配置 Script，規劃配置參數及選項來建立自訂配置。

您已執行第 81 頁的『規劃拓撲、設定及配置路徑』，並已選取『彈性自訂配置』配置路徑。

1. 知道您將使用下列哪一項來配置「商業程序編排器」：
  - 管理主控台的「商業程序編排器」配置頁面
  - `bpeconfig.jacl` 配置 Script
2. 如果您沒有足夠資訊或權限可獨自建立整個配置，請聯絡系統其他部分的負責人員來規劃。例如：
  - 如果組織的 LDAP 伺服器使用的鑑別方法需要您要求使用者 ID 及授權，則您必須要求其相關資訊。
  - 如果您未獲授權建立資料庫，則必須在規劃資料庫中包含資料庫管理者，且資料庫管理者需要資料庫 Script 的副本來自訂及執行。
3. 第 89 頁的『規劃安全、使用者 ID 及權限』
4. 第 94 頁的『規劃「商業程序編排器」的資料庫』
5. 第 104 頁的『規劃商業流程管理程式及人力作業管理程式』
6. 第 105 頁的『規劃人員目錄提供者』

7. 第 106 頁的『規劃商業程序編排器瀏覽器』
8. 第 107 頁的『規劃商業程序編排器觀察程式』
9. 如果您打算使用管理主控台的「商業程序編排器」配置頁面，請確定您已規劃好所有需要在配置頁面輸入的值，如第 113 頁的『「商業程序編排器」配置』中所述。
10. 如果您將使用 `bpeconfig.jacl` 配置 Script：
  - a. 請確定您已規劃好必須在指令行或批次檔中指定的所有選項及參數值。選項及參數將彙總於第 137 頁的『使用 `bpeconfig.jacl` Script 來配置商業程序編排器』，並詳細說明於第 143 頁的『`bpeconfig.jacl` Script 檔』。
  - b. 如果您打算使用批次檔來執行 `bpeconfig.jacl` 配置 Script，請建立批次檔或 Shell Script。

您已計劃好所需要的一切，現在可以執行 第 123 頁的第 4 章，『配置商業程序編排器』。

## 規劃安全、使用者 ID 及權限

規劃配置「商業程序編排器」所需的使用者 ID 和權限。

在配置期間，您需要使用不同的使用者 ID，且必須指定執行時期會使用的其他使用者 ID。在開始配置「商業程序編排器」之前，請確定已先規劃和建立所有使用者 ID。

### 若為「商業程序編排器」配置範例：

您只需要建立新設定檔的權限。在「設定檔管理工具」中，使用建立一般設定檔的選項，當您啟用管理安全時，也會配置「商業程序編排器」範例。不需要其他規劃或使用者 ID，您可以跳過這項作業。

### 若為高度安全配置：

您必須詳細規劃所有使用者 ID，如這項作業所述。

### 若為低度安全配置：

如果不需要完整安全設定，例如在非正式作業系統中，您可以減少使用的使用者 ID 數目。您必須詳細規劃所有使用者 ID，但某些使用者 ID 可以有多種用途。例如，用來建立資料庫綱目的資料庫使用者 ID，也可以在執行時期作為資料來源使用者名稱來連接資料庫。

### 如果您要使用 `bpeconfig.jacl` Script 來配置「商業程序編排器」：

用來執行 `bpeconfig.jacl` Script 的使用者 ID，必須具有 (或將使用者 ID 指定為參數) Script 將執行的配置動作的必要權限。在此情況下，您必須詳細規劃所有使用者 ID。對於可在 `bpeconfig.jacl` Script 中指定為參數的使用者 ID，下表包含參數名稱。設定檔必須已存在。若已啟用 WebSphere 管理安全，則需要 configurator 角色有一個 WebSphere 管理者使用者 ID，您可以用它來呼叫 `wsadmin` 工具。

1. 將這一頁列印在紙張上，方便您將規劃的值寫在最後一欄。讓您在配置「商業程序編排器」時可以參考，並保存在記錄中供未來參考。
2. 規劃您在 WebSphere Process Server 上用來配置「商業程序編排器」的使用者 ID。

表 5. 規劃 WebSphere Process Server 的使用者 ID

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
配置「商業程序編排器」的使用者	配置	登入管理主控台及執行管理 Script。	WebSphere administrator 或 configurator 角色 (若已啟用 WebSphere 管理安全原則)。	
		如果您要執行 bpeconfig.jacl Script 來配置「商業程序編排器」。	執行 Script 時，您也必須為所選取的選項提供必要的使用者 ID。如需相關資訊，請參閱第 143 頁的『bpeconfig.jacl Script 檔』。	

3. 規劃哪些人需要存取 *install\_root* 的子目錄。如果您的安全原則不容許將此存取權授與這些人，則必須提供目錄中的檔案副本給這些人。

表 6. 規劃對 *install\_root* 子目錄的存取權

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
資料庫管理者	配置	執行 Script 以設定下列資料庫：  <b>BPEDB</b> ：「商業程序編排器」的資料庫。  <b>OBSRVDB</b> ：「商業程序編排器觀察程式」的資料庫。	如果您使用 bpeconfig.jacl Script 來配置「商業程序編排器」：  對於 bpeconfig.jacl 在該目錄的子目錄中產生的 createSchema.sql Script (或其副本) 的讀取權： <ul style="list-style-type: none"> <li>在 Windows 平台上：<i>profile_root</i>\dbscripts\ProcessChoreographer\</li> <li>在 Linux、UNIX、i5/OS，以及 z/OS 平台上的「UNIX 系統服務 (USS)」中：<i>profile_root</i>/dbscripts/ProcessChoreographer/</li> </ul>	
			如果您不使用 bpeconfig.jacl Script 來配置「商業程序編排器」：  對該目錄提供的資料庫 Script (或其中的檔案副本) 的讀取權： <ul style="list-style-type: none"> <li>在 Windows 平台上：<i>install_root</i>\dbscripts\ProcessChoreographer\<i>database_type</i></li> <li>在 Linux、UNIX、i5/OS，以及 z/OS 平台上的「UNIX 系統服務 (USS)」中：<i>install_root</i>/dbscripts/ProcessChoreographer/<i>database_type</i></li> </ul> 其中 <i>database_type</i> 是下列其中一項： <ul style="list-style-type: none"> <li>DB2</li> <li>DB2zOSV7</li> <li>DB2zOSV8</li> <li>Db2iSeries</li> <li>Derby</li> <li>Informix</li> <li>Oracle</li> <li>SQLServer</li> </ul>	

表 6. 規劃對 *install\_root* 子目錄的存取權 (繼續)

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
整合開發人員	自訂	若要搭配「輕量型目錄存取通訊協定 (LDAP)」或「虛擬成員管理程式 (VMM)」人員目錄提供者來指派人員，您必須自訂 XSL 轉換檔範例的副本。	<p>對 Staff 目錄或目錄中檔案副本的讀取權：</p> <ul style="list-style-type: none"> <li>在 Windows 平台上：<i>install_root</i>\ProcessChoreographer\Staff</li> <li>在 Linux、UNIX 及 i5/OS 平台上：<i>install_root</i>/ProcessChoreographer/Staff</li> </ul> <p>整合開發人員也需要具有適當目錄的寫入權，才能將自訂的 XSL 轉換檔提供給伺服器使用。</p>	

4. 規劃用來建立、配置及存取「商業程序編排器」所使用資料庫的使用者 ID。

表 7. 規劃 *BPEDB* 資料庫的使用者 ID

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
資料庫管理者	配置之前	建立 <i>BPEDB</i> 資料庫實例。若為 Oracle：建立 <i>BPEDB</i> 資料庫。	建立資料庫。	
資料庫管理者或將執行 <i>bpeconfig.jacl</i> Script 的管理者	配置	除非您使用內嵌的 Derby 資料庫，否則您或您的資料庫管理者必須執行「商業程序編排器」資料庫 Script。	若為 <i>BPEDB</i> 資料庫：變更表格、連接、插入表格及建立索引、綱目、表格、表格空間及視圖。	
資料來源使用者名稱  如果您使用 <i>bpeconfig.jacl</i> Script，則此為 <i>-dbUser</i> 參數。	配置	如果您選取 <b>建立表格</b> 選項，此使用者 ID 會用來建立資料庫表格。	若要使用 <b>建立表格</b> 配置選項，此使用者 ID 也必須獲得授權才能對 <i>BPEDB</i> 資料庫執行下列動作：變更表格、連接、插入表格，以及建立索引、表格和視圖。	
	執行時期	「商業流程管理程式」與「人力作業管理程式」使用此使用者 ID 來連接至 <i>BPEDB</i> 資料庫。	此使用者 ID 必須獲得授權才能對 <i>BPEDB</i> 資料庫執行下列動作：連接、刪除表格、插入表格、選取表格和視圖，以及更新表格。	
	套用服務或修正套件之後	必要的話，套用服務之後會自動更新資料庫綱目。這只有在此使用者 ID 具有必要的資料庫權限時才有效，否則必須手動更新綱目。	此使用者 ID 必須獲得授權才能對 <i>BPEDB</i> 資料庫執行下列動作：變更、建立、插入及選取表格；連接至資料庫；建立及捨棄索引與視圖。	

5. 如果您要配置「商業程序編排器觀察程式」，請規劃用來建立、配置及存取資料庫的使用者 ID。

表 8. 規劃 *OBSRVDB* 資料庫的使用者 ID

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
資料庫管理者	配置之前	建立 <i>OBSRVDB</i> 資料庫實例。若為 Oracle，則建立 <i>OBSRVDB</i> 資料庫。	建立資料庫。	

表 8. 規劃 OBSRVADB 資料庫的使用者 ID (繼續)

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
資料庫管理者或管理者	配置	執行 setupEventCollector 工具或 SQL Script 以建立綱目。	若為 OBSRVADB 資料庫：變更表格、連接、建立函數、插入表格，以及建立索引綱目、表格、表格空間和視圖。  如果您要使用 Java 實作的使用者定義函數，則使用者 ID 也必須獲得授權才能安裝 JAR 檔。	
事件收集程式資料來源使用者名稱	執行時期	連接到觀察程式資料庫。如果您使用「商業程序編排器觀察程式」，且該程式使用 BPEDB 資料庫，請使用與「商業程序編排器」資料來源相同的使用者名稱。	連接到資料庫	

6. 若「商業程序編排器」傳訊引擎的訊息儲存庫 (非內嵌式 Derby 或檔案儲存庫) 會有不同的資料庫，請規劃將用來存取該資料庫的使用者 ID。

表 9. 規劃預先配置的 BPEME 傳訊引擎資料庫的使用者 ID

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
匯流排資料來源使用者名稱  如果您使用 bpeconfig.jacl Script，則此為 -medbUser 參數。	配置及執行時期	此使用者名稱是用來連接到 BPEME 傳訊匯流排的資料來源，以及建立必要的表格和索引。	此使用者 ID 必須獲得授權才能對 BPEME 資料庫執行下列動作：連接、刪除表格、插入表格、選取表格和視圖，以及更新表格。	

7. 規劃「Java 訊息服務 (JMS)」提供者的使用者 ID。

表 10. 規劃 JMS 提供者的使用者 ID

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
JMS 鑑別使用者	執行時期	系統整合匯流排的鑑別別名。在配置「商業程序編排器」時必須指定它。  如果您使用 bpeconfig.jacl Script，則此使用者 ID 及其密碼為 -mqUser 及 -mqPwd 參數。	必須是有效的使用者名稱。它會自動新增至「商業程序編排器」匯流排的「匯流排連接器」角色。	



表 10. 規劃 JMS 提供者的使用者 ID (繼續)

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
JMS API 鑑別使用者	執行時期	任何「商業流程管理程式」JMS API 要求都會使用這個使用者 ID 來處理。  如果您使用 bpeconfig.jacl Script，則此使用者 ID 及其密碼為 -jmsBFMRUNASUser 及 -jmsBFMRUNASPwd 參數。	這些必須是有效的 ID。	
呈報鑑別使用者	執行時期	任何「人力作業管理程式」JMS API 要求都會使用這個使用者 ID 來處理。  如果您使用 bpeconfig.jacl Script，則此使用者 ID 及其密碼為 -jmsHTMRUNASUser 及 -jmsHTMRUNASPwd 參數。		

8. 規劃「商業流程管理程式」及「人力作業管理程式」的 J2EE 角色將對映的群組或使用者 ID。

表 11. 規劃「商業流程管理程式」及「人力作業管理程式」的安全角色

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID、群組或兩者的規劃清單
管理者使用者	執行時期	「商業流程管理程式」及「人力作業管理程式」的系統管理者和監督者安全角色各對映至使用者 ID、群組或兩者的清單。這裡定義的值會建立對映，提供所需的存取權給這個角色中的使用者。  如果您使用 bpeconfig.jacl Script，這些使用者和群組對應於下列參數：	
管理者群組	執行時期		
監督者使用者	執行時期		
監督者群組	執行時期		

9. 如果您要讓「人力作業呈報」傳送特定商業事件的通知電子郵件，且您的「簡易郵件傳送通訊協定 (SMTP)」伺服器需要鑑別，請決定用來連接電子郵件伺服器的使用者 ID。

表 12. 規劃電子郵件伺服器的使用者 ID

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
郵件傳輸使用者	執行時期	「人力作業管理程式」使用此使用者 ID 鑑別已配置的郵件伺服器來傳送「呈報電子郵件」。  如果您使用 bpeconfig.jacl Script，則此為 -mailUser 參數。此密碼是 -mailPwd 參數。	傳送電子郵件。	

10. 如果您要對人力作業使用人員指派，且將使用「輕量型目錄存取通訊協定 (LDAP)」人員目錄提供者 (使用簡式鑑別)，請規劃用來登入 LDAP 伺服器的使用者 ID。

表 13. 規劃 LDAP 伺服器的使用者 ID

使用者 ID 或角色	使用者 ID 的使用時機	使用者 ID 的用途	使用者 ID 必須具有的權限	規劃的使用者 ID
LDAP 外掛程式內容：鑑別別名	執行時期	當配置「輕量型目錄存取通訊協定 (LDAP)」人員目錄提供者時，且該人員目錄提供者使用簡式鑑別來連接 LDAP，例如 mycomputer/My LDAP Alias。當自訂 LDAP 外掛程式的內容時，您需要指定此使用者 ID。	如果 LDAP 伺服器使用簡式鑑別，則此使用者 ID 必須能夠連接 LDAP 伺服器。如果 LDAP 伺服器使用匿名鑑別，則不需要此使用者 ID。	

11. 以必要的權限來建立您規劃的使用者 ID。如果您沒有權限來自行建立所有的使用者 ID，請向適當的管理者提交要求，然後在下表中輸入他們為您建立的使用者 ID 的名稱。

在配置「商業程序編排器」時，您將會知道所需的使用者 ID。

## 規劃「商業程序編排器」的資料庫

規劃「商業程序編排器」的資料庫。取決於設定，您可能需要規劃建立最多三個資料庫，或完全不建立資料庫。

「商業程序編排器」可以與其他程序伺服器元件共用資料庫。「商業流程管理程式」和「人力作業管理程式」會使用 BPEDB 資料庫。若為正式作業系統，請為每一個已配置「商業程序編排器」的部署目標規劃一個專用的資料庫。如果您使用「商業程序編排器觀察程式」，該程式可使用相同的 BPEDB 資料庫，但若額外使用 OBSRVDB 資料庫，則效能會更好。「商業程序編排器」傳訊引擎可共用 SCA 傳訊引擎所使用的資料庫，或擁有它們專屬的 BPEMEDB 資料庫。如需所選取的配置路徑支援哪些資料庫的相關資訊，請參閱第 83 頁的表 4。

- 在非正式作業系統上，如果設定簡單性比效能更重要，您的選項將視所選擇的配置路徑而定：
  - 如果您要使用安裝程式或設定檔管理工具來建立『基本範例』「商業程序編排器」配置，則會建立不同的 Derby BPEDB 資料庫，此資料庫也可供「商業程序編排器觀察程式」使用。若為「商業程序編排器」傳訊引擎，預設值是擁有不同的 Derby 資料庫 (BPEME)，但如果您使用『組織範例』配置，則也可以選擇使用「檔案儲存庫」或共用 WPRCSDB 資料庫。

- 如果您要使用安裝程式或設定檔管理工具來建立一個包括「商業程序編排器」配置的部署環境，則「商業程序編排器」、「商業程序編排器觀察程式」及「商業程序編排器」傳訊引擎將全部使用 WPRCSDB 資料庫。因此，您不需要針對「商業程序編排器」進行任何資料庫規劃。
2. 若為正式作業系統：
    - a. 若效能很重要，請為「商業程序編排器」規劃使用不同的資料庫，如『規劃 BPEDB 資料庫』中所述，否則，請規劃使用 WPRCSDB 一般資料庫。
    - b. 若您將使用「商業程序編排器觀察程式」：
      - 若您想要使其查詢對商業程序效能的影響降至最低，請規劃使用不同的資料庫，如第 99 頁的『規劃 OBSRVDB 資料庫』中所述。
      - 否則，請規劃配置它以使用 BPEDB 資料庫。
    - c. 如果您要針對「商業程序編排器」傳訊引擎使用不同的資料庫來改進效能，請執行第 103 頁的『規劃傳訊引擎資料庫』，否則，請規劃使用「服務元件架構 (SCA)」所使用的預設資料庫。

您已規劃好「商業程序編排器」配置的所有資料庫。

## 規劃 BPEDB 資料庫

規劃「商業程序編排器」的資料庫。

「商業程序編排器」需要資料庫。已為所有支援的資料庫系統提供 SQL Script，來建立及管理資料庫綱目。有了資料庫之後，必須為「商業程序編排器」配置資料庫的 JDBC 存取權。視資料庫系統、拓撲、安裝目的及您選擇使用的管理工具而定，可以自動執行建立資料庫及配置 JDBC 存取權的部分或全部作業。若為正式作業系統，「商業程序編排器」應該有它自己的資料庫，但如果覺得效能不重要，您也可以配置「商業程序編排器」與其他 WebSphere Process Server 元件共用資料庫。

1. 請確定您選擇的 BPEDB 資料庫及配置路徑是相容的：支援的資料庫如下：
  - DB2 UDB for Linux、UNIX 及 Windows
  - DB2 for iSeries™
  - DB2 for z/OS®
  - Derby
  - Informix Dynamic Server
  - Microsoft® SQL Server
  - Oracle

如果您已決定要如何配置「商業程序編排器」，則您選擇的配置路徑將影響您建立資料庫的方式。如果您尚未決定要使用哪一個配置路徑來配置「商業程序編排器」，則識別資料庫需求有助於刪除不支援您的需求的配置路徑。如需每一個配置路徑支援哪些資料庫的詳細資訊，請參閱第 83 頁的表 4。

2. 如果您想要具有最少規劃及下列性質的簡式資料庫設定：
  - 它沒有提供正式作業系統通常需要的效能、可調整性或安全性。
  - 所有資料庫物件是建立在單一表格空間內，例如，預設表格空間。
  - 資料庫伺服器是 WebSphere Process Server 機器的本端伺服器。
  - 用來存取此資料庫的使用者 ID 也具有管理權限。

您需要規劃的選項視您選擇的配置路徑而定：

- 如果您使用**安裝程式**或**設定檔管理工具**來取得「商業程序編排器」配置範例，則會為「商業程序編排器」建立不同的 Derby BPEDB 資料庫，此資料庫不需要進一步規劃。
- 如果您使用管理主控台的**部署環境精靈**來配置「商業程序編排器」，請規劃使用已提供的 SQL Script 副本來建立 BPEDB 資料庫，它將在單一表格空間內建立預設綱目。
- **bpeconfig.jacl** 工具可為任何資料庫配置「商業程序編排器」及 JDBC 存取權。
  - 如果您要以**互動模式**執行 bpeconfig.jacl Script，可選取在現有的資料庫中建立表格。
  - 如果您有一個使用者 ID 具有建立資料庫物件的權限，可以使用 -createDB yes 選項，使 bpeconfig.jacl Script 產生及執行 SQL 檔案，在預設表格空間內建立資料庫物件。在此情況下，也要規劃停止伺服器，並對 wsadmin 公用程式使用 -conntype NONE 選項。如果您要使用 Oracle 或 DB2 for z/OS 資料庫，則資料庫實例必須已存在。若為其他資料庫類型，bpeconfig.jacl 將嘗試建立資料庫實例。如果在建立資料庫或物件時有發生任何錯誤，您可以使用所產生的 SQL Script，就好像使用 -createDB no 選項一樣。
  - 如果您沒有具有建立資料庫物件權限的使用者 ID，則必須使用 -createDB no 選項，使 bpeconfig.jacl Script 產生 SQL 檔案，而在預設表格空間內建立資料庫物件，但它並不會執行這個 Script。在此情況下，要求資料庫管理者為您自訂及執行 Script。

如需工具及其他資料庫參數的相關資訊，請參閱第 143 頁的『bpeconfig.jacl Script 檔』。

- 使用管理主控台的**商業程序編排器配置頁面**：
  - 若要在一般資料庫中建立「商業程序編排器」資料庫物件，請規劃使用預設資料庫實例作為「商業程序編排器」資料來源的目標。
  - 若要重複使用現有的資料庫，請規劃指定現有的資料庫實例作為「商業程序編排器」資料來源的目標。
  - 如果您選取建立表格選項，當「商業程序編排器」第一次使用此資料庫時，將在此資料庫中建立它需要的資料庫物件。這個選項無法用於 DB2 on z/OS 資料庫。
  - 若要使用 Script 建立此資料庫，請不要使用建立表格選項。
- 3. 如果您希望「商業程序編排器」具有下列性質的**高效能**資料庫設定：
  - 資料庫僅供「商業程序編排器」使用。
  - 資料庫伺服器最好在專用機器上，不過，它也可以是 WebSphere Process Server 機器的本端伺服器。
  - 您可以自訂磁碟的表格空間配置，以達到更好的效能。
  - 您可以使用不同於用來管理資料庫的使用者 ID 來存取此資料庫。

您必須規劃下列所有步驟：

4. 如果您尚未規劃資料庫的使用者 ID，請執行第 91 頁的表 7。
5. 規劃磁碟及表格空間的配置。資料庫主機最好有一個快速儲存體子系統，例如網路連結的儲存體或儲存區網路。若為正式作業系統，請將開發及系統測試期間遇到的結果納入考量。資料庫的大小視許多因素而定。以微流程方式執行的程序使用很少的空間，然而，每一個程序範本可能需要數十或數百 KB。

如果您使用個別磁碟，且資料庫系統支援將資料庫表格配置到不同磁碟，請規劃您將使用多少個磁碟以及如何配置它們。硬體輔助磁碟陣列通常會比單一磁碟提供更好的效能。如果您使用下列其中一項：

- DB2
- DB2zOSV7
- DB2zOSV8
- Informix (表格空間即所謂的 dbspaces)
- Oracle

規劃於何處尋找下列 BPEDB 資料庫表格空間：

- AUDITLOG
- COMP
- INDEXTS
- INSTANCE
- LOBTS
- STAFFQRY
- TEMPLATE
- WORKITEM
- SCHEDTS

它們可以全部在一個高效能 RAID 陣列上，但每一個表格空間應該在不同的檔案中，以容許平行存取。請記住，在給定的磁碟數目之下，使用 RAID 配置的效能會比將表格空間配置到不同磁碟更好。例如，以在具有 N 個處理器之專用伺服器上執行的 DB2 資料庫而言，請考慮使用下列準則：

- 若為表格空間，請使用具有 2\*N 個主要磁碟、2\*N 個鏡映磁碟且分段大小為 256 kb 的 RAID-1 陣列。
- 若為資料庫交易日誌，請使用具有 2\*N 個主要磁碟、2\*N 個鏡映磁碟且分段大小為 64 kb 的 RAID-1 陣列。

如果您使用 DB2 資料庫，而且要在 RAID 控制器上使用 15 個磁碟機，請考慮使用下列配置：

- 一個磁碟用於作業系統及分頁 (即 Windows 上的分頁檔、AIX® 及 HP-UX 上的分頁空間，以及 Solaris 上的交換空間)。
- 在 RAID-1 配置中使用八個磁碟 (四個主要磁碟和四個鏡映磁碟) 作為資料庫控制檔和表格空間的一個邏輯磁碟，所使用的分段大小為 256 kb。
- 在 RAID-1 配置中使用六個磁碟 (三個主要磁碟和三個鏡映磁碟) 作為資料庫交易日誌的一個邏輯磁碟，所使用的分段大小為 64 kb。

如果您使用 Oracle 資料庫，請考慮下列準則：

- 對所有檔案、跨越所有磁碟進行一切分段和鏡映 (SAME)，所使用的分段寬度為 1 MB。
- 高可用性的鏡映資料。
- 建立位於磁碟機外半部的分割區 (用於表格空間)。
- 依分割區而非磁碟分成資料子集。
- 使用「自動儲存體管理 (ASM)」檔案系統。



- 請勿區隔重做日誌與其他資料檔案。
6. 請規劃在執行 SQL Script 之前是由您或您的資料庫管理者自訂 SQL Script 來建立資料庫物件。
    - 如果您要使用 **bpeconfig.jacl** 工具來配置「商業程序編排器」，請使用 `-createDB no` 選項。這樣可避免該工具執行它所產生的 SQL Script。所產生的 SQL 檔案是根據已提供給您資料庫的標準 SQL 檔案，但其所有配置參數（提供給 `bpeconfig.jacl` 工具）已預先填入 SQL 檔案中，使所需的自訂作業減至最少。如需工具及其他資料庫參數的相關資訊，請參閱第 137 頁的『使用 `bpeconfig.jacl` Script 來配置商業程序編排器』。
    - 如果您使用管理主控台的**商業程序編排器配置頁面**或**部署環境精靈**來配置「商業程序編排器」，請規劃清除建立表格選項，以確定您不會取得預設綱目。請規劃自訂分送給資料庫系統的標準 SQL 檔案的副本。

如果您想要預覽資料庫的 SQL 檔案，以便規劃您的自訂作業，請尋找及檢視資料庫的 SQL `createSchema.sql` Script，但不要修改它。SQL 檔案位於：

- 在 Windows 平台上：`install_root\dbscripts\ProcessChoreographer\database_type`
- 在 Linux、UNIX、i5/OS，以及 z/OS 平台上的「UNIX 系統服務 (USS)」中：`install_root/dbscripts/ProcessChoreographer/database_type`

其中 `database_type` 是下列其中一項：

- DB2
  - DB2zOSV7
  - DB2zOSV8
  - Db2iSeries
  - Derby
  - Informix
  - Oracle
  - SQLServer
7. 若資料庫伺服器是程序伺服器的遠端伺服器，請在程序伺服器機器上規劃安裝「Java 資料庫連線功能 (JDBC)」驅動程式或資料庫用戶端：
    - 若為第二類型 JDBC 驅動程式：決定要安裝哪一個資料庫用戶端，以及安裝在何處。
    - 若為第四類型 JDBC 驅動程式：尋找驅動程式的 JAR 檔，並決定安裝在何處。
  8. 若資料庫伺服器是程序伺服器的本端伺服器，則存取資料庫所需的 JDBC JAR 檔會隨著資料庫系統進行安裝。尋找並記下這些 JAR 檔的位置。
  9. 如果您使用 DB2 for z/OS，請決定要使用的子系統。規劃在 Script 檔 `createTablespace.sql` 和 `createSchema.sql` 中用來取代儲存體群組名稱、資料庫名稱（不是子系統名稱）及綱目限定元的值。
  10. 決定資料庫所在的伺服器。如果資料庫伺服器在遠端，您需要適當的資料庫用戶端或具有 XA 支援的第四類型 JDBC 驅動程式。
  11. 決定下列配置參數的值，您需要為資料庫指定這些參數：
    - 「Java 資料庫連線功能 (JDBC)」提供者可能是第二類型或第四類型。若為 Oracle，請決定要使用 `oci` 或 `thin` 驅動程式。
    - 資料庫實例（若為 Oracle，為資料庫名稱；若為 DB2 on z/OS，為子系統名稱）。
    - 綱目限定元。預設會使用連線使用者 ID 作為隱含的綱目限定元。
    - 建立綱目的使用者名稱。



- 若您使用第四類型 JDBC 驅動程式：資料庫伺服器的名稱或 IP 位址。
- 資料庫伺服器使用的埠號。唯有當您使用第四類型 JDBC 驅動程式時，才需要這個。
- 鑑別別名的使用者 ID 及密碼。這是在執行時期 jdbc/BPEDB 資料來源用來存取資料庫的使用者 ID。這些是 bpeconfig.jacl 的 -dbUser 及 -dbPwd 參數。

如需這些參數 (包括預設值) 的相關資訊，請參閱第 113 頁的『「商業程序編排器」配置』或第 143 頁的『bpeconfig.jacl Script 檔』中的說明。

#### 12. 規劃支援足夠的平行 JDBC 連線：

- a. 評估「商業程序編排器」BPEDB 資料庫所需的平行 JDBC 連線數目上限。這將視商業程序及使用者數目的本質而定。可透過「商業程序編排器」API 並行連接的用戶端數目上限，加上以 BPEInternalActivationSpec 及 HTMInternalActivationSpec JMS 啟動規格定義的並行端點數目，再加上容許超載狀況的 10% 安全差，就是理想的評估值。
- b. 請確定資料庫系統可支援必要的平行 JDBC 連線數目。
- c. 根據資料庫系統的最佳作法規劃適當設定，以支援預期的平行 JDBC 連線數目。

#### 13. 若為正式作業系統，請規劃下列管理作業：

- 在移入一般正式作業資料之後，調整資料庫。
- 定期刪除資料庫中已完成的程序實例及作業實例。

您已規劃「商業程序編排器」的資料庫。

#### 相關工作

第 518 頁的『平衡硬體資源』

您可以透過平衡硬體資源來改善長時間執行的商業程序的效能。

### 規劃 OBSRVADB 資料庫

規劃「商業程序編排器觀察程式」的資料庫。

「商業程序編排器觀察程式」可使用相同資料庫，但若額外使用 OBSRVADB 資料庫，則效能會更好。若您不重複使用 BPEDB 資料庫，請執行下列作業：

1. 若您規劃有多個事件收集程式實例，且它們將使用相同的資料庫，請為每一個事件收集程式規劃唯一的綱目名稱。若要達到更好的效能，請為每一個事件收集程式規劃資料庫。
2. 決定要用於資料庫的資料庫系統：
  - Derby
  - DB2 UDB for Linux、UNIX 及 Windows
  - DB2 for iSeries
  - DB2 for z/OS
  - Oracle

**限制：**「商業程序編排器觀察程式」不支援使用 Informix 或 SQL Server 資料庫。

3. 決定資料庫所在的伺服器。
4. 如果您尚未規劃資料庫的使用者 ID，請執行第 91 頁的表 8。
5. 若您不使用 bpeconfig.jacl Script 來配置觀察程式及事件收集程式以使用 BPEDB 資料庫，請決定要如何建立 OBSRVADB 資料庫。

## 使用功能表導向的管理工具 `setupEventCollector`

您可以使用此工具，在互動模式下建立資料庫，並在執行期環境中驗證您的輸入。若您使用此工具，請決定您是否要此工具建立 SQL 檔，但不要執行它 – 如果您想要在執行 SQL 之前自訂它，或讓資料庫管理者自訂及執行它，請使用這個選項。如需這個工具的相關資訊，請參閱第 230 頁的『`setupEventCollector` 工具』。

此工具可讓您建立 Java 型使用者定義函數 (UDF) 或 SQL 型 UDF，在這兩個選項之間進行切換，以及安裝及移除支援 UDF 所需的 JAR 檔。若為非 Derby 資料庫，此工具支援使用 Java 型使用者定義函數 (UDF) 或 SQL 型 UDF 建立資料庫。若為 DB2 on z/OS 資料庫，此工具支援使用 Java 型 UDF 或 SQL 型 UDF 建立資料庫。若為 Derby 資料庫，只使用 Java 型 UDF 來建立資料庫。

## 執行 SQL Script

如果您不能使用此工具來存取資料庫，則可能需要使用 SQL Script。若為非 Derby 資料庫，所有 SQL Script 會使用 SQL 實作來建立「商業程序編排器觀察程式」的 UDF，此實作將簡化使用 SQL 型 UDF 配置的作業。  
若為 Derby 資料庫，只使用 Java 型 UDF 來建立資料庫。

## 第一次使用時會自動建立表格

在管理主控台的「商業程序編排器」配置頁面上選取**建立表格**選項，是取得預設資料庫綱目的捷徑。這個選項不適用於高效能系統。若為非 Derby 資料庫，會使用 SQL 型 UDF，以簡化配置作業。這個選項無法用於 DB2 on z/OS 資料庫。若為 Derby 資料庫，只使用 Java 型 UDF 來建立資料庫。

**註：**若您使用 DB2 for z/OS 資料庫，且偏好使用 Java 型 UDF (而非 SQL 型 UDF) 建立此資料庫，則您別無選擇，只能使用功能表導向的管理工具 `setupEventCollector`。若您使用 Derby 資料庫，將使用 Java 型 UDF，因為內嵌的 Derby 資料庫不支援 SQL UDF。

如需 UDF 的相關資訊，請參閱第 215 頁的『商業程序編排器觀察程式的使用者定義函數』。

6. 若您使用 DB2 for Linux、UNIX 或 Windows 資料庫，請規劃下列項目：
  - 資料庫名稱。預設值是 BPEDB。
  - 要用來連接資料庫的使用者 ID。您也必須知道這個使用者 ID 的密碼。
  - 用來建立資料庫物件的資料庫綱目名稱。預設值是連線使用者 ID。
  - 規劃表格空間 OBSVRTS 的完整位置。
  - 決定您是否要使用 SQL 型使用者定義函數 (UDF)，而非預設的 Java 型 UDF。
  - 若您要使用 `setupEventCollector` 工具來設定資料庫，則也要規劃下列各項：
    - 決定要使用的 JDBC 驅動程式類型：
      - 第二類型，使用原生資料庫用戶端連接。這是預設值。
      - 第四類型，透過 JDBC 直接連接。在此情況下，也要確定您知道下列各項：
        - 資料庫伺服器的主機名稱或 IP 位址。預設值是 localhost。
        - 用於資料庫的埠號。預設值是 50000。
    - 尋找安裝 DB2 JDBC 驅動程式檔案 `db2jcc.jar` 及 `db2jcc_license_cu.jar` 的目錄。

7. 若您使用 DB2 for i5/OS 資料庫，請規劃下列項目：
  - 資料庫名稱。若您在原生 iSeries 環境中配置資料庫，例如，在 qshell 中，請使用 \*LOCAL。否則，請使用 \*SYSBAS。
  - 要用來連接資料庫的使用者 ID。您也必須知道這個使用者 ID 的密碼。
  - 用來建立資料庫物件的資料庫綱目名稱。預設值是連線使用者 ID。
  - 決定您是否要使用 SQL 型使用者定義函數 (UDF)，而非預設的 Java 型 UDF。
  - 若您要使用 setupEventCollector 工具來設定資料庫，則也要規劃下列各項：
    - 資料庫伺服器的主機名稱。通常這一律為 localhost。埠號一律為 446。
    - JDBC 驅動程式的目錄：
      - 若資料庫處於原生 iSeries 環境，例如，在 qshell 中，這是 db2\_classes.jar 檔案所在的路徑，它通常是 /QIBM/ProdData/Java400/ext。
      - 若為遠端資料庫，則這是 jt400.jar 檔案所在的路徑。
8. 若您使用 DB2 for z/OS 資料庫，請規劃下列項目：
  - 子系統的位置名稱 (網路名稱)。
  - 儲存體群組名稱。
  - 子系統所知的資料庫名稱。預設值是 OBSRVADB。
  - 要用來連接資料庫的使用者 ID。您也必須知道這個使用者 ID 的密碼。
  - 用來建立資料庫物件的資料庫綱目名稱 (SQLID)。
  - 視您使用的 DB2 版本而定，請規劃要在哪一個儲存體群組中建立表格空間：
    - 若為 DB2 for z/OS 第 7 版，請使用：
      - OBSVRTS 的正規表格空間。
      - OS26201、OS26202、OS26203 及 OS26204 的大型物件 (LOB) 表格空間。
    - 若為 DB2 for z/OS 第 8 版，請使用：
      - OBSVR01、OBSVR02、OBSVR03、OBSVR04、OBSVR05、OBSVR06、OBSVR07 及 OBSVR08 的正規表格空間。
      - OS26201、OS26202、OS26203 及 OS26204 的 LOB 表格空間。
  - 若您想要使用 Java 型使用者定義函數 (UDF)，而非預設的 SQL 型 UDF，請決定您要用來在其中執行函數的 WLM 環境名稱。
  - 若您要使用 setupEventCollector 工具來設定資料庫，則也要規劃下列各項：
    - 決定要使用的 JDBC 驅動程式類型：
      - 第四類型，透過 JDBC 直接連接。在此情況下，也要確定您知道下列各項：
        - 資料庫伺服器的主機名稱或 IP 位址。預設值是 localhost。
        - 用於資料庫的埠號。預設值是 446。
        - JDBC 驅動程式 JAR 檔 db2jcc.jar 及 db2jcc\_license\_cisuz.jar 的目錄。
      - 第二類型，使用原生資料庫用戶端連接。在此情況下，請同時規劃會出現在本區型錄中的資料庫別名。
9. 若您使用 Derby 資料庫，請規劃下列項目：
  - 資料庫名稱。這必須是伺服器檔案系統上的完整路徑。預設值是 *install\_root/databases/BPEDB*。
  - 用來建立資料庫物件的資料庫綱目名稱。預設值是 APP。
  - 若您要使用 setupEventCollector 工具來設定資料庫，則也要規劃下列各項：

- 若您使用 Derby 網路 JDBC 驅動程式，請規劃用來連接資料庫的使用者 ID。您也必須知道這個使用者 ID 的密碼。
  - 決定要使用的 JDBC 驅動程式類型：
    - 內嵌的 JDBC 驅動程式。在此情況下，請同時規劃 JDBC 驅動程式 JAR 檔 `derby.jar` 的目錄。預設位置是 `install_root/derby/lib`。
    - 網路 JDBC 驅動程式。在此情況下，也要確定您知道下列各項：
      - JDBC 驅動程式 JAR 檔 `derbyclient.jar` 的目錄。預設位置是 `install_root/derby/lib`。
      - 若使用 Derby 網路伺服器，請決定 UDF JAR 檔 `bpcodbutil.jar` 在 Derby 網路伺服器上的位置。預設位置是 `install_root/derby/lib`。
      - 資料庫伺服器的主機名稱。預設值是 `localhost`。
      - 用於資料庫的埠號。預設值是 `1527`。
10. 若您使用 Oracle 資料庫，請規劃下列項目：
- SID 名稱。預設值是 `BPEDB`。
  - 決定要用來連接資料庫的 Oracle 使用者 ID。它必須有 `CONNECT` 及 `RESOURCE` 角色。預設使用者 ID 是 `system`。您也必須知道這個使用者 ID 的密碼。
  - 用來建立資料庫物件的資料庫綱目名稱。預設值是用來連接資料庫的使用者 ID。
  - 規劃下列每一個表格空間的完整位置：
    - `OBSVRIDX`
    - `OBSVRLOB`
    - `OBSVRTS`
  - 決定您是否要使用 SQL 型使用者定義函數 (UDF)，而非預設的 Java 型 UDF。
  - 若您要使用 `setupEventCollector` 工具來設定資料庫，則也要規劃下列各項：
    - JDBC 驅動程式檔案 `ojdbc14.jar` 的位置。
    - 資料庫伺服器的主機名稱。預設值是 `localhost`。
    - 用於資料庫的埠號。預設值是 `1521`。
11. 若您在批次模式下使用 `bpeconfig.jacl` 工具並設定 `-createEventCollector yes` 選項，請規劃下列其中一項：
- `-createDB yes` 選項可使該工具執行 `bpeconfig.jacl` 所產生的 SQL Script。您可以使用 `-dbSchema` 參數指定 `BPEDB` 資料庫的綱目限定元，但「商業程序編排器觀察程式」將在相同資料庫中使用相同的綱目。這就是為什麼使用 `-createDB yes` 選項不適用於高效能系統的緣故。
  - `-createDB no` 選項可避免該工具執行它所產生的 SQL Script。所產生的 SQL 檔案是根據已提供給您資料庫的標準 SQL 檔案，但其所有配置參數（提供給 `bpeconfig.jacl` 工具）已預先填入 SQL 檔案中，使所需的自訂作業減至最少。請規劃在執行 SQL Script 之前由您或資料庫管理者自訂所產生的 SQL Script 來建立資料庫物件。如需工具及其他資料庫參數的相關資訊，請參閱第 137 頁的『使用 `bpeconfig.jacl` Script 來配置商業程序編排器』。

**限制：** 您無法在互動模式下使用 `bpeconfig.jacl` Script 來配置「商業程序編排器觀察程式」。

12. 若您使用管理主控台的「商業程序編排器」事件收集程式頁面建立資料庫表格，請規劃下列其中一項：
  - 若為 DB2 on z/OS 以外的所有資料庫類型，您可以使用建立表格選項，使「商業程序編排器」第一次存取該資料庫時，該工具會在指定的資料庫中建立預設綱目。
  - 若您想要執行 SQL Script 來準備資料庫表格，請勿使用建立表格選項。請規劃在執行 SQL Script 之前由您或資料庫管理者自訂 SQL Script 副本來建立資料庫物件。這個選項最適合正式作業系統。
13. 若您想要預覽資料庫的 SQL Script，以便規劃您的自訂作業：請尋找及檢視資料庫的 createSchema\_Observer.sql 檔案，但不要修改它。SQL 檔案位於：
  - 在 Windows 平台上：*install\_root\dbscripts\ProcessChoreographer\database\_type*
  - 在 Linux、UNIX、i5/OS，以及 z/OS 平台上的「UNIX 系統服務 (USS)」中：*install\_root/dbscripts/ProcessChoreographer/database\_type*

其中 *database\_type* 是下列其中一項：

- DB2
- DB2zOSV7
- DB2zOSV8
- Db2iSeries
- Derby
- Oracle

**註：**如果您使用 `bpeconfig.jacl` 工具來配置「商業程序編排器」，請規劃使用該工具所產生的 SQL Script，您並不需要編輯這個 Script，就可以使用位置保留元的值來取代配置參數。所產生的 Script 只有在執行該工具之後才可供使用，但它們是基於以上列出位置的 Script。如果您想要自訂表格空間配置，則仍然必須編輯所產生的 Script 檔。

您已規劃「商業程序編排器觀察程式」的資料庫。

## 規劃傳訊引擎資料庫

您可以針對「商業程序編排器」匯流排的傳訊引擎使用不同的資料庫，以改進效能。

您可以對 WebSphere Process Server 建立的所有匯流排使用一個傳訊資料庫，也就是，「服務元件架構 (SCA)」系統匯流排、SCA 應用程式匯流排、「共用事件基礎架構」匯流排及「商業程序編排器」匯流排。訊息引擎所在叢集的所有成員都必須能夠存取此資料庫，以確保可隨時使用訊息引擎的失效接手功能。若效能很重要，請為「商業程序編排器」傳訊引擎規劃使用專用的資料庫，而不使用 SCA 匯流排及應用程式所使用的預設 MEDB。

1. 如果您要使用**安裝程式或設定檔管理工具**來取得其中一個「商業程序編排器」配置範例，請決定「商業程序編排器」傳訊引擎將使用 Derby Embedded、「檔案儲存庫」或 WPRCSDB 資料庫。
2. 「Java 資料庫連線功能 (JDBC)」提供者。請注意，「檔案儲存庫」及內嵌式的 Derby 資料庫無法用於網路部署環境。
3. 如果您要使用 WebSphere MQ，則必須使用 `bpeconfig.jacl` 配置 Script 來配置「商業程序編排器」。已淘汰使用 WebSphere MQ。
4. 如果您使用 `bpeconfig.jacl` 配置 Script 在獨立式伺服器上配置「商業程序編排器」，請決定要使用的訊息儲存庫：



- SCA 的傳訊引擎資料庫的個別綱目。
  - 不同的 Derby 資料庫。
  - 在執行 bpeconfig.jacl Script 之前，已建立在部署目標上的任何使用者定義資料來源。
  - 檔案儲存庫。
5. 如果您使用管理主控台的「商業程序編排器」配置頁面，請規劃下列配置參數。如需這些參數的相關資訊，請參閱「商業程序編排器」配置頁面上關於第 119 頁的『匯流排』的區段。
    - 本端或遠端匯流排成員位置。
    - 資料庫的名稱。預設值是 BPEME。
    - 綱目名稱。預設值是 MEDBPM00。
  6. 如果您要使用檔案儲存庫或內嵌式 Derby JDBC 提供者，將自動建立訊息儲存庫。
  7. 如果您不使用檔案儲存庫或內嵌式 Derby JDBC 提供者，請規劃下列配置參數。
    - a. 規劃在啟動「商業程序編排器」之前，資料庫已經存在。
    - b. 資料庫伺服器的主機名稱或 IP 位址，以及它所使用的埠號。
    - c. 用來連接至資料庫及建立綱目的使用者名稱。這是您在第 92 頁的表 9 中規劃的使用者 ID。

您已規劃「商業程序編排器」傳訊引擎的資料庫。

## 規劃商業流程管理程式及人力作業管理程式

「商業程序編排器」配置的核心包含「商業流程管理程式」及「人力作業管理程式」。您必須規劃其配置參數。

1. 請確定您知道「Java 訊息服務 (JMS)」提供者使用者 ID，此使用者 ID 將作為「商業流程管理程式」訊息驅動 Bean 的執行身分使用者 ID。在管理主控台和第 92 頁的表 10 中，此使用者 ID 稱為 **JMS API 鑑別使用者**。
2. 請確定您知道「Java 訊息服務 (JMS)」提供者使用者 ID，此使用者 ID 將作為「人力作業管理程式」訊息驅動 Bean 的執行身分使用者 ID。在管理主控台和第 92 頁的表 10 中，此使用者 ID 稱為 **呈報使用者鑑別使用者**。
3. 請確定您知道管理者及監督者的安全角色將對映至哪些群組或使用者 ID。如需詳細資訊，請參閱第 93 頁的表 11。
4. 如果您要「人力作業管理程式」以電子郵件來傳送呈報事件的通知，請指明「簡易郵件傳送通訊協定 (SMTP)」電子郵件伺服器所在的主機名稱或 IP 位址。規劃電子郵件通知的寄件者地址。如果電子郵件服務需要鑑別，請確定您知道用來連接服務的使用者 ID 和密碼。
5. 決定 API 的 Web 服務連結的環境定義根目錄。
  - 在伺服器上配置時：
    - 「商業流程管理程式」的預設值是 `/BFMIF_nodeName_serverName`
    - 「人力作業管理程式」的預設值是 `/HTMIF_nodeName_serverName`
  - 在叢集上配置時：
    - 「商業流程管理程式」的預設值是 `/BFMIF_clusterName`
    - 「人力作業管理程式」的預設值是 `/HTMIF_clusterName`



6. 決定一開始時要啓用「商業流程管理程式」或「人力作業管理程式」的審核日誌，或兩者都啓用。
7. 如果您要使用「商業程序編排器觀察程式」，請決定一開始時是否要配置「商業流程管理程式」來產生「共用事件基礎架構」記載事件。

您已規劃好「商業流程管理程式」與「人力作業管理程式」的所有起始配置參數。稍後，您可以使用管理主控台，隨時變更這些設定。

## 規劃人員目錄提供者

規劃「商業程序編排器」的人員目錄提供者、人員代理、虛擬成員管理程式及「輕量型目錄存取通訊協定 (LDAP)」等設定。

1. 若您要使用人力作業，請決定要使用哪些人員目錄提供者：

### 虛擬成員管理程式 (VMM) 人員目錄提供者

VMM 人員目錄提供者已備妥可使用預先配置給 WebSphere 安全的聯合儲存庫 (亦稱為虛擬成員管理程式) – 使用檔案儲存庫。若您想要使用不同於聯合儲存庫的使用者儲存庫，則需要重新配置聯合儲存庫。VMM 人員目錄提供者支援所有的「商業程序編排器」人員指派功能，包括代理在內。它根據聯合儲存庫提供的功能，例如支援不同的儲存庫類型，如 LDAP、資料庫、檔案型及內容延伸儲存庫。

若要使用 VMM 人員目錄提供者，爲了 WebSphere Application Server 安全，您必須已配置聯合儲存庫。您可以根據 LDAP 或資料庫，使聯合儲存庫與其他儲存庫相關聯。如需這方面的相關資訊，請參閱在聯合儲存庫配置中管理範圍。如需使用聯合儲存庫的相關資訊，請參閱 IBM WebSphere 開發人員技術日誌。

### 輕量型目錄存取通訊協定 (LDAP) 人員目錄提供者

在使用此人員目錄提供者之前，必須先加以配置。請執行步驟 2 中的規劃。

### 系統人員目錄提供者

此人員目錄提供者可直接使用，不需進行配置。請勿在正式作業系統中使用此提供者，它僅供應用程式開發測試之用。

### 使用者登錄人員目錄提供者

此人員目錄提供者可直接使用，不需進行配置。

2. 如果您要使用「輕量型目錄存取通訊協定 (LDAP)」，請規劃下列項目。
  - a. 您需要自訂專屬的 LDAPTransformation.xml 檔案版本。如需該檔案的位置及您可能需要自訂的內容清單，請參閱第 170 頁的『配置 LDAP 人員目錄提供者』。
  - b. 請規劃下列 LDAP 自訂內容：

LDAP 外掛程式內容	必要或選用性	說明
AuthenticationAlias	選用	用來連接 LDAP 的鑑別別名，例如 mycomputer/My LDAP Alias。您必須在管理主控台中定義此別名，作法是按一下安全 → 安全管理、應用程式及基礎架構 → Java 鑑別和授權服務 (JAAS) → J2C 鑑別資料。若未設定此別名，會使用 LDAP 伺服器的匿名登入。
AuthenticationType	選用	若此內容設定爲 simple，代表簡式鑑別，則需要 AuthenticationAlias 參數。否則，若未設定它，會使用匿名鑑別。
BaseDN	必要	所有 LDAP 搜尋作業的基本識別名稱 (DN)，例如 "o=mycompany, c=us"。若要指定根目錄，請指定空字串 ""。

LDAP 外掛程式內容	必要或選用性	說明
Casesentiveness ForObjectclasses	選用	決定 LDAP 物件類別名稱是否區分大小寫。
ContextFactory	必要	設定「Java 命名和目錄介面 (JNDI)」環境定義 Factory，例如， com.sun.jndi.ldap.LdapCtxFactory
ProviderURL	必要	此網址必須指向 LDAP JNDI 目錄伺服器地址。其格式必須符合一般 JNDI 語法，例如，ldap://localhost:389。若為 SSL 連線，請使用 LDAP 的 URL。
SearchScope	必要	所有搜尋作業的預設搜尋範圍。決定 baseDN 內容之下的搜尋深度。請指定下列其中一個值：objectScope、onelevelScope 或 subtreeScope
additionalParameter Name1-5 和 additionalParameter Value1-5	選用	使用這些名稱/值配對設定五個任意 JNDI 內容，以連線至 LDAP 伺服器。

3. 如果您要使用虛擬成員管理程式，請規劃下列項目。
  - a. 您需要自訂專屬的 VMMTransformation.xml 檔案版本。如需該檔案的位置及您可能需要自訂的內容清單，請參閱第 169 頁的『配置虛擬成員管理程式人員目錄提供者』。
4. 如果您要在正式作業環境中使用人員代理，請規劃使用「VMM 內容延伸儲存庫」來儲存代理資訊。這需要資料庫。「內容延伸儲存庫」及隱含選取的資料庫，在整個 WebSphere Cell 內必須是唯一的且可供存取。因為 BPEDB 資料庫在 Cell 內不一定是唯一的，所以無法使用 BPEDB。您可以使用一般資料庫 WPSRCDB 來管理「內容延伸儲存庫」，不過，若為正式作業環境，建議使用與其他 WebSphere Process Server 資料庫無關的資料庫。

您已規劃人員目錄提供者及人員指派選項。

## 規劃商業程序編排器瀏覽器

規劃「商業程序編排器瀏覽器」的配置參數。

若您要使用「商業程序編排器瀏覽器」，可在您配置「商業程序編排器」的同時規劃及配置它，或者，您也可以稍後執行。

1. 決定您要配置的「商業程序編排器瀏覽器」實例數目。您在配置「商業程序編排器」時可以輕易建立第一個實例。擁有多個實例的可能原因，包括：
  - 因為每一個「商業程序編排器瀏覽器」實例只能管理一個「商業程序編排器」配置，如果您的環境中有多個「商業程序編排器」配置，則為每一個配置設定一個「商業程序編排器瀏覽器」實例是合理的。
  - 如果您要自訂「商業程序編排器瀏覽器」，可以由兩個以上不同自訂版本的「商業程序編排器瀏覽器」來管理相同的「商業程序編排器」配置。

若需要更多實例，您必須使用管理主控台來個別建立每個實例，如第 121 頁的『商業程序編排器瀏覽器設定』中所述，或使用配置 Script，如第 183 頁的『使用 clientconfig.jacl Script 檔配置商業程序編排器瀏覽器』中所述。

2. 對於您要的每一個「商業程序編排器瀏覽器」實例，請規劃下列項目：
  - 「商業程序編排器瀏覽器」的環境定義根目錄。它在 Cell 內必須是唯一的。預設值是 /bpc。
  - 在呈報電子郵件中將插入的「商業程序編排器瀏覽器」的 URL。

如果由「商業程序編排器瀏覽器」管理相同的「商業程序編排器」配置，則「商業程序編排器」配置的「人力作業管理程式」只能鏈結到其中一個「商業程序編排器瀏覽器」實例。在此情況下，您必須決定「人力作業管理程式」要使用哪一個「商業程序編排器瀏覽器」URL。

- 查詢傳回的結果數目上限 - 預設值是 10000。
- 這個「商業程序編排器瀏覽器」將管理的「商業程序編排器」實例的部署目標 (伺服器或叢集)。

如需這些配置參數的相關資訊，請參閱第 121 頁的『商業程序編排器瀏覽器設定』。

您已規劃「商業程序編排器瀏覽器」的配置選項。

## 規劃商業程序編排器觀察程式

規劃配置「商業程序編排器觀察程式」及事件收集程式。

若您要使用「商業程序編排器觀察程式」，可在您配置「商業程序編排器」的同時規劃及配置它，或者，您也可以稍後執行。

1. 瞭解不同「商業程序編排器觀察程式」拓撲元素的用途及彼此間的關係。

### 觀察程式應用程式。

您可以在伺服器或叢集上配置多個實例。實例不一定要在已配置「商業程序編排器」的部署目標上。每一個實例透過資料來源連接至一個資料庫綱目。

### 事件收集程式應用程式。

此應用程式必須部署在已配置「共用事件基礎架構 (CEI)」伺服器的伺服器或叢集上。它不一定要部署在已配置「商業程序編排器」的地方。它會從 CEI 接收商業程序，轉換它們，並將它們寫入 OBSRVDB 資料庫中。

### OBSRVDB 資料庫。

事件收集程式及觀察程式是使用相同資料庫進行通訊。若為非正式作業系統，則可以與其他元件共用該資料庫。

您的選擇與「商業程序編排器」設定已有的拓撲無關。如需深入瞭解其可能性，請參閱第 111 頁的『關於商業程序編排器觀察程式』。

2. 請指明設定的目的、機器需求及拓撲含意。

### 簡單設定

若要更簡單的配置及管理，以及較低的效能，請在您已配置「商業程序編排器」及 CEI 的相同部署目標上部署觀察程式及事件收集程式應用程式，並使用本端資料庫系統。

### 高負載正式作業系統：網路部署

使用有多個節點的 Cell，它含有多個叢集。請在 Cell 的任何部署目標上安裝觀察程式應用程式的實例。請在您已配置「共用事件基礎架構 (CEI)」的叢集上安裝事件收集程式應用程式。使用不同的資料庫伺服器。

3. 如果您尚未規劃觀察程式的資料庫，請執行第 99 頁的『規劃 OBSRVDB 資料庫』。
4. 對於您要配置的每一個事件收集程式實例，請規劃下列項目：

- a. 決定要在何處安裝它。每一個部署目標只能安裝一個事件收集程式實例，且部署目標上必須已配置 CEI。
- b. 決定要如何配置此事件收集程式實例：
  - 使用管理主控台頁面。如需這個選項的相關資訊，請參閱第 219 頁的『使用管理主控台來配置商業程序編排器事件收集程式』。
  - 使用互動式 `setupEventCollector` 工具。如需這個選項的相關資訊，請參閱第 217 頁的『使用 `setupEventCollector` 工具來配置商業程序編排器事件收集程式』。
  - 在建立「商業程序編排器」配置的同時，使用 `bpeconfig.jacl` Script。- `createEventCollector` 選項有預設值 `yes`。

**註：**請不要使用 `bpeconfig.jacl` 來為高效能系統配置「商業程序編排器觀察程式」，因為 `bpeconfig.jacl` 將在與「商業程序編排器」配置相同的部署目標上，配置事件收集程式及觀察程式應用程式，並配置它們共用 `BPEDB` 資料庫。如需這個選項的相關資訊，請參閱第 137 頁的『使用 `bpeconfig.jacl` Script 來配置商業程序編排器』。

您無法在互動模式下使用 `bpeconfig.jacl` 來配置「商業程序編排器觀察程式」。

- c. 規劃資料來源：
  - 如果安全很重要：
    - 如果「商業程序編排器觀察程式」共用與「商業程序編排器」相同的實體資料庫，請為觀察程式資料庫規劃使用不同的資料來源，並規劃它的 JNDI 名稱。
    - 規劃資料庫將使用的鑑別別名。
  - 如果安全不重要：
    - 如果「商業程序編排器觀察程式」共用與「商業程序編排器 (`BPEDB`)」相同的實體資料庫，您可以使用相同的資料來源，以及相同的 JNDI 名稱。
    - 如果您使用相同的資料來源，則也可以使用相同的鑑別別名。
  - 規劃建立具有 `Cell` 範圍的資料來源。
- d. 規劃配置事件收集程式時需要的配置參數。
  - 資料庫的 JNDI 資料來源名稱。
  - 要用於資料庫物件的綱目。預設值是用來連接資料庫的使用者 ID。
  - 要用來連接資料庫的使用者 ID。預設值是 `db2admin`。
  - 使用者 ID 的密碼。
  - 若您使用第四類型 JDBC 連線，請同時收集資料庫伺服器的主機名稱或 IP 位址，以及它所使用的埠號。
  - 決定要在何處部署事件收集程式。此部署目標上必須已配置 CEI，因此，如果 CEI 有另一個叢集，請規劃在相同叢集上部署事件收集程式。
  - 若您要在網路部署環境中部署事件收集程式，必須知道 CEI 匯流排的傳訊引擎是配置在哪一個部署目標上。
  - 如果 CEI 匯流排已啓用安全原則，請規劃將用來鑑別 CEI 匯流排的 JMS 使用者 ID。
  - 決定在配置事件收集程式時是否要啓用 CEI 事件記載商業事件，或是否稍後才要使用管理主控台或執行 Script 來啓用它。

- e. 規劃執行時期配置值，在配置事件收集程式之後，您可能需要自訂這些值以符合您的需求：

- BpcEventTransformerEventCount
- BpcEventTransformerMaxWaitTime
- BpcEventTransformerToleranceTime
- ObserverCreateTables
- 若鑑別別名使用者 ID 不擁有資料庫綱目，請規劃 ObserverSchemaName。

如需這些值的相關資訊，請參閱第 226 頁的『變更商業程序編排器觀察程式的配置參數』。

5. 對於您要配置的每一個「商業程序編排器觀察程式」實例，請規劃下列項目：

- 決定要如何配置此實例：
  - 使用管理主控台頁面。如需這個選項的相關資訊，請參閱第 224 頁的『使用管理主控台來配置商業程序編排器觀察程式』。
  - 使用互動式 setupObserver 工具。如需這個選項的相關資訊，請參閱第 223 頁的『使用 setupObserver 工具來配置商業程序編排器觀察程式』。
  - 在建立「商業程序編排器」配置的同時，使用 bpeconfig.jacl Script。- createObserver 選項有預設值 yes。

**註：**請不要使用 bpeconfig.jacl 來為高效能系統配置「商業程序編排器觀察程式」，因為 bpeconfig.jacl 將在與「商業程序編排器」配置相同的部署目標上，配置事件收集程式及觀察程式應用程式，並配置它們共用 BPEDB 資料庫。如需這個選項的相關資訊，請參閱第 137 頁的『使用 bpeconfig.jacl Script 來配置商業程序編排器』。

- 此實例的部署目標。
- 此實例將監視的事件收集程式所使用的資料來源的 JNDI 名稱。
- 若您要使用管理主控台來配置此實例，則也要規劃下列各項：
  - 此實例的環境定義根目錄。這指定在瀏覽器中用來呼叫到這個「商業程序編排器觀察程式」實例的 URL 部分。預設值是 /bpcobserver。
  - 這個「商業程序編排器觀察程式」實例將以視覺化方式來呈現其資料的事件收集程式實例的部署目標。

如需這些配置參數的相關資訊，請參閱第 122 頁的『「商業程序編排器觀察程式」設定』。

6. 如果您要使用 bpeconfig.jacl Script 來配置「商業程序編排器」：

- 在批次模式中執行 Script 時，預設也會配置事件收集程式及觀察程式應用程式，而且它們將配置在與「商業程序編排器」配置相同的部署目標上。
- 若您不想要使用 bpeconfig.jacl 來配置事件收集程式及觀察程式應用程式的其中一個或兩個，請規劃使用 bpeconfig.jacl 選項 -createEventCollector no 和 -createObserver no 的其中一個或兩個，以防止 bpeconfig.jacl 配置它們。

您已規劃「商業程序編排器觀察程式」及事件收集程式的配置選項。

---

## 關於商業程序編排器

說明「商業流程管理程式」及「人力作業管理程式」提供的功能。



「商業程序編排器」是一種企業工作流程引擎，它支援 WebSphere Application Server 環境中的商業程序及人力作業。這些建構可以用來編排服務，以及整合涉及商業程序中人員的活動。「商業程序編排器」可管理商業程序及人力作業的生命週期、導覽相關聯的模型，以及呼叫適當的服務。

「商業程序編排器」提供了下列功能：

- 商業程序及人力作業支援。商業程序提供標準的方式來使用「Web 服務商業程序執行語言 (WS-BPEL，縮寫為 BPEL)」建立商業程序模型。使用人力作業，可以使用「作業執行語言 (TEL)」來建立涉及人員的互動模型。商業程序及人力作業在「服務導向架構 (SOA)」或「服務元件架構 (SCA)」中均顯現為服務；它們也支援簡單的資料物件及商業物件。
- 開發自訂應用程式的應用程式設計介面，用來與商業程序及人力作業互動。
- 「商業程序編排器」瀏覽器。此 Web 應用程式提供管理商業程序及人力作業的功能。如需相關資訊，請參閱『關於商業程序編排器瀏覽器』。
- 商業程序編排器觀察程式。此 Web 應用程式可讓您觀察執行中商業程序的狀態。如需相關資訊，請參閱第 111 頁的『關於商業程序編排器觀察程式』。

#### 相關工作

規劃配置商業程序編排器

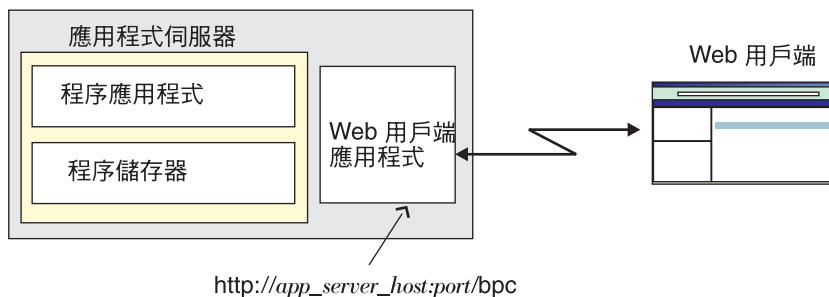
計劃「商業程序編排器」設定與配置參數。

### 關於商業程序編排器瀏覽器

「商業程序編排器瀏覽器」是 Web 應用程式，實作與商業程序及人力作業互動的一般 Web 使用者介面。

您可以在伺服器或叢集上配置一或多個「商業程序編排器瀏覽器」實例。具有 WebSphere Process Server 設定檔的 WebSphere Process Server 安裝，或 WebSphere Process Server 用戶端安裝就已足夠；並不需要在伺服器或叢集上配置「商業程序編排器」。用戶端安裝是將用戶端連接至 WebSphere Process Server 時唯一需要的基礎架構，但是它不包含「商業程序編排器瀏覽器」。然而，如果您擁有部署管理程式，則也可以將「商業程序編排器瀏覽器」安裝在 WebSphere Process Server 用戶端安裝的伺服器上。

單一「商業程序編排器瀏覽器」只可以連接至一個「商業程序編排器」配置，而不需要連接至本端配置。然而，您可以在相同伺服器或叢集上配置多個「商業程序編排器瀏覽器」實例，且每個實例都可以連接至不同的「商業程序編排器」配置。



當您啟動「商業程序編排器瀏覽器」時，您在使用者介面中看到的物件及採取的動作視您所屬的使用者群組及授權給該群組的權限而定。例如，如果您是商業程序管理



者，則負責已部署商業程序的順利運行。您可以檢視程序及作業範例、程序實例、作業實例及其關聯物件的相關資訊。您也可以處理這些物件；例如，您可以啟動新的程序實例、建立及啟動作業、修復及重新啟動失敗的活動、管理工作項目以及刪除整個程序實例及作業實例。然而，如果您是使用者，則只可以檢視及處理那些指派給您的作業。

## 關於商業程序編排器觀察程式

關於商業程序編排器觀察程式。

您可以使用「商業程序編排器觀察程式」，來建立已完成程序的報告。您還可以使用它來檢視執行中程序的狀態。本主題說明架構及可能的配置路徑。

「商業程序編排器觀察程式」使用「共用事件基礎架構 (CEI)」，可收集由 WebSphere Process Server 產生的事件。您可以使用許多預先定義的報告，或定義您的專屬報告，來取得程序、活動或其他聚集資料的數目概觀。您還可以取得關於特定程序或活動的資訊。

「商業程序編排器觀察程式」根據兩個 J2EE 企業應用程式，如下圖所示：

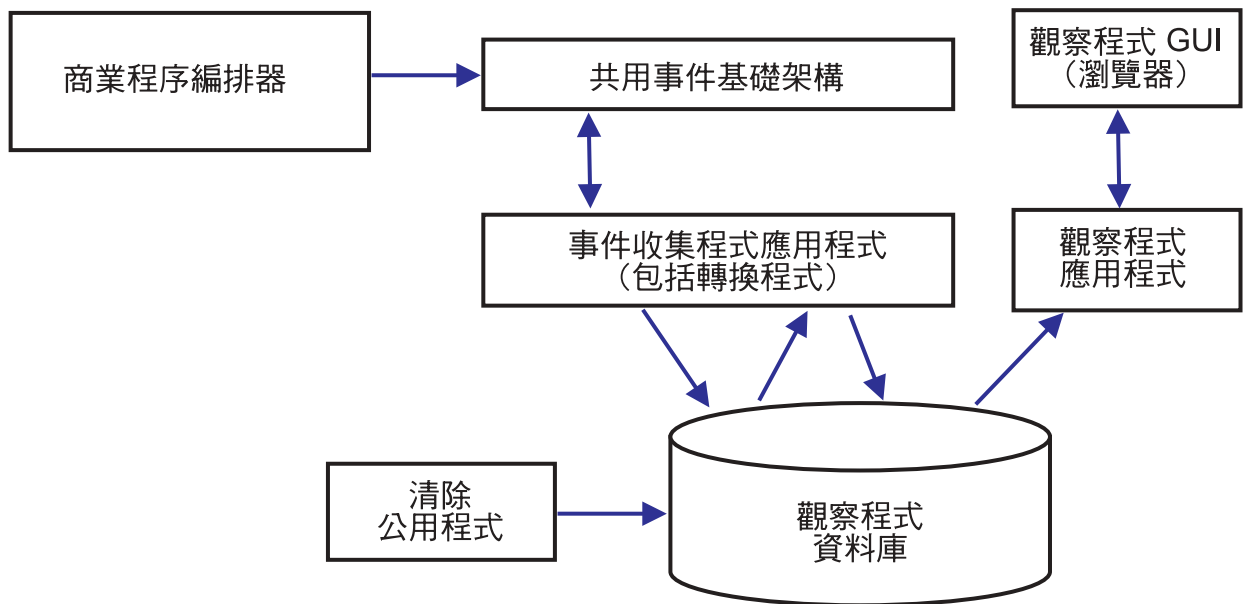


圖 1. 商業程序編排器觀察程式架構

- 事件收集程式應用程式會從 CEI 匯流排讀取事件資訊，並將其儲存在「商業程序編排器觀察程式」資料庫的事件收集程式表格中。
- 觀察程式資料庫是一組儲存事件資料的資料庫表格。
- 會定期地觸發事件轉換程式 J2EE 企業應用程式，將未處理的事件資料轉換為適於「商業程序編排器觀察程式」查詢的格式。
- 觀察程式應用程式會產生報告，並執行其他動作，而使用者可以使用觀察程式圖形使用者介面 (GUI) 來起始設定這些動作。
- 您可以使用 GUI 來產生您的報告。您也可以儲存及擷取您已定義的報告。
- 清除公用程式可以用來從觀察程式資料庫移除記錄，這有助於改進效能。

## 簡式配置

下圖說明效能不是重要考量的簡式配置。

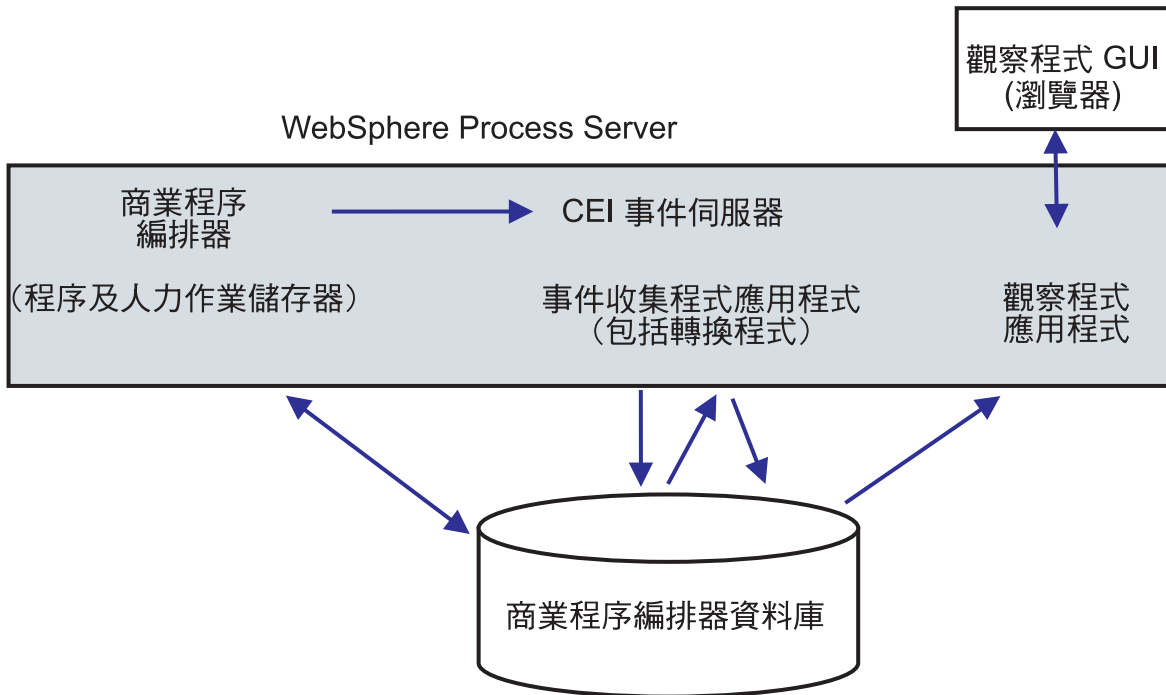


圖 2. 商業程序編排器觀察程式獨立式設定

所有程式都安裝在單一機器上，且「商業程序編排器」及「商業程序編排器觀察程式」使用相同的資料庫。

如果建立範例「商業程序編排器」配置，則會建立這類型的簡式配置。同時，`bpeconfig.jacl` 工具預設會將相同部署目標上的觀察程式及事件收集程式應用程式，配置為「商業程序編排器」配置。將啟用「共用事件基礎架構 (CEI)」記載，且會在「商業程序編排器」Derby 資料庫 `BPEDB` 中建立必要的資料庫綱目。如果效能不是重要的考量內容，則此配置路徑是理想的路徑。

## 高效能配置

提供的互動式配置工具可讓您自由利用「商業程序編排器觀察程式」架構的所有潛在功能。例如，在正式作業系統的理想效能配置中，三個「商業程序編排器」元素會在不同的機器上執行，且「商業程序編排器觀察程式」具有其專屬的資料庫。

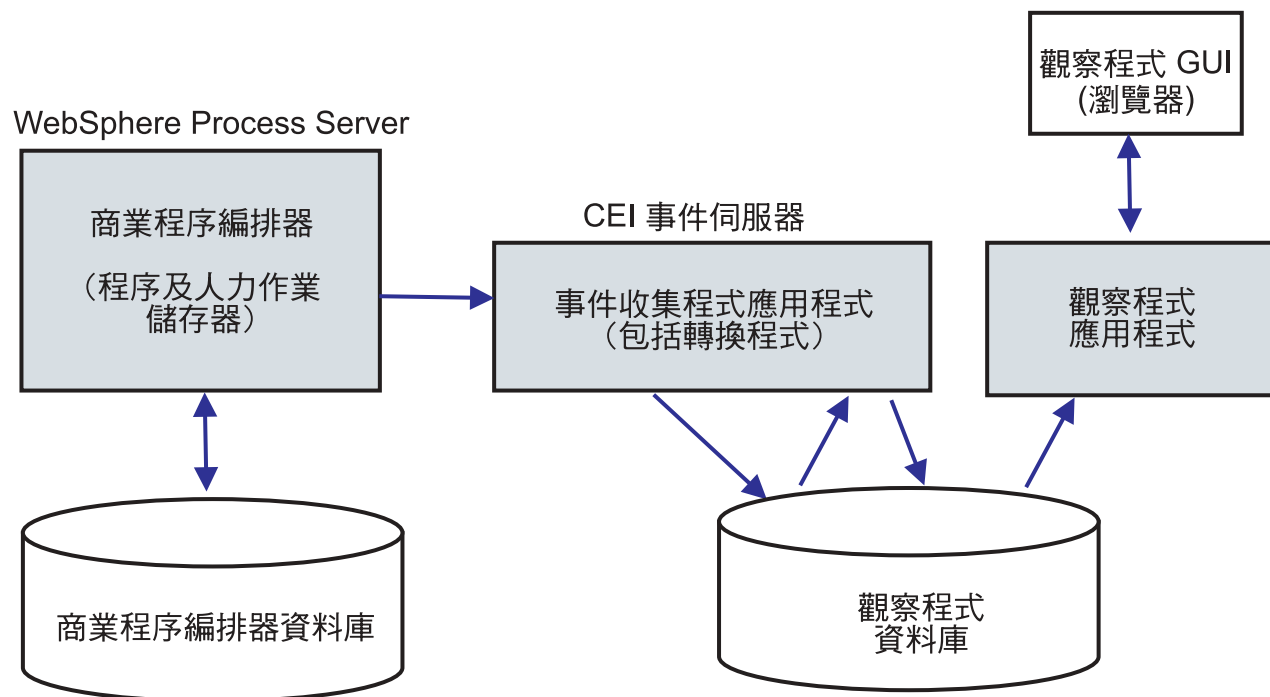


圖 3. 正式作業效能的商業程序編排器觀察程式設定

如果您想要針對「商業程序編排器觀察程式」使用不同的資料庫，或將「商業程序編排器觀察程式」新增至現有的「商業程序編排器」配置，請在叢集設定中，或使用更準確的資料庫選項，來執行第 186 頁的『配置商業程序編排器觀察程式』。

### 在網路部署環境中

如果想要在網路部署環境中配置「商業程序編排器觀察程式」，則適用下列限制。

- CEI 必須已在您的 Cell 中進行配置。
- 如前圖所示，在已配置「CEI 事件」伺服器的部署目標上，必須配置「商業程序編排器」事件收集程式。如果「CEI 事件」伺服器配置在與「商業程序編排器」不同的叢集上，則必須在已配置「CEI 事件」伺服器的部署目標上配置「商業程序編排器」事件收集程式。「商業程序編排器觀察程式」應用程式不需要安裝在與事件收集程式相同的機器上。

## 「商業程序編排器」配置

請利用這個畫面來安裝和配置「商業程序編排器」。

若要檢視此管理主控台頁面，請按一下伺服器 → 叢集 → *cluster\_name* 或伺服器 → 應用程式伺服器 → *server\_name*，然後按一下儲存器下的商業程序編排器儲存器。

此頁分成數個區段。如需每一個區段中的欄位詳細資訊，請參閱：

- 第 114 頁的『資料來源』
- 第 115 頁的『人力作業管理程式郵件階段作業』
- 第 116 頁的『安全』
- 第 118 頁的『狀態觀察程式』

- 第 118 頁的『SCA 連結』
- 第 119 頁的『匯流排』

## 資料來源

在此章節中，您可以指定「商業程序編排器」的資料來源。

## 編輯

按一下此按鈕來編輯資料來源。

## 測試連線

測試到資料來源的連線。

## 資料庫實例

「商業流程管理程式」與「人力作業管理程式」所使用的資料庫名稱。

內容	值
資料類型	字串
預設值	WPRCSDB

## 綱目名稱

要使用的綱目名稱。

如果您要使用自己的綱目而非預設綱目，則只需要變更綱目名稱。

內容	值
資料類型	字串

## 建立表格

如果已選取，則在第一次存取資料庫時就會自動建立表格。若要此選擇生效，資料庫必須已經存在，且所提供的使用者名稱必須具有在資料庫中建立表格及索引的權限。

在正式作業系統中，不建議使用此選項。如果未選取此選項，則不會自動建立表格，您必須執行 Script 來手動建立表格。

內容	值
資料類型	勾選框
預設值	已選取

## 使用者名稱

具有權限連接到資料庫及修改資料的使用者 ID。

如果使用者 ID 具有在資料庫中建立表格及索引的權限，則可以使用自動建立表格的選項，並且在套用服務程式或修正套件後，視需要自動更新資料庫綱目。

內容	值
資料類型	字串

## 密碼

資料來源使用者 ID 的密碼。

內容	值
資料類型	字串

## 伺服器

資料庫伺服器的位址。

請指定主機名稱或 IP 位址，以及埠號。

內容	值
資料類型	字串
範例	localhost:50000

## 提供者

「商業程序編排器」的 JDBC 提供者。

內容	值
資料類型	下拉清單

## 人力作業管理程式郵件階段作業

在此章節中，您可以指定呈報電子郵件的參數。

## 啓用電子郵件服務

如果「人力作業管理程式」要傳送呈報電子郵件通知，則您必須啓用郵件階段作業。

內容	值
資料類型	勾選框
預設值	已選取

## 郵件傳輸主機

「簡易郵件傳送通訊協定 (SMTP)」電子郵件服務所在的主機名稱或 IP 位址。

內容	值
資料類型	字串

## 郵件傳輸使用者

電子郵件服務的使用者 ID。

如果郵件伺服器不需要鑑別，您可以將這個欄位保留空白。

內容	值
資料類型	字串

## 郵件傳輸密碼

郵件傳輸使用者 ID 的密碼。

如果郵件伺服器不需要鑑別，您可以將這個欄位保留空白。

內容	值
資料類型	字串

## 商業程序編排器瀏覽器 URL

指定用於電子郵件鏈結到「商業程序編排器瀏覽器」的 URL。

這個 URL 是用來提供所產生的電子郵件的鏈結，因此接收到電子郵件通知的商業管理者可以按一下鏈結，並在他們的 Web 瀏覽器中檢視相關的商業程序或人力作業。

內容	值
資料類型	字串
範例	http://www.ibm.com:9080/bpc

## 安全

在此章節中，您可以指定功能角色與使用者 ID 及群組的對映，以及對「商業程序編排器」而言必要的鑑別認證。

### 管理者使用者

管理者安全角色對映到指定的使用者 ID。

內容	值
資料類型	字串
預設值	目前登入的使用者

### 管理者群組

管理者安全角色對映到指定的群組。

內容	值
資料類型	字串
預設值	無

### 監視者使用者

系統監視者安全角色對映到指定的使用者 ID。

內容	值
資料類型	字串
預設值	目前登入的使用者

### 監視者群組

系統監視者安全角色對映到指定的群組。



內容	值
資料類型	字串
預設值	無

### JMS 鑑別使用者

系統整合匯流排的鑑別別名。

內容	值
資料類型	字串
預設值	目前登入的使用者

### JMS 鑑別密碼及確認密碼

JMS 鑑別使用者 ID 的密碼。

內容	值
資料類型	字串
預設值	無

### JMS API 鑑別使用者

商業流程管理程式訊息驅動 Bean 的執行身分使用者 ID。

內容	值
資料類型	字串
預設值	目前登入的使用者

### JMS API 鑑別密碼及確認密碼

JMS API 鑑別使用者 ID 的密碼。

內容	值
資料類型	字串
預設值	無

### 呈報使用者鑑別使用者

人力作業管理程式訊息驅動 Bean 的執行身分使用者 ID。

內容	值
資料類型	字串
預設值	目前登入的使用者

### 呈報使用者鑑別密碼及確認密碼

呈報使用者鑑別使用者 ID 的密碼。

內容	值
資料類型	字串
預設值	無

## 狀態觀察程式

在此章節中，您可以針對「商業流程管理程式」及「人力作業管理程式」啓用審核日誌記載及「共用事件基礎架構 (CEI)」記載。

## 商業流程管理程式的審核日誌記載

選取時，即會啓用「商業流程管理程式」的審核日誌記載。

內容	值
資料類型	勾選框
預設值	已選取

## 人力作業管理程式的審核日誌記載

選取時，即會啓用「人力作業管理程式」的審核日誌記載。

內容	值
資料類型	勾選框
預設值	已選取

## 商業流程管理程式的共用事件基礎架構記載

選取時，即會啓用「商業流程管理程式」的「共用事件基礎架構」記載。

內容	值
資料類型	勾選框
預設值	已選取

## 人力作業管理程式的共用事件基礎架構記載

選取時，即會啓用「人力作業管理程式」的「共用事件基礎架構」記載。

內容	值
資料類型	勾選框
預設值	已選取

## SCA 連結

針對「服務元件架構 (SCA)」連結，您可以設定 Web 服務 API 的環境定義根目錄。

## 主機

這個唯讀欄位說明連結「商業流程管理程式」及「人力作業管理程式」主機的環境定義字首，其附加環境定義根目錄。

## 商業流程管理程式的環境定義根目錄

商業流程管理程式 Web 服務的環境定義根目錄。

內容	值
資料類型	字串
在伺服器上配置時的預設值	<code>/BFMIF_nodeName_serverName</code>
在叢集上配置時的預設值	<code>/BFMIF_clusterName</code>

## 人力作業管理程式的環境定義根目錄

人力作業管理程式 Web 服務的環境定義根目錄。

內容	值
資料類型	字串
在伺服器上配置時的預設值	/HTMIF_nodeName_serverName
在叢集上配置時的預設值	/HTMIF_clusterName

## 相對路徑

這個唯讀的欄位會顯示「商業流程管理程式」及「人力作業管理程式」的 SCA 連結相對路徑。

內容	值
資料類型	唯讀字串
商業流程管理程式的相對路徑	/sca/com/ibm/bpe/spi/sca/BFMWS
人力作業管理程式的相對路徑	/sca/com/ibm/task/spi/sca/HTMWS

## 匯流排

如果您要對「商業程序編排器」傳訊引擎使用與「服務元件架構 (SCA)」的配置不同的資料來源，請展開此部分來變更設定。

## 使用預設配置

若已選取，將會使用 SCA 傳訊引擎的現行配置設定。

若您要使用不同的設定，請清除勾選框，以啓用本節中的其他欄位。

內容	值
資料類型	勾選框
預設值	已選取

## 匯流排成員位置

定義要在本端或遠端儲存傳訊引擎的資料。

請選取**本端**或**遠端**。若您選取**遠端**，就會啓用遠端目的地位置選取元及**新建**按鈕。

內容	值
資料類型	圓鈕
預設值	本端

## 遠端目的地位置

指定遠端傳訊引擎儲存的佈署目標。

若清單是空的，或未包含您要選取的位置，請按一下**新建**。

內容	值
資料類型	下拉清單
預設值	無

## 新建

這個按鈕會開啓「瀏覽」部署目標頁面。

選取部署目標後，就會新增目標至遠端目的地位置的清單。

## 編輯

按一下此按鈕來編輯資料來源。

## 測試連線

測試到資料來源的連線。

## 資料庫實例

資料庫的名稱。

內容	值
資料類型	字串
預設值	<code>\${USER_INSTALL_ROOT}\databases\BPEME</code>

## 綱目名稱

要使用的綱目名稱。

內容	值
資料類型	字串
預設值	MEDBM00

## 建立表格

如果已選取，則在第一次存取資料庫時就會自動建立表格。若要此選擇生效，資料庫必須已經存在，且所提供的使用者名稱必須具有在資料庫中建立表格及索引的權限。

如果未選取，則不會自動建立表格，您必須執行 Script 來手動建立表格。若為正式作業系統，最好不要使用由此選項建立的預設表格。

內容	值
資料類型	勾選框
預設值	已選取

## 使用者名稱

具有權限連接到資料庫及修改資料的使用者 ID。

如果使用者 ID 具有在資料庫中建立表格及索引的權限，則可以使用自動建立表格的選項，並且在套用服務程式或修正套件後，視需要自動更新資料庫綱目。

內容	值
資料類型	字串

## 密碼

資料來源使用者 ID 的密碼。

內容	值
資料類型	字串

## 伺服器

資料庫伺服器的位址。

請指定主機名稱或 IP 位址，以及埠號。

內容	值
資料類型	字串
範例	localhost:50000

## 提供者

「商業程序編排器」傳訊引擎的 JDBC 提供者。

若您已將 SCA 配置為使用檔案儲存庫，這個欄位會設為檔案儲存庫，而資料庫參數的欄位則無法使用。選取資料庫提供者時，可以使用資料庫參數。

內容	值
資料類型	下拉清單
預設值	您針對 SCA 配置的提供者。

---

## 商業程序編排器瀏覽器設定

在此處可以設定一般內容。

若要檢視此管理主控台頁面，請按一下伺服器 → 叢集 → *cluster\_name* 或伺服器 → 應用程式伺服器 → *server\_name*，然後在商業整合之下，展開商業程序編排器，然後按一下瀏覽器配置。若要建立新的瀏覽器配置，請按一下新增。若要檢視現存的配置，請按一下實例名稱，或選取實例再按一下編輯。

### 環境定義根目錄

「商業程序編排器瀏覽器」的環境定義根目錄。

這定義瀏覽器要到達「商業程序編排器瀏覽器」所必須使用之 URL 的一部分。

內容	值
資料類型	字串
預設值	/bpc

### 瀏覽器搜尋結果限制

針對查詢返回之結果的最大數目。

內容	值
資料類型	整數
單位	搜尋結果
預設值	10000

## 受管理的「商業程序編排器」儲存器

「商業程序編排器」儲存器的範圍。

這指定此實例連接的「商業程序編排器」儲存器。這可以是多對一關係。每一個「商業程序編排器瀏覽器」實例只能連接至一個「商業程序編排器」儲存器。不過，每一個「商業程序編排器」儲存器可供零或多個「商業程序編排器瀏覽器」實例連接。

內容	值
資料類型	部署目標的下拉清單
預設值	已配置這個「商業程序編排器瀏覽器」實例的部署目標 (伺服器或叢集)。

---

## 「商業程序編排器觀察程式」設定

顯示「商業程序編排器觀察程式」之特定實例的環境定義根目錄及監視目標。

若要檢視此管理主控台頁面，請按一下**伺服器** → **叢集** → **cluster\_name** 或 **伺服器** → **應用程式伺服器** → **server\_name**，然後在**商業整合**之下，展開**商業程序編排器**，然後按一下**觀察程式配置**。若要建立新的觀察程式配置，請按一下**新增**。若要檢視現存的配置，請按一下實例名稱，或選取實例再按一下**編輯**。

### 環境定義根目錄

「商業程序編排器觀察程式」的環境定義根目錄。

這會定義瀏覽器要到達「商業程序編排器觀察程式」必須使用之 URL 的一部分。

內容	值
資料類型	字串
預設值	/bpcobserver

### 將來自這個「商業程序編排器」事件收集程式的監視資料視覺化

選取要由這個「商業程序編排器觀察程式」實例將其資料視覺化的事件收集程式實例。

內容	值
資料類型	下拉清單
預設值	已配置這個「商業程序編排器」觀察程式實例的部署目標 (伺服器或叢集)。



---

## 第 4 章 配置商業程序編排器

在安裝含有商業程序或人力作業的任何企業應用程式之前，必須先配置「商業程序編排器」。

您已完成第 81 頁的第 3 章，『規劃配置商業程序編排器』。

視您選擇的配置路徑而定，請執行下列其中一個動作：

- 若為下列任何非正式作業配置路徑：

- 『基本範例』
- 『含有組織的範例』
- 『非正式作業部署環境』

執行『使用安裝程式或設定檔管理工具來配置商業程序編排器』。

- 若為『正式作業部署環境』配置路徑，請執行第 125 頁的『使用管理主控台的部署環境精靈來配置商業程序編排器』。
- 若為『彈性自訂配置』配置路徑，視您要使用的工具而定，請執行下列其中一項：
  - 第 127 頁的『使用管理主控台的商業程序編排器配置頁面』
  - 第 137 頁的『使用 bpeconfig.jacl Script 來配置商業程序編排器』

已配置「商業程序編排器」。

您可以開始自訂設定。

---

### 使用安裝程式或設定檔管理工具來配置商業程序編排器

有五個簡單的方式可建立非正式作業「商業程序編排器」配置。

您已完成第 84 頁的『規劃建立「商業程序編排器」配置基本範例』，並已決定所要的非正式作業系統特性，如第 83 頁的表 4 中所彙總。

1. 視您選取的配置路徑而定，請執行步驟 1a - 第 124 頁的 1c 的其中一個步驟。
  - a. 如果您想要的「商業程序編排器」配置為『基本範例』，它不包括人員指派及代理的組織範例：
    - 1) 啟動「安裝程式」或「設定檔管理工具」。
      - 若為「安裝程式」：
        - 請確定您選取**一般安裝**選項。
        - 請確定您選取**獨立式伺服器**選項。
        - 請確定您已啓用**管理安全**。
      - 若為「設定檔管理工具」：
        - 請確定您建立 **WebSphere Process Server** 設定檔。
        - 請確定您選取**獨立式伺服器設定檔**選項。
        - 請確定您選取**一般設定檔建立**選項。
        - 請確定您選取**啓用管理安全**。

- b. 如果您想要的「商業程序編排器」配置為『含組織的範例』，它包括人員指派及代理的 15 人組織範例：
  - 1) 啟動「設定檔管理工具」。
  - 2) 請確定您建立 **WebSphere Process Server** 設定檔。
  - 3) 請確定您選取**獨立式伺服器**設定檔選項。
  - 4) 請確定您選取**進階**選項。
  - 5) 請確定您選取**從開發範本建立伺服器**選項。
  - 6) 請確定您選取**啟用管理安全**。
  - 7) 請確定您選取**配置商業程序編排器範例**。
- c. 如果您想要根據部署環境型樣配置的『非正式作業部署環境』「商業程序編排器」：
  - 1) 啟動「安裝程式」或「設定檔管理工具」。
    - 若為「安裝程式」：
      - 請確定您選取**部署環境安裝**選項。
      - 請確定您建立部署管理程式。
      - 您可以根據下列任何型樣配置「商業程序編排器」：
        - 遠端傳訊及遠端支援
        - 遠端傳訊
        - 單一叢集
      - 請確定您已啟用**管理安全**，否則您不會獲得「商業程序編排器」配置。
    - 若為「設定檔管理工具」：
      - 請確定您建立 **WebSphere Process Server** 設定檔。
      - 請確定您選取**部署管理程式**設定檔選項。
      - 您可以根據下列任何型樣配置「商業程序編排器」：
        - 遠端傳訊及遠端支援
        - 遠端傳訊
        - 單一叢集
      - 請確定您已啟用**管理安全**，否則您不會獲得「商業程序編排器」範例。
  - 2) 建立並聯合自訂設定檔。
2. 選擇性的: 執行第 235 頁的『驗證商業程序編排器是否運作』。
3. 選擇性的: 如果您想要變更 **JMS** 鑑別使用者 ID、執行身分使用者 ID 或角色與使用者和群組的對映，請按一下**安全** → **商業整合安全**，來變更安全設定。
4. 選擇性的: 變更「人力作業管理程式」的設定：
  - 如果您要變更任何「人力作業管理程式」設定以呈報電子郵件，例如寄件者位址或「商業程序編排器瀏覽器」的 URL 字首，請按一下**伺服器** → **應用程式伺服器** → **server\_name** 或**伺服器** → **叢集** → **cluster\_name** (如果「商業程序編排器」是配置在叢集上)，然後在**商業整合**下，展開**商業程序編排器儲存器**，再按一下**人力作業管理程式**，並進行變更。

- 如果您想要變更電子郵件伺服器位址或埠號、電子郵件伺服器的使用者 ID 或密碼，請按一下**資源** → **郵件** → **郵件階段作業**，選取 **Cell** 範圍，然後按一下 **HTM 郵件階段作業**，並進行變更。
5. 視您用於「人員指派」的人員目錄提供者類型而定，您可能需要配置人員目錄提供者：
    - 系統和使用者登錄人員目錄提供者可以直接使用，而不需要配置。
    - 如果您使用「輕量型目錄存取通訊協定 (LDAP)」，請執行第 170 頁的『配置 LDAP 人員目錄提供者』。
    - 如果您使用「虛擬成員管理程式」(Virtual Member Manager, VMM)，請執行第 169 頁的『配置虛擬成員管理程式人員目錄提供者』。
  6. 選擇性的: 如果您已配置 VMM，但想要使用人員代理，請執行第 175 頁的『配置人員替代』。
  7. 選擇性的: 如果您要使用群組工作項目，請使用管理主控台來啓用它們。按一下**伺服器** → **應用程式伺服器** → **server\_name** 或**伺服器** → **叢集** → **cluster\_name** (如果「商業程序編排器」是配置在叢集上)，然後在**商業整合**下，展開**商業程序編排器儲存器**，按一下**人力作業管理程式**，並選取**啓用群組工作項目**。

已配置「商業程序編排器」。

---

## 使用管理主控台的部署環境精靈來配置商業程序編排器

使用管理主控台的部署環境精靈，您可以建立一個包括「商業程序編排器」的型樣配置。如果「商業程序編排器」配置有它自己的資料庫，則此配置適用於正式作業系統。

您已執行第 86 頁的『規劃使用管理主控台的部署環境精靈』。

1. 啓動部署環境精靈。在管理主控台中，按一下**伺服器** → **部署環境** → **新建**。當您輸入其他配置參數時，請確定您輸入自己在第 86 頁的『規劃使用管理主控台的部署環境精靈』中規劃的值：
  - a. 您可以根據下列任何型樣配置「商業程序編排器」：
    - 遠端傳訊及遠端支援
    - 遠端傳訊
    - 單一叢集
    - 自訂
  - b. 在安全頁面上，您可以設定使用者名稱和密碼，用作「商業程序編排器」(識別為 WBI\_BPC 元件) 的鑑別別名。
  - c. 在資料庫頁面上，如果您想要對「商業程序編排器」、「商業程序編排器瀏覽器」或「商業程序編排器」傳訊引擎使用不同的資料庫，請將預設資料來源從預設值變更為您所規劃的值。
  - d. 在「商業程序編排器」頁面上，指定您爲此配置規劃的環境定義根目錄、安全參數和郵件階段作業參數。
2. 如果您爲「商業程序編排器」指定不同的資料庫，請執行第 159 頁的『使用 SQL Scripts 來建立商業程序編排器的資料庫』。否則，若爲非 Derby 資料庫，請確定有空的資料庫，這樣「商業程序編排器」就可以在第一次存取該資料庫時在該資料庫中建立預設綱目。

3. 如果您為「商業程序編排器觀察程式」指定不同的資料庫，請執行第 187 頁的『準備商業程序編排器觀察程式的資料庫』。否則，若為非 Derby 資料庫，請確定空的資料庫存在，這樣「商業程序編排器」就可以在第一次存取該資料庫時在該資料庫中建立預設綱目。
4. 如果您為「商業程序編排器」傳訊引擎指定不同的資料庫，請確定該資料庫存在。
  - 如果您想要使用**建立表格**選項，讓傳訊引擎第一次使用該資料庫時建立預設綱目，請授與資料庫使用者 ID 在您所規劃使用的綱目中建立表格和視圖的權限。
  - 否則，如果您**不使用建立表格**選項，則在預設傳訊提供者嘗試存取該資料庫之前，請先建立表格。您可以使用 `install_root` 目錄的 `bin` 子目錄中的 `sibDDLGenerator` 公用程式，產生可用來建立表格的 DDL 檔案。
5. 對於每一個已配置「商業程序編排器」的節點，請確定已設定 JDBC 驅動程式的環境變數。在叢集上，您必須為每一個管理叢集成員的節點執行此動作。
  - a. 按一下**環境** → **WebSphere 變數**，對於**範圍**，請選取已配置「商業程序編排器」的節點。
  - b. 選取 JDBC 提供者的環境變數：
    - 若為 Derby，您不需要設定任何環境變數。
    - 若為使用 CLI 驅動程式的 DB2 on Linux、UNIX、Windows 或 z/OS，請選取 `DB2_JDBC_DRIVER_PATH`。
    - 若為使用 Universal 驅動程式的 DB2 on Linux、UNIX、Windows 或 z/OS，請選取 `DB2UNIVERSAL_JDBC_DRIVER_PATH`。
    - 若為使用原生驅動程式的 DB2 on i5/OS，請選取 `OS400_NATIVE_JDBC_DRIVER_PATH`。
    - 若為使用工具箱驅動程式的 DB2 on i5/OS，請選取 `OS400_TOOLBOX_JDBC_DRIVER_PATH`。
    - 若為 Oracle，請選取 `ORACLE_JDBC_DRIVER_PATH`。
    - 若為 Informix，請選取 `INFORMIX_JDBC_DRIVER_PATH`。
    - 若為使用 WebSphere 內嵌式 ConnectJDBC 驅動程式的 SQL Server，則您不需要設定任何環境變數。
    - 若為使用 DataDirect ConnectJDBC 第四類型驅動程式的 SQL Server，請選取 `CONNECTJDBC_JDBC_DRIVER_PATH`。
  - c. 設定環境變數，以指向 JDBC 驅動程式的 JAR 檔的位置。
6. 啟動「商業程序編排器」：執行第 234 頁的『啟動商業程序編排器』。
7. 選擇性的: 驗證基本「商業程序編排器」配置是否可運作：執行第 235 頁的『驗證商業程序編排器是否運作』。
8. 選擇性的: 變更「人力作業管理程式」的設定：
  - 如果您要變更任何「人力作業管理程式」設定以呈報電子郵件，例如寄件者位址或「商業程序編排器瀏覽器」的 URL 字首，請按一下**伺服器** → **應用程式伺服器** → **server\_name** 或 **伺服器** → **叢集** → **cluster\_name** (如果「商業程序編排器」是配置在叢集上)，然後在**商業整合**下，展開**商業程序編排器儲存器**，再按一下**人力作業管理程式**，並進行變更。
  - 如果您想要變更電子郵件伺服器位址或埠號、電子郵件伺服器的使用者 ID 或密碼，請按一下**資源** → **郵件** → **郵件階段作業**，選取 **Cell** 範圍，然後按一下 **HTM 郵件階段作業**，並進行變更。

9. 視您用於「人員指派」的人員目錄提供者類型而定，您可能需要配置人員目錄提供者：
  - 系統和使用者登錄人員目錄提供者可以直接使用，而不需要配置。
  - 如果您使用「輕量型目錄存取通訊協定 (LDAP)」，請執行第 170 頁的『配置 LDAP 人員目錄提供者』。
  - 如果您使用「虛擬成員管理程式」(Virtual Member Manager, VMM)，請執行第 169 頁的『配置虛擬成員管理程式人員目錄提供者』。
10. 選擇性的: 如果您已配置 VMM，但想要使用人員代理，請執行第 175 頁的『配置人員替代』。
11. 選擇性的: 如果您要使用群組工作項目，請使用管理主控台來啓用它們。按一下**伺服器** → **應用程式伺服器** → **server\_name** 或 **伺服器** → **叢集** → **cluster\_name** (如果「商業程序編排器」是配置在叢集上)，然後在**商業整合**下，展開**商業程序編排器儲存器**，按一下**人力作業管理程式**，並選取**啓用群組工作項目**。
12. 如果您已啓用應用程式安全且擁有呼叫遠端 EJB 方法的長時間執行程序，請確定 Common Secure Interoperability 第 2 版 (CSIv2) 入埠鑑別配置已啓用 CSIv2 身分主張。

「商業程序編排器」已配置給您選取的部署環境。

---

## 使用管理主控台的商業程序編排器配置頁面

說明如何使用管理主控台的「商業程序編排器」配置頁面，在給定的伺服器或叢集上建立配置。

您必須配置必要的資源並安裝「商業程序編排器」應用程式，才能執行包含商業程序或人力作業的應用程式。

1. 如果您在建立預設設定檔時選取「商業程序編排器」配置範例選項，則「商業流程管理程式」、「人力作業管理程式」、「商業程序編排器瀏覽器」及「商業程序編排器觀察程式」都已配置。

若要檢查是否已配置，您可以在管理主控台查看名稱開頭如下的企業應用程式：

- BPCObserver
- BPCECollector
- BPEContainer
- BPCEplorer
- TaskContainer

此配置範例使用 Derby 資料庫，不適用於正式作業系統。因為在部署目標上只能有一個「商業程序編排器」配置，您必須先移除配置範例，如第 237 頁的第 5 章，『移除商業程序編排器配置』所述，才能繼續配置「商業程序編排器」。

2. 如果您有網路部署環境，請確定已配置「服務元件架構 (SCA)」：
  - a. 如果您想要在伺服器上配置「商業程序編排器」，請按一下**伺服器** → **應用程式伺服器** → **serverName**，然後在**商業整合**區段中，按一下**服務元件架構**。
  - b. 如果您想要在叢集上配置「商業程序編排器」，請按一下**伺服器** → **叢集** → **clusterName**，然後在**商業整合**區段中，按一下**服務元件架構**。
  - c. 如果它尚未啓用，請選取**支援服務元件架構元件**。



3. 針對「商業程序編排器」傳訊引擎的資料儲存庫建立資料庫：
  - 如果您想要使用「商業程序編排器」配置頁面上的**建立表格**選項，讓傳訊引擎第一次使用此資料庫時建立預設綱目，請執行下列動作：
    - a. 如果資料庫不存在，請建立它。
    - b. 授與該資料庫使用者 ID 在您規劃使用的綱目中建立表格和視圖的權限。
  - 否則，如果您**不使用建立表格**選項，則在預設傳訊提供者嘗試存取該資料庫之前，請先建立表格。您可以使用 *install\_root* 目錄的 *bin* 子目錄中的 *sibDDLGenerator* 公用程式，產生可用來建立表格的 DDL 檔案。
4. 如果您使用 WebSphere MQ「Java 訊息服務 (JMS)」提供者，則必須建立佇列管理程式和佇列：執行第 178 頁的『建立商業程序編排器的佇列管理程式及佇列』。如果您使用預設的傳訊 JMS 提供者，請勿執行此步驟。
5. 建立「商業程序編排器」的資料庫：
  - 如果您想要使用「商業程序編排器」配置頁面上的**建立表格**選項，讓「商業程序編排器」在第一次使用此資料庫時建立預設綱目，請執行下列動作：
    - a. 如果資料庫不存在，請建立它。
    - b. 授與該資料庫使用者 ID 在您規劃使用的綱目中建立表格和視圖的權限。
  - 否則，如果您**不使用建立表格**選項，則執行第 159 頁的『使用 SQL Scripts 來建立商業程序編排器的資料庫』。
6. 對於每一個已配置「商業程序編排器」的節點，請確定已設定 JDBC 驅動程式的環境變數。在叢集上，您必須為每一個管理叢集成員的節點執行此動作。
  - a. 按一下**環境** → **WebSphere 變數**，對於**範圍**，請選取已配置「商業程序編排器」的節點。
  - b. 選取 JDBC 提供者的環境變數：
    - 若為 Derby，您不需要設定任何環境變數。
    - 若為使用 CLI 驅動程式的 DB2 on Linux、UNIX、Windows 或 z/OS，請選取 `DB2_JDBC_DRIVER_PATH`。
    - 若為使用 Universal 驅動程式的 DB2 on Linux、UNIX、Windows 或 z/OS，請選取 `DB2UNIVERSAL_JDBC_DRIVER_PATH`。
    - 若為使用原生驅動程式的 DB2 on i5/OS，請選取 `OS400_NATIVE_JDBC_DRIVER_PATH`。
    - 若為使用工具箱驅動程式的 DB2 on i5/OS，請選取 `OS400_TOOLBOX_JDBC_DRIVER_PATH`。
    - 若為 Oracle，請選取 `ORACLE_JDBC_DRIVER_PATH`。
    - 若為 Informix，請選取 `INFORMIX_JDBC_DRIVER_PATH`。
    - 若為使用 WebSphere 內嵌式 ConnectJDBC 驅動程式的 SQL Server，則您不需要設定任何環境變數。
    - 若為使用 DataDirect ConnectJDBC 第四類型驅動程式的 SQL Server，請選取 `CONNECTJDBC_JDBC_DRIVER_PATH`。
  - c. 設定環境變數，以指向 JDBC 驅動程式的 JAR 檔的位置。
7. 在管理主控台上，選取您要配置「商業程序編排器」的伺服器或叢集。按下列其中一項：
  - **伺服器** → **應用程式伺服器** → **serverName**



- 伺服器 → 叢集 → *clusterName*

其中 *serverName* 或 *clusterName* 是伺服器或叢集的名稱。

8. 跳至「商業程序編排器」配置頁面：在**儲存器設定**區段中，展開**商業程序編排器儲存器設定**，再按一下**商業程序編排器儲存器**。
9. 驗證是否未配置「商業程序編排器」。應該會出現訊息，指出目前未安裝「商業程序編排器」儲存器（「商業流程管理程式」及「人力作業管理程式」）。

如果已安裝「商業流程管理程式」及「人力作業管理程式」，在繼續下一步之前，請執行第 237 頁的第 5 章，『**移除商業程序編排器配置**』。

10. 對於此伺服器或叢集上的「商業程序編排器」配置，輸入您規劃的值並選取您規劃的選項。如需相關資訊，請參閱第 113 頁的『**「商業程序編排器」配置**』，它分成下列區段：
  - a. 第 114 頁的『**資料來源**』
  - b. 第 115 頁的『**人力作業管理程式郵件階段作業**』
  - c. 第 116 頁的『**安全**』
  - d. 第 118 頁的『**狀態觀察程式**』
  - e. 第 118 頁的『**SCA 連結**』
  - f. 第 119 頁的『**匯流排**』
11. Click **Apply**. Information is displayed reporting the progress deploying and configuring Business Process Choreographer.
12. 如果安裝順利完成，請按一下**儲存主要配置**，然後按一下**儲存**。否則，請捨棄變更，檢查管理主控台及「部署管理程式」或伺服器上的 SystemOut.log 檔案，查看是否有任何錯誤訊息可協助您更正問題，然後重試。
13. 啟動「商業程序編排器」：執行第 234 頁的『**啟動商業程序編排器**』。
14. 選擇性的: 驗證基本「商業程序編排器」配置是否可運作：執行第 235 頁的『**驗證商業程序編排器是否運作**』。
15. 選擇性的: 變更「人力作業管理程式」的設定：
  - 如果您要變更任何「人力作業管理程式」設定以呈報電子郵件，例如寄件者位址或「商業程序編排器瀏覽器」的 URL 字首，請按一下**伺服器 → 應用程式伺服器 → server\_name** 或 **伺服器 → 叢集 → cluster\_name** (如果「商業程序編排器」是配置在叢集上)，然後在**商業整合**下，展開**商業程序編排器儲存器**，再按一下**人力作業管理程式**，並進行變更。
  - 如果您想要變更電子郵件伺服器位址或埠號、電子郵件伺服器的使用者 ID 或密碼，請按一下**資源 → 郵件 → 郵件階段作業**，選取 **Cell** 範圍，然後按一下 **HTM 郵件階段作業**，並進行變更。
16. 視您用於「人員指派」的人員目錄提供者類型而定，您可能需要配置人員目錄提供者：
  - 系統和使用者登錄人員目錄提供者可以直接使用，而不需要配置。
  - 如果您使用「輕量型目錄存取通訊協定 (LDAP)」，請執行第 170 頁的『**配置 LDAP 人員目錄提供者**』。
  - 如果您使用「虛擬成員管理程式 (Virtual Member Manager, VMM)」，請執行第 169 頁的『**配置虛擬成員管理程式人員目錄提供者**』。
17. 選擇性的: 如果您已配置 VMM，但想要使用人員代理，請執行第 175 頁的『**配置人員替代**』。

18. 選擇性的: 如果您要使用群組工作項目, 請使用管理主控台來啓用它們。按一下**伺服器** → **應用程式伺服器** → **server\_name** 或**伺服器** → **叢集** → **cluster\_name** (如果「商業程序編排器」是配置在叢集上), 然後在**商業整合**下, 展開**商業程序編排器儲存器**, 按一下**人力作業管理程式**, 並選取**啓用群組工作項目**。
19. 如果您已啓用應用程式安全且擁有呼叫遠端 EJB 方法的長時間執行程序, 請確定 Common Secure Interoperability 第 2 版 (CSIV2) 入埠鑑別配置已啓用 CSIV2 身分主張。
20. 選擇性的: 如果您尚未安裝和配置「商業程序編排器瀏覽器」, 現在可以開始配置。執行第 183 頁的『配置商業程序編排器瀏覽器』。
21. 選擇性的: 如果您尚未安裝和配置「商業程序編排器觀察程式」, 現在可以開始配置。執行第 186 頁的『配置商業程序編排器觀察程式』。

已配置「商業程序編排器」。

## 「商業程序編排器」配置

請利用這個畫面來安裝和配置「商業程序編排器」。

若要檢視此管理主控台頁面, 請按一下**伺服器** → **叢集** → **cluster\_name** 或**伺服器** → **應用程式伺服器** → **server\_name**, 然後按一下**儲存器**下的**商業程序編排器儲存器**。

此頁分成數個區段。如需每一個區段中的欄位詳細資訊, 請參閱:

- 第 114 頁的『資料來源』
- 第 115 頁的『人力作業管理程式郵件階段作業』
- 第 116 頁的『安全』
- 第 118 頁的『狀態觀察程式』
- 第 118 頁的『SCA 連結』
- 第 119 頁的『匯流排』

### 資料來源

在此章節中, 您可以指定「商業程序編排器」的資料來源。

### 編輯

按一下此按鈕來編輯資料來源。

### 測試連線

測試到資料來源的連線。

### 資料庫實例

「商業流程管理程式」與「人力作業管理程式」所使用的資料庫名稱。

內容	值
資料類型	字串
預設值	WPRCSDB

### 綱目名稱

要使用的綱目名稱。

如果您要使用自己的綱目而非預設綱目，則只需要變更綱目名稱。

內容	值
資料類型	字串

## 建立表格

如果已選取，則在第一次存取資料庫時就會自動建立表格。若要此選擇生效，資料庫必須已經存在，且所提供的使用者名稱必須具有在資料庫中建立表格及索引的權限。

在正式作業系統中，不建議使用此選項。如果未選取此選項，則不會自動建立表格，您必須執行 Script 來手動建立表格。

內容	值
資料類型	勾選框
預設值	已選取

## 使用者名稱

具有權限連接到資料庫及修改資料的使用者 ID。

如果使用者 ID 具有在資料庫中建立表格及索引的權限，則可以使用自動建立表格的選項，並且在套用服務程式或修正套件後，視需要自動更新資料庫綱目。

內容	值
資料類型	字串

## 密碼

資料來源使用者 ID 的密碼。

內容	值
資料類型	字串

## 伺服器

資料庫伺服器的位址。

請指定主機名稱或 IP 位址，以及埠號。

內容	值
資料類型	字串
範例	localhost:50000

## 提供者

「商業程序編排器」的 JDBC 提供者。

內容	值
資料類型	下拉清單

## 人力作業管理程式郵件階段作業

在此章節中，您可以指定呈報電子郵件的參數。

## 啓用電子郵件服務

如果「人力作業管理程式」要傳送呈報電子郵件通知，則您必須啓用郵件階段作業。

內容	值
資料類型	勾選框
預設值	已選取

## 郵件傳輸主機

「簡易郵件傳送通訊協定 (SMTP)」電子郵件服務所在的主機名稱或 IP 位址。

內容	值
資料類型	字串

## 郵件傳輸使用者

電子郵件服務的使用者 ID。

如果郵件伺服器不需要鑑別，您可以將這個欄位保留空白。

內容	值
資料類型	字串

## 郵件傳輸密碼

郵件傳輸使用者 ID 的密碼。

如果郵件伺服器不需要鑑別，您可以將這個欄位保留空白。

內容	值
資料類型	字串

## 商業程序編排器瀏覽器 URL

指定用於電子郵件鏈結到「商業程序編排器瀏覽器」的 URL。

這個 URL 是用來提供所產生的電子郵件的鏈結，因此接收到電子郵件通知的商業管理者可以按一下鏈結，並在他們的 Web 瀏覽器中檢視相關的商業程序或人力作業。

內容	值
資料類型	字串
範例	<a href="http://www.ibm.com:9080/bpc">http://www.ibm.com:9080/bpc</a>

## 安全

在此章節中，您可以指定功能角色與使用者 ID 及群組的對映，以及對「商業程序編排器」而言必要的鑑別認證。

## 管理者使用者

管理者安全角色對映到指定的使用者 ID。

內容	值
資料類型	字串

內容	值
預設值	目前登入的使用者

### 管理者群組

管理者安全角色對映到指定的群組。

內容	值
資料類型	字串
預設值	無

### 監視者使用者

系統監視者安全角色對映到指定的使用者 ID。

內容	值
資料類型	字串
預設值	目前登入的使用者

### 監視者群組

系統監視者安全角色對映到指定的群組。

內容	值
資料類型	字串
預設值	無

### JMS 鑑別使用者

系統整合匯流排的鑑別別名。

內容	值
資料類型	字串
預設值	目前登入的使用者

### JMS 鑑別密碼及確認密碼

JMS 鑑別使用者 ID 的密碼。

內容	值
資料類型	字串
預設值	無

### JMS API 鑑別使用者

商業流程管理程式訊息驅動 Bean 的執行身分使用者 ID。

內容	值
資料類型	字串
預設值	目前登入的使用者

### JMS API 鑑別密碼及確認密碼

JMS API 鑑別使用者 ID 的密碼。

內容	值
資料類型	字串
預設值	無

### 呈報使用者鑑別使用者

人力作業管理程式訊息驅動 Bean 的執行身分使用者 ID。

內容	值
資料類型	字串
預設值	目前登入的使用者

### 呈報使用者鑑別密碼及確認密碼

呈報使用者鑑別使用者 ID 的密碼。

內容	值
資料類型	字串
預設值	無

### 狀態觀察程式

在此章節中，您可以針對「商業流程管理程式」及「人力作業管理程式」啟用審核日誌記載及「共用事件基礎架構 (CEI)」記載。

#### 商業流程管理程式的審核日誌記載

選取時，即會啟用「商業流程管理程式」的審核日誌記載。

內容	值
資料類型	勾選框
預設值	已選取

#### 人力作業管理程式的審核日誌記載

選取時，即會啟用「人力作業管理程式」的審核日誌記載。

內容	值
資料類型	勾選框
預設值	已選取

#### 商業流程管理程式的共用事件基礎架構記載

選取時，即會啟用「商業流程管理程式」的「共用事件基礎架構」記載。

內容	值
資料類型	勾選框
預設值	已選取

#### 人力作業管理程式的共用事件基礎架構記載

選取時，即會啟用「人力作業管理程式」的「共用事件基礎架構」記載。



內容	值
資料類型	勾選框
預設值	已選取

## SCA 連結

針對「服務元件架構 (SCA)」連結，您可以設定 Web 服務 API 的環境定義根目錄。

### 主機

這個唯讀欄位說明連結「商業流程管理程式」及「人力作業管理程式」主機的環境定義字首，其附加環境定義根目錄。

### 商業流程管理程式的環境定義根目錄

商業流程管理程式 Web 服務的環境定義根目錄。

內容	值
資料類型	字串
在伺服器上配置時的預設值	<i>/BFMIF_nodeName_serverName</i>
在叢集上配置時的預設值	<i>/BFMIF_clusterName</i>

### 人力作業管理程式的環境定義根目錄

人力作業管理程式 Web 服務的環境定義根目錄。

內容	值
資料類型	字串
在伺服器上配置時的預設值	<i>/HTMIF_nodeName_serverName</i>
在叢集上配置時的預設值	<i>/HTMIF_clusterName</i>

### 相對路徑

這個唯讀的欄位會顯示「商業流程管理程式」及「人力作業管理程式」的 SCA 連結相對路徑。

內容	值
資料類型	唯讀字串
商業流程管理程式的相對路徑	<i>/sca/com/ibm/bpe/spi/sca/BFMWS</i>
人力作業管理程式的相對路徑	<i>/sca/com/ibm/task/spi/sca/HTMWS</i>

### 匯流排

如果您要對「商業程序編排器」傳訊引擎使用與「服務元件架構 (SCA)」的配置不同的資料來源，請展開此部分來變更設定。

### 使用預設配置

若已選取，將會使用 SCA 傳訊引擎的現行配置設定。

若您要使用不同的設定，請清除勾選框，以啓用本節中的其他欄位。

內容	值
資料類型	勾選框
預設值	已選取

## 匯流排成員位置

定義要在本端或遠端儲存傳訊引擎的資料。

請選取**本端**或**遠端**。若您選取**遠端**，就會啓用遠端目的地位置選取元及**新建**按鈕。

內容	值
資料類型	圓鈕
預設值	本端

## 遠端目的地位置

指定遠端傳訊引擎儲存的佈署目標。

若清單是空的，或未包含您要選取的位置，請按一下**新建**。

內容	值
資料類型	下拉清單
預設值	無

## 新建

這個按鈕會開啓「瀏覽」部署目標頁面。

選取部署目標後，就會新增目標至遠端目的地位置的清單。

## 編輯

按一下此按鈕來編輯資料來源。

## 測試連線

測試到資料來源的連線。

## 資料庫實例

資料庫的名稱。

內容	值
資料類型	字串
預設值	<code>\${USER_INSTALL_ROOT}\databases\BPHEME</code>

## 綱目名稱

要使用的綱目名稱。

內容	值
資料類型	字串
預設值	MEDBM00

## 建立表格

如果已選取，則在第一次存取資料庫時就會自動建立表格。若要此選擇生效，資料庫必須已經存在，且所提供的使用者名稱必須具有在資料庫中建立表格及索引的權限。

如果未選取，則不會自動建立表格，您必須執行 `Script` 來手動建立表格。若為正式作業系統，最好不要使用由此選項建立的預設表格。

內容	值
資料類型	勾選框
預設值	已選取

### 使用者名稱

具有權限連接到資料庫及修改資料的使用者 ID。

如果使用者 ID 具有在資料庫中建立表格及索引的權限，則可以使用自動建立表格的選項，並且在套用服務程式或修正套件後，視需要自動更新資料庫綱目。

內容	值
資料類型	字串

### 密碼

資料來源使用者 ID 的密碼。

內容	值
資料類型	字串

### 伺服器

資料庫伺服器的位址。

請指定主機名稱或 IP 位址，以及埠號。

內容	值
資料類型	字串
範例	localhost:50000

### 提供者

「商業程序編排器」傳訊引擎的 JDBC 提供者。

若您已將 SCA 配置為使用檔案儲存庫，這個欄位會設為檔案儲存庫，而資料庫參數的欄位則無法使用。選取資料庫提供者時，可以使用資料庫參數。

內容	值
資料類型	下拉清單
預設值	您針對 SCA 配置的提供者。

---

## 使用 bpeconfig.jacl Script 來配置商業程序編排器

說明如何使用 bpeconfig.jacl Script，在給定的伺服器或叢集上配置「商業程序編排器」及所有必要資源。

1. 請確定您知道要使用哪些選項和參數。請參閱您在第 81 頁的第 3 章，『規劃配置商業程序編排器』中規劃的值。您必須在批次檔或指令行上併入所有必要參數和選項。否則，系統會提示您輸入任何未提供的必要參數。如需這個 Script、其選項及參數的詳細資訊，請參閱第 143 頁的『bpeconfig.jacl Script 檔』。

選項	敘述
如果伺服器 (或在網路部署環境中，部署管理程式) 沒有執行	<p>使用此選項：</p> <p><code>-conntype NONE</code></p> <p>如果伺服器 (或部署管理程式) 在執行中，請勿使用此選項。</p>
如果已啟用管理安全	<p>包括這些參數：</p> <p><code>-user <i>userName</i></code>  <code>-password <i>userPassword</i></code></p>
如果您不使用預設設定檔	<p>包括此參數：</p> <p><code>-profileName <i>profileName</i></code></p>
如果您未在預設伺服器上配置「商業程序編排器」	<p>包括任一參數：</p> <p><code>-cluster <i>clusterName</i></code></p> <p>或兩個參數：</p> <p><code>-node <i>nodeName</i></code>  <code>-server <i>serverName</i></code></p>
因為這個 <b>Script</b> 一定會建立「商業程序編排器」配置	<p>包括「商業流程管理程式」及「人力作業管理程式」的必要參數：</p> <pre>{-adminBFMUsers <i>userList</i>     -adminBFMGroups <i>groupList</i>} {-monitorBFMUsers <i>userList</i>     -monitorBFMGroups <i>groupList</i>} -jmsBFMRunAsUser <i>userID</i> -jmsBFMRunAsPwd <i>password</i> {-adminHTMUsers <i>userList</i>     -adminHTMGroups <i>groupList</i>} {-monitorHTMUsers <i>userList</i>     -monitorHTMGroups <i>groupList</i>} -jmsHTMRunAsUser <i>userID</i> -jmsHTMRunAsPwd <i>password</i> -contextRootBFM <i>contextRootBFM</i> -contextRootHTM <i>contextRootHTM</i></pre> <p>對於結尾是 <i>Users</i> 和 <i>Groups</i> 的參數配對，您必須指定其中一個或兩個參數。開頭為 <i>contextRoot</i> 的兩個參數為選用項目。</p>
如果您想要啟用簡易郵件傳送通訊協定 (SMTP) 伺服器來傳送「呈報電子郵件」	<p>包括此參數：</p> <p><code>-mailServerName <i>mailServerName</i></code></p> <p>如果郵件伺服器需要鑑別，請同時包括這些參數：</p> <p><code>-mailUser <i>mailUserID</i></code>  <code>-mailPwd <i>mailPassword</i></code></p>

選項	敘述
<p>因為您可以讓 <b>Script</b> 檔建立該資料庫，或只讓它產生 <b>SQL Script</b> 而不執行 <b>Script</b></p>	<p>使用此選項</p> <pre data-bbox="971 264 1230 296">-createDB { yes   no }</pre> <p>如果您選取 <b>yes</b>，<b>bpeconfig.jacl</b> Script 將產生及執行 <b>SQL</b> 檔案，在預設表格空間內建立資料庫物件，這並不適用於高效能系統。在此情況下，也要規劃停止伺服器，並使用 <b>-conntype NONE</b> 選項。</p> <p>若您選取 <b>no</b>，且該資料庫並不存在，則您或資料庫管理者必須執行所產生的 <b>SQL Script</b>。若為高效能系統，請指定 <b>no</b>，因為您需要先自訂 <b>SQL Script</b> 才能執行它。如果您沒有權限自行建立該資料庫，則也要指定 <b>no</b>，這樣您就可以提供 <b>SQL Script</b> 給資料庫管理者來自訂及執行。</p> <p>如果您使用的資料庫有限制支援，則也必須指定 <b>no</b>。</p> <p><b>限制：</b> 這個 <b>Script</b> 無法建立下列類型的資料庫：</p> <ul data-bbox="971 932 1312 1083" style="list-style-type: none"> <li>• DB2 for z/OS</li> <li>• Oracle</li> <li>• 遠端 Microsoft SQL Server</li> <li>• 遠端 Informix Dynamic Server</li> </ul> <p>若您選取 <b>yes</b>，並且在連線模式中執行這個 <b>Script</b>，則如果建立資料庫或綱目所花費的時間超過預設逾時值 3 分鐘，就會失敗。</p>

選項	敘述
<p>因為每一個「商業程序編排器」配置都需要資料庫的存取權</p>	<p>包括此參數：</p> <pre>-dbType databaseType</pre> <p>也提供資料庫類型的必要參數 (如需詳細資訊，請參閱第 143 頁的『bpeconfig.jacl Script 檔』)：</p> <pre>-dbVersion version -dbHome databaseInstallPath -dbJava JDBCDriverPath -dbName databaseName -dbUser databaseUser -dbPwd databasePassword -dbAdmin databaseAdministratorUserID -driverType JDBCDriverType -dbTablespaceDir databaseTablespacePath -dbServerName databaseServerName -dbServerPort databaseServerPort -dbStorageGroup DB2zOSSStorageGroup -dbConnectionTarget DB2zOSSSubSystem -dbSchema schemaQualifier -dbInstance InformixInstance</pre> <p>在叢集的批次模式中執行這個 Script 時，如果資料庫需要 <code>-dbJava</code> 參數，請依下列方式，為每一個管理叢集成員的節點指定此參數：</p> <pre>-dbJava.nodeName JDBCDriverPath _on_nodeName</pre> <p>註：如果您使用下列其中一個資料庫，<code>bpeconfig.jacl</code> 也可以建立資料庫實例：</p> <ul style="list-style-type: none"> <li>• 本端 DB2 for Linux、UNIX 或 Windows</li> <li>• DB2 on iSeries</li> <li>• Derby Embedded</li> <li>• Derby 網路資料庫，且伺服器在執行中</li> </ul>
<p>因為每一個「商業程序編排器」配置都使用 JMS 提供者</p>	<p>包括此參數：</p> <pre>-mqType { WPM   MQSeries }</pre> <p>也提供 JMS 提供者的必要參數 (如需詳細資訊，請參閱第 143 頁的『bpeconfig.jacl Script 檔』)。</p> <pre>-createQM { yes   no } -qmNameGet getQueueManagerName -mqClusterName mqClusterName -qmNamePut putQueueManagerName -mqHome MQInstallationDirectory -mqUser JMSProviderUserID -mqPwd JMSProviderPassword</pre> <p>註：MQSeries® 選項已淘汰。</p>



選項	敘述
如果您使用 <code>-mqType WPM</code> 選項，請指定傳訊引擎儲存庫設定。	包括下列參數： <pre>-meStoreType { FILESTORE   DATASTORE } -mqCreateTables { true   false } -mqSchemaName mqSchemaName -mqDataSource datasourceName -medbUser meDatabaseUser -medbPwd meDatabasePassword</pre>
因為這個 <b>Script</b> 一定會配置「商業程序編排器瀏覽器」	包括下列這些選用性參數： <pre>-hostName explorerVirtualHostname -explorerHost explorerURL -remoteNode nodeName -remoteServer serverName -remoteCluster clusterName -contextRootExplorer explorerContextRoot -maxListEntries maximum</pre> <p>如需這些參數的相關資訊，包括預設值在內，請參閱第 183 頁的『使用 <code>clientconfig.jacl</code> Script 檔配置商業程序編排器瀏覽器』。</p>
知道您想要在部署目標上安裝及配置「商業程序編排器觀察程式」或事件收集程式應用程式	使用這些選項： <pre>-createEventCollector { yes   no } -createObserver { yes   no }</pre> <p>這些選項只能在批次模式中執行 <code>bpeconfig.jacl</code> 時使用，並不適用於高效能系統。若為正式作業系統，請執行第 186 頁的『配置商業程序編排器觀察程式』。</p>

- 如果您在建立預設設定檔時選取「商業程序編排器」配置範例選項，則「商業流程管理程式」、「人力作業管理程式」、「商業程序編排器瀏覽器」及「商業程序編排器觀察程式」都已配置。

若要檢查是否已配置，您可以在管理主控台查看名稱開頭如下的企業應用程式：

- BPCObserver
- BPCECollector
- BPEContainer
- BPCEplorer
- TaskContainer

此配置範例使用 Derby 資料庫，不適用於正式作業系統。因為在部署目標上只能有一個「商業程序編排器」配置，您必須先移除配置範例，如第 237 頁的第 5 章，『移除商業程序編排器配置』所述，才能繼續配置「商業程序編排器」。

- 如果您有網路部署環境，請確定已配置「服務元件架構 (SCA)」：
  - 如果您想要在伺服器上配置「商業程序編排器」，請按一下 **伺服器** → **應用程式伺服器** → **serverName**，然後在**商業整合**區段中，按一下**服務元件架構**。
  - 如果您想要在叢集上配置「商業程序編排器」，請按一下 **伺服器** → **叢集** → **clusterName**，然後在**商業整合**區段中，按一下**服務元件架構**。
  - 如果它尚未啟用，請選取**支援服務元件架構元件**。

4. 如果您使用 WebSphere Platform Messaging (WPM) 作為 JMS 提供者，且不使用 `-meStoreType DATASTORE` (含 Derby Embedded 資料庫) 或 `-meStoreType FILESTORE` 選項，然後為「商業程序編排器」傳訊引擎的資料儲存庫建立資料庫：
    - 如果您想要使用 `-mqCreateTables yes` 選項，讓傳訊引擎第一次使用此資料庫時建立預設綱目，請執行下列動作：
      - a. 如果資料庫不存在，請建立它。
      - b. 授與該資料庫使用者 ID 在您規劃使用的綱目中建立表格和視圖的權限。
    - 否則，如果您要使用 `-mqCreateTables no` 選項，則在預設傳訊提供者嘗試存取該資料庫之前，請先建立表格。您可以使用 `install_root` 目錄的 `bin` 子目錄中的 `sibDDLGenerator` 公用程式，產生可用來建立表格的 DDL 檔案。
  5. 若您規劃使用 `-createDB yes` 選項執行所產生的 SQL Script 來建立資料庫綱目：
    - a. 如果您使用下列其中一個資料庫：
      - DB2 for z/OS
      - Oracle
      - 遠端 Microsoft SQL Server
      - 遠端 Informix Dynamic Server，如果資料庫不存在，請根據資料庫文件，手動建立空的資料庫。
    - b. 請確定資料庫用戶端 (例如 `db2.exe`) 位於 Scripting 用戶端的路徑上。
    - c. 請確定應用程式伺服器已停止。
  6. 呼叫 `bpeconfig.jacl` Script 檔時，可在提供已規劃選項和配置參數的批次模式中，或在互動模式中。如需 Script 檔的詳細資訊，請參閱第 143 頁的『`bpeconfig.jacl` Script 檔』。
  7. 如果您使用 `-createDB no` 選項來延遲建立資料庫，或 `bpeconfig.jacl` Script 建立資料庫失敗，則您或資料庫管理者應該執行第 156 頁的『使用已產生的 SQL Script 來建立商業程序編排器的資料庫綱目』中所述的動作，然後才能在步驟 8 中啟動「商業程序編排器」。
- 註：**如果您使用本端資料庫，且在步驟 8 中啟動「商業程序編排器」時它會存在，但您並未執行第 156 頁的『使用已產生的 SQL Script 來建立商業程序編排器的資料庫綱目』中所述的動作，則「商業程序編排器」第一次存取該資料庫時，將會建立預設綱目。
8. 啟動「商業程序編排器」：執行第 234 頁的『啟動商業程序編排器』。
  9. 選擇性的: 驗證基本「商業程序編排器」配置是否可運作：執行第 235 頁的『驗證商業程序編排器是否運作』。
  10. 選擇性的: 如果您想要變更 JMS 鑑別使用者 ID、執行身分使用者 ID 或角色與使用者和群組的對映，請按一下 **安全** → **商業整合安全**，來變更安全設定。
  11. 選擇性的: 變更「人力作業管理程式」的設定：
    - 如果您要變更任何「人力作業管理程式」設定以呈報電子郵件，例如寄件者位址或「商業程序編排器瀏覽器」的 URL 字首，請按一下 **伺服器** → **應用程式伺服器** → **server\_name** 或 **伺服器** → **叢集** → **cluster\_name** (如果「商業程序編排器」是配置在叢集上)，然後在 **商業整合** 下，展開 **商業程序編排器儲存器**，再按一下 **人力作業管理程式**，並進行變更。

