

Workday Studio

Product Summary

March 10, 2026



目錄

設定注意事項：Workday Studio.....	10
Studio 基礎知識.....	12
下載並安裝 Windows 版 Workday Studio.....	12
下載並安裝 MacOS 版 Workday Studio.....	13
概念：Workday Studio.....	15
概念：整合開發生命週期.....	16
概念：MediationContext 物件.....	17
概念：整合系統和事件.....	18
參照：介面基礎知識.....	18
整合設計.....	19
組合併.....	19
建立集合專案.....	19
將步驟新增至元件.....	20
新增子系元素.....	21
連接集合元素.....	21
配置集合元素屬性.....	21
管理收款和專案.....	22
宣告名稱空間字首.....	23
建立路徑變數.....	24
概念：集合調色板.....	24
概念：元件和步驟.....	25
概念：集合註解.....	26
概念：組合專案的相關資源檔案.....	26
概念：快速大綱檢視.....	27
參照：組合屬性.....	27
參照：泳道屬性.....	27
參照：連線行屬性.....	28
裝配件傳輸.....	28
在 Studio 中使用列舉做為啟動參數.....	28
概念：集合傳輸.....	29
概念：列舉.....	30
參照：Workday-In 傳輸屬性.....	31
參照：使用當地文字輸入傳輸屬性.....	32
參照：Workday-Out-Rest 傳輸屬性.....	34
參照：Workday-Out-Soap 傳輸屬性.....	35
參照：本端傳出傳輸屬性.....	36
參照：HTTP 傳出傳輸屬性.....	37
參照：FTP 傳出傳輸屬性.....	39
參照：FTPS 傳出傳輸屬性.....	43
參照：SFTP 傳出傳輸屬性.....	46
參照：XMPP 傳出傳輸屬性.....	50
參照：自訂傳出傳輸屬性.....	51
參照：電子郵件傳出傳輸屬性.....	52
HTTP 傳出傳輸子元素.....	58
概念：HTTP-Out 傳輸子元素.....	58

參照：HTTP-基本驗證屬性.....	58
參照：HTTP 自訂驗證屬性.....	59
參照：HTTPS 屬性.....	60
FTP-Out 傳輸子元素.....	60
概念：FTP-Out 傳輸子元素.....	60
參照：Proxy 屬性屬性.....	61
參照：防火牆-屬性內容.....	61
參照：用戶端金鑰屬性.....	61
參照：伺服器金鑰屬性.....	62
SFTP-Out 子系元素.....	62
概念：SFTP-Out 子系元素.....	62
參照：Proxy 屬性屬性.....	62
參照：私人金鑰屬性.....	63
元件.....	63
概念：元件.....	63
參照：非同步中介元件屬性.....	64
參照：同步中介元件屬性.....	64
參照：自訂中介元件屬性.....	65
參照：路由元件屬性.....	65
參照：拆分器元件屬性.....	66
參照：匯總器元件屬性.....	66
路由元件子系元素.....	66
概念：路由元件子系元素.....	66
參照：自訂策略屬性.....	67
參照：Regex-策略屬性.....	68
參照：最佳實際教學應用程式 - 策略屬性.....	68
參照：循環策略屬性.....	68
參照：XPath-策略屬性.....	68
參照：故障轉移策略屬性.....	69
參照：所有策略屬性.....	69
參照：文件迭代器屬性.....	69
參照：循環策略屬性.....	70
參照：選擇路由屬性.....	70
參照：子路由屬性.....	70
拆分器元件子系元素.....	70
概念：拆分器元件子系元素.....	70
參照：自訂拆分屬性.....	71
參照：標準拆分屬性.....	71
參照：XPath 拆分器屬性.....	72
參照：XML 串流拆分器屬性.....	72
參照：JSON 拆分器屬性.....	72
參照：解壓縮拆分器屬性.....	73
參照：Mtable 拆分器屬性.....	73
標準拆分子元素.....	73
概念：標準拆分子元素.....	73
參照：標頭結尾包含屬性.....	74
參照：標頭固定明細屬性.....	74
參照：內容固定線屬性.....	74
參照：內容-開始-結束屬性.....	74
參照：內容開始於屬性.....	74
參照：頁尾固定線屬性.....	75
參照：頁尾開始位置屬性.....	75
匯總器組成要素子系元素.....	75
概念：匯總器組成要素子系元素.....	75
參照：自訂批次策略屬性.....	76
參照：大小-批次-策略屬性.....	76