- 如果您想要變更電子郵件伺服器位址或埠號、電子郵件伺服器的使用者 ID 或密碼，請按一下**資源** → **郵件** → **郵件階段作業**，選取 **Cell** 範圍，然後按一下 **HTM 郵件階段作業**，並進行變更。
12. 視您用於「人員指派」的人員目錄提供者類型而定，您可能需要配置人員目錄提供者：
    - 系統和使用者登錄人員目錄提供者可以直接使用，而不需要配置。
    - 如果您使用「輕量型目錄存取通訊協定 (LDAP)」，請執行第 170 頁的『配置 LDAP 人員目錄提供者』。
    - 如果您使用「虛擬成員管理程式」(Virtual Member Manager, VMM)，請執行第 169 頁的『配置虛擬成員管理程式人員目錄提供者』。
  13. 選擇性的: 如果您已配置 VMM，但想要使用人員代理，請執行第 175 頁的『配置人員替代』。
  14. 選擇性的: 如果您要使用群組工作項目，請使用管理主控台來啓用它們。按一下**伺服器** → **應用程式伺服器** → **server\_name** 或**伺服器** → **叢集** → **cluster\_name** (如果「商業程序編排器」是配置在叢集上)，然後在**商業整合**下，展開**商業程序編排器儲存器**，按一下**人力作業管理程式**，並選取**啓用群組工作項目**。
  15. 如果您已啓用應用程式安全且擁有呼叫遠端 EJB 方法的長時間執行程序，請確定 Common Secure Interoperability 第 2 版 (CSIv2) 入埠鑑別配置已啓用 CSIv2 身分主張。
  16. 選擇性的: 如果您尚未安裝和配置「商業程序編排器瀏覽器」，現在可以開始配置。執行第 183 頁的『配置商業程序編排器瀏覽器』。
  17. 選擇性的: 如果您尚未安裝和配置「商業程序編排器觀察程式」，現在可以開始配置。執行第 186 頁的『配置商業程序編排器觀察程式』。

已配置「商業程序編排器」。

## bpeconfig.jacl Script 檔

此 Script 檔會在伺服器或叢集上配置「商業程序編排器」及所有必要的資源。

### 目的

此 Script 可以使用互動方式或在批次模式下執行。它可以建立本端資料庫及必要的傳訊資源，以及選擇性地配置「商業程序編排器瀏覽器」及「商業程序編排器觀察程式」。

### 位置

bpeconfig.jacl Script 檔位於「商業程序編排器」config 目錄中：

- 在 Linux、UNIX 及 i5/OS 平台上：位於目錄 *install\_root/ProcessChoreographer/config* 中
- 在 Windows 平台上：位於目錄 *install\_root\ProcessChoreographer\config* 中

### 限制

此 Script 具有下列限制：

#### DB2 for z/OS 資料庫

bpeconfig.jacl Script 無法建立 DB2 for z/OS 資料庫。您必須手動建立它。

## DB2 資料庫

如果選取「Universal 驅動程式第四類型」，則即使 DB2 已安裝在本端，`bpeconfig.jacl Script` 還是無法建立資料庫。

## Oracle 資料庫

`bpeconfig.jacl Script` 無法建立 Oracle 資料庫。如果您想要針對「商業程序編排器」使用 Oracle 資料庫，則必須手動建立該資料庫。

## Microsoft SQL Server 資料庫

`bpeconfig.jacl Script` 無法建立遠端資料庫。若要建立本端資料庫，請使用第二類型 JDBC 驅動程式，而且不要指定 `-dbServerName` 參數。如果您想要針對「商業程序編排器」使用遠端 Microsoft SQL Server 資料庫，則必須手動建立該資料庫。

## 在獨立式伺服器環境中執行 Script

在獨立式伺服器環境中：

- 只有當應用程式伺服器不在執行中時，才會併入 `-conntype NONE` 選項。
- 如果伺服器正在執行，而且已啓用 WebSphere 管理安全，則併入 `-user` 及 `-password` 選項。
- 如果您沒有配置預設設定檔，則會新增 `-profileName` 選項。

## 在網路部署環境中執行 Script

在網路部署環境中：

- 在部署管理程式節點上執行 Script。
- 只有當部署管理程式不在執行中時，才會併入 `-conntype NONE` 選項。
- 如果啓用 WebSphere 管理安全，則併入 `-user` 及 `-password` 選項。
- 如果您沒有配置預設設定檔，則會新增 `-profileName` 選項。

## 以非互動方式配置商業程序儲存器、商業程序編排器瀏覽器及商業程序編排器觀察程式

如果您在指令行上提供必要的參數，則不會提示您輸入它們。若要配置「商業程序編排器」，請輸入下列其中一個指令：

在 Linux 及 UNIX 平台上，如果現行目錄是 `install_root`，請輸入指令：  
`bin/wsadmin.sh -f ProcessChoreographer/config/bpeconfig.jacl parameters`

在 i5/OS 平台上，如果現行目錄是 `install_root`，請輸入指令：  
`bin/wsadmin -f ProcessChoreographer/config/bpeconfig.jacl parameters`

在 Windows 平台上，如果您的現行目錄是 `install_root`，請輸入指令：  
`bin\wsadmin -f ProcessChoreographer/config/bpeconfig.jacl parameters`

其中 `parameters` 如下：

```
-conntype NONE
-user userName
-password userPassword
-profileName profileName
{-node nodeName -server serverName}
-cluster clusterName
```

```

-adminBFMUsers userList
-adminBFMGroups groupList
-monitorBFMUsers userList
-monitorBFMGroups groupList
-jmsBFMRunAsUser userID
-jmsBFMRunAsPwd password
-adminHTMUsers userList
-adminHTMGroups groupList
-monitorHTMUsers userList
-monitorHTMGroups groupList
-jmsHTMRunAsUser userID
-jmsHTMRunAsPwd password
-contextRootBFM contextRootBFM
-contextRootHTM contextRootHTM
-mailServerName mailServerName
-mailUser mailUserID
-mailPwd mailPassword
-hostName VirtualHostname
-explorerHost explorerURL
-remoteNode nodeName
-remoteServer serverName
-remoteCluster clusterName
-contextRootExplorer explorerContextRoot
-compileJSPs { yes | no }
-maxListEntries max
-createDB { yes | no }
-dbType databaseType
-dbVersion version
-dbHome databaseInstallPath
-dbJava JDBCDriverPath
-dbName databaseName
-dbUser databaseUser
-dbPwd databasePassword
-driverType JDBCDriverType
-dbTablespaceDir databaseTablespacePath
-dbServerName databaseServerName
-dbServerPort databaseServerPort
-dbStorageGroup DB2zOSSStorageGroup
-dbConnectionTarget DB2zOSSSubSystem
-dbSchema schemaQualifier
-dbInstance InformixInstance
-mqType JMSProviderType
-createQM { yes | no }
-qmNameGet getQueueManagerName
-mqClusterName mqClusterName
-qmNamePut putQueueManagerName
-mqHome MQInstallationDirectory
-mqUser JMSProviderUserID
-mqPwd JMSProviderPassword
-meStoreType { FILESTORE | DATASTORE }
-mqSchemaName mqSchemaName
-mqCreateTables { true | false }
-mqDataSource datasourceName
-medbUser meDatabaseUser
-medbPwd meDatabasePassword
-createEventCollector { yes | no }
-createObserver { yes | no }

```

**註:** 上方參數的一部分是選用參數，視針對其他參數提供的值而定。參數與條件之間的相依關係會決定參數是選用的還是必要的，在下列說明中會針對每個參數說明其相依關係。在指令行上沒有指定任何必要的參數時，會以其列出順序以互動方式提示您。

## 參數

當使用 `wsadmin` 呼叫 Script 時，您可以使用下列參數：

**-conntype NONE**

這指定沒有可用的管理連線。只有在應用程式伺服器 (限獨立式) 或部署管理程式 (限網路部署) 未執行時，才會併入此選項。

**-user *userName***

如果啟用 WebSphere 管理安全，則必須提供使用者 ID 以進行鑑別。

**-password *userPassword***

如果啟用 WebSphere 管理安全，則必須提供使用者 ID *userName* 的密碼。

**-profileName *profileName***

其中 *profileName* 是使用者定義的設定檔名稱。如果您沒有配置預設設定檔，請指定此選項。設定檔必須已存在。

**-node *nodeName***

其中 *nodeName* 是要配置「商業程序編排器」的節點名稱。如果您只有一個節點，且只有一個伺服器，則此參數是選用參數。

**-server *serverName***

其中 *serverName* 是要配置「商業程序編排器」的伺服器名稱。如果您只有一個節點，且只有一個伺服器，則此參數是選用參數。

**-adminBFMUsers *userList***

其中 *userList* 是來自使用者登錄的使用者名稱清單，會將 BPESystemAdministrator Java 2 Enterprise Edition (J2EE) 角色對映到這些使用者。分隔字元是垂直線 (|)。需要有這個內容，才可安裝商業程序儲存器。此參數沒有預設值。必須設定 `adminBFMUsers` 或 `adminBFMGroups` 選項其中一個，或者兩個都設定。

**-adminBFMGroups *groupList***

其中，*groupList* 是來自使用者登錄的群組名稱清單，會將 BPESystemAdministrator J2EE 角色對映到這些群組。分隔字元是垂直線 (|)。需要有這個內容，才可安裝商業程序儲存器。此參數沒有預設值。必須設定 `adminBFMUsers` 或 `adminBFMGroups` 選項其中一個，或者兩個都設定。

**-monitorBFMUsers *userList***

其中，*userList* 是來自使用者登錄的使用者名稱清單，會將 BPESystemMonitor J2EE 角色對映到這些使用者。分隔字元是垂直線 (|)。需要有這個內容，才可安裝商業程序儲存器。此參數沒有預設值。必須設定 `monitorBFMUsers` 或 `monitorBFMGroups` 其中一個，或者兩個都設定。

**-monitorBFMGroups *groupList***

其中，*groupList* 是來自使用者登錄的群組名稱清單，會將 BPESystemMonitor J2EE 角色對映到這些群組。分隔字元是垂直線 (|)。需要有這個內容，才可安裝商業程序儲存器。此參數沒有預設值。必須設定 `monitorBFMUsers` 或 `monitorBFMGroups` 其中一個，或者兩個都設定。

**-jmsBFMRunAsUser *userID***

其中，*userID* 是來自使用者登錄，用於 J2EE 角色 JMSAPIUser 的執行身分使用者 ID。需要有這個內容才能配置商業程序儲存器。此參數沒有預設值。必須設定此項。

**-jmsBFMRunAsPwd *password***

其中，*password* 是 `jmsBFMRunAsUser` 使用者 ID 的密碼。需要有這個內容才能配置商業程序儲存器。此參數沒有預設值。必須設定此項。



**-adminHTMUsers** *userList*

其中，*userList* 是來自使用者登錄的使用者名稱清單，會將 TaskSystemAdministrator Java 2 Enterprise Edition (J2EE) 角色對映到這些使用者。分隔字元是垂直線 (|)。需要有這個內容才能安裝人力作業儲存器。此參數沒有預設值。必須設定 adminHTMUsers 或 adminHTMGroups 選項其中一個，或者兩個都設定。

**-adminHTMGroups** *groupList*

其中，*groupList* 是來自使用者登錄的群組名稱清單，會將 TaskSystemAdministrator J2EE 角色對映到這些群組。分隔字元是垂直線 (|)。需要有這個內容才能安裝人力作業儲存器。此參數沒有預設值。必須設定 adminHTMUsers 或 adminHTMGroups 選項其中一個，或者兩個都設定。

**-monitorHTMUsers** *userList*

其中，*userList* 是來自使用者登錄的使用者名稱清單，會將 TaskSystemMonitor J2EE 角色對映到這些使用者。分隔字元是垂直線 (|)。需要有這個內容才能安裝人力作業儲存器。此參數沒有預設值。必須設定 monitorHTMUsers 或 monitorHTMGroups 其中一個，或者兩個都設定。

**-monitorHTMGroups** *groupList*

其中，*groupList* 是來自使用者登錄的群組名稱清單，會將 TaskSystemMonitor J2EE 角色對映到這些群組。分隔字元是垂直線 (|)。需要有這個內容才能安裝人力作業儲存器。此參數沒有預設值。必須設定 monitorHTMUsers 或 monitorHTMGroups 其中一個，或者兩個都設定。

**-jmsHTMRunAsUser** *userID*

其中，*userID* 是來自使用者登錄，用於 J2EE 角色 EscalationUser 的執行身分使用者 ID。需要有這個內容才能配置人力作業儲存器。此參數沒有預設值。必須設定此項。

**-jmsHTMRunAsPwd** *password*

其中，*password* 是 jmsHTMRunAsUser 使用者 ID 的密碼。需要有這個內容才能配置人力作業儲存器。此參數沒有預設值。必須設定此項。

**-contextRootBFM** *contextRootBFM*

其中，*contextRootBFM* 是「Web 服務端點 URL」的環境定義根目錄。若為「商業流程管理程式 (BFM)」，在伺服器上預設環境定義根目錄是 /BFMIF\_{\$nodeName}\_{\$serverName}。在叢集上，預設值是 /BFMIF\_clusterName。

**-contextRootHTM** *contextRootHTM*

其中，*contextRootHTM* 是「Web 服務端點 URL」的環境定義根目錄。若為「人力作業管理程式 (HTM)」，在伺服器上，預設環境定義根目錄是 /HTMIF\_{\$nodeName}\_{\$serverName}。在叢集上，預設值是 /HTMIF\_clusterName。

**-mailServerName** *mailServerName*

其中，*mailServerName* 是「人力作業管理程式」要用來傳送通知郵件的郵件伺服器主機名稱。當配置郵件階段作業時需要它。如果未將此參數設為空值，則會跳過郵件階段作業配置。預設值是本端主機的完整主機名稱。

**-mailUser** *mailUserID*

其中，*mailUserID* 是要存取郵件伺服器的使用者 ID。只有在郵件伺服器需要鑑別時，才需要此參數。否則，可以略過。建立郵件階段作業時需要此參數，「人力作業管理程式」才能傳送通知郵件。預設值是空白，只有當不需要任何鑑別時才適用。

**-mailPwd** *mailPassword*

其中 *mailPassword* 是使用者 ID *mailUserID* 的密碼。只有在郵件伺服器需要鑑別時，才需要此參數。否則，可以略過。建立郵件階段作業時需要有此參數，「人力作業管理程式」才能傳送通知郵件。

**-hostName** *VirtualHostname*

其中 *VirtualHostname* 是虛擬主機，「商業程序編排器」以及「商業流程管理程式」和「人力作業管理程式」API 的 Web 服務連結會在其上執行。預設值是 `default_host`。

**-maxListEntries** *maximum*

其中 *maximum* 是「商業程序編排器瀏覽器」針對查詢所傳回的結果數上限。預設值是 10000。

**-explorerHost** *explorerURL*

其中，*explorerURL* 是「商業程序編排器瀏覽器」的 URL。如果未針對非叢集環境指定此參數，則會計算預設值，例如 `http://localhost:9080`。「人力作業管理程式」會使用此參數的值來鏈結至此瀏覽器實例。

**-precompileJSPs { no | yes }**

判定 Java Server Pages (JSP) 是否經過前置編譯。預設值是 `no`。請注意，無法除錯經過前置編譯的 JSP。

**-remoteNode** *nodeName*

如果不需要連接至本端「商業程序編排器」配置，請使用此參數及 `remoteServer`。如果未指定此參數，則會預設為 `-node` 參數的值。

**-remoteServer** *serverName*

如果不需要連接至本端「商業程序編排器」配置，請使用此參數及 `remoteNode`。如果未指定此參數，則會預設為 `-server` 參數的值。

**-remoteCluster** *clusterName*

如果不需要連接至本端「商業程序編排器」配置，且未指定 `remoteNode` 及 `remoteServer`，請使用此參數。如果未指定此參數，則會預設為 `-cluster` 參數的值。

**-contextRootExplorer** *contextRootExplorer*

其中，*contextRootExplorer* 是「商業程序編排器瀏覽器」的環境定義根目錄。預設值是 `/bpc`，這會導致預設 URL 為 `http://host:port/bpc`。在 WebSphere Cell 內，環境定義根目錄必須是唯一的。

**-createDB { yes | no }**

可能值是 `yes` 或 `no`。如果設為 `yes`，Script 將建立資料庫。若為 z/OS 資料庫及 Oracle，此 Script 無法建立資料庫，它只能建立表格空間及表格。若為其他資料庫類型，則預設值是 `yes`。若為正式作業系統，請使用 `no`。如果使用 `yes`，則呼叫 `bpeconfig.jacl` 的指令提示必須設定適當的路徑，才能執行對應的資料庫指令 (例如，`db2.exe`)。

**-dbType** *databaseType*

其中，*databaseType* 是資料庫類型。需要此項才能安裝商業程序儲存器，以建立資料庫或資料庫表格，以及來建立資料來源。沒有預設值。可能的值如下：

- Derby
- DB2
- zOS-DB2
- Informix

- iSeries-DB2
- MSSQL
- Oracle

**-dbVersion** *version*

其中，*version* 是資料庫版本號碼。它沒有預設值。只有下列的資料庫類型才需要它：

- 若為 DB2 for z/OS，則 *version* 的值必須為 7、8 或 9。
- 若為 Oracle，*version* 的值必須為 9 或 10。
- 若為 MSSQL，如果資料庫沒有 Unicode 支援，則 *version* 的值必須為 2000，如果資料庫具有 Unicode 支援，則為 2000U。

**-dbHome** *databaseInstallPath*

其中，*databaseInstallPath* 是資料庫系統的安裝目錄。只有在 `createDB` 參數設為 **Yes** 時，Informix 才需要此參數，而 DB2 則選擇性地需要此參數。它是用來建立資料庫或資料庫表格，以及來建立資料來源。預設值以及需求視資料庫及平台而定：

若為 **DB2**：

- 在 Windows 平台上，預設值是 `current_drive\Program Files\IBM\SQLLIB`，其中 *current\_drive* 是現行磁碟機代號。
  - 在 Solaris 平台上，預設值是 `/export/home/${dbUser}/sqllib`。
  - 在其他平台上，預設值是 `/home/${dbUser}/sqllib`。
- 目錄 `${dbHome}/bnd` 及 `${dbHome}/bin` 必須存在。

若為 **Informix**：

- 在 Windows 平台上，預設值是 `current_drive\Program Files\Informix`，其中 *current\_drive* 是現行磁碟機代號。
  - 在 Solaris 及 HP-UX 平台上，預設值是 `/opt/informix`。
  - 在 Linux 及 AIX 平台上，預設值是 `/usr/informix`。
- 檔案 `${dbHome}/jdbc/lib/ifxjdbc.jar` 必須存在。

**-dbJava** *JDBCdriverPath*

其中，*JDBCdriverPath* 是 JDBC 驅動程式所在的目錄。只有下列的資料庫及驅動程式類型組合才需要這個參數：

- DB2 Universal 與第四類型驅動程式。預設值是 `databaseInstallPath/java`。
- DB2 for i5/OS 與第二類型 (原生) 驅動程式。預設值是 `/QIBM/ProdData/Java400/ext`。
- DB2 for i5/OS 與第四類型 (工具箱) 驅動程式。預設值是 `/QIBM/ProdData/HTTP/Public/jt400/lib/java`。
- DB2 for z/OS 與第四類型驅動程式。預設值是 `databaseInstallPath/java`。
- Informix。預設值是 `databaseInstallPath/jdbc/lib`。
- MSSQL DataSource 與 DataDirect 驅動程式類型。沒有預設值。
- Oracle。預設值是 `databaseInstallPath/jdbc/lib`。

其中 *databaseInstallPath* 是資料庫系統的安裝目錄。

在叢集上以批次模式執行 Script 時，如果資料庫需要 `-dbJava` 參數，則請以下列方式針對每個管理叢集成員的節點指定此參數：

`-dbJava.nodeName JDBCPath_on_nodeName`

其中 `JDBCPath` 是 JDBC 驅動程式的路徑，而 `nodeName` 是節點名稱。

**-dbName** *databaseName*

其中，*databaseName* 是「商業程序編排器」資料庫的名稱。它是用來建立資料庫或資料庫表格，以及來建立資料來源。預設值是 `BPEDB`。

- 若為 Oracle，此項是 `TNS`。
- 若為 Derby Network (非 Derby Embedded)，這必須是絕對路徑名稱。
- 若為 i5/OS，則是資料庫名稱或 IASP 硬體裝置名稱。使用「工具箱」JDBC 驅動程式時，預設值是 `*SYSBAS`，而使用「原生」驅動程式時，預設值是 `*LOCAL`。

**-dbUser** *databaseUser*

其中，*databaseUser* 是要存取資料庫的使用者 ID。它是用來建立資料來源。預設值視資料庫及平台而定：

- 若為 Windows 平台上的 DB2：`"db2admin"`
- 若為其他平台上的 DB2：`"db2inst1"`
- 若為 Derby Network：目前登入使用者的使用者 ID
- 若為 Informix：`"informix"`
- 若為 Oracle：`"system"`
- 若為 MSSQL：目前登入使用者的使用者 ID

**-dbPwd** *databasePassword*

其中，*databasePassword* 是使用者 ID *databaseUser* 的密碼。

**-driverType** *JDBCType*

其中，*JDBCType* 是 JDBC 驅動程式的類型。它是用來建立資料來源。

- 若為 DB2，可能的值是 `Universal` 或 `CLI`。
- 若為 i5/OS 上的 DB2：可能值是 `native` 或 `toolbox`。
- 若為 Derby：可能值是 `Embedded` 或 `Network`。
- 若為 Oracle，可能的值是 `oci8` 或 `thin`。
- 若為 MSSQL，可能的值是 `Embedded` 或 `DataDirect`。

**-dbTablespaceDir** *databaseTablespacePath*

其中，*databaseTablespacePath* 是建立資料庫表格空間的目錄。它是用來建立資料庫及資料庫表格。只有下列的資料庫類型才需要這個參數：

- 若為 Oracle，沒有預設值。您必須提供一個值。
- 若為 DB2，預設值是空白，表示沒有建立任何表格空間。

**-dbServerName** *databaseServerName*

其中 *databaseServerName* 是管理「商業程序編排器」資料庫的名稱伺服器。它是用來建立資料來源。

- 若為 DB2，預設值是空白。若為 DB2 UDB，此參數是選用參數，且如果沒有指定它，會針對 DB2 配置第二類型 JDBC 驅動程式，否則將配置第四類型 JDBC 提供者。
- 若為 i5/OS 上的 DB2：指定伺服器簡短名稱。使用「工具箱」JDBC 驅動程式時，預設值是本端主機的簡短名稱。
- 若為所有其他資料庫類型，預設值是本端主機的完整主機名稱。

**-dbServerPort** *databaseServerPort*

其中，*databaseServerPort* 是「商業程序編排器」資料庫伺服器的 TCP/IP 埠。如果指定了 *dbServerName*，則需要這個參數。

- 若為 DB2，預設值是 50000。
- 若為 Derby Network，預設值是 1527。
- 若為 Informix，預設值是 1526。
- 若為 MSSQL，預設值是 1433。
- 若為驅動程式類型為 thin 的 Oracle，預設值是 1521。

**-dbStorageGroup** *DB2zOSSStorageGroup*

其中 *DB2zOSSStorageGroup* 是用來建立「商業程序編排器」資料庫表格的儲存體群組。只有 z/OS 上的 DB2 才需要這個參數。沒有預設值，且不能為空白。

**-dbConnectionTarget** *DB2zOSSubSystem*

其中 *DB2zOSSubSystem* 是用來建立「商業程序編排器」資料庫表格及資料來源的 DB2 連線目標位置。只有 z/OS 上的 DB2 才需要這個參數。預設值是 BPEDB。

**-dbSchema** *schemaQualifier*

若為 i5/OS，則 *schemaQualifier* 是集合名稱，而預設值是 BPEDB。若為其他所有平台，則 *schemaQualifier* 是用來建立「商業程序編排器」資料庫表格及資料來源的綱目限定元。預設值是空白，這表示使用隱含綱目限定元 (視使用的資料庫類型而定)。

**-dbInstance** *InformixInstance*

其中，*InformixInstance* 是「商業程序編排器」Informix 資料庫的實例名稱。預設值是 *ids1*。

**-mqType** *JMSProviderType*

其中，*JMSProviderType* 是要用於「商業程序編排器」的「Java 訊息服務 (JMS)」提供者類型。它是用來建立佇列管理程式及佇列、接聽埠或 *ActivationSpecs*，以及 *Queue Connection Factory*。

其中 *JMSProviderType* 是下列其中一個值：

**WPM** 代表預設傳訊 (WebSphere Platform Messaging)。恆可使用這個選項。

**MQSeries**

代表 WebSphere MQ。此選項要求已安裝產品 WebSphere MQ。使用此值即將淘汰。

**-createQM** { *yes* | *no* }

控制 Script 是否建立本端 WebSphere MQ 佇列管理程式。只有在參數 *mqType* 的值是 *MQSeries* (即將淘汰) 時，此選項才會生效。這個參數的預設值是 *yes*。如果您不想讓 Script 建立 WebSphere MQ 佇列管理程式，例如您想要在與執行 Script 不同的伺服器上建立佇列管理程式時，請使用值 *no*。

**-qmNameGet** *getQueueManagerName*

其中，*getQueueManagerName* 是 GET 要求的佇列管理程式名稱。它是用來建立佇列管理程式及佇列，以及建立接聽埠及 *Queue Connection Factory*。它不得包含 - 字元。*getQueueManagerName* 的預設值是 *BPC\_nodeName\_serverName*。只有在參數 *mqType* 的值是 *MQSeries* (即將淘汰) 時，此選項才會生效。



**-mqClusterName** *mqClusterName*

其中 *mqClusterName* 是佇列管理程式要加入的 WebSphere MQ 叢集的名稱。這是選用參數。預設值是 MQCluster。只有在參數 *mqType* 的值是 MQSeries (即將淘汰) 時，此選項才會生效。

**-qmNamePut** *putQueueManagerName*

其中，*putQueueManagerName* 是 PUT 要求的佇列管理程式名稱。只有當已經設定 *mqClusterName* 參數時才使用它。它是用來建立佇列管理程式及佇列，以及建立接聽埠及 Queue Connection Factory。它不能包含 - 字元，且不能與針對 *qmNameGet* 參數指定的佇列管理程式名稱相同。*putQueueManagerName* 的預設值是 *BPCC\_nodeName\_serverName*。

**-mqHome** *MQInstallationDirectory*

其中，*MQInstallationDirectory* 是 WebSphere MQ 的安裝目錄。這是用來建立佇列管理程式及佇列 (僅限 Windows 平台)，以及用來建立接聽埠及 Queue Connection Factory。如果設定 WebSphere 變數 MQ\_INSTALL\_ROOT，則會使用其值，且不會修改。只有在參數 *mqType* 的值是 MQSeries (即將淘汰) 時，此選項才會生效。

如果沒有設定 MQ\_INSTALL\_ROOT，*MQInstallationDirectory* 使用的預設值視平台而定：

**Windows 平台：**

*current\_drive*\Program Files\IBM\WebSphere MQ

**AIX：** /usr/mqm

**i5/OS：**

/QIBM/ProdData/mqm

**Solaris、HP-UX 及 Linux：**

/opt/mqm

**-mqUser** *JMSProviderUserID*

其中，*JMSProviderUserID* 是要存取 JMS 提供者的使用者 ID。

- 如果 *mqType* 的值是 WPM，則此參數是用來根據「商業程序編排器」SI 匯流排進行鑑別；預設值是目前登入的使用者。
- 如果 *mqType* 的值是 MQSeries，則在 Linux 及 UNIX 平台上，此參數是用來建立佇列管理程式及佇列。*JMSProviderUserID* 的預設值是 *mqm*。

**-mqPwd** *JMSProviderPassword*

其中，*JMSProviderPassword* 是針對 *mqUser* 提供的使用者 ID 的密碼。這個參數沒有預設值。

**-meStoreType** { *FILESTORE* | *DATASTORE* }

設定「商業程序編排器」訊息引擎的訊息儲存庫類型。如果供應 *mqDataSource* 參數，則此參數是設為 *DATASTORE*。如果「服務元件架構 (SCA)」使用 *FILESTORE*，則此參數是設為 *FILESTORE*。Network Deployment 環境不支援 *FILESTORE*。如果未設定 *mqDataSource*，且 SCA 使用 *DATASTORE* 作為其訊息儲存庫類型，則會繼承 SCA 訊息引擎資料庫設定 (例如資料庫類型、JDBC 提供者及資料庫伺服器)。在此情況下，必須設定不同的資料庫綱目 (請參閱下面的 *mqSchemaName*)，也可以改寫 *mqCreateTables* 旗標。

**-mqSchemaName** *mqSchemaName*

其中，*mqSchemaName* 是預設 JMS 提供者傳訊引擎的資料庫綱目名稱。預設值是 *BPME*。只有在 *meStoreType* 設為 *DATASTORE* 時，才會使用此選項。



### **-mqCreateTables** { *true* | *false* }

此布林參數控制預設 JMS 提供者是否在第一次連線時，自動在訊息引擎資料庫中建立其表格。如果「服務元件架構 (SCA)」的這個旗標設為 *true*，則此參數也會設為 *true*。如果 SCA 的這個旗標設為 *false*，則此參數也會設為 *false*。只有在 *mqType* 選項設為 *WPM* 且 *meStoreType* 設為 *DATASTORE* 時，才會使用此選項。

### **-medbUser** *MEDBUserID*

其中 *MEDBUserID* 是用來存取傳訊引擎資料庫的使用者 ID。此參數的預設值是 *dbUser* 參數的值。只有在 *meStoreType* 參數設為 *DATASTORE*，且不是透過 Derby Embedded JDBC 提供者存取傳訊引擎資料庫時，才需要此參數。

### **-medbPwd** *MEDBPassword*

其中 *MEDBPassword* 是針對 *medbUser* 提供的使用者 ID 的密碼。此參數沒有預設值。

### **-createEventCollector** { *yes* | *no* }

當在批次模式中執行時，預設值是 *yes*，這樣會導致配置「商業程序編排器觀察程式」所需的「商業程序編排器」事件收集程式應用程式。使用此選項時，不可以指定不同資料庫，預設會使用 *BPEB* 資料庫，這表示此選項不適用於高效能系統。如果您不想安裝它，請將這個參數的值設為 *no*。

### **-createObserver** { *yes* | *no* }

當在批次模式中執行時，預設值是 *yes*，這樣會導致配置「商業程序編排器觀察程式」應用程式。只有在非互動式模式中才可以使用此選項。使用此選項時，不可以指定不同資料庫，預設會使用 *BPEB* 資料庫，這表示此選項不適用於高效能系統。如果您不想安裝它，請將這個參數的值設為 *no*。

## 以互動方式執行配置 Script

此範例說明執行 *bpeconfig.jacl* Script 來安裝及配置商業程序儲存器，其使用現有的 *DB2* 資料庫、人力作業儲存器及「商業程序編排器瀏覽器」。

**限制:** 以互動方式執行時，此 Script 無法配置「商業程序編排器觀察程式」，也無法配置必要的事件收集程式應用程式。如果您想要使用「商業程序編排器觀察程式」，您必須執行第 186 頁的『配置商業程序編排器觀察程式』。

1. 在伺服器上，或若為網路部署，則在部署管理程式上，啟動 Script：

- 在 Linux 及 UNIX 平台上，輸入指令：

```
install_root/bin/wsadmin.sh
  -f install_root/ProcessChoreographer/config/bpeconfig.jacl
  ( [-user userName] [-password password] | [-conntype NONE])
  [-profileName profileName]
```

- 在 i5/OS 平台上，輸入指令：

```
install_root/bin/wsadmin
  -f install_root/ProcessChoreographer/config/bpeconfig.jacl
  ( [-user userName] [-password password] | [-conntype NONE])
  [-profileName profileName]
```

- 在 Windows 平台上，請輸入指令：

```
install_root\bin\wsadmin.bat
  -f install_root\ProcessChoreographer\config\bpeconfig.jacl
  ( [-user userName] [-password password] | [-conntype NONE])
  [-profileName profileName]
```

2. 以互動方式輸入對顯示問題的回應：

- a. 在網路部署環境中，會提供您可以在其中配置的伺服器或叢集。如果它不是正確的伺服器或叢集，請輸入 **No** 以提供下一個伺服器或叢集。如果它是正確的伺服器或叢集，請輸入 **Yes**。
- b. 若為問題 `Install the business process container?`，請輸入 **Yes**。
- c. 若為問題 `User(s) to add to role BPSystemAdministrator`，請輸入將執行商業程序管理者角色之使用者的使用者 ID。
- d. 若為問題 `Group(s) to add to role BPSystemAdministrator`，請輸入來自網域使用者登錄，且對映到商業程序管理者角色的群組。
- e. 若為問題 `User(s) to add to role BPSystemMonitor`，請輸入將執行商業程序監視者角色之使用者的使用者 ID。
- f. 若為問題 `Group(s) to add to role BPSystemMonitor`，請輸入來自網域使用者登錄，且對映到商業程序監視者角色的群組。
- g. 若為問題 `Run-as UserId for role JMSAPIUser`，請輸入將用於 `JMSAPIUser` 角色的執行身分使用者 ID。
- h. 輸入執行身分使用者 ID 的密碼。
- i. 若為問題 `Use WebSphere default messaging or WebSphere MQ [WPM/MQSeries]?`，請選取您想要使用的 JMS 提供者。
- j. 輸入下列項目：
  - 1) 若為問題 `Virtual Host for the SCA Web Service [default_host]:`，請按 **Enter** 鍵接受「服務元件架構 (SCA)」Web 伺服器虛擬主機的預設值 `default_host`。
  - 2) 若為問題 `Context root for the SCA Web Service [/BFMIF_PNODE_server1]:`，請按 **Enter** 鍵接受預設值 `BFMIF_nodeName_serverName`。
- k. 若為問題 `Create the DataSource for the Process Choreographer database?`，請輸入 **Yes**。
- l. 若為問題 `Create DataSource for a Derby, a DB2, an Informix, an Oracle, or an SQL Server database [Derby/DB2/zOS-DB2/iSeries-DB2/Informix/Oracle/MSSQL]?`，在此範例中，請輸入 **DB2**。選取不同的資料庫會導致其他資料庫特定的問題。
- m. 輸入資料庫名稱。
- n. 在 `Database schema name (may be empty)` 提示上，按 **Enter** 鍵以使用隱含網目限定元。
- o. 若為問題 `Universal or CLI?`，請按 **Enter** 鍵選取預設 `Universal JDBC` 驅動程式。
- p. 若為問題 `DB2 User ID`，請輸入用來存取資料庫的使用者 ID。
- q. 輸入資料庫使用者 ID 的密碼。
- r. 若為問題 `Database server name (may be empty, set to use the type 2 driver)`，請輸入管理資料庫的伺服器名稱。
- s. 若為問題 `Database server port`，請輸入資料庫伺服器埠 (例如，50000)。
- t. 在 `JDBC driver directory on [yourHost]` 提示上，輸入 `DB2 JDBC` 驅動程式 `JAR` 檔案所在的目錄。
- u. 若為問題 `Create the Process Choreographer database objects?`，如果目前登入的使用者 ID 具有足夠的權限可以建立資料庫，且現行環境中已設定 `DB2`

(例如， PATH 上有 'db2' 執行檔)，則可以輸入 **Yes**，否則，如果目前登入的使用者 ID 沒有足夠的權限可以建立資料庫，請輸入 **No**。

如果回答是 **Yes**：

- 1) 若為問題 DB2 tablespace directory (may be empty)，請按 **Enter** 鍵將它保留空白。
- 2) 若為問題 Is 'BPEDB' an existing database (the Process Choreographer schema must not yet exist)，則會根據環境予以提示。
- v. 如果您收到問題 User ID for access to Process Choreographer SI bus，請輸入用來存取預設 JMS 提供者的使用者 ID。
- w. 輸入 SI 匯流排鑑別使用者 ID 的密碼。
- x. For the question Install the task container?, enter **Yes**.
- y. 若為問題 User(s) to add to role TaskSystemAdministrator，請輸入將執行作業管理者角色之使用者的使用者 ID。
- z. 若為問題 Group(s) to add to role TaskSystemAdministrator，請輸入來自網域使用者登錄，且對映到作業管理者角色的群組。
- aa. 若為問題 User(s) to add to role TaskSystemMonitor，請輸入將執行作業監視者角色之使用者的使用者 ID。
- ab. 若為問題 Run-as UserID for role EscalationUser，請輸入呈報使用者角色的執行身分使用者 ID，例如 db2admin。
- ac. 輸入呈報使用者 ID 的密碼。如果使用的使用者 ID 與步驟 第 154 頁的 2g 相同，則會隱藏此提示。
- ad. 若為問題 Context root for the SCA Web Service [/HTMIF\_nodeName\_serverName]:，請輸入「服務元件架構 (SCA)」Web 伺服器的環境定義根目錄，或按 **Enter** 鍵接受預設值。
- ae. 若為問題 Create the mail notification session for the human task manager?，如果您不想為「人力作業管理程式」建立郵件通知階段作業，請輸入 **No**。否則，請輸入 **Yes**，並指定郵件傳輸主機。您可以選擇性地指定使用者 ID 及密碼。
- af. For the question Maximum number of list entries for the Process Choreographer Explorer，press **Enter** to use the default value 10000.
- ag. 若為問題 Context root for the Business Process Choreographer Explorer [/bpc]:，請輸入「商業程序編排器瀏覽器」的環境定義根目錄，或按 **Enter** 鍵使用預設值 /bpc。
- ah. 若為問題 Install the Business Process Choreographer Explorer?，請輸入 **Yes** 安裝「商業程序編排器瀏覽器」，然後針對問題 Precompile JSPs?，如果您想要讓 Java Server Pages (JSP) 經過前置編譯，請輸入 **Yes**，否則請輸入 **No**。若為遠端「商業程序編排器瀏覽器」，針對問題 Node of Process Choreographer to connect to [PNODE]:，請輸入要連接的「商業程序編排器」節點名稱，並針對問題 Server of Process Choreographer to connect to [server1]:，輸入要連接的「商業程序編排器」伺服器名稱，或按 **Enter** 鍵接受預設值。
- ai. Various information is displayed, for example providing the URL of the Business Process Choreographer Explorer. For example:

```
*****
* NOTE: The Process Choreographer URL will be used by the
* Human Task Manager on server server1 of node viennaNode01
* to link to this Explorer instance. Set an empty URL to not create this link.
*****
URL for this Process Choreographer Explorer [http://host_name:9080/bpc]:
```

Enter the URL for this Business Process Choreographer Explorer instance, or press **Enter** to accept the default.

- aj. 會顯示提示，提示何處可以找到可用來配置「商業程序編排器觀察程式」的 Script 檔。

To interactively configure the EventCollector, please use the script `setupEventCollector` located in `install_root\ProcessChoreographer\config`.  
To interactively configure the Observer, please use the script `setupObserver` located in `install_root\ProcessChoreographer\config`.

3. In case of problems, check the log files.

## 日誌檔

如果您在使用 `bpeconfig.jacl` Script 檔建立配置時出現問題，請檢查下列日誌檔：

- `bpeconfig.log`
- `wsadmin.traceout`

兩個檔案都可以在設定檔的 `logs` 目錄中找到：

- 在 Linux、UNIX 及 i5/OS 平台上：在目錄 `profile_root/logs` 中
- 在 Windows 平台上：在目錄 `profile_root\logs` 中

如果是連線模式執行 Script，則也請檢查檔案 `SystemOut.log` 及 `SystemErr.log`，這些檔案可以在根據 `wsadmin` Script 用戶端所連接應用程式伺服器或部署管理程式命名的 `logs` 目錄的子目錄中找到。

### 相關工作

第 137 頁的『使用 `bpeconfig.jacl` Script 來配置商業程序編排器』

說明如何使用 `bpeconfig.jacl` Script，在給定的伺服器或叢集上配置「商業程序編排器」及所有必要資源。

## 使用已產生的 SQL Script 來建立商業程序編排器的資料庫綱目

`bpeconfig.jacl` Script 會產生 SQL Script，來建立「商業程序編排器」的資料庫物件。

您已使用 `bpeconfig.jacl` Script 來配置「商業程序編排器」，並已使用 `-createDB no` 選項來延遲建立資料庫物件，或 `bpeconfig.jacl` Script 無法建立資料庫。

在已產生的 SQL 檔案中，已取代您在配置「商業程序編排器」時提供的所有相關配置參數。您可能想要高效能「商業程序編排器」配置的資料庫，或是資料庫管理者必須為您建立資料庫，或兩者都需要。

1. 在您執行 bpeconfig.jacl Script 的節點上尋找已產生的 SQL Script。

選項	敘述
若為 Linux 及 UNIX	<ul style="list-style-type: none"> <li>如果您已指定綱目字首，則所產生的 Script 為：<i>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/database_schema/createSchema.sql</i>。</li> <li>如果您未指定綱目字首，則所產生的 Script 為：<i>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/createSchema.sql</i>。</li> </ul>
若為 i5/OS	所產生的 Script 為： <i>profile_root/dbscripts/ProcessChoreographer/database_type/collection_name/createSchema.sql</i> 。
若為 Windows	<ul style="list-style-type: none"> <li>如果您已指定綱目字首，則所產生的 Script 為：<i>profile_root\dbscripts\ProcessChoreographer\database_type\database_name\database_schema\createSchema.sql</i></li> <li>如果您未指定綱目字首，則所產生的 Script 為：<i>profile_root\dbscripts\ProcessChoreographer\database_type\database_name\createSchema.sql</i></li> </ul> <p>註：若為 SQL Server，也有一個稱為 createSchemaUnicode.sql 的版本，如果您的資料庫是配置為 Unicode，則應該使用這個版本。</p>
若為 z/OS	<p>有一個稱為 createSchema.sql 的 ASCII SQL Script，以及一個功能相等的 EBCDIC DDL Script，稱為 createSchema.ddl：</p> <ul style="list-style-type: none"> <li>如果您已指定綱目字首，則這兩個檔案都位於 <i>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/database_schema</i></li> <li>如果您未指定綱目字首，則這兩個檔案都位於 <i>profile_root/dbscripts/ProcessChoreographer/database_type/database_name</i></li> </ul>

其中：

*database\_type*

是下列其中一項：

- DB2
- DB2zOSV7
- DB2zOSV8
- Db2iSeries
- Derby
- Informix

- Oracle
- SQLServer

*database\_name*  
是資料庫的名稱。

*database\_schema*  
是綱目的名稱，如果您正在使用某個綱目的話。

*collection\_name*  
是集合的名稱；僅適用於 iSeries 上的 DB2。

2. 如果資料庫尚未存在，而且它不是 Derby 資料庫，請資料庫管理者根據您在第 95 頁的『規劃 BPEDB 資料庫』及第 89 頁的『規劃安全、使用者 ID 及權限』中規劃的值，建立資料庫及使用者 ID。
3. 若是遠端資料庫，請將所產生的 Script 複製到遠端節點。如果您無權執行此動作，請將 Script 副本提供給資料庫管理者，並與她討論您的需求。
4. 您或您的資料庫管理者必須自訂 SQL Script：
  - a. 若為高效能系統，請指定如何配置您在第 95 頁的『規劃 BPEDB 資料庫』的第 96 頁的 5 步驟中規劃的磁碟及表格空間。
5. 使用下列其中一個指令，在資料庫主機上執行 SQL Script：

選項	敘述
若為 Linux、UNIX 或 Windows 上的 DB2	db2 -tf createSchema.sql
若為 iSeries 上的 DB2	db2 -tf createSchema.sql
若為 z/OS 上的 DB2	若為所有 ASCII 版本： db2 -tf createSchema.sql  若為 EBCDIC 版本： db2 -tf createSchema.dll
若為 Derby 資料庫	java -Dij.protocol=jdbc:derby: -Dij.database=BPEDB org.apache.derby.tools.ij createSchema.sql
若為 Informix 資料庫	dbaccess <i>databaseName</i> createSchema.sql
若為 Oracle 資料庫	sqlplus <i>userID/password</i> @ <i>database_name</i> @createSchema.sql
若為 SQL Server 資料庫	若為 ASCII 資料庫： sqlcmd -U <i>userID</i> -P <i>password</i> -d <i>database_name</i> -i createSchema.sql  若為 Unicode 資料庫： sqlcmd -U <i>userID</i> -P <i>password</i> -d <i>database_name</i> -i createSchemaUnicode.sql

6. 配置「Java 資料庫連線功能 (JDBC)」來存取遠端資料庫：請在任一項上執行下列步驟：
  - 在已配置「商業程序編排器」的每一個叢集成員上。



- 在執行「商業程序編排器」但沒有本端資料庫的任何伺服器上。
- 否則，如果伺服器具有本端資料庫，請不要執行下列步驟。
- a. 在管理應用程式伺服器的伺服器上，安裝適當的第二類型資料庫用戶端或第四類型 JDBC 驅動程式。
- b. 如果您是使用第二類型 JDBC 驅動程式，請讓資料庫用戶端可識別新的資料庫。資料庫必須透過別名分類且可存取。如果您是使用第二類型 JDBC 驅動程式，請執行下列動作，讓資料庫用戶端可識別新的資料庫：

**若為 Derby**

此步驟不適用，因為只支援第四類型 JDBC 提供者。

**若為 DB2 Universal Database™**

資料庫必須編目且可透過別名來存取。

**若為 DB2 for iSeries**

資料庫必須編目且可透過別名來存取。

**若為 DB2 for z/OS**

資料庫必須編目且可透過別名來存取。

**若為 Informix Dynamic Server**

此步驟不適用，因為只支援第四類型 JDBC 提供者。

**若為 Microsoft SQL Server**

此步驟不適用，因為只支援第四類型 JDBC 提供者。

**若為 Oracle**

使用 TCP 網路服務名稱 (TNS) 來存取資料庫。

- c. 使用管理主控台，測試資料庫的連線。
  - 1) 按一下**資源** → **JDBC** → **商業整合資料來源**
  - 2) 必要的話，請選取不同範圍並按一下**套用**。

**註：**若為叢集式「商業程序編排器」配置，資料來源定義在叢集層次中。若為非叢集配置，資料來源是在伺服器層次上定義。

- 3) 尋找並選取具有 JNDI 名稱 jdbc/BPEDB 的資料來源。
- 4) 按一下**測試連線**。
- 5) 應該會出現訊息，指出測試連線成功。

「商業程序編排器」資料庫已存在，而且可從任何已配置「商業程序編排器」的遠端伺服器或叢集成員中存取該資料庫。

---

## 使用 SQL Scripts 來建立商業程序編排器的資料庫

建立可在正式作業環境或進階拓撲中使用的資料庫，並使它可在遠端存取。

提供 SQL Script 是為建立及管理所有支援的資料庫系統的資料庫綱目。使用這個方法，「商業程序編排器」會具有自己的資料庫，這對於效能而言很重要。在獨立式伺服器設定中，各個應用程式伺服器上的「商業程序編排器」配置都有專用的資料庫。在叢集式「商業程序編排器」設定中，所有叢集成員共用資料庫。

Script 會指派個別儲存體給不同類型的資料庫物件，例如，範本資料及實例資料。這種分隔可以啓用更好的負載平衡及效能調整。若要達到最好的效能，您可以在執行 Script 之前自訂它們。

1. 在資料庫所在的伺服器上，根據資料庫系統的說明來建立資料庫。

- 第 161 頁的『建立商業程序編排器的 Derby 資料庫』。
  - 第 161 頁的『建立商業程序編排器的 DB2 for i5/OS 資料庫』。
  - 第 162 頁的『建立商業程序編排器的 DB2 for Linux、UNIX 及 Windows 資料庫』。
  - 第 163 頁的『建立商業程序編排器的 DB2 for z/OS 資料庫』。
  - 第 165 頁的『建立商業程序編排器的 Informix Dynamic Server 資料庫』。
  - 第 166 頁的『建立商業程序編排器的 Microsoft SQL Server 資料庫』。
  - 第 167 頁的『建立商業程序編排器的 Oracle 資料庫』。
2. 配置「Java 資料庫連線功能 (JDBC)」來存取遠端資料庫：請在任一項上執行下列步驟：
- 在已配置「商業程序編排器」的每一個叢集成員上。
  - 在執行「商業程序編排器」但沒有本端資料庫的任何伺服器上。
  - 否則，如果伺服器具有本端資料庫，請不要執行下列步驟。
- a. 在管理應用程式伺服器的伺服器上，安裝適當的第二類型資料庫用戶端或第四類型 JDBC 驅動程式。
- b. 如果您是使用第二類型 JDBC 驅動程式，請讓資料庫用戶端可識別新的資料庫。資料庫必須透過別名分類且可存取。如果您是使用第二類型 JDBC 驅動程式，請執行下列動作，讓資料庫用戶端可識別新的資料庫：
- 若為 Derby**  
此步驟不適用，因為只支援第四類型 JDBC 提供者。
- 若為 DB2 Universal Database**  
資料庫必須編目且可透過別名來存取。
- 若為 DB2 for iSeries**  
資料庫必須編目且可透過別名來存取。
- 若為 DB2 for z/OS**  
資料庫必須編目且可透過別名來存取。
- 若為 Informix Dynamic Server**  
此步驟不適用，因為只支援第四類型 JDBC 提供者。
- 若為 Microsoft SQL Server**  
此步驟不適用，因為只支援第四類型 JDBC 提供者。
- 若為 Oracle**  
使用 TCP 網路服務名稱 (TNS) 來存取資料庫。
- c. 使用管理主控台，測試資料庫的連線。
- 1) 按一下**資源** → **JDBC** → **商業整合資料來源**
  - 2) 必要的話，請選取不同範圍並按一下**套用**。
- 註：**若為叢集式「商業程序編排器」配置，資料來源定義在叢集層次中。若為非叢集配置，資料來源是在伺服器層次上定義。
- 3) 尋找並選取具有 JNDI 名稱 jdbc/BPEDB 的資料來源。
  - 4) 按一下**測試連線**。
  - 5) 應該會出現訊息，指出測試連線成功。

「商業程序編排器」資料庫已存在，而且可從任何已配置「商業程序編排器」的遠端伺服器或叢集成員中存取該資料庫。

## 建立商業程序編排器的 Derby 資料庫

使用這項作業來建立「商業程序編排器」的 Derby 資料庫。

Derby 資料庫系統隨附於 WebSphere Process Server。

若要建立 Derby Embedded 資料庫 BPEDB，請執行下列動作：

1. 請執行下列其中一個動作，準備執行資料庫建立 Script 檔：
  - 若準備在預設位置中建立資料庫，請在適當的設定檔目錄中手動建立 `databases` 子目錄。在 Windows 平台上，建立 `profile_root\databases`。在 Linux、UNIX 及 i5/OS 平台上，建立 `profile_root/databases`。切換至新目錄。
  - 若準備建立非預設位置的資料庫位置，請切換至您要建立新資料庫的目錄。
2. 將資料庫建立 Script 複製到您在 1 步驟中建立的目錄。
  - 在 Linux、UNIX 及 i5/OS 平台上，複製檔案 `install_root/dbscripts/ProcessChoreographer/Derby/createDatabase.sql`
  - 在 Windows 平台上，複製檔案 `install_root\dbscripts\ProcessChoreographer\Derby\createDatabase.sql`
3. 檢查伺服器上是否已配置 Java。輸入下列指令：

```
java -version
```

如果收到錯誤訊息，則在步驟 5 中，當您執行資料庫建立 Script 時，必須在 Java 指令前面加上 Java 執行檔的完整路徑：

  - 在 Windows 平台上，新增路徑：`install_root\java\bin\`
  - 在 Linux、UNIX 及 i5/OS 平台上，新增路徑 `install_root/java/bin/`
4. 根據標頭中的指示，自訂資料庫建立 Script `createDatabase.sql` 的副本。您必須包括資料庫的名稱。在 Windows 平台上，請避免使用「記事本」編輯器，因為此編輯器所顯示的檔案無法讓使用者閱讀。
5. 建立資料庫。從要建立資料庫的目錄中，執行資料庫建立 Script 檔 `createDatabase.sql` 的自訂版本，如標頭中所述。

「商業程序編排器」的資料庫已存在。

## 建立商業程序編排器的 DB2 for i5/OS 資料庫

使用這項作業來建立「商業程序編排器」的 DB2 for i5/OS 資料庫綱目。

您已完成第 95 頁的『規劃 BPEDB 資料庫』。

1. 在管理該資料庫的機器上：如果擁有該資料庫的使用者 ID 沒有集合，請建立集合。
2. 將建立綱目 Script 複製到管理該資料庫的機器上。
  - 在 Linux、UNIX 及 i5/OS 平台上，它位於 `install_root/dbscripts/ProcessChoreographer/Db2iSeries/createSchema.sql`。
  - 在 Windows 平台上，它位於 `install_root\dbscripts\ProcessChoreographer\Db2iSeries\createSchema.sql`。
- a. 如果您想要在「商業程序編排器」第一次使用該資料庫時建立預設綱目，而且您會在配置頁面上選取**建立表格**選項，請跳過此步驟。

- b. 如果您不選取**建立表格**選項，請製作 SQL 檔案 createSchema.sql 的本端副本，並根據您在第 95 頁的『規劃 BPEDB 資料庫』和第 89 頁的『規劃安全、使用者 ID 及權限』中規劃的值來自訂它。
3. 建立資料庫物件。在 qshell 環境中，執行 Script 的副本。例如，若 Script 是在現行目錄中，請輸入此指令：
 

```
db2 -tf createSchema.sql
```
4. 若是遠端資料庫，請使用「工具箱」JDBC 驅動程式。請從資料庫主機下載 JAR 檔 /QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar。
5. 若是本端資料庫，請使用「原生」JDBC 驅動程式。請確定 classpath 包括 /QIBM/ProdData/Java400/ext/db2\_classes.jar。

「商業程序編排器」的 DB2 for i5/OS 綱目存在。

## 建立商業程序編排器的 DB2 for Linux、UNIX 及 Windows 資料庫

使用這項作業來建立「商業程序編排器」的 DB2 資料庫。

您已完成第 95 頁的『規劃 BPEDB 資料庫』。

1. 在管理該資料庫的伺服器上安裝 DB2。
2. 在所有使用第二類型「Java 資料庫連線功能 (JDBC)」驅動程式來存取該資料庫的遠端應用程式伺服器上，安裝 DB2 用戶端。
3. 在管理應用程式伺服器的伺服器上，切換至資料庫系統的「商業程序編排器」配置 Script 所在的目錄：
  - 在 Windows 系統上，請輸入：
 

```
cd install_root\dbscripts\ProcessChoreographer\DB2
```
  - 在 UNIX 和 Linux 系統上，輸入：
 

```
cd install_root/dbscripts/ProcessChoreographer/DB2
```
4. 如果資料庫位在不同於應用程式伺服器的伺服器上：
  - a. 將下列 SQL Script 複製到您的資料庫伺服器：
 

```
createDatabase.sql
createTablespace.sql
createSchema.sql
dropSchema.sql
dropTablespace.sql
```
  - b. 切換至您複製 SQL Script 的目錄。
5. 如果您要使用現有的資料庫，請跳至步驟 第 163 頁的 9，以建立表格空間和綱目。Unicode 支援：請確定資料庫支援 Unicode (UTF-8)。若沒有 Unicode 支援，則無法儲存 Java 程式碼中可處理的所有字元，當用戶端使用不相容的字碼頁時，將會發生字碼頁轉換問題。
6. 在資料庫伺服器上建立 DB2 實例。
7. 如果您有「對稱多重處理器 (SMP)」伺服器，請檢查可供 DB2 使用的處理器數目。檢查您的授權：
  - 在 AIX 系統上，輸入指令：
 

```
/usr/opt/db2_08_01/adm/db2licm -l
```
  - 在其他 UNIX 或 Linux 系統上，輸入此指令：
 

```
/opt/IBM/db2/V8.1/adm/db2licm -l
```

必要的話，請使用 `db2clim` 指令或「DB2 授權中心」，變更處理器授權的數目。

8. 建立新的資料庫：

- a. 請確定您使用的使用者 ID 具有資料庫系統的管理者權限。
- b. 如果您要建立適用於獨立開發、評估或示範用途的非正式作業資料庫 BPEDB，請輸入下列指令：

```
db2 -tf createDatabase.sql
```

若為正式作業環境，請考慮使用專用的表格空間儲存器並調整 DB2 參數。

9. 若要建立表格空間和綱目：

- a. 根據您在第 95 頁的『規劃 BPEDB 資料庫』中的規劃，自訂 `createTablespace.sql` 表格空間建立 Script 的副本。
- b. 請確定您具有資料庫系統的管理者權限。您用來建立綱目的使用者 ID 必須是您配置商業程序儲存器的資料來源時，所指定的同一個使用者 ID。
- c. 請確定您已附加至正確的實例。檢查 `DB2INSTANCE` 環境變數。
- d. 若要連接至資料庫 *databaseName*，請在 DB2 指令行處理器中輸入下列指令：

```
db2 connect to databaseName
```

- e. 若要建立表格空間，請輸入指令：

```
db2 -tf createTablespace.sql
```

請確定 Script 輸出沒有錯誤。如果發生錯誤，您可以使用 `dropTablespace.sql` Script 來捨棄表格空間。

- f. 若要建立綱目 (表格、索引及視圖)，請在 DB2 指令行處理器中輸入下列指令：

```
db2 -tf createSchema.sql
```

請確定 Script 輸出沒有錯誤。如果您要捨棄綱目，請使用 `dropSchema.sql` Script。

**註：**如果您此刻不建立表格空間和綱目，則稍後必須使用**建立表格**選項，使「商業程序編排器」第一次嘗試使用該資料庫時，建立預設的表格空間和綱目。

10. 在可從遠端存取該資料庫的每一個應用程式伺服器上：

- a. 輸入下列指令來編目資料庫：

```
db2 catalog database databaseName as databaseAlias at node nodeName
```

如需編目資料庫的相關資訊，請參閱 DB2 文件。

- b. 輸入下列指令，驗證您可以連接資料庫：

```
db2 connect to databaseName user userID  
db2 connect reset
```

「商業程序編排器」的資料庫已存在。

## 建立商業程序編排器的 DB2 for z/OS 資料庫

使用這項作業來建立「商業程序編排器」的 DB2 for z/OS 資料庫。

您已完成第 95 頁的『規劃 BPEDB 資料庫』。

本主題說明如何建立 DB2 for z/OS 資料庫，以及如何驗證應用程式伺服器所在的伺服器可以連接此資料庫。

1. 您必須已在 UNIX、Linux、Windows 或 i5/OS 伺服器上安裝 WebSphere Process Server。
2. 在資料庫所在的 z/OS 伺服器上：
  - a. 登入原生的 z/OS 環境。
  - b. 如果安裝多個 DB2 系統，請決定您要使用的子系統。
  - c. 記下 DB2 子系統所接聽的 IP 埠。
  - d. 建立資料庫及儲存體群組。執行下列動作之一：
    - 使用 DB2 管理功能表來建立新的資料庫及儲存體群組。
    - 根據標頭中的指示編輯 createDatabase.sql Script 檔的副本，然後執行您的副本。它位於下列目錄中：
      - 在 Windows 平台上：`install_root\dbscripts\ProcessChoreographer\database_type`
      - 在 Linux、UNIX、i5/OS，以及 z/OS 平台上的「UNIX 系統服務 (USS)」中：`install_root/dbscripts/ProcessChoreographer/database_type`
 其中 `database_type` 是下列其中一項：
      - DB2zOSV7
      - DB2zOSV8
  - e. 記下名稱。
  - f. 決定從執行 WebSphere Process Server 的遠端伺服器上，用來連接資料庫的使用者 ID。基於安全理由，此使用者 ID 通常不會是您用來建立資料庫的使用者 ID。
  - g. 將資料庫和儲存體群組的存取權授與使用者 ID。使用者 ID 也必須具有為資料庫建立新表格的許可權。
  - h. 決定您是否要在連接的使用者 ID 的綱目中建立表格及視圖，或是否要自訂綱目限定元。如果單一使用者 ID 存取含有同名表格的多個資料庫，則您必須使用不同的綱目限定元來避免名稱衝突。
3. 在 WebSphere Process Server 所在的伺服器上：
  - a. 請確定您已安裝 DB2 Connect™ Gateway。DB2 Connect Gateway 是 DB2 UDB ESE 套件的一部分，但也可以分開安裝。
  - b. 在 DB2 指令行視窗中，使用下列指令來編目遠端資料庫：
 

```
catalog tcpip node zosnode remote host_name server IP_port ostype mvs
catalog database location as database_alias at node zosnode authentication dcs
catalog dcs database database_alias parms ',,INTERRUPT_ENABLED'
```
  - c. 請確定已指定同步點管理程式實例名稱。輸入下列指令：
 

```
db2 update dbm cfg using SPM_NAME host_name
db2 update dbm cfg using SPM_LOG_FILE_SZ log_file_size
```
  - d. 請注意下列資訊：
 

DB2 for z/OS 與 DB2 for Linux、UNIX 及 Windows 之間有一項重大差異。DB2 for Linux, UNIX, and Windows 沒有子系統的概念，但 DB2 for z/OS 有此概念。為了避免資料庫名稱和子系統名稱發生混淆，必須知道由於 DB2 for z/OS 是在子系統中執行，`catalog node` 和 `catalog database` 指令必須指定適當的子系統。在 DB2 for Linux、UNIX 及 Windows 上，沒有子系統名稱的概念，所以 `catalog` 指令建立鏈結的資料庫名稱，實際上就是 DB2 for z/OS 子系統的名稱。
  - e. 輸入下列指令，驗證您可以建立遠端子系統的連線：



```
db2 connect to subsystem user userid using password
```

- f. 在管理應用程式伺服器的伺服器上，切換至資料庫系統的「商業程序編排器」配置 Script 所在的目錄：
  - 在 Windows 系統上，視您的 DB2 版本而定，輸入下列其中一個指令：

```
cd install_root\dbscripts\ProcessChoreographer\DB2zOSV7
```

```
cd install_root\dbscripts\ProcessChoreographer\DB2zOSV8
```
  - 在 UNIX、Linux 及 i5/OS 系統上，視您的 DB2 版本而定，輸入下列其中一個指令：

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV7
```

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV8
```
- g. 根據您在第 95 頁的『規劃 BPEDB 資料庫』中的規劃，自訂 createTablespace.sql 表格空間建立 Script 的副本。將 @STOGRP@ 取代為儲存體群組名稱，且將 @DBNAME@ 取代為資料庫名稱 (不是子系統名稱)。
- h. 執行您自訂的 createTablespace.sql Script 版本，如 Script 的標頭所述。如果您要捨棄表格空間，請自訂及執行 dropTablespace.sql Script。
- i. 編輯 createSchema.sql Script。
  - 1) 將 @STOGRP@ 取代為儲存體群組名稱。
  - 2) 將 @DBNAME@ 取代為資料庫名稱 (不是子系統名稱)。
  - 3) 將 @SCHEMA@ 取代為綱目限定元，或從 Script 中移除 @SCHEMA@ (包括隨後的句點)。自訂綱目限定元只能用於 DB2 Universal JDBC Driver，且需要將配置 customSQLID 內容設為適當的值。
- j. 執行您自訂的 createSchema.sql Script 版本，如 Script 的標頭所述。如果這個 Script 沒有作用，或者您想要移除表格及視圖，請使用 dropSchema.sql Script 來捨棄綱目，但在執行 Script 之前，必須先取代 @SCHEMA@。

「商業程序編排器」的資料庫已存在。

## 建立商業程序編排器的 Informix Dynamic Server 資料庫

使用這項作業來建立「商業程序編排器」的 Informix Dynamic Server 資料庫。

您已完成第 95 頁的『規劃 BPEDB 資料庫』。

1. 在資料庫所在的電腦上安裝 Informix 伺服器。
2. 建立 Informix 伺服器實例。請確定 Informix 環境變數設定正確。尤其，*INFORMIXSERVER* 必須指向新的實例，而 *ONCONFIG* 必須指向實例的配置檔。如需不同環境變數及其需要如何設定的詳細資訊，請參閱 Informix Dynamic Server 文件。請確定「全球語言支援 (GLS)」的相關環境變數設為 Unicode (UTF-8) 支援。儲存 Java 程式碼中可處理的所有字元需要「Unicode 支援」。
3. 在下列位置複製並配置「Java 資料庫連線功能 (JDBC)」驅動程式：
  - 所有使用資料庫伺服器的遠端應用程式伺服器。
4. 在管理應用程式伺服器的伺服器上，切換至資料庫系統的「商業程序編排器」配置 Script 所在的目錄：
  - 在 Windows 系統上，請輸入：

```
cd install_root\dbscripts\ProcessChoreographer\Informix
```
  - 在 UNIX 和 Linux 系統上，輸入：

```
cd install_root/dbscripts/ProcessChoreographer/Informix
```

5. 如果資料庫與應用程式伺服器位於不同的伺服器：
  - a. 從應用程式伺服器的 ProcessChoreographer 目錄中，將作業系統的 Script 複製到資料庫所在伺服器上的適當目錄：若為 Linux 和 UNIX 上的 Informix，請複製下列檔案：

```
createDatabase.sql
createSchema.sql
dropSchema.sql
createDbSpace.sh
dropDbSpace.sh
```

若為 Windows 上的 Informix，請複製下列檔案：

```
createDatabase.sql
createSchema.sql
dropSchema.sql
createDbSpace.bat
dropDbSpace.bat
```

- b. 切換至您複製 SQL Script 的目錄。
6. 如果您要使用預設值來建立適用於獨立開發、評估或示範用途的非正式作業資料庫，請輸入指令：

```
dbaccess - createDatabase.sql
```

此指令會為您使用的使用者 ID 建立 Informix 資料庫 BPEDB。請確定 Script 輸出沒有錯誤。您可以使用 dropSchema.sql Script，只捨棄綱目，或使用 SQL 指令 DROP DATABASE，捨棄整個資料庫。

7. 如果您要建立正式作業系統的資料庫，您必須手動建立資料庫：
  - a. 建立資料庫，例如 BPEDB。
  - b. 為您的資料庫建立 Dbspaces。

在 Windows 系統上，閱讀 createDbSpace.bat 檔案中的指示。將 Script 中的值參數調整為適用於環境的值。

在 UNIX 和 Linux 系統上，閱讀 createDbSpace.sh 檔案中的指示。將 Script 中的值參數調整為適用於環境的值。

- c. 輸入下列指令，執行 Script 來建立綱目：

```
dbaccess databaseName createSchema.sql
```

其中，*databaseName* 是資料庫的名稱，例如 BPEDB。

- d. 檢查 Script 輸出是否有任何錯誤。如果您要捨棄綱目，請使用 dropSchema.sql Script。

「商業程序編排器」的資料庫已存在。

## 建立商業程序編排器的 Microsoft SQL Server 資料庫

使用這項作業來建立「商業程序編排器」的 Microsoft SQL Server 資料庫。

您已完成第 95 頁的『規劃 BPEDB 資料庫』。

1. 在管理該資料庫的伺服器上安裝 Microsoft SQL Server。請確定您已選取選項來建立區分大小寫的實例。如果您已有使用不區分大小寫選項所建立的 SQL Server，請執行重新建置主要工具，將對照設定變更為區分大小寫。請確定伺服器支援 Unicode。

2. 請確定資料庫伺服器 and 「分散式交易協調器 (DTC)」正在執行中。請確定資料庫伺服器已配置為可進行分散式交易。如需詳細資訊，請參閱 Microsoft SQL Server 的文件。
3. 在管理應用程式伺服器的伺服器上，切換至資料庫系統的「商業程序編排器」配置 Script 所在的目錄：
  - 在 Windows 系統上，請輸入：

```
cd install_root\dbscripts\ProcessChoreographer\SQLServer
```
  - 在 UNIX 和 Linux 系統上，輸入：

```
cd install_root/dbscripts/ProcessChoreographer/SQLServer
```
4. 如果資料庫與應用程式伺服器位於不同的伺服器：
  - a. 從應用程式伺服器的 SQLServer 目錄中，將資料庫的 SQL Script 複製到資料庫所在伺服器上的適當目錄：

```
createDatabase.sql
createDatabaseUnicode.sql
createSchema.sql
createSchemaUnicode.sql
dropSchema.sql
```
  - b. 切換至您複製 SQL Script 的目錄。
5. 執行下列動作之一：
  - 如果您要建立適用於獨立開發、評估或示範用途的非正式作業 SQL Server 資料庫『BPEDB』：
    - a. 執行下列其中一個 Script，如檔案標頭中所述。
      - createDatabase.sql
      - createDatabaseUnicode.sql (Unicode 資料庫)例如，輸入：

```
sqlcmd -U userID -P password -i createDatabase.sql
```
    - b. 請確定 Script 輸出沒有錯誤。如果發生錯誤，您可以使用 dropSchema.sql Script 來捨棄綱目。
  - 如果您要建立正式作業 SQL Server 資料庫，請手動建立資料庫：
    - a. 建立資料庫，例如『BPEDB』。
    - b. 若要建立綱目，請執行下列其中一個 Script，如檔案標頭中所述。
      - createSchema.sql
      - createSchemaUnicode.sql (如果已建立 Unicode 資料庫)例如，輸入：

```
sqlcmd -U userID -P password -i createSchema.sql
```
    - c. 請確定 Script 輸出沒有錯誤。如果發生錯誤，您可以使用 dropSchema.sql Script 來捨棄綱目。

「商業程序編排器」的資料庫已存在。

## 建立商業程序編排器的 Oracle 資料庫

使用這項作業來建立「商業程序編排器」的 Oracle 資料庫。

您已完成第 95 頁的『規劃 BPEDB 資料庫』。

1. 在資料庫所在的電腦上安裝 Oracle 伺服器。請確定您使用位於 lib32 子目錄中的 32 位元 Oracle 程式庫。
2. 在 Linux 和 UNIX 系統上，請確定已為 root 使用者設定環境變數 `ORACLE_BASE` 和 `ORACLE_HOME`。
3. 檢查類別路徑，確定 JDBC 驅動程式使用 `ojdbc14.jar` 檔案。
4. 在 Linux 和 UNIX 系統上，對 `/usr/lib` 目錄中的下列 Oracle 程式庫建立軟式鏈結：
  - 若為 Oracle 10g：鏈結至：`libclnt.so.10.1`。
  - 若為 Oracle 9i：鏈結至：`libnnz10.so`、`libclnt.so.10.1`、`libclntsh.so.10.1` 及 `libocijdbc10.so`。

如需如何設定 Oracle OCI 用戶端的詳細資訊，請參閱 Oracle 提供的文件。

5. 使用「資料庫配置輔助程式」建立 Oracle 資料庫，例如，使用名稱 `BPEDB`。沒有 Script 可以快速建立「商業程序編排器」的預設 Oracle 資料庫。請確定您已選取資料庫的 `JServer` 選項。建立的資料庫必須具有 Unicode 字碼頁。
6. 請輸入下列指令以啟動 Oracle 接聽器：

```
lsnrctl start
```

7. 選擇性的：如果您不要自訂表格空間和綱目，您可以跳過這項作業的其餘步驟，在此情況下，當「商業程序編排器」第一次嘗試使用資料庫時，將會建立預設的表格空間和綱目。
8. 在管理應用程式伺服器的伺服器上，切換至資料庫系統的「商業程序編排器」配置 Script 所在的目錄：

- 在 Windows 系統上，請輸入：

```
cd install_root\dbscripts\ProcessChoreographer\Oracle
```

- 在 UNIX 和 Linux 系統上，輸入：

```
cd install_root/dbscripts/ProcessChoreographer/Oracle
```

9. 如果資料庫位在不同於應用程式伺服器的伺服器上：
  - a. 從應用程式伺服器所在伺服器上的「商業程序編排器」子目錄中，將下列 Oracle 配置 SQL Script 複製到資料庫所在伺服器上的適當目錄：

```
createSchema.sql
createTablespace.sql
dropSchema.sql
dropTablespace.sql
```

- b. On your database server, change to the directory where you copied the SQL scripts.

10. 如果您不要在預設實例中建立綱目，請將 `ORACLE_SID` 環境變數設定為您在 5 步驟中建立的資料庫 SID。
11. 請確定執行這些 Script 的使用者至少具有下列資料庫專用權：`CREATE SESSION`、`CREATE TABLESPACE`、`DROP TABLESPACE`、`CREATE TABLE` 及 `CREATE VIEW`。
12. 根據您在第 95 頁的『規劃 BPEDB 資料庫』中的規劃，自訂 `createTablespace.sql` 表格空間建立 Script 的副本。
13. 若要建立表格空間，請執行 `createTablespace.sql` Script。基於測試目的，您可以讓所有表格空間使用相同位置，並將路徑當作指令行引數傳遞至 Script，例如，在 Windows 系統上，若使用使用者 ID 『`bpeuser`』、密碼 『`bpepwd`』、資料庫名稱 『`BPEDB`』、表格空間路徑 `d:\mydb\ts`，請輸入：

```
sqlplus bpeuser/bpepwd@BPEDB @createTablespace.sql d:\mydb\ts
```

如果您要捨棄表格空間，則可以使用 `dropTablespace.sql` Script。

- 請確定擁有表格的使用者，在前一個步驟所建立的所有表格空間上獲授與足夠配額。
- 根據檔案開頭的指示來編輯綱目建立 Script `createSchema.sql`，並將位置保留元 `@SCHEMA@` 取代為綱目的名稱。如果 `@SCHEMA@` 與執行 `createSchema.sql` Script 的使用者不同，請確定此使用者具有下列資料庫專用權：`CREATE ANY TABLE`、`ALTER ANY TABLE`、`CREATE ANY INDEX` 及 `CREATE ANY VIEW`。
- 若要建立綱目，請執行 `createSchema.sql` Script。例如，在 Windows 系統上，輸入：

```
sqlplus bpeuser/bpepwd@BPEDB @createSchema.sql
```

「商業程序編排器」的資料庫已存在。

## 配置人員目錄提供者

您可以使用這項作業，配置「輕量型目錄存取通訊協定 (LDAP)」或「虛擬成員管理程式 (VMM)」人員目錄提供者，「商業程序編排器」會使用它來決定誰可以啟動程序，或誰可以要求活動或作業。

每一種支援的人員目錄服務都需要一個對應的人員目錄提供者。支援的人員目錄提供者如下：

表 14. 支援的人員目錄提供者

人員目錄提供者	人員目錄提供者選項
「輕量型目錄存取通訊協定 (LDAP)」目錄	LDAP 人員目錄提供者
虛擬成員管理程式	VMM 人員目錄提供者
本端作業系統使用者登錄	系統人員目錄提供者
WebSphere Application Server 使用者登錄	使用者登錄人員目錄提供者

已利用預設配置來安裝所有這些外掛程式。您可以搭配使用使用者登錄及系統外掛程式與預設配置。若為 VMM，預設配置通常就足夠了。

## 配置虛擬成員管理程式人員目錄提供者

配置「虛擬成員管理程式 (VMM)」人員目錄提供者，讓「商業程序編排器」執行人員指派，決定誰可以啟動程序，或誰可以要求活動或作業。預設人員目錄提供者已備妥可供使用，而且只在您引進自訂的人員指派準則時才需要配置。

您已配置聯合儲存庫。

- 備份 VMM 的標準轉換檔，並給予不同的名稱，例如 `myVMMTransformation.xml`。
  - 在 Windows 平台上，此檔案位於 `install_root\ProcessChoreographer\Staff\VMMTransformation.xml`
  - 在 Linux、UNIX 及 iOS 平台上，此檔案位於 `install_root/ProcessChoreographer/Staff/VMMTransformation.xml`

**警告：** 請不要修改轉換檔的原始版本，因為在未來套用服務套件或修正套件時可能會直接改寫而不會出現警告。

2. 如果已在叢集上配置「商業程序編排器」，請使轉換檔的副本可在每一個管理叢集成員的 WebSphere Process Server 安裝上使用。
3. 在管理主控台中，按一下資源 → 人員目錄提供者。
4. 選取適當的節點。

選項	敘述
若為獨立式設定檔：	只顯示一個節點。
在單一伺服器上已配置「商業程序編排器」的網路部署環境中：	選取包含伺服器的節點。
在叢集上已配置「商業程序編排器」的網路部署環境中：	您必須在每一個管理叢集成員的節點上配置人員目錄提供者。選取第一個節點、配置該節點上的人員目錄提供者，然後對所有管理叢集成員的其他節點重複配置。

5. 若要建立新的 VMM 人員目錄配置：
    - a. 按一下 **VMM 人員目錄提供者**。
    - b. 在**其他內容**中，選取**人員目錄配置**。
    - c. 按一下**新建** → **瀏覽**，然後選取您在步驟 第 169 頁的 1 中複製的「可延伸樣式表語言 (XSL)」轉換檔副本。
    - d. 按**下一步**。
    - e. 在**一般內容**區段中，輸入新人員目錄配置的管理名稱。
    - f. 選擇性的: 輸入說明。
    - g. 輸入唯一的「Java 命名和目錄介面 (JNDI)」名稱，讓系統識別這個配置。例如，`bpe/staff/myvmmconfiguration`  
**註:** 沒有其他配置參數
    - h. 按一下**確定**，然後按一下**儲存**。
  6. 若要啟動提供者配置，請停止並啟動一台以上您已配置提供者的伺服器。
  7. 選擇性的: 如果您想要變更 XSL 轉換檔:
    - a. 依需要編輯 XSL 檔。
    - b. 請重新啟動伺服器。在網路部署環境中，新版本的 XSL 檔必須可供所有伺服器使用，而且必須重新啟動所有伺服器。
  8. 如果您在任何步驟中遇到問題，請參閱 *WebSphere Process Server 疑難排解 PDF*。
- 已配置 VMM 人員目錄提供者。

## 配置 LDAP 人員目錄提供者

您可以使用這項作業來配置「輕量型目錄存取通訊協定 (LDAP)」人員目錄提供者，讓「商業程序編排器」執行人員指派，決定誰可以啟動程序，或誰可以要求活動或作業。

您已執行第 105 頁的『規劃人員目錄提供者』所述的 LDAP 規劃。

LDAP 人員目錄提供者配置是利用指向本端 LDAP 伺服器的 URL 來起始設定的。稍後您必須變更 URL，以指向真正的 LDAP 伺服器，這個伺服器通常位於應用程式伺服器的遠端。LDAP 人員目錄提供者是針對容許匿名存取的 LDAP 伺服器所配置的。



1. 備份 LDAP 的標準轉換檔，並給予不同的名稱，例如 myLDAPTransformation.xml。
  - 在 Windows 平台上，此檔案位於 `install_root\ProcessChoreographer\Staff\LDAPTransformation.xml`
  - 在 Linux、UNIX 及 i5/OS 平台上，此檔案位於 `install_root/ProcessChoreographer/Staff/LDAPTransformation.xml`
2. 選擇性的: 改寫轉換檔的副本，以符合組織的 LDAP 綱目。您可以現在或稍後執行第 172 頁的『修改 LDAP 轉換檔』。
 

**警告:** 請不要修改轉換檔的原始版本，因為在未來套用服務或修正套件時可能會直接改寫而不會出現警告。
3. 如果已在叢集上配置「商業程序編排器」，請使轉換檔的副本可在每一個管理叢集成員的 WebSphere Process Server 安裝上使用。
4. 在管理主控台中，按一下資源 → 人員目錄提供者。
5. 選取適當的節點。

選項	敘述
若為獨立式設定檔：	只顯示一個節點。
在單一伺服器上已配置「商業程序編排器」的網路部署環境中：	選取包含伺服器的節點。
在叢集上已配置「商業程序編排器」的網路部署環境中：	您必須在每一個管理叢集成員的節點上配置人員目錄提供者。選取第一個節點、配置該節點上的人員目錄提供者，然後對所有管理叢集成員的其他節點重複配置。

6. 若要在選取的節點上建立新的 LDAP 配置：
  - a. 按一下 **LDAP 人員目錄提供者**。
  - b. 在「其他內容」下，按一下**人員目錄配置**。
  - c. 按一下**新建** → **瀏覽**，然後選取您的轉換檔副本。
  - d. 按**下一步**。
  - e. 輸入人員目錄配置的管理名稱。
  - f. 輸入說明。
  - g. 輸入讓人力作業用來參照這個提供者的「Java 命名和目錄介面 (JNDI)」名稱。例如，`bpe/staff/ldapsrver1`
  - h. 按一下**套用**。
  - i. 按一下**自訂內容**。
  - j. 對於您在 第 105 頁的 2 中規劃的每一個必要內容和任何選用內容，請按一下內容名稱、輸入值，然後按一下**確定**。
 

**註:** 若為選用的其他內容，您可以設定針對 JNDI 定義的內容，例如，啟用 LDAP 轉介。若為 `providerURL`，您可以指定以 `ldap://` 或 `ldaps://` 開頭的 URL。
  - k. 若要套用變更，請按一下**儲存**。
7. 選擇性的: 如果您想要變更 XSL 轉換檔:
  - a. 依需要編輯 XSL 檔。
  - b. 請重新啟動伺服器。在網路部署環境中，新版本的 XSL 檔必須可供所有伺服器使用，而且必須重新啟動所有伺服器。

- 若要啟動提供者配置，請停止並啟動一台以上您已配置提供者的伺服器。
- 如果您在任何這些步驟中有問題，請參閱 *WebSphere Process Server 疑難排解 PDF*。

現在，人力作業及程序可以使用人員指派服務來解析人員指派查詢，也可以決定哪些人可以執行哪些活動。

## 修改 LDAP 轉換檔

說明如何修改 LDAP 轉換 XSL 檔案，以符合組織的 LDAP 綱目。

預設 LDAPTransformation.xml 檔案會將預先定義的人員指派準則對映至 LDAP 查詢，而這些查詢會使用 WebSphere 使用的預設 LDAP 綱目元素。此綱目假設存在下列情況：

- 群組項目的 LDAP 物件類別為 groupOfName。
- 包含群組的成員 DN 的群組項目屬性為 member。
- 人員項目的 LDAP 物件類別為 inetOrgPerson。
- 包含人員項目中登入 ID 的屬性為 uid。
- 包含人員電子郵件位址的人員項目屬性為 mail。
- 包含人員經理識別名稱的人員項目屬性為 manager。

如果 LDAP 綱目使用不同的物件類別及屬性名稱，則必須變更所使用 LDAP 轉換檔中的這些設定。請製作原始 LDAPTransformation.xml 檔案的副本 (如第 170 頁的『配置 LDAP 人員目錄提供者』中所述)。

**警告：** 請不要修改轉換檔的原始版本，因為在未來套用服務套件或修正套件時可能會直接改寫而不會出現警告。

編輯檔案的變數宣告部分，以變更所有人員指派準則的設定，通常就已足夠：

```
<xsl:variable name="DefaultGroupClass">groupOfNames</xsl:variable>
<xsl:variable name="DefaultGroupClassMemberAttribute">member</xsl:variable>

<xsl:variable name="DefaultPersonClass">inetOrgPerson</xsl:variable>
<xsl:variable name="DefaultUserIDAttribute">uid</xsl:variable>
<xsl:variable name="DefaultMailAttribute">mail</xsl:variable>
<xsl:variable name="DefaultManagerAttribute">manager</xsl:variable>
```

您可以在 XSL 範本內套用變更，這些範本會轉換個別人員指派準則 (如下列範例所示)。

### 範例：GroupMembers

將群組項目的物件類別變更為 groupOfUniqueNames、將包含成員 DN 清單的群組項目屬性變更為 uniqueMember，並將包含登入的人員項目屬性變更為 cn：

```
<slldap:usersOfGroup>
...

<slldap:attribute>
<xsl:attribute name="name">uniqueMember</xsl:attribute>
<xsl:attribute name="objectclass">groupOfUniqueNames</xsl:attribute>
<xsl:attribute name="usage">recursive</xsl:attribute>
</slldap:attribute>

...
<slldap:attribute>
<xsl:attribute name="name">cn</xsl:attribute>
<xsl:attribute name="objectclass">inetOrgPerson</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</slldap:attribute>
...
<slldap:resultObject>
<xsl:attribute name="objectclass">groupOfUniqueNames</xsl:attribute>
```

```

<xsl:attribute name="usage">recursive</xsl:attribute>
<ldap:resultAttribute>
<xsl:attribute name="name">uniqueMember</xsl:attribute>
<xsl:attribute name="destination">intermediate</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

<ldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">cn</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

</ldap:usersOfGroup>

```

### 範例：GroupMembersWithoutFilteredUsers

將 LDAP 過濾器運算子變更為 `>=`。

```

<ldap:StaffQueries>
<ldap:usersOfGroup>
...
</ldap:usersOfGroup>

<ldap:intermediateResult>
<xsl:attribute name="name">filteredusers</xsl:attribute>
<ldap:search>
<xsl:attribute name="filter">
<xsl:value-of select="staff:parameter[@id='FilterAttribute']"/>
  >=
<xsl:value-of select="staff:parameter[@id='FilterValue']"/>
</xsl:attribute>
...
<ldap:search>
...
</ldap:intermediateResult>
...
</ldap:StaffQueries>

```

### 範例：GroupSearch

將搜尋屬性變更為 `MyType`、將物件類別變更為 `mypersonclass`，並將包含登入 ID 的屬性變更為 `myuid`。

```

<ldap:StaffQueries>
...
<ldap:search>
<xsl:attribute name="filter">
(&
...
<xsl:if test="staff:parameter[@id='MyType']!="">
(<xsl:value-of select="$GS_Type"/>=>
<xsl:value-of select="staff:parameter[@id='Type']"/>)
)
...
</xsl:attribute>

<ldap:attribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</ldap:attribute>

```

```

...
<slldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<slldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</slldap:resultAttribute>
<slldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</slldap:resultAttribute>
<slldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</slldap:resultAttribute>
</slldap:resultObject>

<slldap:search>
</slldap:StaffQueries>

```

## 範例：Manager of Employee

將包含管理者 DN 的屬性變更爲 managerentry，並將管理者登入 ID 屬性的來源變更爲 name。

```

<slldap:StaffQueries>

<slldap:intermediateResult>
...
<slldap:user>
...
<xsl:attribute name="name">managerentry</xsl:attribute>
...
<slldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<slldap:resultAttribute>
<xsl:attribute name="name">managerentry</xsl:attribute>
<xsl:attribute name="destination">intermediate</xsl:attribute>
</slldap:resultAttribute>
</slldap:resultObject>
</slldap:user>
</slldap:intermediateResult>

<slldap:user>
...
<xsl:attribute name="name">name</xsl:attribute>
...
<slldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<slldap:resultAttribute>
<xsl:attribute name="name">name</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</slldap:resultAttribute>
<slldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</slldap:resultAttribute>
<slldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</slldap:resultAttribute>
</slldap:resultObject>

</slldap:user>
</slldap:StaffQueries>

```

## 範例：PersonSearch

將搜尋屬性變更爲 MyAttribute、將物件類別變更爲 mypersonclass，並將傳回屬性的來源變更爲 myuid。

```

<ldap:StaffQueries>
...
<ldap:search>
<xsl:attribute name="filter">
(&
...
<xsl:if test="staff:parameter[@id='MyAttribute']!="">
(<xsl:value-of select="$PS_UserID"/>=
<xsl:value-of select=staff:parameter[@id='UserID']"/>)
)
</xsl:if>
...
</xsl:attribute>

<ldap:attribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</ldap:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>
</ldap:search>
</ldap:StaffQueries>

```

## 範例：Users

將傳回屬性的來源變更為 myuid，並將物件類別變更為 mypersonclass。

```

<ldap:user>
...
<xsl:attribute name="attribute">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>

<ldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

</ldap:user>

```

---

## 配置人員替代

建立並啟動「虛擬成員管理程式 (VMM)」內容延伸儲存庫，讓「商業程序編排器」支援使用者替代。

您已對「聯合儲存庫」配置 WebSphere 安全，而且如果您已引進自訂的人員指派準則，也已執行第 169 頁的『配置虛擬成員管理程式人員目錄提供者』。您知道您是否將使用檔案登錄、內容延伸登錄或現有的 LDAP 綱目，以儲存內容延伸。

1. 將兩個屬性 (『isAbsent』 作為單值字串，『substitutes』 作為多值字串) 新增至 PersonAccount 的 VMM 定義：

a. 尋找 wimxmlextension.xml 檔：

- 在 Linux、UNIX 及 i5/OS 平台上，它位於 *profile\_root/config/cells/cell\_name/wim/model* 中
- 在 Windows 平台上，它位於 *profile\_root\config\cells\cell\_name\wim\model* 中

b. 製作 wimxmlextension.xml 檔的備份副本。

c. 編輯 wimxmlextension.xml 檔的原始副本，並確定包含下列定義，這些定義將使用者代理所需的兩個屬性新增至 PersonAccount 實體類型：

```
<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
  dataType="STRING" multiValued="false" propertyName="isAbsent">
  <wim:applicableEntityTypeNames>PersonAccount
  </wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
  dataType="STRING" multiValued="true" propertyName="substitutes">
  <wim:applicableEntityTypeNames>PersonAccount
  </wim:applicableEntityTypeNames>
</wim:propertySchema>
```

如果您是使用檔案登錄 fileRegistry.xml，請跳至步驟 第 178 頁的 4。

2. 如果使用內容延伸登錄來保留代理資訊，請執行下列動作。如需設定內容延伸儲存庫的相關資訊，請參閱配置聯合儲存庫配置中的內容延伸儲存庫。

a. 請確定有資料庫可用來儲存內容延伸。

b. 請確定伺服器類別路徑上有 JDBC 驅動程式類別。按一下環境 → WebSphere 變數以檢查。必要的話，按一下應用程式伺服器 → *server\_name* → 程序定義 → Java 虛擬機器 → 配置，將 JDBC 驅動程式新增至類別路徑。若為 DB2，將 db2jcc.jar,db2jcc\_license\_cu.jar 及 db2jcc\_license\_cisuz.jar 新增至伺服器的類別路徑，然後按一下套用 → 儲存

c. 使用管理主控台來配置 VMM 的 DB2 Universal JDBC 驅動程式提供者及第四類型資料來源。將資料來源的 webSphereDefaultIsolationLevel 自訂內容設定為值 2。如需變更預設隔離層次的相關資訊，請參閱變更非 CMP 應用程式的預設隔離層次及說明如何使用新的自訂內容 webSphereDefaultIsolationLevel 來這樣做。

d. 請重新啟動伺服器。

e. 製作 wimlaproperties.xml 檔的備份副本。

- 在 Linux、UNIX 及 i5/OS 平台上，它位於 *profile\_root/config/cells/cell\_name/wim/model* 中
- 在 Windows 平台上，它位於 *profile\_root\config\cells\cell\_name\wim\model* 中

f. 編輯 wimlaproperties.xml 檔的原始副本，並新增下列定義：

```
<wimprop:property wimPropertyName="isAbsent" dataType="String" valueLength="128" multiValued="false">
<wimprop:applicableEntityName>
  <wimprop:entityName>PersonAccount</wimprop:entityName>
</wimprop:applicableEntityName>
</wimprop:property>

<wimprop:property wimPropertyName="substitutes" dataType="String" valueLength="128" multiValued="true">
<wimprop:applicableEntityName>
  <wimprop:entityName>PersonAccount</wimprop:entityName>
</wimprop:applicableEntityName>
</wimprop:property>
```



- g. 請確定應用程式伺服器 (或在網路部署環境中，部署管理程式) 正在執行中。請明白不要對 `wsadmin` 公用程式使用 `-conntype NONE` 選項。
- h. 使用 VMM 管理作業 `setupIdMgrPropertyExtensionRepositoryTables`，在「內容延伸儲存庫」資料庫中建立代理內容。如需詳細資料，請參閱使用 `wsadmin` 指令來設定項目對映儲存庫、內容延伸儲存庫或自訂登錄資料庫儲存庫。例如，在 Windows 平台上使用 DB2 資料庫：

```
$AdminTask setupIdMgrPropertyExtensionRepositoryTables {
-reportSqlError true
-schemaLocation install_root\etc\wim\setup
-laPropXML install_root\etc\wim\setup\wimlaproperties.xml
-databaseType db2
-dbURL jdbc:db2:
-dbDriver com.ibm.db2.jcc.DB2Driver
-dbAdminId userID
-dbAdminPassword password }
```

- i. 如果您是使用「輕量型目錄存取通訊協定 (LDAP)」使用者儲存庫，請編輯 Cell 的 `wim` 子目錄的 `config` 子目錄中的 `wimconfig.xml` 檔，並新增下列項目，從 LDAP 儲存庫中排除代理屬性：

```
<config:repositories xsi:type="config:LdapRepositoryType"
adapterClassName="com.ibm.ws.wim.adapter.LdapAdapter"
id="ldaprepo1" ...>
...
<config:attributeConfiguration>
<config:propertiesNotSupported name="isAbsent"/>
<config:propertiesNotSupported name="substitutes"/>
</config:attributeConfiguration>
```

- j. 啟動延伸內容儲存庫：

- 1) 使用 `setIdMgrPropertyExtensionRepository` 指令。如需詳細資料，請參閱使用 `wsadmin` 指令來設定項目對映儲存庫、內容延伸儲存庫或自訂登錄資料庫儲存庫。例如，使用名稱為 `VMMDB` 的 DB2 資料庫、名稱為 `VMMDS` 的資料來源：

```
$AdminTask setIdMgrPropertyExtensionRepository {
-dataSourceName jdbc/VMMDS
-databaseType db2
-dbURL jdbc:db2:VMMDB
-dbAdminId userID
-dbAdminPassword password
-JDBCClass com.ibm.db2.jcc.DB2Driver
-entityRetrievalLimit 10 }
```

- 2) 驗證 `wimconfig.xml` 檔是否包含如下的項目：

```
<config:propertyExtensionRepository
adapterClassName="com.ibm.ws.wim.lookaside.LookasideAdapter"
id="LA"
databaseType="db2"
dataSourceName="jdbc/VMMDS"
dbAdminId="userID"
dbAdminPassword="{xor}PasswordXOR"
dbURL="jdbc:db2:VMMDB"
entityRetrievalLimit="10"
JDBCClass="com.ibm.db2.jcc.DB2Driver"/>
```

- 3. 如果使用 LDAP 網目來保留代理資訊：它可能有或可能沒有『`isAbsent`』及『`substitutes`』的定義 (可能具有不同的名稱)。您是否具有現有的定義，或您將建立新的定義，請確定下列：
  - a. LDAP 目錄必須容許寫入作業。
  - b. 缺席資訊的屬性 (『`isAbsent`』) 必須是屬於類型「布林」或「字串」。
  - c. 定義哪一個人員可以代理的屬性 (『`substitutes`』) 必須是屬於類型「字串」、多值，而且允許最多 128 個字元的長度。
  - d. 如果您現有或選擇的屬性名稱不是『`isAbsent`』及『`substitutes`』，則必須按一下 **伺服器** → **應用程式伺服器** → **server\_name** 或 **伺服器** → **叢集** → **cluster\_name**，定義您的屬性名稱，然後在商業整合下，展開商業程序編排

器，然後按一下人力作業管理程式 → 配置 → 自訂內容，再對自訂內容 Substitution.SubstitutesAttribute 及 Substitution.AbsenceAttribute 設定所需的名稱。

4. 請重新啟動伺服器。
5. 啓用「人力作業管理程式」中的代理：
  - a. 使用管理主控台，按一下伺服器 → 應用程式伺服器 → **server\_name** 或伺服器 → 叢集 → **cluster\_name**，然後在商業整合下，展開商業程序編排器，並按一下人力作業管理程式 → 執行時期
  - b. 若要啓用代理，請選取啓用代理。
  - c. 如果不容許任何管理者對其他使用者執行代理，請取消勾選將代理管理限制為管理者選項。

註：這個設定並不會影響使用者為自己變更代理的能力。

- d. 按一下套用。
6. 如果您在任何這些步驟中有問題，請參閱疑難排解 *WebSphere Process Server* PDF。

人員指派服務已配置為支援使用者代理缺席的使用者。

---

## 建立商業程序編排器的佇列管理程式及佇列

本文說明如何建立 WebSphere MQ 佇列管理程式和佇列。

必須已安裝 WebSphere MQ。

註：已淘汰 WebSphere MQ 的支援。

如果您使用 WebSphere MQ 作為外部「Java 訊息服務 (JMS)」提供者，則必須建立佇列管理程式和佇列。

1. 選擇性的：如果您在建立正式作業系統，請規劃佇列管理程式將使用的硬碟。對持續佇列資料及 WebSphere MQ 日誌使用預設位置可能會對佇列管理程式的效能造成負面影響。請考量根據 WebSphere MQ 文件的建議，變更這些位置。
2. 如果您不是建立 WebSphere MQ 叢集設定，請執行下列動作：
  - a. 請確定使用者 ID 具有建立 WebSphere MQ 佇列的權限。
  - b. 建立佇列管理程式及佇列：在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\config
createQueues.bat queueManager
```

在 UNIX 及 Linux 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
createQueues.sh queueManager
```

其中 *queueManager* 是現有佇列管理程式的名稱，或指定給新佇列管理程式的名稱。如果指定的佇列管理程式已存在，則會用來建立佇列。如果佇列管理程式不存在，則在建立預設佇列之前，將會先建立並啓動佇列管理程式。

- - 
  - 
  -
3. 如果您建立的 WebSphere 叢集設定會使用 WebSphere MQ 叢集，則只需要執行建立叢集佇列管理程式和佇列。
4. 如果您建立的 WebSphere 叢集設定會使用中央佇列管理程式，請執行下列動作：

- a. 從 WebSphere Process Server 機器所在伺服器上的 ProcessChoreographer 目錄的 config 子目錄中，將建立佇列 Script 檔複製到中央佇列管理程式所在的伺服器：
  - 如果中央佇列管理程式位於 Windows 工作站，請複製檔案：createQueues.bat
  - 如果中央佇列管理程式位於 UNIX 或 Linux 伺服器，請複製檔案：createQueues.sh
- b. 在佇列管理程式所在的伺服器上，請確定已安裝 WebSphere MQ，且您的使用者 ID 具有建立 WebSphere MQ 佇列的權限。
- c. 建立佇列管理程式及佇列：在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\config
createQueues.bat queueManager
```

在 Linux 及 UNIX 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
createQueues.sh queueManager
```

其中 *queueManager* 是指定給新佇列管理程式的名稱。

- d. 輸入下列指令，為新的佇列管理程式新增接聽器：

```
runmqtsr -t tcp -p port -m queueManager
```

其中 *port* 是接聽器接聽的埠號。

佇列管理程式及佇列已存在。

## 建立商業程序編排器的叢集佇列管理程式及佇列

如果您建立的「商業程序編排器」的 WebSphere 叢集設定會使用 WebSphere MQ 叢集，則必須建立佇列管理程式、佇列、叢集、儲存庫、通道及接聽器。

1. 如果您的 WebSphere 叢集由 UNIX 節點組成，請在每一個節點上執行下列動作：
  - a. 請確定使用者 ID 具有建立 WebSphere MQ 佇列的權限。
  - b. 輸入下列指令，建立取得及放置佇列管理程式、使它們成為 WebSphere MQ 叢集的成員，並建立佇列：

```
cd install_root/ProcessChoreographer/config
createQueues.sh getQueueManager clusterName putQueueManager
```

其中：

*getQueueManager*

給與取得佇列管理程式的唯一名稱。這個佇列管理程式管理所有本端佇列。

*clusterName*

WebSphere MQ 叢集的名稱，所有佇列管理程式都是該叢集的成員。

*putQueueManager*

放置佇列管理程式的唯一名稱。這個佇列管理程式不管理任何佇列，以確保訊息可以跨越取得佇列進行配送。

如果佇列管理程式已存在，則會使用它們。如果佇列管理程式不存在，則會建立並使用它們。

- c. 輸入下列指令，啟動 WebSphereMQ 指令處理器：

```
runmqsc getQueueManager
```

- d. 若為複雜的設定，建議輸入下列 MQ 指令，啓用佇列管理程式的遠端管理：
- e. 如果這個佇列管理程式將是 WebSphere MQ 叢集的儲存庫，請輸入 MQ 指令：
- f. 輸入下列 MQ 指令，將佇列管理程式的傳送端及接收端通道定義至每一個不在這個伺服器上管理的儲存庫。對於每一個叢集接收端通道：

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
  CHLTYPE(CHLTYPE) +
  CLUSTER('clusterName') +
  CLUSNL(' ') +
  CONNAME('repositoryIP-Address(port)') +
  DESCR('Cluster receiver channel at repositoryQueueManager TCP/IP') +
  MAXMSGL(4194304) +
  TRPTYPE(TCP) +
  MCAUSER('principal') +
  REPLACE
```

對於每一個叢集傳送端通道：

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
  CHLTYPE(CLUSSDR) +
  CONNAME('repositoryIP-Address(port)') +
  CLUSTER('clusterName') +
  CLUSNL(' ') +
  DESCR('Cluster sender channel to repositoryQueueManager TCP/IP') +
  MAXMSGL(4194304) +
  TRPTYPE(TCP) +
  MCAUSER('targetPrincipal') +
  REPLACE +
  NPMSPEED (NORMAL)
```

其中：

*repositoryQueueManager*

管理儲存庫的佇列管理程式名稱。

*clusterName*

WebSphere MQ 叢集的名稱，所有佇列管理程式都是該叢集的成員。

*repositoryIP-Address*

儲存庫佇列管理程式所在節點的 IP 位址。

*port* 儲存庫佇列管理程式正在使用的 IP 埠。

*principal*、*targetPrincipal*

針對接收端及傳送端通道使用的 MCAUSER。如需此值的相關資訊，請參閱 WebSphere MQ 文件。

- g. 對於每一個佇列管理程式，請輸入 MQ 指令，啓動接聽器：

```
runmqtsr -t tcp -p port -m QueueManager
```

2. 如果您的 WebSphere 叢集由 Windows 節點組成，請在每一個節點上執行下列動作：
  - a. 請確定使用者 ID 具有建立 WebSphere MQ 佇列的權限。
  - b. 輸入下列指令，建立「取得」佇列管理程式、使它成為 WebSphere MQ 叢集的成員，並建立佇列：

```
cd install_root\ProcessChoreographer\config
createQueues.bat getQueueManager clusterName putQueueManager
```

其中：

*getQueueManager*

給與取得佇列管理程式的唯一名稱。這個佇列管理程式管理所有本端佇列。

*clusterName*

WebSphere MQ 叢集的名稱，所有佇列管理程式都是該叢集的成員。

*putQueueManager*

放置佇列管理程式的唯一名稱。這個佇列管理程式不管理任何佇列，以確保訊息可以跨越取得佇列進行配送。

如果佇列已存在，則會加以使用。如果佇列不存在，則會建立並使用它們。

- c. 輸入下列指令，啟動 WebSphereMQ 指令處理器：

```
runmqsc queueManager
```

- d. 若為複雜的設定，建議輸入下列 MQ 指令，啟用佇列管理程式的遠端管理：

```
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') TYPE(CHLTYPE)
```

- e. 如果這個佇列管理程式將是 WebSphere MQ 叢集的儲存庫，請輸入 MQ 指令：

```
ALTER QMGR REPOS('clusterName') REPOSNL(' ')
```

- f. 輸入下列 MQ 指令，將佇列管理程式的傳送端及接收端通道定義至每一個不在這個伺服器上管理的儲存庫。對於每一個叢集接收端通道：

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
  CHLTYPE(CLUSRCVR) +
  CLUSTER('clusterName') +
  CLUSNL(' ') +
  CONNAME('repositoryIP-Address(port)') +
  DESCR('Cluster receiver channel at repositoryQueueManager TCP/IP') +
  MAXMSGL(4194304) +
  TRPTYPE(TCP) +
  MCAUSER('principal') +
  REPLACE
```

對於每一個叢集傳送端通道：

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
  CHLTYPE(CLUSSDR) +
  CONNAME('repositoryIP-Address(port)') +
  CLUSTER('clusterName') +
  CLUSNL(' ') +
  DESCR('Cluster sender channel to repositoryQueueManager TCP/IP') +
  MAXMSGL(4194304) +
  TRPTYPE(TCP) +
  MCAUSER('principal') +
  REPLACE +
  NPMSPEED (NORMAL)
```

其中：

*repositoryQueueManager*

管理儲存庫的佇列管理程式名稱。

*clusterName*

WebSphere MQ 叢集的名稱，所有佇列管理程式都是該叢集的成員。

*repositoryIP-Address*

儲存庫佇列管理程式所在節點的 IP 位址。

*port* 儲存庫佇列管理程式正在使用的 IP 埠。

*principal*

要使用的 MCAUSER。如需此值的相關資訊，請參閱 WebSphere MQ 文件。

g. 對於每一個佇列管理程式，請輸入 MQ 指令，啟動接聽器：

```
runmqtsr -t tcp -p port -m QueueManager
```

3. 選擇性的: 若要驗證伺服器上的通道狀態，請輸入 MQ 指令：

```
display chstatus(*)
```

佇列管理程式、佇列、叢集、儲存庫、通道及接聽器都存在。

## 概觀：配置商業程序編排器瀏覽器

「商業程序編排器瀏覽器」提供用來管理商業程序及處理人力作業的使用者介面。它是 Java 2 Enterprise Edition (J2EE) Web 應用程式，基礎為 JavaServer Faces (JSF) 技術及「商業程序編排器瀏覽器」元件。

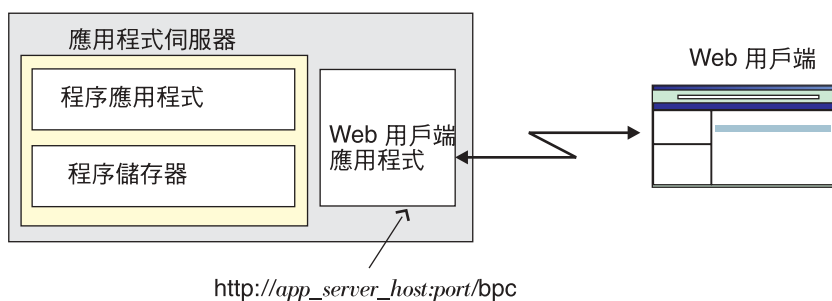
- 第 110 頁的『關於商業程序編排器瀏覽器』
- 第 183 頁的『配置商業程序編排器瀏覽器』

### 關於商業程序編排器瀏覽器

「商業程序編排器瀏覽器」是 Web 應用程式，實作與商業程序及人力作業互動的一般 Web 使用者介面。

您可以在伺服器或叢集上配置一或多個「商業程序編排器瀏覽器」實例。具有 WebSphere Process Server 設定檔的 WebSphere Process Server 安裝，或 WebSphere Process Server 用戶端安裝就已足夠；並不需要在伺服器或叢集上配置「商業程序編排器」。用戶端安裝是將用戶端連接至 WebSphere Process Server 時唯一需要的基礎架構，但是它不包含「商業程序編排器瀏覽器」。然而，如果您擁有部署管理程式，則也可以將「商業程序編排器瀏覽器」安裝在 WebSphere Process Server 用戶端安裝的伺服器上。

單一「商業程序編排器瀏覽器」只可以連接至一個「商業程序編排器」配置，而不需要連接至本端配置。然而，您可以在相同伺服器或叢集上配置多個「商業程序編排器瀏覽器」實例，且每個實例都可以連接至不同的「商業程序編排器」配置。



當您啟動「商業程序編排器瀏覽器」時，您在使用者介面中看到的物件及採取的動作視您所屬的使用者群組及授權給該群組的權限而定。例如，如果您是商業程序管理者，則負責已部署商業程序的順利運行。您可以檢視程序及作業範例、程序實例、作業實例及其關聯物件的相關資訊。您也可以處理這些物件；例如，您可以啟動新的程序實例、建立及啓動作業、修復及重新啓動失敗的活動、管理工作項目以及刪除整個程序實例及作業實例。然而，如果您是使用者，則只可以檢視及處理那些指派給您的作業。



## 配置商業程序編排器瀏覽器

您可以執行 Script 或使用管理主控台來配置「商業程序編排器瀏覽器」。

您已配置「商業程序編排器」。

適用下列一項以上：

- 您尚未安裝「商業程序編排器瀏覽器」。
- 您想要管理現有的「商業程序編排器」配置。
- 您想要將另一個「商業程序編排器瀏覽器」實例新增至已管理的「商業程序編排器」配置。

若要配置「商業程序編排器瀏覽器」，請執行下列其中一項：

- 如果想要使用 Script，請執行『使用 clientconfig.jacl Script 檔配置商業程序編排器瀏覽器』。
- 如果想要使用管理主控台，請執行『使用管理主控台來配置商業程序編排器瀏覽器』。

「商業程序編排器瀏覽器」已完成配置，隨時可以使用。

啓動「商業程序編排器瀏覽器」。

## 使用管理主控台來配置商業程序編排器瀏覽器

您可以使用管理主控台來配置「商業程序編排器瀏覽器」。

1. 按一下**伺服器** → **應用程式伺服器** → **server\_name** 或 **伺服器** → **叢集** → **cluster\_name**，然後在**商業整合**之下，展開**商業程序編排器**，再按一下**商業程序編排器瀏覽器**。
2. 若要建立新的瀏覽器配置，請按一下**新增**。
3. 輸入下列欄位的值：
  - 在部署目標伺服器或叢集上，**環境定義根目錄**必須是唯一的。
  - **瀏覽器搜尋結果限制**。
  - **受管理的「商業程序編排器」儲存器**

如需這些欄位的相關資訊，請參閱第 121 頁的『商業程序編排器瀏覽器設定』。

4. 按一下**套用**。會顯示訊息以指出進度。
5. 選擇性的: 如果報告任何問題，請檢查 SystemOut.log 檔案。
6. 啓動名稱爲 BPCEplorer\_scope 的企業應用程式。其中 scope 識別您已配置「商業程序編排器瀏覽器」的伺服器或叢集。

「商業程序編排器瀏覽器」已完成配置，隨時可以使用。

啓動「商業程序編排器瀏覽器」。

## 使用 clientconfig.jacl Script 檔配置商業程序編排器瀏覽器

此 Script 檔會在伺服器或叢集上配置「商業程序編排器瀏覽器」及所有必要的資源。

## 目的

此 Script 檔會配置「商業程序編排器瀏覽器」。此 Script 檔可以使用互動方式或批次模式執行。

## 位置

clientconfig.jacl Script 檔位於「商業程序編排器」config 目錄中：

- 在 Linux、UNIX 及 i5/OS 平台上：位於目錄 *install\_root/ProcessChoreographer/config* 中
- 在 Windows 平台上：位於目錄 *install\_root\ProcessChoreographer\config* 中

## 在獨立式伺服器環境中執行 Script

在獨立式伺服器環境中：

- 只有當應用程式伺服器不在執行中時，才會併入 `-conntype NONE` 選項。
- 如果伺服器正在執行，而且已啓用 WebSphere 管理安全，則併入 `-user` 及 `-password` 選項。
- 如果您沒有配置預設設定檔，則會新增 `-profileName` 選項。

## 在網路部署環境中執行 Script

在網路部署環境中：

- 在部署管理程式節點上執行 Script。
- 只有當部署管理程式不在執行中時，才會併入 `-conntype NONE` 選項。
- 如果啓用 WebSphere 管理安全，則併入 `-user` 及 `-password` 選項。
- 如果您沒有配置預設設定檔，則會新增 `-profileName` 選項。

## 以非互動方式配置商業程序編排器瀏覽器

請將現行目錄變更為 *install\_root*，並執行下列作業：

在 Linux 及 UNIX 平台上，輸入下列指令：

```
bin/wsadmin.sh -f ProcessChoreographer/config/clientconfig.jacl options
```

在 i5/OS 上，輸入下列指令：

```
bin/wsadmin -f ProcessChoreographer/config/clientconfig.jacl options
```

在 Windows 平台上，請輸入指令：

```
bin\wsadmin.bat -f ProcessChoreographer/config/clientconfig.jacl options
```

其中 *options* 如下：

```
( [-user userName][-password password] | [-conntype NONE] )  
  [-profileName profileName]  
( [-node nodeName][-server serverName] )  
  [-cluster clusterName]  
  [-contextRootExplorer explorerContextRoot]  
  [-hostName explorerVirtualHostname]  
  [-precompileJSPs { yes | no }]
```

```
( ( [-remoteNode nodeName] [-remoteServer serverName] )
  | [-remoteCluster clusterName] )
  [-maxListEntries maximum]
  [-explorerHost explorerURL]
```

**註:** 如果您在指令行上提供所有必要參數，就不會再提示您輸入它們。在指令行上沒有指定任何必要的參數時，會以其列出順序以互動方式提示您。

## 參數

當使用 `wsadmin` 呼叫 `Script` 時，您可以使用下列參數：

### **-node** *nodeName*

其中，*nodeName* 是要在其中配置「商業程序編排器瀏覽器」的節點名稱。如果未指定此參數，則預設值是本端節點。

### **-server** *serverName*

其中，*serverName* 是要在其中配置「商業程序編排器瀏覽器」的伺服器名稱。如果您只有一個節點，且只有一個伺服器，則此參數是選用參數。

### **-cluster** *clusterName*

其中，*clusterName* 是要在其中配置「商業程序編排器瀏覽器」的叢集名稱。這是選用參數。請不要在獨立式伺服器環境中指定這個選項，也不要指定節點及伺服器時指定這個選項。

### **-contextRootExplorer** *contextRootExplorer*

其中，*contextRootExplorer* 是「商業程序編排器瀏覽器」的環境定義根目錄。在 WebSphere Cell 內，環境定義根目錄必須是唯一的。預設值是 `/bpc`。

### **-hostName** *VirtualHostname*

其中 *VirtualHostname* 是虛擬主機，「商業程序編排器」以及「商業流程管理程式」和「人力作業管理程式」API 的 Web 服務連結會在其上執行。預設值是 `default_host`。

### **-precompileJSPs** { *no* | *yes* }

判定 Java Server Pages (JSP) 是否經過前置編譯。預設值是 `no`。請注意，無法除錯經過前置編譯的 JSP。

### **-remoteNode** *nodeName*

如果不需要連接至本端「商業程序編排器」配置，請使用此參數及 `remoteServer`。如果未指定此參數，則會預設為 `-node` 參數的值。

### **-remoteServer** *serverName*

如果不需要連接至本端「商業程序編排器」配置，請使用此參數及 `remoteNode`。如果未指定此參數，則會預設為 `-server` 參數的值。

### **-remoteCluster** *clusterName*

如果不需要連接至本端「商業程序編排器」配置，且未指定 `remoteNode` 及 `remoteServer`，請使用此參數。如果未指定此參數，則會預設為 `-cluster` 參數的值。

### **-maxListEntries** *maximum*

其中 *maximum* 是「商業程序編排器瀏覽器」針對查詢所傳回的結果數上限。預設值是 10000。

### **-explorerHost** *explorerURL*

其中，*explorerURL* 是「商業程序編排器瀏覽器」的 URL。此參數的值是用來將受

管理「商業程序編排器」配置的「人力作業管理程式」，鏈結至此特定「商業程序編排器瀏覽器」實例。此參數預設為空字串，這表示不會建立鏈結。您可以稍後使用管理主控台來建立或變更鏈結。

## 日誌檔

如果您在使用 `clientconfig.jacl` Script 檔建立配置時發生問題，請檢查下列日誌檔：

- `clientconfig.log`
- `wsadmin.traceout`

兩個檔案都可以在設定檔的 `logs` 目錄中找到：中

- 在 Linux、UNIX 及 i5/OS 平台上：在目錄 `profile_root/logs` 中
- 在 Windows 平台上：在目錄 `profile_root\logs` 中

如果是以連線模式執行 Script，則也請檢查檔案 `SystemOut.log` 及 `SystemErr.log`，這些檔案可以在根據 `wsadmin` Script 用戶端所連接應用程式伺服器或部署管理程式命名的 `logs` 目錄的子目錄中找到。

---

## 配置商業程序編排器觀察程式

使用「商業程序編排器觀察程式」是選用項目，但在使用之前，您必須先設定資料庫並安裝應用程式。

您已執行第 111 頁的『關於商業程序編排器觀察程式』。

您想要配置具有其專屬資料庫的「商業程序編排器觀察程式」。

1. 執行第 187 頁的『準備商業程序編排器觀察程式的資料庫』
2. 執行第 217 頁的『配置商業程序編排器事件收集程式應用程式』。
3. 執行第 223 頁的『配置商業程序編排器觀察程式應用程式』。
4. 執行第 226 頁的『變更商業程序編排器觀察程式的配置參數』。
5. 執行第 225 頁的『啓用商業程序編排器的記載功能』。
6. 執行第 234 頁的『驗證商業程序編排器觀察程式』。

「商業程序編排器觀察程式」已配置且在運作中。

您可以使用「商業程序編排器觀察程式」來產生報告，如第 308 頁的『報告商業程序及活動』所述。

## 移除商業程序編排器觀察程式 6.0.1 版範例

說明如何移除 6.0.1 版隨附的「商業程序編排器觀察程式」範例。

如果您從 WebSphere Process Server 6.0.1 版升級，且使用「商業程序編排器觀察程式」範例，您必須先移除範例，才能配置最新版的「商業程序編排器觀察程式」。應用程式本身不會移轉，但資料庫視圖和索引必須加以捨棄。範例所收集的資料不會移轉，也無法使用。

捨棄「商業程序編排器觀察程式」和事件收集程式所使用的資料庫視圖和索引。

1. 如果使用 DB2 資料庫：
  - a. 連接到觀察程式綱目所在的資料庫。

- b. 尋找並執行 Script，以捨棄觀察程式範例 DB2 綱目：
  - 在 Windows 平台上：`install_root\ProcessChoreographer\sample\observer\dropObserverSampleSchema_DB2.sql`。
  - 在 Linux 及 UNIX 平台上：`install_root/ProcessChoreographer/sample/observer/dropObserverSampleSchema_DB2.sql`。

例如，輸入：

```
db2 -tf dropObserverSampleSchema_DB2.sql
```

2. 如果您使用 Cloudscape™ 資料庫，請在指令視窗中執行下列動作：
  - a. 將 `derby.jar` 及 `derbytools.jar` 檔案新增至 CLASSPATH 中。
    - 在 Windows 平台上：它們位於 `install_root\derby\bin\embedded` 中。
    - 在 Linux 及 UNIX 平台上：它們位於 `install_root/derby/bin/embedded` 中。
  - b. 尋找並執行 Script，以捨棄觀察程式範例 Cloudscape 綱目，如檔案中的備註所述：
    - 在 Windows 平台上：`install_root\ProcessChoreographer\sample\observer\dropObserverSampleSchema_Cloudscape.sql`。
    - 在 Linux 及 UNIX 平台上：`install_root/ProcessChoreographer/sample/observer/dropObserverSampleSchema_Cloudscape.sql`。

例如，輸入此指令：

```
java -Dij.protocol=jdbc:derby:  
-Dij.database=OBSVRDB  
org.apache.derby.tools.ij  
dropObserverSampleSchema_Cloudscape.sql
```

已移除「商業程序編排器觀察程式」範例。

## 準備商業程序編排器觀察程式的資料庫

為您的資料庫執行動作。

### 準備商業程序編排器觀察程式的 DB2 Universal 資料庫

您可以使用 Script 或互動式工具來準備此資料庫。

#### 使用 SQL Scripts 來準備商業程序編排器觀察程式的 DB2 Universal 資料庫:

本文說明如何使用 `createTablespace_Observer.sql` 及 `createSchema_Observer.sql` Script，在 Linux、UNIX 及 Windows 平台上準備 DB2 Universal 資料庫。

您的資料庫必須已存在。您可以使用現有的資料庫，或根據資料庫文件來建立新的資料庫。若要執行這項作業，您必須具有目的地資料庫的管理權限。

1. 切換至資料庫的配置 Script 所在的「商業程序編排器」子目錄。
  - 在 Linux 及 UNIX 平台上，切換至 `install_root/dbscripts/ProcessChoreographer/DB2` 目錄。
  - 在 Windows 平台上，切換至目錄 `install_root\dbscripts\ProcessChoreographer\DB2`。
2. 將所有 `*Observer.sql` Script 檔複製到資料庫伺服器。
3. 在資料庫伺服器上，切換至您複製 Script 檔的目錄。
4. 建立表格空間：

- a. 編輯 createTablespace\_Observer.sql Script 檔 (根據檔案頂端的指示)。
  - b. 執行表格空間建立 Script 檔，請輸入指令：  
db2 -tf createTablespace\_Observer.sql
  - c. 請確定 Script 輸出沒有錯誤。如果發生錯誤，您可以使用 dropTablespace\_Observer.sql Script 檔來捨棄表格空間。
5. 建立綱目 (表格、索引及視圖)。
- a. 根據檔案開頭的指示，編輯 createSchema\_Observer.sql Script 檔。
  - b. 在 DB2 指令行處理器中，輸入下列指令：  
db2 -tf createSchema\_Observer.sql
  - c. 請確定 Script 輸出沒有錯誤。如果您要捨棄綱目，請使用 dropSchema\_Observer.sql Script 檔。
6. 如果您想要使用 Java 實作而非 SQL 實作的「商業程序編排器觀察程式」UDF，請執行第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』。
7. 使用管理主控台，建立指向資料庫的 XA 資料來源，並測試連線。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

#### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

### 使用 setupEventCollector 工具來準備商業程序編排器觀察程式的 DB2 Universal 資料庫:

本文說明如何使用互動式功能表導向工具及 createTablespace\_Observer.sql script，在 Linux、UNIX 及 Windows 平台上準備 DB2 Universal 資料庫。

您的資料庫必須已存在，且必須對它具有管理權限。

1. 如果您使用第二類型 JDBC 連線：
  - a. 準備指令行環境：
    - 在 Linux 及 UNIX 平台上，對 DB2 實例執行 db2profile。
    - 在 Windows 上，開啓 DB2 指令視窗。
  - b. 如果您的資料庫在遠端，請在本端 DB2 實例上編目資料庫。
2. 建立表格空間：
  - a. 切換至資料庫的 SQL Script 所在的「商業程序編排器」子目錄。
    - 在 Linux 及 UNIX 平台上，切換至 `install_root/dbscripts/ProcessChoreographer/DB2` 目錄。
    - 在 Windows 平台上，切換至目錄 `install_root\dbscripts\ProcessChoreographer\DB2`。



- b. 製作 createTablespace\_Observer.sql Script 檔的副本。
- c. 根據檔案開頭的指示，編輯 createTablespace\_Observer.sql Script 檔的副本。
- d. 執行表格空間建立 Script 檔，請輸入指令：

```
db2 -tf createTablespace_Observer.sql
```

- e. 請確定 Script 輸出沒有錯誤。如果發生錯誤，您可以使用 dropTablespace\_Observer.sql Script 檔來捨棄表格空間。
3. 如果您使用的使用者 ID 不是資料庫管理者，請確定它具有下列許可權：  
GRANT CREATETAB, CONNECT, CREATE\_EXTERNAL\_ROUTINE ON DATABASE  
TO USER *user\_name*;  
GRANT USE OF TABLESPACE *tablespace\_name* TO USER *user\_name*;

其中，*user\_name* 是使用者 ID，而 *tablespace\_name* 是所有觀察程式表格空間名稱的清單，在 createTablespace\_Observer.sql Script 中可以找到這些名稱。

4. 切換至配置 Script 所在的「商業程序編排器」目錄。

在 Linux 及 UNIX 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\config
```

5. 啓動此工具以設定事件收集程式，如第 230 頁的『setupEventCollector 工具』中所述。
6. 準備資料庫：

- a. 當您看見：

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

選取選項 1，以準備事件收集程式和觀察程式應用程式的資料庫。

- b. 當您看見：

```
Prepare a database for the WebSphere Business Process Choreographer  
Event Collector and Observer
```

```
Select the type of your database provider:
```

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle

0) Exit Menu

輸入 d 以選取 DB2 Universal。

- c. 此工具可讓您建立 SQL 檔案，交由資料庫管理者執行，而不是以現行使用者 ID 執行。當您看見：

Do you want to create an SQL file only (delay database preparation)?  
y) yes  
n) no

- 如果您不想要延遲執行 SQL，請輸入 n。
- 如果您想要延遲執行 SQL，請輸入 y。您會看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?  
y) yes  
n) no

- 如果您想要在資料庫內檢查所輸入的值，請輸入 y。
- 否則，請輸入 n。

視您輸入的指令而定，可能不會看到下列所有提示。跳過您沒看到的步驟。

- d. 如果您看見：

Specify the JDBC driver type to be used:

- 2) Connect using type 2 (using a native database client)
- 4) Connect using type 4 (directly via JDBC)

指定 JDBC 驅動程式類型：

- 如果您使用原生資料庫用戶端，請輸入 2。
- 否則，請輸入 4，以選取第四類型 JDBC 驅動程式。

- e. 如果您看到下列其中一個提示：

Specify the name of your database: [BPEDB]  
Specify the name of database in local catalog: [BPEDB]

資料庫名稱或別名。

**註：**預設值 BPEDB 是「商業程序編排器」所使用的相同資料庫。若為高效能系統，您應該使用不同的資料庫。

- f. 如果您看見：

Specify the hostname of the database server: [localhost]

輸入資料庫伺服器的主機名稱或 IP 位址。

- g. 如果您看見：

Specify the port where the database server is listening: [50000]

輸入資料庫伺服器的埠號。

- h.

Specify the directory of your JDBC driver: [D:\opt\SQLLIB\java]

輸入 JDBC 驅動程式的 db2jcc.jar 及 db2jcc\_license\_cu.jar JAR 檔所在的目錄。

- i. 如果您看見：

Specify userid to connect to the database 'database\_name' [db2admin] :  
Specify the password for userid 'user\_ID' :

輸入使用者 ID 及密碼，以連接資料庫。畫面上會顯示下列內容：

Trying to connect to database 'database\_name', using user 'user\_ID'  
Connected to 'database\_name'

j. 如果您看見：

Specify the database schema to be used. [user\_ID]：

輸入用於資料庫物件的資料庫綱目 (集合名稱)。如果您輸入空格字元或將欄位保留空白，則會使用使用者 ID 的綱目。

k. 當您看見：

Choose the implementation of the Observer user-defined functions.

Note: The Java UDFs are more precise, but they require a jar file installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果您想要使用更精確的 Java 型使用者定義函數 (UDF)，這需要在資料庫中安裝 JAR 檔，請輸入 1。
- 如果您要使用較不精確的 SQL 型 UDF，請輸入 2。

您會看到類似下列內容的項目：

```
Checking for required tablespace(es) ['OBSVRTS']
All required tablespaces were found.
Loading the jar file 'install_root\lib\bpcodbutil.jar' into the database.
The jar file 'install_root\lib\bpcodbutil.jar' was successfully installed.
```

The setup of the database completed successfully.

7. 使用管理主控台，建立指向資料庫的 XA 資料來源，並測試連線。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

#### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

## 準備商業程序編排器觀察程式的 DB2 for iSeries 資料庫

您可以使用 Script 或互動式工具來準備此資料庫。

### 使用 SQL Script 來準備商業程序編排器觀察程式的 DB2 for iSeries 資料庫:

i5/OS qshell 環境中的 本文說明如何使用 createSchema\_Observer.sql script，用來準備 DB2 for iSeries 資料庫。

您的集合必須已存在。您可以使用現有的集合，或根據資料庫文件來建立新的集合。

1. 在 qshell 環境中，切換至資料庫的配置 Script 所在的「商業程序編排器」子目錄。  
輸入：

```
cd install_root/dbscripts/ProcessChoreographer/DB2iSeries
```

2. 將所有 \*Observer.sql Script 檔複製到資料庫伺服器。
3. 在資料庫伺服器上，切換至您複製 Script 檔的目錄。
4. 建立綱目 (表格、索引及視圖)。
  - a. 根據檔案開頭的指示，編輯 createSchema\_Observer.sql Script 檔。
  - b. 在 DB2 指令行處理器中或在 qshell 中，輸入下列指令：
 

```
db2 -tf createSchema_Observer.sql
```
  - c. 請確定 Script 輸出沒有錯誤。如果您要捨棄綱目，請使用 dropSchema\_Observer.sql Script 檔。
5. 如果您想要使用 Java 實作的必要使用者定義函數 (UDF)，請執行第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』。
6. 使用管理主控台，建立指向資料庫的 XA 資料來源，並測試連線。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

#### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

### 使用 setupEventCollector 工具來準備商業程序編排器觀察程式的 DB2 for iSeries 資料庫：

本文說明如何使用互動式功能表導向工具，從 i5/OS qshell 環境內準備 DB2 for iSeries 資料庫。

您可以使用現有的集合，或根據資料庫文件來建立新的集合。

1. 啟動 qshell 環境。
2. 切換至配置 Script 所在的「商業程序編排器」目錄。輸入：
 

```
cd install_root/ProcessChoreographer/config
```
3. 啟動此工具以設定事件收集程式，如第 230 頁的『setupEventCollector 工具』中所述。
4. 準備資料庫：
  - a. 當您看見：
    - 1) Prepare a database for the Event Collector and Observer
    - 2) Install the Event Collector application
    - 3) Remove the Event Collector application and related objects
    - 4) Change configuration settings of an installed Event Collector
    - 5) Drop the database schema of the Event Collector and Observer
    - 6) Administer Observer related user-defined functions
  - 0) Exit Menu

選取選項 1，以準備事件收集程式和觀察程式應用程式的資料庫。

- b. 當您看見：

Prepare a database for the WebSphere Business Process Choreographer Event Collector and Observer

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle
  
- 0) Exit Menu

輸入 i 以選取 DB2 for iSeries。

- c. 此工具可讓您建立 SQL 檔案，交由資料庫管理者執行，而不是以現行使用者 ID 執行。當您看見：

Do you want to create an SQL file only (delay database preparation)?

- y) yes
- n) no

- 如果您不想要延遲執行 SQL，請輸入 n。
- 如果您想要延遲執行 SQL，請輸入 y。您會看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?

- y) yes
- n) no

- 如果您想要在資料庫內檢查所輸入的值，請輸入 y。
- 否則，請輸入 n。

視您輸入的指令而定，可能不會看到下列所有提示。跳過您沒看到的步驟。

- d. 如果您看見：

Specify the JDBC driver to be used:

- 1) Connect using the IBM Toolbox for Java JDBC driver
- 2) Connect using the native JDBC driver

Your selection: [2]

- 如果您要配置遠端資料庫，請輸入 1，以選取 IBM Toolbox for Java JDBC 驅動程式。
- 如果您要配置本端資料庫，請輸入 2，以選取原生 JDBC 驅動程式。

- e. 如果您看見：

Specify the name of database in local catalog: [\*LOCAL]

或

Specify the name of your database: [\*SYSBAS]

輸入服務 ID，或接受預設值。

- f. 如果您看見：

Specify the hostname of the database server: [localhost]

輸入資料庫伺服器的主機名稱或 IP 位址。

- g. 如果您看見：

Specify the directory of your JDBC driver:  
[/QIBM/ProdData/HTTP/Public/jt400/lib]

輸入 JDBC 驅動程式 JAR 檔所在的目錄。

- 若為原生驅動程式 (db2\_classes.zip)，這通常是 /QIBM/ProdData/Java400/ext。
- 若為工具箱驅動程式 (jt400.jar)，這通常是 /QIBM/ProdData/HTTP/Public/jt400/lib

h. 如果您看見：

Specify userid to connect to the database 'database\_name' [db2admin] :  
Specify the password for userid 'user\_ID' :

輸入使用者 ID 及密碼，以連接資料庫。

i. 如果您看見：

Specify the database schema to be used. [user\_ID] :

輸入用於資料庫物件的資料庫綱目 (集合名稱)。您必須指定已存在的綱目。如果您輸入空格字元或將欄位保留空白，則會使用使用者 ID 的綱目。

j. 如果您看見：

Note: The Java UDFs are more precise, but they require a jar file installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果您想要使用更精確的 Java 型使用者定義函數 (UDF)，這需要在資料庫中安裝 JAR 檔，請輸入 1。
- 如果您要使用較不精確的 SQL 型 UDF，請輸入 2。

k. 當資料庫完成準備之後，畫面上會顯示下列訊息：

The setup of the database completed successfully.

5. 如果您使用不同的資料庫 (不是 BPEDB)，請使用管理主控台來建立指向資料庫的 XA 資料來源，並測試連線。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

使用 setupEventCollector 工具，從遠端系統準備 DB2 for iSeries 資料庫：



本文說明如何使用互動式功能表導向工具，從遠端 Linux、Windows 或 UNIX 系統，準備「商業程序編排器觀察程式」的 DB2 for iSeries 資料庫。

您可以使用現有的集合，或根據資料庫文件來建立新的集合。要使用的集合必須已存在。

1. 若要從遠端準備資料庫，您需要下載 IBM Toolbox JDBC 驅動程式，以連接至 iSeries 電腦。下載之後，請記下 Jar 檔 jt400.jar 的位置。
2. 啟動指令行環境。
3. 切換至配置 Script 所在的「商業程序編排器」目錄。
  - 在 Windows 平台上，輸入：  
`cd install_root\ProcessChoreographer\config`
  - 在 Linux 及 UNIX 平台上，輸入：  
`cd install_root/ProcessChoreographer/config`
4. 啟動此工具以設定事件收集程式，如第 230 頁的『setupEventCollector 工具』中所述。
5. 準備資料庫：
  - a. 當您看見：
    - 1) Prepare a database for the Event Collector and Observer
    - 2) Install the Event Collector application
    - 3) Remove the Event Collector application and related objects
    - 4) Change configuration settings of an installed Event Collector
    - 5) Drop the database schema of the Event Collector and Observer
    - 6) Administer Observer related user-defined functions
  
    - 0) Exit Menu

選取選項 1，以準備事件收集程式和觀察程式應用程式的資料庫。

- b. 當您看見：

Prepare a database for the WebSphere Business Process Choreographer  
Event Collector and Observer

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle
  
- 0) Exit Menu

輸入 i 以選取 DB2 for iSeries。

- c. 此工具可讓您建立 SQL 檔案，交由資料庫管理者執行，而不是以現行使用者 ID 執行。當您看見：

Do you want to create an SQL file only (delay database preparation)?

  - y) yes
  - n) no
  - 如果您不想要延遲執行 SQL，請輸入 n。
  - 如果您想要延遲執行 SQL，請輸入 y。您會看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?  
y) yes  
n) no

- 如果您想要在資料庫內檢查所輸入的值，請輸入 y。
- 否則，請輸入 n。

視您輸入的指令而定，可能不會看到下列所有提示。跳過您沒看到的步驟。

d. 如果您看見：

Specify the name of your database: [\*SYSBAS]

輸入服務 ID，或接受預設值。

e. 如果您看見：

Specify the hostname of the database server: [localhost]

輸入資料庫伺服器的主機名稱或 IP 位址。

f. 如果您看見：

Specify the directory of your JDBC driver:  
[/QIBM/ProdData/HTTP/Public/jt400/lib]

輸入您下載 JDBC 驅動程式檔案 jt400.jar 的目錄。

g. 如果您看見：

Specify userid to connect to the database 'database\_name' [db2admin] :  
Specify the password for userid 'user\_ID' :

輸入使用者 ID 及密碼，以連接資料庫。

h. 如果您看見：

Specify the database schema to be used. [user\_ID] :

輸入用於資料庫物件的資料庫綱目 (集合名稱)。您必須指定已存在的綱目。如果您輸入空格字元或將欄位保留空白，則會使用使用者 ID 的綱目。

i. 如果您看見：

Note: The Java UDFs are more precise, but they require a  
jar file installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果您想要使用更精確的 Java 型使用者定義函數 (UDF)，這需要在資料庫中安裝 JAR 檔，請輸入 1。
- 如果您要使用較不精確的 SQL 型 UDF，請輸入 2。

j. 當資料庫完成準備之後，畫面上會顯示下列訊息：

The setup of the database completed successfully.

6. 如果您使用不同的資料庫 (不是 BPEDB)，請使用管理主控台來建立指向資料庫的 XA 資料來源，並測試連線。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

## 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

## 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

## 準備商業程序編排器觀察程式的 DB2 for z/OS 資料庫

您可以從遠端或在「UNIX 系統服務」內準備此資料庫。

### 在 USS 中建立商業程序編排器觀察程式的 DB2 for z/OS 資料庫:

本文說明如何使用互動式功能表導向工具 及 createTablespace\_Observer.sql Script (位於 z/OS 機器上的「UNIX 系統服務 (USS)」中) 來建立 DB2 for z/OS 資料庫。

#### 1. 準備 DB2 環境：

- a. 登入原生的 z/OS 環境。
- b. 如果安裝多個 DB2 系統，請決定您要使用的子系統。
- c. 記下 DB2 子系統所接聽的 IP 埠。
- d. 決定子系統的位置名稱。若要找出位置名稱，請檢查「DB2 系統」畫面，或針對您的子系統選取 DB2 管理功能表選項執行 **SQL 陳述式**，並輸入下列 SQL 查詢：

```
select current server from sysibm.sysdummy1
```
- e. 建立儲存體群組並記下名稱，例如 OBSVRSG。
- f. 如果您要使用新的資料庫，請建立新的資料庫，例如 OBSVRSG。需要的話，您可以重複使用現有的資料庫和儲存體群組，例如「商業程序編排器」資料庫 BPEDB。
- g. 決定要使用的綱目限定元 (\_SQLID)。
- h. 決定用來設定資料庫的使用者 ID *user\_ID*。這不是在執行時期用來存取資料庫的使用者 ID。
- i. 請確定使用者 ID 具有下列存取資料庫和儲存體群組的權限：
  - 儲存體群組的使用許可權。
  - 資料庫 OBSVRDB 的使用許可權。
  - 在資料庫 OBSVRDB 內建立表格空間的許可權。
  - 在資料庫 OBSVRDB 內建立表格的許可權。
- j. 如果您想要使用 Java 實作的「商業程序編排器」使用者定義函數 (UDF)，請確定該使用者 ID 也具有下列權限：
  - 在 SYSIBM.SYSJAROBJECTS 上選取的許可權。
  - 對綱目 SQLJ 執行下列儲存程序的權限：
    - INSTALL\_JAR
    - REMOVE\_JAR
    - REPLACE\_JAR

- DB2\_INSTALL\_JAR
  - DB2\_REMOVE\_JAR
  - DB2\_REPLACE\_JAR
  - 執行屬於集合 DSNJAR 套件的許可權。
- k. 如果您想要使用 Java 實作的「商業程序編排器」使用者定義函數 (UDF)，請準備 DB2 環境來執行 Java 使用者定義函數及解譯的 Java 常式。執行下列動作：
- 1) 啓用 DB2 提供的儲存程序和定義 DB2 Universal JDBC Driver 使用的表格
  - 2) 設定已解譯 Java 常式的環境
- 請記下此程序期間建立的 WLM 應用程式環境的名稱。

2. 登入 USS。

3. 建立表格空間：

- a. 切換至資料庫系統的「商業程序編排器觀察程式」資料庫 Script 所在的目錄。視您的 DB2 版本而定，輸入下列其中一個指令：

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV7
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV8
```

- b. 編輯 createTablespace\_Observer.sql Script。將 @STOGRP@ 取代為儲存體群組名稱，且將 @DBNAME@ 取代為資料庫名稱 (不是子系統名稱)。

**註:** 提供的 SQL 檔是 ASCII 格式。視您用來檢視、編輯或執行此檔案的工具而定，您可能需要將檔案轉換成 EBCDIC。若要將檔案轉換成 EBCDIC，請輸入下列指令：

```
iconv -t IBM-1047 -f ISO8859-1 createTablespace_Observer.sql > createTablespace_Observer_EBCDIC.sql
```

若要轉換回 ASCII，請輸入指令：

```
iconv -t ISO8859-1 -f IBM-1047 createTablespace_Observer_EBCDIC.sql > createTablespace_Observer_ASCII.sql
```

- c. 確定已連接您的資料庫，然後執行您自訂的 createTablespace\_Observer.sql Script 版本。

4. 啓動此工具以設定事件收集程式，如第 230 頁的『setupEventCollector 工具』中所述。

5. 準備資料庫：當您看見：

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

- a. 選取選項 1，準備事件收集程式應用程式的資料庫。
- b. 輸入 7 或 8，選取您的 DB2 on z/OS 版本號碼。
- c. 此工具可讓您建立 SQL 檔案，交由資料庫管理者執行，而不是以現行使用者 ID 執行。當您看見：

```
Do you want to create an SQL file only (delay database preparation)?
y) yes
n) no
```

- 如果您不想要延遲執行 SQL，請輸入 n。
- 如果您想要延遲執行 SQL，請輸入 y。您會看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?  
y) yes  
n) no

- 如果您想要在資料庫內檢查所輸入的值，請輸入 y。
- 否則，請輸入 n。

視您輸入的指令而定，可能不會看到下列所有提示。跳過您沒看到的步驟。

d. 如果您看見：

Specify the database location name:  
(as returned by SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1):

輸入資料庫的位置名稱。這是您在步驟 第 197 頁的 1d 中記下的值。

e. 如果您看見：

Specify the name of the database as known by the subsystem [*subsystem*]

輸入資料庫在 z/OS 主機上的子系統中的名稱。這是您在步驟 第 197 頁的 1f 中記下的值。

f. 如果您看見：

Specify the hostname of the z/OS DB2 database server: [*localhost*]

輸入資料庫伺服器的主機名稱。

g. 如果您看見：

Specify the port where the database subsystem is listening:

輸入資料庫子系統所使用的埠號。這是您在步驟 第 197 頁的 1c 中記下的值。

h. 如果您看見：

Specify userid to connect to the database '*database\_alias*' [*db2admin*] :

輸入使用者 ID 以連接資料庫。這是步驟 第 197 頁的 1h 所述的使用者 ID *user\_ID*。

i. 如果您看見：

Specify the password for userid '*user\_ID*' :

輸入使用者 ID 的密碼。

j. 如果您看見：

Trying to connect to database '*database\_alias*', using user '*user\_ID*'  
Connected to '*database\_alias*'  
Specify the database schema to be used. [*user\_ID*] :

輸入用於資料庫物件的資料庫綱目，或按 Enter 鍵來接受預設值，也就是用來連接資料庫的使用者 ID。這是綱目限定元 *\_SQLID*。

k. 如果您看見：

Note: The Java UDFs are more precise, but they require a jar file  
installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果您想要使用更精確的 Java 型使用者定義函數 (UDF)，這需要在資料庫中安裝 JAR 檔，請輸入 1。
- 如果您要使用較不精確的 SQL 型 UDF，請輸入 2。

l. 如果您看見：

Specify the DB2 storage group name to be used. [OBSVRSG] :

輸入步驟 第 197 頁的 1e 中的儲存體群組名稱，或按 Enter 鍵來接受預設值。

m. 如果您看見：

Specify the WLM environment name where the UDF should run. [] :

輸入在步驟 第 198 頁的 1k 中記下的 WLM 環境。

n. 在檢查必要的表格空間並載入 JAR 檔至資料庫之後，下列訊息指出順利完成：

The setup of the database completed successfully.

6. 使用管理主控台，建立指向資料庫的 XA 資料來源。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

#### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

#### 從遠端系統建立商業程序編排器觀察程式的 DB2 for z/OS 資料庫:

本文說明如何使用互動式功能表導向工具 及 createTablespace\_Observer.sql Script，在 Linux、UNIX 或 Windows 系統上建立 DB2 for z/OS 資料庫。

您必須已在 Linux、UNIX 或 Windows 伺服器上安裝 WebSphere Process Server。

1. 在資料庫所在的 z/OS 伺服器上：

- a. 登入原生的 z/OS 環境。
- b. 如果安裝多個 DB2 系統，請決定您要使用的子系統。
- c. 記下 DB2 子系統所接聽的 IP 埠。
- d. 建立儲存體群組並記下名稱，例如 OBSVRSG。
- e. 如果您要使用新的資料庫，請建立新的資料庫，例如 OBSVRSG。需要的話，您可以重複使用現有的資料庫和儲存體群組，例如「商業程序編排器」資料庫 BPEDB。
- f. 決定要使用的綱目限定元 (\_SQLID)。



- g. 決定用來設定資料庫的使用者 ID *user\_ID*。這不是在執行時期用來存取資料庫的使用者 ID。
  - h. 請確定使用者 ID 具有下列存取資料庫和儲存體群組的權限：
    - 儲存體群組的使用許可權。
    - 資料庫 OBSVRDB 的使用許可權。
    - 在資料庫 OBSVRDB 內建立表格空間的許可權。
    - 在資料庫 OBSVRDB 內建立表格的許可權。
  - i. 如果您想要使用 Java 實作的「商業程序編排器」使用者定義函數 (UDF)，請確定該使用者 ID 也具有下列權限：
    - 在 SYSIBM.SYSJAROBJECTS 上選取的許可權。
    - 對綱目 SQLJ 執行下列儲存程序的權限：
      - INSTALL\_JAR
      - REMOVE\_JAR
      - REPLACE\_JAR
      - DB2\_INSTALL\_JAR
      - DB2\_REMOVE\_JAR
      - DB2\_REPLACE\_JAR
    - 執行屬於集合 DSNJAR 套件的許可權。
  - j. 如果您想要使用 Java 實作的「商業程序編排器」使用者定義函數 (UDF)，請準備 DB2 環境來執行 Java 使用者定義函數及解譯的 Java 常式。執行下列動作：
    - 1) 啓用 DB2 提供的儲存程序和定義 DB2 Universal JDBC Driver 使用的表格
    - 2) 設定已解譯 Java 常式的環境

請記下此程序期間建立的 WLM 應用程式環境的名稱。
2. 在 WebSphere Process Server 所在的伺服器上：
- a. 安裝適合的 DB2 用戶端。

**註：**如果您規劃使用原生 DB2 用戶端來連接至遠端資料庫 (使用第二類型 JDBC 連線)，請確定您已安裝 DB2 Connect Gateway。DB2 Connect Gateway 是 DB2 UDB ESE 套件的一部分，但也可以分開安裝。

- b. 如果您使用原生 DB2 用戶端，請將遠端資料庫編目，並驗證您可以建立其連線。在 DB2 指令行視窗中輸入下列指令：

```
catalog tcpip node zosnode remote host_name server IP_port ostype mvs
catalog database location as database_alias at node zosnode
authentication dcs
catalog dcs database database_alias parms ',,INTERRUPT_ENABLED'
```

其中

*zosnode*

是遠端 z/OS 節點的本端別名。

*host\_name*

是遠端 z/OS 機器的 TCP/IP 位址或別名。

*IP\_port*

是 DB2 子系統所接聽的埠號。

### *database\_alias*

是用來存取遠端資料庫的本端別名。

### *location*

是遠端 DB2 位置名稱。若要找出位置名稱，請登入 TSO，使用其中一種可用的查詢工具，在選取的子系統上輸入下列 SQL 查詢。

```
select current server from sysibm.sysdummy1
```

若要驗證您可以連接至遠端系統，請輸入：

```
db2 connect to database_alias user userid using password
```

- c. 切換至資料庫系統的「商業程序編排器觀察程式」資料庫 Script 所在的目錄：
  - 在 Linux 及 UNIX 平台上，視您的 DB2 版本而定，輸入下列其中一個指令：

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV7
```

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV8
```
  - 在 Windows 平台上，視您的 DB2 版本而定，輸入下列其中一個指令：

```
cd install_root\dbscripts\ProcessChoreographer\DB2zOSV7
```

```
cd install_root\dbscripts\ProcessChoreographer\DB2zOSV8
```
- d. 編輯 createTablespace\_Observer.sql Script。將 @STOGRP@ 取代為儲存體群組名稱，且將 @DBNAME@ 取代為資料庫名稱 (不是子系統名稱)。
- e. 執行您自訂的 createTablespace\_Observer.sql Script 版本。如果您要捨棄表格空間，請使用 dropTablespace\_Observer.sql Script。
- f. 切換至配置 Script 所在的「商業程序編排器」子目錄。

在 Linux 及 UNIX 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\config
```

- g. 啟動此工具以設定事件收集程式，如第 230 頁的『setupEventCollector 工具』中所述。
- h. 選取選項 1，準備事件收集程式應用程式的資料庫。
- i. 輸入 7 或 8，選取您的 DB2 on z/OS 版本號碼。
- j. 此工具可讓您建立 SQL 檔案，交由資料庫管理者執行，而不是以現行使用者 ID 執行。當您看見：

```
Do you want to create an SQL file only (delay database preparation)?
```

```
y) yes  
n) no
```

- 如果您不想要延遲執行 SQL，請輸入 n。
- 如果您想要延遲執行 SQL，請輸入 y。您會看到：

```
Even if you want to delay the configuration,  
your entered values can be checked within the database.
```

```
Do you want to perform these checks?
```

```
y) yes  
n) no
```

- 如果您想要在資料庫內檢查所輸入的值，請輸入 y。
- 否則，請輸入 n。

視您輸入的指令而定，可能不會看到下列所有提示。跳過您沒看到的步驟。

k. 如果您看見：

Specify the JDBC driver type to be used:

- 2) Connect using type 2 (using a native database client)
- 4) Connect using type 4 (directly via JDBC)

指定 JDBC 驅動程式類型：

- 如果您使用原生資料庫用戶端，請輸入 2。
- 否則，請輸入 4，以選取第四類型 JDBC 驅動程式。

l. 如果您看見：

Specify the name of database in local catalog: [BPEDB]

輸入您的資料庫在本端 DB2 用戶端上編目的名稱，這是您在步驟 第 201 頁的 2b 中用於 *database\_alias* 的值。

m. 如果您看見：

Specify the location name/connection target: []

輸入要連接的子系統的位置名稱。

**註：**若要決定位置名稱，請使用 SQL 處理器登入，並執行下列 SQL 陳述式：

```
SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1
```

n. 如果您看見：

Specify the name of the database as known by the subsystem: [OBSVRDB]

輸入在 z/OS 主機上的子系統內用來辨識此資料庫的名稱。

o. 如果您看見：

Specify the hostname of the database server: [localhost]

Specify the port where the database server is listening: [446]

輸入 z/OS 資料庫伺服器使用的主機名稱及埠號。

p. 如果您看見：

Specify the directory of your JDBC driver: []

輸入 DB2 JDBC 驅動程式的 db2jcc.jar 及 db2jcc\_license\_cisuz.jar JAR 檔所在的目錄。

q. 如果您看見：

Specify userid to connect to the database 'database\_name' [db2admin] :

Specify the password for userid 'user\_ID' :

輸入使用者 ID 及密碼，以連接資料庫。這是步驟 第 201 頁的 1g 所述的使用者 ID *user\_ID*。

r. 如果您看見：

Specify the database schema to be used. [user\_ID] :

輸入用於資料庫物件的資料庫綱目名稱。

s. 如果您看見：

Note: The Java UDFs are more precise, but they require a jar file installed to the database.

Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果您想要使用更精確的 Java 型使用者定義函數 (UDF)，這需要在資料庫中安裝 JAR 檔，請輸入 1。
- 如果您要使用較不精確的 SQL 型 UDF，請輸入 2。

t. 如果您看見：

Specify the DB2 storage group name to be used. [OBSVRSG] :

輸入步驟 第 200 頁的 1d 中的儲存體群組名稱。

u. 如果您看見：

Specify the WLM environment name where the UDF should run. [] :

輸入在步驟 第 201 頁的 1j 中記下的 WLM 環境。在檢查必要的表格空間並載入 JAR 檔至資料庫之後，下列訊息指出順利完成：

The setup of the database completed successfully.

3. 使用管理主控台，建立指向資料庫的 XA 資料來源。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

#### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

## 準備商業程序編排器觀察程式的 Derby 資料庫

您可以使用 Script 或互動式工具來準備此資料庫。

### 使用 SQL Script 來準備商業程序編排器觀察程式的 Derby 資料庫:

本文說明如何使用 createSchema\_Observer.sql script 來準備 Derby 資料庫。

您必須建立「商業程序編排器觀察程式」資料庫的綱目。您可以在現有的資料庫中建立它，或由 Script 檔為您建立新的資料庫。

1. 切換至資料庫的配置 Script 所在的「商業程序編排器」子目錄。

- 在 Linux、UNIX 及 i5/OS 平台上，切換至目錄 `install_root/dbscripts/ProcessChoreographer/Derby`。
- 在 Windows 平台上，切換至目錄 `install_root\dbscripts\ProcessChoreographer\Derby`。

2. 執行下列動作：

- a. 在 Derby 網路伺服器環境中，將 \*Observer.sql Script 複製到網路伺服器。也將 JAR 檔 bpcodbutil.jar 從 *install\_root* 目錄的 lib 子目錄複製到資料庫伺服器上的相同目錄。
  - b. 在文字編輯器中，閱讀 Script 檔 createSchema\_Observer.sql 標頭中的指示。如果您要建立新資料庫，請在資料庫名稱後面附加 ;create=true。例如，若資料庫名稱是 OBSVRDB，請將 -Dij.database=OBSVRDB 參數取代為 -Dij.database=OBSVRDB;create=true
- 註:** 在 Windows 平台上，請避免使用「記事本」編輯器，因為此編輯器所顯示的檔案無法讓使用者閱讀。
- c. 若您使用內嵌式 Derby 驅動程式連接到現有的資料庫，請停止伺服器及使用該資料庫的任何其他應用程式。
  - d. 建立綱目。從您建立資料庫的目錄中，執行 Script 檔 createSchema\_Observer.sql，如 Script 的標頭所述。
  - e. 若發生錯誤，您可以執行 Script 檔 dropSchema\_Observer.sql 來捨棄綱目。
3. 使用管理主控台，建立指向資料庫的 XA 資料來源，並測試連線。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

#### 使用 **setupEventCollector** 工具來準備商業程序編排器觀察程式的 Derby 資料庫:

本文說明如何使用互動式功能表導向工具 **setupEventcollector**，在任何支援的平台上準備 Derby 資料庫。

1. 切換至配置 Script 所在的「商業程序編排器」子目錄。

在 Linux、UNIX 及 i5/OS 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\config
```

2. 若您使用內嵌式 Derby 驅動程式連接到現有的資料庫，請停止伺服器及使用該資料庫的任何其他應用程式。規劃在啟動此工具時使用 `-conntype none`。
3. 啟動此工具以設定事件收集程式，如第 230 頁的『**setupEventCollector** 工具』中所述。
4. 當您看見：

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

選取選項 1，準備事件收集程式應用程式的資料庫。畫面上會顯示下列功能表：

```
Prepare a database for the WebSphere Business Process Choreographer
Event Collector and Observer
```

```
Select the type of your database provider:
```

c) Derby

- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle

0) Exit Menu

5. 輸入 c 以選取 Derby。
6. 此工具可讓您建立 SQL 檔案，交由資料庫管理者執行，而不是以現行使用者 ID 執行。當您看見：

Do you want to create an SQL file only (delay database preparation)?

- y) yes
- n) no

- 如果您不想要延遲執行 SQL，請輸入 n。
- 如果您想要延遲執行 SQL，請輸入 y。您會看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?

- y) yes
- n) no

- 如果您想要在資料庫內檢查所輸入的值，請輸入 y。
- 否則，請輸入 n。

視您輸入的指令而定，可能不會看到下列所有提示。跳過您沒看到的步驟。

7. 如果您看見：

Specify the JDBC driver type to be used:

- 1) Connect using the embedded JDBC driver
- 2) Connect using the network JDBC driver

Your selection: [1]

- 若要使用內嵌式 JDBC 驅動程式連接，請輸入 1。

**重要：** 使用此驅動程式配置資料庫時，請確定沒有其他應用程式（包括 WebSphere Process Server）連接到該資料庫。

- 若要使用網路 JDBC 驅動程式，請輸入 2。

8. 當您看見： 指定資料庫的名稱 [*database\_name*]

輸入資料庫的完整路徑。

**註：** 預設值 ...\BPEDB 是「商業程序編排器」所使用的相同資料庫。爲了提高效率，請使用不同的資料庫。

9. 如果您看見：

Specify the database schema to be used. [APP] :

輸入用於資料庫物件的資料庫綱目名稱。如果您輸入空格字元或將欄位保留空白，則會使用預設綱目 APP。

10. 如果您看見：

Specify the hostname of the database server: [localhost]  
指定資料庫伺服器接聽的埠： [1527]

輸入 Derby 網路伺服器的主機名稱及埠號。



11. 如果您看見：

Specify the directory of your JDBC driver: [B:\w\p\derby\lib]

- 若為內嵌式 JDBC 驅動程式，請輸入 derby.jar 檔案所在的目錄。
- 若為網路 JDBC 驅動程式，請輸入 derbyclient.jar 所在的目錄。

12. 如果您看見：

Specify userid to connect to the database *database\_name*: []

- 若伺服器需要鑑別，請輸入已獲授權連接到 Derby 網路伺服器的使用者 ID。
- 否則，不輸入任何值會導致使用 dummy 這個使用者 ID。這是因為 Derby JDBC 驅動程式一律需要使用者 ID 才能連接到網路伺服器。

13. 如果您看見：

The application server must be stopped to update a Derby /  
Cloudscape database.  
This must be done outside wsadmin using 'stopServer *server\_name*'.  
After the server is stopped, come back to this prompt and enter  
'c' to continue.  
Please stop the server '*server\_name*' now.  
Press 'c' to continue, 'a' to abort:

a. 在 wsadmin 之外，使用此指令來停止伺服器：

```
stopServer server_name
```

b. 若您已停止伺服器，請按 c 繼續。否則，請按 a 回到第 205 頁的 4 步驟所顯示的主功能表。

14. 如果您看見：

Specify the database schema to be used. [APP] :

輸入要用於資料庫物件的綱目名稱，或按 Enter 鍵來使用預設值。

15. 請確定您看到下列訊息，它確認資料庫已準備完成：

The setup of the database completed successfully.

16. 如果您看見：

Restart the server now using 'startServer *server\_name*'.  
After the server is up again, come back to this prompt and enter  
'c' to continue.  
Press 'c' to continue, 'a' to abort:

a. 使用下列指令來啟動伺服器：

```
startServer server_name
```

b. 待伺服器啟動之後，回到這個提示，按 c 繼續。否則，請按 a 回到第 205 頁的 4 步驟所顯示的主功能表。

下列訊息指出順利完成：

WASX7074I: Reconnect of SOAP connector to host localhost completed.

17. 使用管理主控台，建立指向資料庫的 XA 資料來源，並測試連線。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

## 準備商業程序編排器觀察程式的 Oracle 資料庫

您可以使用 Script 或互動式工具來準備此資料庫。

使用 SQL Script 來準備商業程序編排器觀察程式的 Oracle 資料庫：

本文說明如何使用 `createTablespace_Observer.sql` 及 `createSchema_Observer.sql` Script 以準備 Oracle 資料庫。

您的資料庫必須已存在。您可以使用現有的資料庫，或根據資料庫文件來建立新的資料庫。

1. 切換至資料庫的配置 Script 所在的「商業程序編排器」子目錄。
  - 在 Linux、UNIX 及 i5/OS 平台上，切換至目錄 `install_root/dbscripts/ProcessChoreographer/Oracle`。
  - 在 Windows 平台上，切換至目錄 `install_root\dbscripts\ProcessChoreographer\Oracle`。
2. 將所有 \*Observer.sql Script 檔複製到資料庫伺服器。
3. 如果您要使用 Java 型「商業程序編排器」使用者定義函數：
  - a. 也將 Jar 檔 `bpcodbutil.jar` 從 `install_root` 目錄的 `lib` 子目錄複製到包含這個 SQL Script 檔的目錄。
  - b. 安裝包含「商業程序編排器觀察程式」的 UDF 的 JAR 檔。
    - 1) 以具有 Oracle 管理權限的使用者身分登入資料庫伺服器，並切換至 JAR 檔 `bpcodbutil.jar` 所在的目錄：
      - 如果資料庫與應用程式伺服器位於相同的伺服器上，請切換至 `install_root` 目錄的 `lib` 子目錄。
      - 如果資料庫與應用程式伺服器位於不同的電腦上，請切換至您已複製 JAR 檔 `bpcodbutil.jar` 的目錄。
    - 2) 輸入下列指令來執行 Oracle `loadjava` 公用程式，以安裝 JAR 檔 `bpcodbutil.jar`：

```
loadjava -user user/password@database
          -schema schema_name
          -resolve bpcodbutil.jar
```

其中：  
`user`、`password` 及 `database` 是使用者 ID、密碼及資料庫名稱的有效值。  
`schema_name` 是要用來儲存類別的綱目名稱。這必須與用於「觀察程式」表格的綱目相同。
    - 3) 萬一發生問題，您可以使用下列指令來捨棄 JAR 檔：

```
dropjava -user user/password@database
          -schema schema_name bpcodbutil.jar
```
4. 建立表格空間
  - a. 根據檔案開頭的指示，編輯 `createTablespace_Observer.sql` Script 檔的副本。
  - b. 根據檔案開頭的指示，執行 `createTablespace_Observer.sql` Script 檔的副本。
  - c. 請確定 Script 輸出沒有錯誤。如果發生錯誤，您可以使用 `dropTablespace_Observer.sql` Script 檔來捨棄表格空間。
5. 建立綱目 (表格、索引及視圖)。
  - a. 根據檔案開頭的指示，編輯並執行 `createSchema_Observer.sql` Script 檔的副本。
  - b. 請確定 Script 輸出沒有錯誤。如果您要捨棄綱目，請使用 `dropSchema_Observer.sql` Script 檔。

6. 使用管理主控台，建立指向資料庫的 XA 資料來源，並測試連線。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

#### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 `setupEventCollector` 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

### 使用 `setupEventCollector` 工具來準備商業程序編排器觀察程式的 Oracle 資料庫:

本文說明如何使用互動式功能表導向工具 及 `createTablespace_Observer.sql` Script 以準備 Oracle 資料庫。

您的資料庫必須已存在。您可以使用現有的資料庫，或根據資料庫文件來建立新的資料庫。

**註:** 若要從 i5/OS 平台建立遠端 Oracle 資料庫，請執行第 207 頁的『使用 SQL Script 來準備商業程序編排器觀察程式的 Oracle 資料庫』。

1. 在 Linux 及 UNIX 平台上：將 `$ORACLE_HOME/bin` 新增至 PATH 變數。
2. 建立表格空間：
  - a. 切換至資料庫的配置 Script 所在的「商業程序編排器」子目錄。
    - 在 Linux 和 UNIX 平台上，切換至目錄 `install_root/dbscripts/ProcessChoreographer/Oracle`。
    - 在 Windows 平台上，切換至目錄 `install_root\dbscripts(ProcessChoreographer)\Oracle`。
  - b. 編輯 `createTablespace_Observer.sql` Script 檔 (根據檔案頂端的指示)。
  - c. 根據檔案開頭的指示，執行表格空間建立 Script 檔。
  - d. 請確定 Script 輸出沒有錯誤。如果發生錯誤，您可以使用 `dropTablespace_Observer.sql` Script 檔來捨棄表格空間。
3. 切換至配置 Script 所在的「商業程序編排器」目錄。

在 Linux 及 UNIX 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\config
```

4. 啟動此工具以設定事件收集程式，如第 230 頁的『`setupEventCollector` 工具』中所述。
5. 準備資料庫：當您看見：
  - 1) Prepare a database for the Event Collector and Observer
  - 2) Install the Event Collector application
  - 3) Remove the Event Collector application and related objects
  - 4) Change configuration settings of an installed Event Collector

- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

執行下列動作：

- a. 選取選項 1，以準備事件收集程式和觀察程式應用程式的資料庫。畫面上會顯示下列功能表：

Prepare a database for the WebSphere Business Process Choreographer  
Event Collector and Observer

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle

0) Exit Menu

- b. 輸入 o 來選取 Oracle。
- c. 此工具可讓您建立 SQL 檔案，交由資料庫管理者執行，而不是以現行使用者 ID 執行。當您看見：

Do you want to create an SQL file only (delay database preparation)?

- y) yes
- n) no

- 如果您不想要延遲執行 SQL，請輸入 n。
- 如果您想要延遲執行 SQL，請輸入 y。您會看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?

- y) yes
- n) no

- 如果您想要在資料庫內檢查所輸入的值，請輸入 y。
- 否則，請輸入 n。

視您輸入的指令而定，可能不會看到下列所有提示。跳過您沒看到的步驟。

- d. 如果您看見：

Specify the database to be used.

Note: Database must already exist.

Specify the name of your database [BPEDB] :

輸入資料庫的 SID 名稱。

- e. 如果您看見：

Specify the hostname where the oracle database resides: [localhost]

輸入資料庫伺服器的主機名稱或 IP 位址。

- f. 如果您看見：

Specify the port where the oracle listener is listening: [1521]

輸入 Oracle 接聽器的埠號。

g. 如果您看見：

Specify userid to connect to the database '*database\_name*' [system] :

輸入使用者 ID 以連接資料庫。預設值是 system。

h. 如果您看見：

Specify the password for userid '*user\_ID*' :

輸入使用者 ID 的密碼。

i. 當您看見：

Choose the implementation of the Observer user-defined functions.

Note: The Java UDFs are more precise, but they require a jar file installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果您想要使用更精確的 Java 型使用者定義函數 (UDF)，這需要在資料庫中安裝 JAR 檔，請輸入 1。
- 如果您要使用較不精確的 SQL 型 UDF，請輸入 2。

您會看到類似下列內容的項目：

```
Trying to connect to database 'database_name', using user 'user_ID'
Connected to 'database_name'
Checking for required tablespace(s) ['OBSVRTS', 'OBSVRL0B', 'OBSVRIDX']
All required tablespaces were found.
Loading the jar file 'install_root\lib\bpcodbutil.jar' into the database.
The jar file 'install_root\lib\bpcodbutil.jar' was successfully installed.
```

The setup of the database completed successfully.

6. 使用管理主控台，建立指向資料庫的 XA 資料來源。

「商業程序編排器觀察程式」的資料庫綱目已備妥。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

#### 相關工作

『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

## 在 Java 及 SQL 使用者定義函數之間進行選取

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間

間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

## 使用 Script 在 Java 及 SQL 使用者定義函數之間進行選取

本文說明如何使用 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

您想要使用 Java 實作而非 SQL 實作的「商業程序編排器觀察程式」UDF。

1. 如果您要使用 Java 實作的「商業程序編排器觀察程式」使用者定義函數 (UDF)，請將 Jar 檔 `bpcodbutil.jar` 從 `install_root` 目錄的 `lib` 子目錄複製到資料庫伺服器上的相同目錄。
2. 如果您想要使用 Java 實作的「商業程序編排器觀察程式」UDF，請安裝 Jar 檔 `bpcodbutil.jar`：

- a. 如果您尚未連接資料庫，請在 DB2 指令行處理器中，輸入下列指令來連接它：

```
db2 connect to databaseName
```

- b. 輸入下列指令來安裝 Jar 檔：

```
db2 call sqlj.install_jar('file:pathURL','schema.BPCODBUTIL')
```

其中 `pathURL` 是 Jar 檔的完整 URL，`schema` 是「商業程序編排器」資料庫的綱目名稱。例如：

- 在 Linux 及 UNIX 平台上，如果 Jar 檔是位於 `/tmp` 目錄中，您必須輸入下列指令：

```
db2 call sqlj.install_jar('file:/tmp/bpcodbutil.jar',  
    'schema.BPCODBUTIL')
```

- 在 Windows 平台上，如果 Jar 檔位於 `c:\tmp` 目錄中，您必須輸入下列指令：

```
db2 call sqlj.install_jar('file:c:/tmp/bpcodbutil.jar',  
    'schema.BPCODBUTIL')
```

**註：** 如果您要捨棄 Jar 檔，請使用下列指令：

```
db2 call sqlj.remove_jar('schema.BPCODBUTIL')
```

3. 捨棄 UDF 的 SQL 實作。根據檔案開頭的指示，編輯 `dropFunctions_Observer.sql` Script 檔。例如，若為 DB2，請在 DB2 指令行處理器中，輸入此指令：

```
db2 -tf dropFunctions_Observer.sql
```

4. 建立 Java 實作的 UDF。根據檔案開頭的指示，編輯 `createFunctionsJava_Observer.sql` Script 檔。例如，若為 DB2，請在 DB2 指令行處理器中，輸入此指令：

```
db2 -tf createFunctionsJava_Observer.sql
```

已切換所使用的 UDF 實作。

## 使用 `setupEventCollector` 工具，在 Java 及 SQL 使用者定義函數之間進行選取

本文說明如何使用互動式功能表導向工具，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。



若為 Derby 資料庫，setupEventCollector 一律使用 Java 型 UDF。若為其他資料庫類型，setupEventCollector 預設會使用 Java 型 UDF，但您可以使用此工具來切換至 SQL 型 UDF。如果您又改變主意，可以使用此工具切換回使用 Java 型 UDF。

1. 啟動此工具以設定事件收集程式，如第 230 頁的『setupEventCollector 工具』中所述。您會看到下列功能表：
  - 1) Prepare a database for the Event Collector and Observer
  - 2) Install the Event Collector application
  - 3) Remove the Event Collector application and related objects
  - 4) Change configuration settings of an installed Event Collector
  - 5) Drop the database schema of the Event Collector and Observer
  - 6) Administer Observer related user-defined functions
- 0) Exit Menu
2. 選取選項 6，以管理觀察程式相關的使用者定義函數。您會看到下列功能表：
  - c) Derby
  - d) DB2 Universal
  - i) DB2 iSeries
  - 7) DB2 V7 on z/OS
  - 8) DB2 V8 on z/OS
  - o) Oracle
3. 如果您使用 DB2 for Linux, UNIX, or Windows 或 DB2 for z/OS，請選取適合您資料庫版本的選項：d、7 或 8
  - a. 當您看到下列功能表時：

Specify which type should be used to connect to the Database:

    - 2) Connect using type 2 (using a native DB2 client)
    - 4) Connect using type 4 (directly via JDBC)

請選取下列選項之一：

    - 2 若為第二類型 JDBC 連線，使用原生 DB2 用戶端。在此情況下，會提示您輸入下列資訊：

資料庫名稱

資料庫使用者 ID

密碼

JDBC 驅動程式的目錄
    - 4 若為第四類型 JDBC 驅動程式，則直接連接。在此情況下，會提示您輸入下列資訊：

資料庫名稱

資料庫伺服器主機名稱

資料庫伺服器埠號

JDBC 驅動程式的目錄

資料庫使用者 ID

密碼
4. 如果您使用 Oracle，請選取選項 o。
  - a. 輸入下列連線資訊：

資料庫伺服器主機名稱

資料庫伺服器埠號

資料庫名稱

資料庫使用者 ID

密碼

### JDBC 驅動程式的目錄

5. 若可以建立資料庫的連線，您會看到可管理觀察程式資料庫之 UDF 的功能表：

6) Administer Observer related user-defined functions

- 1) Activate Java based user-defined functions
- 2) Activate SQL based user-defined functions
- 3) Determine current state
- 4) List, install or remove the jar file containing the java based functions

註：『activate』 選項不適用於 Derby 資料庫。

a. 如果您想要啓動 Java 型 UDF，請選取選項 1。

1) 當您看見：

Specify the database schema to be used:

輸入資料庫綱目的名稱。

2) 當您看見：

WARNING: Switching the UDF implementation type may break any running Observer applications. Continue anyway?

y) yes

n) no

Your selection :

輸入 y 繼續，或輸入 n 不繼續。

3) 如果您繼續，會看到類似下面的文字：

Removing the user-defined functions ...

The jar file with jar\_id 'DB2INST1.BPCODBUTIL' is updated with the current version.

Loading the jar file 'B:\w\p\lib\bpcodbutil.jar' into the database. The jar file 'BPCODBUTIL' was successfully installed.

Creating the Java based user-defined functions ...

4) 下列訊息指出順利完成：

The setup of the database completed successfully.

b. 如果您想要啓動 SQL 型 UDF，請選取選項 2。

1) 當您看見：

Specify the database schema to be used:

輸入資料庫綱目的名稱。

2) 當您看見：

WARNING: Switching the UDF implementation type may break any running Observer applications. Continue anyway?

y) yes

n) no

Your selection :

輸入 y 繼續，或輸入 n 不繼續。

3) 如果您看見：

Removing the user-defined functions ...

Creating the SQL based user-defined functions ...

Do you also want to remove the jar file from the database?

y) yes

n) no

Your selection :

輸入 y，從資料庫中移除 JAR 檔，或輸入 n 不移除它。

4) 下列訊息指出順利完成：

The setup of the database completed successfully.

- c. 選擇性的: 若要決定所選取的 UDF 實作是 Java 或 SQL，而且當 Java 為作用中的情況下，也要驗證是否已安裝 JAR 檔，請選取選項 3。例如，如果 Java 實作為作用中，您應該會看到下列類似訊息：

The active UDF implementation is Java.

Tested functionality of the UDF, is working

- d. 選擇性的: 若要安裝或移除 Java 型 UDF 所需要的 JAR 檔，若要列出已安裝在資料庫中的所有 JAR 檔，請選取選項 4，然後當您看到下列功能表時：

List, install or remove jar files containing the java based functions

1) Install the jar file containing the Observer functions  
into the database

2) Remove the jar file containing the Observer functions  
from the database

3) List installed jar files

0) Exit Menu

- 選取選項 1，以安裝 JAR 檔。
- 選取選項 2，以移除 JAR 檔。
- 選取選項 3，以列出安裝在資料庫中的 JAR 檔。
- 選取選項 0，以結束功能表。

- e. 反覆地選取選項 0，回到第 213 頁的 1 步驟中顯示的功能表。

「商業程序編排器觀察程式」資料庫將使用您選取的 UDF。

## 商業程序編排器觀察程式的使用者定義函數

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

UDF 可以安裝在下列其中一項實作中：

### SQL 實作

使用資料庫系統提供的內建時間函數，將 SQL 實作用於以一般 SQL 實作的 UDF。

因為 SQL 實作只需要執行提供的 SQL Script，所以安裝 SQL 實作會比安裝 Java 實作簡單。若為這些 Script，則安裝時需要較少的管理權限。此外，SQL 實作的效能也會高於 Java 實作。然而，因為內建時間函數的限制，所以 SQL 實

作的 UDF 可能無法如您需要地一般精確。例如，在 DB2 上，內建時間函數假設每個月的長度是 30 天，而這可能會篡改您的結果。

SQL 實作不適用於 Derby 資料庫。

## Java 實作

Java 實作是用於使用 Java 語言實作的 UDF。

若要安裝 Java 實作，請使用資料庫系統提供的機制。Java 實作的 UDF 會授與精確的報告。然而，安裝 Java 實作需要的步驟會多於安裝 SQL 實作，而且需要較多的資料庫管理權限。例如，在 DB2 z/OS 資料庫上，必須設定工作管理程式 (WLM) 環境才能執行 UDF。

根據選擇用來設定資料庫的方式，預設實作會不同：

- 如果將資料庫設定成使用 SQL Script，或使用「第一次觸控時建立表格」功能，則預設會安裝 SQL 實作。
- 如果將資料庫設定成使用 setupEventCollector 工具，或使用設定檔建立精靈中的「商業程序編排器」範例配置 (只有在 Derby 資料庫上才提供)，則預設會安裝 Java 實作。

您可以在初始設定之後變更 UDF 的實作。這會在第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』中加以說明。

### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

第 191 頁的『使用 SQL Script 來準備商業程序編排器觀察程式的 DB2 for iSeries 資料庫』

i5/OS qshell 環境中的 本文說明如何使用 createSchema\_Observer.sql script，用來準備 DB2 for iSeries 資料庫。

第 192 頁的『使用 setupEventCollector 工具來準備商業程序編排器觀察程式的 DB2 for iSeries 資料庫』

本文說明如何使用互動式功能表導向工具，從 i5/OS qshell 環境內準備 DB2 for iSeries 資料庫。

第 194 頁的『使用 setupEventCollector 工具，從遠端系統準備 DB2 for iSeries 資料庫』

本文說明如何使用互動式功能表導向工具，從遠端 Linux、Windows 或 UNIX 系統，準備「商業程序編排器觀察程式」的 DB2 for iSeries 資料庫。

第 197 頁的『在 USS 中建立商業程序編排器觀察程式的 DB2 for z/OS 資料庫』  
本文說明如何使用互動式功能表導向工具及 createTablespace\_Observer.sql Script (位於 z/OS 機器上的「UNIX 系統服務 (USS)」中) 來建立 DB2 for z/OS 資料庫。

第 200 頁的『從遠端系統建立商業程序編排器觀察程式的 DB2 for z/OS 資料庫』  
本文說明如何使用互動式功能表導向工具及 createTablespace\_Observer.sql Script，在 Linux、UNIX 或 Windows 系統上建立 DB2 for z/OS 資料庫。

第 207 頁的『使用 SQL Script 來準備商業程序編排器觀察程式的 Oracle 資料庫』  
本文說明如何使用 createTablespace\_Observer.sql 及 createSchema\_Observer.sql Script 以準備 Oracle 資料庫。

第 209 頁的『使用 setupEventCollector 工具來準備商業程序編排器觀察程式的 Oracle 資料庫』

本文說明如何使用互動式功能表導向工具及 createTablespace\_Observer.sql Script 以準備 Oracle 資料庫。

第 187 頁的『使用 SQL Scripts 來準備商業程序編排器觀察程式的 DB2 Universal 資料庫』

本文說明如何使用 createTablespace\_Observer.sql 及 createSchema\_Observer.sql Script，在 Linux、UNIX 及 Windows 平台上準備 DB2 Universal 資料庫。

第 188 頁的『使用 setupEventCollector 工具來準備商業程序編排器觀察程式的 DB2 Universal 資料庫』

本文說明如何使用互動式功能表導向工具及 createTablespace\_Observer.sql script，在 Linux、UNIX 及 Windows 平台上準備 DB2 Universal 資料庫。

第 230 頁的『setupEventCollector 工具』

使用 setupEventCollector，以互動方式配置或移除「商業程序編排器」事件收集程式應用程式，來設定資料庫及管理資料庫的使用者定義函數。此工具使用 wsadmin Scripting。

## 配置商業程序編排器事件收集程式應用程式

使用互動式工具或管理主控台來安裝及配置事件收集程式應用程式。

「共用事件基礎架構 (CEI)」必須配置在您要安裝事件收集程式應用程式的部署目標上。

若要配置「商業程序編排器」事件收集程式，請執行下列其中一項：

### 使用 setupEventCollector 工具來配置商業程序編排器事件收集程式

本文說明如何使用互動式功能表導向工具 在伺服器或叢集上安裝及配置事件收集程式應用程式。

1. 切換至配置 Script 所在的「商業程序編排器」子目錄。

在 Linux、UNIX 及 i5/OS 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\config
```

2. 啟動此工具以設定事件收集程式，如第 230 頁的『setupEventCollector 工具』中所述。例如：

在 Linux 和 UNIX 平台上：

```
setupEventCollector.sh -server server1
```

在 i5/OS 平台上：

```
setupEventCollector -server server1
```

在 Windows 平台上：

```
setupEventCollector.bat -server server1
```

您會看到 Commands Menu：

Commands Menu

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application

- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

3. 若要安裝「商業程序編排器」事件收集程式應用程式：

- a. 選取選項 2。畫面上會顯示下列內容：

```
Create required objects and install the WebSphere Business Process  
Choreographer Event Collector application ...
```

- b. 若安裝在獨立式伺服器上，您會看到：

```
Working on node 'your_node_name', server 'your_server_name'.
```

- c. 若您在部署管理程式上安裝應用程式，則必須從所有可用目標清單中選取部署目標。例如：

```
Select the deployment target to install to:
```

- 1) Cluster '*cluster1*'
- 2) Node '*Node04*', Server '*managed1*'
- 3) Node '*Node04*', Server '*managed2*'

0) Exit Menu

- d. 當此工具在部署目標上搜尋現有的事件收集程式安裝時，您會看到如下的訊息：

```
Searching for an already installed Event Collector on 'deployment_target'
```

- e. 若已經安裝事件收集程式應用程式的實例，您會看到：

```
Do you want to overwrite the existing application?
```

- o) Overwrite
- a) Abort

- 輸入 o，以改寫現有的事件收集程式應用程式。所有安裝值都可以重新輸入，且事件收集程式應用程式已更新。
- 輸入 a 結束，而不安裝事件收集程式。

4. 當您看見：

```
Specify the JNDI name of the database where the WebSphere Business Process  
Choreographer Event Collector should store the collected events.  
Enter '?' to get a list.  
Your selection : [jdbc/BPEDB]
```

輸入用來連接資料庫的 JNDI 名稱。您也可以輸入 ?，取得所有已登錄資料來源的清單。例如：

```
jdbc/BPEDB  
jdbc/DefaultEJBTimerDataSource  
jdbc/mediation/messageLog
```

5. 當您看見：

```
Specify the database schema to be used.  
Enter a space character or leave empty to use the default schema of the  
datasource. [] :
```

輸入事件收集程式儲存事件的資料庫表格的綱目名稱。若要使用在資料來源定義的鑑別別名中指定的使用者 ID 作為綱目，請輸入空格字元或將欄位保留空白。

已建立所有必要的物件，也已安裝企業應用程式。下列訊息指出順利完成：



WebSphere Business Process Choreographer Event Collector  
installed successfully!

6. If CEI logging is not enabled on the server, you see the following:

Checking if CEI event logging is enabled ...

Warning: The Business process container of *server\_name* has CEI event logging disabled.  
To allow the Event Collector to work correctly, CEI event logging is required.  
Do you want to enable the CEI event logging on *server\_name*? (y/n)

- If you want the script to enable CEI logging on the named server, enter *y*.
- If you do not want the script to enable CEI logging on the named server, enter *n*.

**註:** It is important that CEI logging is enabled when you start working with the Business Process Choreographer Observer.

7. 當出現提示時：

Do you want to save the changes? (y/n)

如果沒有錯誤訊息，請輸入 *y* 以儲存配置。如果有錯誤，請輸入 *n* 來捨棄變更並保留原始配置。請檢查日誌檔 `setupEventCollector.log`，它位於設定檔的 `logs` 目錄中。

例如，在 Windows 上，如果設定檔是 `myServer`，且您的設定檔都儲存在 `install_root\profiles` 中，則日誌檔位於 `install_root\profiles\myServer\logs` 中。

8. 輸入 `0`，以結束功能表。

9. 讓變更生效：

- 如果您在啟動此工具時指定 `-conntype NONE` 選項，則在重新啟動伺服器之後，您的變更就會生效。
- 如果您在啟動此工具時未指定 `-conntype NONE` 選項，且您在安裝「商業程序編排器」事件收集程式期間於伺服器上啟用 CEI 記載功能，請使用管理主控台來停止並重新啟動 `BPEContainer` 應用程式。

已安裝及配置「商業程序編排器」事件收集程式應用程式。

## 使用管理主控台來配置商業程序編排器事件收集程式

本文說明如何使用管理主控台，在給定的伺服器或叢集上安裝「商業程序編排器」事件收集程式的實例。

您已準備好「商業程序編排器觀察程式」資料庫。

1. 在管理主控台中，導覽至「商業程序編排器」事件收集程式配置頁面：按一下**伺服器** → **叢集** → **cluster\_name** 或**伺服器** → **應用程式伺服器** → **server\_name**，然後在**商業整合**之下，展開**商業程序編排器**，再按一下**商業程序編排器事件收集程式**。
2. 若要建立新配置：
  - a. 輸入或選取下列欄位的值：
    - 資料庫實例名稱。
    - 綱目名稱。
    - 啟用或清除第一次使用該資料庫時建立資料庫表格的選項。

- 連接至資料庫的使用者名稱及密碼。
  - 資料庫伺服器的主機名稱或 IP 位址。
  - 資料庫伺服器的埠號。
  - JDBC 提供者。
  - 觀察目標：
    - 受管理商業程序編排器儲存器
    - 現有事件群組名稱
    - 事件群組名稱
- b. 按一下**套用**，以部署應用程式。
- c. 萬一發生問題，請檢查 SystemOut.log 檔案。否則，會儲存主要配置的變更。
- d. 按一下**應用程式** → **企業應用程式**來啟動應用程式，選取應用程式 BPCECollector\_scope，其中 scope 識別部署目標，然後按一下**啟動**。

已配置「商業程序編排器」事件收集程式。

#### 商業程序編排器事件收集程式：

您必須先配置事件收集程式，才能使用「商業程序編排器觀察程式」。

若要檢視此管理主控台頁面，請按一下**伺服器** → **叢集** → **cluster\_name** 或**伺服器** → **應用程式伺服器** → **server\_name**，然後在**商業整合**之下，展開**商業程序編排器**，然後按一下**事件收集程式配置**。

#### 資料來源：

在此章節中，您可以配置「商業程序編排器」事件收集程式的資料來源。

#### 編輯：

按一下此按鈕來編輯資料來源。

#### 測試連線：

測試到已選取資料來源的連線。

#### 資料庫實例：

資料庫的名稱。

內容	值
資料類型	字串
預設值	\${USER_INSTALL_ROOT}\databases\BPOBEC00

#### 綱目名稱：

資料庫綱目的名稱。

內容	值
資料類型	字串
預設值	無

### 建立表格:

如果已選取，則在第一次存取資料庫時就會自動建立表格。若要此選擇生效，資料庫必須已經存在，且所提供的使用者名稱必須具有在資料庫中建立表格及索引的權限。

在正式作業系統中，不建議使用此選項。如果未選取此選項，則不會自動建立表格，您必須執行 Script 來手動建立表格。

內容	值
資料類型	勾選框
預設值	已選取

### 使用者名稱:

具有權限連接到資料庫及修改資料的使用者 ID。

如果使用者 ID 有權限在資料庫中建立表格及索引，則在套用服務套件或修正套件之後，將會視情況來自動更新資料庫綱目。

內容	值
資料類型	字串
預設值	目前登入的使用者。

### 密碼:

資料來源使用者 ID 的密碼。

內容	值
資料類型	字串
預設值	無

### 伺服器:

資料庫伺服器的位址。

指定主機名稱或 IP 位址。

內容	值
資料類型	字串
預設值	無

### 埠:

資料庫伺服器使用的埠號。

內容	值
資料類型	字串
預設值	視所選取的 JDBC 提供者而定。

### 提供者:

JDBC 提供者。

內容	值
資料類型	下拉清單
預設值	DERBY_EMBEDDED

**JMS 使用者名稱:**

內容	值
資料類型	字串
預設值	針對「商業程序編排器」JMS 使用者指定的相同使用者 ID。

**JMS 密碼:**

內容	值
資料類型	字串
預設值	針對「商業程序編排器」JMS 使用者指定的相同密碼。

**觀察目標:**

在此章節中，您可以指定事件收集程式的目標。

內容	值
資料類型	圓鈕
選項	<ul style="list-style-type: none"><li>• 受管理商業程序編排器儲存器</li><li>• 現有事件群組名稱</li><li>• 事件群組名稱</li></ul>

**受管理的「商業程序編排器」儲存器:**

選取已配置的「商業程序編排器」儲存器。

內容	值
資料類型	下拉清單
目錄	所有「商業程序編排器」配置

**事件群組設定檔清單:**

選取群組設定檔。

內容	值
資料類型	下拉清單

**輸入事件群組設定檔清單:**

輸入事件群組設定檔清單。

內容	值
資料類型	字串

內容  
預設值

值  
無

## 配置商業程序編排器觀察程式應用程式

您可以使用工具或管理主控台來配置「商業程序編排器觀察程式」應用程式。

您已配置「商業程序編排器」事件收集程式。

若要配置「商業程序編排器觀察程式」，請執行下列其中一項：

### 使用 **setupObserver** 工具來配置商業程序編排器觀察程式

本文說明如何使用互動式功能表導向工具，安裝「商業程序編排器觀察程式」應用程式的實例，並配置它來連接到特定事件收集程式的資料來源。

您已準備「商業程序編排器觀察程式」資料庫，並使用管理主控台來建立它的資料來源。

1. 切換至配置 `Script` 所在的「商業程序編排器」子目錄。

在 Linux、UNIX 及 i5/OS 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\config
```

2. 啟動此工具以設定觀察程式，如第 232 頁的『`setupObserver` 工具』中所述。您會看到下列功能表：

- 1) Install the Observer application
- 2) Remove the Observer application and related objects
- 3) Change configuration settings of an installed Observer

0) Exit Menu

3. 選取選項 1，安裝「商業程序編排器觀察程式」。畫面上會顯示下列內容：

```
Create required objects and install the WebSphere Business Process  
Choreographer Observer application ...
```

4. 若安裝在獨立式伺服器上，您會看到：

```
Working on node 'your_node_name', server 'your_server_name'.
```

5. 若您在部署管理程式上安裝應用程式，則必須從所有可用目標清單中選取部署目標。例如：

```
Select the deployment target to install to:
```

- 1) Cluster '*cluster1*'
- 2) Node '*Node04*', Server '*managed1*'
- 3) Node '*Node04*', Server '*managed2*'

0) Exit Menu

6. 當您看見：

```
Specify the JNDI name of the database containing the event tables.  
Enter '?' to get a list.  
Your selection : [jdbc/BPEDB]
```

輸入用來連接資料庫的 JNDI 名稱。您也可以輸入 `?`，取得所有已登錄資料來源的清單。例如：

```
jdbc/BPEDB
jdbc/DefaultEJBTimerDataSource
jdbc/mediation/messageLog
```

7. 當您看見：

Specify the database schema to be used.  
Enter a space character or leave empty to use the default schema of the datasource. [] :

輸入事件收集程式儲存事件的資料庫表格的綱目名稱。若要使用在資料來源定義的鑑別別名中指定的使用者 ID 作為綱目，請輸入空格字元或將欄位保留空白。

已建立所有必要的物件，也已安裝企業應用程式。下列訊息指出順利完成：

```
WebSphere BPC Observer installed successfully!
```

8. 出現下列提示時：Do you want to save the changes? (y/n)，如果沒有錯誤訊息，請輸入 `y`，以儲存配置。如果有錯誤，請輸入 `n` 來捨棄變更並保留原始配置。檢查日誌檔 `setupObserver.log`，它位於設定檔的 `logs` 目錄中。例如，在 Windows 上，如果設定檔是 `myServer`，且您的設定檔是儲存在 `install_root\profiles` 中，則該日誌檔位於 `install_root\profiles\myServer\logs` 中。
9. 輸入 `0`，以結束功能表。

已安裝和配置「商業程序編排器觀察程式」。

## 使用管理主控台來配置商業程序編排器觀察程式

本文說明如何使用管理主控台來安裝「商業程序編排器觀察程式」應用程式的實例，及配置它來連接到特定事件收集程式的資料來源。

您已配置「商業程序編排器」事件收集程式。

1. 在管理主控台中，導覽至「商業程序編排器觀察程式」配置頁面：按一下**伺服器** → **叢集** → **cluster\_name** 或**伺服器** → **應用程式伺服器** → **server\_name**，然後在**商業整合**之下，展開**商業程序編排器**，再按一下**商業程序編排器觀察程式**。
2. 若要建立新配置：
  - a. 按一下**新增**。
  - b. 輸入或選取下列欄位的值：
    - 這個實例的環境定義根目錄
    - 將來自這個「商業程序編排器」事件收集程式的監視資料視覺化如需這些配置參數的相關資訊，請參閱第 122 頁的『「商業程序編排器觀察程式」設定』。
  - c. 按一下**套用**，以部署應用程式。
  - d. 萬一發生問題，請檢查 `SystemOut.log` 檔案。否則，會儲存主要配置的變更。
  - e. 按一下**應用程式** → **企業應用程式**來啟動應用程式，選取應用程式 `BPCobserver_scope`，其中 `scope` 識別部署目標，然後按一下**啟動**。

「商業程序編排器觀察程式」已完成配置，隨時可以使用。

您可以重複這項作業，在相同或不同的部署目標上配置「商業程序編排器觀察程式」的多個實例。不過，每一個實例必須連接到不同的事件收集程式資料來源。



## 啓用商業程序編排器的記載功能

本文說明如何啓用「商業程序編排器」的「共用事件基礎架構 (CEI)」事件。

若要使用「商業程序編排器觀察程式」來監視商業程序事件，必須啓用商業程序以產生「共用事件基礎架構 (CEI)」事件。您可以在建立商業程序的模型時指定此項。爲了適當地監視商業程序，至少必須產生『Process Started』事件。如需可使用「商業程序編排器觀察程式」來監視的 CEI 事件清單，請參閱商業程序事件。如需如何啓用商業程序以產生 CEI 事件的相關資訊，請參閱「WebSphere Integration Developer 資訊中心」。

如果您安裝「商業程序編排器」事件收集程式的部署目標與配置「商業程序編排器」的部署目標相同，當您安裝應用程式時，可以使用 `setupEventCollector` 工具來啓用 CEI 記載功能。如果您使用管理主控台安裝「商業程序編排器」事件收集程式，則必須使用 `Script` 或使用管理主控台來啓用 CEI 記載功能。

若要使用 `Jython Script` 來啓用「商業程序編排器」的 CEI 記載功能，請執行『使用 `Script` 啓用商業程序編排器的記載功能』。

若要使用管理主控台來啓用「商業程序編排器」的 CEI 記載功能，請執行 第 251 頁的『使用管理主控台來啓用共用基礎事件及審核追蹤』。

將產生商業程序及活動的「共用事件基礎架構」事件，並可由「商業程序編排器」事件收集程式接收事件。

### 使用 `Script` 啓用商業程序編排器的記載功能

本文說明如何使用 `setStateObserver.py` `Script`，來啓用或停用「商業程序編排器」的「共用事件基礎架構 (CEI)」或審核事件。

#### 位置

`setStateObserver.py` `Script` 位於「商業程序編排器」`config` 目錄中。

#### 執行 `Script`

若要執行 `setStateObserver` `Script`：

在 Linux 及 UNIX 平台上，輸入：

```
install_root/bin/wsadmin.sh -lang jython  
-f install_root/ProcessChoreographer/config/setStateObserver.py
```

在 i5/OS 平台上，輸入：

```
install_root/bin/wsadmin -lang jython  
-f install_root/ProcessChoreographer/config/setStateObserver.py
```

在 Windows 平台上，輸入：

```
install_root\bin\wsadmin.bat -lang jython  
-f install_root\ProcessChoreographer\config\setStateObserver.py
```

#### 參數

`Script` 檔可以採用下列參數：

**-conntype** *NONE*

只有在應用程式伺服器 (限獨立式) 或部署管理程式未執行時，才會併入此選項。

**-node** *nodeName*

其中，*nodeName* 是節點的名稱。如果指定叢集，請不要指定此選項。

**-server** *serverName*

其中，*serverName* 是伺服器名稱。如果指定叢集，請不要指定此選項。

**-cluster** *clusterName*

其中，*clusterName* 是叢集名稱。請不要在獨立式伺服器環境中指定這個選項，也不要指定節點及伺服器時指定這個選項。

**-profileName** *profileName*

其中，*profileName* 是要使用的設定檔名稱。

**-enable ( CEI | AuditLog | CEI;AuditLog)**

選擇性地指定是否啟用 CEI 記載功能及 (或) 審核日誌。

**-disable ( CEI | AuditLog | CEI;AuditLog)**

選擇性地指定是否停用 CEI 記載功能及 (或) 審核日誌。

**-bfm**

選擇性地指定將啟用或停用套用至「商業程序編排器」的「商業流程管理程式」(其執行商業程序)。

**-htm**

選擇性地指定將啟用或停用套用至「商業程序編排器」的「人力作業管理程式」(其執行人力作業)。

## 範例

若要在位於 Linux 或 UNIX 平台的 `server1` 上啟用商業程序事件的 CEI 記載功能：  
`wsadmin.sh -lang jython -f setStateObserver.py -server server1 -enable CEI -bfm`

註：在 Windows 上，請使用 `wsadmin.bat`，在 iOS 上，則使用 `wsadmin`。

## 變更商業程序編排器觀察程式的配置參數

調整「商業程序編排器觀察程式」及事件收集程式應用程式的配置參數對於啟用驗證及改進效能很重要。

### 變更預設值

比起測試系統，預設值更適合於正式作業系統。如果您在設定用於開發或測試的「商業程序編排器」，則應該在驗證配置是否在運行中之前變更以下配置參數：

- 將 `BPCEventTransformerEventCount` 值變更為零。
- 將 `BPCEventTransformerToleranceTime` 值變更為一。

進行這些變更，可確保即使事件產生的速度低於正式作業系統，事件也會在一分鐘內變成可用。

### 配置事件收集程式的參數

調整數值參數會影響事件轉換程式觸發的頻率，以及讓事件可用於「商業程序編排器觀察程式」的經歷時間。

配置參數	資料類型 / 單位	預設值	說明
ObserverSchemaName	字串	未設定	這會識別作為所有資料庫物件字首使用的資料庫綱目。若其留為空白，則預設值是作為字首使用，使用者 ID 用來連接至資料庫。此使用者 ID 設為管理主控台中資料來源定義的一部分。如果您指定此參數的值，則在資料來源指定的使用者 ID 必須具有足夠的權限來存取此綱目的資料庫物件。
BPCEventTransformer EventCount	整數 / 事件	500	事件數目，事件收集程式在該數目之後觸發轉換程式，以將收集的事件轉換為適用於觀察程式應用程式的格式。  當您在部署、測試及實驗時，預設值可能過高，並會導致事件在一段較長時間內無法觀察。若要讓事件更快地變成可用，則可以將此值設為零。接著，之後每一個事件都會觸發轉換程式，並會在觀察程式中變成可見。如果您將值變為零，則任何尚未轉換的已傳遞事件都會在產生新事件時立刻轉換。正式作業系統不建議使用零這個值。
BPCEventTransformer MaxWaitTime	整數 / 分鐘	10	觸發轉換程式之前可以經過的最長時間－即使是未達到 BPCEventTransformer EventCount 所指定的事件數目亦然。
BPCEventTransformer ToleranceTime	整數 / 分鐘	10	觀察程式中事件變成可見的最短經歷時間 (以分鐘為單位)。這可讓相關事件產生可靠性關聯。應該避免使用零這個值，否則可能在達到上一個事件之前處理事件。  當您在部署、測試及實驗時，預設值可能過高，並會導致新事件在 10 分鐘內無法觀察。如果您將此參數設為值 1，則所有長於一分鐘的已轉換事件都會顯示在觀察程式中。
ObserverCreateTables	布林		這個參數指出，當 EJB 第一次連接至資料庫時，是否應建立「觀察程式」綱目。有效值為 'true' 和 'false'。

當事件收集程式從「共用事件基礎架構 (CEI)」接收商業相關事件時，事件會儲存在資料庫中。當一段時間過去，且接收到更多事件時，就會啟動轉換程式。轉換程式會執行已儲存事件的批次轉換，並會將它們以用來產生報告的格式寫回資料庫。只有已由轉換程式處理的事件可用於觀察程式報告。

事件收集程式每次接收新事件時，如果符合以下其中一個或兩個條件，則會啟動轉換程式程序：

- 自從上次啟動轉換程式之後接收的事件數目大於 `BPCEventTransformerEventCount` 的值。
- 自從上次啟動轉換程式之後的時間大於 `BPCEventTransformerMaxWaitTime` 的值 (以分鐘為單位)。

如果讓這些值變得更小一些，則可以更快地使用這些事件來產生報告，但是在轉換少量事件時有一些額外成本。這需要在透過處理較大量事件以取得較佳轉換產量，與讓事件在觀察程式資料庫中儘快變成可用之間取得平衡。

每次啟動轉換程式時，它會處理所有早於 `BPCEventTransformerToleranceTime` 的事件 (以分鐘為單位)。它不會處理最近的事件，因為事件不需要以發生的順序進行發佈。`BPCEventTransformerToleranceTime` 的預設值，會假設接收到並可寫入事件收集程式表格的事件，都不會花費超過 10 分鐘的時間。

## 變更事件收集程式的配置參數

若要變更事件收集程式參數，請執行下列動作：

1. 啟動此工具以設定事件收集程式，如第 230 頁的『`setupEventCollector` 工具』中所述。您會看到下列功能表：
  - 1) Prepare a database for the Event Collector and Observer
  - 2) Install the Event Collector application
  - 3) Remove the Event Collector application and related objects
  - 4) Change configuration settings of an installed Event Collector
  - 5) Drop the database schema of the Event Collector and Observer
  - 6) Administer Observer related user-defined functions
- 0) Exit Menu
2. 選取選項 4，以顯示您可以變更的參數清單：
  - 1) `BPCEventTransformerEventCount`
  - 2) `BPCEventTransformerMaxWaitTime`
  - 3) `BPCEventTransformerToleranceTime`
  - 4) `ObserverCreateTables`
  - 5) `ObserverSchemaName`
- 0) Exit Menu
3. 選取您要變更的參數號碼。此時會顯示參數的名稱、說明、類型、單位及現行值。
4. 若要變更指定的值，請輸入新值，然後按 Enter 鍵。按 Enter 鍵而不輸入新值，會回到參數清單。
5. 如果您要變更另一個參數的值，請從步驟 3 開始重複執行。
6. 輸入 0，以結束清單。系統會詢問您是否要儲存變更。
7. 若要儲存所有變更，請輸入 y，否則輸入 n，以捨棄所有變更。
8. 若要使變更生效，請重新啟動 `BPCECollector` 應用程式。

## 配置觀察程式的參數

ReportAtSnapshotRange 參數的值可能對 Snapshot 報告的效能具有較大影響。

配置參數	資料類型 / 單位	預設值	說明
ObserverSchemaName	字串	未設定	這會識別作為所有資料庫物件字首使用的資料庫綱目。若其留為空白，則預設值是作為字首使用，使用者 ID 用來連接至資料庫。此使用者 ID 設為管理主控台中資料來源定義的一部分。如果您指定此參數的值，則在資料來源指定的使用者 ID 必須具有足夠的權限來存取此綱目的資料庫物件。這必須符合事件收集程式的值。
ReportAtSnapshotRange	整數 / 天數	60	Snapshot 報告透過評估所有早於限定 Snapshot 日期及時間的事件來建置。 ReportAtSnapshotRange 會定義事件可以併入 Snapshot 報告的時段。只有在此期間內產生的事件會由 Snapshot 報告進行評估。  如果此值過高，則可能必須處理大量的事件，而產生報告可能會花費較長時間。嘗試將此值設為商業環境中程序實例的持續時間上限。
ObserverCreateTables	布林		這個參數指出，當 EJB 第一次連接至資料庫時，是否應建立「觀察程式」綱目。有效值為 'true' 和 'false'。

## 變更觀察程式的配置參數

若要變更觀察程式參數，請執行下列動作：

1. 啟動此工具以設定觀察程式，如第 232 頁的『setupObserver 工具』中所述。您會看到下列功能表：
  - 1) 安裝「觀察程式」應用程式
  - 2) 移除「觀察程式」應用程式及相關物件
  - 3) 變更已安裝「觀察程式」的配置設定
- 0) 結束功能表
2. 選取選項 3 以顯示您可以變更的參數清單：
  - 1) ReportAtSnapshotRange
  - 2) ObserverCreateTables
  - 3) ObserverSchemaName
- 0) Exit Menu

3. 選取您要變更的參數號碼。此時會顯示參數的名稱、說明、類型、單位及現行值。
4. 若要變更指定的值，請輸入新值，然後按 **Enter** 鍵。按 **Enter** 鍵而不輸入新值，會回到參數清單。
5. 如果您要變更另一個參數的值，請從步驟 3 開始重複執行。
6. 輸入 0，以結束清單。系統會詢問您是否要儲存變更。
7. 若要儲存所有變更，請輸入 *y*，否則輸入 *n*，以捨棄所有變更。
8. 若要使變更生效，請重新啟動 **BPCObserver** 應用程式。

## setupEventCollector 工具

使用 **setupEventCollector**，以互動方式配置或移除「商業程序編排器」事件收集程式應用程式，來設定資料庫及管理資料庫的使用者定義函數。此工具使用 **wsadmin Scripting**。

### 位置

此工具位於配置 **Script** 的「商業程序編排器」子目錄中：

在 **Linux**、**UNIX** 及 **i5/OS** 平台上：*install\_root/ProcessChoreographer/config*。

在 **Windows** 平台上：*install\_root\ProcessChoreographer\config*。

### 限制

- 在網路部署環境中，您必須在部署管理程式節點上啟動此工具，並使用 **-profileName** 選項指定部署管理程式設定檔。
- 此工具只有英文版。
- 在 **i5/OS** 上，您必須使用 **qshell** 來執行此工具。

### 參數

```
[-conntype SOAP | RMI | JMS | NONE]
[-user userID -password password]
[-profileName profileName]
( [-node nodeName] [-server serverName] ) | ( -cluster clusterName )
[-remove [-silent]]
```

其中：

#### **-conntype** SOAP | RMI | JMS | NONE

**wsadmin** 工具使用的連線模式。在獨立式伺服器環境中僅包括 **-conntype** NONE 選項 (如果應用程式伺服器不在執行中)。在網路部署環境中，如果部署管理程式未執行，則僅包括 **-conntype** NONE 選項。

#### **-user** *userID* **-password** *password*

如果已啟用廣域安全，則還需提供讓此工具使用的有效使用者 ID 及密碼。

#### **-profileName** *profileName*

如果您未配置預設設定檔，則提供您要配置的設定檔名稱。

#### **-node** *nodeName*

節點的名稱。這是選用參數。預設值是本端節點。

#### **-server** *serverName*

伺服器的名稱。這是選用參數。



**-cluster** *clusterName*

叢集名稱 *clusterName* 。這是選用參數。

**-remove**

指定此選項來移除事件收集程式應用程式。如果您未指定此選項，則預設會配置應用程式。

**-silent**

此選項只能搭配 `remove` 選項一起使用。它會讓此工具不輸出任何提示。這是選用參數。

**註:** 如果您未指定 `-node`、`-server` 或 `-cluster` 參數，則在配置期間系統會提示您輸入部署目標。

### 範例：啟動此工具

若要啟動此工具來使用名為 `server1` 的伺服器，請輸入下列其中一項指令。

在 **Linux** 和 **UNIX** 平台上：

```
setupEventCollector.sh -server server1
```

在 **i5/OS** 平台上：

```
setupEventCollector -server server1
```

在 **Windows** 平台上：

```
setupEventCollector.bat -server server1
```

您會看到下列「指令」功能表：

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

### 使用此工具

下列主題說明如何使用此工具來進行特定作業。

#### 相關概念

第 215 頁的『商業程序編排器觀察程式的使用者定義函數』

您可以使用「商業程序編排器觀察程式」，根據會導致 SQL 查詢的時間片段或時間間隔來執行報告。若要執行這些報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定的使用者定義函數 (UDF)。

#### 相關工作

第 217 頁的『配置商業程序編排器事件收集程式應用程式』

使用互動式工具或管理主控台來安裝及配置事件收集程式應用程式。

第 188 頁的『使用 `setupEventCollector` 工具來準備商業程序編排器觀察程式的 DB2 Universal 資料庫』

本文說明如何使用互動式功能表導向工具及 `createTablespace_Observer.sql` script，在 Linux、UNIX 及 Windows 平台上準備 DB2 Universal 資料庫。

第 192 頁的『使用 `setupEventCollector` 工具來準備商業程序編排器觀察程式的 DB2 for iSeries 資料庫』

本文說明如何使用互動式功能表導向工具，從 i5/OS qshell 環境內準備 DB2 for iSeries 資料庫。

第 197 頁的『在 USS 中建立商業程序編排器觀察程式的 DB2 for z/OS 資料庫』

本文說明如何使用互動式功能表導向工具 及 `createTablespace_Observer.sql` Script (位於 z/OS 機器上的「UNIX 系統服務 (USS)」中) 來建立 DB2 for z/OS 資料庫。

第 200 頁的『從遠端系統建立商業程序編排器觀察程式的 DB2 for z/OS 資料庫』

本文說明如何使用互動式功能表導向工具 及 `createTablespace_Observer.sql` Script，在 Linux、UNIX 或 Windows 系統上建立 DB2 for z/OS 資料庫。

第 205 頁的『使用 `setupEventCollector` 工具來準備商業程序編排器觀察程式的 Derby 資料庫』

本文說明如何使用互動式功能表導向工具 `setupEventcollector`，在任何支援的平台上準備 Derby 資料庫。

第 209 頁的『使用 `setupEventCollector` 工具來準備商業程序編排器觀察程式的 Oracle 資料庫』

本文說明如何使用互動式功能表導向工具 及 `createTablespace_Observer.sql` Script 以準備 Oracle 資料庫。

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 `setupEventCollector` 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

第 237 頁的『使用 Script 來移除商業程序編排器配置』

您可以使用這項作業，移除伺服器或叢集中的「商業流程管理程式」、「人力作業管理程式」、「商業程序編排器瀏覽器」及「商業程序編排器觀察程式」配置及相關聯的資源。

### 相關參考

第 226 頁的『變更商業程序編排器觀察程式的配置參數』

調整「商業程序編排器觀察程式」及事件收集程式應用程式的配置參數對於啓用驗證及改進效能很重要。

## setupObserver 工具

使用 `setupObserver` 工具，以互動方式配置或移除「商業程序編排器觀察程式」應用程式，或變更配置參數。此工具使用 `wsadmin` Scripting。

### 位置

此工具位於配置 Script 的「商業程序編排器」子目錄中：

在 Linux、UNIX 及 i5/OS 平台上：`install_root/ProcessChoreographer/config`。

在 Windows 平台上：`install_root\ProcessChoreographer\config`。

### 限制

- 在網路部署環境中，您必須在部署管理程式節點上啓動此工具，並使用 `-profileName` 選項指定部署管理程式設定檔。
- 此工具只有英文版。
- 在 i5/OS 上，您必須使用 `qshell` 來執行此工具。

## 參數

```
[-conntype SOAP | RMI | JMS | NONE]
[-user userID -password password]
[-profileName profileName]
( [-node nodeName] [-server serverName] ) | ( -cluster clusterName )
[-remove [-silent]]
```

其中：

### **-conntype SOAP | RMI | JMS | NONE**

`wsadmin` 工具使用的連線模式。在獨立式伺服器環境中僅包括 `-conntype NONE` 選項 (如果應用程式伺服器不在執行中)。在網路部署環境中，如果部署管理程式未執行，則僅包括 `-conntype NONE` 選項。

### **-user *userID* -password *password***

如果已啟用廣域安全，則還需提供讓此工具使用的有效使用者 ID 及密碼。

### **-profileName *profileName***

如果您未配置預設設定檔，則提供您要配置的設定檔名稱。

### **-node *nodeName***

節點的名稱。這是選用參數。預設值是本端節點。

### **-server *serverName***

伺服器的名稱。這是選用參數。

### **-cluster *clusterName***

叢集名稱 *clusterName*。這是選用參數。

### **-remove**

指定此選項來移除事件收集程式應用程式。如果您未指定此選項，則預設會配置應用程式。

### **-silent**

此選項只能搭配 `remove` 選項一起使用。它會讓此工具不輸出任何提示。這是選用參數。

**註：**如果您未指定 `-node`、`-server` 或 `-cluster` 參數，則在配置期間系統會提示您輸入部署目標。

## 範例：啟動此工具

若要啟動此工具來使用名為 `server1` 的伺服器，請輸入下列其中一項指令。

在 **Linux** 和 **UNIX** 平台上：

```
setupObserver.sh -server server1
```

在 **i5/OS** 平台上：

```
setupObserver -server server1
```

在 **Windows** 平台上：

```
setupObserver.bat -server server1
```

您會看到下列「指令」功能表：

- 1) Install the Observer application
- 2) Remove the Observer application and related objects
- 3) Change configuration settings of an installed Observer
  
- 0) Exit Menu

## 使用此工具

下列主題說明如何使用此工具來進行特定作業。

### 相關工作

第 223 頁的『使用 `setupObserver` 工具來配置商業程序編排器觀察程式』

本文說明如何使用互動式功能表導向工具，安裝「商業程序編排器觀察程式」應用程式的實例，並配置它來連接到特定事件收集程式的資料來源。

第 237 頁的『使用 `Script` 來移除商業程序編排器配置』

您可以使用這項作業，移除伺服器或叢集中的「商業流程管理程式」、「人力作業管理程式」、「商業程序編排器瀏覽器」及「商業程序編排器觀察程式」配置及相關聯的資源。

### 相關參考

第 226 頁的『變更商業程序編排器觀察程式的配置參數』

調整「商業程序編排器觀察程式」及事件收集程式應用程式的配置參數對於啓用驗證及改進效能很重要。

## 驗證商業程序編排器觀察程式

在安裝和配置「商業程序編排器觀察程式」之後，請驗證觀察程式是否正常運作。

「商業程序編排器觀察程式」資料庫最初是空的。

1. 執行會產生商業事件的動作，例如，使用「商業程序編排器瀏覽器」來啓動程序實例。
2. 在瀏覽器中，開啓 URL `http://host:port/context_root/` 來啓動「商業程序編排器觀察程式」。其中，`host` 是執行應用程式伺服器的主機名稱，`port` 是應用程式伺服器的埠號（預設值是 9080），而 `context_root` 通常是 `bpcobserver`。
3. 驗證畫面上是否顯示您預期可供使用的事件。如果沒看到事件，請稍候幾分鐘，重新啓動事件收集程式應用程式，然後重新整理瀏覽器視圖。

**註：**若使用 `BPCEventTransformerMaxWaitTime` 和 `BPCEventTransformerToleranceTime` 的預設值，則可能要經過 20 分鐘之後才會觸發轉換程式，且事件收集程式表格中的事件也停留得夠久，然後才處理並可供使用。如需這些參數的相關資訊，包括如何變更及測試建議值，請參閱第 226 頁的『變更商業程序編排器觀察程式的配置參數』。

4. 若發生問題，請參閱第 559 頁的『商業程序編排器觀察程式疑難排解』。

「商業程序編排器觀察程式」正在運作中。

---

## 啓動商業程序編排器

在配置「商業程序編排器」之後，您必須重新啓動受影響的伺服器或叢集。

若要啓動「商業程序編排器」：

1. 如果您是在伺服器上配置「商業程序編排器」，請重新啓動該伺服器。
2. 如果您是在叢集上配置「商業程序編排器」，請重新啓動該叢集。
3. 請確定應用程式伺服器的 `SystemOut.log` 檔案中沒有任何錯誤訊息。在叢集上，請檢查叢集中所有應用程式伺服器的日誌。

4. 驗證「商業流程管理程式」及「人力作業管理程式」應用程式已順利啟動：在管理主控台上，選取應用程式 → 企業應用程式，驗證其名稱開頭為 `BPEContainer_scope` 和 `TaskContainer_scope` 的應用程式已啟動。

如果您是在應用程式伺服器上配置「商業程序編排器」，則其中的 `scope` 值是 `nodeName_serverName`，如果您是在叢集上配置「商業程序編排器」，則此值為 `clusterName`。

「商業程序編排器」在執行中。

您可以開始驗證「商業程序編排器」是否在運作。

---

## 驗證商業程序編排器是否運作

執行「商業程序編排器」安裝驗證應用程式。

1. 使用管理主控台或 `wsadmin` 指令，安裝 `install_root/installableApps/bpcivt.ear` 中的應用程式。企業應用程式在安裝之後會處於已停止狀態，而內含的任何程序和作業範本會處於已啟動狀態。必須啟動應用程式，才能建立程序或作業實例。
2. 視您配置「商業程序編排器」的位置而定，請確定：
  - 應用程式伺服器正在執行中。
  - 至少有一個叢集成員正在執行中。
3. 確定資料庫系統及傳訊服務正在執行中。
4. 選取應用程式 `BPCIVTApp`，然後按一下**啟動**，啟動應用程式。
5. 驗證應用程式是否運作。使用 Web 瀏覽器，開啓下列頁面：

`http://app_server_host:port_no/bpcivt`

其中 `app_server_host` 是應用程式伺服器的主機的網路名稱，而 `port_no` 是虛擬主機所使用的埠號，您在安裝 `bpcivt.ear` 檔案時已將 `IVT Web` 模組對映至此埠號。該埠號視系統配置而定。您應該會看到一則指出成功的訊息。

6. 選擇性的：停止並移除 `bpcivt` 應用程式。
7. 如果發生錯誤，原因可能是下列其中一個：
  - 如果「商業程序編排器」無法存取資料庫，請檢查資料庫系統是否執行中、是否已正確配置所有資料庫用戶端，以及是否已正確定義資料來源。請確定資料來源的使用者 `ID` 及密碼是有效的。
  - 如果「商業程序編排器」無法讀取輸入佇列，請檢查傳訊服務是否執行中，並確定已正確定義 `JMS` 提供者及 `JMS` 資源。

您的「商業程序編排器」配置的基本功能可以運作。

如果已配置其他選用組件，例如「商業程序編排器觀察程式」、「商業程序編排器瀏覽器」或人員目錄提供者，則需要個別地測試它們。

## 瞭解商業程序編排器的啟動行為

本主題說明「商業程序編排器」為何直到所有企業應用程式都已啟動時才可使用。

啟動或重新啟動「商業程序編排器」時，除非啟動所有企業應用程式，否則不會處理內部佇列中的訊息。此行為無法變更。在重新啟動期間，「商業流程管理程式」及

「人力作業管理程式」無法使用的時間，視啓動所有企業應用程式所需的時間而定。此行為可用來防止導覽任何相關聯且未執行的企業應用程式的程序。

在所有應用程式都已啓動之前開始處理內部佇列中的訊息，會導致 `ClassNotFoundException` 異常。

---

## 聯合已配置商業程序編排器的獨立式節點

如果您的伺服器不是在開發模式中執行，您可以將位於獨立式設定檔的伺服器聯合至新的部署管理程式 `Cell`。

部署管理程式正在執行中，且您知道其主機名稱和埠號。「商業程序編排器」是在獨立式設定檔中的伺服器上配置。獨立式設定檔中的「商業程序編排器」資料庫必須可供部署管理程式 `Cell` 從遠端存取。因此，您的伺服器不能以使用內嵌式 `Derby` 資料庫的「商業程序編排器」配置範例為基礎。此外，傳訊引擎資料庫的資料庫必須可從遠端存取，亦即，它不能是 `Derby Embedded`，也不能是 `FILESTORE`。

您有一個以上的應用程式 (內含商業程序或人力作業) 在獨立式伺服器上執行，且您想要將此伺服器聯合至網路部署環境中。

1. 如果節點包含大量應用程式，請增加管理連接器的逾時值。
2. 從指令行中，執行 `addNode` 指令，並指定 `-includeapps` 和 `-includebuses` 選項。如需此指令及可能發生的錯誤的詳細資訊，請參閱 `WebSphere Application Server Network Deployment` 資訊中心的 `addNode` 指令。例如，如果部署管理程式的主機名稱是 `dmgr_host`，且使用埠 `dmgr_port`，請輸入指令：

```
addNode dmgr_host dmgr_port -includeapps -includebuses
```

例如，如果部署管理程式的主機名稱是 `any.hostname.com`，且使用埠 `9043`，而您的設定檔名稱是 `ProcSvr07`、使用者 ID 是 `admin`、密碼是 `secret`，請輸入指令：

```
addNode any.hostname.com 9043 -profileName ProcSvr07 -username admin  
-password secret -includeapps -includebuses
```

如果有任何先決要件不符，畫面上會顯示錯誤訊息。否則，伺服器會停止，且伺服器會聯合至新的部署管理程式 `Cell`。

3. 啓動伺服器，讓變生效。
4. 如果您無法存取伺服器上執行的商業應用程式，請使用部署管理程式上的管理主控台，確定應用程式伺服器的虛擬主機和別名定義符合新的 `Cell`。

現在，您的應用程式在相同的伺服器上執行，但伺服器現在位於可用部署管理程式來管理的 `Cell` 中。

如有必要，您可以將伺服器提昇至叢集。



---

## 第 5 章 移除商業程序編排器配置

您可以使用這項作業，移除商業程序儲存器、人力作業儲存器、「商業程序編排器瀏覽器」、「商業程序編排器觀察程式」及相關聯的資源。

1. 請確定所有獨立式伺服器、資料庫及應用程式伺服器 (或每個叢集至少一個應用程式伺服器) 正在執行中。
2. 對於包含人力作業或商業程序的每一個企業應用程式，請停止所有人力作業範本及所有商業程序範本。
3. 解除安裝包含人力作業或商業程序的所有企業應用程式。
4. 請執行下列其中一個動作：
  - 若要移除「商業程序編排器」配置、「商業程序編排器瀏覽器」、「商業程序編排器觀察程式」、事件收集程式及相關聯的資源，請執行『使用 Script 來移除商業程序編排器配置』。
  - 如果您要重複使用現有的部分配置，請執行第 240 頁的『使用管理主控台來移除商業程序編排器配置』。

已移除「商業程序編排器」配置。

---

### 使用 Script 來移除商業程序編排器配置

您可以使用這項作業，移除伺服器或叢集中的「商業流程管理程式」、「人力作業管理程式」、「商業程序編排器瀏覽器」及「商業程序編排器觀察程式」配置及相關聯的資源。

您必須先停止所有程序和作業範本、刪除所有程序和作業實例，然後停止並移除包含商業程序或人力作業的所有企業應用程式，才能移除「商業程序編排器」配置。

1. 切換至「商業程序編排器」config 目錄：

在 Windows 平台上，請輸入指令：

```
cd install_root\ProcessChoreographer\config
```

在 Linux、UNIX 及 iOS 平台上，輸入：

```
cd install_root/ProcessChoreographer/config
```

2. 執行 Script `bpeunconfig.jacl`。在下列情況，也要指定適當的選項：
  - 若為獨立式伺服器，請停止應用程式伺服器，然後使用 `-conntype NONE` 選項。這個步驟確保任何 Derby 資料庫都不會鎖定，而且可自動移除。
  - 在網路部署環境中，請執行 Script，如下所示：
    - 如果部署管理程式不在執行中，請使用 `-conntype NONE` 選項，在部署管理程式上執行 Script。
    - 如果部署管理程式正在執行中，請停止要從中移除配置的應用程式伺服器，然後執行 Script，但省略 `-conntype NONE` 選項。

在要移除「商業程序編排器」配置的應用程式伺服器節點上執行 Script 時，Script 可以自動刪除任何本端 Derby 資料庫。

- 如果 WebSphere 管理安全已啓用，則也要指定使用者 ID 及密碼：  
-user *userID* -password *password*
- 如果未從預設設定檔中移除配置，則也要指定設定檔名稱：  
-profileName *profileName*

選項	敘述
若為 <b>Linux</b> 或 <b>UNIX</b> 上的單一伺服器	輸入下列指令： <pre>install_root/bin/wsadmin.sh -f bpeunconfig.jacl -server Server -node Node [-deleteDB deleteDatabase]</pre>
若為 <b>Windows</b> 上的單一伺服器	輸入下列指令： <pre>install_root\bin\wsadmin.bat -f bpeunconfig.jacl -server Server -node Node [-deleteDB deleteDatabase]</pre>
若為 <b>i5/OS</b> 上的單一伺服器	輸入下列指令： <pre>install_root/bin/wsadmin -f bpeunconfig.jacl -server Server -node Node [-deleteDB deleteDatabase]</pre>
若為 <b>Linux</b> 或 <b>UNIX</b> 上的叢集	輸入下列指令： <pre>install_root/bin/wsadmin.sh -f bpeunconfig.jacl -cluster Cluster</pre>
若為 <b>Windows</b> 上的叢集	輸入指令： <pre>install_root\bin\wsadmin.bat -f bpeunconfig.jacl -cluster Cluster</pre>
若為 <b>i5/OS</b> 上的叢集	輸入下列指令： <pre>install_root/bin/wsadmin -f bpeunconfig.jacl -cluster Cluster</pre>

其中：

*userID* 使用者 ID。

*password*

使用者 ID 的密碼。

*profileName*

要配置的設定檔名稱。如果您在配置預設設定檔，則這是選用的選項。

*Server* 應用程式伺服器的名稱。如果只有一個伺服器存在，則這是選用參數。

*Node* 節點的名稱。這是選用的。如果忽略節點，則會使用本端節點。

*Cluster* 叢集名稱。

*deleteDatabase*

布林值，指定是否刪除任何 Derby Embedded 資料庫及 FILESTORE 目錄：

**yes**

**no**

若要使用這個選項，伺服器必須不在執行中。如果您沒有任何非 Derby Embedded 資料庫，而且使用了這個選項，則在執行 Script 之後，您可以跳過步驟 第 239 頁的 4。

如果省略必要的參數，系統會提示您輸入它。

3. 選擇性的: 刪除「商業程序編排器」使用的資料庫。

下列情況同時適用於「商業程序編排器」資料庫與傳訊資料庫：

- `bpeunconfig.jacl` Script 會列出已移除的配置所使用的資料庫。此資料庫清單也會寫入 `install_root/profiles/profileName/logs/bpeunconfig.log` 日誌檔中。您可以使用這個清單，來識別您可能想要手動刪除的資料庫。
  - 使用 Derby 資料庫作為「商業程序編排器」資料庫時，`bpeunconfig.jacl` Script 會選擇性地移除資料庫，除非遭到執行中應用程式伺服器鎖定。如果資料庫已鎖定，請停止伺服器，然後使用 `-conntype NONE` 選項。
  - 當 Derby 是傳訊資料庫時，`bpeunconfig.jacl` Script 會選擇性地移除資料庫，除非遭到執行中應用程式伺服器鎖定。如果資料庫已鎖定，請停止伺服器，然後使用 `-conntype NONE` 選項。
  - 使用 FILESTORE 作為「商業程序編排器」傳訊引擎訊息儲存庫時，使用 `bpeunconfig.jacl` Script 的 `-deleteDB yes` 選項也將刪除相關聯的目錄。
  - 若要移除「商業程序編排器觀察程式」資料庫，請啟動工具來設定事件收集程式（如第 230 頁的『`setupEventCollector` 工具』中所述），並選取選項 **Drop the database schema of the Event Collector and Observer**。
4. 選擇性的: 檢查 `bpeunconfig.log` 日誌檔。它位於 `profile_root` 的 `logs` 子目錄中。
  5. 選擇性的: 如果已使用 WebSphere MQ，請刪除「商業程序編排器」使用的佇列管理程式。
  6. 選擇性的: 手動復原 `bpeunconfig.jacl` 未復原的其餘設定。`bpeunconfig.jacl` 不會復原下列設定，因為無法判斷其他元件是否仍然需要這些設定：
    - 啟用 `WorkAreaService`
    - 啟用 `ApplicationProfileService`
    - 啟用 `ObjectPoolService`
    - 啟用 `StartupBeansService`
    - 啟用 `CompensationService`
    - 啟用 `WorkareaPartitionService`
    - 設定 WebSphere 變數

The Business Process Choreographer applications and associated resources (such as scheduler, data sources, listener ports, connection factories, queue destinations, activation specs, work area partition, mail session, and authentication aliases) have been removed.

## 使用工具來移除商業程序編排器觀察程式及事件收集程式

移除伺服器或叢集中的「商業程序編排器觀察程式」及事件收集程式應用程式及相關聯的資源。

1. 如果您已配置「商業程序編排器觀察程式」，請執行第 232 頁的『`setupObserver` 工具』，並指定 `-remove` 選項
2. 如果您已配置「商業程序編排器」事件收集程式，請執行第 230 頁的『`setupEventCollector` 工具』，並指定 `-remove` 選項。
3. 選擇性的: 如果您已安裝 Java 使用者定義的函數，請移除它們。
4. 選擇性的: 刪除資料庫。

已移除「商業程序編排器觀察程式」及事件收集程式應用程式。

---

## 使用管理主控台來移除商業程序編排器配置

您可以使用這項作業，移除部份或全部「商業程序編排器」配置，包括「商業程序編排器瀏覽器」、「商業程序編排器觀察程式」及相關聯的資源。

您必須先停止所有程序及作業範本、刪除所有作業及程序實例，然後解除安裝包含商業程序或人力作業的所有企業應用程式，才能移除「商業程序編排器」配置。

1. 解除安裝「商業程序編排器」企業應用程式。

a. 顯示企業應用程式。

在管理主控台中，選取**應用程式** → **企業應用程式**。

b. 識別「商業程序編排器」安裝的範圍。

尋找以下列名稱開頭的應用程式名稱：

- BPEContainer\_scope 是「商業流程管理程式」應用程式。
- TaskContainer\_scope 是「人力作業管理程式」應用程式。
- BPCExplorer\_scope 是「商業程序編排器瀏覽器」應用程式。

其中，*scope* 的值視您的配置而定：

- 如果已在應用程式伺服器上配置「商業程序編排器」，則 *scope* 具有值 *nodeName\_serverName* – 即使伺服器稍後已提昇至叢集也一樣。
  - 如果「商業程序編排器」在叢集上配置，則 *scope* 的值為 *clusterName*。
- c. 選擇性的: 如果已配置「商業程序編排器」，請解除安裝「商業流程管理程式」及「人力作業管理程式」應用程式。選取 BPEContainer\_scope 及 TaskContainer\_scope，然後按一下**解除安裝** → **確定** → **儲存**。
- d. 選擇性的: 如果已配置「商業程序編排器瀏覽器」，請解除安裝它。
- 如果已使用預設環境定義根目錄 /bpc，請選取 BPCExplorer\_scope，然後按一下**解除安裝** → **確定** → **儲存**。
  - 否則，選取 BPCExplorer\_scope\_context\_root，然後按一下**解除安裝** → **確定** → **儲存**。

**註:** 解除安裝「商業程序編排器觀察程式」及事件收集程式應用程式說明於步驟 第 244 頁的 10 中。

2. 移除下列您不要重複使用的所有或任何資源：

- a. 選擇性的: 尋找「商業程序編排器」資料來源 (預設名稱是 BPEDataSourcedbType)，並在記下其名稱及其相關聯的鑑別資料別名 (如果有的話)，以及「Java 命名和目錄介面 (JNDI)」名稱後，再移除它 (預設名稱為 jdbc/BPEDB)。

若要尋找資料來源：

- 1) 按一下**資源** → **JDBC** → **資料來源**。
  - 2) 對於**範圍**，選取已配置「商業程序編排器」的伺服器或叢集。
- b. 選擇性的: 若資料庫不是 Derby 資料庫，請移除步驟 2 所識別的資料來源的 JDBC 提供者，除非其中還有您仍需要進一步使用的資料來源。按一下**資源** → **JDBC** → **JDBC 提供者**，選取您資料庫的 JDBC 驅動程式，然後按一下**刪除**。

**註:** 如果「商業程序編排器」配置對 Derby Embedded 資料庫使用內建的預設 JDBC 提供者，則無法刪除這個 JDBC 提供者。

- c. 選擇性的: 移除適當的 Connection Factory 和佇列。
  - 若為預設傳訊，在移除 Connection Factory 之前，請記下其相關聯的鑑別資料別名。然後，移除 JMS Connection Factory 和 JMS 佇列。
    - 1) 按一下**資源** → **JMS** → **Connection Factory**。對於**範圍**，選取已配置「商業程序編排器」的伺服器或叢集。然後，選取 Connection Factory，再按一下**刪除**。
    - 2) 按一下**資源** → **JMS** → **佇列**。對於**範圍**，選取已配置「商業程序編排器」的伺服器或叢集。然後，選取佇列，再按一下**刪除**。
  - 若為 WebSphere MQ，請移除 JMS Queue Connection Factory 及 JMS 佇列。
    - 1) 按一下**資源** → **JMS** → **Queue Connection Factory**。對於**範圍**，選取已配置「商業程序編排器」的伺服器或叢集。然後，選取 Connection Factory，再按一下**刪除**。
    - 2) 按一下**資源** → **JMS** → **佇列**。對於**範圍**，選取已配置「商業程序編排器」的伺服器或叢集。然後，選取佇列，再按一下**刪除**。

若為商業程序儲存器，JNDI 名稱通常如下所示：

```
jms/BPECF  
jms/BPECFC  
jms/BFMJMSReplyCF  
jms/BPEIntQueue  
jms/BPERetQueue  
jms/BPEHldQueue  
jms/BFMJMSAPIQueue  
jms/BFMJMScallbackQueue  
jms/BFMJMSReplyQueue
```

若為人力作業儲存器，JNDI 名稱通常如下所示：

```
jms/HTMCF  
jms/HTMIntQueue  
jms/HTMHldQueue
```

- d. 選擇性的: 如果您是使用 WebSphere 預設傳訊作為 JMS 提供者，請移除啟動規格。
  - 1) 按一下**資源** → **JMS** → **啟動規格**。對於**範圍**，選取已配置「商業程序編排器」的伺服器或叢集。
  - 2) 移除下列啟動規格：
    - BPEInternalActivationSpec
    - BFMJMSAS
    - HTMInternalActivationSpec
- e. 選擇性的: 如果您是使用 WebSphere MQ 作為 JMS 提供者，請移除伺服器的接聽器埠。
  - 1) 按一下**伺服器** → **應用程式伺服器** → **serverName**。
  - 2) 在「通訊」下，按一下**傳訊** → **訊息接聽器服務** → **接聽器埠**。
  - 3) 在「應用程式伺服器」窗格上，移除下列接聽器埠：
    - BPEInternalListenerPort

BPEHoldListenerPort  
HTMInternalListenerPort

如果已在叢集上配置「商業程序編排器」，請對叢集的每一個成員重複這個步驟。

- f. 選擇性的: 刪除鑑別資料別名。
    - 1) 按一下**安全** → **安全管理、應用程式及基礎架構**，然後在**鑑別**區段中，展開**Java 鑑別和授權服務 (JAAS)**，再按一下**J2C 鑑別資料**。
    - 2) 如果步驟 第 240 頁的 2 指定的資料來源有鑑別資料別名，請移除該別名。如果未從 6.0.x 版移轉您的「商業程序編排器」配置，則名稱視下列方式中的部署目標而定：
      - 在名稱為 *serverName* 的伺服器、名稱為 *nodeName* 的節點上配置「商業程序編排器」時，名稱通常為 `BPCDB_nodeName.serverName_Auth_Alias`。
      - 在名稱為 *clusterName* 的叢集上配置「商業程序編排器」時，名稱通常為 `BPCDB_clusterName_Auth_Alias`
    - 3) 如果步驟 第 241 頁的 2c 識別的任何 Connection Factory 有鑑別資料別名，請小心移除該別名：
      - 如果未從 6.0.x 版移轉您的「商業程序編排器」配置，則名稱為 `BPC_Auth_Alias`，而且網路部署環境中的所有「商業程序編排器」配置都會共用它。

**警告:** 如果您正要移除前次的「商業程序編排器」配置，請只移除這個鑑別別名，否則剩餘的「商業程序編排器」配置將停止運作。
      - 如果已從 6.0.x 版移轉您的「商業程序編排器」配置，則名稱通常為 `cellName/BPEAuthDataAliasJMS_scope`，其中 *cellName* 是 Cell 的名稱，而 *scope* 識別部署目標。您可以移除這個鑑別別名，不會影響其他「商業程序編排器」配置。
  - g. 選擇性的: 移除資料來源 JNDI 名稱的排程器配置。
    - 1) 按一下**資源** → **排程器**。
    - 2) 對於**範圍**，選取已配置「商業程序編排器」的伺服器或叢集。
    - 3) 在「排程器」窗格上，記下工作管理員的 JNDI 名稱，然後選取並刪除名稱為 `BPEScheduler` 的排程器。
  - h. 選擇性的: 移除工作管理員。
    - 1) 按一下**資源** → **非同步 Bean** → **工作管理員**。
    - 2) 對於**範圍**，選取已配置「商業程序編排器」的伺服器或叢集。
    - 3) 在「工作管理員」窗格上，選取並刪除您在步驟 2g 中已記下其 JNDI 名稱的工作管理員。
    - 4) 也請刪除具有 JNDI 名稱 `wm/BPENavigationWorkManager` 的工作管理員。
  - i. 選擇性的: 移除工作區分割區。
    - 1) 按一下**伺服器** → **應用程式伺服器** → **serverName**。
    - 2) 在**儲存器設定**區段下，展開**商業程序服務**，再按一下**工作區分割區服務**。
    - 3) 在「應用程式伺服器」窗格上，選取並刪除工作區分割區 `BPECompensation`。
- 如果已在叢集上配置「商業程序編排器」，請對叢集的每一個成員重複這個步驟。



- j. 選擇性的: 移除郵件階段作業。
  - 1) 按一下**資源** → **郵件** → **郵件提供者**。
  - 2) 對於**範圍**，選取 **Cell=cellName**，其中 *cellName* 是 Cell 的名稱。
  - 3) 按一下**內建郵件提供者**。
  - 4) 在**其他內容**區段下，選取**郵件階段作業**。
  - 5) 選取並刪除 **HTMailSession\_scope**，其中 *scope* 是步驟 第 240 頁的 1b 中指定的範圍
3. 選擇性的: 如果您對「商業程序編排器」使用 WebSphere 預設傳訊，則可以刪除匯流排成員、匯流排及資料來源：
  - a. 按一下**服務整合** → **匯流排** → **BPC.cellName.Bus**，然後在**拓撲**區段下，按一下**傳訊引擎**。
  - b. 選取傳訊引擎：
    - **nodeName.serverName-BPC.cellName.Bus** (如果您在伺服器上配置「商業程序編排器」)。
    - **clusterName-BPC.cellName.Bus** (如果您在叢集中配置「商業程序編排器」)。

註： 如果已將「商業程序編排器」配置為使用遠端傳訊引擎，則 *nodeName.serverName* 或 *clusterName* 將不符合您已配置「商業程序編排器」的名稱部署目標。
  - c. 在**其他內容**下，選取**訊息儲存庫**。
    - 如果訊息儲存庫類型為 **DATASTORE**，請記下資料來源的 JNDI 名稱。在伺服器上，資料來源的 JNDI 名稱通常為 **jdbc/com.ibm.ws.sib/nodeName.serverName-BPC.cellName.Bus**。在叢集上，資料來源的 JNDI 名稱通常為 **jdbc/com.ibm.ws.sib/clusterName-BPC.cellName.Bus**。
    - 如果訊息儲存庫類型為 **FILESTORE**，請記下 **Log**、**Permanent store** 及 **Temporary store** 的路徑。
  - d. 跳至**服務整合** → **匯流排** → **BPC.cellName.Bus**，然後在**拓撲**區段下，按一下**匯流排成員**，並移除下列其中一個名稱所指定的匯流排成員：
    - **nodeName:serverName** (如果已在伺服器上配置「商業程序編排器」)。
    - **clusterName** (如果已在叢集上配置「商業程序編排器」)。
  - e. 選擇性的: 如果您已移除匯流排 **BPC.cellName.Bus** 的最後一個成員，則也可以移除匯流排。
  - f. 如果您在步驟 3c 中記下的訊息儲存庫類型為 **DATASTORE**，請按一下**資源** → **JDBC** → **資料來源**。傳訊引擎的範圍可能與您已配置「商業程序編排器」的部署目標不相同。必要的話，嘗試不同範圍，尋找您在步驟 3c 中記下的 JNDI 名稱。如果資料來源是針對 **Derby** 資料庫，請記下資料庫的檔案系統路徑。如果已在叢集上配置「商業程序編排器」，請對叢集的每一個成員重複這個步驟。
4. 刪除 **BPC\_REMOTE\_DESTINATION\_LOCATION** 變數。按一下**環境** → **WebSphere** 變數，對於**範圍**，選取已配置「商業程序編排器」的部署目標，然後選取並刪除 **BPC\_REMOTE\_DESTINATION\_LOCATION** 變數。
5. 重新啟動應用程式伺服器或叢集。
6. 按一下**儲存**，將您的所有刪除項目儲存在主要配置中。
7. 選擇性的: 刪除「商業程序編排器」資料庫。

8. 選擇性的: 如果您是使用 WebSphereMQ，請刪除「商業程序編排器」所使用的佇列管理程式。
9. 如果您對「商業程序編排器」使用 WebSphere 預設傳訊，請刪除訊息引擎的訊息儲存庫；因為無法重複使用它。
  - a. 如果您在步驟 第 243 頁的3c 中記下的訊息儲存庫類型為 FILESTORE，請移除您對 Log、Permanent store 及 Temporary store 記下的目錄。
  - b. 如果您在步驟 第 243 頁的3c 中記下的訊息儲存庫類型為 DATASTORE，請移除資料來源指向的資料庫。如果這是 Derby 資料來源，請刪除您在步驟 第 243 頁的 3f 中記下的檔案系統路徑。通常，Derby 資料庫位置如下：
    - 在 Linux、UNIX 及 i5/OS 平台上：
 

```
profile_root/databases/com.ibm.ws.sib/  
nodeName.serverName-BPC.cellName.Bus
```
    - 在 Windows 平台上：
 

```
profile_root\databases\com.ibm.ws.sib\  
nodeName.serverName-BPC.cellName.Bus
```
10. 選擇性的: 如果您已配置「商業程序編排器觀察程式」，請執行：
  - a. 第 245 頁的『使用管理主控台來移除商業程序編排器觀察程式』（針對每一個觀察程式應用程式實例）。
  - b. 『使用管理主控台來移除商業程序編排器事件收集程式』（針對每一個事件收集程式應用程式實例）。

已移除「商業程序編排器」配置。

## 使用管理主控台來移除商業程序編排器事件收集程式

使用這項作業來移除「商業程序編排器」事件收集程式配置，以及「商業程序編排器觀察程式」所需的相關聯資源。

1. 顯示企業應用程式。

在管理主控台中，選取應用程式 → 企業應用程式。

2. 解除安裝「商業程序編排器」事件收集程式應用程式。選取 BPCECollector\_scope 的勾選框，然後按一下解除安裝 → 確定。其中 scope 識別已配置事件收集程式的伺服器或叢集。
3. 刪除目的地佇列：
  - a. 按一下服務整合 → 匯流排 → **CommonEventInfrastructure\_Bus**。
  - b. 在目的地資源下，按一下目的地。
  - c. 選取下列目的地佇列：
    - BPCCEICConsumerQueueDestination\_scope
    - BPCTransformerQueueDestination\_scope

其中 scope 識別已配置事件收集程式的伺服器或叢集。

- d. 按一下刪除。
4. 刪除 JMS Queue Connection Factory：
  - a. 按一下資源 → **JMS** → **Queue Connection Factory**。

- b. 對於**範圍**，選取已配置事件收集程式的伺服器或叢集。
  - c. 選取 `BPCCEIConsumerQueueConnectionFactory` 的勾選框。
  - d. 按一下**刪除**。
5. 刪除 JMS 佇列：
    - a. 按一下**資源** → **JMS** → **佇列**。
    - b. 選取下列佇列的勾選框：
      - `BPCCEIConsumerQueue_scope`
      - `BPCTransformerQueue_scope`
    - c. 按一下**刪除**。
  6. 刪除 JMS 啟動規格：
    - a. 按一下**資源** → **JMS** → **啟動規格**。
    - b. 選取下列啟動規格的勾選框：
      - `BPCCEIConsumerActivationSpec`
      - `BPCTransformerActivationSpec`
    - c. 按一下**刪除**。
  7. 刪除具有 `BFMEvents` 的伺服器範圍的事件設定檔群組：
    - a. 按一下**服務整合** → **共用事件基礎架構** → **事件服務**。
    - b. 在**其他內容**下，按一下**事件服務**。
    - c. 按一下**預設共用事件基礎架構事件伺服器**。
    - d. 在**其他內容**下，按一下**事件群組**。
    - e. 選取 `BFMEvents` 的勾選框。
    - f. 按一下**刪除**。
  8. 如果已從 6.0.2 版移轉您的配置，請刪除鑑別資料別名：
    - a. 按一下**安全** → **安全管理、應用程式及基礎架構** → **鑑別** → **Java 鑑別和授權服務 (JAAS)** → **J2C 鑑別資料**。
    - b. 選取 `BPCEventCollectorJMSAuthenticationAlias_scope`。
    - c. 按一下**刪除**。
  9. 按一下**儲存**，將變更儲存在主要配置中。
  10. 執行下列 Script，捨棄「商業程序編排器觀察程式」所使用的資料庫、綱目及表格空間，在 Windows 平台上，這些 Script位於目錄 `install_root\dbscripts\ProcessChoreographer\database_type` 中，而在 Linux、UNIX 及 i5/OS 平台上，位於目錄 `install_root/dbscripts/ProcessChoreographer/database_type` 中：
    - `dropSchema_Observer.sql`
    - `dropTablespace_Observer.sql`

已移除「商業程序編排器」事件收集程式配置。

## 使用管理主控台來移除商業程序編排器觀察程式

使用這項作業來移除「商業程序編排器觀察程式」配置及相關聯的資源。

1. 顯示企業應用程式。

在管理主控台中，選取**應用程式** → **企業應用程式**。

2. 尋找「商業程序編排器觀察程式」實例。

尋找名稱以 `BPCObserver_scope` 開頭的應用程式。

- 如果「商業程序編排器觀察程式」安裝在應用程式伺服器上，則 `scope` 的值為 `nodeName_serverName`。
- 如果「商業程序編排器觀察程式」安裝在叢集上，則 `scope` 的值為 `clusterName`。

**註：**如果環境定義根目錄不是預設 `/bpcobserver`，則應用程式名稱也會附加環境定義根目錄 `_contextRoot`。

3. 解除安裝「商業程序編排器觀察程式」應用程式。選取您想要刪除的應用程式實例，然後按一下**解除安裝** → **確定** → **儲存**。

已移除「商業程序編排器觀察程式」配置。

---

## 第 3 篇 管理





---

## 第 6 章 管理商業程序編排器

您可以使用管理主控台或 Script 來管理「商業程序編排器」。

---

### 使用管理主控台來管理商業程序編排器

說明可以使用管理主控台來執行的管理動作。

#### 管理伺服器的補償服務

使用管理主控台，可以在啓動應用程式伺服器時自動啓動補償服務，以及指定回復日誌的位置及大小上限。

當商業程序在伺服器上執行時，必須在應用程式伺服器上啓動補償服務。補償服務是用來管理更新，在完成程序前，某些交易可能更新。當設定新的應用程式伺服器時，依預設會啓用補償服務。

您可以使用管理主控台，檢視並變更應用程式伺服器的補償服務內容。

1. 顯示管理主控台。
2. 在導覽窗格中，按一下**伺服器** → **應用程式伺服器** → **server\_name**。
3. 在「配置」標籤的「儲存器設定」下，按一下**儲存器服務** → **補償服務**。這個動作會顯示一個具有補償服務內容的畫面。請確定已選取在**伺服器啓動時啓用服務**勾選框。如果您在叢集上執行商業程序，請對叢集中的每一個伺服器啓用補償服務。
4. 選擇性的: 必要的話，變更補償服務內容。
5. 按一下**確定**。
6. 若要儲存您的配置，請在管理主控台視窗的「訊息」框中按一下**儲存**。

#### 查詢及重新播放失敗的訊息，使用管理主控台

本文說明如何檢查和重新播放無法處理的商業程序或人力作業的任何訊息。

如果在處理訊息時發生問題，訊息會移至保留佇列或存放佇列。這項作業說明如何判斷是否有任何失敗訊息，以及如何將這些訊息再次傳送至內部佇列。

1. 若要檢查存放及保留佇列中有多少訊息：
  - a. 如果已在叢集上配置「商業程序編排器」，請按一下**伺服器** → **伺服器叢集** → **server\_name** 或 **伺服器** → **叢集** → **cluster\_name**。
  - b. 在**配置**標籤的「商業整合」下，按一下**商業程序編排器**。選擇下列其中一個選項：
    - 若為商業程序，請按一下**商業流程管理程式**。
    - 若為人力作業，請按一下**人力作業管理程式**。

存放佇列及保留佇列中的訊息數會顯示在執行時期標籤的一般內容下。

2. 如果「存放」佇列或「保留」佇列包含訊息，您可以將訊息移至內部工作佇列。

按下列其中一個選項：

- 若為商業程序：**重新播放存放佇列**或**重新播放保留佇列**

- 若為人力作業：**重新播放存放佇列**

**註：**已啟用 WebSphere 管理安全時，只限具有操作員權限的使用者才能看到重新播放按鈕。

「商業程序編排器」會嘗試再次處理所有重新播放的訊息。

#### 相關概念

第 25 頁的『從基礎架構失敗回復』

「商業流程管理程式」提供處理暫時基礎架構失敗的機能。

## 重新整理失敗訊息計數

使用管理主控台來重新整理商業程序或人力作業的失敗訊息計數。

重新整理之前，在存放佇列及保留佇列中顯示的訊息數及訊息異常狀況數會保持固定。這項作業說明如何更新和顯示這些佇列上的訊息數及訊息異常狀況數。

1. 選取適當的應用程式伺服器。

如果已在叢集上配置「商業程序編排器」，請按一下**伺服器** → **應用程式伺服器** → **server\_name** 或**伺服器** → **叢集** → **cluster\_name**。

2. 重新整理訊息計數。

a. 在**配置**標籤的「商業整合」下，按一下**商業程序編排器**。選擇下列其中一個選項：

- 若為商業程序，請按一下**商業流程管理程式**。
- 若為人力作業，請按一下**人力作業管理程式**。

b. 在**執行時期**標籤上，按一下**重新整理訊息計數**。

**一般內容**下會顯示下列更新的值：

- 若為商業程序：在存放佇列及保留佇列中的訊息數
- 若為人力作業：在存放佇列中的訊息數
- 如果在存取佇列時發生任何異常，「訊息」異常狀況欄位中會顯示訊息文字。

在這個頁面上，您也可以重新播放這些佇列中的訊息。

#### 相關概念

第 25 頁的『從基礎架構失敗回復』

「商業流程管理程式」提供處理暫時基礎架構失敗的機能。

## 使用管理主控台來重新整理人員查詢結果

人員查詢的結果是固定的。請使用管理主控台來重新整理人員查詢。

「商業程序編排器」會快取人員查詢的結果，其根據執行時期資料庫中的人員目錄 (例如「輕量型目錄存取通訊協定 (LDAP)」伺服器) 加以評估。如果變更人員目錄，則您可以強制重新評估所指派的人員。

若要重新整理人員查詢：

1. 如果已在叢集上配置「商業程序編排器」，請按一下**伺服器** → **伺服器叢集** → **server\_name** 或**伺服器** → **叢集** → **cluster\_name**。
2. 在**配置**標籤的「商業整合」下，按一下**商業程序編排器** → **人力作業管理程式**。

3. 在**執行時期**標籤上，按一下**重新整理人員查詢**。已重新整理所有人員查詢。

**註:** 已啓用 WebSphere 管理安全時，只限具有操作員權限的使用者才能看到重新整理按鈕。

以這種方式來重新整理人員查詢結果會增加應用程式和資料庫的負載。請考慮使用下列替代方法。

#### 相關概念

第 76 頁的『管理人員指派準則及人員解析結果』

與作業授權角色相關聯的人員指派準則，適用於已部署作業範本或作業實例的生命期限。

#### 相關工作

第 267 頁的『使用管理指令重新整理人員查詢結果』

人員查詢的結果是固定的。使用管理指令來重新整理人員查詢。

第 252 頁的『使用重新整理常駐程式來重新整理人員查詢結果』

如果想要設定定期及自動重新整理所有過期人員查詢結果，請使用這個方法。

## 使用管理主控台來啓用共用基礎事件及審核追蹤

使用這項作業，讓「商業程序編排器」事件當作「共用基礎事件」產生至 共用事件基礎架構、儲存在審核追蹤中或兩者。

若要變更「商業流程管理程式」或「人力作業管理程式」的狀態觀察程式設定，您可以在「配置」標籤上永久地變更，或在「執行時期」標籤上暫時地變更。您在「配置」或「執行時期」標籤上所做的任何選擇，將影響在適當的儲存器中執行的所有應用程式。若要讓變更同時影響「商業流程管理程式」及「人力作業管理程式」，您必須分別變更這兩者的設定。

## 使用管理主控台來變更已配置的記載基礎架構

使用這項作業來變更審核日誌的狀態觀察程式記載功能，或配置的共用事件基礎架構記載功能。

在「配置」標籤上所做的選擇會在下一次啓動伺服器時生效。每當啓動伺服器時，選擇的設定仍然有效。

變更配置，如下所示：

1. 顯示商業流程管理程式或人力作業管理程式窗格。
  - a. 如果已在叢集上配置「商業程序編排器」，請按一下**伺服器** → **應用程式伺服器** → **server\_name** 或**伺服器** → **叢集** → **cluster\_name**。
  - b. 在**配置**標籤的「商業整合」下，按一下**商業程序編排器**。選擇下列其中一個選項：
    - 若為商業程序，請按一下**商業流程管理程式**。
    - 若為人力作業，請按一下**人力作業管理程式**。
2. 在**配置**標籤的「一般內容」區段中，選取要啓用的記載。狀態觀察程式彼此無關：您可以啓用或停用其中之一或兩者都停用。

#### 啓用「共用事件基礎架構」記載功能

選取這個勾選框，啓用根據「共用事件基礎架構」來產生事件。

### 啓用審核日誌

選取這個勾選框，將審核日誌事件儲存在「商業程序編排器」資料庫的審核追蹤表格中。

3. 接受變更。
  - a. 按一下**確定**。
  - b. 在「訊息」框中，按一下**儲存**。

已依您的要求來設定狀態觀察程式。重新啓動伺服器之後，變更就會生效。

重新啓動伺服器，讓變生效。如果已在叢集上配置「商業程序編排器」，請重新啓動叢集。

### 使用管理主控台來變更階段作業的記載基礎架構

使用這項作業來變更審核日誌的狀態觀察程式記載功能，或階段作業的共用事件基礎架構記載功能。

在「執行時期」標籤上所做的選擇會立即生效。選擇的設定在下次啓動伺服器之後就會失效。

變更階段作業基礎架構，如下所示：

1. 顯示商業流程管理程式或人力作業管理程式窗格。
  - a. 如果已在叢集上配置「商業程序編排器」，請按一下**伺服器** → **應用程式伺服器** → **server\_name** 或 **伺服器** → **叢集** → **cluster\_name**。
  - b. 在**配置**標籤的「商業整合」下，按一下**商業程序編排器**。選擇下列其中一個選項：
    - 若為商業程序，請按一下**商業流程管理程式**。
    - 若為人力作業，請按一下**人力作業管理程式**。
2. 在**執行時期**標籤的「一般內容」區段中，選取要啓用的記載。狀態觀察程式彼此無關：您可以啓用或停用其中之一或兩者都停用。

#### 啓用「共用事件基礎架構」記載功能

選取這個勾選框，啓用根據「共用事件基礎架構」來產生事件。

#### 啓用審核日誌

選取這個勾選框，將審核日誌事件儲存在「商業程序編排器」資料庫的審核追蹤表格中。

3. 按一下**確定**，以接受變更。

已依您的要求來設定狀態觀察程式。

### 使用重新整理常駐程式來重新整理人員查詢結果

如果想要設定定期及自動重新整理所有過期人員查詢結果，請使用這個方法。

人員查詢是由指定的人員目錄提供者來解析。結果儲存在「商業程序編排器」資料庫中。若要最佳化權限效能，會快取擷取的查詢結果。當呼叫人員查詢重新整理常駐程式時，會檢查快取內容是否最新。

爲了讓人員查詢結果保持最新，會提供一個常駐程式，定期重新整理過期的人員查詢結果。常駐程式會重新整理所有過期的快取人員查詢結果。

1. 開啟「人力作業管理程式」的自訂內容頁面：
  - a. 如果已在叢集上配置「商業程序編排器」，請按一下**伺服器** → **應用程式伺服器** → **server\_name** 或 **伺服器** → **叢集** → **cluster\_name**。
  - b. 在**配置**標籤的「商業整合」下，按一下**商業程序編排器** → **人力作業管理程式**。
  - c. 選擇下列其中一個選項：
    - 若要永久變更設定，請按一下**配置**標籤。在重新啟動應用程式伺服器之後，變更便為有效的。
    - 若要暫時變更設定，請按一下**執行時期**標籤。變更會立即有效，但是在下次應用程式伺服器重新啟動時將重設。
2. 在**人員查詢重新整理排程**欄位中，使用 WebSphere CRON 行事曆支援的語法來輸入排程。此值會決定常駐程式何時將重新整理任何過期的人員查詢結果。預設值是 "0 0 1 \* \* ?"，它會導致每天早上 1 點重新整理。
3. 在**人員查詢結果的逾時**欄位中，輸入以秒為單位的新值。此值決定將人員查詢結果視為有效的期限。在這個時段之後，人員查詢結果不再視為有效，而且下次執行常駐程式時，將重新整理人員查詢。預設值為一小時。
4. 按一下**確定**。
5. 儲存變更。若要讓您在「配置」標籤上執行的變生效，請重新啟動應用程式伺服器。

新的有效期限值只適用於新的人員查詢，它不適用於現有的人員查詢。

#### 相關概念

第 76 頁的『管理人員指派準則及人員解析結果』

與作業授權角色相關聯的人員指派準則，適用於已部署作業範本或作業實例的生命期限。

---

## 使用 Script 來管理商業程序編排器

說明可以使用 Script 來執行的管理動作。

#### 相關資訊

第 225 頁的『使用 Script 啟用商業程序編排器的記載功能』

本文說明如何使用 `setStateObserver.py` Script，來啟用或停用「商業程序編排器」的「共用事件基礎架構 (CEI)」或審核事件。

## 使用管理指令來刪除審核日誌項目

您可以使用管理指令，刪除「商業流程管理程式」的部分或全部審核日誌項目。

在開始此程序之前，必須符合下列條件：

- 應用程式伺服器必須執行中，因為將透過其刪除審核日誌項目。亦即，無法使用 `wsadmin` 的 `-conntype none` 選項，因為需要伺服器連線。
- 如果已在叢集上配置「商業程序編排器」，則至少有一個叢集成員必須正在執行中。
- 已啟用 WebSphere 管理安全時，您必須具有操作員權限。

您可以使用 `deleteAuditLog.py` Script，從資料庫刪除「商業流程管理程式」的審核日誌項目。

1. 切換至管理 Script 所在的「商業程序編排器」子目錄。

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 i5/OS 平台上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

2. 刪除審核日誌表格中的項目。

在 Windows 平台上，輸入下列一個以上的指令。指令的差異以粗體表示：

```
install_root\bin\wsadmin -lang jython -f deleteAuditLog.py  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root\bin\wsadmin -lang jython -f deleteAuditLog.py  
-node nodeName  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root\bin\wsadmin -lang jython -f deleteAuditLog.py  
-cluster cluster_name  
[-profileName profileName]  
[options]
```

在 Linux 及 UNIX 平台上，輸入下列其中一個以上的指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py  
-node nodeName  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py  
-cluster cluster_name  
[-profileName profileName]  
[options]
```

在 i5/OS 平台上，輸入下列其中一個以上的指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py  
-node nodeName  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py  
-cluster cluster_name  
[-profileName profileName]  
[options]
```



其中：

**-cluster** *cluster\_name*

叢集名稱。如果已針對 WebSphere 叢集配置「商業流程管理程式」，則這是必要選項。

**-node** *nodeName*

當指定伺服器名稱時，這是選用選項。此名稱可識別節點。預設值是本端節點。

**-server** *server\_name*

伺服器的名稱。如果未指定叢集名稱，則這是必要選項。

**-profileName** *profileName*

使用者定義的設定檔名稱。如果您未使用預設的設定檔，請指定此選項。

可用的選項如下：

**-all**

刪除資料庫中的所有審核日誌項目。刪除會在多個交易中執行。每一個交易都會刪除在截塊參數中指定的項目數目，或預設數目。

**-time** *timestamp*

刪除所有比您對 *timestamp* 指定的時間還要早的審核日誌項目。使用的時間是世界標準時間 (UTC)。它的時間格式必須是：YYYY-MM-DD[<sup>T</sup>HH:MM:SS]。如果只指定年、月及日，則小時、分鐘及秒會設為 00:00:00。

*-time* 與 *-processtime* 選項互斥。

**-processtime** *timestamp*

刪除所有屬於程序的審核日誌項目，這個程序是在您對 *timestamp* 指定的時間之前完成。使用與 *-time* 參數相同的時間格式。

*-time* 與 *-processtime* 選項互斥。

**-slice** *size*

與 *-all* 參數搭配使用，*size* 指定在每一個交易中併入的項目數。最佳值視您的資料庫系統的可用日誌大小而定。越高值需要更少的交易，但是可能超出資料庫日誌空間。越低值可能導致 Script 需要更長時間來刪除完畢。截塊參數的預設大小為 250。

**註：**「清除未用人員查詢 Script」的 jacl 版本 (deleteAuditLog.jacl) 即將棄用。此 Script 位於 ProcessChoreographer 目錄的 util 子目錄中，且接受這裡所述的相同參數，但必須省略 *-lang jython* 選項。

## 刪除無效的程序範本

您可以使用管理指令，從「商業程序編排器」資料庫中刪除不再有效的程序範本。

在您開始這個程序之前，將刪除範本的應用程式伺服器必須在執行中。亦即，無法使用 wsadmin 的 *-conntype none* 選項，因為需要伺服器連線。即使已啓用 WebSphere 管理安全，無需特殊權限，也可執行這個指令。如果已在叢集上配置「商業程序編排器」，則至少有一個叢集成員必須執行中。

使用 deleteInvalidProcessTemplate.py Script，從資料庫移除那些範本及所有屬於它們的物件，這些範本及物件並不包含在 WebSphere 配置儲存庫中的任何對應的有效應用程式

中。如果使用者已取消應用程式安裝，或未將其儲存在「配置儲存庫」，則可能會發生這種狀況。這些範本通常沒有影響。它們不會顯示在「商業程序編排器瀏覽器」中。

很少有無法過濾這些範本的情況。接著，必須利用下列 Script，從資料庫移除它們。

您無法使用 Script，從資料庫移除有效應用程式的範本。如果對應應用程式有效，則會檢查這個條件並擲出 ConfigurationError 異常。

1. 切換至管理 Script 所在的「商業程序編排器」子目錄。

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 iOS 平台上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

2. 從資料庫中刪除不再有效的商業程序範本。

若要在 Windows 系統上刪除不再有效的商業程序範本，請輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py
    -server server_name
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py
    -server server_name
    -node nodeName
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py
    -cluster cluster_name
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

若要在 Linux 及 UNIX 系統上刪除不再有效的商業程序範本，請輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py
    -server server_name
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py
    -server server_name
    -node nodeName
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py
    -cluster cluster_name
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

若要在 i5/OS 系統上刪除不再有效的商業程序範本，請輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py
    -server server_name
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py
    -server server_name
    -node nodeName
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py
    -cluster cluster_name
    -templateName templateName
    -validFrom validFromString
    [-profileName profileName]
```

其中：

**-cluster** *cluster\_name*

叢集的名稱。如果已針對 WebSphere 叢集配置「商業程序編排器」，則這是必要選項。您可以指定叢集名稱或伺服器名稱及節點名稱。

**-node** *nodeName*

當指定伺服器名稱時，這是選用選項。此名稱可識別節點。預設值是本端節點。您可以指定伺服器名稱及節點名稱或叢集名稱。

**-server** *server\_name*

伺服器的名稱。如果未指定叢集名稱，則這是必要選項。您可以指定伺服器名稱及節點名稱或叢集名稱。

**-templateName** *templateName*

將刪除的程序範本或作業範本的名稱。

**-validFrom** *validFromString*

顯示範本在管理主控台生效的日期 (以世界標準時間表示)。字串必須具有下列格式：'yyyy-MM-ddThh:mm:ss' (年、月、日、T、小時、分鐘、秒)。例如：2005-01-31T13:40:50

**-profileName** *profileName*

使用者定義的設定檔名稱。如果您未使用預設的設定檔，請指定此選項。

**註：** 清除未用程序 Script 的 jacl 版本 (deleteInvalidProcessTemplate.jacl) 已遭淘汰。此 Script 位於 ProcessChoreographer 目錄的 util 子目錄中，且接受這裡所述的相同參數，但必須省略 `-lang jython` 選項。

## 刪除無效的人力作業範本

您可以使用管理指令，從「商業程序編排器」資料庫中刪除不再有效的人力作業範本。

在您開始這個程序之前，將刪除範本的應用程式式伺服器必須在執行中。亦即，無法使用 wsadmin 的 `-conntype none` 選項，因為需要伺服器連線。即使已啓用 WebSphere