參照：時間批次策略屬性.....	76
參照：自訂整理來源屬性.....	76
參照：訊息內容整理程序屬性.....	77
參照：XML-Message-Content-Collater 屬性.....	77
參照：JSON 匯總屬性.....	78
參照：Zip-File-Collater 屬性.....	79
參照：Mtable-Collater 屬性.....	79
通用元件.....	80
概念：通用元件.....	80
參照：FTP 子組合屬性.....	82
參照：GetEventConfigurations 子組合屬性.....	84
參照：GetEventDocuments 子組合屬性.....	86
參照：GetIntegrationEvent 子組合件屬性.....	87
參照：GetIntegrationSystems 子組合體屬性.....	88
參照：PagedGet 子組合件屬性.....	89
參照：PagedGetLocalPaging 子組合屬性.....	92
參照：PdfPrintStep 子組合屬性.....	94
參照：PutIntegrationEvent 子組合件屬性.....	96
參照：PutIntegrationMessage 子組合件屬性.....	97
參照：SalesforceConnector 子組合件屬性.....	100
PrismAnalytics 子裝配件.....	101
錯誤處理程式元件.....	112
概念：錯誤處理程式元件.....	112
參照：自訂錯誤處理程式元件屬性.....	113
參照：記錄錯誤元件屬性.....	114
參照：傳送錯誤元件屬性.....	115
轉換步驟.....	116
使用文字結構描述檔案將文字轉換為 XML.....	116
概念：轉換步驟.....	118
參照：XSLT 步驟屬性.....	120
參照：XSLT+ 步驟屬性.....	122
參照：FOP 步驟屬性.....	123
參照：文字-Excel 步驟屬性.....	126
參照：文字結構描述步驟屬性.....	128
參照：自動換行步驟屬性.....	129
參照：XML 轉 Java 步驟屬性.....	131
參照：JSON 轉換器屬性.....	132
參照：XML 轉 JSON 步驟屬性.....	133
參照：JSON 轉 XML 步驟屬性.....	135
參照：Java 轉 XML 步驟屬性.....	136
參照：XML 轉 CSV 步驟屬性.....	137
參照：CSV 轉 XML 步驟屬性.....	139
參照：字元轉換步驟屬性.....	141
參照：Mtable-建置器步驟屬性.....	142
參照：Mtable-Writer 步驟屬性.....	143
參照：ETV 步驟屬性.....	144
參照：XTT 步驟屬性.....	145
Mtable-建置器子元素.....	147
概念：Mtable-建置器子系元素.....	147
參照：CSV-Mtable-Reader 屬性.....	147
參照：JSON-Mtable-Reader 屬性.....	147
Mtable-Writer 子元素.....	148
概念：Mtable-Writer 子元素.....	148
參照：CSV-Mtable-Writer 屬性.....	148
參照：JSON-Mtable-Writer 屬性.....	148
記錄和事件處理步驟.....	149

概念：記錄和事件處理步驟.....	149
參照：記錄步驟屬性.....	149
參照：雲端記錄步驟屬性.....	151
訊息處理步驟.....	152
概念：訊息處理步驟.....	152
參照：複製步驟屬性.....	153
參照：JavaScript 步驟屬性.....	154
參照：設定標頭步驟屬性.....	156
參照：寫入步驟屬性.....	157
驗證步驟.....	158
概念：驗證步驟.....	158
參照：評估步驟屬性.....	159
參照：驗證步驟屬性.....	159
參照：驗證計算步驟屬性.....	161
參照：驗證 XPath 步驟屬性.....	162
編碼步驟.....	163
概念：編碼步驟.....	163
概念：PGP 檔案名稱.....	164
參照：PGP 加密步驟屬性.....	165
參照：PGP 解密步驟屬性.....	167
參照：Base64 解碼步驟屬性.....	169
參照：Base64 編碼步驟屬性.....	170
參照：Zip 步驟屬性.....	171
參照：解壓縮步驟屬性.....	173
參照：壓縮步驟屬性.....	174
參照：解壓縮步驟屬性.....	175
其他步驟.....	177
概念：其他步驟.....	177
參照：自訂步驟屬性.....	177
參照：設定動態端點步驟屬性.....	178
參照：儲存步驟屬性.....	179
參照：擷取步驟屬性.....	181
參照：Xmldiff 步驟屬性.....	183
參照：Enqueue-Message 步驟屬性.....	185
集合模組.....	187
建立集合模組.....	187
將組套件模組新增至專案.....	188
概念：集合模組.....	189
子裝配件.....	189
建立子裝配件.....	189
概念：子裝配件.....	190
自訂集合元件.....	190
建立自訂 Java 類和 Spring Bean.....	190
概念：自訂集合元素.....	191
概念：自訂元素封裝.....	193
Spring Bean 配置.....	194
概念：Spring Bean 配置.....	194
參照：全域 Bean 屬性.....	194
參照：個別 Bean 屬性.....	195
參照：別名屬性.....	196
參照：屬性屬性.....	196
參照：構造函數- Arg 屬性.....	196
參照：參照屬性.....	196
參照：清單屬性.....	197
參照：地圖屬性.....	197
訊息建置器.....	197