管理安全，無需特殊權限，也可執行這個指令。如果已在叢集上配置「商業程序編排器」，則至少有一個叢集成員必須執行中。

使用 `deleteInvalidTaskTemplate.py` Script，從資料庫移除那些範本及所有屬於它們的物件，這些範本及物件並不包含在 `WebSphere` 配置儲存庫中的任何對應的有效應用程式中。如果使用者已取消應用程式安裝，或未將其儲存在「配置儲存庫」，則可能會發生這種狀況。這些範本通常沒有影響。它們不會顯示在「商業程序編排器瀏覽器」中。

很少有無法過濾這些範本的情況。接著，必須利用下列 Script，從資料庫移除它們。

您無法使用 Script，從資料庫移除有效應用程式的範本。如果對應應用程式有效，則會檢查這個條件並擲出 `ConfigurationError` 異常。

1. 切換至管理 Script 所在的「商業程序編排器」子目錄。

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 iOS 平台上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

2. 從資料庫中刪除不再有效的人力作業範本。

若要在 Windows 系統上刪除不再有效的人力作業範本，請輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
    -server server_name
    -templateName templateName
    -validFrom validFromString
    -nameSpace nameSpace
    [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
    -server server_name
    -node nodeName
    -templateName templateName
    -validFrom validFromString
    -nameSpace nameSpace
    [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
    -cluster cluster_name
    -templateName templateName
    -validFrom validFromString
    -nameSpace nameSpace
    [-profileName profileName]
```

若要在 UNIX 及 Linux 系統上刪除不再有效的人力作業範本，請輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py
    -server server_name
    -templateName templateName
    -validFrom validFromString
    -nameSpace nameSpace
    [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py
    -server server_name
    -node nodeName
    -templateName templateName
```

```
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py  
-cluster cluster_name  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

若要在 i5/OS 系統上刪除不再有效的人力作業範本，請輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py  
-server server_name  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py  
-server server_name  
-node nodeName  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py  
-cluster cluster_name  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

其中：

**-cluster** *cluster\_name*

叢集名稱。如果已針對 WebSphere 叢集配置「商業程序編排器」，則這是必要選項。您可以指定叢集名稱或伺服器名稱及節點名稱。

**-node** *nodeName*

當指定伺服器名稱時，這是選用選項。此名稱可識別節點。預設值是本端節點。您可以指定伺服器名稱及節點名稱或叢集名稱。

**-server** *server\_name*

伺服器的名稱。如果未指定叢集名稱，則這是必要選項。您可以指定伺服器名稱及節點名稱或叢集名稱。

**-templateName** *templateName*

將刪除的程序範本或作業範本的名稱。

**-validFrom** *validFromString*

顯示範本在管理主控台生效的日期 (以世界標準時間表示)。字串必須具有下列格式：'yyyy-MM-ddThh:mm:ss' (年、月、日、T、小時、分鐘、秒)。例如：2005-01-31T13:40:50

**-nameSpace** *nameSpace*

作業範本的目標名稱空間。

**-profileName** *profileName*

使用者定義設定檔的名稱。如果不是使用預設設定檔，請指定此選項。

**註：**清除未用人員查詢 Script 的 jacl 版本 (`deleteInvalidTaskTemplate.jacl`) 已遭淘汰。此 Script 位於 `ProcessChoreographer` 目錄的 `util` 子目錄中，且接受這裡所述的相同參數，但必須省略 `-lang jython` 選項。

## 刪除已完成的程序實例

使用管理指令，可以選擇性地從「商業程序編排器」資料庫刪除任何為完成、終止或失敗結束狀態的最上層程序實例。

在您開始這個程序之前，將刪除程序實例的應用程式伺服器必須在執行中。亦即，無法使用 `wsadmin` 的 `-conntype none` 選項，因為需要伺服器連線。即使已啟用 `WebSphere` 管理安全，無需特殊權限，也可執行這個指令。如果已在叢集上配置「商業程序編排器」，則至少有一個叢集成員必須執行中。

如果最上層程序實例處於下列其中一種結束狀態，則視為已完成：已完成、已終止、結束或已失敗。您可以指定準則，以選擇性地從資料庫刪除最上層程序實例及其所有關聯資料 (例如活動實例、子程序實例及行內作業實例)。

1. 切換至管理 Script 所在的「商業程序編排器」子目錄。

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 i5/OS 平台上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

2. 從資料庫刪除程序實例。

在 Windows 系統上，輸入下列指令：

```
install_root\bin\wsadmin -lang jython -f deleteCompletedProcessInstances.py  
[[[-node nodeName] -server server_name) | (-cluster cluster_name)]  
(-all | -finished | -terminated | -failed )  
[-templateName templateName [-validFrom timestamp]]  
[-startedBy userID ]  
[-completedBefore timestamp]  
[-profileName profileName]
```

在 Linux 及 UNIX 系統上，輸入下列指令：

```
install_root/bin/wsadmin.sh -lang jython -f deleteCompletedProcessInstances.py  
[[[-node nodeName] -server server_name) | (-cluster cluster_name)]  
(-all | -finished | -terminated | -failed )  
[-templateName templateName [-validFrom timestamp]]  
[-startedBy userID ]  
[-completedBefore timestamp]  
[-profileName profileName]
```

在 i5/OS 系統上，輸入下列指令：

```
install_root/bin/wsadmin -lang jython -f deleteCompletedProcessInstances.py  
[[[-node nodeName] -server server_name) | (-cluster cluster_name)]  
(-all | -finished | -terminated | -failed )  
[-templateName templateName [-validFrom timestamp]]  
[-startedBy userID ]  
[-completedBefore timestamp]  
[-profileName profileName]
```



其中：

**-node** *nodeName*

當指定伺服器名稱時，這是選用選項。此名稱可識別節點。預設值是本端節點。您可以指定伺服器名稱及節點名稱或叢集名稱。

**-server** *server\_name*

伺服器的名稱。如果未指定叢集名稱，則這是必要選項。您可以指定伺服器名稱及節點名稱或叢集名稱。

**-cluster** *cluster\_name*

叢集的名稱。如果已對 WebSphere 叢集配置「商業程序編排器」，則這是必要選項。您可以指定叢集名稱或伺服器名稱及節點名稱。

**-all|-finished|-terminated|-failed**

指定將根據其狀態來刪除程序實例。您可以指定 finished、terminated、failed 或 all 的組合。

**-templateName** *templateName*

您可以選擇性地指定將刪除的程序範本的名稱。如果指定這個選項，也可以使用 validFrom

**-validFrom** *timestamp*

顯示範本在管理主控台生效的日期（以世界標準時間表示）。此選項只能搭配 templateName 選項一起使用。timestamp 字串具有下列格式：'yyyy-MM-dd[Thh:mm:ss]'（年、月、日、T、小時、分鐘、秒）。例如，2006-11-20T12:00:00。如果只指定年、月及日，則小時、分鐘及秒會設為 00:00:00。

**-startedBy** *userID*

您可以選擇性地刪除由給定的使用者 ID 啟動的已完成程序實例。

**-completedBefore** *timestamp*

您可以選擇性地刪除在給定的時間之前完成的已完成程序實例。timestamp 字串具有下列格式：'yyyy-MM-ddThh:mm:ss'（年、月、日、T、小時、分鐘、秒）。例如，2006-07-20T12:00:00

**-profileName** *profileName*

使用者定義的設定檔名稱。如果您未使用預設的設定檔，請指定此選項。

例如，若要刪除在伺服器 *myServer* 的節點 *myNode* 上執行的所有程序實例，這些程序實例處於完成狀態，而且由使用者 Antje 啟動，請執行下列指令：

```
wsadmin -lang jython -f deleteCompletedProcessInstances.py
        -node myNode -server myServer
        -finished
        -startedBy Antje
```

已從資料庫刪除已完成的程序實例。

## 從觀察程式資料庫刪除資料

您可以使用管理指令，以選擇性地從「商業程序編排器觀察程式」資料庫刪除程序實例的所有資料，這些實例需符合指定條件。刪除不需要的資料可以改善產生報告的效能。

您可以利用三種方法，刪除程序實例的觀察程式資訊：

- 若要刪除已在指定時間之前達到結束狀態 `deleted` 的程序實例的觀察程式資料，您必須提供下列參數：`-deletedBefore timestamp`。
- 若要刪除特定範本版本的程序實例的觀察程式資料，而不管其現行狀態，您必須提供下列參數：`-templateName templateName -validFrom timestamp`。
- 若要刪除已在指定時間之前達到指定狀態的特定範本版本的程序實例的觀察程式資料，您必須提供下列參數：`-force -templateName template_name -validFrom timestamp -state state -reachedBefore timestamp`，其中 `-templateName template_name` 及 `-validFrom timestamp` 是選用項目。

若要使用這些方法中任何一個，請執行下列動作：

1. 請確定應用程式伺服器正在執行中。如果已在叢集上配置「商業程序編排器」，則至少有一個叢集成員必須執行中。
2. 切換至管理 `Script` 所在的「商業程序編排器」子目錄。

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 i5/OS 平台上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

3. 輸入指令以從資料庫刪除特定程序實例的觀察程式資料。

在 Windows 平台上，輸入下列指令：

```
install_root\bin\wsadmin -lang jython
-f observerDeleteProcessInstanceData.py
[ -user user_ID -password password ]
( [-node node_name] -server server_name ) | (-cluster cluster_name )
[ -profileName profile_name ]
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
-deletedBefore timestamp
| ( -templateName template_name -validFrom timestamp )
| ( -force [-templateName template_name -validFrom timestamp]
-state state -reachedBefore timestamp )
)
```

在 Linux 及 UNIX 平台上，輸入下列指令：

```
install_root/bin/wsadmin.sh -lang jython
-f observerDeleteProcessInstanceData.py
[ -user user_ID -password password ]
( [-node node_name] -server server_name ) | (-cluster cluster_name )
[ -profileName profile_name ]
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
-deletedBefore timestamp
| ( -templateName template_name -validFrom timestamp )
| ( -force [-templateName template_name -validFrom timestamp]
-state state -reachedBefore timestamp )
)
```

在 i5/OS 平台上，輸入下列指令：

```
install_root/bin/wsadmin -lang jython
-f observerDeleteProcessInstanceData.py
[ -user user_ID -password password ]
( [-node node_name] -server server_name ) | (-cluster cluster_name )
```

```

[ -profileName profile_name ]
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
  -deletedBefore timestamp
  | ( -templateName template_name -validFrom timestamp )
  | ( -force [-templateName template_name -validFrom timestamp]
      -state state -reachedBefore timestamp )
)

```

輸入以下指令：

```

install_root/bin/wsadmin.sh -lang jython
-f observerDeleteProcessInstanceData.py
[ -user user_ID -password password ]
( [-node node_name] -server server_name) | (-cluster cluster_name )
[ -profileName profile_name ]
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
  -deletedBefore timestamp
  | ( -templateName template_name -validFrom timestamp )
  | ( -force [-templateName template_name -validFrom timestamp]
      -state state -reachedBefore timestamp )
)

```

其中：

**-user *user\_ID* -password *password***

如果已在伺服器上啟用 WebSphere 管理安全，則您必須提供一個有權從資料庫進行刪除的使用者 ID，以及相關聯的密碼。

**-node *node\_name***

此名稱可識別節點。這是選用參數。預設值是本端節點。

**-server *server\_name***

伺服器的名稱。預設值是預設伺服器。如果指定這個參數，您不得指定 `cluster` 參數。

**-cluster *cluster\_name***

叢集的名稱。如果指定這個參數，您不得指定 `server` 參數。

**-profileName *profile\_name***

使用者定義的 WebSphere 設定檔名稱。如果您未使用預設的設定檔，請指定此選項。

**-dataSource *datasource\_JNDI\_name***

因為伺服器或叢集可以具有多個觀察程式資料庫，所以這個參數會識別指令將處理的資料庫。預設值為 `jdbc/BPEDB`。

**-dbSchemaName *dbSchemaName***

如果利用特定綱目名稱來設定「觀察程式」資料庫，請使用這個參數。

**-deletedBefore *timestamp***

刪除已在指定時間之前達到狀態 `deleted` 的程序實例的所有觀察程式資料。

*timestamp*

日期和時間是以「世界標準時間 (UTC)」表示，格式如下：`'yyyy-MM-dd[Thh:mm:ss]'` (年、月、日、T、小時、分鐘、秒)。例如，`2008-07-20T12:00:00`。如果只指定年、月及日，則小時、分鐘及秒會設定為 `00:00:00`。

**-templateName** *template\_name*

刪除屬於指定的範本版本的實例的所有觀察程式資料。

**-validFrom** *timestamp*

如果指定 **templateName** 選項，則這是必要項目。

*timestamp*

日期和時間是以「世界標準時間 (UTC)」表示，格式如下：'*yyyy-MM-ddThh:mm:ss*' (年、月、日、T、小時、分鐘、秒)。例如，2008-07-20T12:00:00。

**-force**

強制刪除所有範本或指定範本版本 (已在指定時間之前達到指定狀態) 的程序實例的所有觀察程式資料。如果使用這個選項，也須指定選項 **-state** 及 **-reachedBefore**。 **-templateName** 及 **-validFrom** 是選用選項。

**-state** *state*

指定下列其中一種狀態：

running、terminated、suspended、failed、finished、compensated。

**-reachedBefore** *timestamp*

指定時間，屆時必須已達到指定狀態。

*timestamp*

日期和時間是以「世界標準時間 (UTC)」表示，格式如下：'*yyyy-MM-dd[Thh:mm:ss]*' (年、月、日、T、小時、分鐘、秒)。例如，2008-07-20T12:00:00。如果只指定年、月及日，則小時、分鐘及秒會設定為 00:00:00。

例如，若要刪除從 2007 年 1 月 2 日中午開始有效的程序範本 *my\_template* 實例 (執行於已在 2007 年 7 月 20 日中午之前啟動的伺服器 *my\_server* 中的節點 *my\_node*) 的所有觀察程式資料，請執行下列指令：

```
wsadmin -lang jython -f observerDeleteProcessInstanceData.py
        -node my_node -server my_server
        -force -templateName my_template -validFrom 2007-01-02T12:00:00
        -state running -reachedBefore 2007-07-20T12:00:00
```

如果順利完成，工具會報告已刪除觀察程式資料的實例數，以及已從資料庫刪除的表格項目數。否則，會報告錯誤資訊，而且不會變更資料庫。

## 查詢及重新播放失敗的訊息，使用管理指令

使用管理指令來判斷商業程序或人力作業是否有任何失敗訊息，如果有的話，則嘗試重新處理。

在開始此程序之前，必須符合下列條件：

- 要查詢或重新播放訊息的應用程式伺服器必須正在執行中。亦即，無法使用 **wsadmin** 的 **-conntype none** 選項，因為需要伺服器連線。
- 如果已在叢集上配置「商業程序編排器」，則至少有一個叢集成員必須正在執行中。
- 已啟用 **WebSphere** 管理安全時，您必須具有操作員權限。

如果在處理內部訊息時發生問題，此訊息最後會移至「保留」佇列或「存放」佇列。若要判斷是否有任何失敗訊息，以及將這些訊息再次傳送至內部佇列：

1. 切換至管理 Script 所在的「商業程序編排器」子目錄。

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 i5/OS 平台上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

2. 同時查詢保留佇列與存放佇列上的失敗訊息數。

在 Windows 系統上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root\bin\wsadmin -lang jython -f queryNumberOfFailedMessages.py  
-cluster cluster_name  
[ -bfm | -htm ]  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f queryNumberOfFailedMessages.py  
-node nodeName  
-server server_name  
[ -bfm | -htm ]  
[-profileName profileName]
```

在 Linux 及 UNIX 系統上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin.sh -lang jython -f queryNumberOfFailedMessages.py  
-cluster cluster_name  
[ -bfm | -htm ]  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f queryNumberOfFailedMessages.py  
-node nodeName  
-server server_name  
[ -bfm | -htm ]  
[-profileName profileName]
```

在 i5/OS 系統上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin -lang jython -f queryNumberOfFailedMessages.py  
-cluster cluster_name  
[ -bfm | -htm ]  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f queryNumberOfFailedMessages.py  
-node nodeName  
-server server_name  
[ -bfm | -htm ]  
[-profileName profileName]
```

其中：

**-cluster** *cluster\_name*

叢集的名稱。如果已針對 WebSphere 叢集配置「商業程序編排器」，則這是必要選項。

**-node** *nodeName*

當指定伺服器名稱時，這是選用選項。此名稱可識別節點。預設值是本端節點。

**-server** *server\_name*

伺服器名稱。如果未指定叢集名稱，則這是必要選項。

### **-bfm|-htm**

這些是選用關鍵字。如果兩個選項都未指定，則預設會同時顯示商業程序與人力作業的所有失敗訊息。如果您只想要顯示「商業流程管理程式」存放及保留佇列中的訊息數，請指定 **-bfm** 選項。如果您只想要顯示「人力作業管理程式」存放佇列中的訊息數，請指定 **-htm** 選項。

### **-profileName** *profileName*

使用者定義設定檔的名稱。如果不是使用預設設定檔，請指定此選項。

如果您要檢查本端節點上的伺服器，請輸入：

```
wsadmin -lang jython -f queryNumberOfFailedMessages.py -server server_name
```

### 3. 重新播放存放佇列、保留佇列或兩者上的所有失敗訊息。

在 Windows 系統上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py  
-cluster cluster_name  
-queue replayQueue  
[ -bfm | -htm ]  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py  
-node nodeName  
-server server_name  
-queue replayQueue  
[ -bfm | -htm ]  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py  
-server server_name  
-queue replayQueue  
[ -bfm | -htm ]  
[-profileName profileName]
```

在 Linux 及 UNIX 系統上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py  
-cluster cluster_name  
-queue replayQueue  
[ -bfm | -htm ]  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py  
-node nodeName  
-server server_name  
-queue replayQueue  
[ -bfm | -htm ]  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py  
-server server_name  
-queue replayQueue  
[ -bfm | -htm ]  
[-profileName profileName]
```

在 i5/OS 系統上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin -lang jython -f replayFailedMessages.py  
-cluster cluster_name  
-queue replayQueue  
[ -bfm | -htm ]  
[-profileName profileName]
```



```
install_root/bin/wsadmin -lang jython -f replayFailedMessages.py
                        -node nodeName
                        -server server_name
                        -queue replayQueue
                        [ -bfm | -htm ]
                        [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f replayFailedMessages.py
                        -server server_name
                        -queue replayQueue
                        [ -bfm | -htm ]
                        [-profileName profileName]
```

其中：

**-queue** *replayQueue*

(選用項目) 指定要重新播放的佇列。*replayQueue* 可以是下列其中一個值：

holdQueue (這是預設值)

retentionQueue (只有當指定 -bfm 選項時才有效)

both (指定 -htm 選項時無效)

**-cluster** *cluster\_name*

叢集的名稱。如果已針對 WebSphere 叢集配置「商業程序編排器」，則這是必要選項。

**-node** *nodeName*

當指定伺服器名稱時，這是選用選項。此名稱可識別節點。預設值是本端節點。

**-server** *server\_name*

伺服器的名稱。如果未指定叢集名稱，則這是必要選項。

**-bfm|-htm**

這些是選用且互斥的關鍵字。如果兩個選項都未指定，則預設會同時重新播放商業程序與人力作業的失敗訊息。如果您只要重新播放商業程序的訊息，請指定 -bfm 選項。如果您只要重新播放人力作業的訊息，請指定 -htm 選項。

**-profileName** *profileName*

使用者定義設定檔的名稱。如果不是使用預設設定檔，請指定此選項。

**註：** 清除未用人員查詢 Script 的 jacl 版本 (replayFailedMessages.jacl) 已遭淘汰。此 Script 位於 ProcessChoreographer 目錄的 util 子目錄中，且接受這裡所述的相同參數，但必須省略 -lang jython 選項。

**相關概念**

第 25 頁的『從基礎架構失敗回復』

「商業流程管理程式」提供處理暫時基礎架構失敗的機能。

## 使用管理指令重新整理人員查詢結果

人員查詢的結果是固定的。使用管理指令來重新整理人員查詢。

在開始此程序之前，必須符合下列條件：

- 要查詢或重新播放訊息的應用程式伺服器必須正在執行中。亦即，無法使用 wsadmin 的 -conntype none 選項，因為需要伺服器連線。

- 如果已在叢集上配置「商業程序編排器」，則至少有一個叢集成員必須正在執行中。
- 已啓用 WebSphere 管理安全時，您必須具有操作員權限。

「商業程序編排器」會快取人員查詢的結果，其根據執行時期資料庫中的人員目錄 (例如「輕量型目錄存取通訊協定 (LDAP)」伺服器) 加以評估。如果變更人員目錄，則您可以強制重新評估所指派的人員。

1. 切換至管理 Script 所在的「商業程序編排器」子目錄。

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 iOS 平台上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

2. 強制重新評估所指派的人員。

在 Windows 平台上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py
  -server server_name
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userlist username{,username}...]
  [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py
  -node nodeName
  -server server_name
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userlist username{,username}...]
  [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py
  -cluster cluster_name
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userlist username{,username}...]
  [-profileName profileName]
```

在 Linux 及 UNIX 平台上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py
  -server server_name
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userlist username{,username}...]
  [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py
  -node nodeName
  -server server_name
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userlist username{,username}...]
  [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py
  -cluster cluster_name
```

```
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

在 i5/OS 平台上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py  
-server server_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py  
-node nodeName  
-server server_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py  
-cluster cluster_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

其中：

**-cluster** *cluster\_name*

叢集的名稱。如果已針對 WebSphere 叢集配置「商業程序編排器」，則這是必要選項。

**-node** *nodeName*

當指定伺服器名稱時，這是選用選項。此名稱可識別節點。預設值是本端節點。

**-server** *server\_name*

伺服器的名稱。如果未指定叢集名稱，則這是必要選項。

**-processTemplate** *templateName*

程序範本的名稱。會重新整理為這個程序範本所指派的人員。

**-taskTemplate** *templateName*

作業範本的名稱。會重新整理為這個作業範本所指派的人員。

**-nameSpace** *nameSpace*

作業範本的名稱空間。

**-userlist** *userName*

以逗點區隔的使用者名稱清單。會重新整理所指派且包含指定名稱的人員。使用者清單可以透過引號來括住。如果省略引號，則使用者清單不得在使用者名稱之間包含空白。

**-profileName** *profileName*

使用者定義的設定檔名稱。如果您未使用預設的設定檔，請指定此選項。

**註：** 如果既未指定任何 *templateName*，也未指定 *userlist*，則會重新整理所有儲存在資料庫中的人員查詢。基於效能原因，您可能想要避免這種情況。

註: 「重新整理人員查詢 Script」的 jacl 版本 (refreshStaffQuery.jacl) 即將棄用。此 Script 位於 ProcessChoreographer 目錄的 util 子目錄中, 且接受這裡所述的相同參數, 但必須省略 `-lang jython` 選項。

### 相關概念

第 76 頁的『管理人員指派準則及人員解析結果』

與作業授權角色相關聯的人員指派準則, 適用於已部署作業範本或作業實例的生命期限。

## 使用管理指令來移除未用的人員查詢結果

使用管理指令, 可以從資料庫移除未用的人員查詢結果。

在開始此程序之前, 必須符合下列條件:

- 應用程式伺服器必須在執行中, 將透過該程式刪除未用人員查詢。亦即, 無法使用 `wsadmin` 的 `-conntype none` 選項, 因為需要伺服器連線。
- 如果已在叢集上配置「商業程序編排器」, 則至少有一個叢集成員必須正在執行中。
- 已啓用 WebSphere 管理安全時, 您必須具有操作員權限。

「商業程序編排器」會針對已評估的人員查詢, 在執行時期資料庫中維護使用者名稱清單。雖然已完成程序實例及人力作業 (其使用人員查詢), 但是會在資料庫中維護使用者名稱清單, 直到解除安裝對應的商業程序應用程式。

如果資料庫大小影響效能, 您可以移除在資料庫表格中快取的未用人員清單。

1. 切換至管理 Script 所在的「商業程序編排器」子目錄。

在 Windows 平台上, 輸入:

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 iOS 平台上, 輸入:

```
cd install_root/ProcessChoreographer/admin
```

2. 移除未用人員清單。

在 Windows 平台上, 輸入下列其中一個指令。指令的差異以粗體表示:

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py  
-server server_name  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py  
-node nodeName  
-server server_name  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py  
-cluster cluster_name  
[-profileName profileName]
```

在 Linux 及 UNIX 平台上, 輸入下列其中一個指令。指令的差異以粗體表示:

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py  
-server server_name  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py
    -node nodeName
    -server server_name
    [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py
    -cluster cluster_name
    [-profileName profileName]
```

在 i5/OS 平台上，輸入下列其中一個指令。指令的差異以粗體表示：

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
    -server server_name
    [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
    -node nodeName
    -server server_name
    [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
    -cluster cluster_name
    [-profileName profileName]
```

其中：

**-cluster** *cluster\_name*

叢集的名稱。如果已針對 WebSphere 叢集配置「商業程序編排器」，則這是必要選項。

**-node** *nodeName*

當指定伺服器名稱時，這是選用選項。此名稱可識別節點。預設值是本端節點。

**-server** *server\_name*

伺服器的名稱。如果未指定叢集名稱，則這是必要選項。

**-profileName** *profileName*

使用者定義的設定檔名稱。如果您未使用預設的設定檔，請指定此選項。

畫面上會顯示已從資料庫刪除的項目數。

**註：**「要清除未用人員查詢的 `Script`」的 `jacl` 版本 (`cleanupUnusedStaffQueryInstances.jacl`) 即將棄用。此 `Script` 位於 `ProcessChoreographer` 目錄的 `util` 子目錄中，且接受這裡所述的相同參數，但必須省略 `-lang jython` 選項。





---

## 第 7 章 商業程序編排器瀏覽器入門

視您的使用者角色而定，您可以使用「商業程序編排器瀏覽器」來管理商業程序及人力作業，或使用您的指派作業。

您可以使用「商業程序編排器瀏覽器」來執行下列作業：

- 如果您是商業管理者，則可以管理商業程序的生命週期，並且可以修復商業程序。例如，您可以重新啟動或強制完成單一活動，或補償整個商業程序。如果補償失敗，您可以重試、跳過或停止程序實例。此外，您可以針對商業程序及活動新增及更新自訂內容。
- 如果您是人力作業管理者，則可以管理人力作業的生命週期，以及管理工作指派。例如，您可以指派責任給使用者，或管理使用者的缺席處理及代理人。您也可以變更人力作業的優先順序及商業種類，以及新增或更新自訂內容。
- 如果您是商業使用者，則可以使用「商業程序編排器瀏覽器」，來使用您的指派作業。例如，您可以起始商業程序、服務及人力作業，以及您可以使用、編輯、儲存、完成或釋放人力作業。此外，您也可以標示您的缺席並定義代理人。

此外，「商業程序編排器瀏覽器」會提供一個搜尋功能，您可以使用這個功能，探索商業程序及其相關活動，以及需要注意的人力作業。例如，您可以檢查這些實例的狀態、在相關實例與範本之間導覽，以及擷取程序狀態的圖形視圖，其中包括相關聯的活動及人力作業。

### 相關工作

第 298 頁的『管理作業範本及作業實例』

使用管理主控台或管理指令，可以管理作業範本。使用「商業程序編排器瀏覽器」，可以使用作業實例。

第 301 頁的『管理指派的工作』

在啓動作業之後，您可能需要管理作業所指派的工作，例如，更適當地分配工作群組成員的工作負荷。

第 299 頁的『建立及啓動作業實例』

您可以從任何您有權使用的作業範本建立及啓動作業實例。

第 300 頁的『使用您的作業』

若要使用作業，您必須要求作業，然後執行完成它所需的動作。

---

## 商業程序編排器瀏覽器使用者介面

「商業程序編排器瀏覽器」是獨立式 Web 應用程式，提供一組管理功能，用來管理商業程序及人力作業。該介面由工作列、導覽窗格及工作區組成。

下圖顯示了「商業程序編排器瀏覽器」使用者介面的佈置。



使用者介面包含以下主要區域。

## 工作列

工作列為所有的使用者提供選項，以登出「商業程序編排器瀏覽器」，指定缺席設定，並存取線上說明。如果您擁有系統管理者權限，則工作列還會包括以下選項：

**自訂** 選取這個選項，以新增或移除此「商業程序編排器瀏覽器」實例導覽窗格的視圖。您也可以定義使用者登入時所看到的視圖。

### 定義視圖

選取這個選項，以定義使用者群組的自訂視圖。

### 定義替代方案

選取這個選項，以定義使用者的缺席設定。

## 導覽窗格

導覽窗格包含用來管理物件的視圖鏈結，例如，您啟動的程序實例或授權您管理的人力作業。預設使用者介面包含商業程序及作業的預先定義視圖的鏈結。

系統管理者可以透過新增及移除導覽窗格的預先定義視圖，以及定義新增到導覽窗格的自訂視圖，自訂預設導覽窗格的內容。所有使用者都可以從導覽窗格定義個人化視圖。

## 工作區

工作區包含用來檢視及管理商業程序及人力作業相關物件的頁面。您可以透過按一下導覽窗格中的鏈結、透過按一下動作列中的動作，或者透過按一下工作區頁面中的鏈結，來存取這些頁面。

## 商業程序編排器瀏覽器導覽窗格

使用導覽窗格，可以存取您用來管理商業程序及人力作業物件的視圖，例如程序實例及工作指派。預設使用者介面包含商業程序及作業的預先定義視圖的鏈結。您也可以定義專屬的個人化視圖，這些視圖會新增至導覽窗格中。此外，如果您是系統管理者，則可以定義所有使用者都可以使用的自訂視圖。

### 可用的動作

下列動作可在導覽窗格中使用：

- 導覽至視圖。

按一下視圖名稱，以導覽至該視圖。






- 收合及展開群組。

按一下導覽窗格中某個項目旁邊的箭頭，以展開或收合該項目。

- 定義新搜尋。

按一下**新搜尋**圖示 ()，以搜尋物件或定義個人化視圖。

根據視圖類型，可以從蹣現功能表使用其他動作。圖示指出蹣現功能表是否可用。

- 若要刪除視圖，請按一下**刪除**圖示 ()。
- 若要修改視圖，請按一下**編輯**圖示 ()。
- 若要建立視圖的副本並修改該副本，請按一下**複製**圖示 ()。
- 若要在清單中上下移動視圖，請按一下**上**圖示 () 或**下**圖示 ()。

### 導覽窗格中的預先定義視圖

預設導覽窗格包含以下視圖群組。根據您的系統管理者是否新增或移除導覽窗格的視圖，在「商業程序編排器瀏覽器」導覽窗格中顯示的視圖會有所不同。如果未定義視圖群組的視圖，則不會顯示此群組。

#### 程序範本

程序範本群組包含以下視圖：

##### 我的程序範本

此視圖顯示程序範本清單。從此視圖中，您可以顯示程序範本及其結構的相關資訊，顯示與範本相關聯的程序實例的清單，並啟動程序實例。

#### 程序實例

程序實例群組包含以下視圖：

##### 由我啟動

此視圖顯示您啟動的程序實例。從此視圖中，您可以監視程序實例的進度，並列出與其相關的活動、程序或作業。

### 由我管理

此視圖顯示授權您管理的程序實例。從此視圖中，您可以處理程序實例，例如，暫停或回復程序，或者監視程序實例中活動的進度。

### 重要程序

此視圖顯示處於執行中狀態的程序實例，其中包含處於停止狀態的活動。從此視圖中，您可以處理程序實例，或者列出活動，並在稍後處理它們。

### 終止的程序

此視圖顯示處於終止狀態的程序實例。從此視圖中，您可以處理這些程序實例。

### 失敗的補償

此視圖顯示微流程已失敗的補償動作。

### 活動實例

根據預設值，活動實例群組預設不包含任何視圖。因此，此群組不會顯示在預設導覽窗格中。

### 作業範本

作業範本群組包含以下視圖：

#### 我的作業範本

此視圖顯示作業範本的清單。從此視圖中，您可以建立並啓動作業實例，並顯示與範本相關聯的作業實例的清單。

### 作業實例

作業實例群組包含以下視圖：

#### 我的待辦事項

此視圖顯示授權您處理的作業實例的清單。從此視圖中，您可以使用作業實例、釋出您已要求的作業實例，或者將作業實例轉移給另一位使用者。

#### 所有作業

此視圖顯示您是擁有者、可能的擁有者或編輯者的所有作業。從此視圖中，您可以使用作業實例、釋出您已要求的作業實例，或者將作業實例轉移給另一位使用者。

#### 由我起始

此視圖顯示您起始的作業實例。從此視圖中，您可以使用作業實例、釋出您已要求的作業實例，或者將作業實例轉移給另一位使用者。

#### 由我管理

此視圖顯示授權您管理的作業實例。從此視圖中，您可以處理作業實例，例如，暫停或回復程序、建立作業實例的工作項目，或者顯示作業實例現行工作項目的清單。

#### 我的呈報

此視圖顯示已登入使用者的所有呈報。

## 視圖類型



導覽窗格可以包含以下類型的視圖。根據視圖，可以從蹣現功能表使用其他動作。

預設導覽窗格中的預先定義視圖。


只有在導覽窗格沒有由「自訂導覽樹狀結構及登入視圖」頁面中的系統管理者變更時，才能使用這些視圖群組。蹦現功能表無法用於這些視圖。

自訂視圖與預先定義視圖已由系統管理者新增至導覽窗格。

商業使用者可以按一下視圖名稱，並導覽至該視圖。若為系統管理者，則可以使用蹦現功能表。

- 預先定義視圖由預先定義視圖圖示指出：。系統管理者可以使用蹦現功能表變更導覽窗格中這些視圖的位置。
- 自訂視圖由自訂視圖圖示指出：。系統管理者可以刪除、編輯、複製及移除這些視圖。

個人化視圖。

這些視圖由自訂視圖圖示指出：。只有建立視圖的使用者才看得到這些視圖。使用者可以刪除、編輯、複製及移動視圖。

---

## 啓動商業程序編排器瀏覽器

「商業程序編排器瀏覽器」是一種 Web 應用程式，它可以安裝為商業程序儲存器配置的一部分。在您開始從 Web 瀏覽器使用「商業程序編排器瀏覽器」之前，必須先安裝商業程序儲存器、人力作業儲存器及「商業程序編排器瀏覽器」應用程式，而且應用程式必須執行中。

若要啓動「商業程序編排器瀏覽器」，請完成下列步驟。

1. 將您的 Web 瀏覽器引導至「商業程序編排器瀏覽器」URL。

URL 採用下列格式。URL 的值視如何針對您的安裝作業來配置虛擬主機及環境定義根目錄而定。

`http://app_server_host:port_no/context_root`

其中：

*app\_server\_host*

應用程式伺服器的主機網路名稱，這個應用程式伺服器會提供您想要使用的商業程序應用程式。

*port\_no*

「商業程序編排器瀏覽器」使用的埠號。該埠號視系統配置而定。預設埠號為 9080

*context\_root*

應用程式伺服器上「商業程序編排器瀏覽器」應用程式的根目錄。預設值為 bpc。

2. 如果已啓用安全原則，則必須輸入使用者 ID 及密碼，再按一下**登入**。

畫面上會顯示「商業程序編排器瀏覽器」的起始頁面。根據預設值，這是顯示我的待辦視圖的頁面。

## 自訂商業程序編排器瀏覽器

「商業程序編排器瀏覽器」提供讓管理者管理商業程序及人力作業的使用者介面，以及讓商業使用者使用其指派作業的使用者介面。因為這是一般介面，所以您可能想要自訂特定「商業程序編排器瀏覽器」實例的介面，以處理指派給這個實例的使用者群組的商業需求。

您可以利用各種方法來自訂使用者介面。

### 自訂不同使用者群組的商業程序編排器瀏覽器介面

預設「商業程序編排器瀏覽器」使用者介面中的導覽窗格包含一組預先定義視圖的鏈結。我的待辦視圖是您登入之後顯示的預設視圖。如果您有其中一個「商業程序編排器」系統管理者角色，則可以自訂導覽窗格中顯示的鏈結、您的使用者在登入之後看到的視圖，以及視圖中顯示的資訊。

若要自訂介面，您必須具有 `BPCSystemAdministrator` 權限。

例如，「商業程序編排器瀏覽器」的預設使用者介面不會包括使用商業狀態機器的視圖。您可以新增預先定義的視圖，以使用商業狀態機器的程序範本及程序實例。

或者您可能想為處理客戶訂單的使用者提供不同的介面。該介面跟您提供給處理客戶服務查詢的使用者的介面不一樣。您可以自訂「商業程序編排器瀏覽器」的實例，以使它符合已指派給實例的那些使用者的工作流程型樣。

若要自訂「商業程序編排器瀏覽器」的預設使用者介面，請完成下列步驟。

1. 自訂導覽窗格中的視圖集及預設登入視圖。
  - a. 按一下工作列中的**自訂**。
  - b. 在「自訂導覽樹狀結構及登入視圖」頁面中，選取要併入的視圖，以及取消選取要從導覽窗格移除的視圖。
  - c. 選取您的使用者登入「商業程序編排器瀏覽器」時看到的視圖。

清單包含您在前一個步驟中選取的視圖，以及任何您從「搜尋及定義自訂視圖」頁面中建立的自訂視圖 (請參閱步驟 2)。

- d. 若要儲存變更，請按一下**儲存**。

若要將這個實例的視圖還原為預設視圖，請按一下**還原預設**。這個動作會將導覽窗格重設為預先定義的視圖清單。導覽窗格中的自訂視圖不會受到這個動作影響。

2. 自訂視圖。

您可以指定在這個「商業程序編排器瀏覽器」實例視圖中顯示的資訊。

- a. 按一下工作列中的**定義視圖**。
- b. 在「搜尋及定義自訂視圖」頁面中，選取您想要自訂的視圖類型，例如，程序範本。
- c. 在視圖的「搜尋及定義自訂視圖」頁面中，指定搜尋準則。
- d. 使用「視圖內容」標籤，選取要併入視圖中的內容，以及指定清單內容。



如果這是作業實例視圖或程序實例視圖，請按一下**視圖設定**，以選取一組要新增至視圖中動作列的動作，以及對系統管理者及系統監視者指定要併入視圖中的項目。

- 選取視圖類型：
    - 若要新增管理動作至視圖中的動作列，請選取**管理實例**。
    - 若要將一組可讓已登入使用者使用實例的動作新增至動作列，請選取**使用實例**。
  - 若為系統管理者及系統監視者，您可以將搜尋結果限制為他們自己的實例：
    - 若要在視圖中顯示所有符合搜尋準則的項目，請選取**所有實例**。不管系統管理者是否具有這些項目的工作項目，都會顯示所有項目。
    - 若要只顯示已登入使用者有其工作項目的項目，請選取**個人實例**。
- e. 在**視圖名稱**欄位中輸入視圖的顯示名稱，然後按一下**儲存**。

新的視圖會出現在您的導覽窗格中。使用者會在下次登入「商業程序編排器瀏覽器」時看到新的視圖。

### 定義商業狀態機器的程序範本視圖

雖然已對商業狀態機器的程序範本提供預先定義的視圖，但是您可能想要對此類型的範本定義自己的視圖。

若要建立自訂的視圖，您必須具有 `BPCSystemAdministrator` 權限。

1. 按一下工作列中的**定義視圖**。
2. 在「搜尋及定義視圖」頁面中，選取**搜尋程序範本及定義自訂視圖**。
3. 按一下**內容過濾器** → **自訂內容過濾器**。
  - a. 利用下列設定來新增自訂內容：
    - 在**內容名稱**欄位中，鍵入 `generatedBy`。
    - 在**內容值**欄位中，鍵入 `BusinessStateMachine`。
  - b. 按一下**新增**。
  - c. 依需要新增其他自訂內容。
4. 按一下**視圖內容** → **清單直欄**。
  - a. 在「自訂內容的清單直欄」中，利用下列設定來新增自訂內容：
    - 在**內容名稱**欄位中，鍵入 `generatedBy`。
    - 在**顯示名稱**欄位中，鍵入直欄的顯示名稱，然後按一下**新增**。
  - b. 新增其他直欄至所選取的直欄清單，或從其中移除直欄。
5. 在**視圖名稱**欄位中鍵入查詢的顯示名稱，然後按一下**儲存**。

根據預設值，新視圖的鏈結會新增至導覽窗格中的「程序範本」群組。下次您的使用者登入「商業程序編排器瀏覽器」時，他們會看到這個視圖。

### 定義商業狀態機器的程序實例視圖

雖然已對商業狀態機器的程序實例提供預先定義的視圖，但是您可能想要對此類型的程序實例定義自己的視圖。

若要建立自訂的視圖，您必須具有 `BPCSystemAdministrator` 權限。

1. 按一下工作列中的**定義視圖**。



2. 在「搜尋及定義視圖」頁面中，選取**搜尋程序實例及定義自訂視圖**。
3. 按一下**內容過濾器** → **自訂內容過濾器**。
  - a. 利用下列設定來新增自訂內容：
    - 在**內容名稱**欄位中，鍵入 `generatedBy`。
    - 在**內容值**欄位中，鍵入 `BusinessStateMachine`。
  - b. 按一下**新增**。
  - c. 依需要新增其他自訂內容。
4. 按一下**視圖內容** → **清單直欄**。
  - a. 在「查詢內容的清單直欄」中，新增下列查詢內容。
    - 若要新增商業狀態資訊至視圖，請在**內容名稱**欄位中鍵入 `name`、在**變數名稱**欄位中鍵入 `DisplayState`，並在**名稱空間**欄位中鍵入 `tns`，其中 `tns` 是商業狀態機器的目標名稱空間，字尾為 `-process`。也請在**顯示名稱**欄位中指定直欄的顯示名稱，然後按一下**新增**。
    - 若要新增相關性資訊至視圖，請在**內容名稱**欄位、**變數名稱**欄位及**名稱空間**欄位中提供適當的資訊。這些值衍生自商業狀態機器的定義。也請在**顯示名稱**欄位中提供直欄的顯示名稱。

#### 內容名稱

您針對商業狀態機器定義的相互關係內容的名稱。

#### 變數名稱

如果相互關係集是由送入參數起始，變數名稱具有下列格式：

`operation_name_Input_operation_parameter_name`

其中 `operation_name` 是作業名稱，從起始狀態轉移。

如果相互關係集是由送出參數起始，變數名稱具有下列格式：

`operation_name_Output_operation_parameter_name`

#### 名稱空間

查詢內容的名稱空間，其中 `tns` 是商業狀態機器的目標名稱空間，字尾為 `-process`。


- b. 新增其他自訂內容或查詢內容，或者新增直欄至所選取的直欄清單，或從其中移除直欄。
5. 在**視圖名稱**欄位中鍵入查詢的名稱，然後按一下**儲存**。

根據預設值，新視圖的鏈結會新增至導覽窗格中的「程序實例」群組。下次您的使用者登入「商業程序編排器瀏覽器」時，他們會看到這個視圖。

## 將「商業程序編排器瀏覽器」介面個人化

預設「商業程序編排器瀏覽器」使用者介面中的導覽窗格包含一組預先定義視圖的鏈結，以及針對系統管理者定義的視圖。您可以新增自己的視圖至導覽窗格，例如，監視特定作業或程序的進度。

在「商業程序編排器瀏覽器」中，完成下列步驟以將您的使用者介面個人化。

1. 在您想要定義新視圖的導覽樹狀結構區段中，按一下**新搜尋**圖示 ( )。
2. 在視圖的「搜尋及定義個人化視圖」頁面中，指定搜尋準則。

3. 使用「視圖內容」標籤，選取要併入視圖中的內容，以及指定清單內容。

如果這是作業實例視圖或程序實例視圖，請按一下**視圖設定**，以選取一組要新增至視圖中動作列的動作。如果您是系統管理者及系統監視者，請指定在視圖中併入的項目。

- 選取視圖類型：
    - 若要新增管理動作至視圖中的動作列，請選取**管理實例**。
    - 若要將一組可讓已登入使用者使用實例的動作新增至動作列，請選取**使用實例**。
  - 如果您是系統管理者及系統監視者，可以將搜尋結果限制為自己的實例：
    - 若要在視圖中顯示所有符合搜尋準則的項目，請選取**所有實例**。不管系統管理者是否具有這些項目的工作項目，都會顯示所有項目。
    - 若要只顯示已登入使用者有其工作項目的項目，請選取**個人實例**。
4. 在**視圖名稱**欄位中輸入視圖的顯示名稱，然後按一下**儲存**。

新的視圖會出現在您的導覽窗格中。

## 變更預設 Web 應用程式的外觀

「商業程序編排器瀏覽器」提供一個可以立即使用的 Web 使用者介面，這個使用者介面基礎為 JavaServer Pages (JSP) 檔案及 JavaServer Faces (JSF) 元件。階式樣式表 (CSS) 控制如何呈現 Web 介面。您無需撰寫任何新程式碼，即可修改樣式表，以改造使用者介面使其符合某種外觀和操作方式。

修改樣式表需要深入瞭解階式樣式表。

例如，您可以變更 CSS，以便預設介面符合公司 ID 的準則。

修改樣式表。預設樣式表 `style.css` 包含標頭、導覽窗格及內容窗格中的元素樣式。

### 相關概念

第 273 頁的『商業程序編排器瀏覽器使用者介面』

「商業程序編排器瀏覽器」是獨立式 Web 應用程式，提供一組管理功能，用來管理商業程序及人力作業。該介面由工作列、導覽窗格及工作區組成。

## 商業程序編排器瀏覽器介面中使用的樣式

`style.css` 檔案包含您可變更來修改預設使用者介面的外觀和操作方式的樣式。

`style.css` 檔案包含的樣式適用於預設使用者介面的下列元素：

- 第 282 頁的『橫幅』
- 第 282 頁的『標底』
- 第 282 頁的『功能表列』
- 第 282 頁的『登入頁面』
- 第 282 頁的『導覽器』
- 第 283 頁的『內容畫面』
- 第 283 頁的『指令列』
- 第 283 頁的『清單』
- 第 283 頁的『明細畫面』

- 第 283 頁的『訊息資料』
- 第 284 頁的『標籤窗格』
- 第 284 頁的『搜尋頁面』
- 第 284 頁的『錯誤明細』

此檔案位於下列目錄中：

<profile\_root>\installedApps\<node\_name>\<explorer\_instance>\bpexplorer.war\theme

## 橫幅

樣式名稱	說明
.banner	橫幅的分區。
.banner_left	橫幅中的分區。它是用來嵌入應用程式的標題影像。
.banner_right	橫幅中的分區。例如，您可以用它來顯示其他標誌。

## 標底

樣式名稱	說明
.footer	標底的分區。
.footer_left	標底的分區，例如，您可以用它來顯示應用程式的公司標誌。
.footer_right	標底的分區，例如，您可以用它來顯示其他標誌。

## 功能表列

樣式名稱	說明
.menubar	JSF 子視圖。
.menuContainer	包含功能表項目 (例如，標籤及鏈結) 的儲存器畫面。
.menuItem	功能表列上的項目。

## 登入頁面

樣式名稱	說明
.loginPanel	包含登入表單的畫面。
.loginTitle	表單上的標題。
.loginText	說明性文字。
.loginForm	包含輸入控制項的表單。
.loginValues	決定控制項佈置的表格。
.loginField	用於登入欄位的標籤，例如，「名稱」或「密碼」。
.loginValue	文字輸入欄位。

## 導覽器

樣式名稱	說明
.pageBodyNavigator	包含導覽器的區域。
.navigator	包含清單鏈結的導覽器 JSF 子視圖。

樣式名稱	說明
.navigatorTitle	每個導覽器方塊的標題。
.taskNavigatorTitle	導覽方塊的標題類別。它們用來區別商業程序物件和人力作業物件的清單鏈結。
.navigatorFrame	每個導覽器方塊的分區，例如，繪製邊框。
.navigatorLink	導覽器方塊中的鏈結。
.expanded	在展開導覽器方塊時使用。
.collapsed	在收合導覽器方塊時使用。

## 內容畫面

樣式名稱	說明
.pageBodyContent	包含內容的區域。
.panelContainer	包含清單、明細或訊息的分區畫面。
.panelTitle	已顯示內容的標題，例如，My To-dos。
.panelHelp	包含說明文字及圖示的分區儲存器。
.panelGroup	包含指令列及清單、明細或訊息的分區儲存器。

## 指令列

樣式名稱	說明
.commandbar	指令列區域周圍的分區儲存器。
.button	用於指令列按鈕的樣式。

## 清單

樣式名稱	說明
.list	包含列的表格。
.listHeader	清單標題列中使用的樣式。
.ascending	當清單按此直欄以遞增次序排序時，清單標題類別的樣式。
.descending	當清單按此直欄以遞減次序排序時，清單標題類別的樣式。
.unsorted	當清單未按此直欄排序時，清單標題類別的樣式。

## 明細畫面

樣式名稱	說明
.details	明細畫面周圍的分區儲存器。
.detailsProperty	內容名稱的標籤。
.detailsValue	內容值的文字。

## 訊息資料

樣式名稱	說明
.messageData	訊息周圍的分區儲存器。

樣式名稱	說明
.messageDataButton	訊息表單中「新增」及「移除」按鈕的按鈕樣式。
.messageDataOutput	適用於呈現唯讀文字。
.messageDataValidInput	適用於有效的訊息值。
.messageDataInvalidInput	適用於無效的訊息值。

## 標籤窗格

樣式名稱	說明
.tabbedPane	所有標籤窗格周圍的分區儲存器。
.tabHeader	標籤窗格的標籤標題。
.selectedTab	作用中的標籤標題。
.tab	非作用中的標籤標題。
.tabPane	含括標籤窗格的分區儲存器。
.tabbedPaneNested	包含用於搜尋頁面上的巢狀標籤窗格的分區儲存器。
.tabHeaderSimple	巢狀標籤窗格的標籤標題。
tabHeaderProcess	程序過濾器的巢狀標籤窗格的標籤標題。
.tabHeaderTask	作業過濾器的巢狀標籤窗格的標籤標題。
.tabPaneSimple	含括巢狀標籤窗格的分區儲存器。

## 搜尋頁面

樣式名稱	說明
.searchPane	搜尋畫面的標籤窗格。另請參閱標籤窗格。
.searchPanelFilter	搜尋表單的表格儲存器。
.searchLabel	搜尋表單控制項的標籤。
.summary	含括搜尋摘要窗格的儲存器。
.summaryTitle	在搜尋摘要窗格上所有標題的一般樣式。
.summaryTitleProcess	在搜尋摘要窗格上的程序相關區段的標題樣式。
.summaryTitleTask	在搜尋摘要窗格上的作業相關區段的標題樣式。

## 錯誤明細

樣式名稱	說明
.errorPage	錯誤頁面的標籤窗格。
.errorLink	用來在頁面上呈現按鈕鏈結的樣式。
.errorDetails	具有錯誤明細的標籤窗格。
.errorDetailsStack	具有異常堆疊的標籤窗格。
.errorDetailsMessage	錯誤訊息的文字樣式。

---

## 第 8 章 商業程序編排器觀察程式入門

當商業程序及作業執行時，WebSphere® Process Server 可以發出事件，包含程序實例的狀態變更及其相關活動的相關資訊。您可以使用「商業程序編排器觀察程式」，根據這些事件來擷取統計資訊，並建立關於程序及活動的報告。

您可以定義自己的報告，或使用往下探查方法，以取得特定程序實例、活動實例或行內人力作業的詳細資訊。此外，您也可以匯出報告結果，進行進一步的外部處理。

「商業程序編排器觀察程式」橋接 IT 層次監視與商業層次監視的間隙。透過提供「商業流程管理程式」元件中事件的報告方法，它會協助您瞭解在「商業程序編排器」中發生了什麼情況。

### 相關概念

『商業程序編排器觀察程式使用者介面』

「商業程序編排器觀察程式」是獨立式 Web 應用程式，提供一組產生程序及活動事件報告的功能。該介面由工作列、導覽窗格及工作區組成。

第 286 頁的『商業程序編排器觀察程式導覽窗格』

使用導覽窗格可以選取想要建立的報告類型 (例如程序或活動報告)。您也可以儲存專屬的報告定義，並將這些定義新增至導覽窗格中。

---

### 商業程序編排器觀察程式使用者介面

「商業程序編排器觀察程式」是獨立式 Web 應用程式，提供一組產生程序及活動事件報告的功能。該介面由工作列、導覽窗格及工作區組成。

下圖顯示「商業程序編排器觀察程式」使用者介面的佈置。



使用者介面包含以下主要區域。


### 工作列

工作列提供登出「商業程序編排器觀察程式」的選項，以及一般「說明」頁面的鏈結。

### 導覽窗格

使用者介面左側的導覽窗格包含用來選取要建立的報告類型的鏈結，例如，可以檢視圖表中活動實例的資料。

### 工作區

使用者介面右側的工作區包含用來指定報告定義及檢視報告的頁面。若要存取這些頁面，請按一下導覽窗格中的鏈結。如需頁面的相關資訊，請按一下個別頁面上的說明圖示 。

#### 相關工作

第 285 頁的第 8 章，『商業程序編排器觀察程式入門』

當商業程序及作業執行時，WebSphere® Process Server 可以發出事件，包含程序實例的狀態變更及其相關活動的相關資訊。您可以使用「商業程序編排器觀察程式」，根據這些事件來擷取統計資訊，並建立關於程序及活動的報告。

---

## 商業程序編排器觀察程式導覽窗格

使用導覽窗格可以選取想要建立的報告類型 (例如程序或活動報告)。您也可以儲存專屬的報告定義，並將這些定義新增至導覽窗格中。



## 可用的動作

下列動作可在導覽窗格中使用：

- 收合及展開群組。

按一下導覽窗格中某個項目旁邊的加號 (+)，以展開它，或者按一下減號 (-)，以收合該項目。

- 導覽至預先定義的清單或圖表。

按一下想要報告的實例類型。

- 導覽至程序或活動報告精靈。


按一下**新建報告**圖示 ()，以指定報告類型、報告內容及報告的過濾準則。


- 執行儲存的程序或活動報告。


按一下報告名稱，以執行報告。


- 開啓已儲存程序或活動報告定義的蹦現功能表。


按一下**顯示蹦現功能表**圖示 ()，以使用儲存的報告定義。


- 若要編輯報告定義，請按一下「編輯」圖示 ()。

- 若要複製報告定義，請按一下「複製」圖示 ()。

- 若要刪除報告定義，請按一下「刪除」圖示 ()。

- 若要匯出報告結果，請按一下「匯出」圖示 ()。

- 若要以非同步方式執行搜尋，請按一下**非同步搜尋**圖示 ()。

- 非同步搜尋順利完成之後，導覽窗格中會顯示**非同步搜尋已完成**圖示 ()。按一下報告名稱，以檢視搜尋結果。

- 如果非同步搜尋未順利完成，則會顯示**非同步搜尋失敗**圖示 ()。

## 導覽窗格中的預先定義清單及圖表

下列是導覽窗格包含的預先定義清單及圖表群組。

**清單** 此群組包含下列清單：

### 程序

使用此清單，以檢視在指定的時間範圍期間發出程序事件的程序。會根據程序狀態列出程序。

### 活動

使用此清單，以檢視所選取活動在指定的時間範圍期間達到的狀態。會根據活動狀態列出活動。

### 使用者

使用此清單，以檢視所選取使用者在指定的時間範圍期間執行的活動，以及活動所達到的狀態。會根據其狀態顯示活動。並顯示每個活動的對應使用者。

**警告：** 在收集員工工作效能資料會違反隱私權及資料保護法的國家中，則必須將程序模型定義為不要發出個別使用者的活動事件。

**圖表** 此群組包含下列圖表：

#### 程序 Snapshot

使用此圖表，以檢查在指定的時間內處於不同狀態的程序實例數目。您可以利用長條圖或圓餅圖來檢視資料。

#### 依期間的程序

使用此圖表，以檢查程序實例在指定的期間達到所指定狀態的數目分佈。每一個實例都會顯示在時間片段中，且達到指定狀態。您可以利用折線圖、長條圖或圓餅圖來檢視資料。

#### 活動 Snapshot

使用此圖表，以檢查在指定的時間內處於不同狀態的活動實例數目。您可以利用長條圖或圓餅圖來檢視資料。

#### 依期間的活動

使用此圖表，以檢查活動實例在指定的期間達到所指定狀態的數目分佈。每一個實例都會顯示在時間片段中，且達到指定狀態。您可以利用折線圖、長條圖或圓餅圖來檢視資料。

## 程序及活動報告

導覽窗格會鏈結至下列報告精靈。報告精靈會以**新建報告**圖示指出：。

### 程序報告

使用程序報告，以查詢程序實例事件。這些事件說明程序實例的狀態變更。請使用報告精靈來定義報告的資料。您可以儲存及擷取報告定義。

### 活動報告

使用活動報告，可以查詢活動實例事件。這些事件說明活動實例的狀態變更。請使用報告精靈來指定個別報告。您可以儲存及擷取報告定義。

### 相關工作

第 285 頁的第 8 章, 『商業程序編排器觀察程式入門』

當商業程序及作業執行時，WebSphere® Process Server 可以發出事件，包含程序實例的狀態變更及其相關活動的相關資訊。您可以使用「商業程序編排器觀察程式」，根據這些事件來擷取統計資訊，並建立關於程序及活動的報告。

---

## 第 9 章 管理商業程序及人力作業

商業程序和人力作業已部署和安裝成爲企業應用程式的一部分。您可以使用管理主控台或管理指令來管理程序範本和作業範本，也可以使用「商業程序編排器瀏覽器」來處理程序實例和作業實例。請使用「商業程序編排器觀察程式」來報告商業程序和人力作業。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 33 頁的第 2 章, 『關於人力作業』

人力作業是容許人員及服務互動的元件。

---

### 管理程序範本和程序實例

使用管理主控台或管理指令來管理程序範本。使用「商業程序編排器瀏覽器」來處理程序實例。

程序範本定義企業應用程式內的商業程序。當含有程序範本的企業應用程式已安裝、部署及啓動時，程序範本會進入啓動狀態。您可以使用管理主控台或管理指令來停止和啓動程序範本。「商業程序編排器瀏覽器」中會顯示已啓動的程序範本。

程序實例可以是長時間執行的程序或微流程。請使用「商業程序編排器瀏覽器」來顯示程序範本和程序實例的相關資訊，或者處理程序實例。例如，這些動作可能是啓動程序實例；若爲長時間執行的程序，則可能是其他程序生命週期動作，例如暫停、回復或終止程序實例；或是修復活動。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

### 商業程序管理常見問題

管理商業程序的一組相關常見問題的答案。

- 第 290 頁的『如果程序範本處於已啓動狀態，而其所屬的應用程式處於已停止狀態，會發生什麼狀況？』
- 第 290 頁的『如何停止所建立的新程序實例？』
- 第 290 頁的『當較新的程序範本變成有效時，執行中實例會發生什麼狀況？』
- 第 290 頁的『如果建立執行中實例的範本已停止，該實例會發生什麼狀況？』
- 第 290 頁的『我如何可以識別程序實例是否仍在執行？』
- 第 290 頁的『爲何如果商業程序應用程式包含任何程序實例時，我無法停止它？』

## 如果程序範本處於已啟動狀態，而其所屬的應用程式處於已停止狀態，會發生什麼狀況？

如果目前有效的程序範本處於已啟動狀態，而應用程式處於已停止狀態，則不會從範本建立新程序實例。當應用程式處於已停止狀態時，無法導覽現有的程序實例。

## 如何停止所建立的新程序實例？

使用管理主控台，選取程序範本，然後按一下**停止**。此動作將程序範本置入已停止狀態，且不再從範本建立任何實例。範本停止之後，任何從範本建立程序實例的嘗試都會導致 `EngineProcessModelStoppedException` 錯誤。

## 當較新的程序範本變成有效時，執行中實例會發生什麼狀況？

如果程序範本不再有效，這個事實不會對從範本實例化的執行中實例有任何影響。現有程序實例會繼續執行，直到完成。新舊實例會平行執行，直到所有舊實例都完成，或者直到它們都已終止。

## 如果建立執行中實例的範本已停止，該實例會發生什麼狀況？

將程序範本的狀態變為「已停止」，只會停止所建立的新實例。現有的程序實例會繼續執行，直到依序完成。

## 我如何可以識別程序實例是否仍在執行？

以程序管理者身分登入「商業程序編排器瀏覽器」，並跳至「由我管理程序實例」頁面。此頁面會顯示執行中的程序實例。必要的話，您可以終止並刪除這些程序實例。

## 為何如果商業程序應用程式包含任何程序實例時，我無法停止它？

若要程序實例執行，則其對應的應用程式也必須執行。如果應用程式已停止，則程序實例的導覽無法繼續進行。由於這個原因，您只可以停止商業程序應用程式 (如果它沒有程序實例)。

### 相關工作

第 289 頁的『管理程序範本和程序實例』

使用管理主控台或管理指令來管理程序範本。使用「商業程序編排器瀏覽器」來處理程序實例。

第 291 頁的『使用管理主控台停止及啟動程序範本』

您可以使用管理主控台來個別啟動和停止每一個已安裝的程序範本。

第 292 頁的『使用管理指令停止及啟動程序範本』

除了使用管理主控台之外，管理指令也可以用來停用和啟動程序範本。使用管理指令，停止企業應用程式內的所有程序範本。

第 293 頁的『管理程序生命週期』

在啟動之後，程序會經歷各種狀態，直到結束為止。作為程序管理者，您可以在程序的生命週期中對該程序採取各種動作。

第 293 頁的『啟動新的程序實例』

您可以從任何您有權使用的程序範本啟動新的程序實例。

第 294 頁的『監視程序實例的進度』

您可以監視程序實例的進度，以判定您是否需要採取動作，以便程序可以執行完畢。

#### 第 294 頁的『暫停及回復程序實例』

您可以暫停長時間執行的最上層程序實例。例如，您可能想要這樣做，以便您可以配置稍後程序將使用的後端系統存取權，或修正導致程序實例失敗的問題。而在符合程序的必備項目後，就可以回復執行中的程序實例。

#### 第 295 頁的『終止程序實例』

您可能想要終止程序實例，例如，如果不再需要它所代表的工作或文件、如果沒有人可以完成程序實例、如果您發現程序範本方面的問題，以及它需要重新設計等等。

#### 第 295 頁的『刪除程序實例』

可以建立程序範本的模型，以便在完成程序實例時不會自動加以刪除。您可以在完成這些程序實例之後明確地刪除它們。

#### 第 296 頁的『修復程序及活動』

如果程序發生問題，您可以分析程序，然後修復活動。

#### 第 296 頁的『重新啓動活動』

如果已修復活動，則可以使用新的輸入資料來重新啓動它。

#### 第 297 頁的『強制完成活動』

如果您知道將無法及時完成活動，例如，因為再也無法使用所呼叫的服務，您可以強制完成活動，以便可以繼續執行程序流程。

#### 第 297 頁的『管理微流程的補償服務』

執行微流程時，它可能會發生問題。對於這些狀況，可能已對程序模型中的程序定義補償。補償可讓您復原先前完成的步驟，例如，重設資料及狀態，以便您可以從這些問題回復。

## 使用管理主控台停止及啓動程序範本

您可以使用管理主控台來個別啓動和停止每一個已安裝的程序範本。

如果已啓用 WebSphere 管理安全，請驗證您的使用者 ID 是否具有操作員權限。安裝應用程式的伺服器必須正在執行中。

例如，您必須停止程序範本，才能解除安裝範本所屬的商業程序應用程式。下列步驟說明如何使用管理主控台來停止和啓動程序範本。

1. 選取您要管理的模組。

在管理主控台的導覽窗格中，按一下**應用程式** → **SCA 模組** → **module\_name**。

2. 在 EJB 模組的「配置」頁面的**其他內容**下，按一下**商業程序**，然後按一下程序範本。
3. 停止程序範本。

程序範本的現有實例會繼續執行，直到正常結束為止。不過，您無法依據已停止的範本來建立程序實例。

4. 啓動處於已停止狀態的程序範本。

### 相關概念

#### 第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

#### 第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

## 使用管理指令停止及啓動程序範本

除了使用管理主控台之外，管理指令也可以用來停用和啓動程序範本。使用管理指令，停止企業應用程式內的所有程序範本。

在開始此程序之前，必須符合下列條件：

- 如果已啓用 WebSphere 管理安全，請驗證您的使用者 ID 是否具有操作員權限。
- 將停止或啓動程序範本的應用程式伺服器必須正在執行中。亦即，無法使用 wsadmin 的 `-conntype none` 選項，因為需要伺服器連線。

例如，您必須停止程序範本，才能解除安裝範本所屬的商業程序應用程式。下列步驟說明如何使用管理指令來停止和啓動程序範本。

1. 切換至包含管理 Script 的「商業程序編排器」子目錄。

在 Windows 系統上，請輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 UNIX、Linux 及 i5/OS 系統上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

2. 停止程序範本。

在 Windows 系統上，請輸入：

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

在 UNIX、Linux 及 i5/OS 系統上，輸入：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

其中 *application\_name* 是範本所屬的應用程式的名稱。

程序範本的現有實例會繼續執行，直到正常結束為止。當應用程式停止時，您無法依據已停止的範本來建立程序實例。

3. 啓動程序範本。

在 Windows 系統上，請輸入：

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

在 UNIX、Linux 及 i5/OS 系統上，輸入：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

已啓動程序範本。您可以使用「商業程序編排器瀏覽器」，從程序範本啓動程序實例。

### 相關概念

第 3 頁的第 1 章，『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。



## 管理程序生命週期

在啟動之後，程序會經歷各種狀態，直到結束為止。作為程序管理者，您可以在程序的生命週期中對該程序採取各種動作。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

## 啟動新的程序實例

您可以從任何您有權使用的程序範本啟動新的程序實例。

所有已安裝的程序範本都會顯示在「商業程序編排器瀏覽器」的程序範本清單中。若要啟動新的程序實例，請完成下列步驟。

1. 顯示您有權使用的程序範本。

在導覽窗格的「程序範本」下方按一下**我的程序範本**。

2. 選取程序範本旁的勾選框，然後按一下**啟動實例**。

這個動作會顯示「程序輸入訊息」頁面。

如果程序具有多個作業，則這個動作會顯示一個包含所有可用作業的頁面。選取將啟動程序實例的作業。

3. 提供輸入資料以啟動程序實例。

如果程序是長時間執行的程序，您可以鍵入程序實例名稱。如果未指定名稱，系統產生的名稱會指派給新的程序實例。

完成程序輸入訊息的輸入。

4. 若要啟動程序，請按一下**提交**。

會啟動程序實例。如果商業程序包含需要人機互動的活動，則會產生任何可能的擁有者可以要求的作業。如果您是其中一個可能的擁有者，則這項作業會出現在「我的待辦」頁面上的清單中。

如果程序實例是微流程，則在完成程序之後，會立即顯示程序輸出訊息。若為長時間執行程序，請確保在程序完成之後，不會自動刪除程序實例。在該情況下，程序輸出訊息會顯示在程序實例視圖上。並非所有程序都有輸出訊息，例如，如果程序實作單向作業，則不會顯示輸出訊息。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。



## 監視程序實例的進度

您可以監視程序實例的進度，以判定您是否需要採取動作，以便程序可以執行完畢。

在「商業程序編排器瀏覽器」中，完成下列步驟來監視程序實例的進度。

1. 顯示程序實例的清單。

例如，請在導覽窗格的「程序實例」下方按一下**由我管理**。

2. 選取程序實例旁的勾選框，然後按一下**檢視程序狀態**。

畫面上會顯示「程序狀態」頁面。這個頁面會顯示活動、包括鏈結的轉移及結合條件的鏈結、錯誤處理程式、補償處理程式，以及針對程序定義的事件處理程式。粗體顯示的活動會定義為程序模型中的相關商業。狀態資訊是針對這些活動顯示的。

3. 若要處理活動，請按一下「活動」。

畫面上會顯示「活動」頁面，您可以在這個頁面上採取動作，以便程序可以執行完畢。

### 相關概念

第 3 頁的第 1 章，『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

## 暫停及回復程序實例

您可以暫停長時間執行的最上層程序實例。例如，您可能想要這樣做，以便您可以配置稍後程序將使用的後端系統存取權，或修正導致程序實例失敗的問題。而在符合程序的必備項目後，就可以回復執行中的程序實例。

若要暫停及回復程序實例，您必須具有程序管理者權限。

若要暫停程序實例，程序實例必須處於執行中或失敗中狀態。若要回復程序，程序實例必須處於暫停狀態。

若要暫停或回復程序實例，請在「商業程序編排器瀏覽器」中完成下列步驟。

1. 顯示程序實例的清單。

例如，請在導覽窗格的「程序實例」下方按一下**由我管理**。

2. 暫停程序。

選取程序實例旁的勾選框，然後按一下**暫停**。

3. 選擇其中一個選項以暫停程序實例。

- 若要在手動回復之前暫停程序，請選取**暫停**。
- 若要暫停程序直到某個時間，請選取**暫停程序截止於**並指定日期和時間。
- 若要暫停程序一段時間，請選取**暫停程序適用於**並指定持續時間。

4. 若要確認選擇，請按一下**提交**。

此動作會暫停指定的最上層程序實例。程序實例進入暫停狀態。如果 `autonomy` 屬性設為 `child` 的子程序處於執行中、失敗中、終止中或補償中狀態，則也會一併暫停它們。不過，您仍然可以完成任何屬於程序實例的作用中活動及作業。

若要回復處於暫停狀態的程序實例，請選取程序實例，再按一下**回復**。程序實例及其子程序會進入它們在暫停之前的狀態，例如執行中。程序實例及其子程序便會回復。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

## 終止程序實例

您可能想要終止程序實例，例如，如果不再需要它所代表的工作或文件、如果沒有人可以完成程序實例、如果您發現程序範本方面的問題，以及它需要重新設計等等。

若要終止程序實例，您必須具有程序管理者權限。

在「商業程序編排器瀏覽器」中，完成下列步驟來終止程序實例。如果針對商業程序模型定義補償，則您可以選擇終止具有補償的程序實例。

1. 顯示您可以管理的程序實例。

在導覽窗格的「程序實例」下方按一下**由我管理**。

2. 選取您要停止的程序實例旁的勾選框。

- 若要終止具有補償的程序實例，請按一下**補償**。

這個動作會終止程序實例，並啟動補償處理程序。

- 若要終止沒有補償的程序實例，請按一下**終止**。

這個動作會立即停止程序實例，而不會等待任何未完成的活動或作業。已終止的程序實例不會予以補償。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

## 刪除程序實例

可以建立程序範本的模型，以便在完成程序實例時不會自動加以刪除。您可以在完成這些程序實例之後明確地刪除它們。

若要刪除程序實例，您必須具有程序管理者權限。程序實例必須處於已完成、已失敗、已終止或已補償狀態。

如果對程序模型中的程序範本設定對應內容，則會從「商業程序編排器」資料庫自動刪除已完成的程序實例。

例如，若要從未寫入至審核日誌的程序實例查詢資料，或者如果想要將程序的刪除作業延遲至離峰時間，您可能想要將程序實例保留在您的資料庫中。然而，不再需要的舊程序實例資料會影響磁碟空間及效能。因此，您應該定期刪除不再需要或不再想要維護的程序實例資料。請確定您在離峰時間執行這個維護作業。

您可以使用「商業程序編排器瀏覽器」來刪除已完成的程序實例，例如，刪除個別程序實例，或使用 `deleteCompletedProcessInstances` 管理 Script，以立即刪除數個程序實例。

在「商業程序編排器瀏覽器」中，完成下列步驟來刪除程序實例。

1. 顯示您管理的程序實例。

在導覽窗格的「程序實例」下方按一下**由我管理**。

2. 選取您想要刪除的程序實例，然後按一下**刪除**。

這個動作會從資料庫刪除所選取的程序實例。

#### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

## 修復程序及活動

如果程序發生問題，您可以分析程序，然後修復活動。

「商業程序編排器瀏覽器」提供各種視圖，讓程序管理者監視目前執行中的程序。

- 若要檢視活動處於已停止狀態的程序實例，請在導覽窗格的「程序實例」下方按一下**重要程序**。
- 若要監視特定程序實例的進度，請按一下任何視圖中的**檢視程序狀態**，顯示程序實例的清單。

現在您可以採取動作，以修復擱置中的活動。

#### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動，進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

## 重新啟動活動

如果已修復活動，則可以使用新的輸入資料來重新啟動它。

活動必須處於停止狀態，而相關聯的程序實例則必須處於執行中狀態。

若要重新啟動活動，請在「商業程序編排器瀏覽器」中完成下列步驟。

1. 導覽至活動的「活動」頁面，然後按一下**重新啟動**。

例如，在「由我管理程序實例」頁面上，按一下程序實例的名稱。在「程序實例」頁面上，按一下**活動**標籤，再按一下您想要重新啟動的活動名稱。

2. 指定重新啟動活動所需的輸入資料。

當重新啟動活動時，如果發生錯誤，若要繼續執行程序，請選取**錯誤時繼續**。

3. 如果對活動設定有效期限，請對重新啟動的活動指定期限行爲。
4. 按一下**重新啟動**。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動, 進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

### 強制完成活動

如果您知道將無法及時完成活動, 例如, 因為再也無法使用所呼叫的服務, 您可以強制完成活動, 以便可以繼續執行程序流程。

活動通常必須處於停止狀態。然而, 如果活動是人員活動, 則它也可以處於備妥狀態或要求狀態。相關聯的程序實例必須處於執行中狀態。

若要強制完成活動, 請在「商業程序編排器瀏覽器」中完成下列步驟。

1. 導覽至活動的「活動」頁面, 然後按一下**強制完成**。
2. 指定完成活動所需的資料。
3. 再按一下**強制完成**。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』

商業程序是一組商業相關活動, 進行呼叫就可以達成商業目標。

第 289 頁的『商業程序管理常見問題』

管理商業程序的一組相關常見問題的答案。

### 管理微流程的補償服務

執行微流程時, 它可能會發生問題。對於這些狀況, 可能已對程序模型中的程序定義補償。補償可讓您復原先前完成的步驟, 例如, 重設資料及狀態, 以便您可以從這些問題回復。

對於將補償的微流程, 必須在管理主控台中啟動補償服務。

如果微流程的補償動作失敗, 則程序管理者必須介入以解決問題。

在「商業程序編排器瀏覽器」中, 完成下列步驟以管理失敗的補償動作。

1. 顯示失敗的補償動作清單。

在導覽窗格的「程序實例」下方按一下**失敗的補償**。

畫面上會顯示「失敗的補償」頁面。這個頁面包含指定補償動作失敗原因的相關資訊。這個資訊可協助您判定要採取哪些動作, 來更正失敗的補償服務。

2. 選取活動旁的勾選框, 再按一下其中一個可用的動作。

下列是可用的管理動作：

**略過** 跳過現行補償動作並繼續補償微流程。此動作可能導致非補償活動。

**重試** 如果已採取動作來更正失敗的補償動作, 請按一下**重試**以重新嘗試補償動作。

**停止** 停止補償處理程序。

### 相關概念

第 3 頁的第 1 章, 『關於商業程序』  
商業程序是一組商業相關活動, 進行呼叫就可以達成商業目標。  
第 289 頁的『商業程序管理常見問題』  
管理商業程序的一組相關常見問題的答案。

---

## 管理作業範本及作業實例

使用管理主控台或管理指令, 可以管理作業範本。使用「商業程序編排器瀏覽器」, 可以使用作業實例。

### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

### 相關工作

第 273 頁的第 7 章, 『商業程序編排器瀏覽器入門』  
視您的使用者角色而定, 您可以使用「商業程序編排器瀏覽器」來管理商業程序及人力作業, 或使用您的指派作業。

## 利用管理主控台停止及啓動作業範本

使用管理主控台來個別啓動及停止每一個已安裝的作業範本。

如果已啓用 WebSphere 管理安全, 請驗證您的使用者 ID 是否具有操作員權限。

作業範本定義「服務元件架構 (SCA)」服務, 這些服務在企業應用程式內以獨立式作業表示。當含有作業範本的企業應用程式已安裝、部署及啓動時, 作業範本會進入啓動狀態。

1. 選取您要管理的模組。

在管理主控台的導覽窗格中, 按一下**應用程式** → **SCA 模組** → **module\_name**。

2. 在 EJB 模組的「配置」頁面的**其他內容**下, 按一下**人力作業**, 然後按一下**程序範本**。
3. 若要停止作業範本, 請按一下**停止**。
4. 若要啓動作業範本, 請按一下**啓動**。

### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

## 利用管理指令停止及啓動作業範本

除了使用管理主控台之外, 管理指令也可以用來停用和啓動作業範本。使用管理指令, 停止企業應用程式內的所有作業範本。

如果已啓用 WebSphere 管理安全, 請驗證您是否以具有操作員權限的使用者 ID 登入。

作業範本定義「服務元件架構 (SCA)」服務, 這些服務在企業應用程式內以獨立式作業表示。當含有作業範本的企業應用程式已安裝、部署及啓動時, 作業範本會進入啓動狀態。

1. 切換至包含管理 Script 的「商業程序編排器」子目錄。

在 Windows 系統上，請輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 UNIX、Linux 及 iOS 系統上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

## 2. 停止作業範本。

在 Windows 系統上，請輸入：

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

在 UNIX、Linux 及 iOS 系統上，輸入：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

其中 *application\_name* 是範本所屬的應用程式的名稱。作業範本的現有實例會繼續執行，直到正常結束為止。

## 3. 啓動作業範本。

在 Windows 系統上，請輸入：

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

在 UNIX、Linux 及 iOS 系統上，輸入：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

已啓動作業範本。您可以使用「商業程序編排器瀏覽器」來處理與作業範本相關聯的作業實例。

### 相關概念

第 33 頁的第 2 章，『關於人力作業』

人力作業是容許人員及服務互動的元件。

## 建立及啓動作業實例

您可以從任何您有權使用的作業範本建立及啓動作業實例。

所有已安裝的作業範本都會顯示在「商業程序編排器瀏覽器」的作業範本清單中。若要從作業範本建立及啓動作業實例，請完成下列步驟。

### 1. 顯示您有權使用的作業範本。

在導覽窗格的「作業範本」下方按一下**我的作業範本**。

### 2. 選取作業範本旁的勾選框，然後按一下**啓動實例**。

這個動作會顯示「作業輸入訊息」頁面。

### 3. 提供輸入資料以啓動作業實例。

### 4. 若要啓動作業實例，請按一下**提交**。

作業實例已備妥可以使用。

### 相關概念



第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

#### 相關工作

第 273 頁的第 7 章, 『商業程序編排器瀏覽器入門』  
視您的使用者角色而定, 您可以使用「商業程序編排器瀏覽器」來管理商業程序及人力作業, 或使用您的指派作業。

## 使用您的作業

若要使用作業, 您必須要求作業, 然後執行完成它所需的動作。

如果您是處於備妥狀態作業的可能擁有者或管理者, 則可以要求該作業。如果要求作業, 您會變成該作業的擁有者, 並負責完成它。

您有讀者或編輯者角色的作業也會出現在您的作業清單中。

若要利用「商業程序編排器瀏覽器」來要求及完成作業, 請完成下列步驟。

1. 顯示已指派給您的作業。

按一下**作業實例** → **我的待辦事項**。

這個動作會顯示「我的待辦事項」頁面, 其中列出已指派給您的作業。

2. 要求您想要使用的作業。

選取作業旁的勾選框, 然後按一下**使用**。

這個動作會顯示「作業訊息」頁面。

3. 提供資訊以完成作業。

如果您需要中斷工作, 例如, 因為您需要工作同伴提供更多資訊, 才能完成作業, 請按一下**儲存**, 以儲存您的變更。

4. 按一下**完成**, 以您提供的資訊來完成作業。

您完成的作業處於「完成」狀態。如果您離開作業, 而不完成它, 則作業仍會處於要求狀態。

#### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

#### 相關工作

第 273 頁的第 7 章, 『商業程序編排器瀏覽器入門』  
視您的使用者角色而定, 您可以使用「商業程序編排器瀏覽器」來管理商業程序及人力作業, 或使用您的指派作業。

## 暫停及回復作業實例

您可以暫停作業實例。例如, 若要修正導致作業實例失敗的問題, 您可能想要執行此作業。而在符合作業的必備項目後, 就可以回復執行中的作業實例。

若要暫停及回復作業實例, 您必須具有作業管理者權限。



若要暫停作業實例，作業實例必須處已執行中或失敗中狀態。若要回復作業，作業實例必須處於暫停狀態。

只針對使用 WebSphere Application Server 簡式行事曆的人力作業，支援暫停作業。

若要暫停作業實例，請在「商業程序編排器瀏覽器」中完成下列步驟。

1. 顯示您可以管理的作業實例。

在導覽窗格的「作業實例」下方按一下**由我管理**。

2. 在「作業實例」頁面上，按一下**暫停**。
3. 選擇其中一個選項以暫停作業實例。
  - 若要在手動回復之前暫停作業，請選取**暫停**。
  - 若要暫停作業直到某個時間，請選取**暫停作業截止於**並指定日期和時間。
  - 若要暫停作業一段時間，請選取**暫停作業適用於**並指定持續時間。
4. 若要確認選擇，請按一下**提交**。作業實例會進入已暫停狀態。

若要回復處於暫停狀態的作業實例，請按一下**回復**。

#### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

## 管理人力作業的優先順序

您可以使用人力作業的優先順序，來過濾作業及排序您的作業清單。

若要變更作業實例的優先順序，請在「商業程序編排器瀏覽器」中完成下列步驟。

1. 顯示作業實例的清單。

例如，請在導覽窗格的「作業實例」下方按一下**我的待辦事項**。

2. 選取作業實例旁的勾選框，然後按一下**變更優先順序**。
3. 輸入一個值，然後按一下**提交**。

作業實例的優先順序會設為新值。

若要依優先順序排序作業清單，請按一下表格標頭中的箭頭。

#### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

## 管理指派的工作

在啓動作業之後，您可能需要管理作業所指派的工作，例如，更適當地分配工作群組成員的工作負荷。

工作項目是基於特定原因將商業實體 (例如作業或程序實例) 指派給人員或人員群組。指派原因容許人員在商業程序範例情況中扮演各種角色，例如可能的擁有者、編輯者或管理者。

作業實例可以具有數個與其相關聯的工作項目，因為不同的人員可以具有不同的角色。例如，John、Sarah 及 Mike 全都是作業實例的可能擁有者，而 Anne 是管理者；工作項目是針對所有四種人員產生的。John、Sarah 及 Mike 只會將自己的工作項目看作是其作業清單上的作業。因為 Anne 是管理者，所以她會取得自己對於作業的工作項目，而且她可以管理針對 John、Sarah 及 Mike 產生的工作項目。

有時，您可能需要在啓動作業之後變更所指派的作業，例如，將工作項目從原始擁有者轉移給其他人，或指定您在離開時的缺席設定。您可能也需要建立其他工作項目，或刪除再也不需要的工作項目。

#### 相關概念

第 33 頁的第 2 章，『關於人力作業』  
人力作業是容許人員及服務互動的元件。

#### 相關工作

第 273 頁的第 7 章，『商業程序編排器瀏覽器入門』  
視您的使用者角色而定，您可以使用「商業程序編排器瀏覽器」來管理商業程序及人力作業，或使用您的指派作業。

### 轉移您擁有的作業

如果您是作業的擁有者，則您可能需要將作業轉給另一位使用者，例如，如果其他人需要提供資訊，才能完成作業。

在「商業程序編排器瀏覽器」中，完成下列步驟來轉移您擁有的作業。

1. 顯示您擁有的作業。

在導覽窗格的「作業實例」群組中，按一下**我的待辦**。

2. 選取您要轉移的作業旁的勾選框，然後按一下**轉移**。
3. 轉移作業。

在**新擁有者**欄位中，指定新作業擁有者的使用者 ID，然後按一下**轉移**。您只能將作業轉移給另一位作業的可能擁有者或作業管理者。

轉移的作業會出現在屬於新作業擁有者的作業清單上。

#### 相關概念

第 33 頁的第 2 章，『關於人力作業』  
人力作業是容許人員及服務互動的元件。

#### 相關工作

第 303 頁的『指定缺席設定』  
如果您想要離開辦公室一段時間，請指定作業的代理人。

第 304 頁的『指定使用者的缺席設定』  
如果使用者無法處理作業，例如，如果他們因病休假，請指定使用者作業的代理人。

### 轉移工作項目，您為這些項目的作業啓動者、發送者或管理者

您可能需要在開始作業之後變更工作指派。例如，如果作業擁有者休假中，而且作業必須在這個人回來之前完成，則您可能想要將工作項目轉移給另一位使用者。轉移工作項目的方法視您具有的角色及作業狀態而定。

若要轉移工作項目，您必須具有下列其中一個角色，並且根據指派原因，作業必須處於下列其中一個狀態。

角色	指派原因	作業狀態	工作項目可轉移給下列使用者角色：
擁有者	擁有者	已要求	可能的擁有者、管理者。
起始者	起始者	已過期、已終止、已完成、已失敗或執行中	可能的起始者、管理者。
發送者	發送者	任何作業狀態	可能的實例建立者、管理者。
發送者	可能的起始者	非作用中	任何使用者角色。
管理者	起始者	已過期、已終止、已完成、已失敗或執行中	起始者。
管理者	可能的起始者	非作用中	可能的起始者。
管理者	讀者或管理者	非作用中狀態以外的任何狀態	讀者、管理者。
管理者	可能的擁有者或編輯者	備妥或已要求	可能的擁有者或編輯者。

在「商業程序編排器瀏覽器」中，完成下列步驟來轉移工作項目。

1. 顯示您可以管理的作業實例。

在導覽窗格的「作業實例」下方按一下**由我管理**。

2. 顯示作業實例的工作項目。

在「由我管理的作業實例」頁面中，選取作業實例旁的勾選框，然後按一下**工作項目**。

3. 轉移工作項目。

- a. 在**新擁有者**欄位中，指定新工作項目擁有者的使用者 ID。

- b. 選取一個以上的工作項目，然後按一下**轉移**。

已轉移的工作項目及新的工作項目擁有者會出現在工作項目的清單中。

#### 相關概念

第 33 頁的第 2 章, 『關於人力作業』

人力作業是容許人員及服務互動的元件。

#### 相關工作

第 304 頁的『指定使用者的缺席設定』

如果使用者無法處理作業，例如，如果他們因病休假，請指定使用者作業的代理人。

### 指定缺席設定

如果您想要離開辦公室一段時間，請指定作業的代理人。

若要執行這項作業，需要代理的「虛擬成員管理程式 (VMM)」人員目錄提供者。

視已套用的代理原則而定，在您缺席時有一或多個代理人可接收您的工作指派。每一個作業範本可以有不同的代理原則。

1. 在工作列中，按一下**我的代理人**。
2. 在「我的代理人」頁面上，指定缺席設定並按一下**儲存**。
  - a. 若要啓用您的缺席設定，請選取**我缺席**勾選框。
  - b. 在**我的代理人**欄位中，輸入您的代理人的使用者 ID，然後按一下**新增**。
  - c. 選擇性的：依需要新增進一步的代理人。視已套用的代理原則而定，在您缺席時有一或多個代理人可接收您的工作指派。每一個作業範本可以有不同的代理原則。
  - d. 選擇性的：若要從清單中移除代理人，請選取代理人的使用者 ID，再按一下**移除**。若要選取多位代理人，請按住 **Ctrl** 鍵。
3. 要求您的 **BPESystemAdministrator** 重新整理人員查詢結果。

選取**我缺席**勾選框時，您的代理人將收到您所指派的工作。

在選取**我缺席**勾選框之前，已指派給您的工作指派必須個別轉移。

#### 相關概念

第 74 頁的『缺席者的代理』

代理功能可讓您指定專屬的缺席設定，或您所管理的群組成員的缺席設定。代理原則可定義如何處理指派給缺席使用者的作業及呈報。

第 33 頁的第 2 章，『關於人力作業』

人力作業是容許人員及服務互動的元件。

#### 相關工作

第 302 頁的『轉移您擁有的作業』

如果您是作業的擁有者，則您可能需要將作業轉給另一位使用者，例如，如果其他人需要提供資訊，才能完成作業。

第 169 頁的『配置虛擬成員管理程式人員目錄提供者』

配置「虛擬成員管理程式 (VMM)」人員目錄提供者，讓「商業程序編排器」執行人員指派，決定誰可以啓動程序，或誰可以要求活動或作業。預設人員目錄提供者已備妥可供使用，而且只在您引進自訂的人員指派準則時才需要配置。

第 250 頁的『使用管理主控台來重新整理人員查詢結果』

人員查詢的結果是固定的。請使用管理主控台來重新整理人員查詢。

第 267 頁的『使用管理指令重新整理人員查詢結果』

人員查詢的結果是固定的。使用管理指令來重新整理人員查詢。

## 指定使用者的缺席設定

如果使用者無法處理作業，例如，如果他們因病休假，請指定使用者作業的代理人。

您必須具有 **TaskSystemAdministrator** 權限，才能執行這項作業。若要執行這項作業，需要代理的「虛擬成員管理程式 (VMM)」人員目錄提供者。

1. 在工作列中，按一下**定義代理人**。
2. 在「定義代理人」頁面上，指定缺席設定並按一下**儲存**。
  - a. 輸入您要指定缺席設定的使用者 ID。
  - b. 若要啓用缺席設定，請選取**使用者缺席**勾選框。
  - c. 在**使用者的代理人**欄位中，輸入您要指派的代理人的使用者 ID，然後按一下**新增**。

- d. 選擇性的: 依需要新增進一步的代理人。視已套用的代理原則而定, 在使用者缺席時有一或多個代理人可接收工作指派。每一個作業範本可以有不同的代理原則。
  - e. 選擇性的: 若要從清單中移除代理人, 請選取代理人的使用者 ID, 再按一下**移除**。若要選取多位代理人, 請按住 **Ctrl** 鍵。
3. 重新整理人員查詢結果。

當選取**使用者缺席**勾選框時, 代理人會收到指派給使用者的工作。

在選取**使用者缺席**勾選框之前, 已指派給缺席使用者的工作指派必須個別傳送。

#### **相關概念**

第 74 頁的『缺席者的代理』

代理功能可讓您指定專屬的缺席設定, 或您所管理的群組成員的缺席設定。代理原則可定義如何處理指派給缺席使用者的作業及呈報。

第 33 頁的第 2 章, 『關於人力作業』

人力作業是容許人員及服務互動的元件。

#### **相關工作**

第 302 頁的『轉移工作項目, 您為這些項目的作業啟動者、發送者或管理者』

您可能需要在開始作業之後變更工作指派。例如, 如果作業擁有者休假中, 而且作業必須在這個人回來之前完成, 則您可能想要將工作項目轉移給另一位使用者。轉移工作項目的方法視您具有的角色及作業狀態而定。

第 302 頁的『轉移您擁有的作業』

如果您是作業的擁有者, 則您可能需要將作業轉給另一位使用者, 例如, 如果其他人需要提供資訊, 才能完成作業。

第 169 頁的『配置虛擬成員管理程式人員目錄提供者』

配置「虛擬成員管理程式 (VMM)」人員目錄提供者, 讓「商業程序編排器」執行人員指派, 決定誰可以啟動程序, 或誰可以要求活動或作業。預設人員目錄提供者已備妥可供使用, 而且只在您引進自訂的人員指派準則時才需要配置。

第 250 頁的『使用管理主控台來重新整理人員查詢結果』

人員查詢的結果是固定的。請使用管理主控台來重新整理人員查詢。

第 267 頁的『使用管理指令重新整理人員查詢結果』

人員查詢的結果是固定的。使用管理指令來重新整理人員查詢。

### **建立工作項目**

例如, 當沒有現行可能的擁有者可以接受任何其他工作時, 您可能想要為新的可能擁有者建立工作項目。如果對人員目錄的查詢未傳回任何可能的擁有者, 則您也可能想要建立工作項目。例如, 如果自從啟動程序後組織已變更, 則這種情況可能發生在長時間執行的程序中。

若要建立作業實例的工作項目, 您必須具有該作業的適當角色。如果您是作業管理者, 則在作業實例處於下列其中一個狀態時, 您可以建立它的工作項目: 備妥、要求、執行中、完成或失敗。如果作業實例是從作業範本衍生, 則您也可以在工作處於終止或過期狀態時建立工作項目。

在「商業程序編排器瀏覽器」中, 完成下列步驟以建立工作項目。

1. 顯示您管理的作業實例。

在導覽窗格的「作業實例」下方按一下**由我管理**。

2. 選取您要建立工作項目的作業實例旁的勾選框，然後按一下**建立工作項目**。畫面上會顯示「建立工作項目」頁面。
3. 建立工作項目。
  - a. 在**新擁有者**欄位中，指定新工作項目擁有者的使用者 ID。
  - b. 從**原因**清單中選取一個以上的角色。

這些角色會決定已指派的人員可以對新工作項目執行的動作。

- c. 按一下**建立**。

會為您指定給新工作項目擁有者的每一個角色建立工作項目。新作業會出在已指派給這個人員的作業清單上。

#### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

### 刪除工作項目

例如，如果您建立了錯誤的工作項目，或者如果對不再替公司工作的某人產生工作項目，則您可能想要刪除工作項目。

若要刪除作業實例的工作項目，您必須具有該作業適當的角色。如果您是作業管理者，則在作業實例處於下列其中一個狀態時，您可以刪除工作項目：備妥、已要求、執行中、已完成或失敗。如果作業實例是從作業範本衍生，則您也可以作業實例處於已終止或過期狀態時刪除工作項目。

在「商業程序編排器瀏覽器」中，完成下列步驟來刪除工作項目。

1. 顯示您管理的作業實例。

在導覽窗格的「作業實例」下方按一下**由我管理**。

2. 顯示作業實例的工作項目。

在「由我管理的作業實例」頁面中，選取作業實例並按一下**工作項目**。

3. 刪除工作項目。

選取一個以上的工作項目，然後按一下**刪除**。

會刪除工作項目。

#### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

### 檢視作業呈報

呈報會通知呈報接收者，指出使用者可能發生問題，無法及時完成他們的指派作業。

當作業逾期時，它可能導致呈報。呈報可以導致下列動作：

- 例如，建立新的工作項目，讓管理程式採取動作，以支援解決問題。



- 如果您在配置人力作業儲存器時指定電子郵件設定，則有一封電子郵件會傳送給指定的人員，以通知他們關於呈報的作業。
- 呼叫事件通知處理程式。

若要檢視呈報，請按一下「作業實例」下方的**我的呈報**。

- 若要檢視呈報的相關資訊，請按一下「呈報 ID」。
- 若要檢視所呈報作業的相關資訊，請按一下「作業名稱」。

### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

## 傳送電子郵件以便呈報

當作業逾期時，它可能導致呈報。您可以設定系統，傳送電子郵件至委派的人員，以通知他們關於呈報的資訊。

下列規則適用於呈報電子郵件：

- 您的人員目錄提供者必須支援電子郵件位址的規格，例如「輕量型目錄存取通訊協定 (LDAP)」或虛擬成員管理程式。
- 不支援 **Everybody**、**Nobody**、**Group** 及 **Users by user ID** 人員指派準則。例如，改用 **User records by user ID**。

1. 在 WebSphere Integration Developer 中，以人力作業編輯器對作業執行下列動作。
  - a. 在內容區域的**明細**標籤的作業設定下方，編輯**人員目錄 (JNDI 名稱)** 欄位的值。

將這個欄位的值設為下列其中一項：

- bpe/staff/samplevmconfiguration
- bpe/staff/samplevmconfiguration
- 您選擇的人員目錄配置名稱 (JNDI 名稱)。

- b. 在內容區域的**明細**標籤的呈報設定下方，將**通知類型**欄位的值設為**電子郵件**。
- c. 指定因呈報作業而傳送的電子郵件主體內容。

若要將包括作業特定資訊的變數插入至文字中，請按一下**新增變數**，然後從清單中選取適當的變數。在編輯器中，變數將出現在 "%" 字元之間，但是在傳送電子郵件時，如果在執行期環境中執行期間評估它，將取代它。

如果未指定任何文字，將使用預設訊息文字。

2. 在 WebSphere Process Server 中，執行下列動作。
  - a. 確保已設定簡易郵件傳送通訊協定 (SMTP) 主機。如果已啟用鑑別，請設定 SMTP 主機的「使用者 ID」及密碼。

在管理主控台中，按一下**資源** → **郵件** → **郵件階段作業** → **HTMMailSession\_nodeName\_serverName** 以檢查設定，或按一下**資源** → **郵件** → **郵件階段作業** → **HTMMailSession\_clusterName** (如果已在叢集上配置「商業程序編排器」)。SMTP 主機定義在 Cell 層次上。

- b. 確保您在配置人力作業管理程式時指定的傳送者電子郵件位址 (**傳送者電子郵件位址**) 是有效的電子郵件位址。



在管理主控台中，按一下**伺服器** → **應用程式伺服器** → **server\_name** 以檢查這個設定，或按一下**伺服器** → **叢集** → **cluster\_name** (如果已在叢集上配置「商業程序編排器」)。在**配置**標籤的「商業整合」區段中，按一下**商業程序編排器** → **人力作業管理程式**。

如果呈報電子郵件發生問題，請檢查 SystemOut.log 檔找出錯誤訊息。

#### 相關概念

第 33 頁的第 2 章, 『關於人力作業』  
人力作業是容許人員及服務互動的元件。

---

## 在商業程序編排器瀏覽器建立及編輯自訂內容

建立新的自訂內容，可以指定程序實例、活動實例或作業實例的其他內容。

若要建立實例的自訂內容，請在「商業程序編排器瀏覽器」中完成下列步驟。

1. 顯示程序實例、活動實例或作業實例的清單，然後按一下實例名稱以開啓明細頁面。

例如，若要開啓作業實例的清單，請在導覽窗格的「作業實例」下方按一下**我的待辦**。

2. 在「自訂內容」標籤上，按一下**新增**。
3. 在**內容名稱**欄位中輸入自訂內容的名稱，並在**內容值**欄位中輸入一個值。
4. 選擇性的: 若要新增其他自訂內容，請跳至步驟 2。
5. 選擇性的: 若要移除新的自訂內容，請按一下自訂內容旁的**刪除**圖示。
6. 選擇性的: 若要變更自訂內容的內容名稱或值，請按一下自訂內容並輸入新值。
7. 按一下**儲存**。在儲存自訂內容之後，您就無法變更內容名稱，而且也無法刪除自訂內容。

---

## 報告商業程序及活動

在處理商業程序及活動期間，當程序、活動或作業變更狀態時，即可產生事件。這些事件會加以儲存，並可以利用「商業程序編排器觀察程式」來建立報告，例如，分析程序瓶頸，或評估從活動呼叫服務的可靠性。

您可以使用預先定義的報告，或對程序及活動建立使用者定義的報告。

#### 相關概念

『Snapshot 報告』

使用 Snapshot 報告，可以判定特定日期及時間的活動或程序的狀態。

第 310 頁的『期間報告』

使用期間報告，可以決定在一段時間內，特定活動或程序事件發生的頻率。

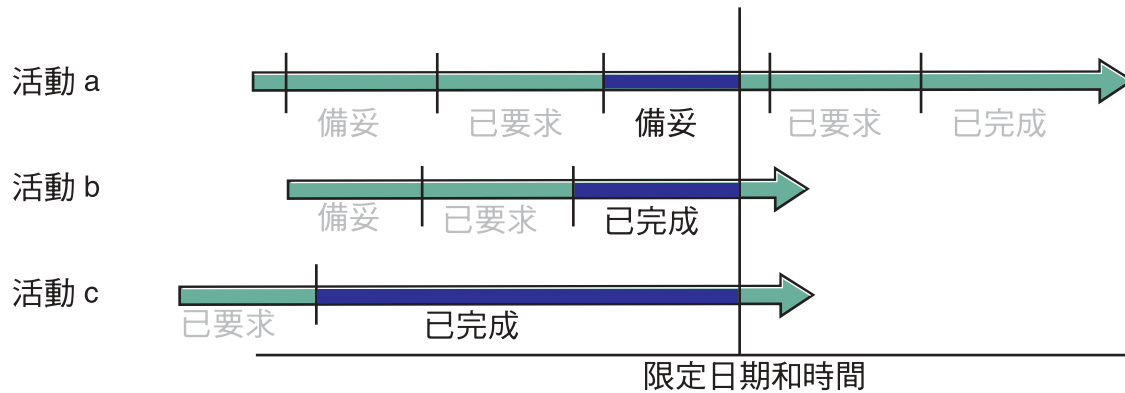
第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

### Snapshot 報告

使用 Snapshot 報告，可以判定特定日期及時間的活動或程序的狀態。

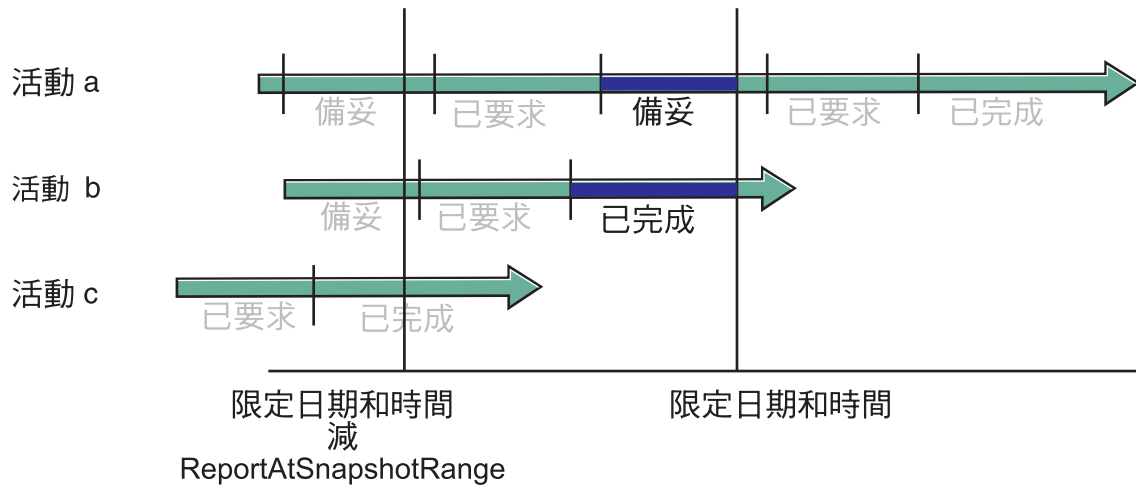
例如，您要瞭解午夜執行的程序實例的數目。對於每個程序或活動實例，「商業程序編排器觀察程式」會尋找指定日期及時間之前的最後一個事件，並評估產生的狀態。以下狀態圖解會顯示事件如何限定 Snapshot 報告。



Snapshot 包括一個處於備妥狀態的活動 (活動 a)，以及兩個處於已完成狀態的活動 (活動 b 及 c)。

### 配置參數 ReportAtSnapshotRange

如果「觀察程式」資料庫包含涵蓋較長一段時間的程序實例資料，則取得 Snapshot 可能會耗用一些時間。若要避免查詢不相關的事件，則使用 ReportAtSnapshotRange 配置參數。報告中只會考量發生在指定時間及日期減去 ReportAtSnapshotRange 配置參數值之後較新的事件。以下狀態圖解顯示事件如何在設定 ReportAtSnapshotRange 參數時限定 Snapshot 報告。



Snapshot 包括一個處於備妥狀態的活動 (活動 a)，以及一個處於已完成狀態的活動 (活動 b)。該報告不會傳回活動 c 的狀態。

### 產生報告週期

您可以定義 Snapshot 報告的產生報告週期。使用這個選項，可以建立包含多個日期的重複 Snapshot 的報告。例如，您要報告三月中每天已啟動程序的數目。您無需每天單獨進行報告。相反地，您可以定義開始日期為 3 月 1 日，開始日期之後的 Snapshot 數目

爲 31，且 Snapshot 之間的時間爲 1 天。產生的報告包含包括時間片段數目的其他直欄。每個時間片段的值表示當月中的日期。

### 相關工作

第 314 頁的『建立預先定義的 Snapshot 圖表』

使用預先定義的 Snapshot 圖表，可以查看程序實例或活動實例在指定的日期和時間的分佈情況。

第 317 頁的『建立使用者定義的 Snapshot 報告』

您可以定義使用者定義的報告，在指定的日期和時間取得狀態資訊的 Snapshot。您也可以建立報告，包含報告期間內定期點的狀態 Snapshot，例如每一個月第一天的午夜。

第 226 頁的『變更商業程序編排器觀察程式的配置參數』

調整「商業程序編排器觀察程式」及事件收集程式應用程式的配置參數對於啓用驗證及改進效能很重要。

第 308 頁的『報告商業程序及活動』

在處理商業程序及活動期間，當程序、活動或作業變更狀態時，即可產生事件。這些事件會加以儲存，並可以利用「商業程序編排器觀察程式」來建立報告，例如，分析程序瓶頸，或評估從活動呼叫服務的可靠性。

第 312 頁的『使用預先定義的清單及圖表』

預先定義的清單及圖表提供往下探查方法，使您取得執行時期實體的狀態及事件資訊。利用往下探查程序中的每一個步驟，進一步定義您有興趣的資訊種類。例如，您可以指定日期及其他過濾準則，以檢視長條圖中活動實例的資料。

第 316 頁的『建立使用者定義的報告』

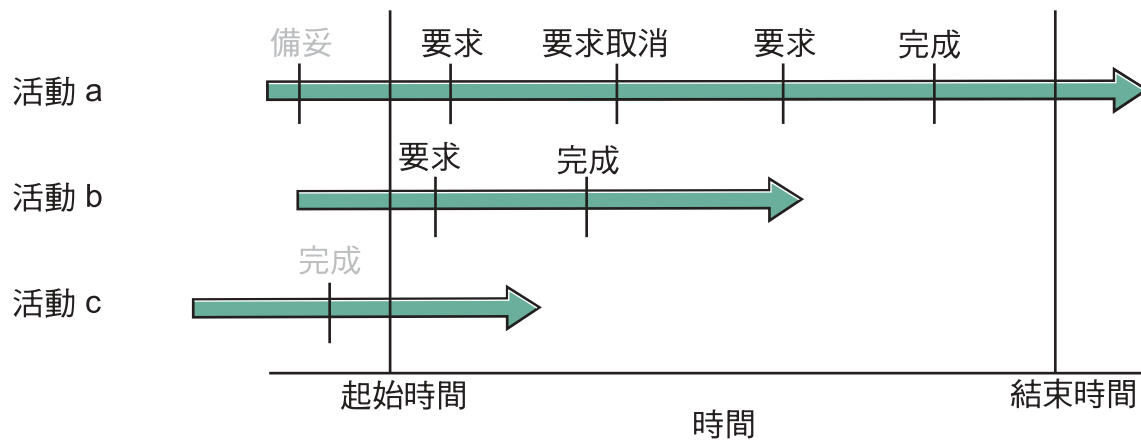
使用者定義的程序及活動報告比預先定義的清單及圖表更有彈性。此外，您也可以儲存並重複使用您的報告定義，而且您可以匯出報告結果。

## 期間報告

使用期間報告，可以決定在一段時間內，特定活動或程序事件發生的頻率。

使用期間視圖，可以指定產生報告期間的起始與結束日期。報告會涵蓋這兩個日期之間的時間。例如，您要瞭解一天中已要求的人員活動數目。

以下狀態圖解會顯示事件如何限定期間報告。涵蓋以下範例中所顯示期間的報告包括六個活動事件；活動 a 包含四個事件，而活動 b 包含兩個事件。活動 c 在產生報告期間的開始之前完成，因此報告中不包含其事件。



這表示如果查詢此期間完成事件的數目，則結果為 2。

### 產生報告週期

您可以定義期間報告的產生報告週期。使用這個選項，可以建立涵蓋多個期間的報告。例如，您要報告最後 12 個月中每個月啟動程序的數目。您無需每個月單獨進行報告。相反地，您可以定義開始日期為 1 月 1 日，開始日期之後的時間片段數目為 12，且時間片段的長度為 1 個月。產生的報告包含包括時間片段數目的其他直欄。每個時間片段的值表示月份。

#### 相關工作

第 314 頁的『建立預先定義的期間圖表』

使用預先定義的期間圖表，可以查看在某個時段達到指定狀態的程序實例或活動實例數目的配送。每一個實例都會顯示在時間片段中，且達到指定狀態。

第 319 頁的『建立使用者定義的期間報告』

您可以針對某個時段發生的程序或活動事件，建立使用者定義的報告。您也可以根據報告週期建立涵蓋多個期間的報告。

第 308 頁的『報告商業程序及活動』

在處理商業程序及活動期間，當程序、活動或作業變更狀態時，即可產生事件。這些事件會加以儲存，並可以利用「商業程序編排器觀察程式」來建立報告，例如，分析程序瓶頸，或評估從活動呼叫服務的可靠性。

第 312 頁的『使用預先定義的清單及圖表』

預先定義的清單及圖表提供往下探查方法，使您取得執行時期實體的狀態及事件資訊。利用往下探查程序中的每一個步驟，進一步定義您有興趣的資訊種類。例如，您可以指定日期及其他過濾準則，以檢視長條圖中活動實例的資料。

第 316 頁的『建立使用者定義的報告』

使用者定義的程序及活動報告比預先定義的清單及圖表更有彈性。此外，您也可以儲存並重複使用您的報告定義，而且您可以匯出報告結果。

## 時間處理

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

### 時間戳記

在資料庫中，時間戳記以世界標準時間 (UTC) 格式儲存。輸入及顯示的時間戳記一律以執行使用者介面所在位置的當地時間表示。這表示如果您指定某個產生報告週期的 Snap-

shot 報告，且產生報告週期跨時調整，以適用於日光節約時間，則在變更時鐘之後，日期和時間的變化約在一小時內。

例如，如果您指定產生報告週期的 Snapshot 報告，並在冬季於上午 8:00 進行第一次 Snapshot，接下來每 24 小時進行一次 Snapshot，則 Snapshot 在日光節約時間期間在上午 9:00 進行。

## 持續時間 (月及年)

如果您指定產生報告週期的報告，例如，您以月或年為單位提供時間片段長度，則每個個別時間片段的長度會根據行事曆進行改變。這可讓您指定每個時間片段代表一年中某月的報告。

### 相關工作

第 308 頁的『報告商業程序及活動』

在處理商業程序及活動期間，當程序、活動或作業變更狀態時，即可產生事件。這些事件會加以儲存，並可以利用「商業程序編排器觀察程式」來建立報告，例如，分析程序瓶頸，或評估從活動呼叫服務的可靠性。

『使用預先定義的清單及圖表』

預先定義的清單及圖表提供往下探查方法，使您取得執行時期實體的狀態及事件資訊。利用往下探查程序中的每一個步驟，進一步定義您有興趣的資訊種類。例如，您可以指定日期及其他過濾準則，以檢視長條圖中活動實例的資料。

第 316 頁的『建立使用者定義的報告』

使用者定義的程序及活動報告比預先定義的清單及圖表更有彈性。此外，您也可以儲存並重複使用您的報告定義，而且您可以匯出報告結果。

## 使用預先定義的清單及圖表

預先定義的清單及圖表提供往下探查方法，使您取得執行時期實體的狀態及事件資訊。利用往下探查程序中的每一個步驟，進一步定義您有興趣的資訊種類。例如，您可以指定日期及其他過濾準則，以檢視長條圖中活動實例的資料。

下列是可用的預先定義清單及圖表類型：

- 清單
- 程序及活動 Snapshot 圖表
- 依期間的程序及活動實例圖表

### 相關概念

第 308 頁的『Snapshot 報告』

使用 Snapshot 報告，可以判定特定日期及時間的活動或程序的狀態。

第 310 頁的『期間報告』

使用期間報告，可以決定在一段時間內，特定活動或程序事件發生的頻率。

第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

### 相關工作

第 315 頁的『範例：使用預先定義的圖表』

這個範例實務會示範如何使用預先定義的圖表。

『範例：使用預先定義的清單』

這個範例實務會示範如何使用預先定義清單。

## 使用預先定義的清單建立報告

您可以使用預先定義的清單，報告在指定的時段內發生的程序或活動事件數目，依狀態排序。您也可以使用清單，向下探查至特定實例的事件。此外，您也可以匯出每一個狀態的報告結果。

### 1. 選取清單類型。

程序實例、活動實例及與使用者相關聯的活動可以使用預先定義的清單。

### 2. 輸入您有興趣的時段的開始日期及結束日期，然後按一下**繼續**。

根據清單類型，會顯示程序範本清單、活動範本，或使用者清單及其相關聯的實例數。

### 3. 選取您感興趣實例的勾選框，然後按一下**實例 Snapshot**。

所選取實例的事件會顯示在標籤窗格中。每一個頁面都會顯示處於特定狀態中的實例。

### 4. 選擇性的: 若要查看所有事件，以及如需特定實例的相關資訊，請按一下實例名稱。

### 5. 選擇性的: 若要以 CSV 格式匯出報告的資料，請按一下**匯出**。選取您想要開啓已產生的匯出資料，還是儲存它，然後按一下**確定**。此時會匯出目前顯示狀態的報告資料。

## 相關概念

第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

## 相關工作

『範例：使用預先定義的清單』

這個範例實務會示範如何使用預先定義清單。

## 範例：使用預先定義的清單

這個範例實務會示範如何使用預先定義清單。

您的工廠生產不同的項目 Item1、Item2 及 Item3。您的製造及出貨程序已建立模型，並利用 WebSphere Process Server 以 SOA 程序執行。每一個客戶訂單是以適當程序範本的專用程序實例來代表。在項目出貨給客戶之後，您的出貨程序為結束狀態「完成」。如果客戶取消訂單，則對應程序實例會終止並達到已終止狀態。

若要查看上個月有多少客戶取消 Item1、Item2 或 Item3 的訂單，您可以查看處於已終止狀態的程序實例數目。此外，您還想要知道當取消時，訂單的處理進度。

使用預先定義的清單，建立一個視圖來顯示已取消多少個程序，以及取消時，查看程序所處的狀態：

### 1. 在導覽列的清單區段中，選取**程序**。

### 2. 在「搜尋準則」頁面上，輸入您有興趣的時段的開始日期及結束日期，然後按一下**繼續**。「程序範本」頁面會列出所有在觀察期間產生程序的程序範本。對於每一個程序範本，您可以查看「啓動」及「結束」的程序實例數。



3. 在「程序範本」頁面上，選取消單的所有範本，然後按一下**實例 Snapshot**。「程序實例」頁面會列出所有程序實例，依其在觀察期間它們達到的狀態分組。
4. 在「程序實例」頁面上，選取消**終止**標籤，以查看觀察期間的取消總數。
5. 依範本名稱來排序清單，然後評估每個程序範本的取消數。
6. 如需進一步明細，請按一下「終止」的程序實例名稱，以檢視「程序實例明細」頁面。檢查實例的運作時間及經歷時間。

#### 相關工作

第 312 頁的『使用預先定義的清單及圖表』

預先定義的清單及圖表提供往下探查方法，使您取得執行時期實體的狀態及事件資訊。利用往下探查程序中的每一個步驟，進一步定義您有興趣的資訊種類。例如，您可以指定日期及其他過濾準則，以檢視長條圖中活動實例的資料。

第 313 頁的『使用預先定義的清單建立報告』

您可以使用預先定義的清單，報告在指定的時段內發生的程序或活動事件數目，依狀態排序。您也可以使用清單，向下探查至特定實例的事件。此外，您也可以匯出每一個狀態的報告結果。

### 建立預先定義的 Snapshot 圖表

使用預先定義的 Snapshot 圖表，可以查看程序實例或活動實例在指定的日期和時間的分佈情況。

您的瀏覽器必須啓用，才能執行 Macromedia Flash Player 來檢視圖表。

1. 選取消 Snapshot 的類型。

程序實例及活動實例可以使用預先定義的 Snapshot 圖表。

2. 輸入搜尋準則並按一下**繼續**。

畫面上會顯示符合搜尋準則的物件範本清單。

3. 選取消您感興趣的範本的勾選框，然後按一下**繼續已選取的**。

您可以變更圖表類型，將結果顯示為長條圖或圓餅圖。

#### 相關概念

第 308 頁的『Snapshot 報告』

使用 Snapshot 報告，可以判定特定日期及時間的活動或程序的狀態。

第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

#### 相關工作

第 315 頁的『範例：使用預先定義的圖表』

這個範例實務會示範如何使用預先定義的圖表。

### 建立預先定義的期間圖表

使用預先定義的期間圖表，可以查看在某個時段達到指定狀態的程序實例或活動實例數目的配送。每一個實例都會顯示在時間片段中，且達到指定狀態。

您的瀏覽器必須啓用，才能執行 Macromedia Flash Player 來檢視圖表。

例如，您可以使用預先定義的圖表，查看已完成的程序實例在過去 12 個月的分佈情況。



1. 選取期間圖表的類型。

程序實例及活動實例可以使用預先定義的期間圖表。

2. 輸入搜尋準則並按一下**繼續**。

輸入時段的開始日期，並指定時間片段數、每一個時間片段的長度，以及您正要報告的狀態。例如，若要報告過去 12 個月每一個月完成的實例，請指定 12 作為時間片段數，並指定 1 個月作為每一個時間片段的長度。

畫面上會顯示符合搜尋準則的物件範本清單。

3. 選取您感興趣的範本的勾選框，然後按一下**繼續已選取的**。

您可以變更圖表類型，將結果顯示為長條圖、折線圖或圓餅圖。

#### 相關概念

第 310 頁的『期間報告』

使用期間報告，可以決定在一段時間內，特定活動或程序事件發生的頻率。

第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

#### 相關工作

『範例：使用預先定義的圖表』

這個範例實務會示範如何使用預先定義的圖表。

### 範例：使用預先定義的圖表

這個範例實務會示範如何使用預先定義的圖表。

您的工廠生產不同的項目 Item1 及 Item2。您的製造及出貨程序已建立模型，並利用 WebSphere Process Server 以 SOA 程序執行。每一個客戶訂單是以適當程序範本的專用程序實例來代表。

最近您已擴充 Item3 的生產線。您有新的 Item3 訂購範本，而且您想要知道您的生產線在上個月的進度。作為一個指示器，您想要看到過去 30 天的生產訂單數目。

若要以圖表的方式呈現過去 30 天所處理的生產訂單數，請指定一個圖表視圖，顯示在有興趣的期間與 OrderItem3 程序範本相關的所有程序實例：

1. 在導覽列的**圖表**區段中，選取「依期間的程序」，查看程序實例在過去 30 天的統計分佈情況。
2. 指定搜尋準則：
  - a. 輸入觀察期間的開始日期
  - b. 將時間片段數設為 30。
  - c. 將時間片段的長度設為一天。
  - d. 在**專注於狀態**清單中，選取**執行中**，然後按一下**繼續**。

此時會開啓「選取程序範本」頁面，其中包含一個清單，列出與觀察期間發生的程序實例相關的所有程序範本。

3. 選取 OrderItem3 範本，查看與這個程序範本相關的所有程序實例，然後按一下**繼續已選取的**。

4. 「程序實例 Snapshot」頁面會顯示所有在指定的時間內，處於不同狀態的程序實例。
5. 使用折線圖或長條圖，以圖表的方式呈現程序在上個月的進度

您的報告會顯示所有在觀察期間到達執行中狀態的程序實例。

#### **相關工作**

第 312 頁的『使用預先定義的清單及圖表』

預先定義的清單及圖表提供往下探查方法，使您取得執行時期實體的狀態及事件資訊。利用往下探查程序中的每一個步驟，進一步定義您有興趣的資訊種類。例如，您可以指定日期及其他過濾準則，以檢視長條圖中活動實例的資料。

第 314 頁的『建立預先定義的 Snapshot 圖表』

使用預先定義的 Snapshot 圖表，可以查看程序實例或活動實例在指定的日期和時間的分佈情況。

第 314 頁的『建立預先定義的期間圖表』

使用預先定義的期間圖表，可以查看在某個時段達到指定狀態的程序實例或活動實例數目的配送。每一個實例都會顯示在時間片段中，且達到指定狀態。

## **建立使用者定義的報告**

使用者定義的程序及活動報告比預先定義的清單及圖表更有彈性。此外，您也可以儲存並重複使用您的報告定義，而且您可以匯出報告結果。

若為程序報告，您可以取得程序實例的屬性及程序實例所屬活動的相關資訊。若為活動報告，您可以取得活動屬性，以及活動的相關聯程序實例的相關資訊。您可以定義一次性報告，或儲存您的報告定義，以便您可在需要時執行它。包括參數，以在每次執行報告時，變更報告定義的值。

#### **相關概念**

第 308 頁的『Snapshot 報告』

使用 Snapshot 報告，可以判定特定日期及時間的活動或程序的狀態。

第 310 頁的『期間報告』

使用期間報告，可以決定在一段時間內，特定活動或程序事件發生的頻率。

第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

#### **相關工作**

第 321 頁的『範例：使用使用者定義的報告』

這個實務範例會示範如何使用使用者定義報告。

#### **相關參考**

第 322 頁的『報告屬性』

使用屬性，可以定義報告的內容，並過濾結果。可用的屬性視報告類型而定。

第 324 頁的『商業程序編排器觀察程式的商業程序事件』

如果要求監視 WebSphere Integration Developer 中的商業程序元素，則會傳送商業程序事件。這些事件的子集可用於「商業程序編排器觀察程式」。


第 325 頁的『效能相關屬性』

執行報告定義所需的時間可以不同。若要改進產生報告的效能，您可以最佳化報告定義。部分一般規格可協助您評估報告屬性對效能的影響。

## 建立使用者定義的 Snapshot 報告

您可以定義使用者定義的報告，在指定的日期和時間取得狀態資訊的 Snapshot。您也可以建立報告，包含報告期間內定期點的狀態 Snapshot，例如每一個月第一天的午夜。

報告精靈會引導您定義報告。

1. 在導覽窗格中，針對程序報告或活動報告，按一下**新建報告**圖示 ( )。
2. 在「選取報告類型」頁面上，按一下 **Snapshot 報告**並按**下一步**。
3. 在「選取 Snapshot 類型」頁面上，指定您想要何時取得 Snapshot，再按**下一步**。
  - 若要查看現行狀態，請按一下**立即取得 Snapshot**。每次執行報告，即會評估 Snapshot 日期和時間。

畫面上會顯示「指定內容」頁面。繼續執行步驟 5。

- 若要查看特定日期和時間的程序或活動狀態，例如 6 月 10 日上午 8:00，請按一下在**特定日期和時間取得 Snapshot**。

畫面上會顯示「指定 Snapshot 設定」頁面。繼續執行步驟 4。

- 若要查看報告期間內定期點的狀態，請按一下**根據報告週期取得重複的 Snapshot**。

畫面上會顯示「指定 Snapshot 設定」頁面。繼續執行步驟 4。

4. 指定 Snapshot 設定，再按**下一步**。


如果將在特定日期和時間取得 Snapshot，請指定日期和時間設定。您可以指定未來的日期和時間。若要在每次執行報告時變更設定，請選取**使用這些設定作為參數**勾選框。


對於具有報告週期的報告：

- a. 選取您是要設定報告週期的開始日期還是結束日期，再按**下一步**。
  - b. 若要設定報告週期的開始日期，請指定何時將取得第一個 Snapshot。若要設定報告週期的結束日期，請指定何時將取得最後一個 Snapshot。
  - c. 若要定義報告週期的持續時間，請設定 Snapshot 數及每一個 Snapshot 之間的時間。
  - d. 若要在每次執行報告時變更報告週期的設定，請選取**使用這些設定作為參數**勾選框。
5. 在「指定報告內容」頁面上，指定您想要報告包含的資訊，再按**下一步**。

若為具有報告週期的報告，屬性清單已包含 Snapshot 號碼屬性。您無法刪除這個屬性。

- a. 按一下**新增**，以查看您可以併入在報告中的屬性清單；這些屬性會變成報告的直欄標題。屬性的位置會決定報告中直欄的次序。對於每一個屬性，您也可以指定如何在直欄內排序結果。如果您對多個屬性指定排序次序，則會依屬性次序來排序結果。請考量重新排列屬性次序，以變更報告中結果的排序次序。

- 若要修改屬性，請按一下**編輯**圖示 ( )。

- 若要刪除屬性，請按一下**刪除**圖示 ( )。

- 若要變更報告中屬性的位置，請按一下上圖示 (↑) 或下圖示 (↓)。
- b. 例如，若要基於效能原因而限制結果中的項目數，請在**臨界值**欄位中輸入一值，以指定結果數上限。

預設臨界值為 20。如果不需要限制結果，請將值設為 -1。

若要在每次執行報告時變更臨界值，請選取**使用臨界值作為參數**勾選框。

6. 選擇性的: 在「指定過濾器內容」頁面上，設定屬性的過濾準則。

使用過濾準則，限制屬性可以採用的值，因此可使您的報告更明確。報告只包括那些符合所有指定過濾準則的程序及活動。如果已在「指定報告內容」頁面上指定作為聚集的屬性，則過濾準則清單已包含這個屬性的過濾準則。您無法刪除這個過濾器。

- a. 按一下**新增**，以查看您可以指定過濾準則的屬性清單。

- 若為更複雜的值類型，例如時間戳記，請按一下**輸入說明程式**圖示 (💡) 以完成欄位。
- 若要在每次執行報告時變更過濾準則的值，請選取**參數**勾選框。

- b. 按**下一步**。

畫面上會顯示「摘要」頁面。這個頁面會顯示報告定義。

7. 在「摘要」頁面上，執行下列其中一項：

- 如果您的報告定義沒有包含參數，請按一下**執行**。

畫面上會顯示結果報告。

- 如果您的報告定義包含參數，請按**下一步**。

您可以變更參數值，然後按一下**執行**。畫面上會顯示結果報告。

如果報告結果不是您預期的，則您可以按一下**編輯**，以變更報告的設定。

8. 選擇性的: 匯出報告結果。

若要以 CSV 格式匯出報告的資料，請按一下**匯出**。選取您想要開啓已產生的匯出資料，還是將資料儲存在硬碟上，再按一下**確定**。

只在報告清單包含項目時，才會顯示**匯出**按鈕。

9. 選擇性的: 儲存報告定義。

如果這是您想要執行多次的報告，例如顯示每月第 10 天完成的程序實例的每月報告，請按一下**儲存**並輸入報告名稱。報告會出現在導覽窗格中。

### 相關概念

第 308 頁的『Snapshot 報告』

使用 Snapshot 報告，可以判定特定日期及時間的活動或程序的狀態。

第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

### 相關工作

第 321 頁的『範例：使用使用者定義的報告』  
這個實務範例會示範如何使用使用者定義報告。

### 相關參考

第 322 頁的『報告屬性』

使用屬性，可以定義報告的內容，並過濾結果。可用的屬性視報告類型而定。

第 324 頁的『商業程序編排器觀察程式的商業程序事件』

如果要求監視 WebSphere Integration Developer 中的商業程序元素，則會傳送商業程序事件。這些事件的子集可用於「商業程序編排器觀察程式」。


第 325 頁的『效能相關屬性』

執行報告定義所需的時間可以不同。若要改進產生報告的效能，您可以最佳化報告定義。部分一般規格可協助您評估報告屬性對效能的影響。

## 建立使用者定義的期間報告

您可以針對某個時段發生的程序或活動事件，建立使用者定義的報告。您也可以根據報告週期建立涵蓋多個期間的報告。

報告精靈會引導您定義報告。

1. 在導覽窗格中，針對程序報告或活動報告，按一下**新建報告**圖示 ( )。
2. 在「選取報告類型」頁面上，按一下**期間報告**並按**下一步**。
3. 在「選取期間類型」頁面上，指定期間類型，再按**下一步**。

例如，若為程序，您可以選取下列其中一個期間類型：

- 若要查看從指定的日期至今的事件，請按一下**報告到目前為止的所有程序**。
- 若要查看指定期間的事件，請按一下**報告指定期間內的程序**。
- 若要在報告期間查看定期的事件，請按一下**依據報告週期報告程序**。

畫面上會顯示「指定日期和時間」頁面。

4. 指定日期和時間設定，再按**下一步**。


如需報告到目前為止的所有程序，請指定開始日期。每次執行報告，都會產生結束日期。如需報告指定期間中的程序，請指定開始日期及結束日期。您可以指定未來的日期。若要在每次執行報告時變更設定，請選取**使用這些設定作為參數**勾選框。


對於具有報告週期的報告：



- a. 選取您想要設定報告週期的開始日期還是結束日期，再按**下一步**。
  - b. 若要設定報告週期的開始日期，請指定第一個時間片段的開始日期。若要設定報告週期的結束日期，請指定最後一個時間片段的結束日期。
  - c. 若要定義報告週期的持續時間，請設定時間片段總數及每一個時間片段的長度。
  - d. 若要在每次執行報告時變更報告週期的設定，請選取**使用這些設定作為參數**勾選框。
5. 在「指定報告內容」頁面上，指定您想要報告包含的資訊，再按**下一步**。

若為具有報告週期的報告，屬性清單已包含時間片段號碼屬性。您無法刪除這個屬性。

- a. 按一下**新增**，以查看您可以併入在報告中的屬性清單；這些屬性會變成報告的直欄標題。屬性的位置會決定報告中直欄的次序。對於每一個屬性，您也可以指定如何在直欄內排序結果。如果您對多個屬性指定排序次序，則會依屬性次序來排序結果。請考量重新排列屬性次序，以變更報告中結果的排序次序。

- 若要修改屬性，請按一下**編輯**圖示 (  )。

- 若要刪除屬性，請按一下**刪除**圖示 (  )。

- 若要變更報告中屬性的位置，請按一下**上**圖示 (  ) 或**下**圖示 (  )。

- b. 例如，若要基於效能原因而限制結果中的項目數，請在**臨界值**欄位中輸入一值，以指定結果數上限。


預設臨界值為 20。如果不需要限制結果，請將值設為 -1。

若要在每次執行報告時變更臨界值，請選取**使用臨界值作為參數**勾選框。

6. 選擇性的: 在「指定過濾器內容」頁面上，設定屬性的過濾準則。

使用過濾準則，限制屬性可以採取的值，因此可使您的報告更明確。如果已在「指定報告內容」頁面上指定作為聚集的屬性，則過濾準則清單已包含這個屬性的過濾準則。您無法刪除這個過濾器。

- a. 按一下**新增**，以查看您可以指定過濾準則的屬性清單。

- 若為更複雜的值類型，例如時間戳記，請按一下**輸入說明程式**圖示 (  ) 以完成欄位。

- 若要在每次執行報告時變更過濾準則的值，請選取**參數**勾選框。

- b. 按**下一步**。

畫面上會顯示「摘要」頁面。這個頁面會顯示報告定義。

7. 在「摘要」頁面上，執行下列其中一項：

- 如果您的報告定義沒有包含參數，請按一下**執行**。

畫面上會顯示結果報告。

- 如果您的報告定義包含參數，請按**下一步**。

您可以變更參數值，然後按一下**執行**。畫面上會顯示結果報告。

如果報告結果不是您預期的，則您可以按一下**編輯**，以變更報告的設定。

8. 選擇性的: 匯出報告結果。

若要以 CSV 格式匯出報告的資料，請按一下**匯出**。選取您想要開啓已產生的匯出資料，還是將資料儲存在硬碟上，再按一下**確定**。

只在報告清單包含項目時，才會顯示**匯出**按鈕。

9. 選擇性的: 儲存報告定義。

如果這是您定期執行的報告，例如每月報告，請按一下**儲存**並輸入報告名稱。報告會出現在導覽窗格中。

### 相關概念



第 310 頁的『期間報告』

使用期間報告，可以決定在一段時間內，特定活動或程序事件發生的頻率。

第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

### 相關工作

『範例：使用使用者定義的報告』

這個實務範例會示範如何使用使用者定義報告。

### 相關參考

第 322 頁的『報告屬性』

使用屬性，可以定義報告的內容，並過濾結果。可用的屬性視報告類型而定。

第 324 頁的『商業程序編排器觀察程式的商業程序事件』

如果要求監視 WebSphere Integration Developer 中的商業程序元素，則會傳送商業程序事件。這些事件的子集可用於「商業程序編排器觀察程式」。

第 325 頁的『效能相關屬性』

執行報告定義所需的時間可以不同。若要改進產生報告的效能，您可以最佳化報告定義。部分一般規格可協助您評估報告屬性對效能的影響。

## 範例：使用使用者定義的報告

這個實務範例會示範如何使用使用者定義報告。

您的工廠生產不同的項目 Item1、Item2 及 Item3。您的製造及出貨程序已建立模型，並利用 WebSphere Process Server 以 SOA 程序執行。每一個客戶訂單是以適當程序範本的專用程序實例來代表。在項目出貨給客戶之後，您的出貨程序會達到結束狀態「已完成」。如果客戶取消訂單，則對應程序實例會終止並達到已終止狀態。

其中一個取消訂單的客戶抱怨回應時間過長。您想要知道為什麼需要這麼長的時間來處理這個訂單。

針對處於已終止狀態的程序實例及工作時間超過兩天的程序實例，建立使用者定義的報告。此外，您的報告也應該顯示已終止的程序實例出了什麼錯。

### 1. 擷取屬於客戶訂單的程序實例資料。

客戶名稱、地址及訂單號碼都是商業資料的一部分，因此，都包含在程序訊息中。不過，「商業程序編排器觀察程式」無法使用商業物件的內容，因為它不是「共用事件基礎架構 CEI」事件的一部分。不過，您知道您正在尋找處於已終止狀態，而且工作時間超過兩天的程序實例。

- a. 在導覽列的**程序報告**區段中，選取**建立新報告**。
- b. 因為您專注於程序實例的狀態，所以請選取報告類型 **Snapshot 報告**。
- c. 在「選取 Snapshot 類型」頁面上，選取在**特定日期和時間**取得 **Snapshot**。在取消訂單之後，立即指定日期和時間，限定 Snapshot 日期。
- d. 在「報告內容」頁面上，將**程序實例 ID**、**程序運作時間**、**已啟動的程序**及**已完成的程序**新增至您的報告內容。
- e. 在「過濾器內容」頁面上，指定大於 **2 天**的**程序運作時間**及**程序狀態**等於**已終止**作為過濾器內容，然後執行報告。

- f. 在「報告結果」頁面上，檢查程序實例 ID、開始日期及完成日期，以尋找對應於客戶訂單的程序實例。如果報告結果不符合您的期望，例如，如果程序實例的清單太長，請按一下**編輯**，以修改您的搜尋準則。
  - g. 將程序實例 ID 複製至剪貼簿，因為您在步驟 2 中將需要這個 ID。
2. 取得特定程序實例出錯時所顯示的資訊。
    - a. 在導覽列的**程序報告**區段中，選取**建立新報告**。
    - b. 選取報告類型 **Snapshot 報告**。  
 請勿使用「期間報告」類型。您對與 Snapshot 報告相關的屬性有興趣。若要查看差異，請定義並執行屬性完全相同的期間報告。
    - c. 在「選取 Snapshot 類型」頁面上，選取在**特定日期和時間取得 Snapshot**。在取消訂單之後，立即指定日期和時間，限定 Snapshot 日期。
    - d. 在「報告內容」頁面上，將**程序實例 ID**、**活動名稱**、**已啟動的活動**及**已完成的活動**新增至您的報告內容。
    - e. 在「過濾器內容」頁面上，指定**程序實例 ID** 等於 **your\_customer's\_process\_instance\_ID** 作為過濾器內容，然後執行報告。報告會顯示最花時間的活動。
    - f. 選擇性的: 如果您需要進一步資訊來評估延遲的確切起因，請編輯並重新執行您的報告。
    - g. 儲存您的報告定義。
  3. 最後您想要在未來避免如此的情況。您想要在每一個工作天結束時具有一個報告，列出所有作用中的訂單程序，這些訂單程序處於由於資源限制或失敗而超出時間限制的危險狀態。
    - a. 編輯您儲存的報告定義。在「選取 Snapshot 類型」頁面上，將 Snapshot 類型變更為**立即取得 Snapshot**、刪除過濾器內容**程序實例 ID** 等於 **your\_customer's\_process\_instance\_ID**，然後新增表示式大於 1 天的程序運作時間。
    - b. 執行您修改的報告，並檢查是否沒有程序實例符合新的過濾準則。
    - c. 儲存報告，以便您可在每一個工作天結束時執行它。

### 相關工作

第 316 頁的『建立使用者定義的報告』

使用者定義的程序及活動報告比預先定義的清單及圖表更有彈性。此外，您也可以儲存並重複使用您的報告定義，而且您可以匯出報告結果。

第 317 頁的『建立使用者定義的 Snapshot 報告』

您可以定義使用者定義的報告，在指定的日期和時間取得狀態資訊的 Snapshot。您也可以建立報告，包含報告期間內定期點的狀態 Snapshot，例如每一個月第一天的午夜。

第 319 頁的『建立使用者定義的期間報告』

您可以針對某個時段發生的程序或活動事件，建立使用者定義的報告。您也可以根據報告週期建立涵蓋多個期間的報告。

### 報告屬性

使用屬性，可以定義報告的內容，並過濾結果。可用的屬性視報告類型而定。

每個定義為報告內容的屬性都是報告中直欄的名稱。此外，使用屬性，可以過濾查詢的結果。您還可以定義未包含在報告中的屬性的過濾準則。

屬性	說明	Snap-shot 報告	期間報告
完成的活動	活動實例達到下列其中一個結束狀態的時間：已失敗、已完成、已跳過、已終止或已過期。	X	X
活動事件	活動事件的事件碼。	X	X
活動事件計數	活動實例所產生的活動事件的數目。	X	X
活動實例 ID	活動實例 ID。	X	X
活動類型	活動實例的類型。	X	X
活動前次使用者名稱	前次起始此活動動作的使用者名稱。	X	X
活動名稱	活動實例的名稱。	X	X
啟動的活動	啟動活動實例的時間。	X	X
活動狀態	活動實例在事件之後所處於的狀態。	X	X
活動範本 ID	活動範本 ID。	X	X
平均活動持續時間	所有活動實例的平均持續時間，以秒為單位。	X	X
平均程序持續時間	所有程序實例的平均持續時間，以秒為單位。	X	X
事件時間	事件發生的時間。	X	X
異常狀況本文	如果異常狀況觸發活動事件，則異常狀況訊息可以是事件資料的一部分，且稍後會儲存在此欄位中。	X	X
處於狀態的活動數目	處於指定狀態的活動實例數目。	X	
活動事件數目	指定期間發生的活動事件數目。		X
程序事件數目	指定期間發生的程序事件數目。		X
處於狀態的程序數目	處於指定狀態的程序實例數目。	X	
程序活動計數	至少產生一個事件的程序實例的活動數目。	X	X
程序活動事件計數	屬於程序實例的活動事件數目。	X	X
完成的程序	程序實例達到下列其中一個結束狀態的時間：已補償、補償已失敗、已失敗、已結束或已終止。	X	X
程序刪除時間	從「商業程序編排器」資料庫刪除程序的時間。	X	X
程序事件	程序實例事件的事件碼。	X	X
程序事件計數	程序實例所產生的程序事件數目。	X	X
程序實例 ID	程序實例 ID。	X	X
處理前次使用者名稱	前次起始此程序動作的使用者名稱。	X	X
啟動的程序	啟動程序實例的時間。	X	X
程序狀態	程序實例在事件之後所處於的狀態。	X	X
程序範本 ID	程序範本 ID。	X	X
程序範本名稱	與程序實例相關聯的程序範本。	X	X
程序運作時間	程序實例的持續時間。這個值是程序中包含的所有基本活動的運作次數總和。基本活動是沒有結構且不包含其他活動的活動。	X	X

屬性	說明	Snapshot 報告	期間報告
Snapshot 號碼	在報告週期的 Snapshot 報告中，此屬性會識別報告週期中的特定 Snapshot。	X	
時間片段號碼	在報告週期的期間報告中，此屬性會識別報告週期中的特定時間片段。		X
使用者名稱	與事件相關聯的使用者的使用者 ID。	X	X
有效起始時間	程序範本變成有效的時間。	X	X

## 商業程序編排器觀察程式的商業程序事件

如果要求監視 WebSphere Integration Developer 中的商業程序元素，則會傳送商業程序事件。這些事件的子集可用於「商業程序編排器觀察程式」。

商業程序可產生下列類型的事件：

- 『程序事件』
- 『活動事件』

視 WebSphere Integration Developer 中的設定而定，可能會發生 6.0.2 事件及 6.1 事件。

「商業程序編排器觀察程式」不需要事件提供商業資料。

## 程序事件

下表說明您可以使用「商業程序編排器觀察程式」報告的所有程序事件。

碼	說明
21000	啟動的程序
21001	程序已暫停
21002	程序已回復
21004	完成的程序
21005	程序已終止
21019	程序已重新啟動
42001	程序失敗
42003	程序補償中
42004	程序已補償
42046	程序補償已失敗
42009	程序終止中
42010	程序失敗中

## 活動事件

下表說明您可以使用「商業程序編排器觀察程式」報告的所有活動事件。

碼	說明
21006	活動備妥

碼	說明
21007	啟動的活動
21011	完成的活動
21021	已取消要求
21022	活動已要求
21027	活動已終止
21080	活動失敗
21081	活動已過期
42005	活動已跳過
42015	活動已停止
42031	活動已強制重試
42032	活動強制已完成
42036	活動已接收訊息

### 相關參考

第 499 頁的『商業程序事件』

如果要求監視 WebSphere Integration Developer 中的商業程序元素，則會傳送商業程序事件。此處可以找到商業程序可產生的所有事件的清單。

### 效能相關屬性

執行報告定義所需的時間可以不同。若要改進產生報告的效能，您可以最佳化報告定義。部分一般規格可協助您評估報告屬性對效能的影響。

#### 指定過濾器

使用適當的過濾器，可以限制所擷取資料的數量。請考量依日期或其他活動或程序實例內容來限制報告結果。對於 Snapshot 報告，請將 ReportAtSnapshotRange 配置參數設為適當的值。

#### 期間報告與 Snapshot 報告

Snapshot 報告會比期間報告的效能降低更多。

#### 使用報告週期進行報告

使用報告週期定義的報告會降低效能，特別是在對查詢定義了許多期間或 Snapshot 的情況下。

**聚集** 事件總數或實例平均持續時間之類的聚集必須處理大量資料，因此會降低效能。

#### 顯示的結果數目

如果您只要報告的部分結果，請指定臨界值來限制結果中的項目數。這會減少資料庫與使用者介面之間傳送的資料量。

然而，如果您定義了排序，則在資料排序之前，必須收集資料庫中的所有結果資料。在此情況下，減少顯示的結果數不會增進效能。您應該改為設定適當的過濾器表示式。

#### 事件及實例資訊

在「觀察程式」資料庫中，與事件相關的資訊儲存在事件資料庫表格中，而與活動及程序實例相關的資訊儲存在實例資料庫表格中。如果您建立同時包含實例相關資訊以及事件特定資訊的報告，則表格會結合以取得所需資訊。如果建

立只包含一種類型資訊的報告，則不會結合表格。因此，只包含一種類型資訊的報告，通常會比同時查詢實例相關及事件特定資訊的報告具有較佳的效能。

### 相關參考

第 226 頁的『變更商業程序編排器觀察程式的配置參數』

調整「商業程序編排器觀察程式」及事件收集程式應用程式的配置參數對於啓用驗證及改進效能很重要。

## 使用已儲存的使用者定義的報告定義

如果您已儲存報告定義，則可以在需要時執行報告、編輯報告定義，或使用報告定義副本來建立類似的報告。此外，您也可以非同步執行報告，以及匯出報告結果。

### 執行已儲存的使用者定義的報告定義

您可以在需要時執行已儲存的報告定義。如果您的報告包含參數，則可以設定在每次執行報告時，您有興趣的值。

1. 若要執行已儲存的報告定義，請在導覽窗格中按一下報告名稱。
  - 如果您的報告定義沒有包含參數，則會顯示結果報告。
  - 如果您的報告定義包含參數，則會顯示「執行報告」頁面。您可以變更參數值，然後按一下**執行**。



畫面上會顯示結果報告。

2. 選擇性的: 匯出報告結果。



若要以 CSV 格式匯出報告的資料，請按一下**匯出**。選取您想要開啓已產生的匯出資料，還是將資料儲存在硬碟上，再按一下**確定**。

### 非同步執行已儲存的使用者定義報告定義

您可以非同步執行已儲存的報告，以在查詢執行時繼續運作。

若要非同步執行已儲存的報告定義，請按一下**顯示蹦現功能表**圖示 ()，再按一下「非同步搜尋」圖示 ()。



如果您的報告定義包含參數，則會顯示「執行報告」頁面。您可以變更參數值，然後按一下「執行」。

- 在順利地完成非同步搜尋之後，「非同步搜尋已完成」圖示 () 會顯示在導覽窗格中。按一下報告名稱，以檢視搜尋結果。
- 如果非同步搜尋未順利完成，則會顯示「非同步搜尋失敗」圖示 ()。

### 使用蹦現功能表來匯出報告結果

若為已儲存的使用者定義報告，您不需要執行報告，即可匯出報告結果，以進行進一步的外部處理。

只有未包含參數的已儲存使用者定義的報告定義可以使用這個選項。

1. 若要匯出已儲存報告定義的報告結果，請按一下**顯示蹦現功能表**圖示 ()，再按一下「匯出」圖示 ()。



2. 選取您想要開啓已產生的匯出資料，還是儲存它，然後按一下**確定**。此時會匯出報告的資料。

### 使用匯出用戶端來匯出報告結果

若爲已儲存的使用者定義報告，您可以使用匯出用戶端命令行工具來執行報告，並匯出報告結果，以進行進一步的外部處理。

只有未包含參數的已儲存使用者定義的報告定義可以使用這個選項。

匯出用戶端工具 `wps_install_root/ProcessChoreographer/client/exportclient.jar` 必須已安裝在您的本端工作站上。

若要執行報告並匯出報告結果，請使用命令行來啓動匯出用戶端。

在 Windows 平台上，輸入：`java -jar exportclient.jaroptions`

在 Linux、UNIX 及 iOS 平台上，輸入：`java -jar exportclient.jaroptions`

您可以利用格式 `-option value`，直接在命令行上指定選項，並指定內容檔的名稱。選項具有格式 `option=value`。在命令行上指定的選項會比內容檔中指定的選項更優先處理。


下列是有效選項：



表 15. 匯出用戶端的有效選項

選項	說明
help	顯示用法資訊。
verbose	在匯出您可以用於除錯的結果時顯示相關資訊。
unicode	以 UTF-8 編碼匯出結果。預設值是本端作業系統編碼。
o	改寫任何現有檔案。如果檔案已存在，則預設值爲錯誤。
properties	這會定義包含其他選項的完整檔名。
url	「商業程序編排器觀察程式」正在執行中的完整 URL。預設值爲 <code>http://localhost:9080</code>
out	這會定義要儲存匯出結果的完整檔名。預設值爲 <code>report name.csv</code> 。
userid	已啓用安全時，需要有效使用者 ID。
password	已啓用安全時，需要有效密碼。
reportname	需要已儲存的報告定義的名稱。

### 編輯及複製已儲存的使用者定義的報告定義

您可以變更已儲存的報告定義的設定，或使用報告定義的副本，來建立類似的報告。

1. 按一下**顯示蹦現功能表**圖示 ()，並執行下列其中一項：

- 若要編輯報告定義，請按一下**編輯**圖示 ()。
- 若要複製報告定義，請按一下**複製**圖示 ()。

此時會開啓「摘要」頁面。這個頁面會顯示報告的時間設定、報告內容及過濾器設定。

按一下每一個摘要區段下方的鏈結，即可變更對應設定。您無法變更報告類型。

2. 選擇性的: 若要編輯時間設定，請按一下**修改報告的日期及報告週期設定**。

根據您定義的報告類型，會開啓「選取 Snapshot 類型」頁面或「選取期間類型」頁面。

3. 選擇性的: 若要修改報告內容，請按一下**修改結果內容**。

此時會開啓「指定報告內容」頁面。

若為具有報告週期的報告，屬性清單包含 Snapshot 號碼屬性或時間片段號碼屬性，視您定義的報告類型而定。您無法刪除這個屬性。

4. 選擇性的: 若要修改過濾器設定，請按一下**修改過濾器設定**。

此時會開啓「指定過濾器內容」頁面。

5. 在「摘要」頁面上，執行下列其中一項：

- 如果您的報告定義沒有包含參數，請按一下**執行**。

畫面上會顯示結果報告。

- 如果您的報告定義包含參數，請按**下一步**。

您可以變更參數值，然後按一下**執行**。畫面上會顯示結果報告。

如果報告結果不是您預期的，則您可以按一下**編輯**，以變更報告的設定。

6. 在「報告結果」頁面上，按一下**儲存**。如果您將建立報告定義的副本，請輸入新報告的名稱，並重新按一下**儲存**。

新報告會出現在導覽窗格中。

### 相關概念

第 311 頁的『時間處理』

在您的報告中，考量「商業程序編排器觀察程式」處理時間戳記與持續時間的方式。

### 相關參考

第 322 頁的『報告屬性』

使用屬性，可以定義報告的內容，並過濾結果。可用的屬性視報告類型而定。

第 324 頁的『商業程序編排器觀察程式的商業程序事件』

如果要求監視 WebSphere Integration Developer 中的商業程序元素，則會傳送商業程序事件。這些事件的子集可用於「商業程序編排器觀察程式」。



第 325 頁的『效能相關屬性』

執行報告定義所需的時間可以不同。若要改進產生報告的效能，您可以最佳化報告定義。部分一般規格可協助您評估報告屬性對效能的影響。

## 刪除已儲存的使用者定義的報告定義

若要使導覽窗格容易明瞭且易於管理，請刪除過期及多餘的報告定義。

您無法還原已刪除的報告定義。

若要刪除報告定義，請按一下顯示跳現功能表圖示 ()，然後按一下刪除圖示 ()。報告名稱會從導覽窗格中消失不見。



---

## 第 4 篇 開發及部署模組





---

## 第 10 章 開發商業程序及作業的用戶端應用程式

您可以使用塑型工具來建置及部署商業程序和作業。這些程序及作業會在執行時期進行互動，例如，啟動程序，或要求及完成作業。您可以使用「商業程序編排器瀏覽器」與程序及作業互動，或「商業程序編排器」API 以開發這些互動的自訂用戶端。

這些用戶端可以是 Enterprise JavaBeans™ (EJB) 用戶端、Web 服務用戶端，或是利用「商業程序編排器瀏覽器」JavaServer Faces (JSF) 元件的 Web 用戶端。「商業程序編排器」提供 Web 服務的 Enterprise JavaBeans (EJB) API 及介面，讓您開發這些用戶端。EJB API 可以透過任何 Java 應用程式進行存取 (包括另一個 EJB 應用程式)。Web 服務的介面可以從 Java 環境或 Microsoft .Net 環境中進行存取。

### 相關概念

第 14 頁的『商業程序的呼叫範例情況』

商業程序是 SCA (「服務元件架構」) 元件實作類型。它可以將服務公開給其他夥伴，並使用其他夥伴提供的服務。商業程序可以是透過「商業程序編排器 API」變為可用的服務提供者、其他 SCA 服務元件的 SCA 服務提供者，或呼叫其他 SCA 服務元件 (包括其他商業程序) 的 SCA 用戶端。



---

## 第 11 章 開發商業程序及人力作業的 EJB 用戶端應用程式

EJB API 提供一組一般方法來開發 EJB 用戶端應用程式，以與安裝於 WebSphere Process Server 上的商業程序及人力作業搭配使用。

使用這些 Enterprise JavaBeans (EJB) API，可以建立用戶端應用程式以執行下列作業：

- 管理程序及作業的生命週期，範圍包括啟動到完成時刪除
- 修復活動及程序
- 管理及分配工作團隊成員的工作量

EJB API 已提供為兩個 Stateless 階段作業 Enterprise Bean：

- BusinessFlowManagerService 介面提供商業程序應用程式的方法
- HumanTaskManagerService 介面提供作業型應用程式的方法

如需 EJB API 的相關資訊，請參閱 com.ibm.bpe.api 套件及 com.ibm.task.api 套件中的 Javadoc。

下列步驟提供在開發 EJB 用戶端應用程式時需要採取的動作概觀。

1. 決定應用程式要提供的功能。
2. 決定要使用的 Session Bean。

根據想要使用應用程式實作的範例情況，可以使用一個或兩個 Session Bean。

3. 決定應用程式使用者所需的權限。

必須將適當的授權角色指派給應用程式使用者，這樣應用程式使用者才能呼叫您併入應用程式中的方法，以及檢視這些方法傳回的物件和這些物件的屬性。建立適當的 Session Bean 實例時，WebSphere Application Server 會建立環境定義與實例的關聯性。環境定義包含呼叫者的主體 ID、群組成員資格清單及角色的相關資訊。這項資訊是用來檢查呼叫者對每個呼叫的權限。

Javadoc 包含每個方法的授權資訊。

4. 決定如何呈現應用程式。

可以本端或遠端呼叫 EJB API。

5. 開發應用程式。
  - a. 存取 EJB API。
  - b. 使用 EJB API 與程序或作業互動。
    - 查詢資料。
    - 使用資料。

---

### 存取 EJB API

Enterprise JavaBeans (EJB) API 已提供為兩個 Stateless 階段作業 Enterprise Bean。商業程序應用程式及作業應用程式會透過 Bean 的 Home 介面來存取適當的階段作業 Enterprise Bean。

`BusinessFlowManagerService` 介面提供商業程序應用程式的方法，而 `HumanTaskManagerService` 介面則提供作業型應用程式的方法。應用程式可以是任何 Java 應用程式 (包括另一個 Enterprise JavaBeans (EJB) 應用程式)。

## 存取 Session Bean 的遠端介面

EJB 用戶端應用程式會透過 Bean 的遠端 Home 介面，存取 Session Bean 的遠端介面。

Session Bean 可以是 `BusinessFlowManager Session Bean` (程序應用程式) 或 `HumanTaskManager Session Bean` (作業應用程式)。

1. 將 Session Bean 的遠端介面的參照新增至應用程式部署描述子。新增下列其中一個檔案的參照：

- `application-client.xml` 檔案，適用於 Java 2 Platform Enterprise Edition (J2EE) 用戶端應用程式
- `web.xml` 檔案，適用於 Web 應用程式
- `ejb-jar.xml` 檔案，適用於 Enterprise JavaBeans (EJB) 應用程式

下列範例顯示程序應用程式的遠端 Home 介面參照：

```
<ejb-ref>
  <ejb-ref-name>ejb/BusinessFlowManagerHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <home>com.ibm.bpe.api.BusinessFlowManagerHome</home>
  <remote>com.ibm.bpe.api.BusinessFlowManager</remote>
</ejb-ref>
```

下列範例顯示作業應用程式的遠端 Home 介面參照：

```
<ejb-ref>
  <ejb-ref-name>ejb/HumanTaskManagerHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <home>com.ibm.task.api.HumanTaskManagerHome</home>
  <remote>com.ibm.task.api.HumanTaskManager</remote>
</ejb-ref>
```

如果使用 WebSphere Integration Developer 將 EJB 參照新增至部署描述子，則在部署應用程式時會自動建立 EJB 參照的連結。如需新增 EJB 參照的相關資訊，請參閱 WebSphere Integration Developer 文件。

2. 將產生的 Stub 與應用程式包裝在一起。

如果您的應用程式與執行 `BPEContainer` 應用程式或 `TaskContainer` 應用程式所在的「Java 虛擬機器 (JVM)」不同，請完成下列動作：

- a. 如果是程序應用程式，請將 `<install_root>/ProcessChoreographer/client/bpe137650.jar` 檔案與應用程式的企業保存檔 (EAR) 包裝在一起。
- b. 如果是作業應用程式，請將 `<install_root>/ProcessChoreographer/client/task137650.jar` 檔案與應用程式的 EAR 檔包裝在一起。
- c. 將應用程式模組的資訊清單檔案中的 **Classpath** 參數設為包括 JAR 檔。

應用程式模組可以是 J2EE 應用程式、Web 應用程式或 EJB 應用程式。

- d. 如果在商業程序或人力作業中使用複式資料類型，而未在 EJB 應用程式或 Web 應用程式中執行用戶端，請將對應的 XSD 或 WSDL 檔與應用程式的 EAR 檔包裝在一起。
3. 透過「Java 命名和目錄介面 (JNDI)」尋找 Session Bean 的遠端 Home 介面。

下列範例顯示程序應用程式的這個步驟：

```
// Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

// Lookup the remote home interface of the BusinessFlowManager bean
Object result =
    initialContext.lookup("java:comp/env/ejb/BusinessFlowManagerHome");

// Convert the lookup result to the proper type
BusinessFlowManagerHome processHome =
    (BusinessFlowManagerHome)javax.rmi.PortableRemoteObject.narrow
    (result,BusinessFlowManagerHome.class);
```

Session Bean 的遠端 Home 介面包含 EJB 物件的 create 方法。此方法會傳回 Session Bean 的遠端介面。

#### 4. 存取 Session Bean 的遠端介面。

下列範例顯示程序應用程式的這個步驟：

```
BusinessFlowManager process = processHome.create();
```

Session Bean 的存取不保證呼叫者可以執行 Bean 提供的所有動作；呼叫者也必須獲得授權，才能執行這些物件。建立 Session Bean 的實例時，環境定義是與 Session Bean 的實例相關聯。環境定義包含呼叫者的主體 ID、群組成員資格清單，以及指出呼叫者是否具有其中一個「商業程序編排器」J2EE 角色。即使未設定廣域安全原則，還是可以使用環境定義來檢查呼叫者對每個呼叫的權限。如果未設定廣域安全原則，則呼叫者的主體 ID 值為 UNAUTHENTICATED。

#### 5. 呼叫服務介面公開的商業功能。

下列範例顯示程序應用程式的這個步驟：

```
process.initiate("MyProcessModel",input);
```

應用程式呼叫會視作交易來執行。您可以使用下列其中一種方式建立及結束交易：

- 由 WebSphere Application Server 自動進行 (部署描述子指定 TX\_REQUIRED)。
- 由應用程式明確進行。您可以將應用程式呼叫附在一個交易中：

```
// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("jta/usertransaction");

// Begin a transaction
transaction.begin();

// Applications calls ...

// On successful return, commit the transaction
transaction.commit();
```

**提示：**若要防止資料庫鎖定衝突，請避免平行執行如下的陳述式：

```
// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("jta/usertransaction");

transaction.begin();

//read the activity instance
process.getActivityInstance(aiid);
```

```
//claim the activity instance
process.claim(aiid);

transaction.commit();
```

`getActivityInstance` 方法及其他讀取作業會設定讀取鎖定。在這個範例中，活動實例上的讀取鎖定會升級至活動實例上的更新鎖定。當這些交易平行執行時，這可能導致資料庫死鎖。

## 範例

下列顯示內容是作業應用程式步驟 3 到 5 的範例。

```
//Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

//Lookup the remote home interface of the HumanTaskManager bean
Object result =
    initialContext.lookup("java:comp/env/ejb/HumanTaskManagerHome");

//Convert the lookup result to the proper type
HumanTaskManagerHome taskHome =
    (HumanTaskManagerHome)javax.rmi.PortableRemoteObject.narrow
    (result,HumanTaskManagerHome.class);

...

//Access the remote interface of the session bean.
HumanTaskManager task = taskHome.create();

...

//Call the business functions exposed by the service interface
task.callTask(tkiid,input);
```

## 存取 Session Bean 的本端介面

EJB 用戶端應用程式會透過 Bean 的本端 Home 介面，存取 Session Bean 的本端介面。

Session Bean 可以是 `BusinessFlowManager Session Bean` (用於程序應用程式) 或 `HumanTaskManager Session Bean` (用於人力作業應用程式)。

1. 將 Session Bean 的本端介面的參照新增至應用程式部署描述子。新增下列其中一個檔案的參照：
  - `application-client.xml` 檔案，適用於 Java 2 Platform Enterprise Edition (J2EE) 用戶端應用程式
  - `web.xml` 檔案，適用於 Web 應用程式
  - `ejb-jar.xml` 檔案，適用於 Enterprise JavaBeans (EJB) 應用程式

下列範例顯示程序應用程式的本端 Home 介面參照：

```
<ejb-local-ref>
  <ejb-ref-name>ejb/LocalBusinessFlowManagerHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home>com.ibm.bpe.api.LocalBusinessFlowManagerHome</local-home>
  <local>com.ibm.bpe.api.LocalBusinessFlowManager</local>
</ejb-local-ref>
```

下列範例顯示作業應用程式的本端 Home 介面參照：

```

<ejb-local-ref>
  <ejb-ref-name>ejb/LocalHumanTaskManagerHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home>com.ibm.task.api.LocalHumanTaskManagerHome</local-home>
  <local>com.ibm.task.api.LocalHumanTaskManager</local>
</ejb-local-ref>

```

如果使用 WebSphere Integration Developer 將 EJB 參照新增至部署描述子，則在部署應用程式時會自動建立 EJB 參照的連結。如需新增 EJB 參照的相關資訊，請參閱 WebSphere Integration Developer 文件。

2. 透過「Java 命名和目錄介面 (JNDI)」尋找 Session Bean 的本端 Home 介面。

下列範例顯示程序應用程式的這個步驟：

```

// Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

// Lookup the local home interface of the BusinessFlowManager bean

LocalBusinessFlowManagerHome processHome =
    (LocalBusinessFlowManagerHome)initialContext.lookup
    ("java:comp/env/ejb/LocalBusinessFlowManagerHome");

```

Session Bean 的本端 Home 介面包含 EJB 物件的 create 方法。此方法會傳回 Session Bean 的本端介面。

3. 存取 Session Bean 的本端介面。

下列範例顯示程序應用程式的這個步驟：

```
LocalBusinessFlowManager process = processHome.create();
```

Session Bean 的存取不保證呼叫者可以執行 Bean 提供的所有動作；呼叫者也必須獲得授權，才能執行這些物件。建立 Session Bean 的實例時，環境定義是與 Session Bean 的實例相關聯。環境定義包含呼叫者的主體 ID、群組成員資格清單，以及指出呼叫者是否具有其中一個「商業程序編排器」J2EE 角色。即使未設定廣域安全原則，還是可以使用環境定義來檢查呼叫者對每個呼叫的權限。如果未設定廣域安全原則，則呼叫者的主體 ID 值為 UNAUTHENTICATED。

4. 呼叫服務介面公開的商業功能。

下列範例顯示程序應用程式的這個步驟：

```
process.initiate("MyProcessModel",input);
```

應用程式呼叫會視作交易來執行。您可以使用下列其中一種方式建立及結束交易：

- 由 WebSphere Application Server 自動進行 (部署描述子指定 TX\_REQUIRED)。
- 由應用程式明確進行。您可以將應用程式呼叫附在一個交易中：

```

// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("jta/usertransaction");

// Begin a transaction
transaction.begin();

// Applications calls ...

// On successful return, commit the transaction
transaction.commit();

```



**提示:** 若要防止資料庫死鎖，請避免平行執行如下的陳述式：

```
// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("jta/usertransaction");

transaction.begin();

//read the activity instance
process.getActivityInstance(aiid);
//claim the activity instance
process.claim(aiid);

transaction.commit();
```

`getActivityInstance` 方法及其他讀取作業會設定讀取鎖定。在這個範例中，活動實例上的讀取鎖定會升級至活動實例上的更新鎖定。當這些交易平行執行時，這可能導致資料庫死鎖

## 範例

下列顯示內容是適用於作業應用程式步驟 2 到 4 的範例。

```
//Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

//Lookup the local home interface of the HumanTaskManager bean
LocalHumanTaskManagerHome taskHome =
    (LocalHumanTaskManagerHome)initialContext.lookup
    ("java:comp/env/ejb/LocalHumanTaskManagerHome");

...
//Access the local interface of the session bean
LocalHumanTaskManager task = taskHome.create();

...
//Call the business functions exposed by the service interface
task.callTask(tkid,input);
```

---

## 查詢商業程序及作業相關物件

用戶端應用程式會使用商業程序及作業相關物件。您可以查詢資料庫中的商業程序及作業相關物件，來擷取這些物件的特定內容。

在配置「商業程序編排器」期間，關聯式資料庫會與商業程序儲存器及作業儲存器相關聯。此資料庫會儲存所有範本 (模型) 及實例 (執行時期) 資料，用來管理商業程序及作業。您可以使用 SQL 類似的語法來查詢此資料。

您可以執行一次性查詢來擷取物件的特定內容。也可以儲存經常使用的查詢，並將這些儲存查詢併入應用程式中。

### 商業程序及作業相關物件的查詢

使用服務 API 的 `query` 方法或 `queryAll` 方法，可擷取關於商業程序及作業的儲存資訊。

所有使用者都可以呼叫 `query` 方法，而且此方法會傳回具有工作項目的物件內容。只有具有下列其中一種 J2EE 角色的使用者，才可以呼叫 `queryAll` 方法：`BPESystemAdministrator`、`TaskSystemAdministrator`、`BPESystemMonitor` 或 `TaskSystemMonitor`。此方法會傳回儲存在資料庫中所有物件的內容。

所有 API 查詢都會對映至 SQL 查詢。而產生 SQL 查詢的表單則視下列部分而定：

- 查詢是否由具有其中一種 J2EE 角色的人員所呼叫。
- 查詢的物件。提供您預先定義的資料庫視圖，以查詢物件內容。
- 插入存取控制的 from 子句、結合條件及使用者指定的條件。

您可以在查詢中同時併入自訂內容及變數內容。如果您在查詢中併入數個自訂內容或變數內容，這會導致在對應資料庫表格上的自行結合。視資料庫系統而定，這些 query() 呼叫可能具有效能連帶作用。

您也可以使用 createStoredQuery 方法，在「商業程序編排器」資料庫中儲存查詢。當您定義儲存查詢時，可提供查詢準則。當儲存查詢執行時會動態套用準則，也就是說，會在執行時期組譯資料。如果儲存查詢包含參數，則也會在查詢執行時解析這些參數。

如需「商業程序編排器」API 的相關資訊，請參閱程序相關方法的 com.ibm.bpe.api 套件及作業相關方法的 com.ibm.task.api 套件中的 Javadoc。

## API query 方法的語法

「商業程序編排器」API 查詢的語法與 SQL 查詢類似。查詢可以包括 select 子句、where 子句、order-by 子句、跳過值組參數、臨界值參數及時區參數。

查詢的語法視物件類型而定。下表會顯示每個不同物件類型的語法。

表 16.

物件	語法
程序範本	<code>ProcessTemplateData[] queryProcessTemplates (java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timezone);</code>
作業範本	<code>TaskTemplate[] queryTaskTemplates (java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timezone);</code>
商業程序及作業相關資料	<code>QueryResultSet query (java.lang.String selectClause, java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer skipTuples, java.lang.Integer threshold, java.util.TimeZone timezone);</code>

### Select 子句:

查詢函數中的 select 子句會識別查詢要傳回的物件內容。

select 子句會說明查詢結果。它會指定名稱的清單，來識別要傳回的物件內容 (結果的直欄)。它的語法與 SQL SELECT 子句的語法類似；使用逗點可以區隔子句的各個部分。子句的每個部分必須指定來自其中一個預先定義視圖的直欄。直欄必須使用視圖名稱及直欄名稱完整指定。在 QueryResultSet 物件中傳回的直欄，會以與 select 子句中指定的直欄相同的順序顯示。

select 子句不支援 SQL 總計函數，例如 AVG()、SUM()、MIN() 或 MAX()。

若要選取多個名稱/值配對的內容 (例如可以查詢的自訂內容及變數內容)，請將一位數的計數器新增至視圖名稱。此計數器可以使用的值為 1 至 9。

### select 子句的範例

- "WORK\_ITEM.OBJECT\_TYPE, WORK\_ITEM.REASON"  
取得關聯物件的物件類型，以及工作項目的指派原因。
- "DISTINCT WORK\_ITEM.OBJECT\_ID"  
取得所有物件 ID (沒有重複項目)，呼叫者具有其工作項目。
- "ACTIVITY.TEMPLATE\_NAME, WORK\_ITEM.REASON"  
取得呼叫者具有其工作項目的活動名稱，以及其指派原因。
- "ACTIVITY.STATE, PROCESS\_INSTANCE.STARTER"  
取得活動的狀態，以及其關聯程序實例的起始者。
- "DISTINCT TASK.TKIID, TASK.NAME"  
取得所有作業 ID 及名稱 (沒有重複項目)，呼叫者具有其工作項目。
- "TASK\_CPROP1.STRING\_VALUE, TASK\_CPROP2.STRING\_VALUE"  
取得在 where 子句中進一步指定的自訂內容值。
- "QUERY\_PROPERTY1.STRING\_VALUE, QUERY\_PROPERTY2.INT\_VALUE"  
取得可以查詢的變數內容的值。這些部分會在 where 子句中進一步指定。
- "COUNT( DISTINCT TASK.TKIID)"  
計算滿足 where 子句的唯一作業的工作項目數。

### Where 子句:

查詢函數中的 where 子句會說明要套用以查詢網域的過濾準則。

where 子句的語法與 SQL WHERE 子句的語法類似。您不需要將 SQL from 子句或 join 述詞明確地新增至 API where 子句中，在執行查詢時，會自動新增這些建構。如果您不想要套用過濾準則，您必須針對 where 子句指定 null。

where 子句語法支援：

- 關鍵字：AND、OR、NOT
- 比較運算子：=、<=、<、<>、>、>=、LIKE  
LIKE 作業支援針對查詢的資料庫所定義的萬用字元。
- 集作業：IN

也會適用下列規則：

- 將物件 ID 常數指定為 ID('string-rep-of-oid')。
- 將二進位常數指定為 BIN('UTF-8 string')。
- 使用符號常數而非整數列舉。例如，不要指定活動狀態表示式 ACTIVITY.STATE=2，而是指定 ACTIVITY.STATE=ACTIVITY.STATE.STATE\_READY。

- 如果比較陳述式的內容值包含單引號 (')，請變更為雙引號，例如 "TASK\_CPROP.STRING\_VALUE='d'automatisation'"。
- 請透過將一位數的字尾新增至視圖名稱，來參照多個名稱/值配對的內容，例如自訂內容。例如："TASK\_CPROP1.NAME='prop1' AND "TASK\_CPROP2.NAME='prop2'"
- 將時間戳記常數指定為 TS('yyyy-mm-ddThh:mm:ss')。若要參照現行日期，請將 CURRENT\_DATE 指定為時間戳記。

在時間戳記中，您必須至少指定一個日期或時間值：

- 如果您只指定一個日期，則會將時間值設為零。
- 如果您只指定一個時間，則會將日期設為現行日期。
- 如果您指定日期，則年份必須由四位數組成；月份及日期值是選用的。缺少的月份及日期值將設為 01。例如，TS('2003') 與 TS('2003-01-01T00:00:00') 相同。
- 如果您指定時間，則會以 24 小時系統來表示這些值。例如，如果現行日期是 2003 年 1 月 1 日，則 TS('T16:04') 或 TS('16:04') 與 TS('2003-01-01T16:04:00') 相同。

### where 子句的範例

- 比較物件 ID 與現有的 ID

```
"WORK_ITEM.WIID = ID('_WI:800c00ed.df8d7e7c.feffff80.38')"
```

此類型的 where 子句通常是使用來自前一個呼叫的現有物件 ID 動態建立的。如果這個物件 ID 儲存在 wiid1 變數中，則子句的建構如下所示：

```
"WORK_ITEM.WIID = ID('" + wiid1.toString() + '" )"
```

- 使用時間戳記

```
"ACTIVITY.STARTED >= TS('2002-06-1T16.00.00')"
```

- 使用符號常數

```
"WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_OWNER"
```

- 使用布林值 true 及 false

```
"ACTIVITY.BUSINESS_RELEVANCE = TRUE"
```

- 使用自訂內容

```
"TASK_CPROP1.NAME = 'prop1' AND " TASK_CPROP1.STRING_VALUE = 'v1' AND  
TASK_CPROP2.NAME = 'prop2' AND " TASK_CPROP2.STRING_VALUE = 'v2'"
```

### Order by 子句:

查詢函數中的 order by 子句會指定查詢結果集的排序準則。

您可以從用以排序結果的視圖來指定直欄清單。這些直欄必須完整符合視圖及直欄的名稱。最佳作法是指定位在 select 子句中的直欄。

order-by 子句語法與 SQL order-by 子句的語法類似；使用逗點可以區隔子句的每個部分。您也可以指定 ASC 依遞增次序來排序直欄，或指定 DESC 依遞減次序排序直欄。如果不想要排序查詢結果集，則您必須針對 order by 子句指定 null。

排序準則會套用至伺服器，亦即，伺服器的語言環境可用於排序。如果指定多個直欄，則查詢結果集會依第一個直欄的值排序，然後依第二個直欄的值排序，以此類推。您不可以依 SQL 查詢中的相同位置，在 order-by 子句中指定直欄。

### order by 子句的範例

- "PROCESS\_TEMPLATE.NAME"

依程序範本名稱字母順序排序查詢結果。

- "PROCESS\_INSTANCE.CREATED, PROCESS\_INSTANCE.NAME DESC"

依建立日期排序查詢結果，且對於特定日期，依程序實例名稱字母相反順序排序結果。

- "ACTIVITY.OWNER, ACTIVITY.TEMPLATE\_NAME, ACTIVITY.STATE"

依活動擁有者，然後依活動範本名稱，再依活動的狀態排序查詢結果。

### 跳過值組參數:

跳過值組參數可指定查詢結果集值組數目，而這些值組是自查詢結果集開頭開始予以忽略且不會傳回給查詢結果集中的呼叫者。

與臨界值參數搭配使用此參數，可實作用戶端應用程式中的分頁，例如擷取前 20 個項目，然後是下 20 個項目，以此類推。

如果此參數設為 null，且未設定臨界值參數，則會傳回所有限定值組。

### 跳過值組參數的範例

- new Integer(5)

指定不會傳回前五個限定值組。

### 臨界值參數:

查詢函數中的臨界值參數會限制查詢結果集中，從伺服器傳回至用戶端的物件數。

因為正式作業範例情況中的查詢結果集可以包含幾千甚至幾百萬個項目，最佳作法是一律指定臨界值。例如，臨界值參數可以用於圖形使用者介面，因為介面一次只適合顯示少量的項目。如果您相對應地設定臨界值參數，資料庫查詢會更快，且需要從伺服器傳送至用戶端的資料會更少。

如果此參數設為 null，且未設定跳過值組參數，則會傳回所有限定物件。

### 臨界值參數的範例

- new Integer(50)

指定要傳回 50 個限定值組。

### 時區參數:

查詢函數中的時區參數會定義查詢中時間戳記常數的時區。

啟動查詢的用戶端時區與處理查詢的伺服器時區可能不同。例如，使用時區參數，可指定 where 子句中所使用時間戳記常數的時區，來指定當地時間。在查詢結果集中傳回的日期，與查詢中指定的時區相同。

如果參數設為 null，則時間戳記常數假設為「世界標準時間 (UTC)」。

### 時區參數的範例

- `process.query("ACTIVITY.AIID",  
"ACTIVITY.STARTED > TS('2005-01-01T17:40')",  
(String)null,  
(Integer)null,  
java.util.TimeZone.getDefault() );`

會傳回啓動時間晚於當地時間 2005 年 1 月 1 日 17:40 的活動的物件 ID。

- `process.query("ACTIVITY.AIID",  
"ACTIVITY.STARTED > TS('2005-01-01T17:40')",  
(String)null, (Integer)null, (TimeZone)null);`

會傳回啓動時間晚於 UTC 的 2005 年 1 月 1 日 17:40 的活動的物件 ID。例如，此指定時間比美東標準時間早 6 個小時。

### 儲存查詢中的參數:

儲存查詢是儲存在資料庫中，且依名稱識別的查詢。當執行查詢時，會動態組譯限定值組。若要讓儲存查詢可重複使用，您可以在查詢定義中使用執行時期所解析的參數。

例如，您已定義自訂內容來儲存客戶名稱。您可以定義查詢來傳回與特定客戶 ACME Co. 相關聯的作業。若要查詢此資訊，查詢中的 `where` 子句可能與下列範例相似：

```
String whereClause =
"TASK.STATE = TASK.STATE.STATE_READY
AND WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER
AND TASK_CPROP.NAME = 'company' AND TASK_CPROP.STRING_VALUE = 'ACME Co.';
```

若要讓此查詢可重複使用，以便您也可以搜尋客戶 BCME Ltd，您可以針對自訂內容的值使用參數。如果您將參數新增至作業查詢，則它可能與下列範例相似：

```
String whereClause =
"TASK.STATE = TASK.STATE.STATE_READY
AND WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER
AND TASK_CPROP.NAME = 'company' AND TASK_CPROP.STRING_VALUE = '@param1';
```

在執行時期，會從傳遞至 `query` 方法的參數清單解析 `@param1` 參數。下列規則適用於在查詢中使用參數：

- 參數僅限在 `where` 子句中使用。
- 參數是字串。
- 在執行時期，會使用字串取代來取代參數。如果您需要特殊字元，您必須在 `where` 子句中指定這些字元，或在執行時期作為參數的一部分傳入。
- 參數名稱包含字串 `@param`，且連接一個整數。最低的數值是 1，其會指向在執行時期傳遞至查詢 API 的參數清單中的第一個項目。
- 在 `where` 子句內可以多次使用參數；該參數的所有出現次數都由相同的值來取代。

### 查詢結果:

查詢結果集包含查詢的結果。

結果集的元素就是物件內容，而這些物件滿足呼叫者所給定的 `where` 子句，而且授權呼叫者可以進行查看。您可以使用 `API next` 方法的相對方式，或使用 `first` 及 `last` 方法的絕對方式，來讀取元素。因為查詢結果集隱含的游標起始位置是第一個元素之前，所以您必須呼叫第一個或下一個方法，才能讀取元素。您可以使用 `size` 方法來判定集中元素的數目。

查詢結果集中的元素包含工作項目的已選取屬性，以及其相關聯的參照物件，例如活動實例及程序實例。QueryResultSet 元素的第一個屬性 (直欄) 會指定在查詢要求的 select 子句中，所指定的第一個屬性值。QueryResultSet 元素的第二個屬性 (直欄) 會指定在查詢要求的 select 子句中，所指定的第二個屬性值，以此類推。

您可以透過呼叫與屬性類型相容的方法，以及透過指定適當的直欄索引，來擷取屬性的值。直欄索引的編號從 1 開始。

屬性類型	方法
String	getString
OID	getOID
Timestamp	getTimestamp getString getTimestampAsLong
Integer	getInteger getShort getLong getString getBoolean
Boolean	getBoolean getShort getInteger getLong getString
byte[]	getBinary

範例：

執行下列查詢：

```
QueryResultSet resultSet = process.query("ACTIVITY.STARTED,
                                         ACTIVITY.TEMPLATE_NAME AS NAME,
                                         WORK_ITEM.WIID, WORK_ITEM.REASON",
                                         (String)null, (String)null,
                                         (Integer)null, (TimeZone)null);
```

傳回的查詢結果集具有四個直欄：

- 直欄 1 是時間戳記
- 直欄 2 是字串
- 直欄 3 是物件 ID
- 直欄 4 是整數

您可以使用下列方法來擷取屬性值：

```
while (resultSet.next())
{
    java.util.Calendar activityStarted = resultSet.getTimestamp(1);
    String templateName = resultSet.getString(2);
    WIID wiid = (WIID) resultSet.getOID(3);
    Integer reason = resultSet.getInteger(4);
}
```



例如，您可以使用結果集的顯示名稱作為列印表格的標題。這些名稱是視圖的直欄名稱，或查詢中由 AS 子句所定義的名稱。您可以使用下列方法來擷取範例中的顯示名稱：

```
resultSet.getColumnDisplayName(1) returns "STARTED"  
resultSet.getColumnDisplayName(2) returns "NAME"  
resultSet.getColumnDisplayName(3) returns "WIID"  
resultSet.getColumnDisplayName(4) returns "REASON"
```

## 使用者指定的存取條件

透過 API 查詢產生 SQL SELECT 陳述式時，會新增使用者指定的存取條件。這些條件會保證只會將那些物件傳回給滿足呼叫者所指定條件的呼叫者，以及獲得授權的呼叫者。

新增的存取條件是視使用者是否為系統管理者而定。

## 非系統管理者使用者呼叫的查詢

產生的 SQL WHERE 子句結合了 API where 子句與使用者特有的存取控制條件。此查詢只會擷取那些使用者獲權存取的物件，亦即，只限使用者擁有工作項目的那些物件。工作項目代表將使用者或使用者群組指派給商業物件的授權角色（例如作業或程序）。例如，如果使用者 John Smith 是給定作業的可能擁有者角色的成員，則會有代表此關係的工作項目物件。

例如，如果非系統管理者的使用者查詢作業，則在未啟用群組工作項目時，會在 WHERE 子句中新增下列存取條件：

```
FROM TASK TA, WORK_ITEM WI  
WHERE WI.OBJECT_ID = TA.TKIID  
AND ( WI.OWNER_ID = 'user'  
      OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

因此，如果 John Smith 想要取得他本身為可能擁有者的作業清單，則 API where 子句可能會與下列類似：

```
"WORK_ITEM.REASON == WORK_ITEM.REASON.REASON_POTENTIAL_OWNER"
```

此 API where 子句會在 SQL 陳述式中產生下列存取條件：

```
FROM TASK TA, WORK_ITEM WI  
WHERE WI.OBJECT_ID = TA.TKIID  
AND ( WI.OWNER_ID = 'JohnSmith'  
      OR WI.OWNER_ID = null AND WI.EVERYBODY = true)  
AND WI.REASON = 1
```

這也表示如果 John Smith 想要查看他本身是程序讀者或程序管理者的活動及作業，以及他本身沒有工作項目的活動及作業，則必須將 PROCESS\_INSTANCE 視圖的內容新增至查詢的 select、where 或 order-by 子句（例如，PROCESS\_INSTANCE.PIID）。

如果啟用群組工作項目，則會在 WHERE 子句中新增其他存取條件，以容許使用者存取群組可以存取的物件。

## 系統管理者呼叫的查詢

系統管理者可以呼叫 query 方法，來擷取具有相關聯工作項目的物件。在此情況下，會將具有 WORK\_ITEM 視圖的結合新增至產生的 SQL 查詢中，但是沒有 WORK\_ITEM.OWNER\_ID 的存取控制條件。

在此情況下，作業的 SQL 查詢會包含下列項目：

```
FROM TASK TA, WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
```

### queryAll 查詢

只有系統管理者或系統監視者才可以呼叫這類型的查詢。而且不會新增存取控制的條件及 WORK\_ITEM 視圖的結合。這類型的查詢會傳回所有物件的所有資料。

### query 及 queryAll 方法的範例

這些範例顯示各種一般 API 查詢的語法，以及處理查詢時產生的相關聯 SQL 陳述式。

#### 範例：查詢處於備妥狀態的作業：

此範例顯示如何使用 query 方法來擷取登入使用者可以使用的作業。

John Smith 想要取得已指派給他的作業清單。若要讓使用者可以使用作業，則作業必須處於備妥狀態。登入使用者也必須具有作業的可能擁有者工作項目。下列程式碼 Snippet 顯示此查詢的 query 方法呼叫：

```
query( "DISTINCT TASK.TKIID",
      "TASK.KIND IN ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING )
      AND " +
      "TASK.STATE = TASK.STATE.STATE_READY AND " +
      "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

下列動作是在產生 SQL SELECT 陳述式時進行：

- 將存取控制的條件新增至 where 子句。此範例假設未啟用群組工作項目。
- 將常數 (例如 TASK.STATE.STATE\_READY) 取代為其數值。
- 新增 FROM 子句及結合條件。

下列程式碼 Snippet 顯示透過 API 查詢產生的 SQL 陳述式：

```
SELECT DISTINCT TASK.TKIID
FROM TASK TA, WORK_ITEM WI,
WHERE WI.OBJECT_ID = TA.TKIID
AND TA.KIND IN ( 101, 105 )
AND TA.STATE = 2
AND WI.REASON = 1
AND ( WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

若要將 API 查詢限制為特定程序的作業 (例如，sampleProcess)，則查詢會與下列類似：

```
query( "DISTINCT TASK.TKIID",
      "PROCESS_TEMPLATE.NAME = 'sampleProcess' AND "+
      "TASK.KIND IN ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING )
      AND " +
      "TASK.STATE = TASK.STATE.STATE_READY AND " +
      "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

#### 範例：查詢處於已要求狀態的作業：

此範例顯示如何使用 query 方法來擷取登入使用者已要求的作業。

使用者 John Smith 想要搜尋他已要求且仍處於已要求狀態的作業。指定「John Smith 所要求的」的條件是 `TASK.OWNER = 'JohnSmith'`。下列程式碼 Snippet 顯示查詢的 query 方法呼叫：

```
query( "DISTINCT TASK.TKIID",
      "TASK.STATE = TASK.STATE.STATE_CLAIMED AND " +
      "TASK.OWNER = 'JohnSmith'",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

下列程式碼 Snippet 顯示透過 API 查詢產生的 SQL 陳述式：

```
SELECT DISTINCT TASK.TKIID
FROM   TASK TA, WORK_ITEM WI,
WHERE  WI.OBJECT_ID = TA.TKIID
AND    TA.STATE = 8
AND    TA.OWNER = 'JohnSmith'
AND    ( WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

要求作業時，會為作業擁有者建立工作項目。因此，形成 John Smith 已要求作業查詢的替代方式會在查詢中新增下列條件，而不是使用 `TASK.OWNER = 'JohnSmith'`：

```
WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_OWNER
```

然後，查詢會與下列程式碼 Snippet 類似：

```
query( "DISTINCT TASK.TKIID",
      "TASK.STATE = TASK.STATE.STATE_CLAIMED AND " +
      "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_OWNER",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

下列動作是在產生 SQL SELECT 陳述式時進行：

- 將存取控制的條件新增至 where 子句。此範例假設未啟用群組工作項目。
- 將常數 (例如 `TASK.STATE.STATE_READY`) 取代為其數值。
- 新增 FROM 子句及結合條件。

下列程式碼 Snippet 顯示透過 API 查詢產生的 SQL 陳述式：

```
SELECT DISTINCT TASK.TKIID
FROM   TASK TA, WORK_ITEM WI,
WHERE  WI.OBJECT_ID = TA.TKIID
AND    TA.STATE = 8
AND    WI.REASON = 4
AND    ( WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

John 即將休假，因此他的小組領導人 Anne Grant 想要確認他現行的工作量。Anne 擁有系統管理者權限。她所呼叫的查詢與 John 所呼叫的查詢相同。然而，因為 Anne 是管理者，所以產生的 SQL 陳述式會不同。下列程式碼 Snippet 顯示產生的 SQL 陳述式：

```
SELECT DISTINCT TASK.TKIID
FROM   TASK TA, WORK_ITEM WI,
WHERE  TA.TKIID = WI.OBJECT_ID =
AND    TA.STATE = 8
AND    TA.OWNER = 'JohnSmith')
```

因為 Anne 是管理者，所以存取控制條件不會新增至 WHERE 子句中。

**範例：查詢呈報：**

此範例顯示如何使用 query 方法來擷取登入使用者的呈報。

呈報作業時，會建立呈報接收者工作項目。使用者 Mary Jones 想要查看已呈報給她的作業清單。下列程式碼 Snippet 顯示查詢的 query 方法呼叫：

```
query( "DISTINCT ESCALATION.ESIID, ESCALATION.TKIID",
      "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_ESCALATION_RECEIVER",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

下列動作是在產生 SQL SELECT 陳述式時進行：

- 將存取控制的條件新增至 where 子句。此範例假設未啟用群組工作項目。
- 將常數 (例如 TASK.STATE.STATE\_READY) 取代為其數值。
- 新增 FROM 子句及結合條件。

下列程式碼 Snippet 顯示透過 API 查詢產生的 SQL 陳述式：

```
SELECT DISTINCT ESCALATION.ESIID, ESCALATION.TKIID
FROM   ESCALATION ESC, WORK_ITEM WI
WHERE  ESC.ESIID = WI.OBJECT_ID
AND    WI.REASON = 10
AND
( WI.OWNER_ID = 'MaryJones' OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

#### 範例：使用 queryAll 方法:

此範例顯示如何使用 queryAll 方法來擷取屬於程序範本的所有活動。

只有具有系統管理者或系統監視者權限的使用者，才可以使用 queryAll 方法。下列程式碼 Snippet 顯示查詢的 queryAll 方法呼叫，以擷取屬於程序範本 sampleProcess 的所有活動：

```
queryAll( "DISTINCT ACTIVITY.AIID",
         "PROCESS_TEMPLATE.NAME = 'sampleProcess'",
         (String)null, (String)null, (Integer)null, (TimeZone)null )
```

下列程式碼 Snippet 顯示透過 API 查詢產生的 SQL 查詢：

```
SELECT DISTINCT ACTIVITY.AIID
FROM   ACTIVITY AI, PROCESS_TEMPLATE PT
WHERE  AI.PTID = PT.PTID
AND    PT.NAME = 'sampleProcess'
```

因為呼叫是由管理者所呼叫，所以存取控制條件不會新增至產生的 SQL 陳述式中。也不會新增具有 WORK\_ITEM 視圖的結合。這表示查詢會擷取程序範本的所有活動 (包括那些沒有工作項目的活動)。

#### 範例：在查詢中併入查詢內容:

此範例顯示如何使用 query 方法來擷取屬於商業程序的作業。此程序定義的查詢內容是您想要併入搜尋的查詢內容。

例如，您想要搜尋所有屬於商業程序且處於備妥狀態的人力作業。此程序具有值為 CID\_12345 的查詢內容 customerID 及名稱空間。下列程式碼 Snippet 顯示查詢的 query 方法呼叫：

```
query ( " DISTINCT TASK.TKIID, TASK_TEMPL.NAME, TASK.STATE,
        PROCESS_INSTANCE.NAME",
      " QUERY_PROPERTY.NAME = 'customerID' AND " +
      " QUERY_PROPERTY.STRING_VALUE = 'CID_12345' AND " +
      " QUERY_PROPERTY.NAMESPACE =
      'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
      " TASK.KIND IN
```

```

        ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING ) AND " +
" TASK.STATE = TASK.STATE.STATE_READY ",
(String)null, (String)null, (Integer)null, (TimeZone)null );

```

如果您想要在查詢中新增第二個具有給定名稱空間的查詢內容 (例如, **Priority**) , 則查詢的 `query` 方法呼叫會與下列類似 :

```

query ( " DISTINCT TASK.TKIID, TASK_TEMPL.NAME, TASK.STATE,
        PROCESS_INSTANCE.NAME",
" QUERY_PROPERTY1.NAME = 'customerID' AND " +
" QUERY_PROPERTY1.STRING_VALUE = 'CID_12345' AND " +
" QUERY_PROPERTY1.NAMESPACE =
'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
" QUERY_PROPERTY2.NAME = 'Priority' AND " +
" QUERY_PROPERTY2.NAMESPACE =
'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
" TASK.KIND IN
( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING ) AND " +
" TASK.STATE = TASK.STATE.STATE_READY ",
(String)null, (String)null, (Integer)null, (TimeZone)null );

```

如果您在查詢中新增多個查詢內容, 則必須對每個新增的內容進行編號 (如程式碼 `Snippet` 中所示) 。然而, 查詢自訂內容會影響效能; 效能會隨著查詢中自訂內容數目的增加而減少。

#### 範例：在查詢中併入自訂內容:

此範例顯示如何使用 `query` 方法來擷取具有自訂內容的作業。

例如, 您想要搜尋所有具有自訂內容 **customerID** (值為 `CID_12345`) 且處於備妥狀態的人力作業。下列程式碼 `Snippet` 顯示查詢的 `query` 方法呼叫 :

```

query ( " DISTINCT TASK.TKIID ",
" TASK_CPROP.NAME = 'customerID' AND " +
" TASK_CPROP.STRING_VALUE = 'CID_12345' AND " +
" TASK.KIND IN
( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING ) AND " +
" TASK.STATE = TASK.STATE.STATE_READY ",
(String)null, (String)null, (Integer)null, (TimeZone)null );

```

如果您想要擷取作業及其自訂內容, 則查詢的 `query` 方法呼叫會與下列類似 :

```

query ( " DISTINCT TASK.TKIID, TASK_CPROP.NAME, TASK_CPROP.STRING_VALUE",
" TASK.KIND IN
( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING ) AND " +
" TASK.STATE = TASK.STATE.STATE_READY ",
(String)null, (String)null, (Integer)null, (TimeZone)null );

```

下列程式碼 `Snippet` 顯示透過此 API 查詢產生的 SQL 陳述式 :

```

SELECT DISTINCT TA.TKIID , TACP.NAME , TACP.STRING_VALUE
FROM TASK TA LEFT JOIN TASK_CPROP TACP ON (TA.TKIID = TACP.TKIID),
WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
AND TA.KIND IN ( 101, 105 )
AND TA.STATE = 2
AND (WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID IS NULL AND WI.EVERYBODY = 1 )

```

此 SQL 陳述式包含 `TASK` 視圖與 `TASK_CPROP` 視圖之間的外部結合。這表示會擷取滿足 `WHERE` 子句的作業, 即使它們沒有任何自訂內容也一樣。

## 商業程序及人力作業物件查詢的預先定義視圖

針對商業程序及人力作業物件，已提供預先定義的資料庫視圖。當您查詢這些物件的參照資料時，可使用這些視圖。

當您使用預先定義的視圖時，您不需要針對視圖直欄明確新增 `join` 述詞，將會自動新增這些建構。您可以使用服務 API (`BusinessFlowManagerService` 或 `HumanTaskManagerService`) 的一般查詢函數來查詢此資料。您也可以使用對應的 `HumanTaskManagerDelegate` API 方法，或由 `ExecutableQuery` 介面的實作提供的預先定義查詢。

**註：** 視圖可能會包含未說明的直欄。這些直欄僅供內部使用。

### ACTIVITY 視圖：

使用此預先定義的資料庫視圖，以針對活動進行查詢。

表 17. ACTIVITY 視圖中的直欄

直欄名稱	類型	備註
PIID	ID	程序實例 ID。
AIID	ID	活動實例 ID。
PTID	ID	程序範本 ID。
ATID	ID	活動範本 ID。
KIND	整數	活動的類型。可能的值如下：  KIND_INVOKE (21) KIND_RECEIVE (23) KIND_REPLY (24) KIND_THROW (25) KIND_RETHROW (46) KIND_TERMINATE (26) KIND_WAIT (27) KIND_COMPENSATE (29) KIND_SEQUENCE (30) KIND_EMPTY (3) KIND_SWITCH (32) KIND_WHILE (34) KIND_PICK (36) KIND_FLOW (38) KIND_SCOPE (40) KIND_SCRIPT (42) KIND_STAFF (43) KIND_ASSIGN (44) KIND_CUSTOM (45) KIND_FOR_EACH_PARALLEL (49) KIND_FOR_EACH_SERIAL (47)
COMPLETED	時間戳記	完成活動的時間。
ACTIVATED	時間戳記	啟動活動的時間。
FIRST_ACTIVATED	時間戳記	第一次啟動活動的時間。
STARTED	時間戳記	啟動活動的時間。

表 17. ACTIVITY 視圖中的直欄 (繼續)

直欄名稱	類型	備註
STATE	整數	活動的狀態。可能的值如下：  STATE_INACTIVE (1) STATE_READY (2) STATE_RUNNING (3) STATE_PROCESSING_UNDO (14) STATE_SKIPPED (4) STATE_FINISHED (5) STATE_FAILED (6) STATE_TERMINATED (7) STATE_CLAIMED (8) STATE_TERMINATING (9) STATE_FAILING (10) STATE_WAITING (11) STATE_EXPIRED (12) STATE_STOPPED (13)
OWNER	字串	擁有者的主體 ID。
DESCRIPTION	字串	如果活動範本說明包含位置保留元，則此直欄包含已解析位置保留元的活動實例說明。
TEMPLATE_NAME	字串	關聯活動範本的名稱。
TEMPLATE_DESCR	字串	關聯活動範本的說明。
BUSINESS_RELEVANCE	布林	指定該活動是否為商業相關項目。可能的值如下：  <b>TRUE</b> 該活動是商業相關項目。您可以在「商業程序編排器瀏覽器」中檢視活動狀態。  <b>FALSE</b> 該活動不是商業相關項目。
EXPIRES	時間戳記	活動到期的日期和時間。如果活動已過期，則為此事件發生的日期和時間。

#### ACTIVITY\_ATTRIBUTE 視圖:

使用此預先定義的資料庫視圖，以針對活動的自訂內容進行查詢。

表 18. ACTIVITY\_ATTRIBUTE 視圖中的直欄

直欄名稱	類型	備註
AIID	ID	含有自訂內容的活動實例 ID。
NAME	字串	自訂內容的名稱。
VALUE	字串	自訂內容的值。

#### ACTIVITY\_SERVICE 視圖:

使用此預先定義的資料庫視圖，以針對活動服務進行查詢。



表 19. *ACTIVITY\_SERVICE* 視圖中的直欄

直欄名稱	類型	備註
EIID	ID	事件實例的 ID。
AIID	ID	等待事件的活動實例 ID。
PIID	ID	包含事件的程序實例 ID。
VTID	ID	說明事件的服務範本 ID。
PORT_TYPE	字串	埠類型的名稱。
NAME_SPACE_URI	字串	名稱空間的 URI。
OPERATION	字串	服務的作業名稱。

### APPLICATION\_COMP 視圖:

使用此預先定義的資料庫視圖，可以查詢應用程式元件 ID 及作業的預設值。

表 20. *APPLICATION\_COMP* 視圖中的直欄

直欄名稱	類型	備註
ACOID	字串	應用程式元件的 ID。
BUSINESS_ RELEVANCE	布林	元件的預設作業商業相關性原則。作業範本或作業中的定義可以改寫此值。屬性會影響審核追蹤的記載。可能的值如下： <b>TRUE</b> 作業與商業相關且受審核。 <b>FALSE</b> 作業與商業不相關且不受審核。
NAME	字串	應用程式元件的名稱。
SUPPORT_ AUTOCLAIM	布林	元件的預設自動要求原則。如果此屬性設為 <b>TRUE</b> ，則當單一使用者是可能的擁有者時，可以自動要求作業。作業範本或作業中的定義可以改寫此值。
SUPPORT_CLAIM_ SUSP	布林	元件的預設值，決定是否可以要求已暫停的作業。如果此屬性設為 <b>TRUE</b> ，則可以要求已暫停的作業。作業範本或作業中的定義可以改寫此值。
SUPPORT_ DELEGATION	布林	元件的預設作業委派原則。如果此屬性設為 <b>TRUE</b> ，則可以修改作業的工作項目指派。這表示可以建立、刪除或轉移工作項目。
SUPPORT_ FOLLOW_ON	布林	元件的預設後續作業原則。如果此屬性設為 <b>TRUE</b> ，則可以針對作業建立後續作業。作業範本或作業中的定義可以改寫此值。
SUPPORT_ SUB_TASK	布林	元件的預設子作業原則。如果此屬性設為 <b>TRUE</b> ，則可以針對作業建立子作業。作業範本或作業中的定義可以改寫此值。

### ESCALATION 視圖:

使用此預先定義的資料庫視圖，可以查詢呈報資料。

表 21. ESCALATION 視圖中的直欄

直欄名稱	類型	備註
ESIID	字串	呈報實例的 ID。
ACTION	整數	呈報觸發的動作。可能的值如下： <b>ACTION_CREATE_WORK_ITEM (1)</b> 針對每個呈報接收者都建立一個工作項目。 <b>ACTION_SEND_EMAIL (2)</b> 傳送電子郵件至每個呈報接收者。 <b>ACTION_CREATE_EVENT (3)</b> 建立及發佈事件。
ACTIVATION_STATE	整數	如果對應作業達到下列其中一個狀態，則會建立呈報實例： <b>ACTIVATION_STATE_READY (2)</b> 指出人員或參與作業已準備接受要求。 <b>ACTIVATION_STATE_RUNNING (3)</b> 指出起始作業已啟動或在執行中。 <b>ACTIVATION_STATE_CLAIMED (8)</b> 指出已要求作業。 <b>ACTIVATION_STATE_WAITING_FOR_SUBTASK (20)</b> 指出作業正等待子作業的完成。
ACTIVATION_TIME	時間戳記	啟動呈報的時間。
AT_LEAST_EXP_STATE	整數	呈報預期的作業狀態。如果發生逾時，則會將此屬性值與作業狀態進行比較。可能的值如下： <b>AT_LEAST_EXPECTED_STATE_CLAIMED (8)</b> 指定已要求作業。 <b>AT_LEAST_EXPECTED_STATE_ENDED (20)</b> 指定作業處於最終狀態 (FINISHED、FAILED、TERMINATED 或 EXPIRED)。 <b>AT_LEAST_EXPECTED_STATE_SUBTASKS_COMPLETED (21)</b> 指定作業的所有子作業已完成。
ESTID	字串	對應呈報範本的 ID。
FIRST_ESIID	字串	鏈結中第一個呈報的 ID。
INCREASE_PRIORITY	整數	指出提高作業優先順序的方法。可能的值如下： <b>INCREASE_PRIORITY_NO (1)</b> 不提高作業優先順序。 <b>INCREASE_PRIORITY_ONCE (2)</b> 作業優先順序每次都會提高一等。 <b>INCREASE_PRIORITY_REPEATED (3)</b> 每次呈報重複執行時，作業優先順序都會提高一等。
NAME	字串	呈報的名稱。

表 21. ESCALATION 視圖中的直欄 (繼續)

直欄名稱	類型	備註
STATE	整數	呈報的狀態。可能的值如下：  STATE_INACTIVE (1) STATE_WAITING (2) STATE_ESCALATED (3) STATE_SUPERFLUOUS (4)
TKIID	字串	呈報所屬作業實例 ID。

#### ESCALATION\_CPROP 視圖:

使用此預先定義的資料庫視圖，可以查詢呈報的自訂內容。

表 22. ESCALATION\_CPROP 視圖中的直欄

直欄名稱	類型	備註
ESIID	字串	呈報 ID。
NAME	字串	內容名稱。
DATA_TYPE	字串	非字串自訂內容類別的類型。
STRING_VALUE	字串	「字串」類型自訂內容的值。

#### ESCALATION\_DESC 視圖:

使用此預先定義的資料庫視圖，可以查詢呈報的多種語言敘述性資料。

表 23. ESCALATION\_DESC 視圖中的直欄

直欄名稱	類型	備註
ESIID	字串	呈報 ID。
LOCALE	字串	與說明或顯示名稱相關聯的語言環境名稱。
DESCRIPTION	字串	作業範本的說明。
DISPLAY_NAME	字串	呈報的敘述性名稱。

#### ESC\_TEMPL 視圖:

使用此預先定義資料庫視圖，可以查詢呈報範本的資料。

表 24. ESC\_TEMPL 視圖中的直欄

直欄名稱	類型	備註
ESTID	字串	呈報範本的 ID。
ACTION	整數	呈報觸發的動作。可能的值如下：  <b>ACTION_CREATE_WORK_ITEM (1)</b> 針對每個呈報接收者都建立一個工作項目。  <b>ACTION_SEND_EMAIL (2)</b> 傳送電子郵件至每個呈報接收者。  <b>ACTION_CREATE_EVENT (3)</b> 建立及發佈事件。

表 24. ESC\_TEMPL 視圖中的直欄 (繼續)

直欄名稱	類型	備註
ACTIVATION_STATE	整數	<p>如果對應作業達到下列其中一個狀態，則會建立呈報實例：</p> <p><b>ACTIVATION_STATE_READY (2)</b> 指出人員或參與作業已準備接受要求。</p> <p><b>ACTIVATION_STATE_RUNNING (3)</b> 指出起始作業已啟動或在執行中。</p> <p><b>ACTIVATION_STATE_CLAIMED (8)</b> 指定已要求作業。</p> <p><b>ACTIVATION_STATE_WAITING_FOR_SUBTASK (20)</b> 指出作業正等待子作業的完成。</p>
AT_LEAST_EXP_STATE	整數	<p>呈報預期的作業狀態。如果發生逾時，則會將此屬性值與作業狀態進行比較。可能的值如下：</p> <p><b>AT_LEAST_EXPECTED_STATE_CLAIMED (8)</b> 指定已要求作業。</p> <p><b>AT_LEAST_EXPECTED_STATE_ENDED (20)</b> 指定作業處於最終狀態 (FINISHED、FAILED、TERMINATED 或 EXPIRED)。</p> <p><b>AT_LEAST_EXPECTED_STATE_SUBTASKS_COMPLETED (21)</b> 指定作業的所有子作業已完成。</p>
CONTAINMENT_CTX_ID	字串	<p>如果呈報範本屬於行內作業範本，則包含環境定義就是程序範本。如果呈報範本環境定義屬於獨立式作業範本，則包含環境定義就是作業範本。</p>
FIRST_ESTID	字串	<p>呈報範本鏈結中第一個呈報範本的 ID。</p>
INCREASE_PRIORITY	整數	<p>指出提高作業優先順序的方法。可能的值如下：</p> <p><b>INCREASE_PRIORITY_NO (1)</b> 不提高作業優先順序。</p> <p><b>INCREASE_PRIORITY_ONCE (2)</b> 作業優先順序每次都會提高一等。</p> <p><b>INCREASE_PRIORITY_REPEATED (3)</b> 每次呈報重複執行時，作業優先順序都會提高一等。</p>
NAME	字串	<p>呈報範本的名稱。</p>
PREVIOUS_ESTID	字串	<p>呈報範本鏈結中前一個呈報範本的 ID。</p>
TKTID	字串	<p>呈報範本所屬的作業範本 ID。</p>

#### ESC\_TEMPL\_CPROP 視圖:

使用此預先定義的資料庫視圖，可以查詢呈報範本的自訂內容。

表 25. ESC\_TEMPL\_CPROP 視圖中的直欄

直欄名稱	類型	備註
ESTID	字串	呈報範本的 ID。
NAME	字串	內容名稱。
TKTID	字串	呈報範本所屬的作業範本 ID。
DATA_TYPE	字串	非字串自訂內容類別的類型。
VALUE	字串	「字串」類型自訂內容的值。

#### ESC\_TEMPL\_DESC 視圖:

使用此預先定義資料庫視圖，可以查詢呈報範本的多種語言敘述性資料。

表 26. ESC\_TEMPL\_DESC 視圖中的直欄

直欄名稱	類型	備註
ESTID	字串	呈報範本的 ID。
LOCALE	字串	與說明或顯示名稱相關聯的語言環境名稱。
TKTID	字串	呈報範本所屬的作業範本 ID。
DESCRIPTION	字串	作業範本的說明。
DISPLAY_NAME	字串	呈報的敘述性名稱。

#### PROCESS\_ATTRIBUTE 視圖:

使用此預先定義的資料庫視圖，以針對程序的自訂內容進行查詢。

表 27. PROCESS\_ATTRIBUTE 視圖中的直欄

直欄名稱	類型	備註
PIID	ID	含有自訂內容的程序實例 ID。
NAME	字串	自訂內容的名稱。
VALUE	字串	自訂內容的值。

#### PROCESS\_INSTANCE 視圖:

使用此預先定義的資料庫視圖，以針對程序實例進行查詢。

表 28. PROCESS\_INSTANCE 視圖中的直欄

直欄名稱	類型	備註
PTID	ID	程序範本 ID。
PIID	ID	程序實例 ID。
NAME	字串	程序實例的名稱。

表 28. *PROCESS\_INSTANCE* 視圖中的直欄 (繼續)

直欄名稱	類型	備註
STATE	整數	程序實例的狀態。可能的值如下：  STATE_READY (1) STATE_RUNNING (2) STATE_FINISHED (3) STATE_COMPENSATING (4) STATE_INDOUBT (10) STATE_FAILED (5) STATE_TERMINATED (6) STATE_COMPENSATED (7) STATE_COMPENSATION_FAILED (12) STATE_TERMINATING (8) STATE_FAILING (9) STATE_SUSPENDED (11)
CREATED	時間戳記	建立程序實例的時間。
STARTED	時間戳記	啟動程序實例的時間。
COMPLETED	時間戳記	完成程序實例的時間。
PARENT_PIID	ID	母項程序實例的 ID。
PARENT_NAME	字串	母項程序實例的名稱。
TOP_LEVEL_PIID	ID	最上層程序實例的程序實例 ID。如果沒有最上層程序實例，則這是現行程序實例的程序實例 ID。
TOP_LEVEL_NAME	字串	最上層程序實例的名稱。如果沒有最上層程序實例，則這是現行程序實例的名稱。
STARTER	字串	程序實例起始者的主體 ID。
DESCRIPTION	字串	如果程序範本的說明包含位置保留元，則此直欄包含已解析位置保留元的程序實例說明。
TEMPLATE_NAME	字串	關聯程序範本的名稱。
TEMPLATE_DESCR	字串	關聯程序範本的說明。
RESUMES	時間戳記	自動回復程序實例的時間。

### PROCESS\_TEMPLATE 視圖:

使用此預先定義的資料庫視圖，以針對程序範本進行查詢。

表 29. *PROCESS\_TEMPLATE* 視圖中的直欄

直欄名稱	類型	備註
PTID	ID	程序範本 ID。
NAME	字串	程序範本的名稱。
VALID_FROM	時間戳記	可以開始實例化程序範本的時間。
TARGET_NAMESPACE	字串	程序範本的目標名稱空間。
APPLICATION_NAME	字串	程序範本所屬的企業應用程式的名稱。
VERSION	字串	使用者定義的版本。
CREATED	時間戳記	在資料庫中建立程序範本的時間。

表 29. *PROCESS\_TEMPLATE* 視圖中的直欄 (繼續)

直欄名稱	類型	備註
STATE	整數	指定程序範本是否可用於建立程序實例。可能的值如下：  STATE_STARTED (1) STATE_STOPPED (2)
EXECUTION_MODE	整數	指定衍生自此程序範本的程序實例可如何執行。可能的值如下：  EXECUTION_MODE_MICROFLOW (1) EXECUTION_MODE_LONG_RUNNING (2)
DESCRIPTION	字串	程序範本的說明。
COMP_SPHERE	整數	指定程序範本中微流程實例的補償行為；會結合現有的補償領域或建立補償領域。  可能的值如下：  COMP_SPHERE_REQUIRED (2) COMP_SPHERE_SUPPORTS (4)
DISPLAY_NAME	字串	程序的敘述性名稱。

#### QUERY\_PROPERTY 視圖:

使用此預先定義的資料庫視圖，以針對程序層次變數進行查詢。

表 30. *QUERY\_PROPERTY* 視圖中的直欄

直欄名稱	類型	備註
PIID	ID	程序實例 ID。
VARIABLE_NAME	字串	程序層次變數的名稱。
NAME	字串	查詢內容的名稱。
NAMESPACE	字串	查詢內容的名稱空間。
GENERIC_VALUE	字串	未對映至下列其中一種已定義類型的內容類型的字串表示法： STRING_VALUE, NUMBER_VALUE, DECIMAL_VALUE, 或 TIMESTAMP_VALUE。
STRING_VALUE	字串	如果內容類型對映至字串類型，則這是字串值。
NUMBER_VALUE	整數	如果內容類型對映至整數類型，則這是整數值。
DECIMAL_VALUE	十進位	如果內容類型對映至浮點類型，則這是十進位值。
TIMESTAMP_VALUE	時間戳記	如果內容類型對映至時間戳記類型，則這是時間戳記值。

#### TASK 視圖:



使用此預先定義的資料庫視圖，以針對作業物件進行查詢。

表 31. TASK 視圖中的直欄

直欄名稱	類型	備註
TKIID	ID	作業實例的 ID。
ACTIVATED	時間戳記	啓動作業的時間。
APPLIC_ DEFAULTS_ID	ID	指定作業預設值的應用程式元件 ID。
APPLIC_NAME	字串	作業所屬的企業應用程式的名稱。
BUSINESS_REL- EVANCE	布林	指定作業是否為商業相關項目。屬性會影響審核追蹤的記載。可能的值如下：  <b>TRUE</b> 作業與商業相關且受審核。 <b>FALSE</b> 作業與商業不相關且不受審核。
COMPLETED	時間戳記	完成作業的時間。
CONTAINMENT_ CTX_ID	ID	此作業的包含環境定義。此屬性決定作業的生命週期。當刪除作業的包含環境定義時，也會刪除作業。
CTX_AUTHORIZA- TION	整數	允許作業擁有者存取作業環境定義。可能的值如下：  <b>AUTH_NONE</b> 沒有關聯環境定義物件的權限。  <b>AUTH_READER</b> 關聯環境定義物件的作業要求讀者權限，例如，讀取程序實例的內容。
DUE	時間戳記	作業的截止時間。
EXPIRES	時間戳記	作業到期的日期。
FIRST_ACTIVATED	時間戳記	第一次啓動作業的時間。
FOLLOW_ON_TKIID	ID	後續作業的實例 ID。
HIERARCHY_POSI- TION	整數	可能的值如下：  <b>HIERARCHY_POSITION_TOP_TASK (0)</b> 作業階層的最上層作業。  <b>HIERARCHY_POSITION_SUB_TASK (1)</b> 作業是作業階層中的子作業。  <b>HIERARCHY_POSITION_FOLLOW_ON_TASK (2)</b> 作業是作業階層中的後續作業。
IS_AD_HOC	布林	指出是否已在執行時期或從作業範本動態建立此作業。
IS_ESCALATED	布林	指出是否已發生此作業的呈報。
IS_INLINE	布林	指出作業是否為商業程序中的行內作業。
IS_WAIT_FOR_ SUB_TK	布林	指出上層作業是否正在等待子作業達到結束狀態。

表 31. TASK 視圖中的直欄 (繼續)

直欄名稱	類型	備註
KIND	整數	作業類型。可能的值如下：  <b>KIND_HUMAN (101)</b> 說明作業是人力所建立及處理的協同作業。  <b>KIND_WPC_STAFF_ACTIVITY (102)</b> 說明作業是人力作業，其為 WebSphere Business Integration Server Foundation 第 5 版商業程序的人員活動。  <b>KIND_ORIGINATING (103)</b> 說明作業是支援人員與電腦互動的呼叫作業，這可讓人們建立、起始及啟動服務。  <b>KIND_PARTICIPATING (105)</b> 說明作業是支援電腦與人員互動的待辦作業，這可讓人員實作服務。  <b>KIND_ADMINISTRATIVE (106)</b> 說明作業是管理作業。
LAST_MODIFIED	時間戳記	前次修改作業的時間。
LAST_STATE_CHANGE	時間戳記	前次修改作業狀態的時間。
NAME	字串	作業的名稱。
NAME_SPACE	字串	用於分類作業的名稱空間。
ORIGINATOR	字串	作業發送者的主體 ID。
OWNER	字串	作業擁有者的主體 ID。
PARENT_CONTEXT_ID	字串	此作業的上層環境定義。此屬性提供呼叫應用程式元件中對應環境定義的索引鍵。上層環境定義由建立作業的應用程式元件設定。
PRIORITY	整數	作業的優先順序。
RESUMES	時間戳記	自動回復作業的時間。
STARTED	時間戳記	啓動作業的時間 (STATE_RUNNING、STATE_CLAIMED)。
STARTER	字串	作業起始者的主體 ID。

表 31. TASK 視圖中的直欄 (繼續)

直欄名稱	類型	備註
STATE	整數	作業的狀態。可能的值如下： <b>STATE_READY (2)</b> 說明已準備要求作業。 <b>STATE_RUNNING (3)</b> 說明作業已啟動或在執行中。 <b>STATE_FINISHED (5)</b> 說明作業已順利完成。 <b>STATE_FAILED (6)</b> 說明作業未順利完成。 <b>STATE_TERMINATED (7)</b> 說明由於外部或內部要求，已終止作業。 <b>STATE_CLAIMED (8)</b> 說明已要求作業。 <b>STATE_EXPIRED (12)</b> 說明作業由於已超出其指定持續時間而結束。 <b>STATE_FORWARDED (101)</b> 說明作業已完成但有後續作業。
SUPPORT_AUTOCLAIM	布林	指出如果將此作業指派給單一使用者，是否會自動要求此作業。
SUPPORT_CLAIM_SUSP	布林	指出如果此作業已暫停，是否會要求此作業。
SUPPORT_DELEGATION	布林	指出此作業是否支援透過建立、刪除或轉移工作項目的工作委派。
SUPPORT_FOLLOW_ON	布林	指出此作業是否支援後續作業的建立。
SUPPORT_SUB_TASK	布林	指出此作業是否支援子作業的建立。
SUSPENDED	布林	指出作業是否已暫停。
TKTID	ID	作業範本 ID。
TOP_TKIID	ID	最上層作業實例 ID (如果這是子作業的話)。
TYPE	字串	用於分類作業的類型。

#### TASK\_CPROP 視圖:

使用此預先定義的資料庫視圖，可以查詢作業物件的自訂內容。

表 32. TASK\_CPROP 視圖中的直欄

直欄名稱	類型	備註
TKIID	字串	作業實例 ID。
NAME	字串	內容名稱。
DATA_TYPE	字串	非字串自訂內容類別的類型。
STRING_VALUE	字串	「字串」類型自訂內容的值。

### TASK\_DESC 視圖:

使用此預先定義資料庫的視圖，可以查詢作業物件的多種語言敘述性資料。

表 33. TASK\_DESC 視圖中的直欄

直欄名稱	類型	備註
TKIID	字串	作業實例 ID。
LOCALE	字串	與說明或顯示名稱相關聯的語言環境名稱。
DESCRIPTION	字串	作業的說明。
DISPLAY_NAME	字串	作業的敘述性名稱。

### TASK\_TEMPL 視圖:

此預先定義資料庫視圖保留可用來實例化作業的資料。

表 34. TASK\_TEMPL 視圖中的直欄

直欄名稱	類型	備註
TKTID	字串	作業範本 ID。
VALID_FROM	時間戳記	作業範本可用於實例化的時間。
APPLIC_ DEFAULTS_ID	字串	指定作業範本預設值的應用程式元件 ID。
APPLIC_NAME	字串	作業範本所屬的企業應用程式的名稱。
BUSINESS_ REL- EVANCE	布林	指定作業範本是否為商業相關項目。屬性會影響審核追蹤的記載。可能的值如下： <b>TRUE</b> 作業與商業相關且受審核。 <b>FALSE</b> 作業與商業不相關且不受審核。
CONTAINMENT_ CTX_ID	ID	此作業範本的包含環境定義。此屬性決定作業範本的生命週期。當刪除包含環境定義時，也會刪除作業範本。
CTX_ AUTHORIZA- TION	整數	允許作業擁有者存取作業環境定義。可能的值如下： <b>AUTH_NONE</b> 沒有關聯環境定義物件的權限。 <b>AUTH_READER</b> 關聯環境定義物件的作業要求讀者權限，例如，讀取程序實例的內容。
DEFINITION_NAME	字串	「作業執行語言 (TEL)」檔案中作業範本定義的名稱。
DEFINITION_NS	字串	TEL 檔案中作業範本定義的名稱空間。
IS_AD_HOC	布林	指出是否已在執行時期動態建立此作業範本，或者作業何時已部署為 EAR 檔的一部分。
IS_INLINE	布林	指出是否已建立此作業範本模型作為商業程序內的作業。

表 34. TASK\_TEMPL 視圖中的直欄 (繼續)

直欄名稱	類型	備註
KIND	整數	衍生自此作業範本的作業類型。可能的值如下： <b>KIND_HUMAN (101)</b> 說明作業是人力所建立及處理的協同作業。 <b>KIND_ORIGINATING (103)</b> 說明作業是支援人員與電腦互動的呼叫作業，這可讓人們建立、起始及啟動服務。 <b>KIND_PARTICIPATING (105)</b> 說明作業是支援電腦與人員互動的待辦作業，這可讓人員實作服務。 <b>KIND_ADMINISTRATIVE (106)</b> 說明作業是管理作業。
NAME	字串	作業範本的名稱。
NAMESPACE	字串	用於分類作業範本的名稱空間。
PRIORITY	整數	作業範本的優先順序。
STATE	整數	作業範本的狀態。可能的值如下： <b>STATE_STARTED (1)</b> 指定可用於建立作業實例的作業範本。 <b>STATE_STOPPED (2)</b> 指定作業範本已停止。無法從此狀態的作業範本建立作業實例。
SUPPORT_AUTOCLAIM	布林	指出如果將衍生自此作業範本的作業指派給單一使用者，是否會自動要求該作業。
SUPPORT_CLAIM_SUSP	布林	指出如果衍生自此作業範本的作業已暫停，是否會要求該作業。
SUPPORT_DELEGATION	布林	指出衍生自此作業範本的作業是否支援使用建立、刪除或轉移工作項目的工作委派。
SUPPORT_FOLLOW_ON	布林	指出作業範本是否支援後續作業的建立。
SUPPORT_SUB_TASK	布林	指出作業範本是否支援子作業的建立。
TYPE	字串	用於分類作業範本的類型。

#### TASK\_TEMPL\_CPROP 視圖:

使用此預先定義的資料庫視圖，可以查詢作業範本的自訂內容。

表 35. TASK\_TEMPL\_CPROP 視圖中的直欄

直欄名稱	類型	備註
TKTID	字串	作業範本 ID。
NAME	字串	內容名稱。
DATA_TYPE	字串	非字串自訂內容類別的類型。
STRING_VALUE	字串	「字串」類型自訂內容的值。

#### TASK\_TEMPL\_DESC 視圖:

使用此預先定義的資料庫視圖，可以查詢作業範本物件的多種語言敘述性資料。

表 36. *TASK\_TEMPL\_DESC* 視圖中的直欄

直欄名稱	類型	備註
TKTID	字串	作業範本 ID。
LOCALE	字串	與說明或顯示名稱相關聯的語言環境名稱。
DESCRIPTION	字串	作業範本的說明。
DISPLAY_NAME	字串	作業範本的敘述性名稱。

### WORK\_ITEM 視圖:

使用此預先定義的資料庫視圖，以針對工作項目以及程序、作業與呈報的權限資料進行查詢。

表 37. *WORK\_ITEM* 視圖中的直欄

直欄名稱	類型	備註
WIID	ID	工作項目 ID。
OWNER_ID	字串	擁有者的主體 ID。
GROUP_NAME	字串	關聯群組工作清單的名稱。
EVERYBODY	布林	指定是否每個人都擁有此工作項目。
OBJECT_TYPE	整數	<p>相關物件的類型。可能的值如下：</p> <p><b>OBJECT_TYPE_ACTIVITY (1)</b> 指定已為活動建立工作項目。</p> <p><b>OBJECT_TYPE_PROCESS_INSTANCE (3)</b> 指定已為程序實例建立工作項目。</p> <p><b>OBJECT_TYPE_TASK_INSTANCE (5)</b> 指定已為作業建立工作項目。</p> <p><b>OBJECT_TYPE_TASK_TEMPLATE (6)</b> 指定已為作業範本建立工作項目。</p> <p><b>OBJECT_TYPE_ESCALATION_INSTANCE (7)</b> 指定已為呈報實例建立工作項目。</p> <p><b>OBJECT_TYPE_APPLICATION_COMPONENT (9)</b> 指定已為應用程式元件建立工作項目。</p>
OBJECT_ID	ID	關聯物件 (例如，關聯程序或作業) 的 ID。
ASSOC_OBJECT_TYPE	整數	ASSOC_OID 屬性所參照物件 (例如，作業、程序或外部物件) 的類型。使用 OBJECT_TYPE 屬性的值。
ASSOC_OID	ID	與含有工作項目的物件相關聯的物件 ID。例如，包含活動實例 (已為其建立此工作項目) 的程序實例的程序實例 ID (PIID)。

表 37. WORK\_ITEM 視圖中的直欄 (繼續)

直欄名稱	類型	備註
REASON	整數	指派工作項目的原因。可能的值如下：  REASON_POTENTIAL_STARTER (5) REASON_POTENTIAL_INSTANCE_CREATOR (11) REASON_POTENTIAL_STARTER (1) REASON_EDITOR (2) REASON_READER (3) REASON_ORIGINATOR (9) REASON_OWNER (4) REASON_STARTER (6) REASON_ESCALATION_RECEIVER (10) REASON_ADMINISTRATOR (7)
CREATION_TIME	時間戳記	建立工作項目的日期和時間。

## 在查詢中使用變數過濾資料

查詢結果會傳回與查詢準則相符的物件。您可能會想要根據變數值來過濾這些結果。

您可以定義程序在執行時期於其程序模型中使用的變數。您可以宣告這些變數的哪些部分可以進行查詢。

例如，John Smith 撥打他投保的保險公司服務電話，以瞭解他受損車子的保險理賠進度。而理賠管理者使用客戶 ID 來找到該理賠申請。

1. 選擇性的: 列出程序中可以查詢的變數內容。

使用程序範本 ID 來識別程序。如果知道可以查詢的變數，則可以跳過此步驟。

```
List variableProperties = process.getQueryProperties(ptid);
for (int i = 0; i < variableProperties.size(); i++)
{
    QueryProperty queryData = (QueryProperty)variableProperties.get(i);
    String variableName = queryData.getVariableName();
    String name = queryData.getName();
    int mappedType = queryData.getMappedType();
    ...
}
```

2. 列出具有符合過濾準則的變數的程序實例。

在此程序中，客戶 ID 已塑型為可以查詢的變數 customerClaim 部分。因此，您可以使用客戶的 ID 來尋找理賠申請。

```
QueryResultSet result = process.query
("PROCESS_INSTANCE.NAME, QUERY_PROPERTY.STRING_VALUE",
 "QUERY_PROPERTY.VARIABLE_NAME = 'customerClaim' AND " +
 "QUERY_PROPERTY.NAME = 'customerID' AND " +
 "QUERY_PROPERTY.STRING_VALUE like 'Smith%'",
 (String)null, (Integer)null,
 (Integer)null, (TimeZone)null );
```

此動作會傳回查詢結果集，內含程序實例名稱以及 ID 開頭為 Smith 的客戶的客戶 ID 值。



## 管理儲存查詢

儲存查詢提供方式來儲存經常執行的查詢。儲存查詢可以是供所有使用者使用的查詢 (公用查詢)，或屬於特定使用者的查詢 (專用查詢)。

儲存查詢是儲存在資料庫中，且依名稱識別的查詢。專用及公用儲存查詢的名稱可以相同，而來自不同擁有者的專用儲存查詢的名稱也可以相同。

您可以擁有商業程序物件、作業物件或這兩種物件類型組合的儲存查詢。

### 管理公用儲存查詢

公用儲存查詢是由系統管理者所建立。所有使用者都可以使用這些查詢。

身為系統管理者，您可以建立、檢視及刪除公用儲存查詢。如果 API 呼叫中未指定使用者 ID，則會假設該儲存查詢就是公用儲存查詢。

1. 建立公用儲存查詢。

例如，下列程式碼片段會建立程序實例的儲存查詢，並將它儲存為名稱 CustomerOrdersStartingWithA。

```
process.createStoredQuery("CustomerOrdersStartingWithA",
    "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
    "PROCESS_INSTANCE.NAME LIKE 'A%'",
    "PROCESS_INSTANCE.NAME",
    (Integer)null, (TimeZone)null);
```

儲存查詢的結果是排序過且開頭為字母 A 的所有程序實例名稱，以及其關聯程序實例 ID (PIID) 的清單。

2. 執行儲存查詢所定義的查詢。

```
QueryResultSet result = process.query("CustomerOrdersStartingWithA",
    new Integer(0));
```

此動作會傳回滿足準則的物件。在此情況下，是所有開頭為 A 的客戶訂單。

3. 列出可用公用儲存查詢的名稱。

下列程式碼片段顯示如何將傳回的查詢清單限制為只限公用查詢。

```
String[] storedQuery = process.getStoredQueryNames(StoredQueryData.KIND_PUBLIC);
```

4. 選擇性的: 檢查特定儲存查詢所定義的查詢。

儲存專用查詢與儲存公用查詢兩者的名稱可以相同。如果這些名稱相同，則會傳回專用儲存查詢。下列程式碼片段顯示如何只傳回具有指定名稱的公用查詢。如果想要針對作業型物件執行此查詢，請指定 StoredQuery 作為傳回的物件類型，而不是 StoredQueryData。

```
StoredQueryData storedQuery = process.getStoredQuery
    (StoredQueryData.KIND_PUBLIC, "CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();
String owner = storedQuery.getOwner();
```

5. 刪除公用儲存查詢。

下列程式碼片段顯示如何刪除在步驟 1 中建立的儲存查詢。

```
process.deleteStoredQuery("CustomerOrdersStartingWithA");
```

## 管理其他使用者的專用儲存查詢

任何使用者都可以建立專用查詢。這些查詢只適用於查詢擁有者及系統管理者。

身為系統管理者，您可以管理屬於特定使用者的專用儲存查詢。

1. 建立使用者 ID Smith 的專用儲存查詢。

例如，下列程式碼片段會建立程序實例的儲存查詢，並以使用者 ID Smith 的名稱 CustomerOrdersStartingWithA 予以儲存。

```
process.createStoredQuery("Smith", "CustomerOrdersStartingWithA",
    "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
    "PROCESS_INSTANCE.NAME LIKE 'A%'",
    "PROCESS_INSTANCE.NAME",
    (Integer)null, (TimeZone)null,
    (List)null, (String)null);
```

儲存查詢的結果是排序過且開頭為字母 A 的所有程序實例名稱，以及其關聯程序實例 ID (PIID) 的清單。

2. 執行儲存查詢所定義的查詢。

```
QueryResultSet result = process.query
    ("Smith", "CustomerOrdersStartingWithA",
    (Integer)null, (Integer)null, (List)null);
new Integer(0));
```

此動作會傳回滿足準則的物件。在此情況下，是所有開頭為 A 的客戶訂單。

3. 取得屬於特定使用者的專用查詢名稱清單。

例如，下列程式碼片段顯示如何取得屬於使用者 Smith 的專用查詢清單。

```
String[] storedQuery = process.getStoredQueryNames("Smith");
```

4. 檢視特定查詢的明細。

下列程式碼片段顯示如何檢視使用者 Smith 所擁有 CustomerOrdersStartingWithA 查詢的明細。

```
StoredQuery storedQuery = process.getStoredQuery
    ("Smith", "CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();
String owner = storedQuery.getOwner();
```

5. 刪除專用儲存查詢。

下列程式碼片段顯示如何刪除使用者 Smith 所擁有的專用查詢。

```
process.deleteStoredQuery("Smith", "CustomerOrdersStartingWithA");
```

## 使用專用儲存查詢

如果您不是系統管理者，則可以建立、執行及刪除您自己的專用儲存查詢。您也可以使用系統管理者建立的公用儲存查詢。

1. 建立專用儲存查詢。

例如，下列程式碼片段會建立程序實例的儲存查詢，並將它儲存為特定名稱。如果未指定使用者 ID，則假設儲存查詢是已登入使用者的專用儲存查詢。

```
process.createStoredQuery("CustomerOrdersStartingWithA",
    "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
    "PROCESS_INSTANCE.NAME LIKE 'A%'",
    "PROCESS_INSTANCE.NAME",
    (Integer)null, (TimeZone)null);
```

此查詢會傳回排序過且開頭字母為 A 的所有程序實例名稱，以及其相關聯程序實例 ID (PIID) 的清單。

2. 執行儲存查詢所定義的查詢。

```
QueryResultSet result = process.query("CustomerOrdersStartingWithA",
    new Integer(0));
```

此動作會傳回滿足準則的物件。在此情況下，是所有開頭為 A 的客戶訂單。

3. 取得已登入使用者可以存取的儲存查詢名稱清單。

下列程式碼片段顯示如何取得使用者可以存取的公用及專用儲存查詢。

```
String[] storedQuery = process.getStoredQueryNames();
```

4. 檢視特定查詢的明細。

下列程式碼片段顯示如何檢視使用者 Smith 所擁有 CustomerOrdersStartingWithA 查詢的明細。

```
StoredQuery storedQuery = process.getStoredQuery
    ("CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();
String owner = storedQuery.getOwner();
```

5. 刪除專用儲存查詢。

下列程式碼片段顯示如何刪除專用儲存查詢。

```
process.deleteStoredQuery("CustomerOrdersStartingWithA");
```

---

## 開發商業程序的應用程式

商業程序是一組商業相關活動，以特定的順序進行呼叫，來達成商業目標。提供的範例會顯示如何開發適用於程序一般動作的應用程式。

商業程序可以是微流程或長時間執行的程序：

- 微流程是同步執行的短時間執行的商業程序。會在極短的時間之後，將結果傳回給呼叫者。
- 長時間執行的可岔斷程序是以鏈結在一起的一連串活動來執行。在程序中使用特定建構，會導致程序流程中的岔斷，例如，呼叫人力作業、使用同步連結來呼叫服務，或使用計時器驅動的活動。

通常會非同步地導覽程序的平行分支，亦即，並行執行平行分支中的活動。根據活動的類型及交易設定，活動可以在本身的交易中執行。

### 程序實例上動作所需的角色

存取 BusinessFlowManager 介面無法保證呼叫者可以對程序執行所有動作。呼叫者必須登入用戶端應用程式，並具有已授權執行動作的角色。

下表顯示特定角色可對程序實例採取的動作。

動作	呼叫者的主體角色		
	讀者	起始者	管理者
createMessage	x	x	x
createWorkItem			x
delete			x
deleteWorkItem			x
forceTerminate			x
getActiveEventHandlers	x		x
getActivityInstance	x		x
getAllActivities	x		x
getAllWorkItems	x		x
getClientUISettings	x	x	x
getCustomProperties	x	x	x
getCustomProperty	x	x	x
getCustomPropertyNames	x	x	x
getFaultMessage	x	x	x
getInputClientUISettings	x	x	x
getInputMessage	x	x	x
getOutputClientUISettings	x	x	x
getOutputMessage	x	x	x
getProcessInstance	x	x	x
getVariable	x	x	x
getWaitingActivities	x	x	x
getWorkItems	x		x
restart			x
resume			x
setCustomProperty		x	x
setVariable			x
suspend			x
transferWorkItem			x

### 商業程序活動上動作所需的角色

存取 BusinessFlowManager 介面無法保證呼叫者可以對活動執行所有動作。呼叫者必須登入用戶端應用程式，並具有已授權執行動作的角色。

下表顯示特定角色可對活動實例採取的動作。

動作	呼叫者的主體角色				
	讀者	編輯者	可能的擁有者	擁有者	管理者
cancelClaim				x	x
claim			x		x

動作	呼叫者的主體角色				
	讀者	編輯者	可能的擁有者	擁有者	管理者
complete				x	x
createMessage	x	x	x	x	x
createWorkItem					x
deleteWorkItem					x
forceComplete					x
forceRetry					x
getActivityInstance	x	x	x	x	x
getAllWorkItems	x	x	x	x	x
getClientUISettings	x	x	x	x	x
getCustomProperties	x	x	x	x	x
getCustomProperty	x	x	x	x	x
getCustomPropertyNames	x	x	x	x	x
getFaultMessage	x	x	x	x	x
getFaultNames	x	x	x	x	x
getInputMessage	x	x	x	x	x
getOutputMessage	x	x	x	x	x
getVariable	x	x	x	x	x
getVariableNames	x	x	x	x	x
getInputVariableNames	x	x	x	x	x
getOutputVariableNames	x	x	x	x	x
getWorkItems	x	x	x	x	x
setCustomProperty		x		x	x
setFaultMessage		x		x	x
setOutputMessage		x		x	x
setVariable					x
transferWorkItem				x 僅針對可能的 擁有者或管理 者	x

## 管理商業程序的生命週期

呼叫可以啟動程序的「商業程序編排器」API 方法時，就會有程序實例。在程序實例的所有活動都處於結束狀態之前，會繼續導覽程序實例。您可以對程序實例採取各種動作，以管理它的生命週期。

提供的範例會顯示如何開發適用於下列程序的一般生命週期動作的應用程式。

### 啟動商業程序

啟動商業程序的方式是根據程序是微流程或長時間執行的程序而定。啟動程序的服務對啟動程序的方式而言也十分重要；程序可以只具有一個唯一的啟動服務或數個啟動服務。

提供的範例會顯示如何開發適用於啟動微流程及長時間執行程序的一般範例情況的應用程式。

### 執行包含唯一啟動服務程式的微流程:

微流程可以透過接收活動或挑選活動予以啟動。如果微流程是使用接收活動啟動，或挑選活動只有一個 `onMessage` 定義時，則啟動服務會是唯一的。

如果微流程實作要求/回應作業 (即程序包含回覆)，則可以使用 `call` 方法來執行在呼叫中傳遞程序範本名稱作為參數的程序。

如果微流程是單向作業，則請使用 `sendMessage` 方法來執行程序。此範例未涵蓋此方法。

1. 選擇性的: 列出程序範本，以尋找想要執行的程序名稱。

如果已知道程序名稱，則這是選用步驟。

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
PROCESS_TEMPLATE.EXECUTION_MODE.EXECUTION_MODE_MICROFLOW",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);
```

結果會依名稱排序。此查詢傳回的陣列會包含前 50 個排序過而且可以透過 `call` 方法啟動的範本。

2. 啟動具有適當類型的輸入訊息的程序。

在您建立訊息時，必須指定訊息的訊息類型名稱，以包含訊息定義。

```
ProcessTemplateData template = processTemplates[0];
//create a message for the single starting receive activity
ClientObjectWrapper input = process.createMessage
(template.getID(),
template.getInputMessageType());
DataObject myMessage = null;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
myMessage = (DataObject)input.getObject();
//set the strings in the message, for example, a customer name
myMessage.setString("CustomerName", "Smith");
}

//run the process
ClientObjectWrapper output = process.call(template.getName(), input);
DataObject myOutput = null;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
myOutput = (DataObject)output.getObject();
int order = myOutput.getInt("OrderNo");
}
```

此動作會建立程序範本的實例 (CustomerTemplate)，並傳遞一些客戶資料。此作業只有在程序完成時才會傳回。程序的結果 (OrderNo) 則會傳回給呼叫者。

### 執行包含非唯一啟動服務的微流程:

微流程可以透過接收活動或挑選活動予以啟動。如果使用具有多個 `onMessage` 定義的挑選活動啟動微流程，則啟動服務不是唯一的。

如果微流程實作要求/回應作業 (即程序包含回覆)，則可以使用 `call` 方法來執行在呼叫中傳遞啟動服務 ID 的程序。

如果微流程是單向作業，則請使用 `sendMessage` 方法來執行程序。此範例未涵蓋此方法。

1. 選擇性的: 列出程序範本，以尋找想要執行的程序名稱。

如果已知道程序名稱，則這是選用步驟。

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
    PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_MICROFLOW",
"PROCESS_TEMPLATE.NAME",
    new Integer(50),
    (TimeZone)null);
```

結果會依名稱排序。此查詢傳回的陣列會包含前 50 個排序過而且可以啟動為微流程的範本。

2. 判斷要呼叫的啟動服務。

此範例會使用找到的第一個範本。

```
ProcessTemplateData template = processTemplates[0];
ActivityServiceTemplateData[] startActivities =
    process.getStartActivities(template.getID());
```

3. 啟動具有適當類型的輸入訊息的程序。

在您建立訊息時，必須指定訊息的訊息類型名稱，以包含訊息定義。

```
ActivityServiceTemplateData activity = startActivities[0];
//create a message for the service to be called
ClientObjectWrapper input =
    process.createMessage(activity.getServiceTemplateID(),
        activity.getActivityTemplateID(),
        activity.getInputMessageType());
DataObject myMessage = null;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)input.getObject();
    //set the strings in the message, for example, a customer name
    myMessage.setString("CustomerName", "Smith");
}
//run the process
ClientObjectWrapper output = process.call(activity.getServiceTemplateID(),
        activity.getActivityTemplateID(),
        input);
//check the output of the process, for example, an order number
DataObject myOutput = null;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myOutput = (DataObject)output.getObject();
    int order = myOutput.getInt("OrderNo");
}
```

此動作會建立程序範本的實例 (CustomerTemplate)，並傳遞一些客戶資料。此作業只有在程序完成時才會傳回。程序的結果 (OrderNo) 則會傳回給呼叫者。

**啟動內含唯一啟動服務的長時間執执行程序:**



如果啟動服務是唯一的，則可以使用 `initiate` 方法，並將程序範本名稱當作參數來傳遞。使用單一接收或挑選活動啟動長時間執行程序時，以及單一挑選活動只有一個 `onMessage` 定義時，都是這種情況。

1. 選擇性的: 列出程序範本，以尋找想要啟動的程序名稱。

如果已知道程序名稱，則這是選用步驟。

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_LONG_RUNNING",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);
```

結果會依名稱排序。此查詢傳回的陣列會包含前 50 個排序過而且可以透過 `initiate` 方法啟動的範本。

2. 啟動具有適當類型的輸入訊息的程序。

在您建立訊息時，必須指定訊息的訊息類型名稱，以包含訊息定義。如果指定程序實例名稱，則它的開頭不得為底線。如果未指定程序實例名稱，則會使用「字串」格式的程序實例 ID (PIID) 作為名稱。

```
ProcessTemplateData template = processTemplates[0];
//create a message for the single starting receive activity
ClientObjectWrapper input = process.createMessage
(template.getID(),
template.getInputMessageType());
DataObject myMessage = null;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
myMessage = (DataObject)input.getObject();
//set the strings in the message, for example, a customer name
myMessage.setString("CustomerName", "Smith");
}
//start the process
PIID piid = process.initiate(template.getName(), "CustomerOrder", input);
```

此動作會建立實例 (`CustomerOrder`)，並傳遞一些客戶資料。啟動程序時，作業會將新程序實例的物件 ID 傳回給呼叫者。

程序實例起始者會設為要求的呼叫者。此人員會接收程序實例的工作項目。會判斷程序實例的程序管理者、讀者及編輯者，而且會接收程序實例的工作項目。也會判斷後續活動實例。這些都會自動啟動，或者，如果它們是人力作業、接收或挑選活動，則會為可能的擁有者建立工作項目。

#### 啟動內含非唯一啟動服務的長時間執行程序:

長時間執行程序可以透過多個起始接收或挑選活動來啟動。您可以使用 `initiate` 方法來啟動程序。如果啟動服務不是唯一的 (例如，使用多個接收或挑選活動，或具有多個 `onMessage` 定義的挑選活動來啟動程序)，則必須識別要呼叫的服務。

1. 選擇性的: 列出程序範本，以尋找想要啟動的程序名稱。

如果已知道程序名稱，則這是選用步驟。

```

ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_LONG_RUNNING",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);

```

結果會依名稱排序。此查詢傳回的陣列會包含前 50 個排序過而且可以啟動為長時間執行程序的範本。

2. 判斷要呼叫的啟動服務。

```

ProcessTemplateData template = processTemplates[0];
ActivityServiceTemplateData[] startActivities =
process.getStartActivities(template.getID());

```

3. 啟動具有適當類型的輸入訊息的程序。

在您建立訊息時，必須指定訊息的訊息類型名稱，以包含訊息定義。如果指定程序實例名稱，則它的開頭不得為底線。如果未指定程序實例名稱，則會使用「字串」格式的程序實例 ID (PIID) 作為名稱。

```

ActivityServiceTemplateData activity = startActivities[0];
//create a message for the service to be called
ClientObjectWrapper input = process.createMessage
(activity.getServiceTemplateID(),
activity.getActivityTemplateID(),
activity.getInputMessageType());

DataObject myMessage = null;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
myMessage = (DataObject)input.getObject();
//set the strings in the message, for example, a customer name
myMessage.setString("CustomerName", "Smith");
}
//start the process
PIID piid = process.sendMessage(activity.getServiceTemplateID(),
activity.getActivityTemplateID(),
input);

```

此動作會建立實例，並傳遞一些客戶資料。啟動程序時，作業會將新程序實例的物件 ID 傳回給呼叫者。

程序實例起始者會設為要求的呼叫者，而且會接收程序實例的工作項目。會判斷程序實例的程序管理者、讀者及編輯者，而且會接收程序實例的工作項目。也會判斷後續活動實例。這些都會自動啟動，或者，如果它們是人力作業、接收或挑選活動，則會為可能的擁有者建立工作項目。

## 暫停及回復商業程序

在長時間執行的最上層程序實例執行時，您可以將它暫停，並重新回復它以將其完成。

呼叫者必須是程序實例的管理者或商業程序管理者。若要暫停程序實例，則程序實例必須處於執行中或失敗中狀態。

例如，您可能會想要暫停程序實例，以配置稍後要在程序中使用的後端系統的存取。而在符合程序的必備項目時，就可以回復程序實例。您可能也會想要暫停程序以修正導致程序實例失敗的問題，然後在問題修正之後再回復該程序。

1. 取得想要暫停的執行中程序 (CustomerOrder)。

```
ProcessInstanceData processInstance =  
    process.getProcessInstance("CustomerOrder");
```

2. 暫停程序實例。

```
PIID piid = processInstance.getID();  
process.suspend( piid );
```

此動作會暫停指定的最上層程序實例。程序實例進入暫停狀態。如果 `autonomy` 屬性設為 `child` 的子程序處於執行中、失敗中、終止中或補償中狀態，則也會一併暫停它們。也會暫停與這個程序實例相關聯的行內作業；但不會暫停與這個程序實例相關聯的獨立式作業。

在這個狀態下，仍然可以完成啟動的活動，但不會再啟動任何新的活動，例如，可以完成處於已要求狀態的人力作業活動。

3. 回復程序實例。

```
process.resume( piid );
```

此動作會讓程序實例及其子程序進入它們在暫停之前的狀態。

## 重新啟動商業程序

您可以重新啟動處於已完成、已終止、已失敗或已補償狀態的程序實例。

呼叫者必須是程序實例的管理者或商業程序管理者。

重新啟動程序實例與第一次啟動程序實例類似。然而，在重新啟動程序實例時，已經知道程序實例 ID，而且也有實例的輸入訊息。

如果程序有多個可以建立程序實例的接收活動或挑選活動（也稱為接收選擇活動），則會使用所有屬於這些活動的訊息來重新啟動程序實例。如果上述任一活動實作要求/回應作業，則會在導覽相關聯的回覆活動時再次傳送回應。

1. 取得想要重新啟動的程序。

```
ProcessInstanceData processInstance =  
    process.getProcessInstance("CustomerOrder");
```

2. 重新啟動程序實例。

```
PIID piid = processInstance.getID();  
process.restart( piid );
```

此動作會重新啟動指定的程序實例。

## 終止程序實例

有時，擁有程序管理者權限的人員會需要終止已知處於無法復原狀態的最上層程序實例。因為會立即終止程序實例，而不會等待任何未執行的子程序或活動，所以只有在異常狀況下才應該終止程序實例。

1. 擷取要終止的程序實例。

```
ProcessInstanceData processInstance =  
    process.getProcessInstance("CustomerOrder");
```

2. 終止程序實例。

如果要終止程序實例，則可以在使用補償或不使用補償的情況下終止程序實例。

若要使用補償來終止程序實例：

```
PIID piid = processInstance.getID();
process.forceTerminate(piid, CompensationBehaviour.INVOKE_COMPENSATION);
```

若不要使用補償來終止程序實例：

```
PIID piid = processInstance.getID();
process.forceTerminate(piid);
```

如果使用補償來終止程序實例，則會執行程序的補償，就像已在最上層範圍上發生錯誤一般。如果您在沒有補償的情況下終止程序實例，則會立即終止程序實例，而不會等待活動、待辦作業或行內呼叫作業正常結束。

程序及獨立式作業所啟動的應用程式會與強制終止要求未終止的程序相關。如果要終止這些應用程式，則必須在程序應用程式中新增陳述式，以明確終止程序所啟動的應用程式。

## 刪除程序實例

如果已在程序模型中設定程序範本的對應內容，則會自動從「商業程序編排器」資料庫中刪除已完成的程序實例。您可能會想要將程序實例保留在資料庫中，例如，若要查詢未寫入至審核日誌的程序實例中的資料。然而，儲存程序實例資料不只會影響磁碟空間及效能，還會無法建立利用相同相互關係集值的程序實例。因此，您應該定期刪除資料庫中的程序實例資料。

若要刪除程序實例，則您需要有程序管理者權限，而且程序實例必須是最上層程序實例。

下列範例顯示如何刪除所有已完成的程序實例。

1. 列出已完成的程序實例。

```
QueryResultSet result =
    process.query("DISTINCT PROCESS_INSTANCE.PIID",
                 "PROCESS_INSTANCE.STATE =
                 PROCESS_INSTANCE.STATE.STATE_FINISHED",
                 (String)null, (Integer)null, (TimeZone)null);
```

此動作會傳回列出已完成程序實例的查詢結果集。

2. 刪除已完成的程序實例。

```
while (result.next() )
{
    PIID piid = (PIID) result.getOID(1);
    process.delete(piid);
}
```

此動作會從資料庫中刪除選取的程序實例及其行內作業。

## 處理人力作業活動

商業程序中的人力作業活動會透過工作項目指派給組織中的各種人員。啟動程序時，會為可能的擁有者建立工作項目。

當啟動人力作業活動時，會同時建立活動實例及相關聯的待辦作業。處理人力作業活動及工作項目管理會委派給「人力作業管理程式」。活動實例的任何狀態變更都會反映在作業實例中，反之亦然。

可能的擁有者會要求活動。此人員負責提供相關資訊並完成活動。

1. 列出屬於已登入人員而且已可以處理的活動：

```
QueryResultSet result =
    process.query("ACTIVITY.AIID",
        "ACTIVITY.STATE = ACTIVITY.STATE.STATE_READY AND
        ACTIVITY.KIND = ACTIVITY.KIND.KIND_STAFF AND
        WORK_ITEM.REASON =
        WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
        (String)null, (Integer)null, (TimeZone)null);
```

此動作會傳回查詢結果集，內含已登入人員可以處理的活動。

2. 要求要處理的活動：

```
if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
    ClientObjectWrapper input = process.claim(aaid);
    DataObject activityInput = null ;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        activityInput = (DataObject)input.getObject();
        // read the values
        ...
    }
}
```

要求活動時，會傳回活動的輸入訊息。

3. 完成活動的處理時，請完成活動。活動的完成狀態可以是順利完成，或者也可以具有錯誤訊息。如果活動順利完成，則會傳遞輸出訊息。如果活動失敗，則活動會進入已失敗或已停止狀態，而且會傳遞錯誤訊息。您必須建立這些動作的適當訊息。在您建立訊息時，必須指定訊息類型名稱，以包含訊息定義。
  - a. 若要順利完成活動，請建立輸出訊息。

```
ActivityInstanceData activity = process.getActivityInstance(aaid);
ClientObjectWrapper output =
    process.createMessage(aaid, activity.getOutputMessageType());
DataObject myMessage = null ;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myMessage = (DataObject)output.getObject();
    //set the parts in your message, for example, an order number
    myMessage.setInt("OrderNo", 4711);
}

//complete the activity
process.complete(aaid, output);
```

此動作會設定內含序號的輸出訊息。

- b. 若要在發生錯誤時完成活動，請建立錯誤訊息。

```
//retrieve the faults modeled for the human task activity
List faultNames = process.getFaultNames(aaid);

//create a message of the appropriate type
ClientObjectWrapper myFault =
    process.createMessage(aaid, faultNames.get(0));

// set the parts in your fault message, for example, an error number
DataObject myMessage = null ;
if ( myFault.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)myFault.getObject();
```

```

        //set the parts in the message, for example, a customer name
        myMessage.setInt("error",1304);
    }

    process.complete(aiid, (String)faultNames.get(0), myFault);

```

此動作會設定處於已失敗或已停止狀態的活動。如果程序模型中活動的 **continueOnError** 參數設為 **true**，則活動會進入已失敗狀態，而導覽會繼續。如果 **continueOnError** 參數設為 **false**，而且在周遭範圍未捕捉到錯誤，則活動會進入已停止狀態。在這種狀態下，可以使用強制完成或強制重試來修復活動。

## 處理單一人員工作流程

部分工作流程只由一個人執行 (例如，從線上書店訂購書籍)。這類型的工作流程沒有平行路徑。 **completeAndClaimSuccessor** API 支援處理這類型的工作流程。

在線上書店中，購買者需要完成一連串的動作才能訂購書籍。這一連串動作可以實作為一系列的人力作業活動 (待辦作業)。如果購買者決定要訂購數本書，則這等於要求下一個人力作業活動。因為使用者介面定義是與活動相關聯，用以控制使用者介面中的對話框流程，所以這種類型的工作流程也稱為 *頁面流程*。

**completeAndClaimSuccessor** API 會完成人力作業活動，並在已登入人員的相同程序實例中要求下一個人力作業活動。而且會傳回下一個已要求活動的相關資訊 (包含要處理的輸入訊息)。因為下一個活動可以在已完成的相同活動交易內使用，所以程序模型中所有人力作業活動的交易行為都必須設定為 **participates**。

將此範例與同時使用「商業流程管理程式」API 及「人力作業管理程式」API 的範例進行比較。

1. 要求一連串活動中的第一個活動。

```

//
//Query the list of activities that can be claimed by the logged-on user
//
QueryResultSet result =
    process.query("ACTIVITY.AIID",
                  "PROCESS_INSTANCE.NAME = 'CustomerOrder' AND
                   ACTIVITY.STATE = ACTIVITY.STATE.STATE_READY AND
                   ACTIVITY.KIND = ACTIVITY.KIND.KIND_STAFF AND
                   WORK_ITEM.REASON =
                     WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
                  (String)null, (Integer)null, (TimeZone)null);
...
//
//Claim the first activity
//
if (result.size() > 0)
{
    result.first();
    AIID aiid = (AIID) result.getOID(1);
    ClientObjectWrapper input = process.claim(aiid);
    DataObject activityInput = null ;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        activityInput = (DataObject)input.getObject();
        // read the values
        ...
    }
}
}

```

要求活動時，會傳回活動的輸入訊息。



2. 完成活動的處理時，請完成活動，並要求下一個活動。

若要完成活動，則會傳遞輸出訊息。在您建立輸出訊息時，必須指定訊息類型名稱，以包含訊息定義。

```
ActivityInstanceData activity = process.getActivityInstance(aiid);
ClientObjectWrapper output =
    process.createMessage(aiid, activity.getOutputMessageTypeName());
DataObject myMessage = null ;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myMessage = (DataObject)output.getObject();
    //set the parts in your message, for example, an order number
    myMessage.setInt("OrderNo", 4711);
}

//complete the activity and claim the next one
CompleteAndClaimSuccessorResult successor =
    process.completeAndClaimSuccessor(aiid, output);
```

這個動作會設定包含訂單號碼的輸出訊息，並要求序列中的下一個活動。如果將後續活動設定為 `AutoClaim`，而且如果有多個可以遵循的路徑，則會要求所有後續活動，並將隨機活動傳回為下一個活動。如果沒有其他後續活動可以指派給此使用者，則會傳回 `Null`。

如果程序包含可以遵循的平行路徑，而且這些路徑包含人力作業活動，其中已登入使用者可能是這些活動中多個活動的擁有者，則會自動要求隨機活動，並且該隨機活動會傳回為下一個活動。

3. 處理下一個活動。

```
String name = successor.getActivityName();

ClientObjectWrapper nextInput = successor.getInputMessage();
if ( nextInput.getObject() !=
    null && nextInput.getObject() instanceof DataObject )
{
    activityInput = (DataObject)input.getObject();
    // read the values
    ...
}

aiid = successor.getAIID();
```

4. 繼續步驟 2 完成活動。

## 將訊息傳送給等待中活動

您可以使用入埠訊息活動（接收活動、挑選活動中的 `onMessage`、事件處理程式中的 `onEvent`），來同步化執行中程序與「外界」的事件。例如，接收客戶回應資訊要求的電子郵件可能就是這類事件。

您可以使用起始作業，將訊息傳送給活動。

1. 列出等待來自己登入使用者的訊息的活動服務範本，而該使用者位於具有特定程序實例 ID 的程序實例中。

```
ActivityServiceTemplateData[] services = process.getWaitingActivities(piid);
```

2. 將訊息傳送給第一個等待服務。

假設第一個服務就是您想要提供的服務。呼叫者必須是接收訊息的潛在活動起始者，或程序實例的管理者。



```

VTID vtid = services[0].getServiceTemplateID();
ATID atid = services[0].getActivityTemplateID();
String inputType = services[0].getInputMessageTypeName();

// create a message for the service to be called
ClientObjectWrapper message =
    process.createMessage(vtid,atid,inputMessageTypeName);
DataObject myMessage = null;
if ( message.getObject() != null && message.getObject() instanceof DataObject )
{
    myMessage = (DataObject)message.getObject();
    //set the strings in the message, for example, chocolate is to be ordered
    myMessage.setString("Order", "chocolate");
}

// send the message to the waiting activity
process.sendMessage(vtid, atid, message);
}

```

此動作會將指定的訊息傳送給等待活動服務，並傳遞一些訂單資料。

您也可以指定程序實例 ID，確保訊息已傳送給指定的程序實例。如果未指定程序實例 ID，則會將訊息傳送給活動服務，以及訊息中透過相互關係值識別的程序實例。如果已指定程序實例 ID，則會檢查利用相互關係值找到的程序實例，確定該程序實例具有指定的程序實例 ID。

## 處理事件

整個商業程序及它的每個範圍都可以與發生關聯事件時呼叫的事件處理程式相關聯。接收或挑選活動的事件處理程式類似，其中，程序都可以使用事件處理程式來提供 Web 服務作業。

只要對應範圍在執行中，就可以不限次數呼叫事件處理程式。此外，可以同時啟動多個事件處理程式實例。

下列程式碼片段顯示如何取得給定程序實例的作用中事件處理程式，以及如何傳送輸入訊息。

1. 判斷程序實例 ID 的資料，並列出程序的作用中事件處理程式。

```

ProcessInstanceData processInstance =
    process.getProcessInstance( "CustomerOrder2711");
EventHandlerTemplateData[] events = process.getActiveEventHandlers(
    processInstance.getID() );

```

2. 傳送輸入訊息。

此範例會使用找到的第一個事件處理程式。

```

EventHandlerTemplateData event = null;
if ( events.length > 0 )
{
    event = events[0];

    // create a message for the service to be called
    ClientObjectWrapper input = process.createMessage(
        event.getID(), event.getInputMessageTypeName());

    if (input.getObject() != null && input.getObject() instanceof DataObject )
    {
        DataObject inputMessage = (DataObject)input.getObject();
        // set content of the message, for example, a customer name, order number
        inputMessage.setString("CustomerName", "Smith");
    }
}

```

```

        inputMessage.setString("OrderNo", "2711");

        // send the message
        process.sendMessage( event.getProcessTemplateName(),
                             event.getPortTypeNamespace(),
                             event.getPortTypeName(),
                             event.getOperationName(),
                             input );
    }
}

```

此動作會將指定的訊息傳送給程序的作用中事件處理程式。

## 分析程序結果

程序可以公開 Web 服務作業，而這些作業是塑型為「Web 服務說明語言 (WSDL)」單向或要求/回應作業。無法使用 `getOutputMessage` 方法，擷取具有單向介面的長時間執行程序的結果，因為程序沒有輸出。不過，您可以改為查詢變數的內容。

只有在衍生程序實例的來源程序範本未指定自動刪除衍生的程序實例時，才會將程序的結果儲存至資料庫中。

分析程序的結果 (例如，檢查序號)。

```

QueryResultSet result = process.query
    ("PROCESS_INSTANCE.PIID",
     "PROCESS_INSTANCE.NAME = 'CustomerOrder' AND
     PROCESS_INSTANCE.STATE =
     PROCESS_INSTANCE.STATE.STATE_FINISHED",
     (String)null, (Integer)null, (TimeZone)null);
if (result.size() > 0)
{
    result.first();
    PIID piid = (PIID) result.getOID(1);
    ClientObjectWrapper output = process.getOutputMessage(piid);
    DataObject myOutput = null;
    if ( output.getObject() != null && output.getObject() instanceof DataObject )
    {
        myOutput = (DataObject)output.getObject();
        int order = myOutput.getInt("OrderNo");
    }
}

```

## 修復活動

長時間執行的程序可以包含也會長時間執行的活動。這些活動可能會發現未捕捉到的錯誤，並進入已停止狀態。而執行中狀態的活動也會像「沒有回應」。在這兩種情況下，程序管理者可以用多種方式來處理活動，以便可以繼續導覽程序。

「商業程序編排器」API 提供 `forceRetry` 及 `forceComplete` 方法來修復活動。提供的範例會顯示如何將活動的修復動作新增至應用程式中。

## 強制完成活動

長時間執行程序中的活動有時會發現錯誤。如果錯誤處理程式未在含括範圍中捕捉到這些錯誤，而且相關聯的活動範本指定活動在發生錯誤時停止，則活動會進入已停止狀態，讓您可以對其進行修復。在此狀態中，您可以強制完成活動。

例如，如果活動未回應，則也可以強制完成處於執行中狀態的活動。

特定類型的活動會有額外的需求。

### 人力作業活動

您可以在強制完成呼叫中傳遞參數 (例如應該已經送出的訊息或應該已經引發的錯誤)。

### Script 活動

您不可以在強制完成呼叫中傳遞參數。然而，您必須設定需要修復的變數。

### 呼叫活動

您也可以強制完成處於執行中狀態的呼叫活動，而這些活動會呼叫不是子程序的非同步服務。例如，如果呼叫非同步服務而且該服務未回應，則您可能會想要這麼做。

1. 列出處於已停止狀態的停止活動。

```
QueryResultSet result =
    process.query("DISTINCT ACTIVITY.AIID",
        "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
        PROCESS_INSTANCE.NAME='CustomerOrder'",
        (String)null, (Integer)null, (TimeZone)null);
```

此動作會傳回 CustomerOrder 程序實例的已停止活動。

2. 完成活動 (例如，已停止的人力作業活動)。

在此範例中，會傳遞輸出訊息。

```
if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
    ActivityInstanceData activity = process.getActivityInstance(aaid);
    ClientObjectWrapper output =
        process.createMessage(aaid, activity.getOutputMessageType());
    DataObject myMessage = null;
    if ( output.getObject() != null && output.getObject() instanceof DataObject )
    {
        myMessage = (DataObject)output.getObject();
        //set the parts in your message, for example, an order number
        myMessage.setInt("OrderNo", 4711);
    }

    boolean continueOnError = true;
    process.forceComplete(aaid, output, continueOnError);
}
```

此動作會完成活動。如果發生錯誤，則 **continueOnError** 參數會判斷在 forceComplete 要求發生錯誤時要採取的動作。

在此範例中，**continueOnError** 為 true。此值表示如果發生錯誤，則活動會進入已失敗狀態。在處理此錯誤或到達程序範圍之前，會將此錯誤傳送到活動的含括範圍。此程序接著會進入失敗中狀態，而且最後會變成已失敗狀態。

### 重試執行已停止的活動

如果長時間執行程序中的活動在含括範圍內發現未捕捉到的錯誤，而且如果相關聯的活動範本指定活動在發生錯誤時停止，則活動會進入已停止狀態，讓您可以對其進行修復。您可以重試執行活動。

您可以設定活動所使用的變數。除了 Script 活動以外，還可以在強制重試呼叫中傳遞參數（例如活動需要的訊息）。

1. 列出已停止的活動。

```
QueryResultSet result =
    process.query("DISTINCT ACTIVITY.AIID",
        "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
        PROCESS_INSTANCE.NAME='CustomerOrder'",
        (String)null, (Integer)null, (TimeZone)null);
```

此動作會傳回 CustomerOrder 程序實例的已停止活動。

2. 重試執行活動（例如，已停止的人力作業活動）。

```
if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
    ActivityInstanceData activity = process.getActivityInstance(aaid);
    ClientObjectWrapper input =
        process.createMessage(aaid, activity.getOutputMessageType());
    DataObject myMessage = null;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        myMessage = (DataObject)input.getObject();
        //set the strings in your message, for example, chocolate is to be ordered
        myMessage.setString("OrderNo", "chocolate");
    }

    boolean continueOnError = true;
    process.forceRetry(aaid, input, continueOnError);
}
```

此動作會重試活動。如果發生錯誤，則 **continueOnError** 參數會判斷在處理 **forceRetry** 要求期間發生錯誤時要採取的動作。

在此範例中，**continueOnError** 為 **true**。這表示如果在處理 **forceRetry** 要求期間發生錯誤，則活動會進入已失敗狀態。在處理此錯誤或到達程序範圍之前，會將此錯誤傳送到活動的含括範圍。這個程序接著會進入失敗中狀態，而且在程序狀態以失敗狀態結束之前，會執行程序層次上的錯誤處理程式。