根據結構描述或 WSDL 建立 XML 文字.....	197
根據結構描述或 WSDL 建立 XPath 運算式.....	198
概念：Workday Studio 訊息建置器.....	199
參照：訊息元素屬性.....	200
傳輸系結.....	203
概念：傳輸系結.....	203
參照：WSDL-HTTP-Binding 屬性.....	204
參照：WSDL-SOAP-Binding 屬性.....	204
WSDL-SOAP 系結子元素.....	204
概念：WSDL-SOAP-Binding 子元素.....	204
參照：自訂原則屬性.....	205
參照：WS-Rm 屬性.....	205
參照：WS-Security 屬性.....	205
參照：WS-Ut 屬性.....	205
結構描述資源管理器.....	206
顯示 WSDL 和 XSD 檔案.....	206
顯示 Workday Web 服務 WSDL.....	206
顯示自訂 RaaS 報告結構描述.....	207
概念：結構描述探索器.....	207
MVEL.....	207
概念：MVEL 基礎知識.....	207
概念：MVEL 和 Studio.....	210
參照：Studio Helper 物件.....	211
參照：文件存取器 Helper 物件.....	215
參照：啟動參數 Helper 物件.....	218
參照：效用 Helper 物件.....	220
參照：範例 Studio MVEL 運算式.....	221
參照：Studio MVEL 自動轉換語法範例.....	223
參照：實際應用程式計量檢視方法.....	225
Studio 整合服務.....	226
建立屬性對應服務.....	226
建立交付服務.....	228
建立偵聽器服務.....	228
建立擷取服務.....	229
建立序列產生器服務.....	229
建立異動記錄服務.....	230
建立 Report Services.....	231
建立自訂物件服務.....	232
建立資料起始服務.....	233
建立服務參考.....	233
新增對 Workday-Out-REST 傳輸中自訂報告的參照.....	234
新增對 Workday-Out-REST 傳輸中自訂物件的參照.....	235
CLAR 檔案.....	236
產生載貨清單檔案.....	236
匯入 CLAR 檔案.....	236
匯出 CLAR 檔案.....	237
概念：CLAR 檔案.....	237
JAXB 專案.....	238
建立 JAXB 專案.....	238
將 JAXB 專案新增為程序集專案中的依存關係.....	239
概念：Studio 中的 JAXB 支援.....	240
Studio Ivy 外掛程式.....	240
配置 Ivy 外掛程式.....	240
在 Workday Studio 專案中新增 Workday Ivy 屬性.....	240
為具有 Ivy 定義的專案解除依存關係.....	241
將相依 JAR 新增至常春藤項目.....	241