## BusinessFlowManagerService 介面

BusinessFlowManagerService 介面公開用戶端應用程式可以呼叫的商業程序功能。

BusinessFlowManagerService 介面可以呼叫的方法視程序的狀態，或者使用包含該方法之應用程式的人員權限而定。此處列出操作商業程序物件的主要方法。如需這些方法及 BusinessFlowManagerService 介面中其他可用方法的相關資訊，請參閱 `com.ibm.bpe.api` 套件中的 Javadoc。

### 程序範本

程序範本是一種包含商業程序規格，且已版本化、已部署並已安裝的程序模型。您可以發出適當的要求（例如 `sendMessage()`），以實例化及啟動它。伺服器會自動驅動程序實例的執行。

表 38. 程序範本的 API 方法

方法	說明
<code>getProcessTemplate</code>	擷取指定的程序範本。

表 38. 程序範本的 API 方法 (繼續)

方法	說明
queryProcessTemplates	擷取資料庫中所儲存的程序範本。

## 程序實例

下列是與啟動程序實例相關的 API 方法。

表 39. 與啟動程序實例相關的 API 方法

方法	說明
call	建立及執行微流程。
callWithReplyContext	從指定的程序範本使用唯一的啟動服務建立及執行微流程，或使用唯一的啟動服務建立及執行長時間執行的程序。呼叫會非同步地等待結果。
callWithUISettings	建立及執行微流程，並傳回輸出訊息及用戶端使用者介面 (UI) 設定。
initiate	建立程序實例及起始程序實例的處理。針對長時間執行的程序使用此方法。您也可以針對想要激發及忽略的微流程使用此方法。
sendMessage	傳送指定的訊息到指定的活動服務及程序實例。如果不存在具有相同的相互關係集值的程序實例，則會建立它。程序可以有唯一或非唯一的啟動服務。
getStartActivities	傳回可從指定的程序範本啟動程序實例的活動的相關資訊。
getActivityServiceTemplate	擷取指定的活動服務範本。

表 40. 控制程序實例生命週期的 API 方法

方法	說明
suspend	當長時間執行的最上層程序實例處於執行中或失敗中狀態時，將其暫停執行。
resume	回復執行長時間執行的最上層程序實例，其處於已暫停狀態。
restart	重新啟動長時間執行的最上層程序實例，其處於已完成、失敗或終止狀態。
forceTerminate	終止指定的最上層程序實例、其具有子項自主權的子程序，以及其執行中、已要求或等待中的活動。
delete	刪除指定的最上層程序實例，以及其具有子項自主權的子程序。
query	從符合搜尋準則的資料庫中擷取內容。

## 活動

若為呼叫活動，則您可以在程序模型中指定這些活動繼續處於錯誤狀況。如果 `continueOnError` 旗標設為 `false` 且發生無法處理的錯誤，則活動會進入已停止狀態。然

後，程序管理者可以修復活動。例如，`continueOnError` 旗標及相關聯的修復函數可以在長時間執行的程序中使用，其中呼叫活動會不定期失敗，但是模型補償及錯誤處理需要投入太多精力。

下列方法可用來處理及修復活動。

表 41. 控制活動實例生命週期的 API 方法

方法	說明
<code>claim</code>	要求備妥的活動實例，以供使用者處理活動。
<code>cancelClaim</code>	取消活動實例的要求。
<code>complete</code>	完成活動實例。
<code>completeAndClaimSuccessor</code>	完成活動實例，並在相同的程序實例中針對已登入的人員要求下一個活動實例。
<code>forceComplete</code>	強制完成處於執行中或已停止狀態的活動實例。
<code>forceRetry</code>	強制重複處於執行中或已停止狀態的活動實例。
<code>query</code>	從符合搜尋準則的資料庫中擷取內容。

## 變數及自訂內容

該介面提供 `get` 及 `set` 方法來擷取及設定變數的值。您還可以將指定內容與程序及活動實例相關聯，並從其中擷取指定內容。自訂內容名稱及值必須為 `java.lang.String` 類型。

表 42. 變數及自訂內容的 API 方法

方法	說明
<code>getVariable</code>	擷取指定的變數。
<code>setVariable</code>	設定指定的變數。
<code>getCustomProperty</code>	擷取指定活動或程序實例的指定自訂內容。
<code>getCustomProperties</code>	擷取指定活動或程序實例的自訂內容。
<code>getCustomPropertyNames</code>	擷取指定活動或程序實例的自訂內容名稱。
<code>setCustomProperty</code>	儲存指定活動或程序實例的自訂特有值。

## 開發人力作業的應用程式

作業是元件用來將人力作業呼叫為服務或人力作業用來呼叫服務的方法。提供有人力作業的一般應用程式範例。

如需「人力作業管理程式 API」的相關資訊，請參閱 `com.ibm.task.api` 套件中的 Javadoc。

### 啟動呼叫同步化介面的呼叫作業

呼叫作業與「服務元件架構 (SCA)」元件相關聯。作業在啟動時會呼叫 SCA 元件。只有在可以同步呼叫相關聯的 SCA 元件時，才會同步啟動呼叫作業。

例如，這類 SCA 元件可以實作為微流程或簡式 Java 類別。

此範例情況會建立作業範本實例，並傳遞一些客戶資料。而在傳回雙向作業之前，作業會保留執行中狀態。作業結果 (OrderNo) 則會傳回給呼叫者。

1. 選擇性的: 列出作業範本，以尋找想要執行的呼叫作業名稱。

如果已知道作業名稱，則這是選用步驟。

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates
("TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_ORIGINATING",
 "TASK_TEMPL.NAME",
 new Integer(50),
 (TimeZone)null);
```

結果會依名稱排序。此查詢傳回的陣列會包含前 50 個排序過的起始範本。

2. 建立適當類型的輸入訊息。

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage( template.getID());
DataObject myMessage = null ;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)input.getObject();
    //set the parts in the message, for example, a customer name
    myMessage.setString("CustomerName", "Smith");
}
```

3. 建立並同步執行作業。

若要同步執行作業，則其必須是雙向作業。此範例會使用 `createAndCallTask` 方法來建立並執行作業。

```
ClientObjectWrapper output = task.createAndCallTask( template.getName(),
                                                    template.getNamespace(),
                                                    input);
```

4. 分析作業的結果。

```
DataObject myOutput = null;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myOutput = (DataObject)output.getObject();
    int order = myOutput.getInt("OrderNo");
}
```

## 啟動呼叫非同步化介面的呼叫作業

呼叫作業與「服務元件架構 (SCA)」元件相關聯。作業在啟動時會呼叫 SCA 元件。只有在可以非同步呼叫相關聯的 SCA 元件時，才會非同步啟動呼叫作業。

例如，這類 SCA 元件可以實作為長時間執行的程序或單向作業。

此範例情況會建立作業範本實例，並傳遞一些客戶資料。

1. 選擇性的: 列出作業範本，以尋找想要執行的呼叫作業名稱。

如果已知道作業名稱，則這是選用步驟。

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates
("TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_ORIGINATING",
 "TASK_TEMPL.NAME",
 new Integer(50),
 (TimeZone)null);
```



結果會依名稱排序。此查詢傳回的陣列會包含前 50 個排序過的起始範本。

2. 建立適當類型的輸入訊息。

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage( template.getID());
DataObject myMessage = null ;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)input.getObject();
    //set the parts in the message, for example, a customer name
    myMessage.setString("CustomerName", "Smith");
}
```

3. 建立並以非同步執行作業。

此範例會使用 `createAndStartTask` 方法來建立並執行作業。

```
task.createAndStartTask( template.getName(),
                        template.getNamespace(),
                        input,
                        (ReplyHandlerWrapper)null);
```

## 建立及啓動作業實例

這個範例情況顯示如何建立用來定義協同作業 (在 API 中也稱為人力作業) 的作業範本實例，以及如何啓動作業實例。

1. 選擇性的: 列出作業範本，以尋找想要執行的協同作業名稱。

如果已知道作業名稱，則這是選用步驟。

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates
("TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_HUMAN",
 "TASK_TEMPL.NAME",
 new Integer(50),
 (TimeZone)null);
```

結果會依名稱排序。這個查詢傳回的陣列會包含前 50 個排序過的作業範本。

2. 建立適當類型的輸入訊息。

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage( template.getID());
DataObject myMessage = null ;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)input.getObject();
    //set the parts in the message, for example, a customer name
    myMessage.setString("CustomerName", "Smith");
}
```

3. 建立並啓動協同作業；這個範例未指定回覆處理程式。

此範例會使用 `createAndStartTask` 方法來建立並啓動作業。

```
TKIID tkiid = task.createAndStartTask( template.getName(),
                                       template.getNamespace(),
                                       input,
                                       (ReplyHandlerWrapper)null);
```

會針對與作業實例相關的人員建立工作項目。例如，可能的擁有者可以要求新的作業實例。

#### 4. 要求作業實例。

```
ClientObjectWrapper input2 = task.claim(tkiid);
DataObject taskInput = null ;
if ( input2.getObject() != null && input2.getObject() instanceof DataObject )
{
    taskInput = (DataObject)input2.getObject();
    // read the values
    ...
}
```

要求作業實例時，會傳回作業的輸入訊息。

## 處理待辦作業或協同作業

待辦作業 (在 API 中也稱為參與作業) 或協同作業 (在 API 中也稱為人力作業) 是透過工作項目指派給您組織中的各個人員。例如，當程序導覽至人力作業活動時，就會建立待辦作業及其相關聯的工作項目。

其中一位可能的擁有者會要求與工作項目相關聯的作業。此人員負責提供相關資訊並完成作業。

#### 1. 列出屬於已登入人員而且已可以處理的作業。

```
QueryResultSet result =
    task.query("TASK.TKIID",
              "TASK.STATE = TASK.STATE.STATE_READY AND
              (TASK.KIND = TASK.KIND.KIND_PARTICIPATING OR
              TASK.KIND = TASK.KIND.KIND_HUMAN)AND
              WORK_ITEM.REASON =
              WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
              (String)null, (Integer)null, (TimeZone)null);
```

此動作會傳回查詢結果集，內含已登入人員可以處理的作業。

#### 2. 要求要處理的作業。

```
if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    ClientObjectWrapper input = task.claim(tkiid);
    DataObject taskInput = null ;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        taskInput = (DataObject)input.getObject();
        // read the values
        ...
    }
}
```

要求作業時，會傳回作業的輸入訊息。

#### 3. 完成作業的處理時，請完成作業。

作業的完成狀態可以是順利完成，或者也可以具有錯誤訊息。如果作業順利完成，則會傳遞輸出訊息。如果作業失敗，則會傳遞錯誤訊息。您必須建立這些動作的適當訊息。

#### a. 若要順利完成作業，請建立輸出訊息。

```
ClientObjectWrapper output =
    task.createOutputMessage(tkiid);
DataObject myMessage = null ;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
```

```

{
    myMessage = (DataObject)output.getObject();
    //set the parts in your message, for example, an order number
    myMessage.setInt("OrderNo", 4711);
}

//complete the task
task.complete(tkiid, output);

```

此動作會設定內含序號的輸出訊息。作業會進入已完成狀態。

- b. 若要在發生錯誤時完成作業，請建立錯誤訊息。

```

//retrieve the faults modeled for the task
List faultNames = task.getFaultNames(tkiid);

//create a message of the appropriate type
ClientObjectWrapper myFault =
    task.createFaultMessage(tkiid, (String)faultNames.get(0));

// set the parts in your fault message, for example, an error number
DataObject myMessage = null ;
if ( myFault.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)myFault.getObject();
    //set the parts in the message, for example, a customer name
    myMessage.setInt("error",1304);
}

task.complete(tkiid, (String)faultNames.get(0), myFault);

```

此動作會設定內含錯誤碼的錯誤訊息。作業會進入已失敗狀態。

## 暫停及回復作業實例

您可以暫停協同作業實例 (在 API 中也稱為人力作業) 或待辦作業實例 (在 API 中也稱為參與作業)。

作業實例可以處於備妥或已要求狀態。也可以進行呈報。呼叫者必須是作業實例的擁有者、發送者或管理者。

您可以在作業實例執行時將它暫停。例如，您可能會想要做這樣的處理，以收集完成作業所需的資訊。資訊可用時，就可以回復作業實例。

1. 取得已登入使用者所要求的作業清單。

```

QueryResultSet result = task.query("DISTINCT TASK.TKIID",
                                   "TASK.STATE = TASK.STATE.STATE_CLAIMED",
                                   (String)null,
                                   (Integer)null,
                                   (TimeZone)null);

```

此動作會傳回查詢結果集，內含已登入使用者所要求的作業清單。

2. 暫停作業實例。

```

if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    task.suspend(tkiid);
}

```

此動作會暫停指定的作業實例。作業實例會進入已暫停狀態。

3. 回復程序實例。

```
task.resume( tkiid );
```

此動作會讓作業實例進入它暫停之前的狀態。

## 分析作業結果

待辦作業 (在 API 中也稱為參與作業) 或協同作業 (在 API 中也稱為人力作業) 非同步執行。如果在啓動作業時指定回覆處理程式，則會在作業完成時自動傳回輸出訊息。如果未指定回覆處理程式，則必須明確地擷取訊息。

只有在衍生作業實例的來源作業範本未指定自動刪除衍生的作業實例時，才會將作業的結果儲存至資料庫中。

分析作業的結果。

此範例顯示如何檢查已順利完成的作業序號。

```
QueryResultSet result = task.query("DISTINCT TASK.TKIID",
                                   "TASK.NAME = 'CustomerOrder' AND
                                   TASK.STATE = TASK.STATE.STATE_FINISHED",
                                   (String)null, (Integer)null, (TimeZone)null);

if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    ClientObjectWrapper output = task.getOutputMessage(tkiid);
    DataObject myOutput = null;
    if ( output.getObject() != null && output.getObject() instanceof DataObject)
    {
        myOutput = (DataObject)output.getObject();
        int order = myOutput.getInt("OrderNo");
    }
}
```

## 終止作業實例

有時，擁有管理者權限的人員需要去終止已知處於無法復原狀態的作業實例。因為會立即終止作業實例，所以只有在異常狀況下才應該終止作業實例。

1. 擷取要終止的作業實例。

```
Task taskInstance = task.getTask(tkiid);
```

2. 終止作業實例。

```
TKIID tkiid = taskInstance.getID();
task.terminate(tkiid);
```

會立即終止作業實例，而不會等待任何未執行的作業。

## 刪除作業實例

只有在作業實例完成時才會自動刪除作業實例，但前提是這項作業指定於用來衍生實例的關聯作業範本。此範例顯示如何刪除所有已完成且未自動刪除的作業實例。

1. 列出已完成的作業實例。

```
QueryResultSet result =
    task.query("DISTINCT TASK.TKIID",
              "TASK.STATE = TASK.STATE.STATE_FINISHED",
              (String)null, (Integer)null, (TimeZone)null);
```

此動作會傳回列出已完成作業實例的查詢結果集。

2. 刪除已完成的作業實例。

```
while (result.next() )
{
    TKIID tkiid = (TKIID) result.getOID(1);
    task.delete(tkiid);
}
```

## 釋放已要求作業

可能的擁有者要求作業時，此人員會負責完成作業。然而，有時必須釋放要求的作業，這樣另一個可能的擁有者才能要求它。

有時，擁有管理者權限的人員需要釋放要求的作業。例如，必須完成作業但是作業擁有者不在時，可能會發生此狀況。作業的擁有者也可以釋放已要求的作業。

1. 列出特定人員 (例如，Smith) 擁有的已要求作業。

```
QueryResultSet result =
    task.query("DISTINCT TASK.TKIID",
              "TASK.STATE = TASK.STATE.STATE_CLAIMED AND
              TASK.OWNER = 'Smith'",
              (String)null, (Integer)null, (TimeZone)null);
```

此動作會傳回查詢結果集，其中列出特定人員 Smith 所要求的作業。

2. 釋放已要求的作業。

```
if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    task.cancelClaim(tkiid, true);
}
```

此動作會使作業回復為備妥狀態，讓另一位可能的擁有者可以要求它。所有由原始擁有者所設定的輸出或錯誤資料都會予以保留。

## 管理工作項目

在活動實例或作業實例的使用期限期間，可以變更與物件相關聯的人員集合，例如，在人員休假、雇用新人員，或工作量需要進行不同的分配時。若要允許進行這些變更，則可以開發用來建立、刪除或轉移工作項目的應用程式。

工作項目代表因特定原因而將物件指派給使用者或使用者群組。物件通常是人力作業活動實例、程序實例或作業實例。原因是衍生自使用者所具有的物件角色。因為使用者在與物件的關聯上可以具有不同的角色，所以物件可以具有多個工作項目，而且會為每個角色各建立一個工作項目。例如，待辦作業實例可以同時具有管理者、讀者、編輯者及擁有者工作項目。

可以採取以管理工作項目的動作是根據使用者擁有的角色而定，例如，管理者可以建立、刪除及轉移工作項目，而作業擁有者則只可以轉移工作項目。

- 建立工作項目。

```
// query the task instance for which an additional
// administrator is to be specified
QueryResultSet result = task.query("TASK.TKIID",
                                   "TASK.NAME='CustomerOrder'",
                                   (String)null, (Integer)null,
                                   (TimeZone)null);
```

```

if ( result.size() > 0 )
{
    result.first();
    // create the work item
    task.createWorkItem((TKIID)(result.getOID(1)),
                        WorkItem.REASON_ADMINISTRATOR,"Smith");
}

```

此動作會建立擁有管理者角色的使用者 **Smith** 的工作項目。

- 刪除工作項目。

```

// query the task instance for which a work item is to be deleted
QueryResultSet result = task.query("TASK.TKIID",
                                    "TASK.NAME='CustomerOrder'",
                                    (String)null, (Integer)null,
                                    (TimeZone)null);

if ( result.size() > 0 )
{
    result.first();
    // delete the work item
    task.deleteWorkItem((TKIID)(result.getOID(1)),
                        WorkItem.REASON_READER,"Smith");
}

```

此動作會刪除擁有讀者角色的使用者 **Smith** 的工作項目。

- 轉移工作項目。

```

// query the task that is to be rescheduled
QueryResultSet result =
    task.query("DISTINCT TASK.TKIID",
              "TASK.NAME='CustomerOrder' AND
              TASK.STATE=TASK.STATE.STATE_READY AND
              WORK_ITEM.REASON=WORK_ITEM.REASON.REASON_POTENTIAL_OWNER AND
              WORK_ITEM.OWNER_ID='Miller'",
              (String)null, (Integer)null, (TimeZone)null);
if ( result.size() > 0 )
{
    result.first();
    // transfer the work item from user Miller to user Smith
    // so that Smith can work on the task
    task.transferWorkItem((TKIID)(result.getOID(1)),
                          WorkItem.REASON_POTENTIAL_OWNER,"Miller","Smith");
}

```

此動作會將工作項目轉給使用者 **Smith**，讓他可以處理該工作項目。

## 在執行時期建立作業範本及作業實例

您通常會使用塑型工具 (如 **WebSphere Integration Developer**) 來建置作業範本。然後在 **WebSphere Process Server** 中安裝作業範本，並從這些範本建立實例 (例如，使用「商業程序編排器瀏覽器」)。然而，您也可以執行時期建立人工或參與作業實例或範本。

例如，您可能在下列情況也會想要進行這樣的處理：部署應用程式時無法使用作業定義、尚未得知屬於工作流程一部分的作業，或需要作業來涵蓋一組人員間的部分特定協同作業。

若要建立特定「待辦」作業或「協同作業」的模型，您可以建立 `com.ibm.task.api.TaskModel` 類別的實例，並使用它們來建立可重複使用的作業範本，或直接建立執行一次的作業實例。若要建立 `TaskModel` 類別的實例，有一組 `Factory` 方

法可用於 `com.ibm.task.api.ClientTaskFactory` `Factory` 類別。在執行時期建立人力作業的模型是以 `Eclipse Modeling Framework (EMF)` 為基礎。

1. 使用 `createResourceSet` `Factory` 方法來建立 `org.eclipse.emf.ecore.resource.ResourceSet`。
2. 選擇性的: 如果想要使用複式訊息類型, 可採用您可以使用 `Factory` 方法 `getXSDFactory()` 取得的 `org.eclipse.xsd.XSDFactory` 來定義它們, 或使用 `loadXSDSchema` `Factory` 方法直接匯入現有的 XML 綱目。

若要讓 `WebSphere Process Server` 可以使用複式類型, 請將它們部署為企業應用程式的一部分。

3. 建立或匯入類型 `javax.wsdl.Definition` 的「Web 服務定義語言 (WSDL)」定義。

您可以使用 `createWSDLDefinition` 方法來建立新的 WSDL 定義。然後, 您可以將它新增至埠類型及作業。您可以使用 `loadWSDLDefinition` `Factory` 方法直接匯入現有的 WSDL 定義。

4. 使用 `createTTask` `Factory` 方法建立作業定義。

如果想要新增或維護更複雜的作業元素, 您可以使用 `com.ibm.wbit.tel.TaskFactory` 類別, 這個類別可以使用 `getTaskFactory` `Factory` 方法來擷取。

5. 使用 `createTaskModel` `Factory` 方法來建立作業模型, 並將它傳遞給您在步驟 1 中建立的資源組合, 而且它會聚集您同時建立的所有其他成品。
6. 選擇性的: 使用 `TaskModel` `validate` 方法來驗證模型。

使用其中一個「人力作業管理程式」EJB API `create` 方法, 讓 `TaskModel` 參數建立可重複使用的作業範本, 或建立執行一次的作業實例。

#### 相關概念

第 34 頁的『作業範本』

人力作業範本包含已部署作業模型的定義, 此定義是使用 `WebSphere Integration Developer` 所建立, 或是使用「商業程序編排器」API 於執行時期建立的。

## 建立使用簡式 Java 類型的執行時期作業

此範例建立的執行時期作業只會在其介面中使用簡式 Java 類型 (例如, 「字串」物件)。

此範例只會在載入資源的呼叫企業應用程式的環境定義內執行。

1. 存取 `ClientTaskFactory`, 並建立資源集以包含新作業模型的定義。

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

2. 建立 WSDL 定義, 並新增作業的說明。

```
// create the WSDL interface
Definition definition = factory.createWSDLDefinition
    ( resourceSet, new QName( "http://www.ibm.com/task/test/", "test" ) );

// create a port type
PortType portType = factory.createPortType( definition, "doItPT" );

// create an operation; the input and output messages are of type String;
// a fault message is not specified
Operation operation = factory.createOperation
    ( definition, portType, "doIt",
      new QName( "http://www.w3.org/2001/XMLSchema", "string" ),
      new QName( "http://www.w3.org/2001/XMLSchema", "string" ),
      (Map)null );
```



### 3. 建立新人力作業的 EMF 模型。

如果建立作業實例，則不需要有效起始日期 (UTCDate)。

```
TTask humanTask = factory.createTTask( resourceSet,
                                       TTaskKinds.HTASK_LITERAL,
                                       "TestTask",
                                       new UTCDate( "2005-01-01T00:00:00" ),
                                       "http://www.ibm.com/task/test/",
                                       portType,
                                       operation );
```

此步驟會使用預設值來起始設定作業模型的內容。

### 4. 修改人力作業模型的內容。

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance( TBoolean, YES_LITERAL );

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
    taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);
```

### 5. 建立內含所有資源定義的作業模型。

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

### 6. 驗證作業模型，並更正任何發現的驗證問題。

```
ValidationProblem[] validationProblems = taskModel.validate();
```

### 7. 建立執行時期作業實例或範本。

使用 `HumanTaskManagerService` 介面，以建立作業實例或作業範本。因為應用程式只使用簡式 Java 類型，所以您不需要指定應用程式名稱。

- 下列片段會建立作業實例：

```
atask.createTask( taskModel, (String)null, "HTM" );
```

- 下列片段會建立作業範本：

```
task.createTaskTemplate( taskModel, (String)null );
```

如果已建立執行時期作業實例，則現在可以啟動它。如果已建立執行時期作業範本，則現在可以從範本建立作業實例。

## 建立使用複式類型的執行時期作業

此範例會建立在其介面使用複式類型的執行時期作業。已定義複式類型，即用戶端上的本端檔案系統具有內含複式類型說明的 XSD 檔。

此範例只會在載入資源的呼叫企業應用程式的環境定義內執行。

1. 存取 `ClientTaskFactory`，並建立資源集以包含新作業模型的定義。

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

- 將複式類型的 XSD 定義新增至資源集，讓您可以在定義作業時使用它們。

這些檔案是位在與執行程式碼相對的位置。

```
factory.loadXSDSchema( resourceSet, "InputBO.xsd" );
factory.loadXSDSchema( resourceSet, "OutputBO.xsd" );
```

- 建立 WSDL 定義，並新增作業的說明。

```
// create the WSDL interface
Definition definition = factory.createWSDLDefinition
    ( resourceSet, new QName( "http://www.ibm.com/task/test/", "test" ) );
```

```
// create a port type
PortType portType = factory.createPortType( definition, "doItPT" );
```

```
// create an operation; the input message is an InputBO and
// the output message an OutputBO;
// a fault message is not specified
Operation operation = factory.createOperation
    ( definition, portType, "doIt",
      new QName( "http://Input", "InputBO" ),
      new QName( "http://Output", "OutputBO" ),
      (Map)null );
```

- 建立新人力作業的 EMF 模型。

如果建立作業實例，則不需要有效起始日期 (UTCDate)。

```
TTask humanTask = factory.createTTask( resourceSet,
    TTaskKinds.HTASK_LITERAL,
    "TestTask",
    new UTCDate( "2005-01-01T00:00:00" ),
    "http://www.ibm.com/task/test/",
    portType,
    operation );
```

此步驟會使用預設值來起始設定作業模型的內容。

- 修改人力作業模型的內容。

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance( TBoolean, YES_LITERAL );

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
    taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);
```

- 建立內含所有資源定義的作業模型。

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

- 驗證作業模型，並更正任何發現的驗證問題。

```
ValidationProblem[] validationProblems = taskModel.validate();
```

## 8. 建立執行時期作業實例或範本。

使用 `HumanTaskManagerService` 介面，以建立作業實例或作業範本。您必須提供內含資料類型定義的應用程式名稱，才能存取它們。應用程式也必須包含虛擬作業或程序，這樣「商業程序編排器」才會載入應用程式。

- 下列片段會建立作業實例：

```
task.createTask( taskModel, "BOapplication", "HTM" );
```

- 下列片段會建立作業範本：

```
task.createTaskTemplate( taskModel, "BOapplication" );
```

如果已建立執行時期作業實例，則現在可以啟動它。如果已建立執行時期作業範本，則現在可以從範本建立作業實例。

## 建立使用現有介面的執行時期作業

此範例會建立利用已定義介面的執行時期作業，即用戶端上的本端檔案系統具有內含介面說明的檔案。

此範例只會在載入資源的呼叫企業應用程式的環境定義內執行。

1. 存取 `ClientTaskFactory`，並建立資源集以包含新作業模型的定義。

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

2. 存取 `WSDL` 定義及作業的說明。

介面說明是位在與執行程式碼相對的位置。

```
Definition definition = factory.loadWSDLDefinition(
    resourceSet, "interface.wsdl" );
PortType portType = definition.getPortType(
    new QName( definition.getTargetNamespace(), "doItPT" ) );
Operation operation = portType.getOperation(
    "doIt", (String)null, (String)null);
```

3. 建立新人力作業的 `EMF` 模型。

如果建立作業實例，則不需要有效起始日期 (`UTCDate`)。

```
TTask humanTask = factory.createTTask( resourceSet,
    TTaskKinds.HTASK_LITERAL,
    "TestTask",
    new UTCDate( "2005-01-01T00:00:00" ),
    "http://www.ibm.com/task/test/",
    portType,
    operation );
```

此步驟會使用預設值來起始設定作業模型的內容。

4. 修改人力作業模型的內容。

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance( TBoolean, YES_LITERAL );

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
```

```

TEscalationReceiver escalationReceiver =
    taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);

```

5. 建立內含所有資源定義的作業模型。

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

6. 驗證作業模型，並更正任何發現的驗證問題。

```
ValidationProblem[] validationProblems = taskModel.validate();
```

7. 建立執行時期作業實例或範本。

使用 `HumanTaskManagerService` 介面，以建立作業實例或作業範本。您必須提供內含資料類型定義的應用程式名稱，才能存取它們。應用程式也必須包含虛擬作業或程序，這樣「商業程序編排器」才會載入應用程式。

- 下列片段會建立作業實例：

```
task.createTask( taskModel, "BOapplication", "HTM" );
```

- 下列片段會建立作業範本：

```
task.createTaskTemplate( taskModel, "BOapplication" );
```

如果已建立執行時期作業實例，則現在可以啟動它。如果已建立執行時期作業範本，則現在可以從範本建立作業實例。

## 建立使用呼叫應用程式介面的執行時期作業

此範例建立的執行時期作業會使用屬於呼叫應用程式一部分的介面。例如，執行時期作業是建立在商業程序的 Java 片段中，並使用程序應用程式的介面。

此範例只會在載入資源的呼叫企業應用程式的環境定義內執行。

1. 存取 `ClientTaskFactory`，並建立資源集以包含新作業模型的定義。

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();

// specify the context class loader so that following resources are found
ResourceSet resourceSet = factory.createResourceSet
    ( Thread.currentThread().getContextClassLoader() );
```

2. 存取 WSDL 定義及作業的說明。

在包含套件 JAR 檔內指定路徑。

```
Definition definition = factory.loadWSDLDefinition( resourceSet,
    "com/ibm/workflow/metaflow/interface.wsdl" );
PortType portType = definition.getPortType(
    new QName( definition.getTargetNamespace(), "doItPT" ) );
Operation operation = portType.getOperation
    ("doIt", (String)null, (String)null);
```

3. 建立新人力作業的 EMF 模型。

如果建立作業實例，則不需要有效起始日期 (UTCDate)。

```
TTask humanTask = factory.createTTask( resourceSet,
    TTaskKinds.HTASK_LITERAL,
    "TestTask",
    new UTCDate( "2005-01-01T00:00:00" ),
```

```
"http://www.ibm.com/task/test/",
portType,
operation );
```

此步驟會使用預設值來起始設定作業模型的內容。

4. 修改人力作業模型的內容。

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance( TBoolean, YES_LITERAL );

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
    taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);
```

5. 建立內含所有資源定義的作業模型。

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

6. 驗證作業模型，並更正任何發現的驗證問題。

```
ValidationProblem[] validationProblems = taskModel.validate();
```

7. 建立執行時期作業實例或範本。

使用 `HumanTaskManagerService` 介面，以建立作業實例或作業範本。您必須提供內含資料類型定義的應用程式名稱，才能存取它們。

- 下列片段會建立作業實例：

```
task.createTask( taskModel, "WorkflowApplication", "HTM" );
```

- 下列片段會建立作業範本：

```
task.createTaskTemplate( taskModel, "WorkflowApplication" );
```

如果已建立執行時期作業實例，則現在可以啟動它。如果已建立執行時期作業範本，則現在可以從範本建立作業實例。

## HumanTaskManagerService 介面

`HumanTaskManagerService` 介面顯示可由本端或遠端用戶端呼叫的作業相關功能。

可以呼叫的方法視作業狀態及使用包含方法之應用程式的人員權限而定。這裡列出操作作業物件的主要方法。如需這些方法及 `HumanTaskManagerService` 介面中其他可用方法的相關資訊，請參閱 `com.ibm.task.api` 套件中的 Javadoc。

## 作業範本

使用作業範本有下列方法。

表 43. 作業範本的 API 方法

方法	說明
getTaskTemplate	擷取指定的作業範本。
createAndCallTask	從指定的作業範本建立及執行作業實例，同時等待結果。
createAndStartTask	從指定的作業範本建立及啓動作業實例。
createTask	從指定的作業範本建立作業實例。
createInputMessage	建立指定作業範本的輸入訊息。例如，建立可用於啓動作業的訊息。
queryTaskTemplates	擷取資料庫中所儲存的作業範本。

## 作業實例

使用作業實例有下列方法。

表 44. 作業實例的 API 方法

方法	說明
getTask	擷取作業實例；可以是任何狀態的作業實例。
callTask	同步啓動呼叫作業。
startTask	啓動已建立的作業。
suspend	暫停協同作業或待辦作業。
resume	回復協同作業或待辦作業。
terminate	終止指定的作業實例。如果終止呼叫作業，則此動作對呼叫的服務沒有任何影響。
delete	刪除指定的作業實例。
claim	要求處理作業。
update	更新作業實例。
complete	完成作業實例。
cancelClaim	釋出要求的作業實例，以便其他可能的擁有者可以使用它。
createWorkItem	建立作業實例的工作項目。
transferWorkItem	將工作項目轉移給指定的擁有者。
deleteWorkItem	刪除工作項目。

## 呈報

使用呈報有下列方法。

表 45. 使用呈報的 API 方法

方法	說明
getEscalation	擷取指定的呈報實例。

## 自訂內容

作業、作業範本及呈報都可以具有自訂內容。介面提供 `get` 及 `set` 方法來擷取及設定自訂內容的值。您還可以將指定內容與作業實例相關聯，並從中擷取指定內容。自訂內容名稱及值必須為 `java.lang.String` 類型。下列方法可用於作業、作業範本及呈報。

表 46. 變數及自訂內容的 API 方法

方法	說明
<code>getCustomProperty</code>	擷取指定作業實例的指定自訂內容。
<code>getCustomProperties</code>	擷取指定作業實例的自訂內容。
<code>getCustomPropertyNames</code>	擷取作業實例的自訂內容名稱。
<code>setCustomProperty</code>	儲存指定作業實例的自訂特定值。

## 接受的作業動作

可以對作業執行的動作視作業是待辦作業、協同作業、呼叫作業或管理作業而定。

您無法將 `HumanTaskManager` 介面提供的所有動作用於所有類型的作業。下表顯示您可以對每種作業執行的動作。

動作	作業類型			
	待辦作業	協同作業	呼叫作業	管理作業
<code>callTask</code>			X	
<code>cancelClaim</code>	X	X <sup>1</sup>		
<code>claim</code>	X	X <sup>1</sup>		
<code>complete</code>	X	X <sup>1</sup>		X
<code>completeWithFollowOnTask<sup>4</sup></code>	X	X <sup>1</sup>		
<code>completeWithFollowOnTask<sup>5</sup></code>		X <sup>3</sup>	X <sup>3</sup>	
<code>createFaultMessage</code>	X	X	X	X
<code>createInputMessage</code>	X	X	X	X
<code>createOutputMessage</code>	X	X	X	X
<code>createWorkItem</code>	X	X <sup>1</sup>	X	X
<code>delete</code>	X <sup>1</sup>	X <sup>1</sup>	X	X <sup>1</sup>
<code>deleteWorkItem</code>	X	X <sup>1</sup>	X	X
<code>getCustomProperty</code>	X	X <sup>1</sup>	X	X
<code>getDocumentation</code>	X	X <sup>1</sup>	X	X
<code>getFaultNames</code>	X	X <sup>1</sup>		
<code>getFaultMessage</code>	X	X <sup>1</sup>	X	
<code>getInputMessage</code>	X	X <sup>1</sup>	X	
<code>getOutputMessage</code>	X	X <sup>1</sup>	X	
<code>getUsersInRole</code>	X	X <sup>1</sup>	X	X
<code>getTask</code>	X	X <sup>1</sup>	X	X
<code>getUISettings</code>	X	X <sup>1</sup>	X	X
<code>resume</code>	X	X <sup>1</sup>		
<code>setCustomProperty</code>	X	X <sup>1</sup>	X	X



動作	作業類型			
	待辦作業	協同作業	呼叫作業	管理作業
setFaultMessage	X	X <sup>1</sup>		
setOutputMessage	X	X <sup>1</sup>		
startTask	X <sup>1</sup>	X <sup>1</sup>	X	X
startTaskAsSubtask <sup>6</sup>	X	X <sup>1</sup>		
startTaskAsSubtask <sup>7</sup>		X <sup>3</sup>	X <sup>3</sup>	
suspend	X	X <sup>1</sup>		
suspendWithCancelClaim	X	X <sup>1</sup>		
terminate	X <sup>1</sup>	X <sup>1</sup>	X <sup>1</sup>	
transferWorkItem	X	X <sup>1</sup>	X	X
update	X	X <sup>1</sup>	X	X
<b>附註:</b> 1. 只適用於獨立式作業、特定作業及作業範本 2. 只適用於獨立式作業、商業程序中的行內作業及特定作業 3. 只適用於獨立式作業及特定作業 4. 可以包含後續作業的作業類型 5. 可以作為後續作業使用的作業類型 6. 可以包含子作業的作業類型 7. 可以作為子作業使用的作業類型				

## 開發商業程序及人力作業的應用程式

人員會包括在大部分的商業程序範例情況中。例如，當啟動或管理程序時，或當執行人力作業活動時，商業程序需要人員互動。若要支援這些範例情況，您需要同時使用「商業流程管理程式 API」及「人力作業管理程式 API」。

若要在商業程序範例情況中包括人員，您可以在商業程序中併入下列作業種類：

- 行內呼叫作業 (在 API 中也稱為原始作業)。

您可以提供呼叫作業給每一個接收活動、挑選活動的每一個 onMessage 元素，以及事件處理程式的每一個 onEvent 元素。然後，這項作業會控制誰有權啟動程序，或與執行中程序實例通訊。

- 管理作業。

您可以提供管理作業，指定誰有權管理程序，或在程序的失敗活動上執行管理作業。

- 待辦作業 (在 API 中也稱為參與作業)。

待辦作業會實作人力作業活動。這類活動可讓您在程序中包括人員。

商業程序中的人力作業活動代表人員在商業程序範例情況中執行的待辦作業。您可以同時使用「商業流程管理程式 API」及「人力作業管理程式 API」，來實現這些範例情況：

- 商業程序是屬於程序的所有活動的儲存器，包括待辦作業所代表的人力作業活動。當建立程序實例時，會指派唯一的物件 ID (PIID)。
- 在執行程序實例期間啟動人力作業活動時，會建立活動實例，以其唯一的物件 ID (AIID) 加以識別。同時，也會建立行內待辦作業實例，以其物件 ID (TKIID) 加以識別。人力作業活動與作業實例的關係是使用物件 ID 來完成：
  - 活動實例的待辦作業 ID 會設定為相關聯的待辦作業的 TKIID。
  - 作業實例的包含環境定義 ID 會設定為程序實例的 PIID，這個程序實例包含相關聯的活動實例。
  - 作業實例的母環境定義 ID 會設定為相關聯的活動實例的 AIID。
- 所有行內待辦作業實例的生命週期都是由程序實例來管理。刪除程序實例時，也會刪除作業實例。換言之，會自動刪除將包含環境定義 ID 設定為程序實例的 PIID 的所有作業。

## 判斷可以啟動的程序範本或活動

您可以呼叫「商業流程管理程式 API」的 `call`、`initiate` 或 `sendMessage` 方法，來啟動商業程序。如果程序只有一個起始活動，您可以使用需要程序範本名稱的方法簽章作為參數。如果程序具有多個起始活動，您必須明確地識別起始活動。

建立商業程序的模型時，模型器可以決定只有使用者子集才能從程序範本建立程序實例。作法是使行內呼叫作業與程序的起始活動產生關聯，並指定該作業上的授權限制。只容許可能是作業的起始者或管理者的人員，才能建立作業的實例，並從而建立程序範本的實例。

如果行內呼叫作業與起始活動無關，或如果未對作業指定授權限制，則每個人都可以使用起始活動來建立程序實例。

一個程序可以具有多個起始活動，每一個起始活動都具有可能起始者或管理者的不同人員查詢。這表示可以授權使用者使用活動 A 來啟動程序，但不能使用活動 B 來啟動程序。

1. 使用「商業流程管理程式 API」，來建立處於已啟動狀態的程序範本的現行版本清單。

**提示:** `queryProcessTemplates` 方法只會排除那些是應用程式一部分，但尚未啟動的程序範本。因此，如果您使用這個方法而不過濾結果，則方法會傳回程序範本的所有版本，不管它們所處的狀態為何。

```
// current timestamp in UTC format, converted to yyyy-mm-ddThh:mm:ss
String now = (new UTCDate()).toXsdString();
String whereClause = "PROCESS_TEMPLATE.STATE =
PROCESS_TEMPLATE.STATE.STATE_STARTED AND
PROCESS_TEMPLATE.VALID_FROM =
(SELECT MAX(VALID_FROM) FROM PROCESS_TEMPLATE
WHERE NAME=PROCESS_TEMPLATE.NAME AND
VALID_FROM <= TS('" + now + "'))";

ProcessTemplateData[] processTemplates = process.queryProcessTemplates
( whereClause,
  "PROCESS_TEMPLATE.NAME",
  (Integer)null, (TimeZone)null);
```

結果是依程序範本名稱來排序。

2. 建立使用者已獲授權的程序範本清單及起始活動清單。

程序範本清單包含那些具有單一起始活動的程序範本。這些活動不是不安全，就是容許已登入的使用者啟動它們。或者，您可能想要收集至少一個起始活動可以啟動的程序範本。

**提示：** 程序管理者也可以啟動程序實例。若要取得完整的範本清單，您也需要讀取與程序範本相關聯的管理作業範本，以及檢查已登入的使用者是否為管理者。

```
List authorizedProcessTemplates = new ArrayList();
List authorizedActivityServiceTemplates = new ArrayList();
```

3. 判斷每一個程序範本的起始活動。

```
for( int i=0; i<processTemplates.length; i++ )
{
    ProcessTemplateData template = processTemplates[i];
    ActivityServiceTemplateData[] startActivities =
        process.getStartActivities(template.getID());
```

4. 對於每一個起始活動，擷取相關聯的行內呼叫作業範本的 ID。

```
for( int j=0; j<startActivities.length; j++ )
{
    ActivityServiceTemplateData activity = startActivities[j];
    TKID tktid = activity.getTaskTemplateID();
```

- a. 如果呼叫作業範本不存在，則這個起始活動不會保護程序範本的安全。

在此情況下，每個人都可以使用這個起始活動來建立程序實例。

```
boolean isAuthorized = false;
    if ( tktid == null )
    {
        isAuthorized = true;
        authorizedActivityServiceTemplates.add(activity);
    }
```

- b. 如果呼叫作業範本存在，請使用「人力作業管理程式 API」，來檢查已登入使用者的授權。

在範例中，已登入的使用者為 **Smith**。已登入的使用者必須是呼叫作業的可能起始者或管理者。

```
if ( tktid != null )
{
    isAuthorized =
        task.isUserInRole
            (tkid, "Smith", WorkItem.REASON_POTENTIAL_STARTER) ||
        task.isUserInRole(tktid, "Smith", WorkItem.REASON_ADMINISTRATOR);

    if ( isAuthorized )
    {
        authorizedActivityServiceTemplates.add(activity);
    }
}
```

如果使用者具有指定的角色，或未指定角色的人員指派準則，則 `isUserInRole` 方法會傳回 `true` 值。

5. 檢查是否只能使用程序範本名稱來啟動程序。

```
if ( isAuthorized && startActivities.length == 1 )
{
    authorizedProcessTemplates.add(template);
}
```

6. 結束迴圈。

```

    } // end of loop for each activity service template
  } // end of loop for each process template

```

## 處理包括人力作業的單一人員工作流程

部分工作流程只由一個人執行 (例如，從線上書店訂購書籍)。這個範例顯示如何實作訂購書本的一連串動作，作為一系列的人力作業活動 (待辦作業)。「商業流程管理程式」API 及「人力作業管理程式」API 都可用來處理工作流程。

在線上書店中，購買者需要完成一連串的動作才能訂購書籍。這一連串動作可以實作為一系列的人力作業活動 (待辦作業)。如果購買者決定要訂購數本書，則這等於要求下一個人力作業活動。一連串作業的相關資訊是由「商業流程管理程式」維護，而作業本身則是由「人力作業管理程式」維護。

將此範例與只使用「商業流程管理程式」API 的範例進行比較。

1. 使用「商業流程管理程式」API，以取得您想要使用的程序實例。

在這個範例中，CustomerOrder 程序的實例。

```

ProcessInstanceData processInstance =
    process.getProcessInstance("CustomerOrder");
String piid = processInstance.getID().toString();

```

2. 使用「人力作業管理程式」API 來查詢備妥的待辦作業 (參與種類)，這些作業是指定的程序實例的一部分。

使用作業的包含環境定義 ID 來指定包含程序實例。若為單一人員工作流程，查詢會傳回與一連串人力作業活動中的第一個人力作業活動相關聯的待辦作業。

```

//
// Query the list of to-do tasks that can be claimed by the logged-on user
// for the specified process instance
//
QueryResultSet result =
    task.query("DISTINCT TASK.TKIID",
               "TASK.CONTAINMENT_CTX_ID = ID(' + piid + "') AND
               TASK.STATE = TASK.STATE.STATE_READY AND
               TASK.KIND = TASK.KIND.KIND_PARTICIPATING AND
               WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
               (String)null, (Integer)null, (TimeZone)null);

```

3. 要求傳回的待辦作業。

```

if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    ClientObjectWrapper input = task.claim(tkiid);
    DataObject activityInput = null ;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        taskInput = (DataObject)input.getObject();
        // read the values
        ...
    }
}

```

要求作業時，會傳回作業的輸入訊息。

4. 判斷與待辦作業相關聯的人力作業活動。

您可以使用下列其中一項方法，使活動與其作業產生關聯。

- task.getActivityID 方法：  
AIID aiid = task.getActivityID(tkiid);

- 作業物件一部分的母環境定義 ID：  
AIID aiid = null;  
Task taskInstance = task.getTask(tkiid);  
OID oid = taskInstance.getParentContextID();  
if ( oid != null and oid instanceof AIID )  
{  
    aiid = (AIID)oid;  
}

5. 當完成了作業上的工作時，請使用「商業流程管理程式」API，完成作業及其相關聯的人力作業活動，然後要求程序實例中的下一個人力作業活動。

若要完成人力作業活動，會傳遞一則輸出訊息。在您建立輸出訊息時，必須指定訊息類型名稱，以包含訊息定義。

```
ActivityInstanceData activity = process.getActivityInstance(aiid);
ClientObjectWrapper output =
    process.createMessage(aiid, activity.getOutputMessageType());
DataObject myMessage = null ;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myMessage = (DataObject)output.getObject();
    //set the parts in your message, for example, an order number
    myMessage.setInt("OrderNo", 4711);
}
```

```
//complete the human task activity and its associated to-do task,
// and claim the next human task activity
CompleteAndClaimSuccessorResult successor =
    process.completeAndClaimSuccessor(aiid, output);
```

這個動作會設定包含訂單號碼的輸出訊息，並要求序列中的下一個人力作業活動。如果將後續活動設定為 AutoClaim，而且如果有多個可以遵循的路徑，則會要求所有後續活動，並將隨機活動傳回為下一個活動。如果沒有其他後續活動可以指派給此使用者，則會傳回 Null。

如果程序包含可以遵循的平行路徑，而且這些路徑包含人力作業活動，其中已登入使用者可能是這些活動中多個活動的擁有者，則會自動要求隨機活動，並且該隨機活動會傳回為下一個活動。

6. 使用下一個人力作業活動。

```
ClientObjectWrapper nextInput = successor.getInputMessage();
if ( nextInput.getObject() !=
    null && nextInput.getObject() instanceof DataObject )
{
    activityInput = (DataObject)input.getObject();
    // read the values
    ...
}

aiid = successor.getAIID();
```

7. 繼續步驟 5，以完成人力作業活動並擷取下一個人力作業活動。

## 處理異常及錯誤

BPEL 程序可能會在程序的不同位置發現錯誤。

「商業程序執行語言 (BPEL)」錯誤源自：

- Web 服務呼叫 (「Web 服務說明語言 (WSDL)」錯誤)
- 擲出活動
- 「商業程序編排器」可辨識的 BPEL 標準錯誤

有機制可以處理這些錯誤。請使用下列其中一個機制，來處理程序實例所產生的錯誤：

- 將控制權傳遞給對應的錯誤處理程式
- 補償程序中的前一個工作
- 停止程序，並讓人員修復狀況 (強制重試、強制完成)

BPEL 程序也會將錯誤傳回給程序所提供的作業呼叫者。您可以將程序中的錯誤塑型為具有錯誤名稱及錯誤資料的回覆活動。這些錯誤會傳回給 API 呼叫者當作查核異常。

如果 BPEL 程序未處理 BPEL 錯誤，或發生 API 異常，則會將執行時期異常傳回給 API 呼叫者。API 異常的範例是發生在從中建立實例的程序模型不存在時。

下列作業會說明錯誤及異常狀況的處理。

### 處理 API 異常

如果 `BusinessFlowManagerService` 介面或 `HumanTaskManagerService` 介面中的方法未順利完成，則會擲出表示錯誤原因的異常。您可以特別提供指引給呼叫者，以處理此異常。

然而，一般作法是只特別處理一小部分的異常，並提供其他潛在異常的一般指引。所有特定的異常都是繼承自一般 `ProcessException` 或 `TaskException`。這是使用最終 `catch (ProcessException)` 或 `catch (TaskException)` 陳述式捕捉一般異常的最佳作法。因為此陳述式兼顧到所有可能會發生的其他異常，所以它有助於確保應用程式的向上相容性。

### 檢查針對活動設定的錯誤

1. 列出處於失敗或已停止狀態的作業活動。

```
QueryResultSet result =
    process.query("ACTIVITY.AIID",
        "(ACTIVITY.STATE = ACTIVITY.STATE.STATE_FAILED OR
        ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED) AND
        ACTIVITY.KIND=ACTIVITY.KIND.KIND_STAFF",
        (String)null, (Integer)null, (TimeZone)null);
```

此動作會傳回內含已失敗或已停止活動的查詢結果集。

2. 讀取錯誤的名稱。

```
if (result.size() > 0)
{
    result.first();
    AIID aiid = (AIID) result.getOID(1);
    ClientObjectWrapper faultMessage = process.getFaultMessage(aiid);
```

```

DataObject fault = null ;
if ( faultMessage.getObject() != null && faultMessage.getObject()
    instanceof DataObject )
{
    fault = (DataObject)faultMessage.getObject();
    Type type = fault.getType();
    String name = type.getName();
    String uri = type.getURI();
}
}

```

這會傳回錯誤名稱。您也可以分析已停止活動的無法處理異常狀況，而不是擷取錯誤名稱。

## 檢查已停止呼叫活動所發生的錯誤

如果活動發生錯誤，則錯誤類型會判斷您可以用來修復活動的動作。

1. 列出處於已停止狀態的人力作業活動。

```

QueryResultSet result =
    process.query("ACTIVITY.AIID",
        "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
        ACTIVITY.KIND=ACTIVITY.KIND.KIND_INVOKE",
        (String)null, (Integer)null, (TimeZone)null);

```

此動作會傳回內含已停止呼叫活動的查詢結果集。

2. 讀取錯誤的名稱。

```

if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
    ActivityInstanceData activity = process.getActivityInstance(aaid);

    ProcessException excp = activity.getUnhandledException();
    if ( excp instanceof ApplicationFaultException )
    {
        ApplicationFaultException fault = (ApplicationFaultException)excp;
        String faultName = fault.getFaultName();
    }
}
}

```





---

## 第 12 章 開發 Web 服務 API 用戶端應用程式

您可以開發用戶端應用程式，以透過 Web 服務 API 來存取商業程序應用程式及人力作業應用程式。

用戶端應用程式可以在任何 Web 服務用戶端環境中進行開發，包括 Java Web 服務及 Microsoft .NET。

---

### 簡介：Web 服務

Web 服務是使用開放的 XML 型標準及傳輸通訊協定，來與用戶端應用程式交換資料的 Web 型企業應用程式。Web 服務容許使用語言及環境為中性的程式設計模型。

Web 服務使用下列核心技術：

- XML (可延伸標記語言)。XML 會解決資料獨立性的問題。您可使用它來說明資料，也可以將該資料對映到或對映出任何應用程式或程式設計語言。
- WSDL (Web 服務說明語言)。您可使用此 XML 型語言來建立基礎應用程式的說明。此說明會透過作為基礎應用程式與其他 Web 支援應用程式之間的介面，將應用程式轉換為 Web 服務。
- SOAP (簡易物件存取通訊協定)。SOAP 是 Web 的核心通訊協定，且大部分 Web 服務會使用此通訊協定以彼此通訊。

---

### Web 服務元件及控制順序

許多用戶端及伺服器端元件會參與代表 Web 服務要求及回應的控制順序。

控制的一般順序如下所示。

1. 在用戶端上：
  - a. 用戶端應用程式 (由使用者提供) 會發出 Web 服務的要求。
  - b. proxy 用戶端 (也是由使用者提供，但可以使用用戶端公用程式來自動產生) 會將服務要求包裝在 SOAP 要求信封中。
  - c. 用戶端開發基礎架構會將要求遞送至定義為 Web 服務端點的 URL。
2. 網路會使用 HTTP 或 HTTPS 將要求傳輸至 Web 服務端點。
3. 在伺服器端上：
  - a. 一般 Web 服務 API 會接收及解碼要求。
  - b. 要求會由一般「商業流程管理程式」或「人力作業管理程式」元件直接處理，或者會遞送至指定的商業程序或人力作業。
  - c. 傳回的資料會包裝在 SOAP 回應信封中。
4. 網路會使用 HTTP 或 HTTPS 將回應傳輸至用戶端環境。
5. 回到用戶端上：
  - a. 用戶端開發基礎架構會解除 SOAP 回應信封的包裝。
  - b. proxy 用戶端會從 SOAP 回應擷取資料，並將其傳遞至用戶端應用程式。
  - c. 用戶端應用程式會在必要時處理傳回的資料。

---

## Web 服務 API 的概觀

Web 服務 API 可讓您開發用戶端應用程式，以使用 Web 服務來存取在「商業程序編排器」環境中執行的商業程序及人力作業。

「商業程序編排器」Web 服務 API 會提供兩個不同的 Web 服務介面 (WSDL 埠類型)：

- 商業流程管理程式 API。容許用戶端應用程式與微流程及長時間執行的程序互動，例如：
  - 建立程序範本及程序實例
  - 要求現有的程序
  - 依其 ID 查詢程序

如需可能動作的完整清單，請參閱第 370 頁的『開發商業程序的應用程式』。

- 人力作業管理程式 API。容許用戶端應用程式執行下列動作：
  - 建立及啓動作業
  - 要求現有的作業
  - 完成作業
  - 依其 ID 查詢作業
  - 查詢作業的集合。

如需可能動作的完整清單，請參閱第 387 頁的『開發人力作業的應用程式』。

用戶端應用程式可以使用其中一個或兩個 Web 服務介面。

### 範例

存取「人力作業管理程式」Web 服務 API，來處理參與人力作業的用戶端應用程式可能概要如下所示：

1. 用戶端應用程式會發出 query Web 服務呼叫至 WebSphere Process Server，要求由使用者處理的參與作業的清單。
2. 參與作業的清單會傳回至 SOAP/HTTP 回應信封中。
3. 然後，用戶端應用程式會發出 claim Web 服務呼叫，來要求其中一個參與作業。
4. WebSphere Process Server 會傳回作業的輸入訊息。
5. 用戶端應用程式會發出 complete Web 服務呼叫，來完成具有輸出或錯誤訊息的作業。

---

## 商業程序及人力作業的需求



使用 WebSphere Integration Developer 開發商業程序及人力作業，以在「商業程序編排器」上執行時，必須符合特定規則才能透過 Web 服務 API 存取。

需求如下：

1. 商業程序及人力作業的介面必須是使用「document/literal 包裝的」樣式 (在 XML 型 RPC (JAX-RPC 1.1) 規格的 Java API 中定義) 來定義。這是使用 WID 開發的所有商業程序及人力作業的預設樣式。
2. Web 服務作業的商業程序及人力作業所顯示的錯誤訊息，必須包含使用「XML 綱目」元素定義的單一 WSDL 訊息組件。例如：

```
<wsdl:part name="myFault" element="myNamespace:myFaultElement"/>
```

### 相關資訊

-  [XML 型 RPC \(JAX-RPC\) 的 Java API 下載頁面](#)
-  [我應該使用哪個 WSDL 樣式？](#)

---

## 開發用戶端應用程式

用戶端應用程式開發程序是由一些步驟組成。

1. 決定用戶端應用程式需要使用的 Web 服務 API：「商業流程管理程式」API、「人力作業管理程式」API 或這兩者。
2. 從 WebSphere Process Server 環境中匯出必要的檔案。也可以從 WebSphere Process Server 用戶端 CD 複製檔案。
3. 在選擇的用戶端應用程式開發環境中，使用匯出的成品來產生 *Proxy* 用戶端。
4. 選擇性的：產生 *Helper* 類別。如果用戶端應用程式直接與 WebSphere 伺服器上的具體程序或作業互動，則需要 *Helper* 類別。然而，如果用戶端應用程式只會執行一般作業（如發出查詢），則不需要這些類別。
5. 開發用戶端應用程式的程式碼。
6. 將任何必要的安全機制新增至用戶端應用程式。

---

## 複製成品

必須從 WebSphere 環境複製一些成品，以協助建立用戶端應用程式。

有兩種方式可以取得這些成品：

- 從 WebSphere Process Server 環境發佈及匯出它們。
- 從 WebSphere Process Server 用戶端 CD 複製檔案。

### 從伺服器環境中發佈及匯出成品

可以開發用戶端應用程式以存取 Web 服務 API 之前，必須先從 WebSphere 伺服器環境中發佈及匯出一些成品。

要匯出的成品為：

- 「Web 服務定義語言 (WSDL)」檔，說明組成 Web 服務 API 的埠類型及作業。
- 「XML 綱目定義 (XSD)」檔，內含 WSDL 檔中的服務及方法所參照的資料類型定義。
- 其他可說明商業物件的 WSDL 及 XSD 檔。商業物件說明在 WebSphere 伺服器上執行的具體商業程序或人力作業。只有在用戶端應用程式需要透過 Web 服務 API 直接與具體商業程序或人力作業互動時，才需要這些額外檔案。如果用戶端應用程式只會執行一般作業（如發出查詢），則不需要這些額外檔案。

發佈這些成品之後，需要將這些成品複製至用戶端程式設計環境，而在此環境中，會使用這些成品來產生 *Proxy* 用戶端及 *Helper* 類別。

### 相關工作

第 418 頁的『複製用戶端 CD 中的檔案』

WebSphere Process Server 用戶端 CD 內含存取 Web 服務 API 的必要檔案。

## 指定 Web 服務端點位址

Web 服務端點位址是用戶端應用程式必須指定才能存取 Web 服務 API 的 URL。端點位址會寫入至 WSDL 檔中，而此檔案是您匯出以產生用於用戶端應用程式的 Proxy 用戶端。

使用的 Web 服務端點位址是根據 WebSphere 伺服器配置而定：

- 範例情況 1。單一 WebSphere 伺服器。要指定的 WebSphere 端點位址是伺服器的主機名稱及埠號 (例如，**host1:9080**)。
- 範例情況 2。由數部伺服器組成的 WebSphere 叢集。要指定的 WebSphere 端點位址是 Web 服務 API 所在伺服器的主機名稱及埠 (例如，**host2:9081**)。
- 範例情況 3。一部 Web 伺服器作為前端系統。要指定的 WebSphere 端點位址則是該 Web 伺服器的主機名稱及埠 (例如：**host:80**)。

Web 服務端點位址預設會採用 `protocol://host:port/context_root/fixed_path` 格式。其中：

- *protocol*。要在用戶端應用程式與 WebSphere 伺服器之間使用的通訊協定。預設通訊協定是 HTTP。您可以改為選擇使用較安全的 HTTPS (透過 SSL 的 HTTP) 通訊協定。建議使用 HTTPS。
- *host:port*。用來存取 Web 服務 API 所在機器的主機名稱及埠號。這些值會根據 WebSphere 伺服器配置而不同；例如，用戶端應用程式是直接還是透過 Web 伺服器前端系統存取應用程式。
- *context\_root*。您可以自由選擇環境定義根目錄的任何值。然而，選擇的值在每個 WebSphere Cell 內都必須是唯一的。預設值會使用 "node\_server/cluster" 字尾，刪除命名衝突的風險。
- *fixed\_path* 是 `/sca/com/ibm/bpe/api/BFMWS` (「商業流程管理程式」API) 或 `/sca/com/ibm/task/api/HTMWS` (「人力作業管理程式」API)，而且不可以修改。

在配置商業程序儲存器或人力作業儲存器時，會初始指定 Web 服務端點位址：

1. 使用具有管理者權限的使用者 ID 登入管理主控台。
2. 選擇應用程式 → **SCA 模組**。

**註：**您也可以選取應用程式 → 企業應用程式，以顯示所有可用企業應用程式的清單。

3. 從 SCA 模組或應用程式清單中選取 **BPEContainer** (商業程序儲存器) 或 **TaskContainer** (人力作業儲存器)。
4. 從其他內容清單中，選擇提供 **HTTP 端點 URL 資訊**。
5. 從清單中選取其中一個預設字首，或輸入自訂字首。如果用戶端應用程式是要直接連接至 Web 服務 API 所在的應用程式伺服器，則請使用預設字首清單中的字首。否則，請指定自訂字首。
6. 按一下**套用**，以將選取的字首複製至 SCA 模組。
7. 按一下**確定**。URL 資訊會儲存至工作區。

您可以在管理主控台中檢視現行值 (例如，如果是商業程序儲存器：**企業應用程式** → **BPEContainer** → **檢視部署描述子**)。

在匯出的 WSDL 檔中， soap:address 元素的 location 屬性會包含指定的 Web 服務端點位址。例如：

```
<wsdl:service name="BFMWSService">
  <wsdl:port name="BFMWSPort" binding="this:BFMWSBinding">
    <soap:address location=
      "https://myserver:9080/WebServicesAPIs/sca/com/ibm/bpe/api/BFMWS"/>
```

### 相關概念

第 426 頁的『新增安全 (Java Web 服務)』

您必須透過在用戶端應用程式中實作安全機制，來維護 Web 服務通訊的安全。

### 相關工作

第 433 頁的『新增安全 (.NET)』

您可以將安全機制整合至用戶端應用程式，以保護 Web 服務通訊的安全。

『發佈 WSDL 檔』

「Web 服務定義語言 (WSDL)」檔包含使用 Web 服務 API 可以進行的所有作業的詳細說明。「商業流程管理程式」與「人力作業管理程式」Web 服務 API 各有不同的 WSDL 檔。您必須先發佈這些 WSDL 檔，然後再將它們從 WebSphere 環境複製至開發環境，而它們在該環境中是用來產生 Proxy 用戶端。

## 發佈 WSDL 檔

「Web 服務定義語言 (WSDL)」檔包含使用 Web 服務 API 可以進行的所有作業的詳細說明。「商業流程管理程式」與「人力作業管理程式」Web 服務 API 各有不同的 WSDL 檔。您必須先發佈這些 WSDL 檔，然後再將它們從 WebSphere 環境複製至開發環境，而它們在該環境中是用來產生 Proxy 用戶端。

發佈 WSDL 檔之前，請務必指定正確的 Web 服務端點位址。這是用戶端應用程式用來存取 Web 服務 API 的 URL。

您只需要發佈一次 WSDL 檔。

**註：**如果您擁有 WebSphere Process Server 用戶端 CD，則可以改為直接將檔案從該 CD 中複製至用戶端程式設計環境。

### 相關工作

第 429 頁的『產生 Proxy 用戶端 (.NET)』

.NET 用戶端應用程式會使用 *Proxy 用戶端* 與 Web 服務 API 互動。Proxy 用戶端會保護用戶端應用程式，使其不像 Web 服務傳訊通訊協定那樣複雜。

第 414 頁的『指定 Web 服務端點位址』

Web 服務端點位址是用戶端應用程式必須指定才能存取 Web 服務 API 的 URL。端點位址會寫入至 WSDL 檔中，而此檔案是您匯出以產生用於用戶端應用程式的 Proxy 用戶端。

## 發佈商業程序 WSDL：

使用管理主控台來發佈 WSDL 檔。

1. 使用具有管理者權限的使用者 ID 登入管理主控台。
2. 選取應用程式 → **SCA 模組**

**註：**您也可以選取應用程式 → **企業應用程式**，以顯示所有可用企業應用程式的清單。

3. 從 SCA 模組或應用程式的清單中，選擇 **BPEContainer** 應用程式。

4. 從**其他內容**的清單中，選取**發佈 WSDL 檔**
5. 按一下清單中的壓縮檔案。
6. 在出現的「檔案下載」視窗上，按一下**儲存**。
7. 瀏覽至本端資料夾，並按一下**儲存**。

匯出的壓縮檔案命名為 BPEContainer\_WSDLFiles.zip。此壓縮檔案包含可說明 Web 服務的 WSDL 檔，以及在 WSDL 檔內參照的任何 XSD 檔。

#### 發佈人力作業 WSDL:

使用管理主控台來發佈 WSDL 檔。

1. 使用具有管理者權限的使用者 ID 登入管理主控台。
2. 選取**應用程式** → **SCA 模組**

**註:** 您也可以選取**應用程式** → **企業應用程式**，以顯示所有可用企業應用程式的清單。

3. 從 SCA 模組或應用程式的清單中，選擇 **TaskContainer** 應用程式。
4. 從**其他內容**的清單中，選取**發佈 WSDL 檔**
5. 按一下清單中的壓縮檔案。
6. 在出現的「檔案下載」視窗上，按一下**儲存**。
7. 瀏覽至本端資料夾，並按一下**儲存**。

匯出的壓縮檔案命名為 TaskContainer\_WSDLFiles.zip。此壓縮檔案包含可說明 Web 服務的 WSDL 檔，以及在 WSDL 檔內參照的任何 XSD 檔。

#### 匯出商業物件

商業程序及人力作業都有妥善定義的介面，可將它們當成 Web 服務進行外部存取。如果這些介面參照商業物件，則需要將介面定義及商業物件匯出至用戶端程式設計環境。

用戶端應用程式需要與之互動的每個商業物件，都需要重複此程序。

在 WebSphere Process Server 中，商業物件會定義與商業程序或人力作業互動的要求、回應及錯誤訊息格式。這些訊息也可以包含複式資料類型的定義。

例如，若要建立及啟動人力作業，則必須將下列資訊項目傳遞至作業介面：

- 作業範本名稱
- 作業範本名稱空間
- 輸入訊息 (包含格式化商業資料)
- 傳回回應訊息的回應封套
- 傳回錯誤及異常的錯誤訊息

這些項目是封裝在單一商業物件內。Web 服務介面的所有作業都已塑型為「document/literal 包裝」作業。這些作業的輸入及輸出參數都會封裝在封套文件中。其他商業物件則會定義對應的回應及錯誤訊息格式。

若要透過 Web 服務建立及啟動商業程序或人力作業，則用戶端上的用戶端應用程式必須可以使用這些封套物件。



將商業物件從 WebSphere 環境匯出為「Web 服務定義語言 (WSDL)」及「XML 綱目定義 (XSD)」檔，再將資料類型定義匯入至用戶端程式設計環境，然後將它們轉換為 Helper 類別供用戶端應用程式使用，就可以完成此作業。

1. 啟動「WebSphere Integration Developer 工作區」(如果尚未執行)。
2. 選取包含要匯出的商業物件的「程式庫」模組。「程式庫」模組是包含必要商業物件的壓縮檔。
3. 匯出「程式庫」模組。
4. 將匯出的檔案複製至用戶端應用程式開發環境。

## 範例

假設商業程序公開下列 Web 服務作業：

```
<wsdl:operation name="updateCustomer">
  <wsdl:input message="tns:updateCustomerRequestMsg"
    name="updateCustomerRequest"/>
  <wsdl:output message="tns:updateCustomerResponseMsg"
    name="updateCustomerResponse"/>
  <wsdl:fault message="tns:updateCustomerFaultMsg"
    name="updateCustomerFault"/>
</wsdl:operation>
```

而 WSDL 訊息定義為：

```
<wsdl:message name="updateCustomerRequestMsg">
  <wsdl:part element="types:updateCustomer"
    name="updateCustomerParameters"/>
</wsdl:message>
<wsdl:message name="updateCustomerResponseMsg">
  <wsdl:part element="types:updateCustomerResponse"
    name="updateCustomerResult"/>
</wsdl:message>
<wsdl:message name="updateCustomerFaultMsg">
  <wsdl:part element="types:updateCustomerFault"
    name="updateCustomerFault"/>
</wsdl:message>
```

必須在用戶端應用程式執行的所有一般作業 (call、sendMessage 等) 中，使用 <xsd:any> 參數從 Web 服務 API 傳遞及接收具體客戶定義元素 types:updateCustomer、types:updateCustomerResponse 及 types:updateCustomerFault。利用匯出的 XSD 檔產生的 Helper 類別，可以在用戶端應用程式端上建立、序列化及解除序列化這些客戶定義的元素。

### 相關工作

第 430 頁的『建立 BPEL 程序的 Helper 類別 (.NET)』

特定的 Web 服務 API 作業需要有用戶端應用程式，才能使用「document/literal 包裝」樣式元素。用戶端應用程式需要 Helper 類別，協助它們產生必要的封套元素。

第 430 頁的『建立 BPEL 程序的 Helper 類別 (.NET)』

特定的 Web 服務 API 作業需要有用戶端應用程式，才能使用「document/literal 包裝」樣式元素。用戶端應用程式需要 Helper 類別，協助它們產生必要的封套元素。

第 424 頁的『建立 BPEL 程序的 Helper 類別 (Java Web 服務)』

具體 API 要求 (例如, sendMessage 或 call) 中參照的商業物件, 需要有用戶端應用程式才能使用「document/literal 包裝」樣式元素。用戶端應用程式需要有 Helper 類別, 協助它們產生必要的封套元素。

## 使用用戶端 CD 上的檔案

作為從 WebSphere 伺服器環境匯出成品的替代方案, 您可以從 WebSphere Process Server 用戶端 CD 複製產生用戶端應用程式所需的檔案。

在此情況下, 您必須手動修改「商業流程管理程式」API 或「人力作業管理程式」API 的預設 Web 服務端點位址。

如果用戶端應用程式要存取這兩個 API, 則您必須編輯這兩個 API 的預設端點位址。

## 複製用戶端 CD 中的檔案

WebSphere Process Server 用戶端 CD 內含存取 Web 服務 API 的必要檔案。

1. 存取用戶端 CD, 並瀏覽至 ProcessChoreographer\client 目錄。
2. 將必要的檔案複製至用戶端應用程式開發環境。

如果是「商業流程管理程式」API, 請複製:

### **BFMWS.wsdl**

說明「商業流程管理程式」Web 服務 API 中可用的 Web 服務。此檔案包含端點位址。

### **BFMIF.wsdl**

說明「商業流程管理程式」Web 服務 API 中每個 Web 服務的參數及資料類型。

### **BFMIF.xsd**

說明「商業流程管理程式」Web 服務 API 中使用的資料類型。

### **BPCGEN.xsd**

包含「商業流程管理程式」與「人力作業管理程式」Web 服務 API 之間常用的資料類型。

如果是「人力作業管理程式」API, 請複製:

### **HTMWS.wsdl**

說明「人力作業管理程式」Web 服務 API 中可用的 Web 服務。此檔案包含端點位址。

### **HTMIF.wsdl**

說明「人力作業管理程式」Web 服務 API 中每個 Web 服務的參數及資料類型。

### **HTMIF.xsd**

說明「人力作業管理程式」Web 服務 API 中使用的資料類型。

### **BPCGEN.xsd**

包含「商業流程管理程式」與「人力作業管理程式」Web 服務 API 之間常用的資料類型。

註: BPCGen.xsd 檔適用於這兩種 API。

複製檔案之後，您必須手動將 BFMWS.wsdl 或 HTMWWS.wsdl 檔的 Web 服務 API 端點位址，變更為 Web 服務 API 所在的 WebSphere 應用程式伺服器的 Web 服務 API 端點位址。

#### 相關工作

『手動變更 Web 服務端點位址』

如果從用戶端 CD 複製檔案，則必須將 WSDL 檔中指定的預設 Web 服務端點位址，變更為 Web 服務 API 所在伺服器的 Web 服務端點位址。

第 413 頁的『從伺服器環境中發佈及匯出成品』

可以開發用戶端應用程式以存取 Web 服務 API 之前，必須先從 WebSphere 伺服器環境中發佈及匯出一些成品。

### 手動變更 Web 服務端點位址

如果從用戶端 CD 複製檔案，則必須將 WSDL 檔中指定的預設 Web 服務端點位址，變更為 Web 服務 API 所在伺服器的 Web 服務端點位址。

您可以使用管理主控台來設定 Web 服務端點位址，然後再匯出 WSDL 檔。然而，如果是從 WebSphere Process Server 用戶端 CD 複製 WSDL 檔，則必須手動修改預設 Web 服務端點位址。

使用的 Web 服務端點位址是根據 WebSphere 伺服器配置而定：

- 範例情況 1。有單一 WebSphere 伺服器。要指定的 WebSphere 端點位址是伺服器的主機名稱及埠號 (例如，**host1:9080**)。
- 範例情況 2。由數部伺服器組成的 WebSphere 叢集。要指定的 WebSphere 端點位址是 Web 服務 API 所在伺服器的主機名稱及埠 (例如，**host2:9081**)。
- 範例情況 3。一部 Web 伺服器作為前端系統。要指定的 WebSphere 端點位址則是該 Web 伺服器的主機名稱及埠 (例如：**host:80**)。

#### 相關工作

第 418 頁的『複製用戶端 CD 中的檔案』

WebSphere Process Server 用戶端 CD 內含存取 Web 服務 API 的必要檔案。

#### 變更商業流程管理程式 API 端點:

如果從 WebSphere Process Server 用戶端 CD 複製「商業流程管理程式」API 檔，則必須手動編輯預設端點位址。

1. 導覽至目錄，而此目錄內含從用戶端 CD 複製的檔案。
2. 使用文字編輯器或 XML 編輯器來開啓 BFMWS.wsdl 檔。
3. 尋找 soap:address 元素 (直到檔案底端)。
4. 使用在其上執行 Web 服務 API 的伺服器的 HTTP URL，來修改 location 屬性的值。若要執行此作業：
  - a. 選擇性地將 http 取代為 https，以使用較安全的 HTTPS 通訊協定。
  - b. 將 localhost 取代為 Web 服務 API 伺服器端點位址的主機名稱或 IP 位址。
  - c. 將 9080 取代為應用程式伺服器的埠號。
  - d. 將 *BPEContainer\_N1\_server1* 取代為執行 Web 服務 API 的應用程式的環境定義根目錄。預設環境定義根目錄包含：
    - *BPEContainer*。應用程式名稱。

- *NI*。節點名稱。
  - *server1*。伺服器名稱。
- e. 不要修改 URL 的固定部分 (/sca/com/ibm/bpe/api/BFMWS)。

例如，如果應用程式是在伺服器 **s1.n1.ibm.com** 上執行，而且伺服器正在埠 **9080** 上接受 SOAP/HTTP 要求，請修改 soap:address 元素，如下所示：

```
<soap:address location="http://s1.n1.ibm.com:9080/
                    BPEContainer_N1_server1/sca/com/ibm/bpe/api/BFMWS"/>
```

#### 相關概念

第 426 頁的『新增安全 (Java Web 服務)』

您必須透過在用戶端應用程式中實作安全機制，來維護 Web 服務通訊的安全。

#### 相關工作

第 433 頁的『新增安全 (.NET)』

您可以將安全機制整合至用戶端應用程式，以保護 Web 服務通訊的安全。

### 變更人力作業管理程式 API 端點:

如果從 WebSphere Process Server 用戶端 CD 複製「人力作業管理程式 API」檔，則必須手動編輯預設端點位址。

1. 導覽至目錄，而此目錄內含從用戶端 CD 複製的檔案。
2. 使用文字編輯器或 XML 編輯器來開啓 HTMWWS.wsdl 檔。
3. 尋找 soap:address 元素 (直到檔案底端)。
4. 將 location 屬性的值修改為正確的端點位址。若要執行此作業：
  - a. 選擇性地將 http 取代為 https，以使用較安全的 HTTPS 通訊協定。
  - b. 將 localhost 取代為 Web 服務 API 伺服器端點位址的主機名稱或 IP 位址。
  - c. 將 9080 取代為應用程式伺服器的埠號。
  - d. 將 *HTMContainer\_N1\_server1* 取代為執行 Web 服務 API 的應用程式的環境定義根目錄。預設環境定義根目錄包含：
    - *HTMContainer*。應用程式名稱。
    - *NI*。節點名稱。
    - *server1*。伺服器名稱。
  - e. 不要修改 URL 的固定部分 (/sca/com/ibm/task/api/HTMWWS)。

例如，如果應用程式是在伺服器 **s1.n1.ibm.com** 上執行，而且伺服器正在埠 **9081** 上接受 SOAP/HTTPS 要求，請修改 soap:address 元素，如下所示：

```
<soap:address location="https://s1.n1.ibm.com:9081/
                    HTMContainer_N1_server1/sca/com/ibm/task/api/HTMWWS"/>
```

#### 相關概念

第 426 頁的『新增安全 (Java Web 服務)』

您必須透過在用戶端應用程式中實作安全機制，來維護 Web 服務通訊的安全。

#### 相關工作

第 433 頁的『新增安全 (.NET)』

您可以將安全機制整合至用戶端應用程式，以保護 Web 服務通訊的安全。

---

## 在 Java Web 服務環境中開發用戶端應用程式

您可以使用與 Java Web 服務相容的 Java 型開發環境，來開發 Web 服務 API 的用戶端應用程式。

### 產生 Proxy 用戶端 (Java Web 服務)

Java Web 服務用戶端應用程式使用 *Proxy* 用戶端與 Web 服務 API 互動。

Java Web 服務的 Proxy 用戶端包含一些 Java Bean 類別，而用戶端應用程式會呼叫這些類別來執行 Web 服務要求。Proxy 用戶端會將服務參數的組合處理為 SOAP 訊息、透過 HTTP 將 SOAP 訊息傳送給 Web 服務、接收來自 Web 服務的回應，以及將任何傳回的資料傳遞給用戶端應用程式。

因此，基本上，Proxy 用戶端容許用戶端應用程式像本端功能一樣地呼叫 Web 服務。

**註：**您只需要產生一次 Proxy 用戶端。存取相同 Web 服務 API 的所有用戶端應用程式，接著就可以使用同一個 Proxy 用戶端。

在 IBM Web 服務環境中，有兩種方式可以產生 Proxy 用戶端：

- 使用 Rational® Application Developer 或 WebSphere Integration Developer 整合開發環境。
- 使用 WSDL2Java 指令行工具。

其他 Java Web 服務開發環境通常會包括 WSDL2Java 工具或專有用戶端應用程式產生機能。

### 使用 Rational Application Developer 產生 Proxy 用戶端

Rational Application Developer 整合開發環境可讓您產生用戶端應用程式的 Proxy 用戶端。

產生 Proxy 用戶端之前，必須事先從 WebSphere 環境 (或 WebSphere Process Server 用戶端 CD) 匯出可說明商業程序或人力作業 Web 服務介面的 WSDL 檔，並將這些檔案複製至用戶端程式設計環境。

1. 將適當的 WSDL 檔新增至專案：

- 若為商業程序：
  - a. 將匯出檔 `BPEContainer_nodename_servername_WSDLFiles.zip` 解壓縮至暫存目錄。
  - b. 從解壓縮的目錄 `BPEContainer_nodename_servername.ear/b.jar` 匯入子目錄 `META-INF`。
- 若為人力作業：
  - a. 將匯出檔 `TaskContainernodename_servername_WSDLFiles.zip` 解壓縮至暫存目錄。
  - b. 從解壓縮的目錄 `TaskContainer_nodename_servername.ear/h.jar` 匯入子目錄 `META-INF`。

在您的專案中，會建立新的目錄 `wsdl` 及子目錄結構。

2. 修改「Web 服務」精靈內容：

- a. 在 Rational Application Developer 中，選擇**喜好設定** → **Web 服務** → **產生程式碼** → **IBM WebSphere 執行時期**。
- b. 選取**使用無包裝樣式從 WSDL 產生 Java** 選項。

**註:** 如果您無法在**喜好設定**功能表中選取 **Web 服務**選項，則必須先啓用必要的功能，如下所示：**視窗** → **喜好設定** → **工作台** → **功能**。按一下 **Web 服務開發人員**，再按一下**確定**。然後，重新開啓「**喜好設定**」視窗，並變更**產生程式碼**選項。

3. 選取位於新建立 wsdl 目錄的 BFMWS.WSDL 或 HTMWS.WSDL 檔。
4. 按一下滑鼠右鍵並選擇 **Web 服務** → **產生用戶端**。

在繼續剩餘的步驟之前，請確保已啓動伺服器。

5. 在「**Web 服務**」視窗上，按**下一步**以接受所有預設值。
6. 在「**Web 服務選擇**」視窗上，按**下一步**以接受所有預設值。
7. 在「**用戶端環境配置**」視窗上：
  - a. 按一下**編輯**，並將 **Web 服務執行時期**選項變更為 **IBM WebSphere**
  - b. 將 **J2EE 版本**選項變更為 **1.4**。
  - c. 按一下**確定**。
  - d. 按**下一步**。
8. 只在您需要產生一個「**Web 服務**」用戶端，以同時包括「**商業程序**」及「**人力作業**」的「**Web 服務 API**」時，才需要這個步驟，因為在這兩個 **WSDL** 檔案中有重複的方法。
  - a. 在「**Web 服務 Proxy**」視窗上，選取定義名稱空間與套件的自訂對映，然後按一下**確定**。
  - b. 在「**Web 服務用戶端名稱空間與套件對映**」視窗上，新增下列名稱空間及套件：

若為 BFMWS.wsdl：

名稱空間	套件
http://www.ibm.com/xmlns/prod/websphere/business-process/types/6.0	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/business-process/services/6.0	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/business-process/services/6.0/ Binding	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/bpc-common/types/6.0	com.ibm.sca.bpe

若為 HTMWS.wsdl：

名稱空間	套件
http://www.ibm.com/xmlns/prod/websphere/human-task/types/6.0	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/human-task/services/6.0	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/human-task/services/6.0/ Binding	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/bpc-common/types/6.0	com.ibm.sca.task

如果要求確認改寫，請按一下 **YesToAll**。

9. 按一下**完成**。



會產生 Proxy 用戶端 (由一些 Proxy、定位器及 Helper Java 類別組成)，並將之新增至專案中。也會更新部署描述子。

## 使用 WSDL2Java 產生 Proxy 用戶端

WSDL2Java 是產生 Proxy 用戶端的命令行工具。Proxy 用戶端簡化了用戶端應用程式的程式設計。

產生 Proxy 用戶端之前，必須事先從 WebSphere 環境 (或 WebSphere Process Server 用戶端 CD) 匯出可說明商業程序或人力作業 Web 服務 API 的 WSDL 檔，並將這些檔案複製至用戶端程式設計環境。

1. 使用 WSDL2Java 工具產生 Proxy 用戶端：鍵入：

```
wSDL2java options WSDLfilepath
```

其中：

- *options* 包括：

**-noWrappedOperations (-w)**

停用包裝作業的偵測。會產生要求及回應訊息的 Java Bean。

註：這不是預設值。

**-role (-r)**

指定值 **client** 來產生檔案及連結檔，以進行用戶端開發。

**-container (-c)**

要使用的用戶端儲存器。有效引數包括：

**client** 用戶端儲存器

**ejb** Enterprise JavaBeans (EJB) 儲存器。

**none** 沒有儲存器

**Web** Web 儲存器

**-output (-o)**

在其中儲存所產生檔案的資料夾。

如需完整的 WSDL2Java 參數清單，請使用 **-help** 命令行參數，或參閱 WID/RAD 中 WSDL2Java 工具的線上說明。

- *WSDLfilepath* 是從 WebSphere 環境匯出或從用戶端 CD 複製的 WSDL 檔的路徑及檔名。

下列範例會產生「人力作業活動」Web 服務 API 的 Proxy 用戶端：

```
call wSDL2java.bat -r client -c client -noWrappedOperations  
-output c:\ws\proxyClient c:\ws\bin\HTMWS.wsdl
```

2. 將產生的類別檔案併入至專案中。

### 相關工作

第 425 頁的『建立用戶端應用程式 (Java Web 服務)』

用戶端應用程式會將要求傳送給 Web 服務 API，以及接收來自 Web 服務 API 的回應。使用 Proxy 用戶端管理通訊及 Helper 類別來格式化複式資料類型，用戶端應用程式就可以像本端功能一樣地呼叫 Web 服務方法。



## 建立 BPEL 程序的 Helper 類別 (Java Web 服務)

具體 API 要求 (例如, `sendMessage` 或 `call`) 中參照的商業物件, 需要有用戶端應用程式才能使用「document/literal 包裝」樣式元素。用戶端應用程式需要有用戶端 Helper 類別, 協助它們產生必要的封套元素。

若要建立 Helper 類別, 則必須已從 WebSphere Process Server 環境中匯出 Web 服務 API 的 WSDL 檔。

Web 服務 API 的 `call()` 及 `sendMessage()` 作業, 容許與 WebSphere Process Server 上的 BPEL 程序互動。`call()` 作業的輸入訊息需要提供程序輸入訊息的 document/literal 封套。

有一些可能的技術可以產生 BPEL 程序或人力作業的 Helper 類別, 包括:

### 1. 使用 SoapElement 物件。

在 WebSphere Integration Developer 可用的 Rational Application Developer 環境中, Web 服務引擎支援 JAX-RPC 1.1。在 JAX-RPC 1.1 中, SoapElement 物件會擴充「文件物件模型 (DOM)」元素, 因此可能會使用 DOM API 來建立、讀取、載入及儲存 SOAP 訊息。

例如, 假設 WSDL 檔包含工作流程程序或人力作業的下列輸入訊息:

```
<xsd:element name="operation1">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="input1" nillable="true" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

開發程序或人力作業模組時, 會建立 WSDL 檔案。

若要在您的用戶端應用程式中使用 DOM API 來建立對應的 SOAP 訊息:

```
SOAPFactory soapfactoryinstance = SOAPFactory.newInstance();
SOAPElement soapmessage = soapfactoryinstance.createElement
    ("operation1", namespaceprefix, interfaceURI);
SOAPElement inputelement = soapfactoryinstance.createElement("input1");
inputelement.addTextNode( message value);
soapmessage.addChildElement(outputelement);
```

下列範例顯示如何在您的用戶端應用程式中建立 `sendMessage` 作業的輸入參數:

```
SendMessage inWsend = new SendMessage();
inWsend.setProcessTemplateName(processTemplateName);
inWsend.setPortType(portType);
inWsend.setOperation(operationName);
inWsend.set_any(soapmessage);
```

### 2. 使用「WebSphere 自訂資料連結」功能。


下列 developerWorks 文章會說明此技術:


- 如何選擇 Web 服務的自訂對映技術
- 使用 EMF SDO 開發複式 XML 綱目的 Web 服務

**相關工作**

第 416 頁的『匯出商業物件』

商業程序及人力作業都有妥善定義的介面，可將它們當成 Web 服務進行外部存取。如果這些介面參照商業物件，則需要將介面定義及商業物件匯出至用戶端程式設計環境。

 與文件型 Web 服務型樣及策略的交互作業能力

 內含選用 JAX-RPC 1.0/1.1 XML 綱目類型的綱目/WSDL 的「Web 服務」支援

## 建立用戶端應用程式 (Java Web 服務)

用戶端應用程式會將要求傳送給 Web 服務 API，以及接收來自 Web 服務 API 的回應。使用 Proxy 用戶端管理通訊及 Helper 類別來格式化複式資料類型，用戶端應用程式就可以像本端功能一樣地呼叫 Web 服務方法。

開始建立用戶端應用程式之前，請產生 Proxy 用戶端及任何必要的 Helper 類別。

您可以使用任何 Web 服務相容的開發工具 (例如，IBM Rational Application Developer (RAD)) 來開發用戶端應用程式。您可以建置任何類型的 Web 服務應用程式來呼叫 Web 服務 API。

1. 建立新的用戶端應用程式專案。
2. 產生 Proxy 用戶端，並將 Java Helper 類別新增至專案中。
3. 撰寫用戶端應用程式的程式碼。
4. 建置專案。
5. 執行用戶端應用程式。

下列範例顯示如何使用「商業流程管理程式」Web 服務 API。

```
// create the proxy
    BFMIFProxy proxy = new BFMIFProxy();
// prepare the input data for the operation
    GetProcessTemplate iw = new GetProcessTemplate();
    iw.setIdentifier(your_process_template_name);

// invoke the operation
    GetProcessTemplateResponse ow = proxy.getProcessTemplate(iw);

// process output of the operation
    ProcessTemplateType ptd = ow.getProcessTemplate();
    System.out.println("getName= " + ptd.getName());
    System.out.println("getPtid= " + ptd.getPtid());
```

### 相關工作

第 421 頁的『產生 Proxy 用戶端 (Java Web 服務)』

Java Web 服務用戶端應用程式使用 Proxy 用戶端與 Web 服務 API 互動。

第 424 頁的『建立 BPEL 程序的 Helper 類別 (Java Web 服務)』

具體 API 要求 (例如，sendMessage 或 call) 中參照的商業物件，需要有用戶端應用程式才能使用「document/literal 包裝」樣式元素。用戶端應用程式需要有 Helper 類別，協助它們產生必要的封套元素。

第 423 頁的『使用 WSDL2Java 產生 Proxy 用戶端』

WSDL2Java 是產生 Proxy 用戶端的指令行工具。Proxy 用戶端簡化了用戶端應用程式的程式設計。

## 新增安全 (Java Web 服務)

您必須透過在用戶端應用程式中實作安全機制，來維護 Web 服務通訊的安全。

WebSphere Application Server 目前支援下列 Web 服務 API 的安全機制：

- 使用者名稱記號
- 小型認證機構 (LTPA)

### 相關概念

第 57 頁的『人力作業的授權角色』

您可以針對人力作業執行的動作，視授權角色而定。此角色可以是系統層次 J2EE 角色或實例型角色。

第 28 頁的『商業程序的授權角色』

角色是指具有相同授權層次的一組人員。您可以針對商業程序執行的動作，視您的授權角色而定。此角色可以是 J2EE 角色或實例型角色。

### 實作使用者名稱記號

使用者名稱記號安全機制提供使用者名稱及密碼認證。

使用使用者名稱記號安全機制，就可以選擇實作各種回呼處理程式。而根據您的選擇：

- 每次執行用戶端應用程式時，系統會提示您所要提供的使用者名稱及密碼。
- 使用者名稱及密碼會寫入至部署描述子。

在上述任一情況下，所提供的使用者名稱及密碼必須符合對應商業程序儲存器或人力作業儲存器中授權角色的使用者名稱及密碼。

使用者名稱及密碼會封裝在要求訊息信封中，因此在 SOAP 訊息標頭中會顯示「明碼」。因此，強烈建議您將用戶端應用程式配置成使用 HTTPS (透過 SSL 的 HTTP) 通訊協定。這樣會加密所有通訊。指定 Web 服務 API 的端點 URL 位址時，可以選取 HTTPS 通訊協定。

若要定義使用者名稱記號：

1. 建立安全記號：
  - a. 開啓您模組的部署編輯器
  - b. 按一下 **WS 延伸** 標籤。
  - c. 在**服務參照**下，可能會列出下列「Web 服務參照」：
    - service/BFMWSService (適用於商業程序)
    - service/HTMWSService (適用於人力作業)

列出哪些項目視產生 Proxy 用戶端時是新增 BFMWS.wsdl (適用於商業程序)、HTMWWS.wsdl (適用於人力作業) 或兩者而定。

- d. 若為兩個服務參照：
  - 1) 選取其中一個**服務參照**。
  - 2) 展開**要求產生器配置**區段。
  - 3) 展開**安全記號**子區段。
  - 4) 按一下**新增**。此時會開啓「安全記號」視窗。


- 5) 在**名稱**欄位中，鍵入新安全記號的名稱：**UserNameTokenBFM** 或 **UserNameTokenHTM**。
  - 6) 在**記號類型**下拉清單中，選取**使用者名稱**。(本端名稱欄位會自動移入預設值。)
  - 7) 讓 **URI** 欄位保留空白。使用者名稱記號不需要任何 URI 值。
  - 8) 按一下**確定**。
2. 建立記號產生器：
- a. 開啓您模組的**部署編輯器**
  - b. 按一下 **WS 連結**標籤
  - c. 在**服務參照**下，會列出與前一個步驟相同的「Web 服務參照」：
    - service/BFMWSService (適用於商業程序)
    - service/HTMWSService (適用於人力作業)
  - d. 若為兩個服務參照：
    - 1) 選取其中一個**服務參照**。
    - 2) 展開**安全要求產生者連結配置**區段。
    - 3) 展開**記號產生器**子區段。
    - 4) 按一下**新增**。此時會開啓「記號產生器」視窗。
    - 5) 在**名稱**欄位中，鍵入新記號產生器的名稱，例如 "UserNameTokenGeneratorBFM" 或 "UserNameTokenGeneratorHTM"。
    - 6) 在**記號產生器類別**欄位中，確保已選取下列記號產生器類別：**com.ibm.wsspi.wssecurity.token.UsernameTokenGenerator**。
    - 7) 在**安全記號**下拉清單中，選取您早先建立的適當安全記號。
    - 8) 選取**使用值類型**勾選框。
    - 9) 在**值類型**欄位中，選取 **使用者名稱記號**。(本端名稱欄位會自動移入值，以反映您選擇的**使用者名稱記號**。)
    - 10) 在**回呼處理程式**欄位中，鍵入 "com.ibm.wsspi.wssecurity.auth.callback.GUIPromptCallbackHandler" (會在執行用戶端應用程式時，提示您輸入使用者名稱及密碼) 或 "com.ibm.wsspi.wssecurity.auth.callback.NonPromptCallbackHandler"。
    - 11) 如果選擇 **NonPromptCallbackHandler**，則必須在部署描述子的對應欄位中指定有效的使用者名稱及密碼。
    - 12) 按一下**確定**。

### 相關工作

第 414 頁的『指定 Web 服務端點位址』

Web 服務端點位址是用戶端應用程式必須指定才能存取 Web 服務 API 的 URL。端點位址會寫入至 WSDL 檔中，而此檔案是您匯出以產生用於用戶端應用程式的 Proxy 用戶端。

### 相關資訊

 IBM WebSphere Developer 技術異動日誌：WebSphere Application Server 第 6 版的 Web 服務安全

## 實作 LTPA 安全機制

用戶端應用程式在先前建立的安全環境定義內執行時，可以使用「小型認證機構 (LTPA)」安全機制。

只有在用戶端應用程式於已建立安全環境定義的安全環境中執行時，才可以使用 LTPA 安全機制。例如，如果用戶端應用程式是在 Enterprise JavaBeans (EJB) 儲存器中執行，則 EJB 用戶端必須先登入，才能呼叫用戶端應用程式。然後會建立安全環境定義。如果 EJB 用戶端應用程式接著呼叫 Web 服務，則 LTPA 回呼處理程式會從安全環境定義中擷取 LTPA 記號，並將它新增至 SOAP 要求訊息中。在伺服器端上，LTPA 機制會處理 LTPA 記號。

若要實作 LTPA 安全機制：

1. 在 WebSphere Integration Developer 可用的 Rational Application Developer 環境中，選擇 **WS 連結** → **安全要求產生器連結配置** → **記號產生器**。
2. 建立安全記號：
  - a. 開啓您模組的**部署編輯器**
  - b. 按一下 **WS 延伸**標籤。
  - c. 在**服務參照**下，可能會列出下列 **Web 服務參照**：
    - service/BFMWSService (適用於商業程序)
    - service/HTMWSService (適用於人力作業)列出哪些項目視產生 Proxy 用戶端時是新增 BFMWS.wsdl (適用於商業程序)、HTMWWS.wsdl (適用於人力作業) 或兩者而定。
  - d. 若為兩個服務參照：
    - 1) 選取其中一個**服務參照**。
    - 2) 展開**要求產生器配置**區段。
    - 3) 展開**安全記號**子區段。
    - 4) 按一下**新增**。此時會開啓「安全記號」視窗。
    - 5) 在**名稱**欄位中，鍵入新安全記號的名稱：**LTPATokenBFM** 或 **LTPATokenHTM**。
    - 6) 在**記號類型**下拉清單中，選取 **LTPAToken**。(URI 及本端名稱欄位會自動移入預設值。)
    - 7) 按一下**確定**。
3. 建立記號產生器：
  - a. 開啓您模組的**部署編輯器**
  - b. 按一下 **WS 連結**標籤
  - c. 在**服務參照**下，會列出與前一個步驟相同的「Web 服務參照」：
    - service/BFMWSService (適用於商業程序)
    - service/HTMWSService (適用於人力作業)
  - d. 若為兩個服務參照：
    - 1) 選取其中一個**服務參照**。
    - 2) 展開**安全要求產生者連結配置**區段。
    - 3) 展開**記號產生器**子區段。

- 4) 按一下**新增**。此時會開啓「記號產生器」視窗。
- 5) 在**名稱**欄位中，鍵入新記號產生器的名稱，例如 "LTPATokenGeneratorBFM" 或 "LTPATokenGeneratorHTM"。
- 6) 在**記號產生器類別**欄位中，確保已選取下列記號產生器類別：  
**com.ibm.wsspi.wssecurity.token.LTPATokenGenerator**。
- 7) 在**安全記號**下拉清單中，選取您早先建立的適當安全記號。
- 8) 選取**使用值類型**勾選框。
- 9) 在**值類型**欄位中，選取 **LTPAToken**。(URI 及本端名稱欄位會自動移入值，以反映您選擇的 **LTPA** 記號。)
- 10) 在**回呼處理程式**欄位中，鍵入  
"com.ibm.wsspi.wssecurity.auth.callback.LTPATokenCallbackHandler"。
- 11) 按一下**確定**。

在執行時期，**LTPATokenCallbackHandler** 會從現有的安全環境定義中擷取 LTPA 記號，並將它新增至 SOAP 要求訊息中。

## 新增交易支援 (Java Web 服務)

Java Web 服務用戶端應用程式可以配置成容許伺服器端要求處理來參與用戶端的交易，方式是將用戶端應用程式環境定義傳遞為服務要求的一部分。此單位交易支援是以「Web 服務-單位交易 (WS-AT)」規格進行定義。

WebSphere Application Server 會將每個 Web 服務 API 要求執行為不同的單位交易。您可以利用下列其中一種方式，將用戶端應用程式配置成使用交易支援：

- 參與交易。伺服器端要求處理是在用戶端應用程式交易環境定義內執行。然後，如果伺服器在 Web 服務 API 要求執行時發現問題，並進行回復，則也會回復用戶端應用程式的要求。
- 不使用交易支援。WebSphere Application Server 仍然會建立在其中執行要求的新交易，但是不會使用用戶端應用程式交易環境定義來執行伺服器端要求處理。

---

## 在 .NET 環境中開發用戶端應用程式

Microsoft .NET 提供功能強大的開發環境，可在其中透過 Web 服務連接應用程式。

### 產生 Proxy 用戶端 (.NET)

.NET 用戶端應用程式會使用 *Proxy 用戶端* 與 Web 服務 API 互動。Proxy 用戶端會保護用戶端應用程式，使其不像 Web 服務傳訊通訊協定那樣複雜。

若要建立 Proxy 用戶端，則必須先從 WebSphere 環境中匯出一些 WSDL 檔，並將這些檔案複製至用戶端程式設計環境。

**註：**如果您擁有 WebSphere Process Server 用戶端 CD，則可以改為從該 CD 中複製檔案。

Proxy 用戶端包含一組 C# Bean 類別。每個類別都會包含單一 Web 服務公開的所有方法及物件。service 方法會將參數的組合處理為完整的 SOAP 訊息、透過 HTTP 將 SOAP 訊息傳送給 Web 服務、接收來自 Web 服務的回應，以及處理任何傳回的資料。



**註:** 您只需要產生一次 Proxy 用戶端。存取 Web 服務 API 的所有用戶端應用程式接著就可以使用同一個 Proxy 用戶端。

1. 使用 WSDL 指令產生 Proxy 用戶端：鍵入：

```
wSDL options WSDLfilepath
```

其中：

- *options* 包括：

#### **/language**

可讓您指定用來建立 Proxy 類別的語言。預設值是 C#。您也可以指定 **VB** (Visual Basic)、**JS** (JScript) 或 **VJS** (Visual J#) 作為 language 引數。

#### **/output**

輸出檔的名稱 (具有適當的字尾)。例如，proxy.cs

#### **/protocol**

Proxy 類別中實作的通訊協定。**SOAP** 是預設值。

如需完整的 **WSDL.exe** 參數清單，請使用 */?* 指令行參數，或參閱 Visual Studio 中 WSDL 工具的線上說明。

- *WSDLfilepath* 是從 WebSphere 環境匯出或從用戶端 CD 複製的 WSDL 檔的路徑及檔名。

下列範例會產生「人力作業管理程式」Web 服務 API 的 Proxy 用戶端：

```
wSDL /language:cs /output:proxycient.cs c:\ws\bin\HTMWS.wSDL
```

2. 將 Proxy 用戶端編譯為「動態鏈結程式庫 (DLL)」檔案。

#### **相關工作**

第 415 頁的『發佈 WSDL 檔』

「Web 服務定義語言 (WSDL)」檔包含使用 Web 服務 API 可以進行的所有作業的詳細說明。「商業流程管理程式」與「人力作業管理程式」Web 服務 API 各有不同的 WSDL 檔。您必須先發佈這些 WSDL 檔，然後再將它們從 WebSphere 環境複製至開發環境，而它們在該環境中是用來產生 Proxy 用戶端。

## **建立 BPEL 程序的 Helper 類別 (.NET)**

特定的 Web 服務 API 作業需要有用戶端應用程式，才能使用「document/literal 包裝」樣式元素。用戶端應用程式需要有 Helper 類別，協助它們產生必要的封套元素。

若要建立 Helper 類別，則必須已從 WebSphere Process Server 環境中匯出 Web 服務 API 的 WSDL 檔。

Web 服務 API 的 call() 及 sendMessage() 作業，讓 BPEL 程序可以在 WebSphere Process Server 內啟動。call() 作業的輸入訊息需要提供 BPEL 程序輸入訊息的 document/literal 封套。若要產生 BPEL 程序的必要 Bean 及類別，請將 <wsdl:types> 元素複製至新的 XSD 檔，然後使用 xsd.exe 工具產生 Helper 類別。

1. 如果尚未做這樣的處理，請從 WebSphere Integration Developer 中匯出 BPEL 程序介面的 WSDL 檔。
2. 使用文字編輯器或 XML 編輯器來開啓 WSDL 檔。



- 複製 `<wsdl:types>` 元素的所有子元素的內容，並將該內容貼入全新的基本 XSD 檔中。
- 在 XSD 檔上，執行 `xsd.exe` 工具：

**call xsd.exe file.xsd /classes /o**

其中：

**file.xsd**

要轉換的「XML 綱目定義」檔。

**/classes (/c)**

產生對應於所指定 XSD 檔內容的 Helper 類別。

**/output (/o)**

指定所產生檔案的輸出目錄。如果省略此目錄，則預設值會是現行目錄。

例如：

**call xsd.exe ProcessCustomer.xsd /classes /output:c:\temp**

- 將產生的類別檔案新增至用戶端應用程式。例如，如果使用 Visual Studio，則可以使用 **專案** → **新增現有項目** 功能表選項來完成這個作業。

如果 `ProcessCustomer.wsdl` 檔案包含下列項目：

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:bons1="http://com/ibm/bpe/unittest/sca"
  xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="ProcessCustomer"
  targetNamespace="http://ProcessTypes/bpel/ProcessCustomer">
  <wsdl:types>
    <xsd:schema targetNamespace="http://ProcessTypes/bpel/ProcessCustomer"
      xmlns:bons1="http://com/ibm/bpe/unittest/sca"
      xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:import namespace="http://com/ibm/bpe/unittest/sca"
        schemaLocation="xsd-includes/http.com.ibm.bpe.unittest.sca.xsd"/>
      <xsd:element name="doit">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true" type="bons1:Customer"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="doitResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true" type="bons1:Customer"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="doitRequestMsg">
    <wsdl:part element="tns:doit" name="doitParameters"/>
  </wsdl:message>
  <wsdl:message name="doitResponseMsg">
    <wsdl:part element="tns:doitResponse" name="doitResult"/>
  </wsdl:message>
  <wsdl:portType name="ProcessCustomer">
```

```

    <wsdl:operation name="doit">
      <wsdl:input message="tns:doitRequestMsg" name="doitRequest"/>
      <wsdl:output message="tns:doitResponseMsg" name="doitResponse"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>

```

產生的 XSD 檔包含：

```

<xsd:schema xmlns:bons1="http://com/ibm/bpe/unittest/sca"
            xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://ProcessTypes/bpel/ProcessCustomer">
  <xsd:import namespace="http://com/ibm/bpe/unittest/sca"
            schemaLocation="Customer.xsd"/>
  <xsd:element name="doit">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="input1" type="bons1:Customer" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="doitResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="output1" type="bons1:Customer" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

### 相關工作

第 416 頁的『匯出商業物件』

商業程序及人力作業都有妥善定義的介面，可將它們當成 Web 服務進行外部存取。如果這些介面參照商業物件，則需要將介面定義及商業物件匯出至用戶端程式設計環境。

### 相關資訊

 XML 綱目定義工具 (XSD.EXE) 的 Microsoft 文件

## 建立用戶端應用程式 (.NET)

用戶端應用程式會將要求傳送給 Web 服務 API，以及接收來自 Web 服務 API 的回應。使用 Proxy 用戶端管理通訊及 Helper 類別來格式化複式資料類型，用戶端應用程式就可以像本端功能一樣地呼叫 Web 服務方法。

開始建立用戶端應用程式之前，請產生 Proxy 用戶端及任何必要的 Helper 類別。

您可以使用任何 .NET 相容的開發工具 (例如，Visual Studio .NET) 來開發 .NET 用戶端應用程式。您可以建置任何類型的 .NET 應用程式來呼叫一般 Web 服務 API。

1. 建立新的用戶端應用程式專案。例如，在 Visual Studio 中建立 **WinFX Windows 應用程式**。
2. 在專案選項中，新增 Proxy 用戶端的「動態鏈結程式庫 (DLL)」檔案的參照。將內含商業物件定義的所有 Helper 類別新增至專案。例如，在 Visual Studio 中，可以使用 **專案** → **新增現有項目** 選項來完成這個作業。
3. 建立 Proxy 用戶端物件。例如：

```

HTMClient.HTMReference.HumanTaskManagerComponent1Export_HumanTaskManagerHttpService service =
    new HTMClient.HTMReference.HumanTaskManagerComponent1Export_HumanTaskManagerHttpService();

```

4. 宣告要在 Web 服務內傳送或接收的訊息中使用的所有商業物件資料類型。例如：

```
HTMClient.HTMReference.TKIID id = new HTMClient.HTMReference.TKIID();

ClipBG bg = new ClipBG();
Clip clip = new Clip();
```
5. 呼叫特定 Web 服務功能，並指定所有必要參數。例如，若要建立及啟動人力作業：

```
HTMClient.HTMReference.createAndStartTask task =
    new HTMClient.HTMReference.createAndStartTask();
HTMClient.HTMReference.StartTask sTask = new HTMClient.HTMReference.StartTask();

sTask.taskName = "SimpleTask";
sTask.taskNamespace = "http://myProcess/com/acme/task";
sTask.inputMessage = bg;
task.inputTask = sTask;

id = service.createAndStartTask(task).outputTask;
```
6. 使用持續 ID (在前一個步驟的範例中是 id) 識別遠端程序及作業。例如，若要求先前建立的人力作業：

```
HTMClient.HTMReference.claimTask claim = new HTMClient.HTMReference.claimTask();
claim.inputTask = id;
```

#### 相關工作

第 429 頁的『產生 Proxy 用戶端 (.NET)』

.NET 用戶端應用程式會使用 *Proxy* 用戶端與 Web 服務 API 互動。Proxy 用戶端會保護用戶端應用程式，使其不像 Web 服務傳訊通訊協定那樣複雜。

第 430 頁的『建立 BPEL 程序的 Helper 類別 (.NET)』

特定的 Web 服務 API 作業需要有用戶端應用程式，才能使用「document/literal 包裝」樣式元素。用戶端應用程式需要有 Helper 類別，協助它們產生必要的封套元素。

## 新增安全 (.NET)

您可以將安全機制整合至用戶端應用程式，以保護 Web 服務通訊的安全。

這些安全機制可以包括使用者名稱記號 (使用者名稱及密碼)，或自訂二進位及 XML 型安全記號。

1. 下載及安裝 Microsoft .NET 的 Web Services Enhancements (WSE) 2.0 SP3。這可從下列位置取得：

<http://www.microsoft.com/downloads/details.aspx?familyid=1ba1f631-c3e7-420a-bc1e-ef18bab66122&displaylang=en>

2. 如下修改產生的 Proxy 用戶端程式碼。

將：

```
public class Export1_MyMicroflowHttpService : System.Web.Services.Protocols.SoapHttpClientProtocol {
```

變更爲：

```
public class Export1_MyMicroflowHttpService : Microsoft.Web.Services2.WebServicesClientProtocol {
```

**註：**如果執行 WSDL.exe 工具來重新產生 Proxy 用戶端，則這些修改會遺失。

3. 修改用戶端應用程式程式碼，方法是在檔案頂端新增下列各行：

```

using System.Web.Services.Protocols;
using Microsoft.Web.Services2;
using Microsoft.Web.Services2.Security.Tokens;
...

```

4. 新增程式碼，以實作想要的安全機制。例如，下列程式碼會新增使用者名稱及密碼保護：

```

string user = "U1";
string pwd = "password";
UsernameToken token =
    new UsernameToken(user, pwd, PasswordOption.SendPlainText);

me._proxy.RequestSoapContext.Security.Tokens.Clear();
me._proxy.RequestSoapContext.Security.Tokens.Add(token);

```

---

## 查詢商業程序及作業相關物件

您可以使用 Web 服務 API 來查詢「商業程序編排器」資料庫中的商業程序及作業相關物件，來擷取這些物件的特定內容。

「商業程序編排器」資料庫會儲存範本 (模型) 及實例 (執行時期) 資料，用來管理商業程序及作業。

透過 Web 服務 API，用戶端應用程式可以發出查詢，以從資料庫擷取商業程序及作業的相關資訊。

用戶端應用程式可以發出一性查詢來擷取物件的特定內容。您可以儲存經常使用的查詢。之後，用戶端應用程式就可以擷取及使用這些儲存查詢。

### 商業程序及作業相關物件的查詢

使用 Web 服務 API 的查詢介面，可取得關於商業程序及作業的相關資訊。

用戶端應用程式會使用與 SQL 相似的語法來查詢資料庫。

#### Java Web 服務的範例

```

string processTemplateName = "ProcessCustomerLR";
query query1 = new query();
query1.selectClause = "DISTINCT PROCESS_INSTANCE.STARTED, PROCESS_INSTANCE.PIID";
query1.whereClause =
    "PROCESS_INSTANCE.TEMPLATE_NAME = '" + processTemplateName + "'";
query1.orderByClause = "PROCESS_INSTANCE.STARTED";
query1.threshold = null;
query1.timeZone = "UTC"; query1.skipTuples = null;
queryResponse queryResponse1 = proxy.query(query1);

```

從資料庫擷取的資訊會作為查詢結果集，透過 Web 服務 API 傳回。

例如：

```

QueryResultSetType queryResultSet = queryResponse1.queryResultSet;
if (queryResultSet != null) {
    Console.WriteLine("--> QueryResultSetType");
    Console.WriteLine(" . size= " + queryResultSet.size);
    Console.WriteLine(" . numberColumns= " + queryResultSet.numberColumns);
    string indent = " . ";

    // -- the query column info
    QueryColumnInfoType[] queryColumnInfo = queryResultSet.QueryColumnInfo;
    if (queryColumnInfo.Length > 0) {

```

```

        Console.WriteLine();
        Console.WriteLine("= . QueryColumnInfoType size= " + queryColumnInfo.Length);
        Console.Write( " | tableName ");
        for (int i = 0; i < queryColumnInfo.Length ; i++) {
            Console.Write( " | " + queryColumnInfo[i].tableName.PadLeft(20) );
        }
        Console.WriteLine();
        Console.Write( " | columnName ");
        for (int i = 0; i < queryColumnInfo.Length ; i++) {
            Console.Write( " | " + queryColumnInfo[i].columnName.PadLeft(20) );
        }
        Console.WriteLine();
        Console.Write( " | data type ");
        for (int i = 0; i < queryColumnInfo.Length ; i++) {
            QueryColumnInfoType tt = queryColumnInfo[i].type;
            Console.WriteLine( " | " + tt.ToString());
        }
        Console.WriteLine();
    }
    else {
        Console.WriteLine("--> queryColumnInfo= <null>");
    }

    // - the query result values
    string[][] result = queryResultSet.result;
    if (result !=null) {
        Console.WriteLine();
        Console.WriteLine("= . result size= " + result.Length);
        for (int i = 0; i &lt; result.Length; i++) {
            Console.Write(indent +i );
            string[] row = result[i];
            for (int j = 0; j &lt; row.Length; j++ ) {
                Console.Write(" | " + row[j]);
            }
            Console.WriteLine();
        }
    }
    else {
        Console.WriteLine("--> result= <null>");
    }
}
else {
    Console.WriteLine("--> QueryResultSetType= <null>");
}
}

```

查詢函數會根據呼叫者的權限傳回物件。查詢結果集只包含授權呼叫者可以查看的那些物件內容。

提供您預先定義的資料庫視圖，以查詢物件內容。若為程序範本，查詢函數的語法如下所示：

```

ProcessTemplateData[] queryProcessTemplates
    (java.lang.String whereClause,
     java.lang.String orderByClause,
     java.lang.Integer threshold,
     java.util.TimeZone timezone);

```

若為作業範本，查詢函數的語法如下所示：

```

TaskTemplate[] queryTaskTemplates
    (java.lang.String whereClause,
     java.lang.String orderByClause,
     java.lang.Integer threshold,
     java.util.TimeZone timezone);

```

若為其他商業程序及作業相關物件，查詢函數的語法如下所示：

```

QueryResultSet query (java.lang.String selectClause,
                      java.lang.String whereClause,
                      java.lang.String orderByClause,
                      java.lang.Integer skipTuples
                      java.lang.Integer threshold,
                      java.util.TimeZone timezone);

```

查詢介面也包含 queryAll 方法。您可以使用此方法來擷取物件的所有相關資料，例如供監視用途。queryAll 方法的呼叫者必須具有下列其中一個 Java 2 Platform Enterprise Edition (J2EE) 角色：BPESystemAdministrator、BPESystemMonitor、TaskSystemAdministrator 或 TaskSystemMonitor。不會套用使用物件的對應工作項目的權限檢查。

## .NET 的範例

```

ProcessTemplateType[] templates = null;

try {
    queryProcessTemplates iW = new queryProcessTemplates();
    iW.whereClause = "PROCESS_TEMPLATE.STATE=PROCESS_TEMPLATE.STATE.STATE_STARTED";
    iW.orderByClause = null;
    iW.threshold = null;
    iW.timeZone = null;

    Console.WriteLine("--> queryProcessTemplates ... ");
    Console.WriteLine("--> query: WHERE " + iW.whereClause + " ORDER BY " +
        iW.orderByClause + " THRESHOLD " + iW.threshold + " TIMEZONE" + iW.timeZone);

    templates = proxy.queryProcessTemplates(iW);

    if (templates.Length < 1) {
        Console.WriteLine("--> No templates found :-(");
    }
    else {
        for (int i = 0; i < templates.Length ; i++) {
            Console.WriteLine("--> found template with ptid: " + templates[i].ptid);
            Console.WriteLine(" and name: " + templates[i].name);
            /* ... other properties of ProcessTemplateType ... */
        }
    }
}
catch (Exception e) {
    Console.WriteLine("exception= " + e);
}

```

## 查詢參數

每個查詢必須指定許多與 SQL 相似的子句及參數。

查詢由下列部分組成：

- Select 子句
- Where 子句
- Order by 子句
- 跳過值組參數
- 臨界值參數
- 時區參數

### 相關概念

第 341 頁的『Select 子句』

查詢函數中的 select 子句會識別查詢要傳回的物件內容。

第 342 頁的『Where 子句』

查詢函數中的 where 子句會說明要套用以查詢網域的過濾準則。

第 343 頁的『Order by 子句』

查詢函數中的 order by 子句會指定查詢結果集的排序準則。

第 344 頁的『跳過值組參數』

跳過值組參數可指定查詢結果集值組數目，而這些值組是自查詢結果集開頭開始予以忽略且不會傳回給查詢結果集中的呼叫者。

第 344 頁的『臨界值參數』

查詢函數中的臨界值參數會限制查詢結果集中，從伺服器傳回至用戶端的物件數。

第 344 頁的『時區參數』

查詢函數中的時區參數會定義查詢中時間戳記常數的時區。

第 345 頁的『查詢結果』

查詢結果集包含查詢的結果。

## 商業程序及人力作業物件查詢的預先定義視圖

針對商業程序及人力作業物件，已提供預先定義的資料庫視圖。

當您查詢這些物件的參照資料時，可使用這些視圖。當您使用這些視圖時，您不需要針對視圖直欄明確新增 join 述詞，將會自動新增這些建構。您可以使用 Web 服務 API 的查詢函數，來查詢此資料。

### 相關參考

第 352 頁的『ACTIVITY 視圖』

使用此預先定義的資料庫視圖，以針對活動進行查詢。

第 353 頁的『ACTIVITY\_ATTRIBUTE 視圖』

使用此預先定義的資料庫視圖，以針對活動的自訂內容進行查詢。

第 353 頁的『ACTIVITY\_SERVICE 視圖』

使用此預先定義的資料庫視圖，以針對活動服務進行查詢。

第 354 頁的『APPLICATION\_COMP 視圖』

使用此預先定義的資料庫視圖，可以查詢應用程式元件 ID 及作業的預設值。

第 354 頁的『ESCALATION 視圖』

使用此預先定義的資料庫視圖，可以查詢呈報資料。

第 356 頁的『ESCALATION\_CPROP 視圖』

使用此預先定義的資料庫視圖，可以查詢呈報的自訂內容。

第 356 頁的『ESCALATION\_DESC 視圖』

使用此預先定義的資料庫視圖，可以查詢呈報的多種語言敘述性資料。

第 358 頁的『PROCESS\_ATTRIBUTE 視圖』

使用此預先定義的資料庫視圖，以針對程序的自訂內容進行查詢。

第 358 頁的『PROCESS\_INSTANCE 視圖』

使用此預先定義的資料庫視圖，以針對程序實例進行查詢。

第 359 頁的『PROCESS\_TEMPLATE 視圖』

使用此預先定義的資料庫視圖，以針對程序範本進行查詢。

第 360 頁的『QUERY\_PROPERTY 視圖』

使用此預先定義的資料庫視圖，以針對程序層次變數進行查詢。



第 360 頁的『TASK 視圖』

使用此預先定義的資料庫視圖，以針對作業物件進行查詢。

第 363 頁的『TASK\_CPROP 視圖』

使用此預先定義的資料庫視圖，可以查詢作業物件的自訂內容。

第 364 頁的『TASK\_DESC 視圖』

使用此預先定義資料庫的視圖，可以查詢作業物件的多種語言敘述性資料。

第 364 頁的『TASK\_TEMPL 視圖』

此預先定義資料庫視圖保留可用來實例化作業的資料。

第 365 頁的『TASK\_TEMPL\_CPROP 視圖』

使用此預先定義的資料庫視圖，可以查詢作業範本的自訂內容。

第 365 頁的『TASK\_TEMPL\_DESC 視圖』

使用此預先定義的資料庫視圖，可以查詢作業範本物件的多種語言敘述性資料。

第 366 頁的『WORK\_ITEM 視圖』

使用此預先定義的資料庫視圖，以針對工作項目以及程序、作業與呈報的權限資料進行查詢。

## 管理儲存查詢

儲存查詢提供方式來儲存經常執行的查詢。儲存查詢可以是供所有使用者使用的查詢 (公用查詢)，或屬於特定使用者的查詢 (專用查詢)。

儲存查詢是儲存在資料庫中，且依名稱識別的查詢。專用及公用儲存查詢的名稱可以相同，而來自不同擁有者的專用儲存查詢的名稱也可以相同。

您可以擁有商業程序物件、作業物件或這兩種物件類型組合的儲存查詢。

管理公用儲存查詢

公用儲存查詢是由系統管理者所建立。所有使用者都可以使用這些查詢。

管理其他使用者的專用儲存查詢

任何使用者都可以建立專用查詢。這些查詢只適用於查詢擁有者及系統管理者。

使用專用儲存查詢

如果您不是系統管理者，則可以建立、執行及刪除您自己的專用儲存查詢。您也可以使用系統管理者建立的公用儲存查詢。

---

## 第 13 章 開發 JMS 用戶端應用程式

您可以透過「Java 訊息服務 (JMS) API」開發存取商業程序應用程式的用戶端應用程式。

---

### JMS 簡介

WebSphere Process Server 6.1 版支援非同步傳訊 (以 Java 訊息服務 (JMS) 程式設計介面為基礎) 作為通訊方法。

JMS 提供共用方法，讓 Java 用戶端 (用戶端應用程式或 J2EE 應用程式) 可以將要求建立、傳送、接收及讀取為 JMS 訊息。

JMS 是非同步傳訊型介面，其：

- 使用**點對點或發佈/訂閱傳訊**。訊息型架構會將資訊推送給其他應用程式，而這些應用程式並不需要明確要求該資訊。相同的資訊可以平行遞送給多位訂閱者。「商業程序編排器」的 JMS 介面只支援點對點傳訊。
- 提供**獨立的節奏**。JMS 架構是以非同步方式運作，但是也可以模擬同步的要求/回應模式。這容許來源及目標系統同步運作，而不需要互相等待。此功能十分適用於「商業程序編排器」，因為其提供與長時間執行商業程序非同步互動的功能。
- 支援**交易**。交易讓用戶端應用程式可以將訊息群組當成單一基本單位進行傳送或接收。JMS 交易會在伺服器的交易內執行。若為「商業程序編排器」的 JMS 介面，則一般會傳送及接收每個交易的單一訊息。
- **保證資訊遞送**。JMS 架構可以利用交易模式來管理訊息，並確保訊息遞送 (但不提供及時遞送的任何保證)。若為「商業程序編排器」，則此可靠訊息遞送功能處理的是商業程序，因此特別重要。
- 確保**異質架構之間的交互作業能力**。來源及目標應用程式可以在異質環境中作業，而不需要處理與其各自架構相關的通訊及執行問題。
- **讓交換更為流暢**。切換至訊息模式，容許交換更精細的資訊。

---

### 商業程序的需求

使用 WebSphere Integration Developer 開發的商業程序，在「商業程序編排器」上執行時，必須符合特定規則才能透過 JMS API 存取。

需求如下：

1. 商業程序介面必須使用「document/literal 包裝」樣式 (在 XML 型 RPC (JAX-RPC 1.1) 規格的 Java API 中定義) 來定義。這是使用 WebSphere Integration Developer 開發的所有商業程序及人力作業的預設樣式。
2. Web 服務作業的商業程序及人力作業所顯示的錯誤訊息，必須包含使用「XML 綱目」元素定義的單一 WSDL 訊息組件。例如：

```
<wsdl:part name="myFault" element="myNamespace:myFaultElement"/>
```

**相關資訊**

 [XML 型 RPC \(JAX-RPC\) 的 Java API 下載頁面](#)

## 存取 JMS 介面

若要透過 JMS 介面來傳送及接收訊息，應用程式必須先建立 `BPC.cellname.Bus` 的連線、建立階段作業，然後產生訊息生產者及消費者。

程序伺服器接受遵循點對點參照範例的「Java 訊息服務 (JMS)」訊息。傳送或接收 JMS 訊息的應用程式必須執行下列動作。

下列範例假設 JMS 用戶端在受管理環境中執行 (EJB、應用程式用戶端或 Web 用戶端儲存器)。如果您想要在 J2SE 環境中執行 JMS 用戶端，請參閱<http://www-1.ibm.com/support/docview.wss?uid=swg24012804>中的「具有 IBM WebSphere Application Server 的 IBM Client for JMS on J2SE」。

1. 建立 `BPC.cellname.Bus` 的連線。沒有存在針對用戶端應用程式要求的預先配置 `ConnectionFactory`：用戶端應用程式可以使用 JMS API 的 `ReplyConnectionFactory` 或建立專屬的 `ConnectionFactory`，在任一情況下，它都可以使用「Java 命名和目錄介面 (JNDI)」查閱，來擷取 `ConnectionFactory`。JNDI 查閱名稱必須與配置「商業程序編排器」的外部要求佇列時所指定的名稱相同。下列範例假設用戶端應用程式建立自己的 `ConnectionFactory`，名稱為 `"jms/clientCF"`。

```
//Obtain the default initial JNDI context.
Context initialContext = new InitialContext();

// Look up the connection factory.
// Create a connection factory that connects to the BPC bus.
// Call it, for example, "jms/clientCF".
// Also configure an appropriate authentication alias.
ConnectionFactory connectionFactory =
    (ConnectionFactory)initialContext.lookup("jms/clientCF");
```

```
// Create the connection.
Connection connection = connectionFactory.createConnection();
```

2. 建立階段作業，以便可以建立訊息生產者及消費者。

```
// Create a transaction session using auto-acknowledgement.
Session session = connection.createSession(true, Session.AUTO_ACKNOWLEDGE);
```
3. 建立訊息生產者來傳送訊息。JNDI 查閱名稱必須與配置「商業程序編排器」的外部要求佇列時所指定的名稱相同。

```
// Look up the destination of the Business Process Choreographer input queue to
// send messages to.
Queue sendQueue = (Queue) initialContext.lookup("jms/BFMJMSAPIQueue");
```

```
// Create a message producer.
MessageProducer producer = session.createProducer(sendQueue);
```

4. 建立訊息消費者來接收回覆。回覆目的地的 JNDI 查閱名稱可以指定使用者定義的目的地，但是它也可以指定預設 (已定義「商業程序編排器」) 回覆目的地 `jms/BFMJMSReplyQueue`。在這兩種情況下，回覆目的地必須位於 `BPC.<cellname>.Bus`。

```
// Look up the destination of the reply queue.
Queue replyQueue = (Queue) initialContext.lookup("jms/BFMJMSReplyQueue");
```

```
// Create a message consumer.
MessageConsumer consumer = session.createConsumer(replyQueue);
```

5. 傳送訊息。

```

// Start the connection.
connection.start();

// Create a message - see the task descriptions for examples - and send it.
// This method is defined elsewhere ...
String payload = createXMLDocumentForRequest();
TextMessage requestMessage = session.createTextMessage(payload);

// Set mandatory JMS header.
// targetFunctionName is the operation name of JMS API
// (for example, getProcessTemplate, sendMessage)
requestMessage.setStringProperty("TargetFunctionName", targetFunctionName);

// Set the reply queue; this is mandatory if the replyQueue
// is not the default queue (as it is in this example).
requestMessage.setJMSReplyTo(replyQueue);

// Send the message.
producer.send(requestMessage);

// Get the message ID.
String jmsMessageID = requestMessage.getJMSMessageID();

session.commit();

```

#### 6. 接收回覆。

```

// Receive the reply message and analyse the reply.
TextMessage replyMessage = (TextMessage) consumer.receive();

// Get the payload.
String payload = replyMessage.getText();

session.commit();

```

#### 7. 關閉連線並釋放資源。

```

// Final housekeeping; free the resources.
session.close();
connection.close();

```

**註：** 不需要在每次交易之後關閉連線。一旦啟動了連線，在關閉連線之前，都可以交換任意數目的要求及回應訊息。範例顯示在單一商業方法內具有單一呼叫的簡式案例。

---

## 商業程序編排器 JMS 訊息的結構

每個 JMS 訊息的標頭及本文都必須要有預先定義的結構。

Java 訊息服務 (JMS) 訊息包含：

- 用於訊息識別及路由資訊的訊息標頭。
- 訊息本文 (內容)，也就是存放內容的地方。

「商業程序編排器」只支援文字訊息格式。

### 訊息標頭

JMS 容許用戶端存取一些訊息標頭欄位。

「商業程序編排器」JMS 用戶端可以設定下列標頭欄位：

- **JMSReplyTo**

傳送要求回覆的目的地。如果要求訊息中未指定此欄位，則會將回覆傳送至「匯出」介面的預設回覆目的地（「匯出」是用戶端介面，呈現商業程序元件）。此目的地可以使用 `initialContext.lookup("jms/BFMJMSReplyQueue");` 取得

- **TargetFunctionName**

WSDL 作業名稱 (例如, "queryProcessTemplates")。一律必須設定此欄位。請注意, TargetFunctionName 指定這裡說明的通用 JMS 訊息介面的作業。這不應該與可間接呼叫的具體程序或作業所提供的作業混淆 (例如, 使用 `call` 或 `sendMessage` 作業)。

「商業程序編排器」用戶端也可以存取下列標頭欄位：

- **JMSMessageID**

可唯一識別訊息。由 JMS 提供者在訊息傳送時設定。如果用戶端在傳送訊息之前設定 JMSMessageID, 則 JMS 提供者會予以改寫。如果進行鑑別時需要訊息 ID, 則用戶端可以在傳送訊息之後擷取 JMSMessageID。

- **JMSCorrelationID**

鏈結訊息。請不要設定此欄位。「商業程序編排器」回覆訊息會包含要求訊息的 JMSMessageID。

每個回應訊息都會包含下列 JMS 標頭欄位：

- **IsBusinessException**

"False" 表示 WSDL 輸出訊息, 或 "true" 表示 WSDL 錯誤訊息。

ServiceRuntimeExceptions 不會傳回給非同步用戶端應用程式。如果在處理 JMS 要求訊息期間發生嚴重異常, 則會導致執行時期失敗, 因而回復處理此要求訊息的交易。然後會重新遞送 JMS 要求訊息。如果很早就失敗, 則在將訊息當成「SCA 匯出」的一部分處理期間 (例如, 解除序列化訊息時), 會嘗試「SCA 匯出」接收目的地指定的最大失敗遞送重試次數。達到最大失敗遞送次數之後, 會將要求訊息新增至「商業程序編排器」匯流排的系統異常目的地。然而, 如果是在「商業流程管理程式」的 SCA 元件實際處理要求期間失敗, 則 WebSphere Process Server 的失敗事件管理基礎架構會處理失敗要求訊息, 亦即, 如果重試無法解決異常狀況, 則最後會記錄在失敗事件管理資料庫。

## 訊息本文

JMS 訊息本文是包含 XML 文件的「字串」, 而此 XML 文件代表作業的 document/literal 封套元素。

有效要求訊息本文的簡式範例如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<_6:queryProcessTemplates xmlns:_6="http://www.ibm.com/xmlns/prod/
  websphere/business-process/services/6.0">
<whereClause>PROCESS_TEMPLATE.STATE IN (1)</whereClause>
</_6:queryProcessTemplates>
```

### 相關工作

第 445 頁的『檢查回應訊息是否有商業異常』

JMS 用戶端應用程式必須檢查所有回應訊息的訊息標頭, 是否有商業異常。

## JMS 呈現的授權

若要授權使用 JMS 介面，則必須在 WebSphere Application Server 中啟用安全設定。

安裝商業程序儲存器時，角色 **JMSAPIUser** 必須對映至使用者 ID。此使用者 ID 是用來發出所有 JMS API 要求。例如，如果 **JMSAPIUser** 對映至 "User A"，則所有 JMS API 要求會顯示為源自 "User A" 的程序引擎。

必須將下列權限指派給 **JMSAPIUser** 角色：

要求	需要的授權
forceTerminate	程序管理者
sendEvent	可能的活動擁有者或程序管理者

註：其他所有要求則不需要特殊授權。

具有商業程序管理者角色的人員則會授與特殊權限。商業程序管理者是特殊角色，與程序實例的程序管理者不同。商業程序管理者擁有所有專用權。

如果程序實例存在，則不可以從使用者登錄中刪除程序起始者的使用者 ID。如果刪除此使用者 ID，則無法繼續此程序的導覽。您會在系統日誌檔中收到下列異常：

非唯一 ID : <user ID>

## JMS API 概觀

JMS 訊息介面 (之後稱為 "JMS API") 可讓您開發用戶端應用程式，以使用非同步方式存取「商業程序編排器」環境中執行的商業程序。

JMS API 容許用戶端應用程式以非同步方式與微流程及長時間執行程序互動。

JMS API 公開的介面與 Web 服務 API 介面相同，但具有下列例外：

- 使用 Web 服務 API 時，call 作業只可以用來呼叫微流程。然而，如果使用 JMS API，則可以使用 call 作業來呼叫微流程及長時間執行程序。
- 下列不是透過 JMS API 公開的作業：
  - callAsync 作業 (與其相關聯回呼作業搭配)。
  - completeAndClaimSuccessor 及 getParticipatingTask 作業

### 範例 - 執行長時間執行程序

若為與長時間執行程序搭配運作的一般用戶端應用程式，則步驟順序如下：

1. 設定 JMS 環境 (如第 440 頁的『存取 JMS 介面』中所述)。
2. 取得已安裝程序定義的清單：
  - 傳送 queryProcessTemplates。
  - 這樣會傳回 **ProcessTemplate** 物件的清單。
3. 取得起始活動 (具有 createInstance="yes" 的接收或挑選) 的清單：
  - 傳送 getStartActivities。
  - 這樣會傳回 **InboundOperationTemplate** 物件的清單。
4. 建立輸入訊息。這是環境特有的，而且可能需要使用預先部署的程序特定成品。



5. 建立程序實例：
  - 發出 `sendMessage`。

使用 JMS API，也可以使用 `call` 作業，以與商業程序提供的長時間執行要求回應作業進行互動。即使經過很長的一段時間，此作業還是會將作業結果或錯誤傳回給指定的回覆目的地。因此，如果使用 `call` 作業，就不需要使用 `query` 及 `getOutputMessage` 作業來取得程序的輸出或錯誤訊息。

6. 重複下列步驟，以選擇性地取得程序實例的輸出訊息：
  - 發出 `query`，以取得程序實例的已完成狀態。
  - 發出 `getOutputMessage`。
7. 選擇性地使用程序所公開的其他作業：
  - `getWaitingActivities` 或 `getActiveEventHandlers`，以取得 **InboundOperationTemplate** 物件的清單。
  - 建立輸入訊息
  - 使用 `sendMessage` 傳送訊息
8. 選擇性地使用 `getCustomProperties` 及 `setCustomProperties`，以取得及設定程序或內含活動上所定義的自訂內容。
9. (選擇性地) 完成使用程序實例：
  - 傳送 `delete` 及 `terminate`，以完成長時間執行程序的使用。

---

## 開發 JMS 應用程式

您必須使用 Java 2 Enterprise Edition (J2EE) 環境，在 Java 中開發 JMS 用戶端應用程式。

JMS 用戶端應用程式會與 JMS API 交換要求及回應訊息。若要建立要求訊息，用戶端應用程式會以代表對應作業的文件/文字封套的 XML 元素填寫 JMS `TextMessage` 訊息本文。

### 複製成品

可以從 WebSphere 環境複製一些成品，以協助建立 JMS 用戶端應用程式。

只有在使用 `BOXMLSerializer` 來建立 JMS 訊息本文時，才需要使用這些成品。

有兩種方式可以取得這些成品：

- 從 WebSphere Process Server 環境發佈及匯出它們。

若為 WebSphere Process Server 6.1，則可以在 `install_root\ProcessChoreographer\client` 目錄中找到所有用戶端成品。若為 JMS API，則這些成品如下：

`BFMIF.wsdl`  
`BFMIF.xsd`  
`BPCGen.xsd`

- 從 WebSphere Process Server 用戶端 CD 複製檔案。



## 從伺服器環境發佈成品

若要協助開發存取 JMS API 的用戶端應用程式，您可以從 WebSphere 伺服器環境發佈一些成品。

若為 WebSphere Process Server 6.1，所有用戶端成品都將位於 `was_home\ProcessChoreographer\client` 目錄中。若為 JMS API，這些成品為：

```
BFMIF.wsdl
BFMIF.xsd
BPCGen.xsd
```

在發佈這些成品之後，請將它們複製至您的用戶端程式設計環境。

## 複製用戶端 CD 中的檔案

您可以在 WebSphere Process Server 用戶端 CD 上取得存取 JMS API 所需的檔案。

1. 存取用戶端 CD，並瀏覽至 `ProcessChoreographer\client` 目錄。
2. 將必要的檔案複製至用戶端應用程式開發環境

若為 WebSphere Process Server 6.1，所有用戶端成品都將位於 `\ProcessChoreographer\client` 目錄中。若為 JMS API，這些成品為：

```
BFMIF.wsdl
BFMIF.xsd
BPCGen.xsd
```

## 檢查回應訊息是否有商業異常

JMS 用戶端應用程式必須檢查所有回應訊息的訊息標頭，是否有商業異常。

JMS 用戶端應用程式必須先檢查回應訊息標頭中的 **IsBusinessException** 內容。

例如：

```
// receive response message
Message receivedMessage = ((JmsProxy) getToBeInvokedUponObject()).receiveMessage();
String strResponse = ((TextMessage) receivedMessage).getText();

if (receivedMessage.getStringProperty("IsBusinessException") {
    // strResponse is a bussiness fault
    // any api can end w/a processFaultMsg
    // the call api also w/a businessFaultMsg
}
else {
    // strResponse is the output message
}
```

### 相關概念

第 441 頁的『商業程序編排器 JMS 訊息的結構』

每個 JMS 訊息的標頭及本文都必須要有預先定義的結構。



---

## 第 14 章 使用 JSF 元件開發商業程序及人力作業的 Web 應用程式

「商業程序編排器」提供數個 JavaServer Faces (JSF) 元件。您可以擴充及整合這些元件，將商業程序及人力作業功能新增至 Web 應用程式。

您可以使用 WebSphere Integration Developer 來建置您的 Web 應用程式。

1. 建立動態專案並變更「Web 專案功能」內容，以包括 JSF 基本元件。

如需建立 Web 專案的相關資訊，請跳至 WebSphere Integration Developer 的資訊中心。

2. 新增必備的「商業程序編排器瀏覽器」Java 保存檔 (JAR 檔)。

新增下列檔案至您專案的 WEB-INF/lib 目錄：

- bpcclientcore.jar
- bfmclientmodel.jar
- htmclientmodel.jar
- bpcjsfcomponents.jar

如果您是在遠端伺服器上部署 Web 應用程式，也會新增下列檔案。在遠端存取「商業程序編排器」API 時，需要這些檔案。

- bpe137650.jar
- task137650.jar

在 WebSphere Process Server 中，這些檔案都位於下列目錄中：

- 在 Windows 系統上：*install\_root*\ProcessChoreographer\client
  - 在 UNIX、Linux 及 i5/OS 系統上：*install\_root*/ProcessChoreographer/client
3. 將您需要的 EJB 參照新增至 Web 應用程式部署描述子 (web.xml 檔)。

```
<ejb-ref id="EjbRef_1">
  <ejb-ref-name>ejb/BusinessProcessHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <home>com.ibm.bpe.api.BusinessFlowManagerHome</home>
  <remote>com.ibm.bpe.api.BusinessFlowManager</remote>
</ejb-ref>
<ejb-ref id="EjbRef_2">
  <ejb-ref-name>ejb/HumanTaskManagerEJB</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <home>com.ibm.task.api.HumanTaskManagerHome</home>
  <remote>com.ibm.task.api.HumanTaskManager</remote>
</ejb-ref>
<ejb-local-ref id="EjbLocalRef_1">
  <ejb-ref-name>ejb/LocalBusinessProcessHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home>com.ibm.bpe.api.LocalBusinessFlowManagerHome</local-home>
  <local>com.ibm.bpe.api.LocalBusinessFlowManager</local>
</ejb-local-ref>
<ejb-local-ref id="EjbLocalRef_2">
  <ejb-ref-name>ejb/LocalHumanTaskManagerEJB</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home>com.ibm.task.api.LocalHumanTaskManagerHome</local-home>
  <local>com.ibm.task.api.LocalHumanTaskManager</local>
</ejb-local-ref>
```

4. 新增「商業程序編排器瀏覽器」JSF 元件至 JSF 應用程式。

- a. 將您應用程式所需的標籤庫參照新增至 JavaServer Pages (JSP) 檔案。通常，您需要 JSF 及 HTML 標籤庫，以及 JSF 元件所需的標籤庫。

- `<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>`
- `<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>`
- `<%@ taglib uri="http://com.ibm.bpe.jsf/taglib" prefix="bpe" %>`

- b. 新增 `<f:view>` 標籤至 JSP 頁面的本文，以及新增 `<h:form>` 標籤至 `<f:view>` 標籤。

- c. 新增 JSF 元件至 JSP 檔。

根據您的應用程式，新增 List 元件、Details 元件、CommandBar 元件或 Message 元件至 JSP 檔。您可以新增每一個元件的多個實例。

- d. 配置 JSF 配置檔中的受管理 Bean。

根據預設值，配置檔是 `faces-config.xml` 檔。這個檔案位於 Web 應用程式的 `WEB-INF` 目錄中。

根據您新增至 JSP 檔的元件，您也需要新增參照至查詢，以及新增其他封套物件至 JSF 配置檔。若要確保正確的錯誤處理，您也需要同時對 JSF 配置檔中的錯誤頁面定義錯誤 Bean 及導覽目標。

```
<faces-config>
...
<managed-bean>
  <managed-bean-name>BPCErr</managed-bean-name>
  <managed-bean-class>com.ibm.bpc.clientcore.util.ErrorBeanImpl
  </managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

...
<navigation-rule>
...
<navigation-case>
  <description>
    The general error page.
  </description>
  <from-outcome>error</from-outcome>
  <to-view-id>/Error.jsp</to-view-id>
</navigation-case>
...
</navigation-rule>
</faces-config>
```

在觸發錯誤頁面的錯誤狀況下，異常會設定在錯誤 Bean 上。

- e. 實作自訂程式碼，支援 JSF 元件時需要。

5. 部署應用程式。

如果您是在網路部署環境中部署應用程式，請將目標資源「Java 命名和目錄介面 (JNDI)」名稱變更為「商業流程管理程式」API 及「人力作業管理程式」API 可在您的 Cell 中找到的值。

- 如果已在相同受管理 Cell 的另一個伺服器上配置您的商業程序儲存器，則名稱具有下列結構：

`cell/nodes/nodename/servers/servername/com/ibm/bpe/api/BusinessManagerHome`  
`cell/nodes/nodename/servers/servername/com/ibm/task/api/HumanTaskManagerHome`

- 如果已在相同 Cell 的叢集上配置您的商業程序儲存器，則名稱具有下列結構：

`cell/clusters/clustername/com/ibm/bpe/api/BusinessFlowManagerHome`  
`cell/clusters/clustername/com/ibm/task/api/HumanTaskManagerHome`

將 EJB 參照對映至 JNDI 名稱，或以手動方式將參照新增至 `ibm-web-bnd.xmi` 檔。

下表列出參照連結及其預設對映。

表 47. 參照連結與 JNDI 名稱的對映

參照連結	JNDI 名稱	備註
<code>ejb/BusinessProcessHome</code>	<code>com/ibm/bpe/api/BusinessFlowManagerHome</code>	遠端階段作業 Bean
<code>ejb/LocalBusinessProcessHome</code>	<code>com/ibm/bpe/api/BusinessFlowManagerHome</code>	本端階段作業 Bean
<code>ejb/HumanTaskManagerEJB</code>	<code>com/ibm/task/api/HumanTaskManagerHome</code>	遠端階段作業 Bean
<code>ejb/LocalHumanTaskManagerEJB</code>	<code>com/ibm/task/api/HumanTaskManagerHome</code>	本端階段作業 Bean

您已部署的 Web 應用程式包含「商業程序編排器瀏覽器」元件提供的功能。

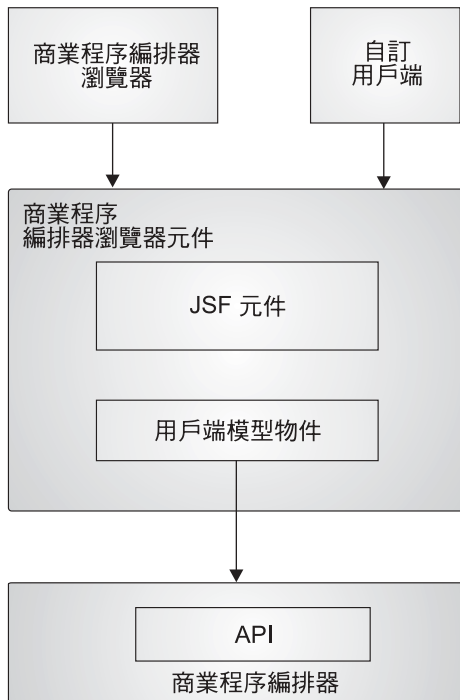
如果您是針對程序及作業訊息使用自訂 JSP，則必須將用來部署 JSP 的 Web 模組對映至自訂 JSF 用戶端所對映的相同伺服器。

---

## 商業程序編排器瀏覽器元件

「商業程序編排器瀏覽器」元件是一組根據 JavaServer Faces (JSF) 技術，可配置且可重複使用的元素。您可以將這些元素內含在 Web 應用程式中。然後，Web 應用程式可以存取已安裝的商業程序及人力作業應用程式。

元件由一組 JSF 元件及一組用戶端模型物件組成。「商業程序編排器」、「商業程序編排器瀏覽器」及其他自訂用戶端的元件關係顯示在下圖中。



## JSF 元件

「商業程序編排器瀏覽器」元件包括以下 JSF 元件。當您建置用於處理商業程序及人力作業的 Web 應用程式時，會將這些 JSF 元件包含在 JavaServer Pages (JSP) 檔案中。

- List 元件

List 元件會在表格中顯示應用程式物件的清單，例如，作業、活動、程序實例、程序範本、工作項目或呈報。這個元件具有相關聯的清單處理程式。

- Details 元件

Details 元件會顯示以下內容：作業、工作項目、活動、程序實例及程序範本。這個元件具有相關聯的明細處理程式。

- CommandBar 元件

CommandBar 元件會顯示包含按鈕的列。這些按鈕代表指令，用來處理明細視圖中的物件或清單中選取的物件。這些物件由清單處理程式或明細處理程式提供。

- Message 元件

Message 元件會顯示可以包含「服務資料物件 (SDO)」或簡式類型的訊息。

## 用戶端模型物件

用戶端模型物件與 JSF 元件一起使用。物件會實作部分基礎「商業程序編排器」API 介面，並覆蓋原始物件。用戶端模型物件會針對標籤提供國家語言支援，並針對部分內容提供轉換器。

---

## JSF 元件中的錯誤處理

JavaServer Faces (JSF) 元件利用預先定義的受管理 Bean `BPCError`，以進行錯誤處理。在觸發錯誤頁面的錯誤狀況下，異常狀況會在錯誤 Bean 上進行設定。

此 Bean 會實作 `com.ibm.bpc.clientcore.util.ErrorBean` 介面。錯誤頁面會在下列狀況中顯示：

- 如果在執行針對清單處理程式定義的查詢期間發生錯誤，且該錯誤由指令的 `execute` 方法作為 `ClientException` 錯誤產生
- 如果 `ClientException` 錯誤由指令的 `execute` 方法產生，且此錯誤既不是 `ErrorsInCommandException` 錯誤，也不會實作 `CommandBarMessage` 介面
- 如果錯誤訊息顯示在元件中，且您遵循訊息的超鏈結

`com.ibm.bpc.clientcore.util.ErrorBeanImpl` 介面的預設實作可用。

介面的定義如下所示：

```
public interface ErrorBean {

    public void setException(Exception ex);

    /*
     * This setter method call allows a locale and
     * the exception to be passed. This allows the
     * getExceptionMessage methods to return localized Strings
     */
    public void setException(Exception ex, Locale locale);

    public Exception getException();
    public String getStack();
    public String getNestedExceptionMessage();
    public String getNestedExceptionStack();
    public String getRootExceptionMessage();
    public String getRootExceptionStack();

    /*
     * This method returns the exception message
     * concatenated recursively with the messages of all
     * the nested exceptions.
     */
    public String getAllExceptionMessages();

    /*
     * This method is returns the exception stack
     * concatenated recursively with the stacks of all
     * the nested exceptions.
     */
    public String getAllExceptionStacks();
}
```

---

## 用戶端模型物件的預設轉換器及標籤

用戶端模型物件會實作「商業程序編排器」API 的對應介面。

`List` 元件及 `Details` 元件可對任何 Bean 操作。您可以顯示 Bean 的所有內容。不過，如果您想要設定用於 Bean 內容的轉換器及標籤，則必須對 `List` 元件使用 `column` 標籤，或對 `Details` 元件使用 `property` 標籤。您可以定義下列 `static` 方法來定義內容的預設轉換器及標籤，而非設定轉換器及標籤。您可以定義下列 `static` 方法：



```

static public String getLabel(String property,Locale locale);
static public com.ibm.bpc.clientcore.converter.SimpleConverter
    getConverter(String property);

```

下表顯示的用戶端模型物件實作對應的「商業流程管理程式」及「人力作業管理程式」API 類別，並提供其內容的預設標籤及轉換器。這個介面封套針對一組內容提供可區分語言環境的標籤及轉換器。下表顯示「商業程序編排器」介面與對應用戶端模型物件的對映。

表 48. 「商業程序編排器」介面如何對映至用戶端模型物件

「商業程序編排器」介面	用戶端模型物件類別
com.ibm.bpe.api.ActivityInstanceData	com.ibm.bpe.clientmodel.bean.ActivityInstanceBean
com.ibm.bpe.api.ActivityServiceTemplateData	com.ibm.bpe.clientmodel.bean.ActivityServiceTemplateBean
com.ibm.bpe.api.ProcessInstanceData	com.ibm.bpe.clientmodel.bean.ProcessInstanceBean
com.ibm.bpe.api.ProcessTemplateData	com.ibm.bpe.clientmodel.bean.ProcessTemplateBean
com.ibm.task.api.Escalation	com.ibm.task.clientmodel.bean.EscalationBean
com.ibm.task.api.Task	com.ibm.task.clientmodel.bean.TaskInstanceBean
com.ibm.task.api.TaskTemplate	com.ibm.task.clientmodel.bean.TaskTemplateBean

## 新增 List 元件至 JSF 應用程式

使用「商業程序編排器瀏覽器」List 元件，可以顯示用戶端模型物件的清單，例如商業程序實例或作業實例。

1. 新增 List 元件至 JavaServer Pages (JSP) 檔。

新增 `bpe:list` 標籤至 `h:form` 標籤。 `bpe:list` 標籤必須包含 `model` 屬性。新增 `bpe:column` 標籤至 `bpe:list` 標籤，以新增將在清單中每一列出現的物件內容。

下列範例顯示如何新增 List 元件，以顯示作業實例。

```

<h:form>

    <bpe:list model="#{TaskPool}">
        <bpe:column name="name" action="taskInstanceDetails" />
        <bpe:column name="state" />
        <bpe:column name="kind" />
        <bpe:column name="owner" />
        <bpe:column name="originator" />
    </bpe:list>

</h:form>

```

`model` 屬性會參照受管理 Bean (TaskPool)。受管理 Bean 提供 Java 物件的清單，清單會重複使用這些物件，然後顯示在個別列。

2. 配置在 `bpe:list` 標籤中參照的受管理 Bean。

若為 List 元件，這個受管理 Bean 必須是 `com.ibm.bpe.jsf.handler.BPCListHandler` 類別的實例。

下列範例顯示如何新增 TaskPool 受管理 Bean 至配置檔。

```

<managed-bean>
<managed-bean-name>TaskPool</managed-bean-name>
<managed-bean-class>com.ibm.bpe.jsf.handler.BPCListHandler</managed-bean-class>

```

```

<managed-bean-scope>session</managed-bean-scope>
  <managed-property>
    <property-name>query</property-name>
    <value>#{TaskPoolQuery}</value>
  </managed-property>
  <managed-property>
    <property-name>type</property-name>
    <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
  </managed-property>
</managed-bean>

<managed-bean>
<managed-bean-name>TaskPoolQuery</managed-bean-name>
<managed-bean-class>sample.TaskPoolQuery</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>
  <managed-property>
    <property-name>type</property-name>
    <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
  </managed-property>
</managed-bean>

<managed-bean>
<managed-bean-name>htmConnection</managed-bean-name>
<managed-bean-class>com.ibm.task.clientmodel.HTMConnection</managed-bean-class>
<managed-bean-scope>application</managed-bean-scope>
  <managed-property>
    <property-name>jndiName</property-name>
    <value>java:comp/env/ejb/LocalHumanTaskManagerEJB</value>
  </managed-property>
</managed-bean>

```

這個範例顯示 TaskPool 具有兩個可配置內容：query 及 type。查詢內容值會參照另一個受管理 Bean (TaskPoolQuery)。類型內容值會指定 Bean 類別，這個類別的內容顯示在所顯示清單的直欄中。相關聯的查詢實例也可以具有一個內容類型。如果指定內容類型，它必須與為清單處理程式所指定的類型相同。

只要查詢的結果可以表示為強式 Bean 的清單，您就可以將任何類型的查詢邏輯新增至 JSF 應用程式。例如，TaskPoolQuery 是使用 com.ibm.task.clientmodel.bean.TaskInstanceBean 物件清單來實作。

### 3. 新增清單處理程式所參照的受管理 Bean 的自訂程式碼。

下列範例顯示如何新增 TaskPool 受管理 Bean 的自訂程式碼。

```

public class TaskPoolQuery implements Query {

    public List execute throws ClientException {

        // Examine the faces-config file for a managed bean "htmConnection".
        //
        FacesContext ctx = FacesContext.getCurrentInstance();
        Application app = ctx.getApplication();
        ValueBinding htmVb = app.createValueBinding("#{htmConnection}");
        htmConnection = (HTMConnection) htmVb.getValue(ctx);
        HumanTaskManagerService taskService =
            htmConnection.getHumanTaskManagerService();

        // Then call the actual query method on the Human Task Manager service.
        //
        QueryResultSet queryResult = taskService.query(
            "DISTINCT TASK.TKIID, TASK.NAME, TASK.KIND, TASK.STATE, TASK.TYPE,"
            + "TASK.STARTED, TASK.ACTIVATED, TASK.DUE, TASK.EXPIRES, TASK.PRIORITY" ,
            "TASK.KIND IN(101,102,105) AND TASK.STATE IN(2)
            AND WORK_ITEM.REASON IN (1)",
            (String)null,

```

```

        (Integer)null,
        (TimeZone)null);
    List applicationObjects = transformToTaskList ( queryResult );
    return applicationObjects ;
}

private List transformToTaskList(QueryResultSet result) {

    ArrayList array = null;
    int entries = result.size();
    array = new ArrayList( entries );

    // Transforms each row in the QueryResultSet to a task instance beans.
    for (int i = 0; i < entries; i++) {
        result.next();
        array.add( new TaskInstanceBean( result, connection ));
    }
    return array ;
}
}

```

TaskPoolQuery Bean 會查詢 Java 物件的內容。這個 Bean 必須實作 com.ibm.bpc.clientcore.Query 介面。當清單處理程式重新整理其內容時，它會呼叫查詢的 execute 方法。這個呼叫會傳回 Java 物件的清單。getType 方法必須傳回所傳回的 Java 物件的類別名稱。

您的 JSF 應用程式現在包含 JavaServer 頁面，其可顯示所要求的物件清單內容，例如您可以使用的作業實例狀態、種類、擁有者及發送者。

## 如何處理清單

List 元件的每個實例都與 com.ibm.bpe.jsf.handler.BPCListHandler 類別的實例相關聯。

此清單處理程式會追蹤在關聯清單中選取的項目，並會提供通知機制，建立清單項目與不同類型項目的明細頁面的關聯。該清單處理程式透過 bpe:list 標籤的 model 屬性，連結至 List 元件。

清單處理程式的通知機制會使用 com.ibm.bpe.jsf.handler.ItemListener 介面進行實作。您可以在 JavaServer Faces (JSF) 應用程式的配置檔中，登錄此介面的實作。

按一下清單中的鏈結，就會觸發通知。所有已設定 action 屬性的直欄，都會呈現鏈結。action 屬性值是 JSF 導覽目標，或傳回 JSF 導覽目標的 JSF 動作方法。

BPCListHandler 類別還會提供 refreshList 方法。您可以在 JSF 方法連結中使用此方法，實作使用者介面控制項，以再次執行查詢。

## 查詢實作

您可以使用清單處理程式，顯示所有類型的物件及其內容。顯示的清單內容視為清單處理程式配置的 com.ibm.bpc.clientcore.Query 介面實作所傳回的物件清單而定。您可以依程式的方式使用 BPCListHandler 類別的 setQuery 方法來設定查詢，或者可以在應用程式的 JSF 配置檔中配置它。

您不僅可以針對「商業程序編排器」API 執行查詢，還可以針對任何其他可從應用程式存取的資訊來源執行查詢，例如，內容管理系統或資料庫。唯一的需求是 execute 方法將查詢結果作為物件的 java.util.List 傳回。

所傳回類型的物件必須保證適當的 `getter` 方法可用於所有內容，這些內容顯示在為其定義查詢的清單直欄中。若要保證所傳回類型的物件符合清單定義，您可以將 `faces` 配置檔中定義的 `BPCListHandler` 實例上類型內容的值，設為所傳回物件的完整類別名稱。您可以在查詢實作的 `getType` 呼叫中傳回此名稱。在執行時期，清單處理程式會檢查物件類型是否符合定義。

若要將錯誤訊息對映至清單中的特定項目，則查詢傳回的物件必須實作具有簽章 `public Object getID()` 的方法。

## 預設轉換器及標籤

查詢傳回的項目必須是 `Bean`，而且它們的類別必須符合指定為 `BPCListHandler` 類別或 `com.ibm.bpc.clientcore.Query` 介面定義中類型的類別。此外，`List` 元件會檢查項目類別或超類別是否實作下列方法：

```
static public String getLabel(String property,Locale locale);
static public com.ibm.bpc.clientcore.converter.SimpleConverter
    getConverter(String property);
```

如果 `Bean` 定義了這些方法，則 `List` 元件會使用此標籤作為清單的預設標籤，而使用 `SimpleConverter` 作為該內容的預設轉換器。您可以使用 `bpe:list` 標籤的 `label` 及 `converterID` 屬性，來改寫這些設定。如需相關資訊，請參閱 `SimpleConverter` 介面及 `ColumnTag` 類別的 `Javadoc`。

## 使用者指定的時區資訊

`JavaServer Faces (JSF)` 元件提供一種公用程式，用以處理 `List` 元件中使用者指定的時區資訊。

`BPCListHandler` 類別會使用 `com.ibm.bpc.clientcore.util.User` 介面，來取得每個使用者的時區與語言環境的相關資訊。`List` 元件預期配置介面實作，其中 `user` 作為 `JavaServer Faces (JSF)` 配置檔的受管理 `Bean` 名稱。如果配置檔中遺漏此項目，則會傳回執行 `WebSphere Process Server` 的時區。

`com.ibm.bpc.clientcore.util.User` 介面的定義如下所示：

```
public interface User {

    /**
     * The locale used by the client of the user.
     * @return Locale.
     */
    public Locale getLocale();
    /**
     * The time zone used by the client of the user.
     * @return TimeZone.
     */
    public TimeZone getTimeZone();

    /**
     * The name of the user.
     * @return name of the user.
     */
    public String getName();
}
```

## List 元件中的錯誤處理

使用 List 元件來顯示 JSF 應用程式中的清單時，可以利用 `com.ibm.bpe.jsf.handler.BPCListHandler` 類別提供的錯誤處理功能。

### 執行查詢或執行指令時發生的錯誤

如果在執行查詢期間發生錯誤，則 `BPCListHandler` 類別會區別存取權不足與其他異常狀況所導致的錯誤。若要捕捉由於存取權不足而導致的錯誤，則由查詢的 `execute` 方法擲出的 `ClientException` 的 `rootCause` 參數必須是 `com.ibm.bpe.api.EngineNotAuthorizedException` 或 `com.ibm.task.api.NotAuthorizedException` 異常狀況。List 元件會顯示錯誤訊息，而非查詢結果。

如果導致錯誤的原因不是存取權不足，則 `BPCListHandler` 類別會將異常狀況物件傳遞至 `com.ibm.bpc.clientcore.util.ErrorBean` 介面的實作，該介面由 JSF 應用程式配置檔的 `BPCError` 索引鍵所定義。當設定異常狀況時，會呼叫錯誤導覽目標。

### 使用清單中顯示的項目時發生的錯誤

`BPCListHandler` 類別會實作 `com.ibm.bpe.jsf.handler.ErrorHandler` 介面。您可以提供這些錯誤的相關資訊，包含 `setErrors` 方法中類型為 `java.util.Map` 的對映參數。此對映中包含 ID 作為索引鍵，異常狀況作為值。ID 必須是導致錯誤之物件的 `getID` 方法所傳回的值。如果已設定對映，且任一 ID 符合清單中顯示的任何項目，則清單處理程式會自動新增包含錯誤訊息的直欄至清單。

若要避免清單中的錯誤訊息過期，則重設錯誤對映。在下列狀況中，對映會自動重設：

- 呼叫 `refreshList` 方法 `BPCListHandler` 類別。
- 在 `BPCListHandler` 類別上設定新查詢。
- `CommandBar` 元件用來觸發清單項目的動作。`CommandBar` 元件會使用此機制，當作數種處理錯誤方法中的一種。

## List 元件：標籤定義

「商業程序編排器瀏覽器」List 元件顯示表格中物件的清單，例如，作業、活動、程序實例、程序範本、工作項目及呈報。

List 元件由 JSF 元件標籤組成：`bpe:list` 及 `bpe:column`。`bpe:column` 標籤是 `bpe:list` 標籤的子元素。

### 元件類別

`com.ibm.bpe.jsf.component.ListComponent`

### 範例語法

```
<bpe:list model="#{ProcessTemplateList}">
  rows="20"
  styleClass="list"
  headerStyleClass="listHeader"
  rowClasses="normal">

  <bpe:column name="name" action="processTemplateDetails"/>
  <bpe:column name="validFromTime"/>
  <bpe:column name="executionMode" label="Execution mode"/>
```

```

    <bpe:column name="state" converterID="my.state.converter"/>
    <bpe:column name="autoDelete"/>
    <bpe:column name="description"/>
</bpe:list>

```

## 標籤屬性

`bpe:list` 標籤的本文只可以包含 `bpe:column` 標籤。呈現表格時，List 元件會疊代應用程式物件的清單，並呈現每一個物件的所有直欄。

表 49. `bpe:list` 屬性

屬性	必要的	說明
<code>buttonStyleClass</code>	否	用來呈現標底區域按鈕的階式樣式表 (CSS) 樣式類別。
<code>cellStyleClass</code>	否	用來呈現個別表格資料格的 CSS 樣式類別。
<code>checkbox</code>	否	判定是否呈現用來選取多個項目的勾選框。屬性值為 <code>true</code> 或 <code>false</code> 。若此值設為 <code>true</code> ，會呈現勾選框直欄。
<code>headerStyleClass</code>	否	用來呈現表格標題的 CSS 樣式類別。
<code>model</code>	是	連結類別為 <code>com.ibm.bpe.jsf.handler.BPCListHandler</code> 的受管理 Bean 的值。
<code>rows</code>	否	頁面上顯示的列數。如果項目數超出列數，則分頁按鈕會顯示在表格的尾端。此屬性不支援值表示式。
<code>rowClasses</code>	否	用來呈現表格中的列的 CSS 樣式類別。
<code>selectAll</code>	否	如果此屬性設為 <code>true</code> ，則依預設會選取清單中的所有項目。
<code>styleClass</code>	否	用來呈現整個表格 (包含標題、列及分頁按鈕) 的 CSS 樣式類別。

表 50. `bpe:column` 屬性

屬性	必要的	說明
<code>action</code>	否	若指定這個屬性，則會在直欄中呈現鏈結。按一下此鏈結時，會觸發 JavaServer Faces 動作方法或 Faces 導覽目標。JavaServer Faces 動作方法具有下列簽章： <code>String method()</code> 。
<code>converterID</code>	否	用來轉換內容值的 Faces 轉換器 ID。如果未設定此屬性，則會使用此模型提供給此內容的任何 Faces 轉換器 ID。
<code>label</code>	否	用作直欄標頭或表格標頭列資料格的標籤的文字或數值連結表示式。如果未設定此屬性，則會使用此模型提供給此內容的任何標籤。
<code>name</code>	是	此直欄中顯示的內容名稱。

## 新增 Details 元件至 JSF 應用程式

使用「商業程序編排器瀏覽器」Details 元件，可以顯示作業、工作項目、活動、程序實例及程序範本的內容。

1. 新增 Details 元件至 JavaServer Pages (JSP) 檔。

新增 `bpe:details` 標籤至 `<h:form>` 標籤。`bpe:details` 標籤必須包含 **model** 屬性。您可以利用 `bpe:property` 標籤，新增內容至 Details 元件。

下列範例顯示如何新增 Details 元件，以顯示作業實例的部分內容。

```
<h:form>

    <bpe:details model="#{TaskInstanceDetails}">
        <bpe:property name="displayName" />
        <bpe:property name="owner" />
        <bpe:property name="kind" />
        <bpe:property name="state" />
        <bpe:property name="escalated" />
        <bpe:property name="suspended" />
        <bpe:property name="originator" />
        <bpe:property name="activationTime" />
        <bpe:property name="expirationTime" />
    </bpe:details>

</h:form>
```

**model** 屬性會參照受管理 Bean `TaskInstanceDetails`。這個 Bean 提供 Java 物件的內容。

2. 配置在 `bpe:details` 標籤中參照的受管理 Bean。

若為 Details 元件，這個受管理 Bean 必須是 `com.ibm.bpe.jsf.handler.BPCDetailsHandler` 類別的實例。這個處理程式類別會包裝 Java 物件，並將其公開內容顯示於明細元件。

下列範例顯示如何新增 `TaskInstanceDetails` 受管理 Bean 至配置檔。

```
<managed-bean>
    <managed-bean-name>TaskInstanceDetails</managed-bean-name>
    <managed-bean-class>com.ibm.bpe.jsf.handler.BPCDetailsHandler</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
    <managed-property>
        <property-name>type</property-name>
        <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
    </managed-property>
</managed-bean>
```

這個範例顯示 `TaskInstanceDetails` Bean 具有可配置的 `type` 內容。類型內容值指定 Bean 類別 (`com.ibm.task.clientmodel.bean.TaskInstanceBean`)，這個類別的內容顯示在所顯示明細的列中。Bean 類別可以是任何 JavaBeans 類別。如果 Bean 提供預設轉換器及內容標籤，則會以與 List 元件相同的方式，來使用轉換器及標籤進行呈現。

您的 JSF 應用程式現在包含可以顯示指定物件明細的 JavaServer 頁面，例如作業實例的明細。

### Details 元件：標籤定義

「商業程序編排器瀏覽器」Details 元件顯示以下各項的內容：作業、工作項目、活動、程序實例及程序範本。



Details 元件由 JSF 元件標籤組成：bpe:details 及 bpe:property。bpe:property 標籤是 bpe:details 標籤的子元素。

## 元件類別

com.ibm.bpe.jsf.component.DetailsComponent

## 範例語法

```
<bpe:details model="#{MyActivityDetails}">
  <bpe:property name="name"/>
  <bpe:property name="owner"/>
  <bpe:property name="activated"/>
</bpe:details>

<bpe:details model="#{MyActivityDetails}" style="style" styleClass="cssStyle">
  style="style"
  styleClass="cssStyle"
</bpe:details>
```

## 標籤屬性

使用 bpe:property 標籤，以同時指定顯示的屬性子集與這些屬性的顯示次序。如果明細標籤不包含任何屬性標籤，則會呈現所有模型物件的可用屬性。

表 51. bpe:details 屬性

屬性	必要的	說明
columnClasses	否	用來呈現直欄的階式樣式表 (CSS) 樣式類別的清單，以逗點區隔。
id	否	元件的 JavaServer Faces ID。
model	是	連結類別為 com.ibm.bpe.jsf.handler.BPCDetailsHandler 的受管理 Bean 的值。
rowClasses	否	用來呈現列的 CSS 樣式類別清單，以逗點區隔。
styleClass	否	用來呈現 HTML 元素的 CSS 類別。

表 52. bpe:property 屬性

屬性	必要的	說明
converterID	否	用來在 JavaServer Faces (JSF) 配置檔中登錄轉換器的 ID。
label	否	內容的標籤。如果未設定此屬性，則用戶端模型類別會提供預設標籤。
name	是	要顯示的內容名稱。此名稱必須對應於某個指定內容，如對應的用戶端模型類別中所定義。

## 新增 CommandBar 元件至 JSF 應用程式

使用「商業程序編排器瀏覽器」CommandBar 元件，可以顯示具有按鈕的橫列。這些按鈕代表的指令可在物件或清單中所選物件的明細視圖上操作。

當使用者按一下使用者介面中的按鈕時，對應指令便會在已選取的物件上執行。您可以在 JSF 應用程式中新增及擴充 CommandBar 元件。

### 1. 新增 CommandBar 元件至 JavaServer Pages (JSP) 檔。

新增 bpe:commandbar 標籤至 <h:form> 標籤。bpe:commandbar 標籤必須包含 model 屬性。

下列範例顯示如何新增 CommandBar 元件，來提供作業實例清單的「重新整理」及「要求」指令。

```
<h:form>

    <bpe:commandbar model="#{TaskInstanceList}">
        <bpe:command commandID="Refresh" >
            action="#{TaskInstanceList.refreshList}"
            label="Refresh"/>

        <bpe:command commandID="MyClaimCommand" >
            label="Claim" >
            commandClass="<customcode>"/>
        </bpe:commandbar>
    </h:form>
```

**model** 屬性參照受管理 Bean。這個 Bean 必須實作 ItemProvider 介面，並提供已選取的 Java 物件。CommandBar 元件通常會與相同 JSP 檔中的 List 元件或 Details 元件搭配使用。在標籤中指定的模型一般都會與相同頁面上的 List 元件或 Details 元件中指定的模型相同。因此，例如對於 List 元件，指令會處理清單中選取的項目。

在這個範例中，**model** 屬性會參照 TaskInstanceList 受管理 Bean。這個 Bean 提供在作業實例清單中選取的物件。Bean 必須實作 ItemProvider 介面。這個介面是由 BPCListHandler 類別及 BPCDetailsHandler 類別實作。

### 2. 選擇性的：配置在 bpe:commandbar 標籤中參照的受管理 Bean。

比方說，如果 CommandBar **model** 屬性參照受管理 Bean (已針對清單或明細處理程式配置)，則不需要任何進一步的配置。如果您變更這些處理程式的配置，或者您使用不同的受管理 Bean，請將實作 ItemProvider 介面的受管理 Bean 新增至 JSF 配置檔。

### 3. 將程式碼 (實作自訂指令) 新增至 JSF 應用程式。

下列程式碼 Snippet 顯示如何撰寫指令類別，來實作 Command 介面。JSP 檔中的 bpe:command 標籤會參照這個指令類別 (MyClaimCommand)。

```
public class MyClaimCommand implements Command {

    public String execute(List selectedObjects) throws ClientException {
        if( selectedObjects != null && selectedObjects.size() > 0 ) {
            try {
                // Determine HumanTaskManagerService from an HTMLConnection bean.
                // Configure the bean in the faces-config.xml for easy access
                // in the JSF application.
                FacesContext ctx = FacesContext.getCurrentInstance();
                ValueBinding vb =
                    ctx.getApplication().createValueBinding("#{htmlConnection}");
                HTMLConnection htmConnection = (HTMLConnection) htmVB.getValue(ctx);
                HumanTaskManagerService htm =
                    htmConnection.getHumanTaskManagerService();

                Iterator iter = selectedObjects.iterator() ;
```

```

while( iter.hasNext() ) {
    try {
        TaskInstanceBean task = (TaskInstanceBean) iter.next() ;
        TKIID tiid = task.getID() ;

        htm.claim( tiid ) ;
        task.setState( new Integer(TaskInstanceBean.STATE_CLAIMED ) ) ;

    }
    catch( Exception e ) {
        ; // Error while iterating or claiming task instance.
        // Ignore for better understanding of the sample.
    }
}
}
catch( Exception e ) {
    ; // Configuration or communication error.
    // Ignore for better understanding of the sample
}
}
return null;
}

// Default implementations
public boolean isMultiSelectEnabled() { return false; }
public boolean[] isApplicable(List itemsOnList) {return null; }
public void setContext(Object targetModel) {; // Not used here }
}

```

這個指令是以下列方式來處理：

- a. 當使用者按一下指令列中的對應按鈕時，即會呼叫指令。 **CommandBar** 元件會從 **model** 屬性中指定的項目提供者擷取已選取的項目，並將已選取的物件清單傳遞至 **commandClass** 實例的 **execute** 方法。
- b. **commandClass** 屬性會參照實作 **Command** 介面的自訂指令實作。這表示指令必須實作 **public String execute(List selectedObjects) throws ClientException** 方法。指令會傳回一個結果，用來判定 **JSF** 應用程式的下一個導覽規則。
- c. 在指令完成之後，**CommandBar** 元件會評估 **action** 屬性。**action** 屬性可以是靜態字串，或是利用方法連結至具有 **public String Method()** 簽章的 **JSF** 動作方法。使用 **action** 屬性，置換指令類別的結果，或明確地指定導覽規則的結果。如果該指令產生了 **ErrorsInCommandException** 異常狀況之外的異常狀況，則不會處理 **action** 屬性。
- d. 如果 **commandClass** 屬性沒有指定指令類別，會立即呼叫動作。例如，對於範例中的重新整理指令，會呼叫 **JSF** 值表示式 **#{TaskInstanceList.refreshList}**，而非指令。

您的 **JSF** 應用程式現在包含 **JavaServer** 頁面，可以實作自訂指令列。

## 如何處理指令

使用 **CommandBar** 元件，可以將動作按鈕新增至您的應用程式。該元件會在使用者介面中建立用於動作的按鈕，並處理按一下按鈕時建立的事件。

這些按鈕會觸發處理 **com.ibm.bpe.jsf.handler.ItemProvider** 介面所傳回物件的功能，例如 **BPCListHandler** 類別，或者 **BPCDetailsHandler** 類別。**CommandBar** 元件使用項目提供者，由 **bpe:commandbar** 標籤中 **model** 屬性的值定義。

當按一下應用程式使用者介面指令列區段中的某個按鈕時，CommandBar 元件會使用以下方法處理關聯事件。

1. CommandBar 元件會識別為產生事件的按鈕指定的 `com.ibm.bpc.clientcore.Command` 介面實作。
2. 如果與 CommandBar 元件相關聯的模型實作 `com.ibm.bpe.jsf.handler.ErrorHandler` 介面，則會呼叫 `clearErrorMap` 方法，以移除上一個事件的錯誤訊息。
3. 會呼叫 `ItemProvider` 介面的 `getSelectedItems` 方法。傳回的項目清單會傳遞至指令的 `execute` 方法，並會呼叫指令。
4. CommandBar 元件會決定 JavaServer Faces (JSF) 導覽目標。如果未在 `bpe:commandbar` 標籤中指定 **action** 屬性，則 `execute` 方法的回覆值會指定導覽目標。如果 **action** 屬性設為 JSF 方法連結，則方法傳回的字串會解譯為導覽目標。**action** 屬性也可以指定明確的導覽目標。

## CommandBar 元件：標籤定義

「商業程序編排器瀏覽器」CommandBar 元件顯示包含按鈕的列。這些按鈕用來處理明細視圖中的物件或清單中選取的物件。

CommandBar 元件由 JSF 元件標籤組成：`bpe:commandbar` 及 `bpe:command`。`bpe:command` 標籤是 `bpe:commandbar` 標籤的子元素。

### 元件類別

`com.ibm.bpe.jsf.component.CommandBarComponent`

### 範例語法

```
<bpe:commandbar model="#{TaskInstanceList}">
    <bpe:command
        commandID="Work on"
        label="Work on..."
        commandClass="com.ibm.bpc.explorer.command.WorkOnTaskCommand"
        context="#{TaskInstanceDetailsBean}" />
    <bpe:command
        commandID="Cancel"
        label="Cancel"
        commandClass="com.ibm.task.clientmodel.command.CancelClaimTaskCommand"
        context="#{TaskInstanceList}" />
</bpe:commandbar>
```

### 標籤屬性

表 53. `bpe:commandbar` 屬性

屬性	必要的	說明
<code>buttonStyleClass</code>	否	用來呈現指令列按鈕的階式樣式表 (CSS) 樣式類別。
<code>id</code>	否	元件的 JavaServer Faces ID。

表 53. `bpe:commandbar` 屬性 (繼續)

屬性	必要的	說明
<code>model</code>	是	連結表示式與實作 <code>ItemProvider</code> 介面的受管理 <code>Bean</code> 的值。此受管理 <code>Bean</code> 通常是 <code>com.ibm.bpe.jsf.handler.BPCListHandler</code> 類別或 <code>com.ibm.bpe.jsf.handler.BPCDetailsHandler</code> 類別，由 <code>List</code> 元件或 <code>Details</code> 元件在同一 <code>JavaServer Pages (JSP)</code> 檔案中作為 <code>CommandBar</code> 元件使用。
<code>styleClass</code>	否	用來呈現指令列的 <code>CSS</code> 樣式類別。

表 54. `bpe:command` 屬性

屬性	必要的	說明
<code>action</code>	否	指令按鈕會觸發的 <code>JavaServer Faces</code> 動作方法或 <code>Faces</code> 導覽目標。此動作傳回的導覽目標會改寫所有其他導覽規則。當未擲出異常或由指令產生 <code>ErrorsInCommandException</code> 異常時，會呼叫此動作。
<code>commandClass</code>	否	指令類別的名稱。 <code>CommandBar</code> 元件會建立此類別的實例，若選取此指令按鈕，則會執行該實例。
<code>commandID</code>	是	指令的 <code>ID</code> 。
<code>context</code>	否	為使用 <code>commandClass</code> 屬性指定的指令提供環境定義的物件。第一次存取指令列時會擷取環境定義物件。
<code>immediate</code>	否	指定何時觸發指令。如果此屬性值是 <code>true</code> ，則會在處理頁面的輸入之前觸發此指令。預設值是 <code>false</code> 。
<code>label</code>	是	指令列中呈現的按鈕標籤。
<code>rendered</code>	否	決定是否呈現某個按鈕。屬性值可以是布林值或值表示式。
<code>styleClass</code>	否	用來呈現按鈕的 <code>CSS</code> 樣式類別。此樣式會置換為了指令列所定義的按鈕樣式。

## 新增 `Message` 元件至 `JSF` 應用程式

使用「商業程序編排器瀏覽器」`Message` 元件，可以呈現 `JavaServer Faces (JSF)` 應用程式中的資料物件及初始類型。

如果訊息類型是初始類型，會呈現標籤及輸入欄位。如果訊息類型是資料物件，則元件會遍訪物件，並呈現物件內的元素。

1. 新增 `Message` 元件至 `JavaServer Pages (JSP)` 檔。

新增 `bpe:form` 標籤至 `<h:form>` 標籤。`bpe:form` 標籤必須包括 `model` 屬性。

下列範例顯示如何新增 `Message` 元件。

```

<h:form>

    <h:outputText value="Input Message" />
    <bpe:form model="#{MyHandler.inputMessage}" readOnly="true" />

    <h:outputText value="Output Message" />
    <bpe:form model="#{MyHandler.outputMessage}" />

</h:form>

```

Message 元件的 **model** 屬性會參照 com.ibm.bpc.clientcore.MessageWrapper 物件。這個封套物件會包裝「服務資料物件 (SDO)」物件或 Java 初始類型，例如 int 或 boolean。在範例中，訊息是由 MyHandler 受管理 Bean 的內容提供。

## 2. 配置在 bpe:form 標籤中參照的受管理 Bean。

下列範例顯示如何新增 MyHandler 受管理 Bean 至配置檔。

```

<managed-bean>
<managed-bean-name>MyHandler</managed-bean-name>
<managed-bean-class>com.ibm.bpe.sample.jsf.MyHandler</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>

    <managed-property>
        <property-name>type</property-name>
        <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
    </managed-property>

</managed-bean>

```

## 3. 新增自訂程式碼至 JSF 應用程式。

下列範例顯示如何實作輸入及輸出訊息。

```

public class MyHandler implements ItemListener {

    private TaskInstanceBean taskBean;
    private MessageWrapper inputMessage, outputMessage

    /* Listener method, e.g. when a task instance was selected in a list handler.
     * Ensure that the handler is registered in the faces-config.xml or manually.
     */
    public void itemChanged(Object item) {
        if( item instanceof TaskInstanceBean ) {
            taskBean = (TaskInstanceBean) item ;
        }
    }

    /* Get the input message wrapper
     */
    public MessageWrapper getInputMessage() {
        try{
            inputMessage = taskBean.getInputMessageWrapper() ;
        }
        catch( Exception e ) {
            ; //...ignore errors for simplicity
        }
        return inputMessage;
    }

    /* Get the output message wrapper
     */
    public MessageWrapper getOutputMessage() {
        // Retrieve the message from the bean. If there is no message, create
        // one if the task has been claimed by the user. Ensure that only
        // potential owners or owners can manipulate the output message.
        try{

```

```

        outputMessage = taskBean.getOutputMessageWrapper();
        if( outputMessage == null
            && taskBean.getState() == TaskInstanceBean.STATE_CLAIMED ) {
            HumanTaskManagerService htm = getHumanTaskManagerService();
            outputMessage = new MessageWrapperImpl();
            outputMessage.setMessage(
                htm.createOutputMessage( taskBean.getID() ).getObject()
            );
        }
    }
    catch( Exception e ) {
        ; //...ignore errors for simplicity
    }
    return outputMessage
}
}
}

```

MyHandler 受管理 Bean 會實作 `com.ibm.jsf.handler.ItemListener` 介面，以便它可以登錄自己作為清單處理程式的項目接聽器。當使用者按一下清單中的項目時，MyHandler Bean 會在其 `itemChanged( Object item )` 方法中得到所選取項目的通知。處理程式會檢查項目類型，然後儲存關聯 `TaskInstanceBean` 物件的參照。若要使用這個介面，請在 `faces-config.xml` 檔中，將項目新增至適當清單處理程式中的 `itemListener` 清單。

MyHandler Bean 提供 `getInputMessage` 及 `getOutputMessage` 方法。這兩種方法皆會傳回 `MessageWrapper` 物件。這些方法會將呼叫委派給參照的作業實例 Bean。如果作業實例 Bean 傳回空值，例如，因為未設定訊息，處理程式會建立並儲存新的空白訊息。Message 元件會顯示 MyHandler Bean 提供的訊息。

您的 JSF 應用程式現在包含 `JavaServer` 頁面，可以呈現資料物件及初始類型。

## Message 元件：標籤定義

「商業程序編排器瀏覽器」Message 元件會在 `JavaServer Faces (JSF)` 應用程式中呈現 `commonj.sdo.DataObject` 物件及初始類型，例如，整數及字串。

Message 元件由 JSF 元件標籤組成：`bpe:form`。

### 元件類別

`com.ibm.bpe.jsf.component.MessageComponent`

### 範例語法

```

<bpe:form model="#{TaskInstanceDetailsBean.inputMessageWrapper}"
    simplification="true" readOnly="true"
    styleClass4table="messageData"
    styleClass4output="messageDataOutput">
</bpe:form>

```

### 標籤屬性

表 55. `bpe:form` 屬性

屬性	必要的	說明
id	否	元件的 <code>JavaServer Faces ID</code> 。



表 55. `bpe:form` 屬性 (繼續)

屬性	必要的	說明
<code>model</code>	是	連結表示式的值，指 <code>commonj.sdo.DataObject</code> 物件或 <code>com.ibm.bpc.clientcore.MessageWrapper</code> 物件。
<code>readOnly</code>	否	如果此屬性設為 <code>true</code> ，則會呈現唯讀格式。根據預設值，此屬性會設為 <code>false</code> 。
<code>simplification</code>	否	如果此屬性設為 <code>true</code> ，則會顯示包含簡式類型且基數為 0 或 1 的內容。根據預設值，此屬性會設為 <code>true</code> 。
<code>style4validinput</code>	否	用來呈現有效輸入的階式樣式表 (CSS) 樣式。
<code>style4invalidinput</code>	否	用來呈現無效輸入的 CSS 樣式。
<code>styleClass4invalidInput</code>	否	用來呈現無效輸入的 CSS 樣式類別名稱。
<code>styleClass4output</code>	否	用來呈現輸出元素的 CSS 樣式類別名稱。
<code>styleClass4table</code>	否	用來呈現訊息元件所呈現的表格的 CSS 表格樣式類別名稱。
<code>styleClass4validInput</code>	否	用來呈現有效輸入的 CSS 樣式類別名稱。

---

## 第 15 章 開發作業及程序訊息的 JSP 頁面

「商業程序編排器瀏覽器」介面會提供預設輸入及輸出表單，來顯示及輸入商業資料。您可以使用 JSP 頁面，提供自訂的輸入及輸出表單。

若要在 Web 用戶端中包括使用者定義的 JavaServer Pages (JSP) 頁面，您必須在 WebSphere Integration Developer 中建立人力作業模型時指定它們。例如，您可以提供特定作業及其輸入及輸出訊息的 JSP 頁面，以及提供特定使用者角色或所有使用者角色的 JSP 頁面。在執行時期，使用者定義的 JSP 頁面會併入使用者介面中，以顯示輸出資料並收集輸入資料。

自訂的表單不是自行包含的網頁；它們是「商業程序編排器瀏覽器」內嵌在 HTML 表單的 HTML 片段，例如，訊息的所有標籤及輸入欄位的片段。

在包含自訂表單的頁面上按一下按鈕時，即會在「商業程序編排器瀏覽器」中提交並驗證輸入。驗證是根據所提供的內容類型，以及瀏覽器中使用的語言環境。如果無法驗證輸入，就會重新顯示相同頁面，並會在 `messageValidationErrors` 要求屬性中提供驗證錯誤的相關資訊。資訊會提供為對映，以將無效內容的「XML 路徑表示式 (XPath)」對映至發生的驗證異常。

若要新增自訂的表單至「商業程序編排器瀏覽器」，請使用 WebSphere Integration Developer 來完成下列步驟。

1. 建立自訂的表單。

使用者定義的 JSP 頁面適用於 Web 介面中使用的輸入及輸出表單，這個頁面需要存取訊息資料。使用 JSP 中的 Java Snippet 或 JSP 執行語言來存取訊息資料。您可以透過要求環境定義來使用表單中的資料。

2. 指派 JSP 頁面給作業。

使用人力作業編輯器來開啓人力作業。在用戶端設定值中，指定使用者定義的 JSP 頁面的位置，以及自訂的表單套用至的角色，例如管理者。「商業程序編排器瀏覽器」的用戶端設定值儲存在作業範本中。在執行時期，會利用作業範本來擷取這些設定。

3. 將使用者定義的 JSP 頁面封裝在 Web 保存檔 (WAR 檔案) 中。

您可以將 WAR 檔案併入企業保存檔 (其模組包含作業) 中，或個別部署 WAR 檔案。如果個別部署 JSP，請讓 JSP 可在部署「商業程序編排器瀏覽器」或自訂用戶端的伺服器上使用。

如果您是針對程序及作業訊息使用自訂 JSP，則必須將用來部署 JSP 的 Web 模組對映至自訂 JSF 用戶端所對映的相同伺服器。

自訂的表單會在執行時期呈現在「商業程序編排器瀏覽器」中。

## 使用者定義的 JSP 片段

使用者定義的 JavaServer Pages (JSP) 片段包含在 HTML 表單標籤中。在執行時期，「商業程序編排器瀏覽器」將這些片段併入呈現的頁面中。

輸入訊息的使用者定義 JSP 片段內含在輸出訊息的 JSP 片段之前。

```
<html....>
  ...
  <form...>
    Input JSP (display task input message)

    Output JSP (display task output message)

  </form>
  ...
</html>
```

因為使用者定義的 JSP 片段內含在 HTML 表單標籤中，所以您可以新增輸入元素。輸入元素的名稱必須符合資料元素的「XML 路徑語言 (XPath)」表示式。將提供的字首值加在輸入元素名稱的字首很重要：

```
<input id="address"
       type="text"
       name="${prefix}/selectPromotionalGiftResponse/address"
       value="${messageMap['/selectPromotionalGiftResponse/address']}"
       size="60"
       align="left" />
```

字首值作為要求屬性提供。屬性會確保輸入名稱在括弧的表單中是唯一的。字首由「商業程序編排器瀏覽器」產生，且不應該進行變更：

```
String prefix = (String)request.getAttribute("prefix");
```

只有訊息可以在給定的環境定義中進行編輯時，才可以設定字首元素。根據人力作業的狀態，輸出資料可使用數種方式顯示。例如，如果作業處於已要求狀態，則可以修改輸出資料。然而，如果作業處於已完成狀態，則只可以顯示資料。在 JSP 片段中，您可以測試字首元素是否存在，並相對應地呈現訊息。以下 JSTL 陳述式會顯示您如何可以測試是否已設定字首元素。

```
...
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
...
<c:choose>
  <c:when test="${not empty prefix}">
    <!--Read/write mode-->
  </c:when>
  <c:otherwise>
    <!--Read-only mode-->
  </c:otherwise>
</c:choose>
```

---

## 第 16 章 建立外掛程式以自訂人力作業功能

「商業程序編排器」提供在處理人力作業期間所發生事件的事件處理基礎架構。也會提供外掛點，讓您調整功能，使其符合您的需求。您可以使用服務提供者介面 (SPI) 來建立自訂外掛程式，以處理事件和處理人員查詢。

您可以建立人力作業 API 事件及呈報通知事件的外掛程式。您也可以建立外掛程式，以處理從人員解析傳回的結果。例如，在尖峰期間，您可能會想要將使用者新增至結果清單中，以協助平衡工作量。

您可以將外掛程式登錄在不同層次 (廣域層次上的所有作業，應用程式元件中的作業、與作業範本相關聯的所有作業，或單一作業實例)。

---

### 建立 API 事件處理程式

在 API 方法操作人力作業時會發生 API 事件。請使用 API 事件處理程式外掛程式服務提供者介面 (SPI) 來建立外掛程式，以處理 API 或具有相等 API 事件的內部事件所傳送的作業事件。

請完成下列步驟來建立 API 事件處理程式。

1. 撰寫可以實作 `APIEventHandlerPlugin2` 介面或擴充 `APIEventHandler` 實作類別的類別。此類別可以呼叫其他類別的方法。
  - 如果使用 `APIEventHandlerPlugin2` 介面，則必須實作 `APIEventHandlerPlugin2` 介面及 `APIEventHandlerPlugin` 介面的所有方法。
  - 如果擴充 `SPI` 實作類別，請改寫需要的方法。

這個類別是在 Java 2 Enterprise Edition (J2EE) Enterprise JavaBeans (EJB) 應用程式的環境定義中執行。請確定此類別及其 `Helper` 類別遵循 EJB 規格。

**提示：** 如果想要從此類別呼叫 `HumanTaskManagerService` 介面，請不要呼叫方法以更新已產生事件的作業。此動作會導致資料庫死鎖。

2. 將外掛程式類別及其 `Helper` 類別組譯至 JAR 檔。

如果 `Helper` 類別是供數個 J2EE 應用程式使用，則可以將這些類別包裝在登錄為共用程式庫的不同 JAR 檔中。

3. 在 JAR 檔的 `META-INF/services/` 目錄中，建立外掛程式的服務提供者配置檔。

配置檔提供用來識別及載入外掛程式的機制。這個檔案符合 Java 2 服務提供者介面規格。

- a. 建立名稱為 `com.ibm.task.spi.plugin_nameAPIEventHandlerPlugin` 的檔案，其中 `plugin_name` 是外掛程式的名稱。

例如，如果外掛程式的名稱為 `Customer`，並且實作 `com.ibm.task.spi.APIEventHandlerPlugin` 介面，則配置檔的名稱就會是 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`。

- b. 在檔案的第一行 (不是備註行也不是空白行) 中，指定在步驟 1 中建立的外掛程式類別的完整名稱。

例如，如果外掛程式類別的名稱為 `MyAPIEventHandler`，並且位在 `com.customer.plugins` 套件中，則配置檔的第一行必須包含下列項目：  
`com.customer.plugins.MyAPIEventHandler`。

您擁有可安裝的 JAR 檔，而此檔案包含處理 API 事件的外掛程式，以及可以用來載入外掛程式的服務提供者配置檔。

**提示：** 您只可以有一個 `eventHandlerName` 內容，用來同時登錄 API 事件處理程式及通知事件處理程式。如果想要同時使用 API 事件處理程式及通知事件處理程式，則外掛程式實作的名稱必須相同，例如，`Customer` 作為 SPI 實作的事件處理程式名稱。

您可以使用單一類別或兩個不同類別來實作這兩個外掛程式。在這兩種情況下，則需要在 JAR 檔的 `META-INF/services/` 目錄中建立兩個檔案 (例如，`com.ibm.task.spi.CustomerNotificationEventHandlerPlugin` 及 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`)。

將外掛程式實作及 `Helper` 類別包裝在單一 JAR 檔中。

您現在需要安裝及登錄外掛程式，讓人力作業儲存器可以在執行時期使用此外掛程式。您可以登錄具有作業實例、作業範本或應用程式元件的 API 事件處理程式。

#### 相關概念

第 49 頁的『呼叫作業的範例情況』

此處說明可以用來呼叫作業的各種方式。

## API 事件處理程式

當修改人力作業或變更其狀態時，會發生 API 事件。若要處理這些 API 事件，會在修改作業之前 (事件前方法)，及 API 呼叫剛傳回之前 (事件後方法)，直接呼叫事件處理程式。

如果事件前方法擲出 `ApplicationVetoException` 異常，則不會執行 API 動作，異常會傳回到 API 呼叫者，且也會回復與事件相關聯的交易。如果內部事件觸發事件前方法且擲出 `ApplicationVetoException` 異常，則不會執行內部事件 (例如自動要求)，但異常也不會傳回到用戶端應用程式。在此情況下，會將參考訊息寫入 `SystemOut.log` 檔。如果 API 方法在處理期間擲出異常，則會捕捉該異常並傳遞至事件後方法。在事件後方法傳回之後，異常會重新傳遞至呼叫者。

下列規則適用於事件前方法：

- 事件前方法會接收關聯 API 方法或內部事件的參數。
- 事件前方法可以擲出 `ApplicationVetoException` 異常，來防止處理繼續進行。

下列規則適用於事件後方法：

- 事件後方法會接收提供給 API 呼叫的參數，以及回覆值。如果 API 方法實作擲出異常，則事件後方法也會接收到異常。
- 事件後方法無法修改回覆值。
- 事件後方法無法擲出異常；會記錄執行時期異常，但會將其忽略。

若要實作 API 事件處理程式，您可以使用 `APIEventHandlerPlugin2` 介面，其會延伸 `APIEventHandlerPlugin` 介面，或延伸預設 `com.ibm.task.spi.APIEventHandler` SPI 實作類別。如果您的事件處理程式繼承自預設實作類別，則它一律實作最新版本的 SPI。如果您升級至更新版本的「商業程序編排器」，則僅需略做變更就可以利用新的 SPI 方法。

如果您同時具有通知事件處理程式及 API 事件處理程式，則這兩個事件處理程式必須具有相同的名稱，因為您只能登錄一個事件處理程式名稱。

---

## 建立通知事件處理程式

通知事件是在呈報人力作業時產生。「商業程序編排器」提供用來處理呈報的功能 (例如建立呈報工作項目或傳送電子郵件)。您可以建立通知事件處理程式，以自訂呈報的處理方式。

若要實作通知事件處理程式，可以使用 `NotificationEventHandlerPlugin` 介面，也可以擴充預設 `com.ibm.task.spi.NotificationEventHandler` 服務提供者介面 (SPI) 實作類別。

請完成下列步驟來建立通知事件處理程式。

1. 撰寫可以實作 `NotificationEventHandlerPlugin` 介面或擴充 `NotificationEventHandler` 實作類別的類別。此類別可以呼叫其他類別的方法。

如果使用 `NotificationEventHandlerPlugin` 介面，則必須實作所有介面方法。如果擴充 SPI 實作類別，請改寫需要的方法。

這個類別是在 Java 2 Enterprise Edition (J2EE) Enterprise JavaBeans (EJB) 應用程式的環境定義中執行。請確定此類別及其 `Helper` 類別遵循 EJB 規格。

此外掛程式是使用 `EscalationUser` 角色的權限進行呼叫。配置人力作業儲存器時，會定義此角色。

**提示:** 如果想要從此類別呼叫 `HumanTaskManagerService` 介面，請不要呼叫方法以更新已產生事件的作業或呈報。此動作會導致資料庫死鎖。

2. 將外掛程式類別及其 `Helper` 類別組譯至 JAR 檔。

如果 `Helper` 類別是供數個 J2EE 應用程式使用，則可以將這些類別包裝在登錄為共用程式庫的不同 JAR 檔中。

3. 在 JAR 檔的 `META-INF/services/` 目錄中，建立外掛程式的服務提供者配置檔。

配置檔提供用來識別及載入外掛程式的機制。這個檔案符合 Java 2 服務提供者介面規格。

- a. 建立名稱為 `com.ibm.task.spi.plug-in_nameNotificationEventHandlerPlugin` 的檔案，其中 `plug-in_name` 是外掛程式的名稱。

例如，如果外掛程式的名稱為 `HelpDeskRequest` (事件處理程式名稱)，並且實作 `com.ibm.task.spi.NotificationEventHandlerPlugin` 介面，則配置檔的名稱就會是 `com.ibm.task.spi.HelpDeskRequestNotificationEventHandlerPlugin`。

- b. 在檔案的第一行 (不是備註行也不是空白行) 中，指定在步驟 1 中建立的外掛程式類別的完整名稱。



例如，如果外掛程式類別的名稱為 `MyEventHandler`，並且位在 `com.customer.plugins` 套件中，則配置檔的第一行必須包含下列項目：  
`com.customer.plugins.MyEventHandler`。

您擁有可安裝的 JAR 檔，而此檔案包含處理通知事件的外掛程式，以及可以用來載入外掛程式的服務提供者配置檔。您可以登錄具有作業實例、作業範本或應用程式元件的 API 事件處理程式。

**提示：** 您只可以有一個 `eventHandlerName` 內容，用來同時登錄 API 事件處理程式及通知事件處理程式。如果想要同時使用 API 事件處理程式及通知事件處理程式，則外掛程式實作的名稱必須相同，例如，`Customer` 作為 SPI 實作的事件處理程式名稱。

您可以使用單一類別或兩個不同類別來實作這兩個外掛程式。在這兩種情況下，則需要在 JAR 檔的 `META-INF/services/` 目錄中建立兩個檔案（例如，`com.ibm.task.spi.CustomerNotificationEventHandlerPlugin` 及 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`）。

將外掛程式實作及 `Helper` 類別包裝在單一 JAR 檔中。

您現在需要安裝及登錄外掛程式，讓人力作業儲存器可以在執行時期使用此外掛程式。您可以登錄具有作業實例、作業範本或應用程式元件的通知事件處理程式。

#### 相關概念

第 42 頁的『呈報』

呈報是人力作業未在指定的時間量動作時自動引發的警示。例如，如果作業未要求，或者未在定義的時間限制內完成。您可以為作業指定一個以上的呈報。這些呈報可以平行啟動，或者作為呈報鏈結啟動。

---

## 建立對人員查詢結果進行後處理的外掛程式

人員解析會傳回指派給特定角色的使用者清單（例如，作業的可能擁有者）。您可以建立外掛程式，以變更人員解析所傳回的人員查詢結果。例如，若要改善工作量平衡，則您可能會有外掛程式，從查詢結果中移除已有高工作量的使用者。

您只可以有一個後處理外掛程式；這表示外掛程式必須處理所有作業的人員查詢結果。您的外掛程式可以新增或移除使用者，或者變更使用者或群組資訊。例如，也可以將結果類型從使用者清單變更為群組或每個人。

因為外掛程式是在人員解析完成之後執行，所以已套用任何必須保留機密性或安全的規則。外掛程式會接收在人員解析期間所移除使用者（在 `HTM_REMOVED_USERS` 對映機碼中）的相關資訊。您必須確定外掛程式使用此環境定義資訊來保留任何可能具有的機密性或安全規則。

若要實作人員查詢結果的後處理，請使用 `StaffQueryResultPostProcessorPlugin` 介面。介面具有的方法可以修改作業、呈報、作業範本及應用程式元件的查詢結果。

請完成下列步驟，建立對人員查詢結果進行後處理的外掛程式。

1. 撰寫可以實作 `StaffQueryResultPostProcessorPlugin` 介面的類別。

您必須實作所有介面方法。此類別可以呼叫其他類別的方法。



這個類別是在 Java 2 Enterprise Edition (J2EE) Enterprise JavaBeans (EJB) 應用程式的環境定義中執行。請確定此類別及其 Helper 類別遵循 EJB 規格。

**提示:** 如果想要從此類別呼叫 HumanTaskManagerService 介面，請不要呼叫方法以更新已產生事件的作業。此動作會導致資料庫死鎖。

下列範例顯示如何變更為 SpecialTask 的作業的編輯者角色。

```
public StaffQueryResult processStaffQueryResult
    (StaffQueryResult originalStaffQueryResult,
     Task task,
     int role,
     Map context)
{
    StaffQueryResult newStaffQueryResult = originalStaffQueryResult;
    StaffQueryResultFactory staffResultFactory =
        StaffQueryResultFactory.newInstance();
    if (role == com.ibm.task.api.WorkItem.REASON_EDITOR &&
        task.getName() != null &&
        task.getName().equals("SpecialTask"))
    {
        UserData user = staffResultFactory.newUserData
            ("SuperEditor",
             new Locale("en-US"),
             "SuperEditor@company.com");
        ArrayList userList = new ArrayList();
        userList.add(user);

        newStaffQueryResult = staffResultFactory.newStaffQueryResult(userList);
    }
    return(newStaffQueryResult);
}
```

2. 將外掛程式類別及其 Helper 類別組譯至 JAR 檔。

如果 Helper 類別是供數個 J2EE 應用程式使用，則可以將這些類別包裝在登錄為共用程式庫的不同 JAR 檔中。

3. 在 JAR 檔的 META-INF/services/ 目錄中，建立外掛程式的服務提供者配置檔。

配置檔提供用來識別及載入外掛程式的機制。這個檔案符合 Java 2 服務提供者介面規格。

- a. 建立名稱為 `com.ibm.task.spi.plugin_nameStaffQueryResultPostProcessorPlugin` 的檔案，其中 `plug-in_name` 是外掛程式的名稱。

例如，如果外掛程式的名稱為 `MyHandler`，並且實作 `com.ibm.task.spi.StaffQueryResultPostProcessorPlugin` 介面，則配置檔的名稱就會是 `com.ibm.task.spi.MyHandlerStaffQueryResultPostProcessorPlugin`。

- b. 在檔案的第一行 (不是備註行也不是空白行) 中，指定在步驟 1 中建立的外掛程式類別的完整名稱。

例如，如果外掛程式類別的名稱為 `StaffPostProcessor`，並且位在 `com.customer.plugins` 套件中，則配置檔的第一行必須包含下列項目：`com.customer.plugins.StaffPostProcessor`。您擁有可安裝的 JAR 檔，而此檔案包含可以對人員查詢結果進行後處理的外掛程式，以及可以用來載入外掛程式的服務提供者配置檔。

4. 安裝外掛程式。

人員查詢結果只可以有一個後處理外掛程式。您必須將外掛程式安裝為共用程式庫。

5. 登錄外掛程式。
  - a. 在管理主控台中，跳至「人力作業管理程式」的「自訂內容」頁面（應用程式伺服器 → **server\_name** → 人力作業儲存器 → 自訂內容）。
  - b. 新增名稱爲 **Staff.PostProcessorPlugin** 的自訂內容，以及給予外掛程式的名稱值（在此範例中爲 MyHandler）。

#### 相關概念

第 77 頁的『共用人員指派』

若爲特定的作業角色，則作業範本的所有實例都會使用相同的人員指派準則。這是因爲所有作業實例都是從相同的作業範本進行實例化。爲了避免重新執行人員查詢，作業範本的作業實例會共用查詢的結果。

---

## 安裝外掛程式

若要使用外掛程式，則必須安裝外掛程式，讓作業儲存器可以存取它。

安裝外掛程式的方式是根據外掛程式是只供一個 Java 2 Enterprise Edition (J2EE) 應用程式使用還是由數個應用程式使用而定。

請完成下列其中一個步驟，以安裝外掛程式。

- 安裝外掛程式，供單一 J2EE 應用程式使用。

將外掛程式 JAR 檔新增至應用程式 EAR 檔。在 WebSphere Integration Developer 的部署描述子編輯器中，將外掛程式的 JAR 檔安裝爲主要 Enterprise JavaBeans (EJB) 模組的 J2EE 應用程式的專案公用程式 JAR 檔。

- 安裝外掛程式，供數個 J2EE 應用程式使用。

將 JAR 檔放入 WebSphere Application Server 共用程式庫中，並建立程式庫與需要存取外掛程式的應用程式的關聯性。若要可以在網路部署環境中使用 JAR 檔，請在每部伺服器上以手動方式分送 JAR 檔，然後針對每個 Cell 安裝一次共用程式庫。

您現在可以登錄外掛程式。

---

## 登錄外掛程式

您可以在作業儲存器成品階層的不同層次上登錄外掛程式。例如，廣域層次上的所有作業、應用程式元件的作業、與作業範本相關聯的所有作業，或單一作業實例。

登錄多個外掛程式時，支援範圍設定。這表示使用的是登錄在低階作業儲存器成品階層上的外掛程式（如作業實例），而不是使用登錄在高階上的外掛程式（如作業範本或應用程式元件）。所有階層層次都支援範圍設定。而作業儲存器會使用登錄在最低階層層次上的外掛程式。

您可以使用下列其中一種方式來登錄外掛程式。

- 在作業模型中登錄外掛程式。

在 WebSphere Integration Developer 的作業編輯器的作業內容區域的「明細」頁面中，於事件處理程式名稱欄位中指定事件處理程式的名稱。

- 登錄在執行時期建立的特定作業或作業範本的外掛程式。

使用 `TTask` 類別的 `setEventHandlerName` 方法，以登錄事件處理程式的名稱。

- 在執行時期，變更作業實例的已登錄事件處理程式。

使用 `update(Task task)` 方法，以在執行時期使用作業實例的不同事件處理程式。呼叫者必須具有作業管理者權限，才能更新此內容。

- 在廣域層次上登錄外掛程式。

在人力作業儲存器的「自訂內容」頁面的管理主控台上，定義外掛程式的自訂內容。自訂內容的值即是外掛程式名稱。



---

## 第 17 章 安裝商業程序和人力作業應用程式

您可以將包含商業程序、人力作業或這兩者的「服務元件架構 (SCA)」模組配送至部署目標。部署目標可以是伺服器或叢集。

對於您要安裝應用程式的每一個應用程式伺服器或叢集，驗證是否已安裝及配置「商業流程管理程式」和「人力作業管理程式」(或兩者)。

例如，您可以從管理主控台、指令行或執行管理 Script 來安裝商業程序和作業應用程式。

在安裝商業程序或人力作業應用程式之後，所有商業程序範本及人力作業範本都會進入啟動狀態中。您可以依據這些範本來建立程序實例和作業實例。

您必須啟動應用程式，才能建立程序實例或作業實例。

### 相關概念

第 478 頁的『商業程序及人力作業的部署』

WebSphere Integration Developer 或服務部署產生程序或作業的部署碼時，每個程序元件或作業元件都會對映至一個階段作業 Enterprise Bean。所有部署碼都會包裝在企業應用程式 (EAR) 檔案中。此外，針對每個程序，在企業應用程式安裝期間，代表此程序中 Java 程式碼的 Java 類別都會產生並內嵌於 EAR 檔案中。要部署的每個新版本模型都必須封裝在新的企業應用程式中。

『如何在網路部署環境中安裝商業程序及人力作業應用程式』

在網路部署環境中安裝程序範本或人力作業範本時，應用程式安裝會自動執行下列動作。

---

### 如何在網路部署環境中安裝商業程序及人力作業應用程式

在網路部署環境中安裝程序範本或人力作業範本時，應用程式安裝會自動執行下列動作。

應用程式是以非同步方式安裝在階段中。每個階段都必須順利完成，才能開始下一個階段。

1. 應用程式安裝開始於部署管理程式。

在此階段期間，會在 WebSphere 配置儲存庫中配置商業程序範本及人力作業範本。也會驗證應用程式。如果發生錯誤，則會在 System.out 檔案、System.err 檔案或在部署管理程式上的 FFDC 項目中報告這些錯誤。

2. 應用程式安裝會繼續在節點代理程式上繼續進行。

在此階段期間，會觸發某個應用程式伺服器實例上的應用程式安裝。此應用程式伺服器實例可能是部署目標的一部分，或是部署目標。如果部署目標是具有多個叢集成員的叢集，則從此叢集的叢集成員中任意選擇伺服器實例。如果在此階段期間發生錯誤，則會在 SystemOut.log 檔案、SystemErr.log 檔案或在節點代理程式上的 FFDC 項目中報告這些錯誤。

3. 應用程式會在伺服器實例上執行。

在此階段期間，會將程序範本及人力範本部署至部署目標上的「商業程序編排器」資料庫。如果發生錯誤，則會在 `System.out` 檔案、`SystemErr.log` 檔案或在此伺服器實例上的 FFDC 項目中報告這些錯誤。

#### 相關工作

第 477 頁的第 17 章, 『安裝商業程序和人力作業應用程式』

您可以將包含商業程序、人力作業或這兩者的「服務元件架構 (SCA)」模組配送至部署目標。部署目標可以是伺服器或叢集。

---

## 商業程序及人力作業的部署

WebSphere Integration Developer 或服務部署產生程序或作業的部署碼時，每個程序元件或作業元件都會對映至一個階段作業 Enterprise Bean。所有部署碼都會包裝在企業應用程式 (EAR) 檔案中。此外，針對每個程序，在企業應用程式安裝期間，代表此程序中 Java 程式碼的 Java 類別都會產生並內嵌於 EAR 檔案中。要部署的每個新版本模型都必須封裝在新的企業應用程式中。

當您安裝包含商業程序或人力作業的企業應用程式時，這些項目會適當地在「商業程序編排器」資料庫中儲存為商業程序範本或人力作業範本。依預設，新安裝的範本會處於已啟動狀態。然而，新安裝的企業應用程式會處於已停止狀態。每個已安裝的企業應用程式都可以個別啟動及停止。

您可以部署許多不同版本的程序範本或作業範本，每個範本位於不同的企業應用程式中。安裝新的企業應用程式時，會如下所示判斷安裝的範本版本：

- 如果範本名稱及目標名稱空間不存在，則會安裝新的範本
- 如果範本名稱及目標名稱空間與現有範本的範本名稱及目標名稱空間相同，但是有效起始日期不同，則會安裝現有範本的新版本

**註：** 範本名稱是衍生自元件的名稱，而不是衍生自商業程序或人力作業。

如果您沒有指定有效起始日期，則日期決定方式如下所示：

- 如果使用 WebSphere Integration Developer，則有效起始日期是建立人力作業或商業程序模型的日期。
- 如果使用服務部署，則有效起始日期是執行 `serviceDeploy` 指令的日期。只有協同作業會取得安裝應用程式的日期，以作為有效起始日期。

#### 相關工作

第 477 頁的第 17 章, 『安裝商業程序和人力作業應用程式』

您可以將包含商業程序、人力作業或這兩者的「服務元件架構 (SCA)」模組配送至部署目標。部署目標可以是伺服器或叢集。

---

## 以互動方式安裝商業程序及人力作業應用程式

您可以在執行時期使用 `wsadmin` 工具和 `installInteractive Script`，以互動方式安裝應用程式。如果因為您使用管理主控台來安裝應用程式而無法變更某些設定，您可以使用此 `Script` 來變更這些設定。

執行下列步驟，以互動方式來安裝商業程序應用程式。

1. 啟動 `wsadmin` 工具。

在 `profile_root/bin` 目錄中，輸入 `wsadmin`。

## 2. 安裝應用程式。

在 `wsadmin` 命令行提示上，輸入下列指令：

```
$AdminApp installInteractive application.ear
```

其中 `application.ear` 是含有程序應用程式的企業保存檔的完整名稱。將會透過一連串作業來提示您，讓您變更應用程式的值。

## 3. 儲存配置變更。

在 `wsadmin` 命令行提示上，輸入下列指令：

```
$AdminConfig save
```

您必須儲存變更，才能將更新傳送至主要配置儲存庫。如果 `Scripting` 程序結束，但您尚未儲存變更，則會捨棄變更。

## 配置程序應用程式資料來源及設定參照設定

對於特定的資料庫基礎架構，您可能需要配置程序應用程式來執行 `SQL` 陳述式。這些 `SQL` 陳述式可能來自資訊服務活動，也可能是您在安裝程序或啟動實例期間執行的陳述式。

在安裝應用程式時，您可以指定下列類型的資料來源：

- 在程序安裝期間用來執行 `SQL` 陳述式的資料來源
- 在啟動程序實例期間用來執行 `SQL` 陳述式的資料來源
- 用來執行 `SQL` 片段活動的資料來源

執行 `SQL` 片段活動所需的資料來源是在 `tDataSource` 類型的 `BPEL` 變數中定義。`SQL` 片段活動所需的資料庫綱目和表格名稱是在 `tSetReference` 類型的 `BPEL` 變數中定義。您可以為這兩個變數配置起始值。

您可以使用 `wsadmin` 工具來指定資料來源。

1. 使用 `wsadmin` 工具，以互動方式安裝程序應用程式。
2. 逐步執行作業，直到您抵達更新資料來源和設定參照的作業為止。

為您的環境配置這些設定。下列範例顯示您在這些作業中可以變更的設定。

## 3. 儲存變更。

### 範例：使用 `wsadmin` 工具來更新資料來源和設定參照

對於安裝程序期間或啟動程序時所使用的起始變數值和陳述式，您可以在**更新資料來源**作業中變更資料來源值。在**更新設定參照**作業中，您可以配置與資料庫綱目和表格名稱相關的設定。

Task [24]: Updating data sources

```
//Change data source values for initial variable values at process start
```

```
Process name: Test
// Name of the process template
Process start or installation time: Process start
// Indicates whether the specified value is evaluated
//at process startup or process installation
```



```

Statement or variable: Variable
// Indicates that a data source variable is to be changed
Data source name: MyDataSource
// Name of the variable
JNDI name:[jdbc/sample]:jdbc/newName
// Sets the JNDI name to jdbc/newName
Task [25]: Updating set references

// Change set reference values that are used as initial values for BPEL variables

Process name: Test
// Name of the process template
Variable: SetRef
// The BPEL variable name
JNDI name:[jdbc/sample]:jdbc/newName
// Sets the JNDI name of the data source of the set reference to jdbc/newName
Schema name: [IISAMPLE]
// The name of the database schema
Schema prefix: []:
// The schema name prefix.
// This setting applies only if the schema name is generated.
Table name: [SETREFTAB]: NEWTABLE
// Sets the name of the database table to NEWTABLE
Table prefix: []:
// The table name prefix.
// This setting applies only if the prefix name is generated.

```

---

## 使用管理主控台來解除安裝商業程序和人力作業應用程式

您可以使用管理主控台，解除安裝包含商業程序或人力作業的應用程式。

若要解除安裝包含商業程序或人力作業的應用程式，必須符合下列必要條件：

- 如果已在獨立式伺服器上安裝應用程式，則伺服器必須執行中，而且有權存取「商業程序編排器」資料庫。
- 如果已在叢集上安裝應用程式，則部署管理程式及至少一個叢集成員必須正在執行中。叢集成員有權存取「商業程序編排器」資料庫。
- 如果已在受管理伺服器安裝應用程式，則部署管理程式及這個伺服器必須正在執行中。伺服器必須有權存取「商業程序編排器」資料庫。
- 所有屬於應用程式的商業程序範本及人力作業範本必須處於停止狀態。
- 沒有任何商業程序或人力作業範本實例呈現在任何狀態中。
- 

若為作為開發及單元測試環境的獨立式伺服器環境，伺服器可以配置為在開發模式中執行。這個配置不需要停止範本，而且不會呈現任何實例。不過，這個配置對於正式作業環境而言無效。

若要解除安裝含有商業程序或人力作業的企業應用程式，請執行下列動作：

1. 停止應用程式中的所有程序和作業範本。

此動作會防止建立程序和作業實例。

- a. 在管理主控台導覽窗格中，按一下**應用程式** → **SCA 模組**。
- b. 選取包含要停止之範本的模組。
- c. 在「其他內容」下，視需要按一下**商業程序**或**人力作業**或這兩者。
- d. 按一下適當的勾選框，選取所有程序和作業範本。

e. 按一下**停止**。

對於含有商業程序範本或人力作業範本的所有 EJB 模組，重複執行此步驟。

2. 驗證資料庫、每個叢集至少有一個應用程式伺服器，以及已部署應用程式的獨立式伺服器正在執行中。

在網路部署環境中，對於已安裝應用程式的每一個叢集，部署管理程式、所有受管理的獨立式應用程式伺服器，以及至少一個應用程式伺服器必須正在執行中。

3. 驗證應用程式沒有任何商業程序實例或人力作業實例。

必要的話，管理者可以使用「商業程序編排器瀏覽器」來刪除任何程序或作業實例。

4. 停止並解除安裝應用程式：
  - a. 在管理主控台導覽窗格中，按一下**應用程式** → **企業應用程式**。
  - b. 選取您要解除安裝的應用程式，然後按一下**停止**。

如果應用程式中仍有任何程序實例或作業實例存在，則此步驟會失敗。

- c. 再次選取您要解除安裝的應用程式，然後按一下**解除安裝**。
  - d. 按一下**儲存**，儲存變更。

應用程式已解除安裝。

---

## 使用管理指令來解除安裝商業程序和人力作業應用程式

除了使用管理主控台之外，管理指令也可用來解除安裝含有商業程序或人力作業的應用程式。

若要解除安裝包含商業程序或人力作業的應用程式，必須符合下列必要條件：

- 如果已在獨立式伺服器上安裝應用程式，則伺服器必須執行中，而且有權存取「商業程序編排器」資料庫。
- 如果已在叢集上安裝應用程式，則部署管理程式及至少一個叢集成員必須正在執行中。叢集成員有權存取「商業程序編排器」資料庫。
- 如果已在受管理伺服器安裝應用程式，則部署管理程式及這個伺服器必須正在執行中。伺服器必須有權存取「商業程序編排器」資料庫。
- 所有屬於應用程式的商業程序範本及人力作業範本必須處於停止狀態。
- 沒有任何商業程序或人力作業範本實例呈現在任何狀態中。
- 

若為作為開發及單元測試環境的獨立式伺服器環境，伺服器可以配置為在開發模式中執行。這個配置不需要停止範本，而且不會呈現任何實例。不過，這個配置對於正式作業環境而言無效。

此外，如果已啓用廣域安全，也請驗證您的使用者 ID 是否具有操作員權限。

請確定管理用戶端連接的伺服器程序正在執行中。為了確保管理用戶端會自動連接伺服器程序，請勿使用 `-conntype NONE` 選項作為指令選項。

下列步驟說明如何使用 `bpcTemplates.jacl` Script 來解除安裝含有商業程序範本或人力作業範本的應用程式。您必須停止範本，才能解除安裝範本所屬的應用程式。您可以使用 `bpcTemplates.jacl` Script，只需要一個步驟就可以停止並解除安裝範本。

在解除安裝應用程式之前，您可以刪除與應用程式中的範本相關聯的程序實例或作業實例，例如，使用「商業程序編排器瀏覽器」。您也可以在 `bpcTemplates.jacl` Script 上使用 **-force** 選項，只需要一個步驟就可以刪除與範本相關聯的任何實例、停止範本，然後解除安裝範本。

**注意:**

因為 **-force** 選項會刪除所有程序實例和作業實例資料，請小心使用這個選項。

1. 切換至「商業程序編排器」範例目錄。

在 Windows 平台上，輸入：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 及 iOS 平台上，輸入：

```
cd install_root/ProcessChoreographer/admin
```

2. 停止範本並解除安裝對應的應用程式。

在 Windows 平台上，輸入：

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        [-user user_name]  
                        [-password user_password]  
                        -uninstall application_name  
                        [-force]
```

在 Linux、UNIX 及 iOS 平台上，輸入：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        [-user user_name]  
                        [-password user_password]  
                        -uninstall application_name  
                        [-force]
```

其中：

*user\_name*

如果已啟用廣域安全原則，請提供鑑別用的使用者 ID。

*user\_password*

如果已啟用廣域安全原則，請提供鑑別用的使用者密碼。

*application\_name*

如果已啟用廣域安全原則，請提供鑑別用的使用者密碼。

應用程式已解除安裝。

---

## 第 5 篇 監視商業程序及作業



---

## 第 18 章 監視商業程序及人力作業

監視程序及人力作業是透過 WebSphere Integration Developer 中的監視窗格來控制。不管是否將啓用審核追蹤或是否將產生事件，都必須遵循這種方式。

WebSphere Process Server 包括 共用事件基礎架構，它會提供管理事件資料的標準格式及機制。

每當發生需要監視的情況，而且 共用事件基礎架構 服務可用時，「商業程序編排器」就會發出事件。這些事件符合「共用基礎事件」規格。您可以使用一般工具來處理這些事件。

您也可以使用 Java Snippet，來建立及傳送使用者資料事件。如需相關資訊，請參閱有關傳送事件的 共用事件基礎架構 文件。





---

## 第 19 章 監視商業程序事件

為代表商業程序而產生的事件，由與狀況無關的資料，以及商業程序事件特定的資料組成。已說明商業程序事件特定的屬性及元素。

商業程序事件可以擁有以下種類的事件內容。

---

### 商業程序的特定事件資料

在商業程序中，事件與程序、活動、範圍、鏈結及變數相關。會說明其中每個事件類型的物件特定內容。

若為「商業程序編排器」6.1 版，會發生兩種事件格式：

#### WebSphere Business Monitor 6.0.2 格式

若在 WebSphere Integration Developer 6.0.2 內有模型化程序，或在 WebSphere Integration Developer 6.1 內已啟用 WebSphere Business Monitor 6.0.2 格式模式，則會發生 WebSphere Business Monitor 6.0.2 格式事件。如果未另行指定，則這些事件的物件特定內容會撰寫為字串類型的 *extendedDataElement* XML 元素。

#### WebSphere Business Monitor 6.1 格式

若在 WebSphere Integration Developer 6.1 內有模型化程序，且已啟用 WebSphere Business Monitor 6.1 格式模式，則會發生 WebSphere Business Monitor 6.1 格式事件。這些事件的物件特定內容會撰寫為「共用基礎事件」的 *eventPointData* 資料夾內的 *xs:any* 插槽中的 XML 元素，且承載量訊息會寫入 *applicationData* 區段中。XML 的結構定義於「XML 綱目定義 (XSD)」檔案 *BFMEvents.xsd* 中。此檔案可在 *install\_root\ProcessChoreographer\client* 目錄中找到。

### 程序

程序實例的事件具有下列物件特定的事件內容：

屬性	說明
<i>processTemplateName</i>	實例衍生來源的程序範本名稱
<i>processTemplateValidFrom</i>	範本的有效起始日期
<i>processTemplateId</i>	程序範本的 ID
<i>processInstanceDescription</i>	選用項目：程序實例的說明

屬性	說明
processInstanceExecutionState	代表程序狀態的字串值。格式為： <i>state number-state description</i> 。此屬性可以具有下列其中一個值：  1 - STATE_READY 2 - STATE_RUNNING 3 - STATE_FINISHED 4 - STATE_COMPENSATING 5 - STATE_FAILED 6 - STATE_TERMINATED 7 - STATE_COMPENSATED 8 - STATE_TERMINATING 9 - STATE_FAILING 10 - STATE_INDOUBT 11 - STATE_SUSPENDED 12 - STATE_COMPENSATION_FAILED
principal	代表其執行現行執行步驟的主體。這通常是程序的起始者。
PayloadType	承載量類型。字串的值可以是下列其中一個：none、digest 或 full。此值取決於 WebSphere Integration Developer 中的設定。如果沒有可用的明確監視定義，例如，是否已選取 <b>啟用預設事件</b> ，則預設承載量類型為 full。

## 活動及範圍

活動及範圍具有下列物件特定的事件內容：

屬性	說明
processTemplateName	實例衍生來源的程序範本名稱。
processTemplateValidFrom	範本的有效起始日期。
activityTemplateName	選用項目：實例衍生來源的活動範本名稱。
activityInstanceDescription	選用項目：活動實例的說明。

屬性	說明
activityKind	<p>識別活動類型的字串值。此值具有下列格式：<i>kind number-kind description</i>。此屬性可以具有下列其中一個值：</p> <ul style="list-style-type: none"> <li>3 - KIND_EMPTY</li> <li>21 - KIND_INVOKE</li> <li>23 - KIND_RECEIVE</li> <li>24 - KIND_REPLY</li> <li>25 - KIND_THROW</li> <li>26 - KIND_TERMINATE</li> <li>27 - KIND_WAIT</li> <li>29 - KIND_COMPENSATE</li> <li>30 - KIND_SEQUENCE</li> <li>32 - KIND_SWITCH</li> <li>34 - KIND_WHILE</li> <li>36 - KIND_PICK</li> <li>38 - KIND_FLOW</li> <li>40 - KIND_SCOPE</li> <li>42 - KIND_SCRIPT</li> <li>43 - KIND_STAFF</li> <li>44 - KIND_ASSIGN</li> <li>45 - KIND_CUSTOM</li> <li>46 - KIND_RETHROW</li> <li>47 - KIND_FOR_EACH_SERIAL</li> <li>48 - KIND_FOR_EACH_PARALLEL</li> <li>1000 - SQLSnippet</li> <li>1001 - RetrieveSet</li> <li>1002 - InvokeInformationService</li> <li>1003 - AtomicSQLSnippetSequence</li> </ul>
狀態	<p>代表活動狀態的字串值。格式為：<i>state number-state description</i>。請注意，活動的狀態碼不同於用於程序的狀態碼。此屬性可以具有下列其中一個值：</p> <ul style="list-style-type: none"> <li>1 - STATE_INACTIVE</li> <li>2 - STATE_READY</li> <li>3 - STATE_RUNNING</li> <li>4 - STATE_SKIPPED</li> <li>5 - STATE_FINISHED</li> <li>6 - STATE_FAILED</li> <li>7 - STATE_TERMINATED</li> <li>8 - STATE_CLAIMED</li> <li>9 - STATE_TERMINATING</li> <li>10 - STATE_FAILING</li> <li>11 - STATE_WAITING</li> <li>12 - STATE_EXPIRED</li> <li>13 - STATE_STOPPED</li> </ul>
bpelId	代表活動的 wpc:id 屬性的字串值。
PayloadType	<p>承載量類型。字串的值可以是下列其中一個：none、digest 或 full。此值取決於 WebSphere Integration Developer 中的設定。如果沒有可用的明確監視定義，例如，是否已選取啟用預設事件，則預設承載量類型為 full。</p>

## 鏈結

鏈結具有下列物件特定的事件內容：

屬性	說明
processTemplateName	實例衍生來源的程序範本名稱
processTemplateValidFrom	範本的有效起始日期
flowBpelId	代表包含鏈結的流程活動之 wpc:id 屬性的字串值。
elementName	已評估鏈結的名稱
說明	鏈結的說明。只有在程序模型中指定此屬性時，才會包含此屬性。
PayloadType	承載量類型。字串的值可以是下列其中一個：none、digest 或 full。此值取決於 WebSphere Integration Developer 中的設定。如果沒有可用的明確監視定義，例如，是否已選取 <b>啟用預設事件</b> ，則預設承載量類型為 full。

## 變數

變數具有下列物件特定的事件內容。

屬性	說明
processTemplateName	實例衍生來源的程序範本名稱。
processTemplateValidFrom	範本的有效起始日期。
variableName	已變更變數的名稱。
variableData	<p>要求 WBI Monitor 相容事件時產生。變數內容的 XML 表示法。資料物件的每個內容都以巢狀延伸資料元素形式報告。元素類型可能是具有適當值的 'boolean' 或 'string' 類型。如果變數 <i>variableName</i> 尚未起始設定，則沒有 <i>variableData</i> 元素。</p> <p>此屬性只用於 WebSphere Business Monitor 6.0.2 格式的事件。若為 WebSphere Business Monitor 6.1 格式事件，變數的內容會寫入 <i>applicationData</i> 區段，它包含其名稱設為變數名稱的一個內容元素。</p>
variableData_BO	<p>要求非 WBI Monitor 相容事件時產生。此元素是 'noValue' 類型並包含變數內容的 XML 表示法。資料物件的每個內容都以巢狀延伸資料元素形式報告。如果變數 <i>variableName</i> 尚未起始設定，則沒有 <i>VariableData_BO</i> 元素。</p> <p>此屬性只用於 WebSphere Business Monitor 6.0.2 格式的事件。若為 WebSphere Business Monitor 6.1 格式事件，變數的內容會寫入 <i>applicationData</i> 區段，它包含其名稱設為變數名稱的一個內容元素。</p>
bpelId	代表活動的 wpc:id 屬性的字串值。
PayloadType	承載量類型。字串的值可以是下列其中一個：none、digest 或 full。此值取決於 WebSphere Integration Developer 中的設定。如果沒有可用的明確監視定義，例如，是否已選取 <b>啟用預設事件</b> ，則預設承載量類型為 full。

## 商業程序事件的延伸名稱

此延伸名稱指出事件的承載量。您可在這裡找到商業程序事件的所有延伸名稱及其對應承載量的清單。

此延伸名稱包含用作「共用基礎事件」的 *extensionName* 屬性值的字串值。這也是提供事件其他相關資料的 XML 元素的名稱。事件元素的名稱為大寫，例如 BPC.BFM.BASE，XML 元素的名稱為大小寫混合格式，例如，*BPCEventCode*。除另行指出外，所有資料元素都是 string 類型。

下列延伸名稱可用於商業程序事件：

- 『BPC.BFM.BASE』
- 第 492 頁的 『BPC.BFM.PROCESS.BASE』
- 第 492 頁的 『BPC.BFM.PROCESS.STATUS』
- 第 493 頁的 『BPC.BFM.PROCESS.START』
- 第 493 頁的 『BPC.BFM.PROCESS.FAILURE』
- 第 493 頁的 『BPC.BFM.PROCESS.CORREL』
- 第 493 頁的 『BPC.BFM.PROCESS.WISTATUS』
- 第 493 頁的 『BPC.BFM.PROCESS.WITRANSFER』
- 第 494 頁的 『BPC.BFM.PROCESS.ESCALATED』
- 第 494 頁的 『BPC.BFM.PROCESS.EVENT』
- 第 494 頁的 『BPC.BFM.PROCESS.PARTNER』
- 第 495 頁的 『BPC.BFM.ACTIVITY.BASE』
- 第 496 頁的 『BPC.BFM.ACTIVITY.STATUS』
- 第 496 頁的 『BPC.BFM.ACTIVITY.FAILURE』
- 第 496 頁的 『BPC.BFM.ACTIVITY.MESSAGE』
- 第 497 頁的 『BPC.BFM.ACTIVITY.CLAIM』
- 第 497 頁的 『BPC.BFM.ACTIVITY.WISTATUS』
- 第 497 頁的 『BPC.BFM.ACTIVITY.WITRANSFER』
- 第 497 頁的 『BPC.BFM.ACTIVITY.FOREACH』
- 第 498 頁的 『BPC.BFM.ACTIVITY.ESCALATED』
- 第 498 頁的 『BPC.BFM.ACTIVITY.EVENT』
- 第 498 頁的 『BPC.BFM.LINK.STATUS』
- 第 498 頁的 『BPC.BFM.VARIABLE.STATUS』

### BPC.BFM.BASE

BPC.BFM.BASE 繼承 WBIMonitoringEvent 的 XML 元素。

表 56. BPC.BFM.BASE 的 XML 元素

XML 元素	說明
<i>BPCEventCode</i>	可識別事件本質的「商業程序編排器」事件碼。
<i>processTemplateName</i>	程序範本的名稱。此名稱可與顯示名稱不同。

表 56. BPC.BFM.BASE 的 XML 元素 (繼續)

XML 元素	說明
<i>processTemplateValidFrom</i>	程序範本的有效起始時間屬性。
<i>eventLocalCounter</i>	本端計數器是用來探索相同交易中發生的兩個事件的次序。若為微流程實例，此計數器會重新建構所有產生的事件次序。若為長時間執行程序，本端計數器會指出現行導覽交易中的次序。

## BPC.BFM.PROCESS.BASE

BPC.BFM.PROCESS.STATUS 繼承 第 491 頁的『BPC.BFM.BASE』的 XML 元素。

表 57. BPC.BFM.PROCESS.BASE 的 XML 元素

XML 元素	說明
<i>processInstanceExecutionState</i>	下列格式的程序現行執行狀態： <state code>-<state name>。此屬性可以具有下列其中一個值：  1 - STATE_READY 2 - STATE_RUNNING 3 - STATE_FINISHED 4 - STATE_COMPENSATING 5 - STATE_FAILED 6 - STATE_TERMINATED 7 - STATE_COMPENSATED 8 - STATE_TERMINATING 9 - STATE_FAILING 10 - STATE_INDOUBT 11 - STATE_SUSPENDED 12 - STATE_COMPENSATION_FAILED

## BPC.BFM.PROCESS.STATUS

BPC.BFM.PROCESS.STATUS 繼承 『BPC.BFM.PROCESS.BASE』的 XML 元素。

表 58. BPC.BFM.PROCESS.STATUS 的 XML 元素

XML 元素	說明
<i>processTemplateId</i>	程序範本的 ID。
<i>processInstanceDescription</i>	程序實例的說明。
<i>principal</i>	與此事件相關聯的使用者名稱。

## BPC.BFM.PROCESS.START

BPC.BFM.PROCESS.START 繼承 第 492 頁的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 59. BPC.BFM.PROCESS.START 的 XML 元素

XML 元素	說明
<i>username</i>	若為 BPC.BFM.PROCESS.START，這是要求程序啟動或重新啟動的使用者名稱。

## BPC.BFM.PROCESS.FAILURE

BPC.BFM.PROCESS.FAILURE 繼承 第 492 頁的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 60. BPC.BFM.PROCESS.FAILURE 的 XML 元素

XML 元素	說明
<i>processFailedException</i>	導致程序失敗的異常訊息。

## BPC.BFM.PROCESS.CORREL

BPC.BFM.PROCESS.CORREL 繼承 第 492 頁的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 61. BPC.BFM.PROCESS.CORREL 的 XML 元素

XML 元素	說明
<i>correlationSet</i>	相互關係集實例，格式如下： <pre>&lt;?xml version="1.0"?&gt; &lt;correlationSet name="correlation set   name"&gt;   &lt;property name="property name"     value="property value"/&gt;* &lt;/correlationSet&gt;</pre>

## BPC.BFM.PROCESS.WISTATUS

BPC.BFM.PROCESS.WISTATUS 繼承 第 492 頁的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 62. BPC.BFM.PROCESS.WISTATUS 的 XML 元素

XML 元素	說明
<i>username</i>	若為 BPC.BFM.PROCESS.WISTATUS，這是已建立或刪除其工作項目的使用者清單。

## BPC.BFM.PROCESS.WITRANSFER

BPC.BFM.PROCESS.WITRANSFER 繼承 第 492 頁的『BPC.BFM.PROCESS.STATUS』的 XML 元素。



表 63. BPC.BFM.PROCESS.WITRANSFER 的 XML 元素

XML 元素	說明
<i>current</i>	工作項目的現行擁有者的使用者名稱。此位使用者的工作項目已傳送給他人。
<i>target</i>	工作項目的新擁有者的使用者名稱。

## BPC.BFM.PROCESS.ESCALATED

BPC.BFM.PROCESS.ESCALATED 繼承 第 492 頁的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 64. BPC.BFM.PROCESS.ESCALATED 的 XML 元素

XML 元素	說明
<i>escalationName</i>	呈報的名稱。
<i>operation</i>	若為 BPC.BFM.PROCESS.ESCALATED，是指與呈報行內呼叫作業的事件處理程式相關聯的作業。
<i>portTypeName</i>	與呈報行內呼叫作業的事件處理程式相關聯的作業的埠類型名稱。
<i>portTypeNamespace</i>	與呈報行內呼叫作業的事件處理程式相關聯的作業的埠類型名稱空間。

## BPC.BFM.PROCESS.EVENT

BPC.BFM.PROCESS.EVENT 繼承 第 492 頁的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 65. BPC.BFM.PROCESS.EVENT 的 XML 元素

XML 元素	說明
<i>message</i> 或 <i>message_BO-</i>	服務的輸入訊息或輸出訊息，為「字串」或商業物件 (BO) 表示法。此格式取決於 WebSphere Integration Developer 的事件監視器標籤上是否選取監視相容事件選項。  此屬性只用於 WebSphere Business Monitor 6.0.2 格式的事件。若為 WebSphere Business Monitor 6.1 格式事件，訊息的內容會寫入 <i>applicationData</i> 區段，它包含其名稱設為訊息名稱的一個內容元素。
<i>operation</i>	已接收事件的作業名稱。
<i>portTypeName</i>	與事件處理程式相關聯的作業的埠類型名稱。
<i>portTypeNamespace</i>	與事件處理程式相關聯的作業的埠類型名稱空間。

## BPC.BFM.PROCESS.PARTNER

BPC.BFM.PROCESS.PARTNER 繼承 第 492 頁的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 66. BPC.BFM.PROCESS.PARTNER 的 XML 元素

XML 元素	說明
<i>partnerLinkName</i>	夥伴鏈結的名稱。

## BPC.BFM.ACTIVITY.BASE

BPC.BFM.ACTIVITY.BASE 繼承 第 491 頁的『BPC.BFM.BASE』的 XML 元素。

表 67. BPC.BFM.ACTIVITY.BASE 的 XML 元素

XML 元素	說明
<i>activityKind</i>	<p>活動類型，例如，順序或呼叫。格式為：<code>&lt;kind code&gt;-&lt;kind name&gt;</code>。此屬性可以具有下列其中一個值：</p> <ul style="list-style-type: none"> <li>3 - KIND_EMPTY</li> <li>21 - KIND_INVOKE</li> <li>23 - KIND_RECEIVE</li> <li>24 - KIND_REPLY</li> <li>25 - KIND_THROW</li> <li>26 - KIND_TERMINATE</li> <li>27 - KIND_WAIT</li> <li>29 - KIND_COMPENSATE</li> <li>30 - KIND_SEQUENCE</li> <li>32 - KIND_SWITCH</li> <li>34 - KIND_WHILE</li> <li>36 - KIND_PICK</li> <li>38 - KIND_FLOW</li> <li>40 - KIND_SCOPE</li> <li>42 - KIND_SCRIPT</li> <li>43 - KIND_STAFF</li> <li>44 - KIND_ASSIGN</li> <li>45 - KIND_CUSTOM</li> <li>46 - KIND_RETHROW</li> <li>47 - KIND_FOR_EACH_SERIAL</li> <li>48 - KIND_FOR_EACH_PARALLEL</li> <li>1000 - SQLSnippet</li> <li>1001 - RetrieveSet</li> <li>1002 - InvokeInformationService</li> <li>1003 - AtomicSQLSnippetSequence</li> </ul>

表 67. BPC.BFM.ACTIVITY.BASE 的 XML 元素 (繼續)

XML 元素	說明
<i>state</i>	活動實例的現行狀態，格式為： <code>&lt;state code&gt;</code> - <code>&lt;state name&gt;</code> 。此屬性可以具有下列其中一個值：  1 - STATE_INACTIVE 2 - STATE_READY 3 - STATE_RUNNING 4 - STATE_SKIPPED 5 - STATE_FINISHED 6 - STATE_FAILED 7 - STATE_TERMINATED 8 - STATE_CLAIMED 9 - STATE_TERMINATING 10 - STATE_FAILING 11 - STATE_WAITING 12 - STATE_EXPIRED 13 - STATE_STOPPED
<i>bpelId</i>	BPEL 檔案中活動的 <code>wpc:id</code> 屬性。它對程序模型中的活動而言是唯一的。

## BPC.BFM.ACTIVITY.STATUS

BPC.BFM.ACTIVITY.STATUS 繼承 第 495 頁的『BPC.BFM.ACTIVITY.BASE』的 XML 元素。

表 68. BPC.BFM.ACTIVITY.STATUS 的 XML 元素

XML 元素	說明
<i>activityTemplateName</i>	活動範本的名稱。這可與顯示名稱不同。
<i>activityTemplateId</i>	活動範本的內部 ID。
<i>activityInstanceDescription</i>	活動實例的說明。
<i>principal</i>	要求活動的使用者名稱。

## BPC.BFM.ACTIVITY.FAILURE

BPC.BFM.ACTIVITY.FAILURE 繼承 『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 69. BPC.BFM.ACTIVITY.FAILURE 的 XML 元素

XML 元素	說明
<i>activityFailedException</i>	導致活動失敗的異常。

## BPC.BFM.ACTIVITY.MESSAGE

BPC.BFM.ACTIVITY.MESSAGE 繼承 『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 70. BPC.BFM.ACTIVITY.MESSAGE 的 XML 元素

XML 元素	說明
<i>message</i> 或 <i>message_BO</i>	服務的輸入或輸出訊息，為字串或商業物件 (BO) 表示法。此格式取決於 WebSphere Integration Developer 的事件監視器標籤上是否選取監視相容事件選項。  此屬性只用於 WebSphere Business Monitor 6.0.2 格式的事件。若為 WebSphere Business Monitor 6.1 格式事件，訊息的內容會寫入 <i>applicationData</i> 區段，它包含其名稱設為訊息名稱的一個內容元素。

## BPC.BFM.ACTIVITY.CLAIM

BPC.BFM.ACTIVITY.CLAIM 繼承 第 496 頁的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 71. BPC.BFM.ACTIVITY.CLAIM 的 XML 元素

XML 元素	說明
<i>username</i>	對於 BPC.BFM.ACTIVITY.CLAIM，這是要求其作業的使用者。

## BPC.BFM.ACTIVITY.WISTATUS

BPC.BFM.ACTIVITY.WISTATUS 繼承 第 496 頁的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 72. BPC.BFM.ACTIVITY.WISTATUS 的 XML 元素

XML 元素	說明
<i>username</i>	對於 BPC.BFM.ACTIVITY.WISTATUS，這是與工作項目相關聯的清單使用者。

## BPC.BFM.ACTIVITY.WITRANSFER

BPC.BFM.ACTIVITY.WITRANSFER 繼承 第 496 頁的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 73. BPC.BFM.ACTIVITY.WITRANSFER 的 XML 元素

XML 元素	說明
<i>current</i>	工作項目的現行擁有者的使用者名稱。此位使用者的工作項目已傳送給他人。
<i>target</i>	工作項目的新擁有者的使用者名稱。

## BPC.BFM.ACTIVITY.FOREACH

BPC.BFM.ACTIVITY.FOREACH 繼承 第 496 頁的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 74. BPC.BFM.ACTIVITY.FOREACH 的 XML 元素

XML 元素	說明
<i>parallelBranchesStarted</i>	已啟動的分支數目。

## BPC.BFM.ACTIVITY.ESCALATED

BPC.BFM.ACTIVITY.ESCALATED 繼承 第 496 頁的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 75. BPC.BFM.ACTIVITY.ESCALATED 的 XML 元素

XML 元素	說明
<i>escalationName</i>	呈報的名稱。
<i>operation</i>	若為 BPC.BFM.ACTIVITY.ESCALATED，是指與呈報行內呼叫作業的事件處理程式相關聯的作業。

## BPC.BFM.ACTIVITY.EVENT

BPC.BFM.ACTIVITY.EVENT 繼承 第 496 頁的『BPC.BFM.ACTIVITY.MESSAGE』的 XML 元素。

表 76. BPC.BFM.ACTIVITY.EVENT 的 XML 元素

XML 元素	說明
<i>operation</i>	已接收事件的作業名稱。

## BPC.BFM.LINK.STATUS

BPC.BFM.LINK.STATUS 繼承 第 491 頁的『BPC.BFM.BASE』的 XML 元素。

表 77. BPC.BFM.LINK.STATUS 的 XML 元素

XML 元素	說明
<i>elementName</i>	鏈結的名稱。
<i>description</i>	鏈結的說明。
<i>flowBpelId</i>	定義鏈結所在的流程活動 ID。

## BPC.BFM.VARIABLE.STATUS

BPC.BFM.VARIABLE.STATUS 繼承 第 491 頁的『BPC.BFM.BASE』的 XML 元素。

表 78. BPC.BFM.VARIABLE.STATUS 的 XML 元素

XML 元素	說明
<i>variableName</i>	變數的名稱。

表 78. BPC.BFM.VARIABLE.STATUS 的 XML 元素 (繼續)

XML 元素	說明
<i>variableData</i> 或 <i>variableData_BO</i>	<p>如果變數 <i>variableName</i> 尚未起始設定，則沒有 <i>variableData</i> 或 <i>VariableData_BO</i> 元素。變數的資料是以「字串」或商業物件 (BO) 表示。此格式取決於 WebSphere Integration Developer 的事件監視器標籤上是否選取監視相容事件選項。</p> <p>此屬性只用於 WebSphere Business Monitor 6.0.2 格式的事件。若為 WebSphere Business Monitor 6.1 格式事件，變數的內容會寫入 <i>applicationData</i> 區段，它包含其名稱設為變數名稱的一個內容元素。</p>
<i>bpellId</i>	變數的「商業程序編排器」ID。
<i>principal</i>	更新變數的使用者名稱。

## 商業程序事件

如果要求監視 WebSphere Integration Developer 中的商業程序元素，則會傳送商業程序事件。此處可以找到商業程序可產生的所有事件的清單。

當程序或活動的狀態變更時，會產生事件。商業程序可產生下列類型的事件：

- 第 500 頁的『程序事件』
- 第 501 頁的『活動事件』
- 第 503 頁的『活動範圍事件』
- 第 504 頁的『鏈結事件』
- 第 504 頁的『變數事件』

### XML 綱目定義 (XSD) 檔案

事件結構說明於「XML 綱目定義 (XSD)」檔案 BFMEvents.xsd 中。此檔案位於 *install\_root*\ProcessChoreographer\client 目錄中。

### 表格直欄的索引鍵

下表中的直欄包含：

**碼** 包含事件的號碼。若為 WebSphere Business Monitor 6.0.2 格式事件，此值會作為具有此名稱 *BPCEventCode* 的延伸資料元素寫入「共用基礎事件」中。若為 WebSphere Business Monitor 6.1 格式事件，此值會寫入「共用基礎事件」的 *xs:any* 插槽中。

#### 延伸名稱

*extensionName* 包含的字串值定義「共用基礎事件」中所包含的事件特定資訊。這也是提供事件其他相關資料的 XML 元素的名稱。如需延伸名稱的進一步資訊，請參閱第 491 頁的『商業程序事件的延伸名稱』。

**狀況** 是指商業程序事件的狀況名稱。如需狀況的詳細資料，請參閱第 504 頁的『商業程序事件的狀況』。

## 事件本質

EventNature 參數中商業程序元素的事件狀況的指標，如 WebSphere Integration Developer 中所示。

## 程序事件

下表說明所有程序事件。

碼	延伸名稱	狀況	事件本質	說明
21000	BPC.BFM.PROCESS.START	啓動	ENTRY	啓動的程序
21001	BPC.BFM.PROCESS.STATUS	報告	SUSPENDED	程序已暫停。若要暫停程序實例，請使用「商業程序編排器瀏覽器」。
21002	BPC.BFM.PROCESS.STATUS	報告	RESUMED	程序已回復。只可回復暫停的程序。若要回復程序實例，請使用「商業程序編排器瀏覽器」。
21004	BPC.BFM.PROCESS.STATUS	停止	EXIT	完成的程序
21005	BPC.BFM.PROCESS.STATUS	停止	TERMINATED	程序已終止。若要終止程序實例，請使用「商業程序編排器瀏覽器」。
21019	BPC.BFM.PROCESS.START	報告	RESTARTED	程序已重新啓動
21020	BPC.BFM.PROCESS.STATUS	損毀	DELETED	程序已刪除
42001	BPC.BFM.PROCESS.FAILURE	失敗	FAILED	程序失敗
42003	BPC.BFM.PROCESS.STATUS	報告	COMPENSATING	程序補償中。若要補償程序實例，請使用「商業程序編排器瀏覽器」。
42004	BPC.BFM.PROCESS.STATUS	停止	COMPENSATED	程序已補償
42009	BPC.BFM.PROCESS.STATUS	報告	TERMINATING	程序終止中
42010	BPC.BFM.PROCESS.STATUS	報告	FAILING	程序失敗中
42027	BPC.BFM.PROCESS.CORREL	報告	CORRELATION	相互關係集已起始設定。當程序實例的新相互關係集已起始設定時會產生。例如，當具有起始相互關係集的吸收活動接收到訊息時，就是這種情況。
42041	BPC.BFM.PROCESS.WISTATUS	報告	WI_DELETED	程序工作項目已刪除
42042	BPC.BFM.PROCESS.WISTATUS	報告	WI_CREATED	程序工作項目已建立



碼	延伸名稱	狀況	事件本質	說明
42046	BPC.BFM.PROCESS.STATUS	失敗	COMPFAILED	程序補償已失敗
42047	BPC.BFM.PROCESS.EVENT	報告	EV_RECEIVED	程序事件已接收。若要定義事件，請使用程序介面。當與程序相關聯的事件處理程式啟動時，會產生此事件。
42049	BPC.BFM.PROCESS.ESCALATED	報告	EV_ESCALATED	程序事件已呈報。呈報行內呼叫作業時會產生此事件，它定義在程序層次上，並與 onEvent 事件處理程式相關聯。
42056	BPC.BFM.PROCESS.WITRANSFER	報告	WI_TRANSFERRED	程序工作項目已轉移
42058	BPC.BFM.PROCESS.PARTNER	報告	PA_CHANGE	程序夥伴已變更。當新的端點參照指派給夥伴鏈結時，會產生此事件。

對於程序事件，下列事件相關性範圍 ID 具有下列內容：

- ECSCurrentID 提供程序實例的 ID。
- ECSParentID 在現行程序的程序實例開始事件之前提供 ECSCurrentID 的值。

## 活動事件

下表說明所有活動事件。

碼	延伸名稱	狀況	事件本質	說明
21006	BPC.BFM.ACTIVITY.MESSAGE	啟動	CREATED	活動已備妥。啟動人力作業活動時，會產生此事件。
21007	若為呼叫活動： BPC.BFM.ACTIVITY.MESSAGE。若為所有其他活動類型： BPC.BFM.ACTIVITY.STATUS	啟動	ENTRY	活動已啟動。對於「呼叫」活動，可使用商業物件承載量。
21011	若為呼叫、人力作業、接收及回覆活動： BPC.BFM.ACTIVITY.MESSAGE。若為挑選活動： BPC.BFM.ACTIVITY.EVENT。若為所有其他活動類型： BPC.BFM.ACTIVITY.STATUS	停止	EXIT	活動已完成。若為呼叫、人力作業、接收及回覆活動，可使用商業物件承載量。
21021	BPC.BFM.ACTIVITY.STATUS	報告	DEASSIGNED	要求已取消。取消人力作業活動的要求時，會產生此事件。

碼	延伸名稱	狀況	事件本質	說明
21022	BPC.BFM.ACTIVITY.CLAIM	報告	ASSIGNED	已要求活動。要求人力作業活動時，會產生此事件。
21027	BPC.BFM.ACTIVITY.STATUS	停止	TERMINATED	活動已終止。因為錯誤處理對於活動所指派至的範圍或程序產生影響，而可能終止長時間執行活動。
21080	BPC.BFM.ACTIVITY.FAILURE	已失敗	FAILED	活動已失敗
21081	BPC.BFM.ACTIVITY.STATUS	報告	EXPIRED	活動已過期。您可以定義呼叫及行內人力作業活動的期限。
42005	BPC.BFM.ACTIVITY.STATUS	報告	SKIPPED	活動已跳過。此事件只可套用於已定義結合行為的活動。如果結合行為評估為 false，則會跳過活動且會產生跳過事件。
42012	BPC.BFM.ACTIVITY.MESSAGE	報告	OUTPUTSET	活動輸出訊息已設定。可使用商業物件承載量。
42013	BPC.BFM.ACTIVITY.MESSAGE	報告	FAULTSET	活動錯誤訊息已設定。可使用商業物件承載量。
42015	BPC.BFM.ACTIVITY.STATUS	停止	STOPPED	活動已停止
42031	BPC.BFM.ACTIVITY.STATUS	報告	FRETRIED	活動已強制重試。若要強制活動重試，請使用「商業程序編排器瀏覽器」。
42032	BPC.BFM.ACTIVITY.STATUS	停止	FCOMPLETED	活動已強制完成。若要強制活動完成，請使用「商業程序編排器瀏覽器」。
42036	BPC.BFM.ACTIVITY.MESSAGE	報告	EXIT	活動已接收訊息
42037	BPC.BFM.ACTIVITY.STATUS	報告	CONDTRUE	迴圈條件為 true
42038	BPC.BFM.ACTIVITY.STATUS	報告	CONDFALSE	迴圈條件為 false
42039	BPC.BFM.ACTIVITY.WISTATUS	報告	WI_DELETED	工作項目已刪除。此事件只可套用於挑選、行內人力作業及接收事件。

碼	延伸名稱	狀況	事件本質	說明
42040	BPC.BFM.ACTIVITY.WISTATUS	報告	WI_CREATED	工作項目已建立。此事件只可套用於挑選、行內人力作業及接收事件。
42050	BPC.BFM.ACTIVITY.ESCALATED	報告	ESCALATED	活動已呈報。此事件只可套用於挑選、行內人力作業及接收事件。
42054	BPC.BFM.ACTIVITY.WISTATUS	報告	WI_REFRESHED	活動工作項目已重新整理。此事件只可套用於挑選、行內人力作業及接收事件。
42055	BPC.BFM.ACTIVITY.WITRANSFER	報告	WI_TRANSFERRED	工作項目已轉移。此事件只可套用於挑選、行內人力作業及接收事件。
42057	BPC.BFM.ACTIVITY.FOREACH	報告	BRANCHES_STARTED	對於每項 - 活動分支已啟動

對於活動事件，下列事件相關性範圍 ID 具有下列內容：

- *ECSCurrentID* 提供活動的 ID。
- *ECSParentID* 提供包含程序的 ID。

### 活動範圍事件

下表說明所有活動範圍事件。

碼	延伸名稱	狀況	事件本質	說明
42020	BPC.BFM.ACTIVITY.STATUS	啟動	ENTRY	範圍已啟動
42021	BPC.BFM.ACTIVITY.STATUS	報告	SKIPPED	範圍已跳過
42022	BPC.BFM.ACTIVITY.FAILURE	失敗	FAILED	範圍已失敗
42023	BPC.BFM.ACTIVITY.STATUS	報告	FAILING	範圍終止中
42024	BPC.BFM.ACTIVITY.STATUS	停止	TERMINATED	範圍已終止
42026	BPC.BFM.ACTIVITY.STATUS	停止	EXIT	範圍已完成
42043	BPC.BFM.ACTIVITY.STATUS	報告	COMPENSATING	範圍補償中
42044	BPC.BFM.ACTIVITY.STATUS	停止	COMPENSATED	範圍已補償
42045	BPC.BFM.ACTIVITY.STATUS	失敗	COMPFAILED	範圍補償已失敗
42048	BPC.BFM.ACTIVITY.EVENT	報告	EV_RECEIVED	活動事件已接收
42051	BPC.BFM.ACTIVITY.ESCALATED	報告	EV_ESCALATED	範圍事件已呈報

活動範圍事件是活動事件類型，其語法如 BPC.BFM.ACTIVITY.STATUS 所述。

對於活動範圍事件，下列事件相關性範圍 ID 具有下列內容：

- *ECSCurrentID* 提供範圍的 ID。

- ECSParentID 提供包含程序的 ID。

## 鏈結事件

下表說明所有鏈結事件。

碼	延伸名稱	狀況	事件本質	說明
21034	BPC.BFM.LINK.STATUS	報告	CONDTRUE	鏈結評估為 true
42000	BPC.BFM.LINK.STATUS	報告	CONDFALSE	鏈結評估為 false

對於鏈結事件，下列事件相關性範圍 ID 具有下列內容：

- ECSCurrentID 提供鏈結的來源活動 ID。
- ECSParentID 提供包含程序的 ID。

## 變數事件

下表說明變數事件。

碼	延伸名稱	狀況	事件本質	說明
21090	BPC.BFM.VARIABLE.STATUS	報告	CHANGED	變數更新。可使用商業物件承載量。

對於變數事件，下列事件相關性範圍 ID 具有下列內容：

- ECSCurrentID 提供包含程序的 ID。
- ECSParentID 是現行程序的程序實例開始事件之前的 ECSCurrentID。

---

## 商業程序事件的狀況

在不同狀況下可產生商業程序事件。這些狀況的資料在狀況元素中說明。

商業程序事件可包含下列其中一個狀況元素。

狀況名稱	共用基礎事件的内容	
啓動	categoryName 會設為 StartSituation。	
	situationType	
	類型	StartSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	START_COMPLETED
停止	categoryName 會設為 StopSituation。	
	situationType	
	類型	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	STOP_COMPLETED

狀況名稱	共用基礎事件的内容	
損毀	categoryName 會設為 DestroySituation。	
	situationType	
	類型	DestroySituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
失敗	categoryName 會設為 StopSituation。	
	situationType	
	類型	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	UNSUCCESSFUL
	situationQualifier	STOP_COMPLETED
報告	categoryName 會設為 ReportSituation。	
	situationType	
	類型	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS



---

## 第 20 章 監視人力作業事件

為代表人力作業而產生的事件，由與狀況無關的資料，以及人力作業事件特定的資料組成。已說明人力作業事件特定的屬性及元素。

人力作業事件可以擁有以下種類的事件內容。

---

### 人力作業的特定事件資料

事件是代表作業及呈報而建立的。會說明其中每個事件類型的物件特定內容。

若為「商業程序編排器」6.1 版，會發生兩種事件格式：

#### WebSphere Business Monitor 6.0.2 格式

若在 WebSphere Integration Developer 6.0.2 內有模型化作業，或在 WebSphere Integration Developer 6.1 內已啟用 WebSphere Business Monitor 6.0.2 格式模式，則會發生 WebSphere Business Monitor 6.0.2 格式事件。如果未另行指定，則這些事件的物件特定內容會撰寫為字串類型的 *extendedDataElement* XML 元素。

#### WebSphere Business Monitor 6.1 格式

若在 WebSphere Integration Developer 6.1 內有模型化作業，且已啟用 WebSphere Business Monitor 6.1 格式模式，則會發生 WebSphere Business Monitor 6.1 格式事件。這些事件的物件特定內容會撰寫為「共用基礎事件」的 *eventPointData* 資料夾內的 *xs:any* 插槽中的 XML 元素。XML 的結構定義於「XML 綱目定義 (XSD)」檔案 *HTMEvents.xsd* 中。此檔案位於 *install\_root\ProcessChoreographer\client* 目錄中。

### 作業

作業事件具有下列物件特定的事件內容。

屬性	說明
<i>taskTemplateName</i>	實例衍生來源的作業範本名稱。
<i>taskTemplateValidFrom</i>	範本的有效起始日期。
<i>taskTemplateId</i>	實例衍生來源的作業範本 ID。
<i>taskInstanceDescription</i>	預設語言環境中作業實例的說明。
<i>PayloadType</i>	承載量類型。字串的值可以是下列其中一個： <i>none</i> 、 <i>digest</i> 或 <i>full</i> 。此值取決於 WebSphere Integration Developer 中的設定。如果沒有可用的明確監視定義，則預設承載量類型為 <i>full</i> 。

### 呈報

呈報具有下列物件特定的事件內容：

屬性	說明
<i>taskTemplateName</i>	實例衍生來源的作業範本名稱。



屬性	說明
taskTemplateValidFrom	範本的有效起始日期。
taskTemplateId	實例衍生來源的作業範本 ID。
escalationName	呈報的名稱。
escalationInstanceDescription	選用項目：呈報實例的說明。
PayloadType	承載量類型。字串的值可以是下列其中一個：none、digest 或 full。此值取決於 WebSphere Integration Developer 中的設定。如果沒有可用的明確監視定義，則預設承載量類型為 full。

## 人力作業事件的延伸名稱

此延伸名稱指出人力作業事件的承載量。您可在這裡找到人力作業事件的所有延伸名稱及其對應承載量的清單。

此延伸名稱包含用作「共用基礎事件」的 *extensionName* 屬性值的字串值。這也是提供事件其他相關資料的 XML 元素的名稱。事件元素的名稱為大寫，例如 BPC.HTM.BASE，XML 元素的名稱為大小寫混合格式，例如，*HTMEventCode*。除另行指出外，所有資料元素都是 string 類型。

下列延伸名稱可用於人力作業事件：

- 『BPC.HTM.BASE』
- 第 509 頁的 『BPC.HTM.TASK.BASE』
- 第 509 頁的 『BPC.HTM.TASK.STATUS』
- 第 509 頁的 『BPC.HTM.TASK.FOLLOW』
- 第 509 頁的 『BPC.HTM.TASK.MESSAGE』
- 第 510 頁的 『BPC.HTM.TASK.INTERACT』
- 第 510 頁的 『BPC.HTM.TASK.FAILURE』
- 第 510 頁的 『BPC.HTM.TASK.WISTATUS』
- 第 510 頁的 『BPC.HTM.TASK.WITRANSFER』
- 第 510 頁的 『BPC.HTM.ESCALATION.STATUS』
- 第 511 頁的 『BPC.HTM.ESCALATION.WISTATUS』
- 第 511 頁的 『BPC.HTM.ESCALATION.WITRANSFER』

### BPC.HTM.BASE

BPC.HTM.BASE 繼承 WBIMonitoringEvent 的 XML 元素。

表 79. BPC.HTM.BASE 的 XML 元素

XML 元素	說明
<i>HTMEventCode</i>	可識別事件類型號碼的「商業程序編排器」事件碼。可能的事件碼在下表中列出。

## BPC.HTM.TASK.BASE

BPC.HTM.TASK.BASE 繼承 第 508 頁的『BPC.HTM.BASE』的 XML 元素。

表 80. BPC.HTM.TASK.BASE 的 XML 元素

XML 元素	說明
<i>taskTemplateId</i>	範本的 ID。
<i>taskTemplateName</i>	作業範本的名稱。這可與顯示名稱不同。
<i>taskTemplateValidFrom</i>	作業範本的有效起始日期和時間。

## BPC.HTM.TASK.STATUS

BPC.HTM.TASK.STATUS 繼承 『BPC.HTM.TASK.BASE』的 XML 元素。

表 81. BPC.HTM.TASK.STATUS 的 XML 元素

XML 元素	說明
<i>taskInstanceDescription</i>	作業的說明。

## BPC.HTM.TASK.FOLLOW

BPC.HTM.TASK.FOLLOW 繼承 『BPC.HTM.TASK.BASE』的 XML 元素。

表 82. BPC.HTM.TASK.FOLLOW 的 XML 元素

XML 元素	說明
<i>followTaskId</i>	作為後續作業啟動的作業 ID。

## BPC.HTM.TASK.MESSAGE

BPC.HTM.TASK.MESSAGE 繼承 『BPC.HTM.TASK.STATUS』的 XML 元素。

表 83. BPC.HTM.TASK.FOLLOW 的 XML 元素

XML 元素	說明
<i>message</i> 或 <i>message_BO</i>	<p>包含輸入或輸出訊息的「字串」或商業物件表示法。此格式取決於 WebSphere Integration Developer 的事件監視器標籤上是否選取監視相容事件選項。</p> <p>此屬性只用於 WebSphere Business Monitor 6.0.2 格式的事件。若為 WebSphere Business Monitor 6.1 格式事件，訊息的內容會寫入 <i>applicationData</i> 區段，它包含其名稱設為訊息名稱的一個內容元素。</p>

## BPC.HTM.TASK.INTERACT

BPC.HTM.TASK.INTERACT 繼承 第 509 頁的『BPC.HTM.TASK.STATUS』的 XML 元素。

表 84. BPC.HTM.TASK.INTERACT 的 XML 元素

XML 元素	說明
<i>username</i>	若為 BPC.HTM.TASK.INTERACT，這是與作業相關聯的使用者名稱。

## BPC.HTM.TASK.FAILURE

BPC.HTM.TASK.FAILURE 繼承 第 509 頁的『BPC.HTM.TASK.STATUS』的 XML 元素。

表 85. BPC.HTM.TASK.FAILURE 的 XML 元素

XML 元素	說明
<i>taskFailedException</i>	包含以分號 (;) 區隔的 <i>faultNameSpace</i> 及 <i>faultName</i> 的字串。

## BPC.HTM.TASK.WISTATUS

BPC.HTM.TASK.WISTATUS 繼承 第 509 頁的『BPC.HTM.TASK.STATUS』的 XML 元素。

表 86. BPC.HTM.TASK.WISTATUS 的 XML 元素

XML 元素	說明
<i>username</i>	若為 BPC.BPC.TASK.WISTATUS，這是已建立或刪除其工作項目的使用者清單。

## BPC.HTM.TASK.WITRANSFER

BPC.HTM.TASK.WITRANSFER 繼承 第 509 頁的『BPC.HTM.TASK.STATUS』的 XML 元素。

表 87. BPC.HTM.TASK.WITRANSFER 的 XML 元素

XML 元素	說明
<i>current</i>	若為 BPC.HTM.TASK.WITRANSFER，這是現行使用者的名稱。此位使用者的工作項目已傳送給他人。
<i>target</i>	若為 BPC.HTM.TASK.WITRANSFER，這是工作項目接收者的使用者名稱。

## BPC.HTM.ESCALATION.STATUS

BPC.HTM.ESCALATION.STATUS 繼承 第 509 頁的『BPC.HTM.TASK.BASE』的 XML 元素。

表 88. *BPC.HTM.ESCALATION.STATUS* 的 XML 元素

XML 元素	說明
<i>escalationName</i>	呈報的名稱。
<i>escalationInstanceDescription</i>	呈報的說明。

## BPC.HTM.ESCALATION.WISTATUS

*BPC.HTM.ESCALATION.WISTATUS* 繼承 第 510 頁的『*BPC.HTM.ESCALATION.STATUS*』的 XML 元素。

表 89. *BPC.HTM.ESCALATION.WISTATUS* 的 XML 元素

XML 元素	說明
<i>username</i>	若為 <i>BPC.HTM.ESCALATION.WISTATUS</i> ，這是其工作項目已呈報的使用者清單。

## BPC.HTM.ESCALATION.WITRANSFER

*BPC.HTM.ESCALATION.WITRANSFER* 繼承 第 510 頁的『*BPC.HTM.ESCALATION.STATUS*』的 XML 元素。

表 90. *BPC.HTM.ESCALATION.WITRANSFER* 的 XML 元素

XML 元素	說明
<i>current</i>	若為 <i>BPC.HTM.ESCALATION.WITRANSFER</i> ，這是現行使用者的名稱。此位使用者的工作項目已傳送給他人。
<i>target</i>	若為 <i>BPC.HTM.ESCALATION.WITRANSFER</i> ，這是工作項目接收者的使用者名稱。

---

## 人力作業事件

如果要求監視 WebSphere Integration Developer 中的作業元素，則會傳送人力作業事件。此處可以找到人力作業可產生的所有事件清單。

當作業的狀態變更時，會產生事件。人力作業可產生下列類型的事件：

- 第 512 頁的『作業事件』
- 第 513 頁的『呈報事件』

**註：**如果作業模型中商業相關性旗標設為 `true`，則只會產生特定作業的事件。

行內作業的事件會作為活動事件產生。如需這些事件的清單，請參閱第 499 頁的『商業程序事件』。

### XML 綱目定義 (XSD) 檔案

事件結構說明於「XML 綱目定義 (XSD)」檔案 `HTMEvents.xsd` 中。此檔案位於 `install_root\ProcessChoreographer\client` 目錄中。

## 表格直欄的索引鍵

下表中的直欄包含：

**碼** 包含事件的號碼。若為 WebSphere Business Monitor 6.0.2 格式事件，此值會作為具有此名稱 *HTMEventCode* 的延伸資料元素寫入「共用基礎事件」中。若為 WebSphere Business Monitor 6.1 格式事件，此值會寫入「共用基礎事件」的 *xs:any* 插槽中。

### 延伸名稱

包含作為「共用基礎事件」的 *extensionName* 屬性值使用的字串值。

如果 WebSphere Business Integration Modeler 用於建立基礎作業模型，則其承載量中包含訊息資料的事件延伸名稱，可由其後跟隨其他字元的雜湊字元 (#) 進行延伸。這些其他字元用於區別帶有不同訊息物件的「共用基礎事件」。發出訊息資料的事件也包含其他巢狀 *extendedDataElements*，以報告資料物件的內容。如需相關資訊，請參閱 WebSphere Business Integration Modeler 的文件。

**狀況** 是指人力作業事件的狀況名稱。如需狀況的詳細資料，請參閱第 513 頁的『人力作業事件中的狀況』。

### 事件本質

*EventNature* 參數中商業程序元素的事件狀況的指標，如 WebSphere Integration Developer 中所示。

## 作業事件

下表說明所有作業事件。

碼	延伸名稱	狀況	事件本質	說明
51001	BPC.HTM.TASK.INTERACT	報告	CREATED	作業已建立
51002	BPC.HTM.TASK.STATUS	損毀	DELETED	作業已刪除
51003	BPC.HTM.TASK.STATUS	啟動	ENTRY	作業已啟動
51004	BPC.HTM.TASK.STATUS	停止	EXIT	作業已完成
51005	BPC.HTM.TASK.STATUS	報告	DEASSIGNED	已取消要求
51006	BPC.HTM.TASK.INTERACT	報告	ASSIGNED	作業已要求
51007	BPC.HTM.TASK.STATUS	停止	TERMINATED	作業已終止
51008	BPC.HTM.TASK.FAILURE	失敗	FAILED	作業已失敗
51009	BPC.HTM.TASK.STATUS	報告	EXPIRED	作業已過期
51010	BPC.HTM.TASK.STATUS	報告	WAITFORSUBTASK	等待子作業
51011	BPC.HTM.TASK.STATUS	停止	SUBTASKCOMPLETED	已完成子作業
51012	BPC.HTM.TASK.STATUS	報告	RESTARTED	作業已重新啟動
51013	BPC.HTM.TASK.STATUS	報告	SUSPENDED	作業已暫停
51014	BPC.HTM.TASK.STATUS	報告	RESUMED	作業已回復
51015	BPC.HTM.TASK.FOLLOW	報告	COMPLETEDFOLLOW	作業已完成且後續作業已啟動
51101	BPC.HTM.TASK.STATUS	報告	UPDATED	作業內容已更新
51103	BPC.HTM.TASK.MESSAGE	報告	OUTPUTSET	輸出訊息已更新。可使用商業物件承載量。

碼	延伸名稱	狀況	事件本質	說明
51104	BPC.HTM.TASK. MESSAGE	報告	FAULTSET	錯誤訊息已更新。可使用商業物件承載量。
51201	BPC.HTM.TASK. WISTATUS	損毀	WI_DELETED	工作項目已刪除
51202	BPC.HTM.TASK. WISTATUS	報告	WI_CREATED	工作項目已建立
51204	BPC.HTM.TASK. WITRANSFER	報告	WI_TRANSFERRED	工作項目已轉移
51205	BPC.HTM.TASK. WISTATUS	報告	WI_REFRESHED	工作項目已重新整理

對於作業事件，下列事件相關性範圍 ID 具有下列內容：

- ESCcurrentID 提供作業實例的 ID。
- ECSParentID 是作業實例事件之前的 ESCCurrentID。

### 呈報事件

下表說明所有作業呈報事件。

碼	延伸名稱	狀況	事件本質	說明
53001	BPC.HTM.ESCALATION. STATUS	報告	ENTRY	呈報已激發
53201	BPC.HTM.ESCALATION. WISTATUS	損毀	WI_DELETED	工作項目已刪除
53202	BPC.HTM.ESCALATION. WISTATUS	報告	WI_CREATED	工作項目已建立
53204	BPC.HTM.ESCALATION. WITRANSFER	報告	WI_TRANSFERRED	呈報已轉移
53205	BPC.HTM.ESCALATION. WISTATUS	報告	WI_REFRESHED	工作項目已重新整理

對於作業事件，下列事件相關性範圍 ID 具有下列內容：

- ESCcurrentID 提供呈報的 ID。
- ECSParentID 提供關聯作業實例的 ID。

## 人力作業事件中的狀況

在不同狀況下可產生人力作業事件。這些狀況的資料在狀況元素中說明。

人力作業事件可包含下列其中一個狀況元素。

狀況名稱	共用基礎事件的内容	
啓動	categoryName 會設為 StartSituation。	
	situationType	
	類型	StartSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	START_COMPLETED

狀況名稱	共用基礎事件的内容	
停止	categoryName 會設為 StopSituation。	
	situationType	
	類型	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	STOP_COMPLETED
損毀	categoryName 會設為 DestroySituation。	
	situationType	
	類型	DestroySituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
失敗	categoryName 會設為 StopSituation。	
	situationType	
	類型	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	UNSUCCESSFUL
	situationQualifier	STOP_COMPLETED
報告	categoryName 會設為 ReportSituation。	
	situationType	
	類型	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS



---

## 第 6 篇 調整



---

## 第 21 章 調整商業程序

使用這項作業來改善商業程序的效能。

順利執行商業程序之後，可以執行這項作業來改善效能。

1. 定義如何測量基準線效能，以及想要最佳化的測量。

例如，針對部分商業應用程式，是偏好減少一般使用者在尖峰負荷情況下的回應時間。而針對其他應用程式，則系統可以處理交易的速率其重要性可能會高於每個交易的實際持續時間。

2. 製作基準線測量。

在適用來調整應用程式的負載、時間及星期幾狀況下，製作基準線測量。一般而言，最重要的基準線測量是產量及回應時間。產量值是在達到特定瓶頸容量之後進行測量，例如 100% 的 CPU 負載、最大磁碟 I/O 或 100% 的網路 I/O。在低伺服器使用率期間測量的單一程序實例的回應時間值最為可靠。

3. 調整應用程式。

應用程式可以包含多個程序。因為微流程的執行效果比長時間執行程序還要好。如果不需要持續性，而且某個交易中的單一執行緒可以處理功能，則優先選用建模微流程。也請考量將長時間執行程序的分支分隔成微流程。此外，同步服務呼叫通常比非同步服務呼叫還要快。因此，基於效能原因，會優先選用同步服務呼叫，雖然這不是長時間執行程序中的預設行為。

在長時間執行程序中，您可以變更交易界限。在大部分情況下，可以減少交易界限數來改善效能。不過，只有透過效能測試，才能找出交易界限的最佳數目。也請考量在您的程序而非序列化活動中使用平行執行路徑，並在資料的序列化及解除序列化也很昂貴時，將透過您的程序流動的資料大小及複雜性降至最低。也請將針對記載而產生的事件數目降至最低。

4. 調整程序。

根據應用程式是使用長時間執行的程序還是微流程而定，執行下列其中一個步驟：

- 若要調整長時間執行的程序，請執行第 518 頁的『調整長時間執行的程序』中說明的步驟。這些程序是要長時間執行的，但是事件或人機互動可以中斷這些程序。因此，它們的效能是根據「商業程序編排器」資料庫及傳訊服務的效能而定。
  - 若要調整微流程，請執行第 528 頁的『調整微流程』中說明的步驟。這些程序只需短時間執行。它們只是使用資料庫來記載審核日誌(如果啓用的話)，以及擷取範本資訊。而不會使用傳訊支援來儲存持續資料。這些程序不包含人機互動。
5. 檢閱現行配置，以找出可以去除的效能瓶頸。

需要考慮的可能性包括：

- 安裝更多的處理器、更多的記憶體以及更快速的磁碟。
- 將資料庫日誌儲存在與資料不同的實體磁碟上，以及將資料分佈到數個磁碟上。
- 使用 DB2，而不是使用 Cloudscape，以取得最佳效能。

6. 在基準線測量的類似負載條件下，重複測量基準性能測試。

保留應用程式效能測量的永久記錄，以客觀地測量效能的任何未來變更。

商業程序已配置成明顯地加速執行。

---

## 調整長時間執行的程序

使用這項作業來改善長時間執行的商業程序的效能。

長時間執行的程序可以包括使用者互動、非同步呼叫、多個接收、挑選及事件處理程式這類項目；它們使用資料庫及傳訊子系統來儲存持續狀態。下列主題說明如何改善長時間執行程序的效能。

### 平衡硬體資源

您可以透過平衡硬體資源來改善長時間執行的商業程序的效能。

開始調整系統之前，請驗證所用的電腦已達到良好的平衡，亦即可用資源（CPU、記憶體及 I/O）的比例正確。一部電腦如果具有一個以上極快速的 CPU，但是記憶體很少或 I/O 效能不佳，則很難調整。如果是可岔斷程序，則多個快速磁碟機所提供的良好 I/O 效能與足夠的處理能力和足夠的記憶體一樣重要。

若為正式作業系統，建議使用個別機器來執行 DBMS 及應用程式伺服器，讓資料庫與應用程式伺服器分開。若為高負載或高可用性配置，請考量在數台機器上使用 WebSphere 叢集執行商業程序，以及對資料庫使用個別機器。

1. 在資料庫機器上，確定配置足夠的磁碟。
2. 配置足夠的記憶體。

配置的記憶體數量是根據平台而定：

- 若為具有 4 GB 實體記憶體及本端資料庫管理系統的 32 位元 Windows 系統，請使用下列記憶體配置：
  - 512 MB 用於 Windows 系統
  - 768 MB 用於 WebSphere Application Server
  - 1.5 GB 用於資料庫 (如果您是使用 DB2 的話)。如果您是使用 Oracle，不超過 1 GB 用於「系統廣域區域 (SGA)」，以及 500 MB 用於「程式廣域區域 (PGA)」。
- 若為具有 8 GB 實體記憶體及本端資料庫管理系統的 64 位元 AIX 系統，請使用下列記憶體配置：
  - 512 MB 用於 AIX 系統
  - 1024 MB 用於 WebSphere Application Server
  - 5 GB 用於資料庫。請將 4 GB 配置給程序資料庫，而將 1 GB 配置給傳訊資料庫。

**提示:** 若要協助確保最佳效能，請不要將所有記憶體配置給資料庫，因為檔案快取 (舉例來說) 也會耗用記憶體。因為可用的記憶體不足，所以請避免必須將資料交換至磁碟的狀況。

- 若為 i5/OS 系統，請使用查看系統狀態 (WRKSYSSTS) 指令，協助您防止系統分頁記憶體。如果發生大量分頁錯誤，請採取下列一個以上的動作：

- a. 增加 WebSphere Process Server 或 Enterprise Service Bus 伺服器子系統記憶體儲存區可用的記憶體
  - b. 將 WebSphere Process Server 或 Enterprise Service Bus 伺服器移至另一個記憶體儲存區
  - c. 從 WebSphere Process Server 或 Enterprise Service Bus 伺服器子系統記憶體儲存區中移除工作
- 細部調整應用程式伺服器的資料堆大小。

**註:** 如果您的應用程式伺服器在 i5/OS 系統上執行，您無法調整資料堆大小。

3. 觀察網路使用率。應用程式的效能也會根據訊息可以在伺服器與資料庫伺服器之間傳遞的速度而定。如果可能，請減少網路中的「延遲」。
4. 將工作量移至其他伺服器。

考量可以移至其他伺服器的應用程式或子系統。

電腦硬體現已達到良好的平衡。

#### 相關工作

第 95 頁的『規劃 BPEDB 資料庫』

規劃「商業程序編排器」的資料庫。

第 523 頁的『調整應用程式伺服器』

使用這項作業來調整應用程式伺服器。

## 指定起始 DB2 資料庫設定

使用這項作業來指定起始 DB2 資料庫設定。請注意，這項資訊只當作範例。

**警告:** 下列資訊與「商業程序編排器」資料庫有關。如需調整 WebSphere 預設傳訊資料庫的相關資訊，請參閱 WebSphere Application Server Network Deployment 資訊中心內的調整及解決傳訊引擎資料儲存庫的問題。

若要資料庫運作正常，請指定起始資料庫設定。此外，也會使用兩個有不同分段大小的個別邏輯磁碟。也會對每一個實例使用一個資料庫，所以當傳訊引擎使用資料庫作為資料儲存庫時，它們可以使用「商業程序編排器」資料庫進行傳訊，或對傳訊資料庫使用個別的資料庫機器。

**註:** 這個資訊不適用於 DB2 UDB for iSeries，因為這些設定是這個資料庫類型的內建設定。

1. 區隔日誌檔與資料檔。

將資料庫日誌檔放在與資料不同的磁碟機上，可以改善效能，但前提是要有足夠的可用磁碟機空間。

例如，如果在 Windows 系統上使用 DB2，則可以輸入下列指令，將 BPEDB 資料庫的日誌檔位置變更為 F:\db2logs 目錄：