從 Studio 專案中移除 Workday Ivy 屬性.....	242
整合部署.....	242
新增人脈.....	242
新增人脈群組.....	243
編輯人脈.....	244
編輯人脈群組.....	244
移除人脈.....	245
匯入連結方.....	245
匯出連結方.....	246
連線至 Workday.....	246
將整合部署至 Workday.....	247
在 Workday 中檢視已部署的整合.....	247
將 Studio 配置為使用 HTTPS Proxy Server.....	248
概念：Studio 連結.....	248
概念：Cloud Explorer.....	248
參照：連線錯誤.....	249
Workday Studio 中的多重要素驗證.....	250
在 Workday Studio 中啟用多重要素驗證.....	250
整合配置.....	251
將整合系統使用者帳戶指派給整合.....	251
概念：整合系統安全性.....	252
整合啟動.....	252
從 Studio 啟動整合.....	252
複製啟動 URL.....	253
建立啟動配置.....	253
編輯啟動配置.....	254
整合監控與疑難排解.....	255
流程監控.....	255
概念：整合事件 ID.....	255
參照：流程監控工具列按鈕.....	255
合併報告檢視器.....	255
載入整合的合併報告.....	255
將合併報告儲存至您的工作區.....	256
使用合併報告檢視 Workday 整合事件詳細資料.....	257
使用合併報告檢視整合事件的要求訊息.....	257
使用合併報告檢視器事件記錄監控整合.....	258
使用合併報告檢視器設定檔記錄監控整合.....	259
重播合併報告的程序集執行.....	259
將合併報告稽核事件連結至程序集元素.....	260
在組合圖上顯示合併報告註解.....	260
概念：合併報告檢視器.....	261
參照：合併報告錯誤訊息.....	261
單元集合測試.....	265
建立程序集專案的 AUnit 測試框架.....	265
概念：A 單元測試.....	265
參照：功能單元測試註解.....	267
參照：功能單元測試動作.....	268
Web 服務測試.....	269

使用 WSDL 檔案測試 Web 服務.....	269
使用 XSD 檔案測試 Web 服務.....	270
從 Schema Explorer 測試個別 Web 服務.....	270
使用要求建置器精靈測試 Get 作業和要求訊息.....	271
配置 Web 服務測試人員驗證.....	272
配置 Web 服務測試工具 X.509 驗證.....	272
配置 Web Service Tester Advanced WS-Security.....	273
概念：Web 服務測試.....	273
參照：Web 服務測試工具列按鈕.....	274
程序集偵錯.....	275
偵錯程序集.....	275
概念：程序集偵錯器.....	276
Studio 管理.....	277
Studio 喜好設定.....	277
概念：集合編輯器喜好設定.....	277
概念：人脈喜好設定.....	278
概念：部署喜好設定.....	278
概念：HTTPS 喜好設定.....	278
概念：Ivy 喜好設定.....	278
概念：連結喜好設定.....	278
概念：移動設備清單 (MEL) 驗證喜好設定.....	279
概念：套裝整合喜好設定.....	279
概念：執行階段喜好設定.....	279
概念：伺服器喜好設定.....	279
概念：啟動畫面登入喜好設定.....	279
概念：狀態列喜好設定.....	279
概念：更新喜好設定.....	279
概念：XPath Explorer 喜好設定.....	279
參照：程序集偵錯執行階段喜好設定.....	279
參照：程序集偵錯檢視喜好設定.....	280
參照：Cloud Explorer 喜好設定.....	280
參照：合併報告檢視器喜好設定.....	281
參照：Java 專案設定喜好設定.....	281
參照：通知喜好設定.....	282
參照：流程監控喜好設定.....	282
參照：Web 服務測試人員喜好設定.....	282
Studio 更新.....	282
更新 Studio.....	282
Workday 程序集執行階段變更.....	283
概念：程序集版本.....	283
參照：結構描述層級的版本變更.....	283
參照：XML 層級的版本變更.....	286
參照：Saxon XSLT 處理器變更.....	288
範例程序集.....	289
範例：執行 Hello Cloud 範例程序集.....	289
範例：執行 Hello Workday Web 服務範例程序集.....	290
範例：執行常見使用模式範例程序集.....	290
範例：執行偵錯範例程序集.....	292
範例：執行 Workday-In 功能範例程序集.....	293
範例：執行錯誤處理範例程序集.....	294
範例：執行 AUnit 範例測試.....	295
範例：執行 FOPStep 範例程序集.....	296

範例：針對員工範例程序集執行擷取組織資料.....	297
範例：執行 PrismAnalytics 範例集合.....	298
第三方薪資介面範例.....	300
步驟：執行薪資介面範例.....	300
建立薪資介面安全群組.....	300
建立 PayData 擷取整合系統.....	301
測試 PayData 擷取整合系統.....	302
部署 Studio PayData 整合.....	303
將整合系統使用者指派給 PayData 整合.....	304
連結 PayData 擷取與 PayData 整合.....	304
啟動連結的整合.....	305

設定注意事項：Workday Studio

註：本節所述的解決方案不屬於 Workday 服務。請參閱「[法律公告](#)」以取得詳細資料。

本主題內容可為您規劃配置和使用 Workday Studio 時提供參考。主要內容包括：

- 設定工時確認的原因
- 如何搭配 Workday 的其他功能使用
- 後續影響以及跨產品交互作用
- 安全要求事項與業務流程配置
- 實作前應考量的問題和限制