```
db2 UPDATE DB CFG FOR BPEDB USING NEWLOGPATH F:\db2logs
```

2. 建立表格空間。

建立資料庫之後，請明確建立表格空間。WebSphere Application Server 安裝的 ProcessChoreographer 子目錄中的「商業程序編排器」提供範例 Script，可建立表

格空間。請自訂這些 Script 以符合特定範例情況的需求。建立表格空間的目標是要將輸入及輸出作業分散到可供 DB2 使用的磁碟機上，而且磁碟機數目越多越好。這些 Script 預設會建立下列表格空間：

#### **AUDITLOG**

包含程序及作業的審核追蹤表格。根據您使用審核功能的情形，對此表格空間中的表格存取可能十分頻繁。如果關閉審核功能，則不會存取此表格空間中的表格。

**COMP** 包含「商業程序編排器第 5 版」的商業程序補償表格。根據可補償程序及活動的百分比，此表格空間中的表格可能需要高磁碟頻寬。如果商業程序內未使用補償作業，則不會使用此表格空間中的表格。

#### **INSTANCE**

保留程序實例及作業表格。不論所執行的長時間執行程序的種類為何，一律都會密集使用此項。如果可能，請將此表格空間分散到數個磁碟機上。

#### **SCHEDTS**

包含 WebSphere 排程元件所使用的表格。因為排程器中使用的快取機制，所以排程器表格空間的表格存取數通常很低。

#### **STAFFQRY**

包含用來暫時儲存人員查詢結果的表格，而這些查詢結果是取自人員登錄(如「輕量型目錄存取通訊協定 (LDAP)」。當商業程序包含許多人員活動時，會頻繁地存取此表格空間中的表格。

#### **TEMPLATE**

包含的表格可以儲存程序及作業的範本資訊。在部署應用程式期間，會移入表格。而在執行時期，存取率則十分低。在部署期間，不會更新資料，且只會插入新的資料。

#### **WORKITEM**

保留工作項目處理所需的表格。工作項目是用來與人力作業互動。根據商業程序中的人力作業數目，此表格空間中的表格存取率差異極大，範圍從低存取率到極高的存取率都有可能。因為也會產生工作項目來支援管理長時間執行程序，所以即使未使用明確的人力作業，存取率也不會是零。

若要建立高效能的資料庫，請執行下列動作：

a. 建立資料庫。

在 Windows 上，可以指定目的地磁碟機。此指令會在目的地磁碟機上與伺服器上預設 DB2 實例同名的目錄中建立資料庫。例如，如果要在磁碟機 D: 上建立資料庫，而本端預設實例是 DB2，則資料庫資料會放至 D:\DB2 中。因此，若要在「商業程序編排器」的 D: 目錄中建立 DB2 資料庫，請輸入下列指令：

```
CREATE DATABASE BPEDB ON D: USING CODESET UTF-8 TERRITORY en-us;
```

在 UNIX 及 Linux 上，輸入下列指令：

```
CREATE DATABASE BPEDB ON /wasdbfs USING CODESET UTF-8 TERRITORY en-us;
```

其中 /wasdbfs 指定目錄。

b. 在所需磁碟上建立表格空間。

例如，下列 Script 是根據 createTablespaceDb2.ddl 檔案而來，而該檔案是位  
在所安裝的 WebSphere Application Server ProcessChoreographer 子目錄中。它  
會在 Windows 系統上使用單一高效能磁碟驅動器，來建立表格空間。

```
-- Scriptfile to create tablespaces for DB2 UDB
-- Replace occurrence of @location@ in this file with the location
-- where you want the tablespace containers to be stored, then run:
-- db2 connect to BPEDB
-- db2 -tf createTablespaceDb2.ddl
```

```
CREATE TABLESPACE TEMPLATE MANAGED BY SYSTEM USING( 'D:/BPE/TEMPLATE' );
CREATE TABLESPACE STAFFQRY MANAGED BY SYSTEM USING( 'D:/BPE/STAFFQRY' );
CREATE TABLESPACE AUDITLOG MANAGED BY SYSTEM USING( 'D:/BPE/AUDITLOG' );
CREATE TABLESPACE COMP MANAGED BY SYSTEM USING( 'D:/BPE/COMP' );
CREATE TABLESPACE INSTANCE MANAGED BY SYSTEM USING( 'D:/BPE/INSTANCE', 'D:/BPE/INSTANCE' );
CREATE TABLESPACE WORKITEM MANAGED BY SYSTEM USING( 'D:/BPE/WORKITEM' );
CREATE TABLESPACE SCHEDTS MANAGED BY SYSTEM USING( 'D:/BPE/SCHEDTS' );
```

c. 建立表格。

執行針對個別資料庫提供的 Script，建立「商業程序編排器」表格。例如，若為  
DB2，將使用 ProcessChoreographer 目錄中的 createSchemaDb2.ddl 檔案。

3. 調整資料庫。

使用產能規劃工具來起始設定資料庫。

如果使用 DB2，請從「商業程序編排器」資料庫的闡現功能表中選取 **DB2 配置顧問**，從「DB2 控制中心」中啟動 DB2 配置顧問。請執行下列動作：

a. 將記憶體配置給 DB2。

在**伺服器**中，只將它實際可用的最大記憶體配置給 DB2，而不交換。

b. 指定工作量類型。

在**工作量**中，選取**混合** (查詢及交易)。

c. 在**交易**中，指定交易長度以及每分鐘處理的預估交易數。

選取**超過 10**，指出使用長交易。

然後，在**每分鐘的交易數**欄位中，選取每分鐘處理的預估交易數。若要決定此數目，請假設程序中的每個活動都有一個交易。因此，一分鐘可以執行的交易數如下：

$$\text{每分鐘執行的交易數} = \text{每分鐘完成的程序數} * \text{每個程序中的活動數}$$

d. 調整資料庫，使其具有較快的交易效能以及較慢的回復。

在**優先順序**中，選取**較快的交易效能**。

e. 如果可能，請調整資料庫，其正式作業中已移入一般資料量。在**已移入**中，選取**是**。否則，請選取**否**。



f. 調整平行連線設定。

在**連線**中，指定與應用程式伺服器間可執行的最大平行連線數。判斷此值的準則如下：

- 需要的資料庫連線數是根據與 WebSphere Application Server 的「Java 資料庫連線功能 (JDBC)」連線數決定。JDBC 連線是透過 JDBC 連線儲存區提供，而此連線儲存區是位在 WebSphere Application Server 中。如果有  $p$  個 JDBC 連線，則需要  $p * 1.1$  個資料庫連線。第 523 頁的『調整應用程式伺服器』會說明如何預估  $p$  的實際值。
- 如果「商業程序編排器」及資料庫是安裝在相同的實體伺服器上，則「商業程序編排器」不需要遠端資料庫連線。然而，因為需要有遠端連線才能管理遠端資料庫，所以請指定較小值，而不是零。
- 如果「商業程序編排器」及 DB2 是安裝在不同的伺服器上，請根據先前說明的本端連線規則來設定遠端應用程式的數目。

g. 在**隔離**中，選取**讀取穩定性**。「商業程序編排器」需要此隔離層次。

配置顧問會顯示建議的變更。您可以立即套用變更，也可以將它們儲存至檔案，稍後再套用。

長時間執行的程序會以現行環境及載入條件下的最快速度來執行。

## 指定起始 Oracle 資料庫設定

加強 Oracle 資料庫的效能及可調整性，主要是透過最佳化資料庫檔案的佈置、配置足夠記憶體給緩衝區快取以啓用有效快取，並調整資料庫參數。

1. 配置足夠空間給緩衝區快取。

使用記憶體內快取，以產生資料庫存取的低延遲回應時間。這表示緩衝區快取必須夠大。將緩衝區快取大小至少設定為 700 MB，然後監視快取用量，並在必要時增加快取大小。

2. 調整日誌檔大小，以減少日誌檔交換

Oracle 實例的交易日誌位於數個以循環方式使用的檔案中。當某個現行日誌檔案滿載時，即會切換現行日誌檔案，以容許保存最後一個作用中日誌。因為交換日誌檔是昂貴作業，所以請調整日誌檔大小，讓這些切換不會經常發生；750 MB 是很好的起始值。然後，監視交易率及平均日誌大小，並在需要時調整此值。

3. 調整下列資料庫參數。

### UNDO\_TABLESPACE

請確定不使用超過其大小限制 70 個百分比 (%) 的復原表格空間。

### OPEN\_CURSORS

這個參數的預設值為 50。不過，此值經常不足。您可以對這個參數使用的最高值視您的作業系統而定。在大部分作業系統上，支援最多 1000 的值。

### MAX\_SHARED\_SERVERS

指定可以同步執行的共用伺服器程序數目上限。使用這個參數，保留其他程序的程序插槽，例如專用伺服器。如果指定了 MAX\_SHARED\_SERVERS 參數的值，它應該大於或等於 SHARED\_SERVERS 參數的值，而且應該小於 PROCESSES 參數的值。比方說，如果有 150 個並行使用者，這個參數的良好起始值為 MAX\_SHARED\_SERVERS=70

## 規劃傳訊引擎設定

您可以這項作業來規劃傳訊引擎的起始設定。

若要達到長時間執行程序的最佳效能，請調整傳訊系統，以取得持續訊息的最大效能。若為資料儲存庫後端類型，則優先選用檔案儲存庫，因為它的執行效果很好。如果您的環境在叢集中執行，而且您無法使用檔案儲存庫，請使用資料庫資料儲存庫。

如果您使用 WebSphere Application Server 的服務整合功能，請遵循 WebSphere Application Server Network Deployment 資訊中心內調整服務整合內容的設定的給定指示，以設定並調整傳訊引擎的資料儲存庫。

您的傳訊引擎會以最佳狀況運作。

### 相關工作

第 103 頁的『規劃傳訊引擎資料庫』

您可以針對「商業程序編排器」匯流排的傳訊引擎使用不同的資料庫，以改進效能。

## 調整應用程式伺服器

使用這項作業來調整應用程式伺服器。

啟動這項作業之前，您必須已指定資料庫的起始設定，如第 519 頁的『指定起始 DB2 資料庫設定』中所述。

若要確保商業程序儲存器能以最佳狀況執行，則需要調整伺服器設定。

1. 預估每個商業程序儲存器需要的應用程式伺服器資源。
  - a. 一個資料來源，可以將商業程序狀態資訊讀取及寫入至伺服器範圍 DB2 Universal JDBC Driver Provider (XA) 的 BPEDDataSourceDb2 資料庫中
  - b. 新增下列項目，計算程序導覽的最大並行交易數  $t$ ：
    - 透過「商業程序編排器」API 並行連接的最大用戶端數
    - JMS 啟動規格 BPEInternalActivationSpec 中定義的並行端點數
    - JMS 啟動規格 HTMInternalActivationSpec 中定義的並行端點數若要檢視程序伺服器的啟動規格，請在管理主控台中按一下**資源** → **JMS 提供者** → **預設傳訊** → **啟動規格**。
  - c. 若為「商業程序編排器」資料庫，計算所需的平行 JDBC 連線數， $p = 1.1 * t$   
 $p$  的值不得大於資料庫容許的連線數。
  - d. 若為傳訊資料庫，計算所需的平行 JDBC 連線數， $m = t + x$ ，其中  $x$  是額外 JMS 階段作業數，容許產生額外訊息並且必須予以服務的超載狀況
  - e. 將「SQL 陳述式」快取大小設定為 50
2. 調整「商業程序編排器」資料庫的 JDBC 提供者設定 (BPEDB)。
  - a. 將**最大連線數**設為  $p$  值。 $p$  的值不得大於資料庫容許的連線數。
  - b. 將 **SQL 陳述式快取大小** 設為 300。
3. 重複步驟 2，調整 SCA 傳訊引擎資料儲存庫的 JDBC 提供者設定。
4. 調整傳訊資料庫的 JDBC 提供者設定。

將**最大連線數**設為  $m$  值。

## 5. 調整資料堆大小。

這裡是 32 位元系統上伺服器資料堆大小的部分準則。這些準則不適用在 i5/OS 系統上執行的伺服器。

- 256 MB 太低，因此會導致效能不佳。
- 512 MB 對許多系統的起始資料堆大小而言已足夠。
- 1024 MB 是合理的上限。

若為 64 位元系統，1 到 2 GB 是資料堆的合理大小。

## 6. 調整商業程序所使用的任何服務。請確定所支援的服務，已經調整到可以處理「商業程序編排器」的並行作業及工作量需求。

已調整應用程式伺服器，以改善效能。

### 相關工作

第 518 頁的『平衡硬體資源』

您可以透過平衡硬體資源來改善長時間執行的商業程序的效能。

## 細部調整資料庫

使用這項作業來細部調整資料庫。

商業程序儲存器及商業程序必須在執行中。

**註：**如果不是使用 DB2，請參閱資料庫管理系統文件，以取得監視資料庫效能、識別和去除瓶頸以及細部調整其效能的相關資訊。此主題的其餘部分會提供對 DB2 使用者的建議。不過，這個資訊不適用於 DB2 UDB for iSeries。

### 1. 根據其用量及命中率指派緩衝池的大小

緩衝池命中率指出可以從已在儲存區的資料滿足的資料庫要求百分比。它應該接近 100 百分比，但是任何超過 90 百分比的值皆可接受。增加緩衝池的 **SIZE** 參數，直到您取得滿意的命中率。監視總記憶體配置。如果您讓緩衝池變得太大，系統就會開始交換。在此情況下，請降低緩衝池的大小，或讓其他記憶體可供使用。

如果是使用 DB2 第 8 版，您可以計算緩衝池命中率。您可以從緩衝池 Snapshot 中取得計算所需的值。使用下列指令來取得 Snapshot：

```
DB2 get snapshot for all bufferpools
```

如需計算命中率的相關資訊，請參閱 DB2 第 8 版資訊中心。

如果是使用 DB2 第 9 版，請使用 BP\_HITRATIO 管理視圖，來擷取命中率資訊。如需這個視圖的相關資訊，請參閱 DB2 第 9 版資訊中心。

「DB2 配置顧問」會建議緩衝池大小的值，但是依預設，「商業程序編排器」資料庫只會使用 IBMDEFAULTBP 預設緩衝池。您可以使用下列指令，設定這個緩衝池的大小：

```
DB2 ALTER BUFFERPOOL IBMDEFAULTBP SIZE 120000
```

這個指令會顯示緩衝池及其大小 (以頁為單位)，以及每一頁的大小：

```
DB2 select BPNAME, NPAGES, PAGESIZE from syscat.bufferpools
```

### 2. 調整鎖定清單空間，以協助確保最佳效能。

所有鎖定都需要儲存體，因此這個儲存體會受到限制。要求鎖定數超出這個限制的交易必須中斷，因此它們會降低效能。

- a. 請檢查 DB2 實例的 db2diag.log 檔案。

請尋找與下列範例類似的項目：

```
2005-07-24-15.53.42.078000 Instance:DB2 Node:000
PID:2352(db2syscs.exe) TID:4360 Appid:*LOCAL.DB2.027785142343
data management sqlEscalateLocks Probe:4 Database:BPEDB
```

```
ADM5503E The escalation of "10" locks on table
"DB2ADMIN.ACTIVITY_INSTANCE_B_T" to lock intent
"X" has failed. The SQLCODE is "-911".
```

這種類型的訊息指出已超出鎖定空間。

- b. 增加 MAXLOCKS 及 LOCKLIST 參數的值。

這些參數控制鎖定呈報中的資料庫行爲。鎖定呈報會將相同表格中的數個個別低階鎖定轉換爲單一表格層次鎖定。

如果交易不只使用鎖定清單的 MAXLOCKS 參數，則資料庫管理程式會將這些鎖定轉換爲單一表格鎖定，以便不超出鎖定空間限制。不過，鎖定呈報會顯著地增加死鎖的機率。因此，請將 MAXLOCKS 參數的值增加至 60 百分比。

請將 LOCKLIST 參數的值大約增加至  $10 * p$ ，其中  $p$  是任何時間需要的平行 JDBC 連線數上限的預估值。例如，如果使用  $p=50$  的值調整「商業程序編排器」資料庫 BPEDB 的大小，請輸入下列指令：

```
db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500
```

3. 如果您已使用「DB2 配置顧問」，則資料庫產量可能會十分良好。不過，您可以使用下列方式進一步改善效能：

- 遵循 DB2 線上文件、書籍及文章中說明的資料庫調整的最佳作法。
- 調整下列 DB2 參數：

#### **AVG\_APPLS**

這個參數設得太高會比設得太低來得好。例如，如果最多有 20 個已連接的應用程式，請將 AVG\_APPLS 設爲 50。

#### **DLCHKTIME**

這個參數指定死鎖偵測時間範圍。預設值爲 10 秒。

#### **LOCKTIMEOUT**

這個參數指定應用程式等待鎖定的時間。預設值爲 -1，表示應用程式會等到授與鎖定或發生死鎖爲止。這個參數的值應該一律大於 DLCHKTIME 參數的值，讓死鎖被報告爲死鎖，而不會被報告爲鎖定逾時。這個參數的良好起始值爲 30 秒。如果負載測試顯示交易時間的長度超過 30 秒，您可能想要設定更高的值。

#### **LOGBUFSZ**

增加 DB2 日誌的緩衝區大小，可減少必須將已滿的日誌緩衝區寫入至磁碟的頻率。

#### **LOG\_FILSIZ**

增加日誌檔的大小，可減少它們的交換頻率。

4. 依據工作量需求，調整資料庫及資料庫管理程式設定。在配置顧問配置資料庫之後，您也可以調整下列設定：

#### MINCOMMIT

強烈建議使用值 1。「DB2 配置顧問」可能會建議其他值。

#### NUM\_IOCLEANERS

若為只查詢應用程式，請將值設定為 0，若為正規處理，請使用 1 與系統中磁碟驅動器數目之間的值 (另請參閱 NUM\_IOSERVERS 參數)。若為大型緩衝池，愈高的數目通常愈有利。

#### NUM\_IOSERVERS

必須符合資料庫所在實體磁碟數。IO 伺服器的數目至少應該和磁碟的數目相同。IO 伺服器不會使用許多系統資源，因此將值設得太高會比設得太低來得好。

5. 更新資料庫的 DB2 統計資料。

在最初將負載置於系統之後，或每當資料庫中的資料磁區大量變更時，請考量更新包含統計資訊的 DB2 系統型錄表格。使用 RUNSTATS 指令來更新統計資訊。

RUNSTATS 指令是最適合使用 Script 來執行。下列範例顯示這類 Script。它假設您是以使用者 bpeuser 身分，採用密碼 password 登入，而且您已連接至「商業程序編排器」資料庫 BPEDB。DB2 指令會產生一個 Windows 指令檔，更新 BPEDB 資料庫中相關表格空間內所有表格的統計資訊。因為不會經常存取或更新資訊，所以會省略 TEMPLATE 表格空間表格。

```
db2 -x "select 'db2 runstats on table '
        concat rtrim(tabschema)
        concat '.'
        concat tablename
        concat ' with distribution and detailed indexes all '
from syscat.tables
其中
        type='T' AND
        TBSPACEID IN (select TBSPACEID from sysibm.systablespace
        where TBSPACE IN ('INSTANCE', 'WORKITEM', 'STAFFQUERY',
        'AUDITLOG', 'SCHEDTS'))" > runStatsScript.cmd

echo db2 connect reset >> runStatsScript.cmd
```

結果指令檔會更新指定表格的統計資訊。它包含如下的項目：

```
db2 runstats on table BPEUSER.ACTIVITY_INSTANCE_B_T with distribution and
detailed indexes all
db2 runstats on table BPEUSER.AUDIT_LOG_T with distribution and
detailed indexes all
...
db2 connect reset
```

您可能想要延伸指令檔，以在呼叫 RUNSTATS 指令之前，執行 REORG 指令。請參閱 DB2 文件，以取得如何使用 REORG 指令來重組資料庫表格的相關資訊。

6. 避免死鎖。

當至少有兩個交易封鎖彼此的資源存取時，便會發生死鎖。死鎖可能是由於不良的資料庫配置所造成。它們也可能是因為使用「商業程序編排器」API 的方式所造成。若要避免死鎖，資料庫中物件的每一個 API 呼叫或查詢都應該在自己的交易中執行。

「商業流程管理程式」可以從資料庫死鎖中回復。不過，由於偵測到死鎖與重試回復交易之間的時間，可能會造成主要效能影響。因此，基於效能原因，應該避免死鎖。

若要檢查死鎖，請檢查 db2diag.log 檔，並使用 DB2 監視器。

- a. 增加 db2diag.log 檔的記載層次，以取得資料庫內瓶頸的相關資訊。

將 DIAGLEVEL 參數值從 3 (預設值) 增加至 4，以包括錯誤、警告及參考訊息。您可以使用下列指令來變更值：

```
db2 update dbm cfg using DIAGLEVEL 4
```

- b. 建立 DB2 事件監視器。

事件監視器提供某些事件 (例如，死鎖) 的相關資訊。

- 1) 使用下列指令來建立事件監視器：

```
db2 create event monitor monitor_name for statements, connections,  
transactions, deadlocks with details write to file file_name
```

- 2) 使用下列指令來啟動事件監視器：

```
db2 set event monitor monitor_name state=1
```

- 3) 使用下列指令來收集資訊：

```
db2evmon -db database_name -evm monitor_name output_file_name
```

- c. 使用資料庫 Snapshot 監視器來收集統計資訊。

Snapshot 監視器會使用資料庫監視器開關。若為資料庫實例，監視器開關具有下列預設值：

Buffer pool (DFT\_MON\_BUFPOOL) = ON

Lock (DFT\_MON\_LOCK) = ON

Sort (DFT\_MON\_SORT) = OFF

Statement (DFT\_MON\_STMT) = OFF

Table (DFT\_MON\_TABLE) = OFF

Timestamp (DFT\_MON\_TIMESTAMP) = ON

Unit of work (DFT\_MON\_UOW) = OFF

若要查看資料庫實例的現行設定，請使用下列指令，然後搜尋所有開頭為 DFT\_MON\_ 的參數：

```
db2 get dbm cfg
```

這些設定不同於資料庫的設定。若為資料庫，監視器記錄開關具有下列預設值：

Switch list for db partition number 0

Buffer Pool Activity Information (BUFFERPOOL) = ON

Lock Information (LOCK) = ON

Sorting Information (SORT) = OFF SQL

Statement Information (STATEMENT) = OFF

Table Activity Information (TABLE) = OFF

Take Timestamp Information (TIMESTAMP) = ON



Unit of Work Information (UOW) = OFF

若要查看資料庫的現行設定，請使用下列指令：

```
db2 get monitor switches
```

- 若要更新其中一個資料庫監視器 (例如，鎖定監視器) 的設定，請使用下列指令：

```
db2 update monitor switches using lock on
```

這個設定只適用於現行資料庫階段作業。

- 若要更新資料庫實例的其中一個監視器開關 (例如，鎖定監視器) 的設定，請使用下列指令：

```
db2 update dbm cfg using DFT_MON_LOCK OFF
```

若要啟動設定，請重新啟動資料庫實例。

- 在啟用 Snapshot 監視器之前，請使用下列指令來重設計數器：

```
db2 reset monitor all
```

- 若要在重新啟動資料庫實例之後取得現行 Snapshot，請使用下列指令：

```
db2 get snapshot for all on database_name output_file_name
```

長時間執行的程序會以現行環境及載入條件下的最快速度來執行。

## 細部調整傳訊提供者

使用這項作業來改善傳訊提供者的效能。

如果您使用 WebSphere Application Server 的服務整合功能，請參閱 WebSphere Application Server 資訊中心中的調整及解決傳訊引擎資料儲存庫的問題。

已改善傳訊提供者的效能。

---

## 調整微流程

使用這項作業來改善微流程的效能。

微流程是在記憶體內執行，而且沒有任何使用者互動或持續傳訊支援。只有在啟用微流程的審核日誌或「共用事件基礎架構 (CEI)」時，才需要存取資料庫。微流程的處理程式是在單一執行緒中進行，而且一般是在單一交易中。微流程的效能主要是根據呼叫的服務而定。然而，如果伺服器可用的記憶體太少，則會降低微流程的效能。

1. 調整「Java 虛擬機器 (JVM)」資料堆大小。

因為較大的資料堆大小可減少需要的記憶體回收週期數，所以增加 Java 資料堆大小，可以改善微流程的產量。請盡量讓此值小一點，以避免將資料堆交換至磁碟。如需伺服器資料堆大小的準則，請參閱第 523 頁的『調整應用程式伺服器』中的相關步驟。

2. 調整 JVM 記憶體回收作業。使用「產量記憶體回收器」可以達到最佳產量，然而，記憶體回收會暫停 100-1000 毫秒 (視資料堆大小而定)。如果回應時間比產量重要，則請使用「低暫停記憶體回收器」。
3. 調整 Object Request Broker (ORB) 執行緒儲存區大小。如果遠端用戶端連接至伺服器端 ORB，則請確定「ORB 執行緒儲存區」中有足夠的可用執行緒。



4. 調整預設執行緒儲存區大小。若要增加可以並行執行的微流程數目，則必須增加預設執行緒儲存區大小。若要使用管理主控台來變更值，請按一下**伺服器** → **應用程式伺服器** → **伺服器** → **新增內容** → **執行緒儲存區** → **預設值**。

微流程會以現行環境及載入條件下的最快速度來執行。

---

## 調整包含人力作業的商業程序

有不同的方式可以改進包含人力作業的商業程序的效能。

下列主題說明如何調整包含人力作業的商業程序。

### 減少人力作業的並行存取

當兩個以上人員嘗試要求相同的人力作業時，只有一位人員會成功。其他人會被拒絕存取。

只有一位人員可以要求人力作業。如果同時有數個人嘗試使用相同的人力作業，則會增加衝突機率。因為資料庫的鎖定等待或回復，衝突會導致延遲。避免或減少衝突發生率的部分方式如下所示：

- 如果並行存取較高，請限制可以存取特定人力作業的使用者數目。
- 透過使用智慧型要求機制，避免來自用戶端的不必要人力作業查詢。例如，您可能執行下列其中一個步驟：
  - 如果第一個要求不成功，請嘗試從清單要求另一個項目。
  - 一律要求隨機人力作業。
  - 減少作業的可能擁有者數目，例如透過將作業指派至具有較少成員的群組。
  - 透過指定用來擷取清單之查詢的臨界值，來限制作業清單的大小。另請考量使用過濾來限制相符數目。例如，您可以過濾作業的內容，只顯示優先順序為一的作業，或從現在開始 24 小時內到期的作業。若為行內作業，您也可以使用自訂內容或查詢內容，過濾與作業相關聯的商業資料。若要執行此種過濾，您必須在擷取作業清單的查詢上指定適當的 `where` 子句。
  - 最小化或避免動態人員查詢，也就是說，使用取代變數的人員查詢。
  - 使用人力作業查詢的用戶端快取機制，來避免同時執行數個查詢。

### 減少查詢回應時間

減少資料庫用來回應查詢的時間。

當您使用自訂用戶端時，請確定查詢已設定臨界值。從可用性觀點看，通常不希望擷取成千上百個項目，因為資料庫作業的數目越多，要完成作業的時間就越長，也因為使用者一次只可以管理少量的結果。透過指定臨界值，您可最小化資料庫負載及網路資料傳輸，且有助於確保用戶端可以快速呈現資料。

處理傳回大量項目的查詢的更好方式是重新撰寫查詢，以傳回較小的項目結果集。您可以透過只查詢特定程序實例的工作項目，或只具有特定日期的工作項目，來執行此動作。

您也可以透過使用過濾準則來減少查詢結果。

## 避免掃描整個表格

當您使用查詢應用程式設計介面 (API)，來列出資料庫中的物件時，您可以指定過濾器來縮小您想要擷取的結果。在這些過濾器中，您可以指定物件屬性的值及範圍。

當處理資料庫查詢時，過濾器資訊會轉換為「結構化查詢語言 (SQL)」陳述式中的 WHERE 子句。這些 WHERE 子句會將物件屬性對映到受影響資料庫表格中的直欄名稱。

如果查詢指定的過濾器沒有轉換為索引表格直欄，則 SQL 陳述式可能將導致掃描表格。此掃描具有負面的效能影響，並增加死鎖的風險。如果每天只發生幾次，則還可容忍其對效能的影響，但如果每分鐘發生數次，則會降低效率。

在此情況下，自訂索引可以大大減少該影響。在實際客戶的情況下，自訂索引協助將 API 回應時間從 25 秒減少到 300 毫秒。無需讀取 724 000 列的資料庫表格，只需讀取六列。如果在「商業程序編排器」資料庫中定義實例表格的暫時旗標，則這有助於 DB2 最佳化程式決定適當的資料存取計劃。此旗標指定一律會使用索引，而不是表格掃描，即使具有空的或幾乎全空的表格也是一樣。

視您指定的過濾準則而定，索引中可能不會併入部分直欄。如果在此情況下，且如果使用表格掃描，導致降低查詢效能，請檢查陳述式的存取路徑，例如使用 DB2 Explain。必要的話，請定義新索引。

## 最佳化作業及程序查詢

擷取作業及程序清單的 query 及 queryAll API 呼叫可能導致複式 SQL 查詢，包括多個資料庫表格的組合。資料的最佳表示法可協助處理效能需求，尤其針對多個使用者同時存取作業清單的人工工作流程應用程式。

如果已針對查詢調整「商業程序編排器」，則即使在高負載下，回應時間通常仍位於足夠大小的系統上的子秒區域中。您可以套用標準資料庫計算，以計算查詢的回應時間。

大容量人工工作流程範例情況最適合利用查詢表格來調整。查詢表格提供與特定查詢相關的預先計算資料集。例如，當查詢執行時，查詢內容必須透過資料庫與作業或程序實例結合。如果使用查詢表格，在執行查詢時，不再需要計算這些 SQL 結合。




查詢表格的實作及維護效能高於標準資料庫調整技術。在使用查詢表格之前，仔細考量標準資料庫最佳化技術，例如索引、日誌檔配送及記憶體。

支援兩種查詢表格的方法：具體化視圖及自訂表格。根據維護成本、開發成本，以及作業及程序清單查詢所傳回的現行資料需求，判斷是否使用具體化視圖或自訂表格：

- 使用具體化視圖來利用非同步更新機制，以提供最佳的查詢及程序導覽效能。
  - 只在使用具體化視圖時，才會發生更新
  - 設定、使用及維護相對而言比較簡單
  - 可以實作，不會變更應用程式來源程式碼
- 使用自訂表格，利用 query 或 queryAll 介面，包括來自標準查詢中其他應用程式的資料。此外，自訂表格也可以用來提供作業及程序查詢所需資料的最佳表示法
  - 資料庫觸發程式或其他技術可以用來同步更新針對作業及程序清單查詢而最佳化的自訂表格

- 查詢必須加以變更，才能查詢自訂表格中提供的資料

#### 相關資訊

-  商業程序編排器 `query()` 及 `queryAll` 方法：最佳作法
-  調整人工工作流程
-  DB2 資訊中心：具體化查詢表格



---

## 第 22 章 調整商業程序編排器瀏覽器

下列建議提供各種方法，來改善「商業程序編排器瀏覽器」的效能。

1. 考量增加伺服器的資料堆大小上限。

Web 用戶端會自然地增加您系統上的負載。連接至您伺服器的用戶端越多，必須保存在記憶體中的物件就越多。因此，考量增加伺服器的資料堆大小上限。這會改善您應用程式的回應時間，而且會增加可以同時使用應用程式的使用者數目上限。

2. 調整 Web 儲存器執行緒儲存區。

執行緒儲存區的大小及執行緒閒置逾時可能會影響 Web 儲存器的效能。您可以使用管理主控台來變更這些設定 (伺服器 → 應用程式伺服器 → **server\_name** → 執行緒儲存區 → **Web 儲存器**)。

- a. 調整儲存區大小上限及下限。

Web 用戶端應用程式的所有 HTTP 要求都是使用來自 Web 儲存器執行緒儲存區的執行緒來處理的。您可以調整儲存區大小的下限及上限，來影響 Web 用戶端的效能。

儲存區中的執行緒數目上限不代表您的應用程式伺服器可以同時處理的要求數目。如果儲存區中的所有執行緒都在使用中，則其他要求會置於佇列中，直到它們可以指派給執行緒。如果用戶端要求等待指派執行緒，則用戶端的回應時間會增加。不過，如果數目上限設定太高，則系統可能會超載，導致用戶端的回應時間更糟。它也可能導致其他應用程式的執行急速減慢。

若要判斷變更儲存器大小是否可能導致效能增益，您可以使用「Tivoli® 效能檢視器」，監視執行緒上的負載 (PercentMaxed 計數器) 及 Web 儲存器模組的作用中執行緒數目 (ActiveThreads 計數器)。如果 PercentMaxed 計數器的值持續保持兩位數，則 Web 儲存器可能遭遇瓶頸。在此情況下，請增加執行緒數目。如果作用中執行緒的數目低於儲存區中的執行緒數目，則降低執行緒儲存區大小可能導致效能增益。

- b. 調整執行緒無活動逾時。

執行緒無活動逾時定義應該經過多少秒無活動之後，才收回執行緒。變更此值也可能影響您的回應時間。0 值指出沒有等待時間。

3. 降低大型清單的臨界值。

如果您是使用大型作業或程序清單，您可能想要降低清單的搜尋限制，以避免收集使用者不存取的資料。

### 相關工作

第 183 頁的『使用管理主控台來配置商業程序編排器瀏覽器』  
您可以使用管理主控台來配置「商業程序編排器瀏覽器」。

### 相關參考

第 183 頁的『使用 clientconfig.jacl Script 檔配置商業程序編排器瀏覽器』  
此 Script 檔會在伺服器或叢集上配置「商業程序編排器瀏覽器」及所有必要的資源。



---

## 第 23 章 調整商業程序編排器觀察程式

產生報告所需的時間會不同，而且會視許多因素而異。下列建議提供各種方式來改善「商業程序編排器觀察程式」的效能。

### 選擇適當的資料庫管理系統

「商業程序編排器觀察程式」可以使用 DB2、Oracle 或 Derby 資料庫。如果效能不是主要考量，而且資料不多，則 Derby 資料庫最適合用來開發、示範及原型化。然而，在正式作業系統中，因為 DB2 或 Oracle 資料庫在處理大量資料時較為快速，所以請使用它們。

### 更新資料庫統計資訊

若為 DB2 及 Oracle 資料庫，如果您在擁有移入的正式作業資料庫時更新資料庫統計資訊，則可以大幅改善效能。

- 若要更新 DB2 資料庫的統計資訊，請輸入下列指令：

```
RUNSTATS ON TABLE schema_prefix.EVENT_ACT_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.EVENT_PRC_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.INST_ACT_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.INST_PRC_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.OPEN_EVENTS_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.QUERY_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.SLICES_T FOR INDEXES ALL;
```

- 若要更新 Oracle 資料庫的統計資訊，請輸入下列指令：

```
ANALYZE TABLE schema_prefix.EVENT_ACT_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.EVENT_PRC_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.INST_ACT_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.INST_PRC_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.OPEN_EVENTS_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.QUERY_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.SLICES_T COMPUTE STATISTICS;
```

其中 *schema\_prefix* 是在建立「商業程序編排器觀察程式」資料庫時使用的資料庫綱目名稱。如需更新資料庫統計資訊的相關資訊，請參閱資料庫的文件。

### 僅發出相關事件

在 WebSphere Integration Developer 中，您可以極為詳細地定義活動或程序的記載。如果也針對包含活動的程序產生事件，則只有「商業程序編排器觀察程式」才能辨識活動審核事件。事件收集程式應用程式會略過與程序無關的活動事件，而且這些事件也不會儲存在資料庫中。若要減少所發出事件的數目，請執行下列步驟：

1. 選取想要審核的程序範本，並停止發出您不感興趣的程序的事件。
2. 選取想要審核的這個程序範本的活動。請確認是否可以省略部分事件，而不影響報告結果。

若要讓「商業程序編排器觀察程式」提供精確的活動或程序內容，您應該審核所有事件類型或不要審核任何事件類型。



## 選擇使用者定義的函數 (UDF) 實作

若要建立報告，則「商業程序編排器觀察程式」需要在資料庫中安裝一些特定 UDF。提供 UDF 作為 SQL 型實作及 Java 型實作。SQL 實作執行的速度會比 Java 實作快，但是有一些缺點。如果使用 Java 實作，請考慮切換至 SQL 實作。

## 使用不同的資料庫

如果「商業程序編排器觀察程式」使用的資料庫與「商業程序編排器」相同，則它們會對彼此的效能產生負面影響。如果「商業程序編排器觀察程式」具有不同的資料庫，則會有較好的效能。也請考慮在不同的資料庫伺服器上管理「觀察程式」資料庫。

## 使用不同的機器

如果「商業程序編排器觀察程式」安裝在管理其他應用程式 (例如 BPEContainer 或 TaskContainer 應用程式) 的機器上，請考慮在資源足以符合效能預期的不同機器上執行「商業程序編排器觀察程式」。

## 增加逾時值

產生報告需要的時間較長。如果時間過久，則可能會發生 JDBC 驅動程式的交易逾時或連線逾時。如果發生此狀況，請如下所示增加逾時值：

1. 在管理主控台中，導覽至伺服器 → 應用程式伺服器 → **server\_name** → 交易服務。
2. 如果總交易生命期限逾時值小於最大交易逾時值，請讓它們的值相同。
3. 如果效能仍然發生問題，請將總交易生命期限逾時值設為 0，並增加最大交易逾時值。
4. 如果效能仍然發生問題，請將總交易生命期限逾時及最大交易逾時值都設為 0，並增加 JDBC 驅動程式的連線逾時值。若要執行此作業，請導覽至資料來源的連線儲存區內容。請按一下 JDBC → JDBC 提供者 > JDBC 提供者 → 資料來源 → **data\_source\_name** → 連線儲存區內容，並增加連線逾時值。

在伺服器叢集中，必須調整所有叢集成員的交易逾時值。

## 刪除不必要的資料

報告效能是視觀察程式資料庫中的實例及事件資料數量而定。如果報告查詢大量資料，則會降低效能。如果減少「商業程序編排器觀察程式」資料庫中的程序及活動實例數目，則可以改善報告效能。定期刪除不必要或舊的資訊，有助於改善效能。

### 相關工作

第 211 頁的『在 Java 及 SQL 使用者定義函數之間進行選取』

您可以使用 setupEventCollector 工具或執行 Script，在「商業程序編排器觀察程式」資料庫中的 Java 型及 SQL 型使用者定義函數 (UDF) 之間進行切換。

第 261 頁的『從觀察程式資料庫刪除資料』

您可以使用管理指令，以選擇性地從「商業程序編排器觀察程式」資料庫刪除程序實例的所有資料，這些實例需符合指定條件。刪除不需要的資料可以改善產生報告的效能。

---

## 第 7 篇 疑難排解



---

## 第 24 章 商業程序編排器配置疑難排解

您可以使用這個主題，解決與「商業程序編排器」配置及其「商業流程管理程式」或「人力作業管理程式」元件相關的問題。

本節的目的是幫助您瞭解「商業流程管理程式」或「人力作業管理程式」未如預期運作的原因，並協助您解決問題。下列作業著重在問題判斷，並尋找解決方案，以解決配置期間可能發生的問題。

### 相關資訊



WebSphere Process Server 疑難排解手冊

---

### 商業程序編排器日誌檔

這說明可以找到「商業程序編排器」配置的日誌檔的位置。

#### 建立設定檔

「商業程序編排器」的設定檔動作會寫入至設定檔工具的 logs 目錄中的 bpcaugment.log 檔。您可以在相同目錄的 createBPCObjects.traceout 檔中找到更詳細的追蹤資料。在 Windows 系統上，這些檔案位於 *install\_root*/logs/manageprofiles/*profileName*/logs 目錄中，而在 Linux、UNIX 及 i5/OS 系統上，它們位於 *install\_root*/logs/manageprofiles/*profileName*/logs 中。

如果在設定檔精靈中選取配置範例選項，則它會呼叫 bpeconfig.jacl Script，而且動作會記載至設定檔 logs 目錄的 bpeconfig.log 檔中。這個目錄位於 *profile\_root* 目錄中。

#### 管理 Script

使用 wsadmin 執行的所有「商業程序編排器」Script，都會記載至設定檔工具的 logs 目錄中的 wsadmin.traceout 檔。然而，因為每次呼叫 wsadmin 時都會改寫此檔案，所以請確定先儲存此日誌檔，然後再重新呼叫 wsadmin。

#### 配置相關的 Script

bpeconfig.jacl、bpeupgrade.jacl、clientconfig.jacl 及 bpeunconfig.jacl 這些 Script 檔會將它們的日誌檔以 bpeconfig.log、bpeupgrade.log、clientconfig.log 及 bpeunconfig.log 名稱寫入至 logs 目錄中。

下列配置 Script 會將它們的日誌檔分別寫入至 logs 目錄的 setupObserver.log 及 setupEventCollector.log 檔案中。

- setUpEventCollector.bat 及 setupObserver.bat (Windows 系統)
- setUpEventCollector.sh 及 setupObserver.sh (Linux 及 UNIX 系統)
- setUpEventCollector 及 setupObserver (i5/OS 系統)

也請檢查 wsadmin.traceout 檔。

## 管理公用程式 Script

ProcessChoreographer 目錄的 admin 子目錄中的管理 Script 不會寫入它們自己的日誌檔。請檢查 wsadmin.traceout 檔及應用程式伺服器日誌檔。

---

## 商業程序編排器資料庫及資料來源疑難排解

使用這項作業來解決「商業程序編排器」資料庫及資料來源的問題。

「商業流程管理程式」及「人力作業管理程式」兩者都需要資料庫。如果沒有資料庫，則內含商業程序及人力作業的企業應用程式會無法運作。

- 如果使用 DB2：
  - 如果使用 DB2 Universal JDBC Driver 第四類型，並在測試「商業程序編排器」資料來源的連線或在啟動伺服器時，收到 DB2 內部錯誤（如 "com.ibm.db2.jcc.a.re: XAER\_RMERR: The DDM parameter value is not supported. DDM parameter code point having unsupported value: 0x113f DB2ConnectionCorrelator: NF000001.PA0C.051117223022"），請執行下列動作：
    1. 檢查資料來源的類別路徑設定。在預設設定中，WebSphere 變數 `{DB2UNIVERSAL_JDBC_DRIVER_PATH}` 可以指向 WebSphere Process Server 內嵌式 DB2 Universal JDBC 驅動程式，而該驅動程式是位在 `universalDriver_wbi` 目錄中。
    2. 驅動程式版本可能與 DB2 伺服器版本不相容。請確定使用所安裝資料庫中的原始 `db2jcc.jar` 檔案，而不是 WebSphere Process Server 內嵌式 DB2 Universal JDBC 驅動程式。必要時，請將 WebSphere 變數 `{DB2UNIVERSAL_JDBC_DRIVER_PATH}` 的值變更為指向原始 `db2jcc.jar` 檔案。
    3. 請重新啟動伺服器。
  - 如果 DB2 實例的 `db2diag.log` 檔案包含 ADM5503E 這類訊息（如下所示）：

```
2004-06-25-15.53.42.078000 Instance:DB2 Node:000
PID:2352(db2syscs.exe) TID:4360 Appid:*LOCAL.DB2.027785142343
data management sqlEscalateLocks Probe:4 Database:BPEDB

ADM5503E The escalation of "10" locks on table "GRAALFS .ACTIVITY_INSTANCE_T"
to lock intent "X" has failed. The SQLCODE is "-911"
```

請增加 LOCKLIST 值。例如，若要將值設為 500，請輸入下列 DB2 指令：

```
db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500
```

這可以大幅改善效能。
  - 若要避免死鎖，請確定資料庫系統是配置成使用足夠的記憶體，尤其是緩衝池。若為 DB2，請使用「DB2 配置顧問」來決定配置的合理值。
  - 如果收到的錯誤提及資料來源實作類別 `COM.ibm.db2.jdbc.DB2XADataSource`：
    - 檢查 JDBC 提供者的類別路徑定義是否正確。
    - 檢查元件管理的鑑別別名是否已設定為 `BPCDB_nodeName.serverName_Auth_Alias`（如果已在伺服器上配置「商業程序編排器」）及 `BPCDB_clusterName_Auth_Alias`（如果已在叢集上配置「商業程序編排器」）。
- 如果您使用 Derby：
  - 如果在 Linux 或 UNIX 系統上收到「開啓太多檔案」錯誤，請增加可用的檔案控點數，例如，增加至 4000 以上。如需如何增加可用檔案控點數的相關資訊，請參閱作業系統的文件。

- 如果在嘗試呼叫 `ij` 指令行處理器時收到「找不到 Java 類別」異常，請確定已設定 Java 環境，且 `classpath` 環境變數包括下列 JAR 檔：
  - `derby.jar`
  - `derbytools.jar`
- 如果您是使用內嵌式的 Derby 驅動程式，但無法使用 Derby 工具 (如 `ij`) 連接至 Derby 資料庫，而且收到下列異常：

```
ERROR XJ040: Failed to start database 'c:\WebSphere\AppServer\profiles\profile_name\databases\BPEDB',
see the next exception for details.
ERROR XSDB6: Another instance of Derby may have already booted the database
c:\WebSphere\AppServer\profiles\profile_name\databases\BPEDB.
```

確定一次只有一個應用程式存取 Derby 資料庫。

- 如果您已寫入一個使用「商業程序編排器 API」的用戶端，但未先鑑別使用者，您應該修改用戶端，在使用 API 之前先執行登入。在移轉之後，J2EE 角色 `BPEAPIUser` 及 `TaskAPIUser` 會設定為值 `Everyone`，如此可以在啓用應用程式安全時維持不需要登入的 6.0.x 行爲，以維護舊版相容性。在修正用戶端之後，您必須將這些角色變更為值 `AllAuthenticated`，以防止未經鑑別的使用者存取 API。若爲新的安裝，這些角色預設爲值 `AllAuthenticated`。
- 如果在安裝包含商業程序或人力作業的企業應用程式時收到資料庫錯誤，請確定商業程序儲存器使用的資料庫系統正在執行中，而且可以存取。安裝企業應用程式時，會將所有程序範本及作業範本寫入至「商業程序編排器」資料庫。
- 如果使用國家字元時發生問題。請確定建立的資料庫支援 `Unicode` 字集。
- 如果在資料庫中找不到表格及視圖，而且未啓用建立綱目選項，請檢查下列情況：
  - 如果已配置資料庫綱目限定元，請檢查下列情況：
    - 綱目限定元必須符合資料庫中的綱目。它必須與 `Script` 中使用的綱目相同。
    - 必須將使用資料庫表格及視圖的專用權授與使用者。
  - 如果未配置任何綱目限定元，請確保：
    - 使用者的鑑別別名必須是用來執行 `Script` 的同一個使用者 ID，或必須符合 `Script` 中使用的綱目限定元。
    - 必須將使用資料庫表格及視圖的專用權授與使用者。
- 如果已啓用建立綱目選項，但找不到資料庫表格及視圖，將自動使用下列條件建立資料庫表格及物件：
  - 如果已配置綱目限定元，將使用綱目限定元來建立表格及視圖。
  - 如果未配置任何綱目限定元，將使用使用者 ID 來建立表格及視圖。

---

## 已啓用代理時，作業儲存器無法啓動

您已建立新的「商業程序編排器」配置，並已啓用代理，或者已針對現有配置啓用代理。

### 症狀

`TaskContainer_suffix` 應用程式無法啓動。

### 原因

已啓用人員代理，但未正確配置。

## 解決方案

若要讓代理運作：

- 您必須啟用應用程式安全。
- 您的「人力作業管理程式」必須使用「虛擬成員管理程式 (VMM)」人員目錄提供者 (聯合儲存庫)。
- 如果您想要使用 LDAP 目錄，則必須將 VMM 配置為使用 LDAP。

作為暫行解決方法，您可以停用「人力作業管理程式」的代理，然後重新啓動作業儲存器：

1. 使用管理主控台，按一下**伺服器** → **應用程式伺服器** → **server\_name** 或**伺服器** → **叢集** → **cluster\_name**，然後在**商業整合**下，展開**商業程序編排器**，並按一下**人力作業管理程式** → **執行時期**
2. 取消勾選**啟用代理**選項。
3. 按一下**套用**。
4. 重新啓動 `TaskContainer_suffix` 應用程式。

重新啓用代理之前，請確定您已執行下列各項：

1. 第 105 頁的『**規劃人員目錄提供者**』
2. 第 169 頁的『**配置虛擬成員管理程式人員目錄提供者**』
3. 第 175 頁的『**配置人員替代**』

### 相關工作

第 554 頁的『**人員指派疑難排解**』

使用下列資訊來協助解決指派人員給授權角色時的相關問題。

---

## 6.0.x 商業程序編排器 API 用戶端無法在 6.1 環境中執行

升級至 WebSphere Process Server 6.1 版時，您並未移轉「6.0.x 商業程序編排器」API 用戶端。當您嘗試在 6.1 環境中執行用戶端時，用戶端失敗。

### 症狀

如下的異常會寫入至 `SystemOut.log` 檔：

```
[9/6/07 21:05:27:093 PDT] 00000045 ExceptionUtil E CNTR0020E: EJB threw an unexpected (non-declared) exception during invocation of method "processMessage" on bean "BeanId(validateDataApp#validateDataEJB.jar#component.validateItem, null)".
Exception data: javax.ejb.AccessLocalException: ;
nested exception is: com.ibm.websphere.csi.CSIAccessException:
SECJ0053E: Authorization failed for /UNAUTHENTICATED while invoking
(Home)com/ibm/bpe/api/BusinessFlowManagerHome create:4
securityName: /UNAUTHENTICATED;accessID: UNAUTHENTICATED is not granted any of the required
roles: BPEAPIUser
com.ibm.websphere.csi.CSIAccessException: SECJ0053E: Authorization failed for
/UNAUTHENTICATED while invoking (Home)com/ibm/bpe/api/BusinessFlowManagerHome
create:4 securityName: /UNAUTHENTICATED;accessID: UNAUTHENTICATED is not granted any of the required
roles: BPEAPIUser
at com.ibm.ws.security.core.SecurityCollaborator.performAuthorization(SecurityCollaborator.java:484)
at com.ibm.ws.security.core.EJSSecurityCollaborator.preInvoke(EJSSecurityCollaborator.java:218)
at com.ibm.ejs.container.EJSContainer.preInvokeForStatelessSessionCreate(EJSContainer.java:3646)
at com.ibm.ejs.container.EJSContainer.preInvoke(EJSContainer.java:2868)
at com.ibm.bpe.api.EJSLocalStatelessGenericBusinessFlowManagerEJBHome_a412961d.create(Unknown Source)
```



## 原因

如果您的「商業程序編排器」API 用戶端不包括使用者鑑別，則它會根據安全孔。在 WebSphere Process Server 6.1 版中，已修正這個導致用戶端失敗的安全孔。

## 解決方案

修改您的 API 用戶端，強制使用者在使用 API 之前登入用戶端。

作為暫行解決方法，您可以變更 BPEAPIUser 及 TaskAPIUser 角色的對映。若要變更對映：

1. 在管理主控台中，按一下應用程式 → 企業應用程式 → **BPEContainer\_suffix**，然後在明細內容下，按一下安全角色與使用者/群組對映
2. 將 BPEAPIUser 角色從 All authenticated 變更為 Everyone，然後按一下確定。
3. 對於 TaskContainer\_suffix 及 TaskAPIUser 角色，重複步驟 2。
4. 在修改用戶端之後，您必須將這些角色變更回 All authenticated，以防止未經鑑別的使用者存取 API。

---

## 啓用商業程序編排器的追蹤功能

本文說明聯絡支援中心之前應該進行的處理。

### 啓用追蹤功能

「商業程序編排器」追蹤使用標準 WebSphere Process Server 追蹤機制。這必須用一般方式啓用。

追蹤規格如下所示：

```
com.ibm.bpe.*=all:com.ibm.task.*=all:com.ibm.ws.staffsupport.*=all
```

其中 `com.ibm.bpe.*=all` 會追蹤商業程序，而 `com.ibm.task.*=all` 會追蹤人力作業。人力作業的剩餘部分 (人員目錄提供者) 是透過 `com.ibm.ws.staffsupport` 追蹤。

### 傳送給支援中心的内容


啓用追蹤功能之後，請重建問題範例情況，然後提供下列檔案：

- WebSphere Application Server FFDC 日誌 (位於 `ffdc` 資料夾中)
- 下列日誌檔：
  - SystemOut.log
  - SystemErr.log
  - trace.log

在 Linux、UNIX 及 i5/OS 系統上，這些檔案位於 `profile_root/logs/server_name` 目錄。在 Windows 平台上，它們位於 `profile_root\logs\server_name` 目錄中。

如果您的問題範例情況導致許多記載，可能會以如 `SystemOut_07.10.01_11.00.51.log` 的名稱建立日誌的備份檔。您可以使用管理主控台來變更所建立的備份檔數目，以及日誌檔的大小。增加這兩個值以確保您擷取所有資料，可能是不錯的作法。

### 相關資訊

 WebSphere Process Server 疑難排解手冊

---

## 第 25 章 商業程序及人力作業疑難排解

使用此主題來解決與商業程序及人力作業相關的問題。

下列作業著重在疑難排解於執行商業程序或作業期間發生的問題。

---

### 商業程序及人力作業應用程式的安裝疑難排解

在 ND 環境中安裝內含商業程序、人力作業或這兩者的應用程式時，會在部署管理程式 SystemErr.log 檔中收到異常。

#### 症狀

在 ND 環境中安裝內含商業程序、人力作業或這兩者的應用程式時，會在部署管理程式 SystemErr.log 檔中發現下列異常：

```
SystemErr R com.ibm.ws.management.commands.sib.SIBAdminCommandException:
CWSJA0012E: Messaging engine not found.
at com.ibm.ws.management.commands.sib.SIBAdminCommandHelper.createDestination
(SIBAdminCommandHelper.java:787)
at com.ibm.ws.management.commands.sib.CreateSIBDestinationCommand.afterStepsExecuted
(CreateSIBDestinationCommand.java:459)
at com.ibm.websphere.management.cmdframework.provider.AbstractTaskCommand.execute
(AbstractTaskCommand.java:547)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdminHelper.call(SIBAdminHelper.java:136)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdminHelper.createSIBDestination
(SIBAdminHelper.java:112)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdmin.createDestination(SIBAdmin.java:327)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.createDestination
(SIBDestinationTask.java:263)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.preInstallModule
(SIBDestinationTask.java:71)
at com.ibm.ws.sca.internal.deployment.SCATaskBase.installModule(SCATaskBase.java:57)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.processArtifacts
(SIBDestinationTask.java:228)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.install
(SIBDestinationTask.java:287)
at com.ibm.ws.sca.internal.deployment.SCAInstallTask.performInstallTasks
(SCAInstallTask.java:116)
at com.ibm.ws.sca.internal.deployment.SCAInstallTask.performTask
(SCAInstallTask.java:61)
at com.ibm.ws.management.application.SchedulerImpl.run(SchedulerImpl.java:253)
at java.lang.Thread.run(Thread.java:568)
```

#### 原因

遺漏 "SCA.SYSTEM.cellName.Bus" 匯流排的匯流排成員。

#### 解決方案

在管理主控台中，按一下服務整合 → 匯流排 → **SCA.SYSTEM.cellName.Bus**。在「拓撲」區段中，按一下匯流排成員。請將想要安裝商業程序或人力作業應用程式的伺服器或叢集新增為匯流排成員，然後重新啟動受影響的伺服器或叢集，再試著重新安裝應用程式。

---

## 商業程序的執行疑難排解

本文說明執行一般商業程序時，所遇問題的解決方案。

您可以在「商業程序編排器瀏覽器」的 IBM 技術支援頁面中，搜尋錯誤訊息碼。

1. 在錯誤頁面上，按一下 **搜尋其他資訊** 鏈結。這會開始在 IBM 技術支援站台上搜尋錯誤碼。此站台只提供英文版的資訊。
2. 將錯誤頁面上顯示的錯誤訊息碼複製至剪貼簿。錯誤碼的格式為 CWWBcnnnc，其中每個 c 都是一個字元，而 nnnn 是 4 位數數字。請跳至 WebSphere Process Server 技術支援頁面。
3. 將錯誤碼貼入 **Additional search terms** 欄位，並按一下 **Go**。

下列主題會提供特定問題的解決方案。

### 停止內含微流程的應用程式時發生 **ClassCastException**

SystemOut.log 檔包含的 **ClassCastException** 異常，大約是在內含微流程的應用程式已停止時發生。

#### 原因

停止應用程式時，會從類別路徑中移除 EAR 檔中所含的類別。不過，需要這些類別的微流程實例可能仍然在執行中。

#### 解決方案

請執行下列動作：

1. 先停止微流程程序範本。從現在開始，就無法再從該範本中啟動新的微流程實例。
2. 至少等待微流程執行的持續時間上限，讓所有執行中實例都可以完成。
3. 停止應用程式。

### 呼叫 **processMessage** 方法期間發生非預期的異常 (訊息：CNTR0020E)

商業程序儲存器已停止，而且用戶端無法連接至伺服器。

#### 解決方案

驗證商業程序儲存器是否執行中。

### **XPath** 查詢從陣列中傳回非預期的值

使用 XPath 查詢存取陣列中的成員會傳回非預期的值。

#### 原因

此問題的一般原因是假設陣列中的第一個元素的索引值為零。但在陣列中進行 XPath 查詢時，第一個元素的索引值為一。

#### 解決方案

檢查在陣列中從第一個元素開始使用的索引值。

## 活動因未處理的錯誤而停止 (訊息：CWWBE0057I)

系統日誌包含 CWWBE0057I 訊息時，程序會處於「執行中」狀態，但是無法繼續導覽現行路徑。

### 原因

如果發生下列所有狀況，則呼叫活動、行內人力作業及 Java 片段都會進入已停止狀態：

- 活動引發錯誤
- 含括範圍中未處理錯誤
- 活動的 `continueOnError` 屬性設為 `no`

### 解決方案

此問題的解決方案需要兩個層次的動作：

1. 管理者必須手動修復已停止的活動實例。例如，若要強制完成或強制重試已停止的活動實例。
2. 必須探索失敗原因。在部分情況下，失敗是由模型中必須更正的建模錯誤所造成。

第 372 頁的『管理商業程序的生命週期』

呼叫可以啟動程序的「商業程序編排器」API 方法時，就會有程序實例。在程序實例的所有活動都處於結束狀態之前，會繼續導覽程序實例。您可以對程序實例採取各種動作，以管理它的生命週期。

第 383 頁的『修復活動』

長時間執行的程序可以包含也會長時間執行的活動。這些活動可能會發現未捕捉到的錯誤，並進入已停止狀態。而執行中狀態的活動也會像「沒有回應」。在這兩種情況下，程序管理者可以用多種方式來處理活動，以便可以繼續導覽程序。

## 未補償微流程

微流程已經呼叫服務，且程序失敗，但是未呼叫復原服務。

### 解決方案

必須符合各種條件才能觸發微流程的補償作業。請檢查下列各項：

1. 登入「商業程序編排器瀏覽器」，並按一下**失敗補償**，以檢查補償服務是否失敗且需要修復。
2. 只有在回復微流程的交易時，才會觸發微流程的補償作業。請檢查是否為這個狀況。
3. 微流程的 `compensationSphere` 屬性必須設為 `required`。
4. 只有在對應的轉遞服務未參與微流程的交易時，才會執行補償服務。請確定轉遞服務未參與導覽交易，例如，在程序元件的參照上，將「服務元件架構 (SCA)」限定元 `suspendTransaction` 設為 `True`。

## 長時間執行的程序似乎已停止

長時間執行的程序處於執行中狀態，但是似乎未予處理。

## 原因

這類行為的原因可能有數種：

1. 導覽訊息重試過太多次，而且已移至「保留」或「存放」佇列。
2. 來自「服務元件架構 (SCA)」基礎架構的回覆訊息反覆失敗。
3. 程序正在等待事件、逾時或是等待傳回長時間執行的呼叫或作業。
4. 程序中的活動處於已停止狀態。

## 解決方案

上述每個原因都需要不同的更正動作：

1. 執行查詢及重新播放失敗的訊息 (使用管理主控台)。
2. 檢查管理主控台的失敗事件管理視圖中是否有任何失敗事件。
  - 如果有任何來自「服務元件架構 (SCA)」回覆訊息的失敗事件，請重新啟動訊息。
  - 否則，強制完成或強制重試長時間執行的活動。
3. 檢查是否有活動處於已停止狀態，並且修復這些活動。如果系統日誌包含 CWWBE0057I 訊息，則可能也需要更正模型，如訊息：CWWBE0057I 中所述。

## 呼叫另一個 EAR 檔中的同步子程序失敗

如果長時間執行的程序同步呼叫另一個程序，而且子程序位於另一個企業保存檔 (EAR) 中，則子程序呼叫會失敗。

產生的異常的範例：

```
com.ibm.ws.sca.internal.ejb.util.EJBStubAdapter com.ibm.ws.sca.internal.ejb.util.EJBStubAdapter#003
Exception:
java.rmi.AccessException: CORBA NO_PERMISSION 0x49424307 No; nested exception is:
org.omg.CORBA.NO_PERMISSION: The WSCredential does not contain a forwardable token.
Please enable Identity Assertion for this scenario.
vmcid: 0x49424000 minor code: 307 completed: No
at com.ibm.CORBA.iiop.UtilDelegateImpl.mapSystemException(UtilDelegateImpl.java:202)
at javax.rmi.CORBA.Util.mapSystemException(Util.java:84)
```

## 原因

因為子程序呼叫導致遠端 EJB 方法呼叫，所以呼叫另一個 EAR 檔中的同步子程序時，必須啟用 Common Secure Interoperability 第 2 版 (CSIv2) 身分主張。

## 解決方案

配置 CSIv2 入埠鑑別及 CSIv2 離埠鑑別。

## 執行期間發生非預期的異常 (訊息：CWWBA0010E)

佇列管理程式未執行，或「商業程序編排器」配置包含錯誤的資料庫密碼。

## 解決方案

請檢查下列各項：

1. 如果 systemout.log 檔包含 "javax.jms.JMSException: MQJMS2005: failed to create MQQueueManager"，請啟動佇列管理程式。

2. 確定儲存在「商業程序編排器」配置中的資料庫管理者密碼，符合資料庫中設定的資料庫管理者密碼。

### 事件不明 (訊息：CWWBE0037E)

嘗試將事件傳送給程序實例或啟動新的程序實例，會導致「CWWBE0037E：事件不明。」異常。

#### 原因

此錯誤的一般原因是將訊息傳送給程序，但是已導覽接收或挑選活動，因此這個程序實例無法再次使用該訊息。

#### 解決方案

若要更正此問題：

- 如果現有程序實例應該要使用此事件，則必須傳遞相互關係集值，而這些值符合尚未導覽對應的接收或挑選活動的現有程序實例。
- 如果事件應該要啟動新的程序實例，則相互關係集值不得符合現有的程序實例。

如需在商業程序中使用相互關係集的相關資訊，請參閱 [Technote 1171649](#)。

### 找不到或無法建立程序實例 (訊息：CWWBA0140E)

嘗試將事件傳送給程序實例，會導致 'CreateRejectedException' 訊息。

#### 原因

此錯誤的一般原因是訊息傳送給無法實例化新程序實例的接收或挑選活動，因為接收或挑選活動的 `createInstance` 屬性設為 `no`，而且與訊息一起傳遞、此活動所使用相互關係集的值不符合任何現有的程序實例。

#### 解決方案

若要更正此問題，則必須傳遞符合現有程序實例的相互關係集值。

如需在商業程序中使用相互關係集的相關資訊，請參閱 [BPEL 程序中的相互關係集](#)。

### 程序實例的失敗狀態不容許執行所要求的 `sendMessage` 動作 (訊息：CWWBE0126E)

嘗試將事件傳送給程序實例，會導致 'EngineProcessWrongStateException' 訊息。

#### 原因

這個錯誤的一般原因是將訊息傳送給接收或挑選活動，以建立新的程序實例，但是新的程序實例無法實例化。如果隨著訊息針對這個活動所使用的相互關係集而傳遞的值符合已處於失敗狀態的現有程序實例，就會發生這個狀況。

#### 解決方案

若要更正這個問題，您必須刪除現有的程序實例，或傳遞不符合現有程序實例的相互關係集值。如需在商業程序中使用相互關係集的相關資訊，請參閱 [BPEL 程序中的相互關係集](#)。



## Java 片段中的 Uninitialized 變數或 NullPointerException

在商業程序中使用 uninitialized 變數會導致不同的異常。

### 症狀

這類異常為：

- 在執行 Java 片段或 Java 表示式（讀取或操作變數的內容）期間，會擲出 NullPointerException。
- 在執行指派、呼叫、回覆或擲出活動期間，會擲出 BPEL 標準錯誤 "uninitializedVariable" (訊息 CWWBE0068E)。

### 原因

啓動程序時，商業程序中的所有變數都會具有空值，這樣不會預先初始設定變數。在 Java 片段或 Java 表示式中使用 uninitialized 變數，會導致 NullPointerException。

### 解決方案

變數必須先起始設定才能使用。這是透過指派活動完成，例如，變數需要出現在指派的 to-spec 上，變數也可以在 Java 片段內起始設定。

## 標準錯誤異常 "missingReply" (訊息：CWWBE0071E)

微流程或長時間執行程序的執行導致 BPEL 標準錯誤 "missingReply" (訊息：CWWBE0071E)，或者在系統日誌或 SystemOut.log 檔中發現此錯誤。

### 原因

雙向作業必須傳送回覆。如果程序結束，而未導覽回覆活動，則會產生此錯誤。在下列任一情況下會發生此錯誤：

- 跳過回覆活動。
- 發生錯誤，而且對應的錯誤處理程式未含回覆活動。
- 發生錯誤，而且沒有對應的錯誤處理程式。

### 解決方案

更正模型，以確保回覆活動一律是在程序結束之前執行。

## 序列化平行路徑

流程活動內有兩個以上的平行呼叫活動，但是呼叫活動卻循序執行。

### 解決方案

- 若要達到實際平行化，每個路徑都必須位在不同交易中。請將所有平行呼叫活動的 'transactional behavior' 屬性設為 'commit before' 或 'requires own'。
- 如果您是使用 Derby 或 Oracle 作為資料庫系統，則程序引擎會將執行平行路徑序列化。您無法變更此行為。

## 將巢狀資料物件複製至另一個資料物件，會毀損來源物件的參照

資料物件 Father 包含另一個資料物件 Child。在 Java 片段或用戶端應用程式內，會在資料物件 Mother 的子結構中提取並設定包含 Child 的物件。而資料物件 Father 的 Child 參照會消失。

### 原因

Child 的參照從 Father 移至 Mother。

### 解決方案

如果在 Java 片段或用戶端應用程式中執行這類資料轉換作業，而且想要保留 Father 中的參照，請先複製資料物件，再將它指派給另一個物件。下列程式碼片段說明如何執行這項作業：

```
BOCopy copyService = (BOCopy)ServiceManager.INSTANCE.locateService
    ("com/ibm/websphere/bo/BOCopy");
DataObject Child = Father.get("Child");
DataObject BCopy = copyService.copy(Child);
Mother.set("Child", BCopy);
```

## CScope 無法使用

啟動微流程或執行長時間執行程序中的導覽步驟失敗，出現的確認訊息指出：「後置條件違規!(cscope != null)」。

### 原因

在特定情況下，程序引擎會使用補償服務，但是尚未啟用補償服務。

### 解決方案

啟用補償服務，如管理的 PDF 中所述。

---

## 使用與程序或作業相關的訊息

說明如何取得「商業程序編排器」訊息的相關資訊，這些訊息會寫入顯示畫面或日誌檔。

屬於「商業程序編排器」的訊息的前面會加上字首 CWWB (程序相關訊息) 或 CWTK (作業相關訊息)。這些訊息的格式為 *PrefixComponentNumberTypeCode*。類型碼可以是：

- I** 參考訊息
- W** 警告訊息
- E** 錯誤訊息

在執行程序及作業時，訊息會顯示在「商業程序編排器瀏覽器」中，或是新增至 SystemOut.log 檔及追蹤資料中。如果這些檔案中提供的訊息文字不足以協助您解決問題，則可以使用 WebSphere Application Server 症狀資料庫來尋找其他資訊。若要檢視「商業程序編排器」訊息，請使用 WebSphere 日誌分析器來檢查 activity.log 檔。

1. 啟動 WebSphere 日誌分析器。

執行下列其中一個 Script：

- 在 Windows 系統上：*install\_root/bin/waslogbr.bat*
  - 在 Linux、UNIX 及 i5/OS 系統上，*install\_root/bin/waslogbr.sh*
2. 選擇性的: 按一下**檔案** → **更新資料庫** → **WebSphere Application Server 症狀資料庫**，以檢查最新版的症狀資料庫。
  3. 選擇性的: 載入活動日誌。
    - a. 選取活動日誌檔。
      - *profile\_root/profiles/profile\_name/logs/activity.log* 檔 (Windows 系統)
      - *profile\_root/profiles/profile\_name/logs/activity.log* 檔 (Linux、UNIX 及 i5/OS 系統)
    - b. 按一下**開啓**。

---

## 商業程序及人力作業的管理疑難排解

此文章說明如何解決一些商業程序及人力作業的一般問題。

下列資訊可以協助您對商業程序及人力作業的問題除錯。

- 如果嘗試在商業程序應用程式仍然有程序實例時將其停止，則管理主控台會停止回應。嘗試停止應用程式之前，必須先停止商業程序，以便不再建立新的實例，並執行下列其中一項：

- 等待所有現有程序實例依序結束。
- 終止並刪除所有程序實例。

只有這樣，才能安全地停止程序應用程式。如需防止此問題的相關資訊，請參閱 [Technote 1166009](#)。

- 如果嘗試在人力作業應用程式仍然有作業實例時將其停止，則管理主控台會停止回應。若要停止應用程式，則必須：

1. 停止人力作業，以便不再建立新的實例。
2. 執行下列動作之一：
  - 等待所有現有作業實例依序結束。
  - 終止並刪除所有作業實例。
3. 停止作業應用程式。

- 由呼叫作業啟動的長時間執行商業程序無法啟動。JSP Snippet 讓使用者可以使用呼叫作業。在下列範例中，使用同步呼叫型樣 `createAndCallTask`。在此情況下，長時間執行商業程序無法啟動：

```
HumanTaskManager htm = ...
TaskTemplate taskTemplate = htm.getTaskTemplate( "start the process" );
Task task = htm.createAndCallTask( taskTemplate.getTKID() );
while (task.getState() != TASK.TASK_STATE_FINISHED)
{
    Sleep(100);
}
```

長時間執行情序由數個交易組成，而且其呼叫樣式為非同步。因此，必須使用非同步呼叫型樣 `createAndStartTask` 來啟動它。

```
HumanTaskManager htm = ...
TaskTemplate taskTemplate = htm.getTaskTemplate( "start the process" );
Task task = htm.createAndStartTask( taskTemplate.getTKID() );
```

```
while (task.getState() != TASK.TASK_STATE_FINISHED)
{
    Sleep(100);
}
```

此外，JSP 部署描述子中的交易屬性也必須設定為 `NotSupported`。這確保不需交易即可執行程式碼 Snippet，而且 `createAndStartTask` 方法會開啓新的交易，以啓動程序實例。這個交易是在 `createAndStartTask` 方法返回，並可以看見訊息時才確定。

為已完成以外的狀態準備 "while" 迴圈是一種很好的作法。比方說，如果執行程序期間活動失敗，則結束狀態可能是 `TASK.TASK_STATE_FAILED`。

---

## 呈報電子郵件疑難排解

您可以使用這個資訊，解決與呈報電子郵件相關的問題。

當人力作業不是依預期進行時，會觸發呈報。呈報會建立工作項目。它也可以將受到呈報影響的電子郵件傳送給使用者。如果您有呈報電子郵件方面的問題，請使用這裡的資訊，協助您解決問題。

- 檢查 `SystemOut.log` 檔，以找出與人員指派或電子郵件位址相關的錯誤訊息。
- 如果 `SystemOut.log` 檔沒有包含任何相關訊息，請針對郵件階段作業伺服器啓用除錯模式。

在管理主控台中，按一下資源 → 郵件 → 郵件階段作業 → `HTMailSession_server`，然後選取 **啓用除錯模式** 勾選框。傳送呈報電子郵件時，除錯資訊會寫入至 `SystemOut.log` 檔案中。

- 如果您是使用虛擬成員管理程式作為人員目錄提供者，而且您有電子郵件位址方面的問題，請啓用 `Staff.Diagnosis` 自訂內容。
  1. 在管理主控台中，按一下應用程式伺服器 → `server_name`。
  2. 在商業整合之下，展開商業程序編排器，再按一下人力作業管理程式配置。
  3. 在配置標籤的其他內容之下，按一下自訂內容 → **Staff.Diagnosis**，然後在值欄位中，鍵入 `on`。

傳送呈報電子郵件時，人員指派的相關資訊會寫入至 `SystemOut.log` 檔案中。

- 檢查「人力作業管理程式」存放佇列是否包含訊息。
  1. 在管理主控台中，按一下應用程式伺服器 → `server_name`。
  2. 在商業整合之下，展開商業程序編排器，再按一下人力作業管理程式配置。
  3. 在執行時期標籤中，按一下重新播放存放佇列。存放佇列中的訊息會顯示在存放佇列訊息欄位中。

如果存放佇列包含訊息，請檢查伺服器的 `First Failure Data Capture` 目錄，以取得錯誤的相關資訊。

- 檢查自訂內容的值，以取得重設電子郵件的次數及傳送電子郵件的逾時。
  1. 在管理主控台中，按一下應用程式伺服器 → `server_name`。
  2. 在商業整合之下，展開商業程序編排器，再按一下人力作業管理程式配置。
  3. 在配置標籤的其他內容之下，按一下自訂內容。
  4. 檢查 `EscalationEmail.RetryTimeout` 及 `EscalationEmail.MaxRetries` 欄位的值。

### EscalationEmail.RetryTimeout

指定「人力作業管理程式」直到重新傳送失敗的電子郵件通知前要等待多久時間。這個欄位的預設值為 3600 秒。(一小時) 如果重試失敗，則每次重試失敗，重試逾時會動態地加倍。根據預設值，如果第一次重試失敗，則在兩小時之後會進行另一次重試。

### EscalationEmail.MaxRetries

指定「人力作業管理程式」嘗試重新傳送失敗的通知的次數。這個欄位的預設值為 4 次重試。如果這個欄位的值設定為 0，則不會重新傳送失敗的電子郵件通知。如果所有重試都失敗，則訊息會放入存放佇列中。對於「人力作業管理程式」，您可以在管理主控台的**執行時期**標籤中，看到存放佇列中的訊息。如果您重新播放訊息，這相當於第一次傳送電子郵件。

---

## 人員指派疑難排解

使用下列資訊來協助解決指派人員給授權角色時的相關問題。

本資訊涵蓋下列問題：

- 部署人員目錄提供者期間發生錯誤
- 人員目錄中的項目未反映在工作項目指派中
- 人員目錄的變更未立即反映在工作項目指派中
- 作業或程序實例發生非預期的人員指派
- 已停止人力作業
- 與人員指派相關的錯誤及警告訊息
- 群組工作項目及 "Group" 人員指派準則的問題
- 清除儲存的人員指派結果

您也可以[在技術支援搜尋頁面中搜尋其他資訊](#)。

### 部署人員目錄提供者期間發生錯誤

如果您是使用「輕量型目錄存取通訊協定 (LDAP)」人員目錄提供者，則部署可能失敗，因為提供者配置參數的值不正確。

- 請確定已設定所有必要參數。
- 若要將 baseDN 參數設為 LDAP 目錄樹狀結構的根目錄，請指定空字串；再將 baseDN 參數設為兩個單引號 (') 字元 (``)。請勿使用雙引號 (")。如果無法設定 baseDN 參數，則部署期間會發生 NullPointerException 異常。

### 人員目錄中的項目未反映在所指派的工作項目中

「人員查詢」可擷取的使用者 ID 數目上限由 Threshold 變數指定，此變數是在使用中的 XSL 轉換檔中定義。例如，針對 LDAP 人員目錄提供者使用的 XSL 轉換檔為 LDAPTransformation.xsl。在 Linux、UNIX 及 iOS 平台上，這個檔案位於 *install-root/ProcessChoreographer/Staff* 目錄中，在 Windows 平台上，則位於 *install-root/ProcessChoreographer\Staff* 目錄中。Threshold 預設值為 20。若要變更此值：

1. 使用您自己的 XSL 檔案版本，建立新的人員目錄提供者配置。
2. 根據您的需求，在 XSL 檔案中修改下列項目：

```
<xsl:variable name="Threshold">20</xsl:variable>
```

註：如果您指定很大的 Threshold 值，可能會導致降低效能。因此，請勿指定大於 100 的值。

### 人員目錄的變更不會立即反映在所指派的工作項目中

「商業程序編排器」會在執行時期資料庫中快取對人員目錄 (例如 LDAP 伺服器) 評估的人員指派結果。當人員目錄有變動時，不會立即反映在資料庫快取中。

管理手冊說明重新整理此快取的三種方式：

- **使用管理主控台來重新整理人員查詢結果。**如果您有重大的變更，需要重新整理幾乎所有人員查詢的結果，請使用此方法。
- **使用管理指令來重新整理人員查詢結果。**如果您使用 wsadmin 工具來撰寫管理 Script，或者只想要立即重新整理一部分的人員查詢結果，請使用此方法。
- **使用 refresh 常駐程式來重新整理人員查詢結果。**使用此方法來設定自動地定期重新整理所有過期的人員查詢結果。

註：上述方法都無法針對 Group 動詞來重新整理使用者的群組成員資格關聯。此群組成員資格會快取在使用者的登入階段作業中 (WebSphere 安全 LTPA 記號)，依預設兩個小時之後會過期。程序導覽所用的程序起始者 ID 的群組成員資格清單，永遠都不會重新整理。

### 作業或程序實例出現非預期指派的人員

如果您未替作業的某些角色定義「人員指派準則」，或者如果人員指派失敗或未傳回結果，則會指派預設的人員。這些預設值可能會導致非預期的使用者授權；例如，程序起始者可能取得程序管理者權限。此外，許多權限會由相依的構件所繼承。例如，程序管理者也可能變成所有行內作業的管理者。

下表說明哪些情況適用哪些預設值：

表 91. 商業程序的角色

商業程序的角色	如果程序模型中未定義角色 ...	如果程序模型中已定義角色，但「人員指派」失敗或未傳回適當的結果 ...
程序管理者	程序起始者變成程序管理者	發生下列異常，且未啟動程序： EngineAdministratorCannotBeResolvedException
程序讀者	沒有讀者	沒有讀者

表 92. 行內人力作業及其呈報的角色

行內人力作業及其呈報的角色	如果作業模型中未定義角色 ...	如果作業模型中已定義角色，但「人員指派」失敗或未傳回適當的結果 ...
作業管理者	只套用繼承關係	只套用繼承關係
作業可能的實例建立者	每個人都變成可能的實例建立者	每個人都變成可能的實例建立者
作業可能的起始者	每個人都變成可能的起始者	每個人都變成可能的起始者
作業可能的擁有者	每個人都變成可能的擁有者	管理者變成可能的擁有者
作業編輯者	沒有編輯者	沒有編輯者
作業讀者	只套用繼承關係	只套用繼承關係
呈報接收者	管理者變成呈報接收者	管理者變成呈報接收者



下列繼承規則適用於行內作業：

- 程序管理者變成所有行內作業、其子作業、後續作業及呈報的管理者。
- 程序讀者變成所有行內作業、其子作業、後續作業及呈報的讀者。
- 作業管理者變成呈報所有子作業、後續作業及所有這些作業的管理者。
- 作業讀者變成呈報所有子作業、後續作業及所有這些作業的讀者。
- 任何作業角色的成員變成呈報此作業、子作業及後續作業的讀者。
- 呈報接收者變成已呈報作業的讀者。

表 93. 獨立式人力作業及其呈報的角色

獨立式人力作業及其呈報的角色	如果作業模型中未定義角色 ...	如果作業模型中已定義角色，但人員指派失敗或未傳回正確的結果 ...
作業管理者	發送者變成管理者	未啓動作業
作業可能的實例建立者	每個人都變成可能的實例建立者	每個人都變成可能的實例建立者
作業可能的起始者	發送者變成可能的起始者	未啓動作業
可能的擁有者	每個人都變成可能的擁有者	管理者變成可能的擁有者
編輯者	沒有編輯者	沒有編輯者
讀者	只套用繼承關係	只套用繼承關係
呈報接收者	管理者變成呈報接收者	管理者變成呈報接收者

下列繼承規則適用於獨立式作業：

- 作業管理者變成呈報所有子作業、後續作業及所有這些作業的管理者。
- 作業讀者變成呈報所有子作業、後續作業及所有這些作業的讀者。
- 任何作業角色的成員變成呈報此作業、子作業及後續作業的讀者。
- 呈報接收者變成已呈報作業的讀者。

**註:** 當使用「商業流程管理程式」API 呼叫方法時，BPESystemAdministrator 角色的成員具有管理者權限，而 BPESystemMonitor 角色的成員具有讀者權限。

**註:** 當使用「人力作業管理程式」API 呼叫方法時，TaskSystemAdministrator 角色的成員具有管理者權限，而 TaskSystemMonitor 角色的成員具有讀者權限。

### 已停止人力作業

如果您遇到下列一個以上的問題：

- 即使已順利開始導覽商業程序，也無法要求人力作業。
- SystemOut.log 檔案包含下列訊息：CWWB0057I：程序 'MyProcess' 的活動 'MyStaffActivity' 因一個未處理的失敗而停止...

這些問題指出可能未啓用 WebSphere Application Server 安全。使用人員權限的人力作業和程序需要已啓用安全原則，且已配置使用者登錄。請採行下列步驟：

1. 檢查是否已啓用 WebSphere 安全原則。在管理主控台中，跳至安全 → 廣域安全，並確定已選取啓用廣域安全勾選框。
2. 檢查是否已配置使用者登錄。在管理主控台中，跳至安全 → 使用者登錄，並檢查作用中的使用者登錄屬性。
3. 重新啓動活動 (如果已停止)。



## 和指派人員相關的錯誤和警告訊息

在指派人員期間存取人員目錄時，可能發生一些常見的錯誤。若要查看這些錯誤的詳細資料，您可以使用下列追蹤設定來啟用追蹤：`com.ibm.bpe.*=all:com.ibm.task.*=all:com.ibm.ws.staffsupport.ws.*=all`

下列常見的錯誤情況會以警告或錯誤訊息來表示：

- `trace.log` 檔案中的 `Could not connect to LDAP server`，表示無法連接 LDAP 伺服器。請檢查網路設定、您使用的人員目錄提供者的配置（尤其是提供者 URL），並驗證您的 LDAP 伺服器是否需要 SSL 連線。
- `System.out` 或 `System.err` 檔案中的 `javax.xml.transform.TransformerException: org.xml.sax.SAXParseException: Element type "xsl:template" must be followed by either attribute specifications, ">" or "/>"`，表示無法讀取 `LDAPTransformation.xsl` 檔案。請檢查「人員指派配置」和已配置的 XSLT 檔案是否有錯誤。
- `trace.log` 檔案中的 `LDAP object not found. dn: uid=unknown,cn=users,dc=ibm,dc=com [LDAP: error code 32 - No Such Object]`，表示找不到 LDAP 項目。請檢查作業模型的「人員指派準則」（動詞）參數及 LDAP 目錄內容在作業模型中是否有不符情形。
- `trace.log` 檔案中的 `Requested attribute "uid" not found in: uid=test222,cn=users,dc=ibm,dc=com`，表示在所查詢的 LDAP 物件中找不到屬性。請檢查作業模型的「人員指派準則」（動詞）參數及 LDAP 目錄內容在作業模型中是否有不符情形。也要檢查人員指派配置的 XSLT 檔是否有錯誤。

## 群組工作項目及 "Group" 人員指派準則的問題

如果您是使用 Group 人員指派準則，則可能會發生下列狀況：

- 即使指定群組名稱，群組成員仍然未獲授權：
  - 使用「本端 OS」登錄來管理 WebSphere 安全原則時，請指定群組簡短名稱，使用 LDAP 登錄時，請使用群組 dn。
  - 請確定您知道群組名稱區分大小寫。

這種情形可能是因為您已配置 WebSphere 安全使用 LDAP 使用者登錄，且選取**權限不區分大小寫**選項。如果這樣，請取消選取該選項，或以全部大寫來指定 LDAP 群組 dn。

- 群組成員資格的變更未立即反映在權限中。當受到影響的使用者仍在已登入狀態時，就可能發生這種情形。使用者的群組成員資格會快取在其登入階段作業中，且依預設在兩個小時之後到期。您可以等待登入階段作業到期（預設是兩小時），或重新啟動應用程式伺服器。「人力作業管理程式」提供的重新整理方法不適用於這個人員指派準則。請注意，程序起始者的群組成員資格清單永遠都不會重新整理。

## 清除儲存的人員指派結果

人員指派結果儲存在資料庫中。所有儲存的人員指派結果會隨著人員指派重新整理而變更。如果包含作業實例（導致人員指派結果的計算）的作業範本被刪除，則也會刪除儲存的人員指派結果。不過，如果只有使用儲存的人員指派結果的作業實例被刪除，則不會刪除儲存的人員指派結果。

若要避免資料庫中出現大量的儲存及不必要的人員指派結果，請在作業範本的環境定義中採取下列步驟：

1. 請評量您的人員指派準則定義是否導致共用或不共用的人員指派結果。
2. 如果發生不共用的指派結果，請考量對人員指派結果進行清除程序。讓清除間隔根據預期的作業實例數，以及每個清除間隔的不共用人員指派結果數。如需如何套用 Script 型清除程序的相關資訊，請參閱使用管理指令來移除未用的人員查詢結果。

### 相關參考

第 541 頁的『已啟用代理時，作業儲存器無法啟動』

您已建立新的「商業程序編排器」配置，並已啟用代理，或者已針對現有配置啟用代理。

---

## 疑難排解商業程序編排器瀏覽器

使用此項，可以解決與「商業程序編排器瀏覽器」相關的問題。

使用下列資訊，可以解決與「商業程序編排器瀏覽器」相關的問題。

- 如果您嘗試利用瀏覽器來存取「商業程序編排器瀏覽器」，但是收到錯誤訊息，而非登入頁面，請嘗試下列動作：
  - 使用管理主控台，確定 Web 用戶端應用程式 `BPCExplorer_node_name_server_name` 實際上部署並執行在伺服器上。
  - 在管理主控台中，於應用程式頁面的「檢視部署描述子」下方，驗證環境定義根目錄是否為您在設定「商業程序編排器瀏覽器」時使用的環境定義根目錄。
- 如果在使用「商業程序編排器瀏覽器」時收得錯誤訊息，請按一下錯誤頁面上的**搜尋相關資訊**鏈結。這會開始在 IBM 技術支援站台上搜尋錯誤碼。此站台只提供英文版的資訊。請將「商業程序編排器瀏覽器錯誤」頁面上顯示的錯誤訊息複製至剪貼簿。錯誤碼的格式為 `CWWBcnnnc`，其中每個 `c` 都是一個字元，而 `nnnn` 是 4 位數數字。請跳至 WebSphere Process Server 技術支援頁面。將錯誤碼貼入 **Additional search terms** 欄位，並按一下 **Go**。
- 如果您收到 `StandardFaultException`，標準錯誤為 `missingReply` (訊息 `CWWBE0071E`)，則這顯示程序模型有問題。如需解決此問題的相關資訊，請參閱第 552 頁的『商業程序及人力作業的管理疑難排解』。
- 如果您可以登入「商業程序編排器瀏覽器」，但並未顯示部分項目，或如果未啟用某些動作，則這指出授權方面的問題。

這個問題的可能解決方案包括：

- 使用管理主控台來確保已啟用 WebSphere 管理安全。
- 檢查您是否已使用正確身分 ID 登入「商業程序編排器瀏覽器」。如果您利用不是程序或作業管理者的使用者 ID 來登入，將看不到或不啟用所有管理視圖及選項。
- 使用 WebSphere Integration Developer，檢查或修改在商業程序中定義的權限設定。
- 錯誤訊息 `CWWBU0001E`：「呼叫 `BFMConnection` 函數時發生通訊錯誤」或「呼叫 `HTMConnection` 函數時發生通訊錯誤」。這個錯誤可以指出商業程序儲存器或人力作業儲存器已停止，因此用戶端無法連接至伺服器。驗證商業程序儲存器及人力作業儲存器是否執行中且可存取。巢狀異常可能包含問題的進一步明細。

- 錯誤訊息 WWBU0024E：無法建立與本端商業程序 EJB 的連線，原因為「命名異常」。如果使用者嘗試在商業程序儲存器未執行時登入，則會擲出這個錯誤。驗證應用程式 `BPEContainer_InstallScope` 是否在執行中，其中 `InstallScope` 是 `cluster_name` 或 `hostname_servername`。

#### 相關工作

第 546 頁的『商業程序的執行疑難排解』

本文說明執行一般商業程序時，所遇問題的解決方案。

## 商業程序編排器觀察程式疑難排解

如果商業程序編排器觀察程式發生問題，請參閱本主題中的資訊。

### 症狀

「商業程序編排器觀察程式」的「歡迎使用」頁面上未顯示事件。

### 原因及解決方案

「商業程序編排器觀察程式」資料庫未包含任何事件，或事件尚未轉換。

原因	解決方案
CEI 記載未啓用。	請確定商業程序儲存器的 CEI 記載已啓用。請參閱啓用商業程序編排器觀察程式的記載，以啓用 CEI 記載。
共用事件基礎架構事件伺服器或商業程序編排器事件收集程式不在執行中。	使用管理主控台，檢查共用事件基礎架構事件伺服器及商業程序編排器事件收集程式是否執行。
商業程序的事件監視已停用。	請確定在 WebSphere Integration Developer 程序模型的定義中已啓用事件監視。如需如何啓用商業程序事件監視的建議，請參閱 WebSphere Integration Developer 資訊中心。
事件轉換程式未觸發。	請重新啓動商業程序編排器事件收集程式，以觸發事件轉換程式。
商業程序編排器事件收集程式的不適當配置設定會使資料無法呈現在觀察程式中。	請呼叫 <code>setupEventCollector</code> 配置 Script，以變更 <code>BPCEventTransformerEventCount</code> 、 <code>BPCEventTransformerMaxWaitTime</code> 及 <code>BPCEventTransformerToleranceTime</code> 的商業程序編排器事件收集程式配置設定。如需變更商業程序編排器事件收集程式配置設定，請參閱變更配置參數。

### 如果問題依然存在

- 請檢查伺服器的系統日誌檔 `SystemOut.log` 中的錯誤訊息。
- 請檢查商業程序編排器事件收集程式及商業程序編排器觀察程式的部署及配置。若要檢查配置設定，請呼叫 `setupEventCollector` 及 `setupObserver` 配置 Script。如需如何變更「商業程序編排器」事件收集程式配置設定的進一步資訊，請參閱第 226 頁的『變更商業程序編排器觀察程式的配置參數』。

- 在管理主控台中啓用觀察程式元件的追蹤功能：記載與追蹤 → server1 → 診斷追蹤服務 → 變更日誌明細層次。將 com.ibm.bpe.observer.\* 的明細層次設為 all，並重新啓動 BPCECollector 及 BPCObserver 應用程式。

## 使用程序相關及作業相關審核追蹤資訊

說明商業程序及人力作業的事件類型及資料庫結構。

必須啓用商業程序儲存器、作業儲存器或這兩者的記載功能。

如果已啓用記載，則對於商業程序或人力作業的每一個導覽步驟，資訊都會寫入至審核日誌或 共用事件基礎架構 (CEI) 日誌中。如需 CEI 的相關資訊，請參閱監視 *WebSphere Process Server* PDF。下列主題說明商業程序及人力作業的事件類型及資料庫結構。

### 商業程序的審核事件類型

此處說明在處理商業程序期間可寫入審核日誌的事件類型。

對於要記載的事件，必須符合下列條件：

- 商業程序儲存器已啓用對應的審核日誌記載類型
- 程序模型中對應的實體必須已啓用事件

下表列出商業程序執行時可發生的審核事件碼。

表 94. 程序實例事件

審核事件	事件碼
PROCESS_STARTED	21000
PROCESS_SUSPENDED	21001
PROCESS_RESUMED	21002
PROCESS_COMPLETED	21004
PROCESS_TERMINATED	21005
PROCESS_RESTARTED	21019
PROCESS_DELETED	21020
PROCESS_FAILED	42001
PROCESS_COMPENSATING	42003
PROCESS_COMPENSATED	42004
PROCESS_TERMINATING	42009
PROCESS_FAILING	42010
PROCESS_CORRELATION_SET_INITIALIZED	42027
PROCESS_COMPENSATION_INDOUBT	42030
PROCESS_WORKITEM_DELETED	42041
PROCESS_WORKITEM_CREATED	42042
PROCESS_COMPENSATION_FAILED	42046
PROCESS_EVENT_RECEIVED	42047
PROCESS_EVENT_ESCALATED	42049
PROCESS_WORKITEM_TRANSFERRED	42056

表 94. 程序實例事件 (繼續)

審核事件	事件碼
PROCESS_PARTNER_CHANGED	42058

表 95. 活動事件

審核事件	事件碼
ACTIVITY_READY	21006
ACTIVITY_STARTED	21007
ACTIVITY_COMPLETED	21011
ACTIVITY_CLAIM_CANCELED	21021
ACTIVITY_CLAIMED	21022
ACTIVITY_TERMINATED	21027
ACTIVITY_FAILED	21080
ACTIVITY_EXPIRED	21081
ACTIVITY_LOOPED	42002
ACTIVITY_SKIPPED	42005
ACTIVITY_TERMINATING	42008
ACTIVITY_FAILING	42011
ACTIVITY_OUTPUT_MESSAGE_SET	42012
ACTIVITY_FAULT_MESSAGE_SET	42013
ACTIVITY_STOPPED	42015
ACTIVITY_FORCE_RETRIED	42031
ACTIVITY_FORCE_COMPLETED	42032
ACTIVITY_UNDO_STARTED	42033
ACTIVITY_UNDO_SKIPPED	42034
ACTIVITY_UNDO_COMPLETED	42035
ACTIVITY_MESSAGE_RECEIVED	42036
ACTIVITY_LOOP_CONDITION_TRUE	42037
ACTIVITY_LOOP_CONDITION_FALSE	42038
ACTIVITY_WORKITEM_DELETED	42039
ACTIVITY_WORKITEM_CREATED	42040
ACTIVITY_ESCALATED	42050
ACTIVITY_WORKITEM_REFRESHED	42054
ACTIVITY_WORKITEM_TRANSFERRED	42055
ACTIVITY_PARALLEL_BRANCHES_STARTED	42057

表 96. 變數相關事件

審核事件	事件碼
VARIABLE_UPDATED	21090

表 97. 控制鏈結事件

審核事件	事件碼
LINK_EVALUATED_TO_TRUE	21034
LINK_EVALUATED_TO_FALSE	42000

表 98. 程序範本事件

審核事件	事件碼
PROCESS_INSTALLED	42006
PROCESS_UNINSTALLED	42007

表 99. 範圍實例事件

審核事件	事件碼
SCOPE_STARTED	42020
SCOPE_SKIPPED	42021
SCOPE_FAILED	42022
SCOPE_FAILING	42023
SCOPE_TERMINATED	42024
SCOPE_COMPLETED	42026
SCOPE_COMPENSATING	42043
SCOPE_COMPENSATED	42044
SCOPE_COMPENSATION_FAILED	42045
SCOPE_EVENT_RECEIVED	42048
SCOPE_EVENT_ESCALATED	42051

## 人力作業的審核事件類型

此處說明在處理人力作業期間可寫入審核日誌的事件類型。

對於要記載的事件，必須符合下列條件：

- 人力作業儲存器已啓用對應的審核日誌記載類型
- 作業模型中對應的實體必須已啓用事件

下表列出人力作業執行時可發生的審核事件碼。

表 100. 作業實例事件

審核事件	事件碼
TASK_CREATED	51001
TASK_DELETED	51002
TASK_STARTED	51003
TASK_COMPLETED	51004
TASK_CLAIM_CANCELLED	51005
TASK_CLAIMED	51006
TASK_TERMINATED	51007
TASK_FAILED	51008

表 100. 作業實例事件 (繼續)

審核事件	事件碼
TASK_EXPIRED	51009
TASK_WAITING_FOR_SUBTASK	51010
TASK_SUBTASKS_COMPLETED	51011
TASK_RESTARTED	51012
TASK_SUSPENDED	51013
TASK_RESUMED	51014
TASK_COMPLETED_WITH_FOLLOW_ON	51015
TASK_UPDATED	51101
TASK_OUTPUT_MESSAGE_UPDATED	51103
TASK_FAULT_MESSAGE_UPDATED	51104
TASK_WORKITEM_DELETED	51201
TASK_WORKITEM_CREATED	51202
TASK_WORKITEM_TRANSFERRED	51204
TASK_WORKITEM_REFRESHED	51205

表 101. 作業範本事件

審核事件	事件碼
TASK_TEMPLATE_INSTALLED	52001
TASK_TEMPLATE_UNINSTALLED	52002

表 102. 呈報實例事件

審核事件	事件碼
ESCALATION_FIRED	53001
ESCALATION_WORKITEM_DELETED	53201
ESCALATION_WORKITEM_CREATED	53202
ESCALATION_WORKITEM_TRANSFERRED	53204
ESCALATION_WORKITEM_REFRESHED	53205

## 商業程序的審核追蹤資料庫視圖的結構

AUDIT\_LOG\_B 資料庫視圖提供商業程序的審核日誌相關資訊。

若要讀取審核追蹤的內容，請使用 SQL 或任何支援讀取資料庫表格及視圖的其他管理工具。

審核事件與程序實體有關。審核事件類型取決於事件參照的實體。審核事件類型包括：

- 程序範本事件 (PTE)
- 程序實例事件 (PIE)
- 活動實例事件 (AIE)
- 變數相關事件 (VAR)
- 控制鏈結事件 (CLE)
- 範圍相關事件 (SIE)。



如需審核事件類型碼的清單，請參閱第 560 頁的『商業程序的審核事件類型』。

下表說明 AUDIT\_LOG\_B 審核追蹤視圖的結構。其中列出直欄的名稱、事件類型並提供直欄的簡要說明。

行內作業在 AUDIT\_LOG\_B 審核追蹤視圖中記載，而不是在 TASK\_LOG 審核追蹤視圖中記載。例如，要求行內參與作業會導致 ACTIVITY\_CLAIMED 事件；而不會產生作業相關的事件。

表 103. AUDIT\_LOG\_B 審核追蹤視圖的結構

名稱	PTE	PIE	AIE	VAR	CLE	SIE	說明
AIID			x				與現行事件相關的活動實例 ID。
ALID	x	x	x	x	x	x	審核日誌項目的 ID。
EVENT_TIME	x	x	x	x	x	x	事件發生時的時間戳記 (格式為「世界標準時間 (UTC)」)。
EVENT_TIME_UTC	x	x	x	x	x	x	事件發生時的時間戳記 (格式為「世界標準時間 (UTC)」)。
AUDIT_EVENT	x	x	x	x	x	x	發生的事件類型。
PTID	x	x	x	x	x	x	與現行事件相關的程序的程序範本 ID。
PIID		x	x	x	x	x	與現行事件相關的程序實例的程序實例 ID。
VARIABLE_NAME				x			與現行事件相關的變數名稱。
SIID						x	與事件相關的範圍實例 ID。
PROCESS_TEMPL_NAME	x	x	x	x	x	x	與現行事件相關的程序範本的程序範本名稱。
TOP_LEVEL_PIID		x	x	x	x	x	與現行事件相關的最上層程序的 ID。
PARENT_PIID		x	x	x	x	x	母項程序的程序實例 ID，如果沒有母項程序，則為 null。
VALID_FROM	x	x	x	x	x	x	與現行事件相關的程序範本的有效起始日期。
VALID_FROM_UTC	x	x	x	x	x	x	與現行事件相關的程序範本的有效起始日期 (格式為「世界標準時間 (UTC)」)。
ATID			x				與現行事件相關的活動範本的 ID。
ACTIVITY_NAME			x			x	發生事件的活動名稱。

表 103. AUDIT\_LOG\_B 審核追蹤視圖的結構 (繼續)

名稱	PTE	PIE	AIE	VAR	CLE	SIE	說明
ACTIVITY_KIND			x				<p>發生事件的活動類型。可能的值如下：</p> <p>KIND_EMPTY 3            KIND_INVOKE 21            KIND_RECEIVE 23            KIND_REPLY 24            KIND_THROW 25            KIND_TERMINATE 26            KIND_WAIT 27            KIND_COMPENSATE 29            KIND_SEQUENCE 30            KIND_SWITCH 32            KIND_WHILE 34            KIND_PICK 36            KIND_FLOW 38            KIND_SCRIPT 42            KIND_STAFF 43            KIND_ASSIGN 44            KIND_CUSTOM 45            KIND_RETHROW 46            KIND_FOR_EACH_SERIAL 47            KIND_FOR_EACH_PARALLEL 49</p> <p>這些是針對 ActivityInstanceData.KIND_* 定義的常數</p>
ACTIVITY_STATE			x				<p>與事件相關的活動狀態。可能的值如下：</p> <p>STATE_INACTIVE 1            STATE_READY 2            STATE_RUNNING 3            STATE_SKIPPED 4            STATE_FINISHED 5            STATE_FAILED 6            STATE_TERMINATED 7            STATE_CLAIMED 8            STATE_TERMINATING 9            STATE_FAILING 10            STATE_WAITING 11            STATE_EXPIRED 12            STATE_STOPPED 13</p> <p>這些是針對 ActivityInstanceData.STATE_* 定義的常數</p>
CONTROL_LINK_NAME					x		與現行鏈結事件相關的鏈結名稱。
PRINCIPAL		x	x	x	x	x	主體的名稱。不會針對 PROCESS_DELETED 事件設定此項。
VARIABLE_DATA				x			variable updated 事件的變數資料。

表 103. AUDIT\_LOG\_B 審核追蹤視圖的結構 (繼續)

名稱	PTE	PIE	AIE	VAR	CLE	SIE	說明
EXCEPTION_TEXT		x	x			x	導致活動或程序失敗的異常訊息。適用於：  PROCESS_FAILED ACTIVITY_FAILED SCOPE_FAILED
DESCRIPTION		x	x	x	x	x	活動或程序的說明，可能包含已解析的取代變數。
CORR_SET_INFO		x					起始設定程序開始時間的相互關係集的字串表示法。與 processCorrelationSetInitialized 事件 (42027) 一起提供。
USER_NAME		x	x				已變更其工作項目的使用者名稱。這適用於下列事件： <ul style="list-style-type: none"> <li>• 程序實例工作項目已刪除</li> <li>• 活動實例工作項目已刪除</li> <li>• 程序實例工作項目已建立</li> <li>• 活動實例工作項目已建立</li> </ul>
ADDITIONAL_INFO		x	x			x	此欄位的內容取決於事件的類型：  <b>ACTIVITY_WORKITEM_TRANSFERRERD, PROCESS_WORK_ITEM_TRANSFERRERD</b> 接收工作項目的使用者名稱。  <b>ACTIVITY_WORKITEM_CREATED, ACTIVITY_WORKITEM_REFRESHED, ACTIVITY_ESCALATED</b> 建立或重新整理其工作項目的所有使用者清單 (以 ';' 區隔)。如果清單只包含一位使用者，則 USER_NAME 欄位會填入此使用者的使用者名稱，而 ADDITIONAL_INFO 欄位將為空白 (null)。  <b>PROCESS_EVENT_RECEIVED, SCOPE_EVENT_RECEIVED</b> 如果可用，則為事件處理程式接收的作業類型。會使用下列格式：'{ port type namespace }' port type name ':' operation name。不會針對 'onAlarm' 事件設定此欄位。

### 人力作業的審核追蹤資料庫視圖的結構

TASK\_AUDIT\_LOG 資料庫視圖提供人力作業的審核日誌相關資訊。

行內作業在 AUDIT\_LOG\_B 視圖中記載。所有其他作業類型在 TASK\_AUDIT\_LOG 視圖中記載。

若要讀取審核追蹤的內容，請使用 SQL 或任何支援讀取資料庫表格及視圖的其他管理工具。

審核事件與作業實體有關。審核事件類型取決於事件參照的實體。審核事件類型包括：

- 作業實例事件 (TIE)
- 作業範本事件 (TTE)
- 呈報實例事件 (EIE)

下表說明 TASK\_AUDIT\_LOG 審核追蹤視圖的結構。其中列出直欄的名稱、事件類型並提供直欄的簡要說明。

行內作業在 AUDIT\_LOG\_B 審核追蹤視圖中記載，而不是在 TASK\_AUDIT\_LOG 審核追蹤視圖中記載。例如，要求行內參與作業會導致 ACTIVITY\_CLAIMED 事件；而不會產生作業相關的事件。

表 104. TASK\_AUDIT\_LOG 審核追蹤視圖的結構

名稱	TIE	TTE	EIE	說明
ALID	x	x	x	審核日誌項目的 ID。
AUDIT_EVENT	x	x	x	發生的事件類型。如需審核事件碼的清單，請參閱第 562 頁的『人力作業的審核事件類型』。
CONTAINMENT_ CTX_ID	x	x		包含環境定義的 ID，例如 ACOID、PTID 或 PIID。
DESCRIPTION	x		x	已解析的說明字串，其中說明內的位置保留元由其現行值取代。所有受影響的語言都記載在此直欄中，格式化為 XML 文件。只會記載其說明包含建立之類事件的位置保留元的語言或已明確針對更新之類事件進行更新的語言。
ESIID			x	與現行事件相關的呈報實例 ID。
ESTID			x	與現行事件相關的呈報範本 ID。
EVENT_TIME	x	x	x	事件發生時的時間 (格式為「世界標準時間 (UTC)」)。
FAULT_NAME	x			錯誤訊息的名稱。此屬性適用於下列其中一個事件： TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FAULT_NAME_SPACE	x			錯誤訊息類型的名稱空間。此屬性適用於下列其中一個事件： TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FOLLOW_ON_TKIID	x			後續作業實例的 ID。
MESSAGE_DATA	x			新建或更新的輸入、輸出或錯誤訊息的內容。
NAME	x	x	x	與事件相關聯的作業實例、作業範本或呈報實例的名稱。
NAMESPACE	x	x		與事件相關聯的作業實例、作業範本或呈報實例的名稱空間。

表 104. TASK\_AUDIT\_LOG 審核追蹤視圖的結構 (繼續)

名稱	TIE	TTE	EIE	說明
NEW_USER				已轉移或建立的工作項目的新擁有者。如果值透過 USERS 欄位變成可用，此值可能為 null。另請參閱欄位 USERS。此屬性適用於下列事件：
	x			TASK_WORKITEM_CREATED
	x			TASK_WORKITEM_TRANSFERRED
			x	ESCALATION_WORKITEM_CREATED
OLD_USER			x	ESCALATION_WORKITEM_TRANSFERRED
			x	ESCALATION_WORKITEM_DELETED
	x			TASK_WORKITEM_TRANSFERRED
	x			TASK_WORKITEM_DELETED
PARENT_CONTEXT_ID	x			作業母項環境定義的 ID，例如活動範本或作業實例。只對子作業及後續作業設定此屬性。
PARENT_TASK_NAME	x			母項作業實例或範本的名稱。只對子作業及後續作業設定此屬性。
PARENT_TASK_NAMESP	x			母項作業實例或範本的名稱空間。只對子作業及後續作業設定此屬性。
PARENT_TKIID	x			母項作業實例的 ID。
PRINCIPAL	x	x	x	其要求觸發了事件的主體名稱。
TASK_KIND	x	x		作業的類型。可能的值如下：  KIND_HUMAN 101 KIND_ORIGINATING 103 KIND_PARTICIPATING 105 KIND_ADMINISTRATIVE 106
TASK_STATE	x			作業或作業範本的狀態。作業範本的可能值為：  STATE_STARTED 1 STATE_STOPPED 2 作業實例的可能值為：  '1' : 'STATE_INACTIVE' '2' : 'STATE_READY' '3' : 'STATE_RUNNING' '5' : 'STATE_FINISHED' '6' : 'STATE_FAILED' '7' : 'STATE_TERMINATED' '8' : 'STATE_CLAIMED' '12' : 'STATE_EXPIRED' '101' : 'FORWARDED'
TKIID	x		x	作業實例的 ID。
TKTID	x	x		作業範本的 ID。
TOP_TKIID	x			頂端作業實例的 ID。

表 104. TASK\_AUDIT\_LOG 審核追蹤視圖的結構 (繼續)

名稱	TIE	TTE	EIE	說明
USERS	x		x	指派給作業或呈報工作項目的新使用者 ID。如果值透過 NEW_USER 欄位變成可用，此值可能為 null。請參閱欄位 NEW_USER，以取得此屬性適用於的事件清單。
VALID_FROM		x		與現行事件相關的作業範本的有效起始日期。
WORK_ITEM_REASON	x		x	指派工作項目的原因。可能的值如下：  POTENTIAL_OWNER 1 EDITOR 2 READER 3 OWNER 4 POTENTIAL_STARTER 5 STARTER 6 ADMINISTRATOR 7 POTENTIAL_SENDER 8 ORIGINATOR 9 ESCALATION_RECEIVER 10 POTENTIAL_INSTANCE_CREATOR 11  針對與工作項目相關的所有事件設定原因：對與呈報工作項目相關的事件設定 ESCALATION_RECEIVER，而其他原因適用於與作業工作項目相關的事件。





---

## 第 8 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。本書在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，其他非 IBM 產品、程式、或服務在運作上的評價與驗證，其責任屬於使用者。

在這本書或文件中可能包含著 IBM 所擁有之專利或專利申請案。本書使用者並不享有前述專利之任何授權。您可以用書面方式來查詢授權，來函請寄到：

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

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

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

**下列段落若與該國之法律條款抵觸，即視為不適用：**IBM 僅以「現狀」提供本書，而不提供任何明示或默示之保證 (包括但不限於可商用性或符合特定效用的保證)。若有些地區在某些交易上並不允許排除上述保證，則該排除無效。

本資訊中可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。同時，IBM 得隨時改進並 (或) 變動本書中所提及的產品及 (或) 程式。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供保證。該網站上的資料，並非本 IBM 產品所用資料的一部分，如因使用該網站而造成損害，其責任由貴客戶自行負責。

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

本程式之獲授權者若希望取得本程式之相關資訊，以便達到下列目的：(i) 在獨立建立的程式與其他程式 (包括本程式) 之間交換資訊；以及 (ii) 相互使用已交換之資訊。則請與位於下列地址之人員聯絡：

*IBM Corporation  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
U.S.A.*

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之「IBM 客戶合約」、「國際程式授權合約」或任何同等合約之條款，提供本資訊中所說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。本文件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。

此資訊包含日常企業運作所使用的資料和報告的範例。為求儘可能地完整說明，範例包括了個人、公司、品牌和產品的名稱。這些名稱全為虛構，如有雷同，純屬巧合。

著作權授權：

本資訊包含原始語言的範例應用程式，用以說明各種作業平台上的程式設計技術。貴客戶得為開發、使用、行銷或散佈運用樣本程式之作業平台的應用程式程式介面所撰寫的應用程式之目的，免費複製、修改並散佈這些樣本程式。此些範例並未在所有情況下完整測試。故 IBM 不保證或默示保證這些程式之可靠性、服務性或功能。

這些範例程式的任何部份或任何衍生著作的每一份拷貝，都必須具有下列著作權聲明：(c) (貴公司名稱) (年)。部份程式碼衍生自 IBM Corp. 範例程式。(c) Copyright IBM Corp. \_輸入年份\_. All rights reserved.

如果您檢視此資訊的電子檔，則相片及顏色圖例可能不會出現。

## 程式設計介面資訊

程式設計介面資訊 (如果提供的話) 可協助您使用此程式建立應用軟體。

一般用途的程式設計介面可讓您撰寫取得此程式工具服務的應用軟體。

然而，此資訊可能還包含診斷、修改及調整資訊。提供的診斷、修改及調整資訊可協助您對應用軟體進行除錯。

**警告：**因為此診斷、修改及調整資訊可能會發生變更，所以請勿將其用作程式設計介面。

## 商標及服務標記

AIX、DB2、i5/OS、IBM、IBM 標誌、Informix、iSeries、MQSeries、WebSphere 及 z/OS 是 International Business Machines Corporation 在美國及 (或) 其他國家或地區的註冊商標，而 Cloudscape、DB2 Connect 及 DB2 Universal Database 是 International Business Machines Corporation 在美國及 (或) 其他國家或地區的商標。

Java 及所有以 Java 為基礎的商標是 Sun Microsystems, Inc. 在美國及 (或) 其他國家或地區的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及 (或) 其他國家或地區的商標。

UNIX 是 The Open Group 在美國及其他國家或地區的註冊商標。

Linux 是 Linus Torvalds 在美國及 (或) 其他國家或地區的註冊商標。

其他公司、產品或服務名稱，可能是其他公司的商標或服務標誌。

此產品包括 Eclipse Project (<http://www.eclipse.org>) 所開發的軟體。



IBM WebSphere Process Server for Multiplatforms 6.1.0 版



## 讀者意見表

為使本書盡善盡美，本公司極需您寶貴的意見；懇請您閱讀後，撥冗填寫下表，惠予指教。

請於下表適當空格內，填入記號 (✓)；我們會在下一版中，作適當修訂，謝謝您的合作!

評估項目	評估意見	備註
正確性	內容說明與實際程序是否符合	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	參考書目是否正確	<input type="checkbox"/> 是 <input type="checkbox"/> 否
一致性	文句用語及風格，前後是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	實際產品介面訊息與本書中所提是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
完整性	是否遺漏您想知道的項目	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	字句、章節是否有遺漏	<input type="checkbox"/> 是 <input type="checkbox"/> 否
術語使用	術語之使用是否恰當	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	術語之使用，前後是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
可讀性	文句用語是否通順	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	有否不知所云之處	<input type="checkbox"/> 是 <input type="checkbox"/> 否
內容說明	內容說明是否詳盡	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	例題說明是否詳盡	<input type="checkbox"/> 是 <input type="checkbox"/> 否
排版方式	本書的形狀大小，版面安排是否方便閱讀	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	字體大小，顏色編排，是否有助於閱讀	<input type="checkbox"/> 是 <input type="checkbox"/> 否
目錄索引	目錄內容之編排，是否便於查找	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	索引語錄之排定，是否便於查找	<input type="checkbox"/> 是 <input type="checkbox"/> 否
※評估意見為 "否" 者，請於備註欄提供建議。		

其他：(篇幅不夠時，請另外附紙說明。)

---



---



---



---



---



---



---



---



---



---

上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。  
 註：您也可將寶貴的意見以電子郵件寄至 [tscadmin@tw.ibm.com](mailto:tscadmin@tw.ibm.com)，謝謝。

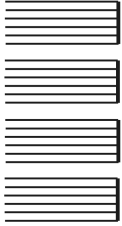


WebSphere Process Server for Multiplatforms  
版本 6.1.0 商業程序編排器

折疊線

110 台北市信義區松仁路 7 號 3 樓

臺灣國際商業機器股份有限公司  
大中華研發中心 軟體國際部 啟



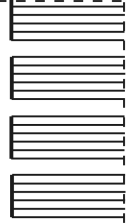
廣 告 回 信
台灣北區郵政管理局 登記證
北台字第 00176 號

(免貼郵票)

寄件人 姓名：  
地址：

寄

折疊線





**IBM**