請參閱詳細任務說明，取得完整的配置詳細資料。

內容

Studio 是一種以 Eclipse™ 為基礎的整合開發環境 (IDE)，可讓您：

- 建立自訂整合。
- 與外部端點交換大量資料。
- 將整合上傳至 Workday。

Studio 是一個外部軟體應用程式，您可以將其與 Workday 的其他軟體分開安裝。

商業效益

Studio 可讓您執行以下動作，藉此提高生產力：

- 建立的整合可支援比企業介面建置器 (EIB) 或連接器範本更大的設計靈活性。
- 偵錯自訂整合的 XML 來源代碼。
- 透過 Workday 匯出和匯入大量資料。
- 將大型資料集轉換為特定格式。例如，您可以將 CSV 資料集轉換為 XML 格式。

使用個案

您需要：

- 無法透過 EIB 或連接器範本使用的自訂整合。範例：在資料集上實作 XML 至文字 (XTT) 轉換的整合。
- 內傳整合功能，可將大量資料從多個資料來源匯入 Workday。
- 一種外傳整合功能，可將 Workday 中的大量資料以特定格式匯出。
- 對自訂整合的 XML 來源代碼執行單元測試。

需要考慮的問題

問題	注意事項
您的組織是否與 Oracle 簽訂了長期支援協議？	如果是，您可以使用 Oracle JDK 1.8 安裝 Studio。 如果不是，您可以使用 OpenJDK 8 安裝 Studio。
您使用哪個平台來安裝 Studio？	為您的平台安裝適當的安裝檔案： • MacOS (64 位元) • Windows (32 位元) • Windows (64 位元)

問題	注意事項
您是否參加過 Studio 的培訓？	對於有興趣參加 Studio 培訓的學員，Workday 提供了「使用 Workday Studio 建立整合」課程。前往培訓目錄了解更多資訊。

建議

Workday 建議您先在預覽沙箱環境中部署並測試自訂的 Studio 整合，再將其轉移至正式環境用戶。

要求

Studio 整合需要與 Java 版本 8 相容的 Java™ 開發套件 (JDK)。此外，Eclipse IDE 需要與 Java 版本 17 相容的 JDK。如果您：

- 具有 Oracle 的授權合約，使用 Oracle JDK 1.8 和 Oracle JDK 1.17。
- 未與 Oracle 簽署協議，請使用 OpenJDK 8 和 OpenJDK 17。

兩個 JDK 都是必要的。

限制

Studio 整合輸出檔案的大小上限為：

- 個別檔案 1 GB (壓縮)
- 所有檔案總計 3 GB (壓縮)。

用戶設定

在 Workday 的「編輯用戶設定 - 整合」任務中勾選「需要來源代碼」核取方塊，即可存取 Studio 整合的來源代碼。存取 Studio 整合的來源代碼，可讓您找出錯誤並提高效率。

安全性

請考慮在「整合」功能區中設定這些網域。

網域	注意事項
整合建置	將整合從 Studio 部署至 Workday。
整合配置	在 Workday 中編輯 Studio 整合
整合偵錯	在 Workday 中偵錯 Studio 整合
整合事件	在 Workday 中啟動 Studio 整合。
整合安全性	在 Workday 中為您的 Studio 整合建立整合系統使用者 (ISU) 帳戶。

業務流程

對於 Studio 整合，您可以配置：

- 「整合」功能區中的「整合流程事件」業務流程，用於控制整合的執行方式。
- 整合業務流程步驟，在業務流程期間啟動整合。範例：您可以將整合步驟新增至「完成 I-9 表」業務流程。此步驟會啟動整合，以傳送業務流程中員工的資料。

報告

Workday 為 Studio 整合提供以下報告：

報告	說明
整合事件	顯示： - 進行中或已完成的整合事件摘要。 - 每個整合事件的狀態。
檢視整合系統	顯示 Workday 用戶中所有可用的整合系統。

連結和觸點

根據整合的產品領域、用途和資料來源，Studio 會與 Workday 的不同部分互動：

- 傳入整合可在 Workday 中新增和更新資料。
- 外傳整合會從 Workday 擷取資料，但不會變更 Workday 中的資料。

Workday 提供了一個「觸點套件」，其中包含的資源可幫助您了解用戶之間的配置關係。在 Workday Community 深入了解 [Workday Touchpoints Kit \(Workday 觸點套件\)](#)。

Studio 基礎知識

下載並安裝 Windows 版 Workday Studio

執行這項作業的原因和時機

Workday Studio 是建立在 Eclipse 開發平台之上。除了平台和第三方依存關係外，它還包含 150 多個 Eclipse 外掛程式。若要使用此功能，您需要 2 個版本的 Java™ 開發套件 (JDK)：

- Eclipse 需要與 Java 版本 17 相容的 JDK。
- Studio 執行時需要與 Java 版本 8 相容的 JDK。

安裝兩個 JDK，然後確保：

- 您的 Java 執行階段環境已設為 Java 8。
- 您的 JDK 編譯器法規遵循層級已設為 Java 1.8。

程序

- 下載並安裝與 Java 版本 17 相容的 JDK。您可以從各種來源下載免費的開放原始碼實作 OpenJDK。
範例：

- 選取適合您作業系統的 [Zulu OpenJDK 17](#)。

註：

- Apple 系統上的使用者必須使用 Apple 系統上的 ARM JDK17。x86_64 JDK 將無法運作。
- Studio 需要 JDK 專為與 Java 17 相容，而最低要求為與 Java 17 不相容。

- 下載並安裝與 Java 版本 8 相容的 JDK。同樣，您可以選擇免費的開放原始碼實作 OpenJDK。
範例：

- 選取適合您作業系統的 [Zulu OpenJDK 8](#)

註：

- Apple 系統上的使用者必須使用 Apple 系統上的 ARM JDK8。x86_64 JDK 將無法運作。
- Studio 需要與 Java 8 完全相容的 JDK，最低要求是與 Java 8 不相容。

- 下載 Windows 版 Workday Studio [安裝程式](#)。

- 若要啟動安裝，請連按兩下可執行安裝檔案。若要在 C:\Program Files 資料夾中安裝 Studio，您必須在可執行檔上按一下滑鼠右鍵並選取「以管理者身分執行」，以「管理者」的身分執行。

註：

您可以使用指令提示符應用程式來執行 Studio 靜默安裝。在指令提示符中，導覽至您儲存安裝檔案的目錄，然後輸入：

```
<Installation_Filename>.exe /S /J=<Java_Home> /D=<Installation_Directory>
```

其中引數：

- /S 指定靜默安裝。
- /J 指定 JDK 17 路徑。
- /RD 安裝 Workday 報告設計工具。
- /D 指定安裝資料夾 (此引數必須為命令中的最後一個)。

範例：workday-studio-x86_64.exe /S /J=C:\Program Files\Zulu\zulu17.62.17-ca-jdk17.0.18-win_x64 /D=C:\Studio

- 閱讀授權協議，然後選取「接受條款」核取方塊。按一下下一步。
- 瀏覽至 JDK 17 安裝位置，然後按一下下一步。例如，JDK 安裝位置 (共 個)：C:\Program Files\Zulu\zulu17.62.17-ca-jdk17.0.18-win_x64。
- 如果 JVM 詳細資料正確無誤，請按一下 確認。否則，請按一下 返回。
- 選取 報告設計 工具包，將其與 Workday Studio 一起安裝。您也可以稍後從 Studio 中安裝報告設計工具。
- 瀏覽至您要安裝 Studio 的目錄。按一下下一步。
- 當 Studio 顯示「安裝完成」訊息時，請按一下下一步，然後按一下 完成。

註：使用 Java 17 的 Eclipse 無法使用使用 Java 8 的 Eclipse 建立的工作區。啟動時，您可以更新現有的工作區，使其與 Java 17 相容。或者，您可以建立全新的相容工作區。更新工作區後，您無法再將其與較早版本的 Studio 搭配使用。

- 為確保 Java 8 可做為 Studio 中的 Java 執行階段環境，請導覽至其「喜好設定」面板：視窗 > 喜好設定 > Java > 安裝的 JRE

如果缺少 Java 8，請按一下 新增，然後選取 標準虛擬機 做為 JRE 類型。指定您安裝 JDK 8 的目錄。

- 確定已選取 Java 8 JRE 旁邊的核取方塊。
- 檢查執行環境是否完全相符：視窗 > 喜好設定 > Java > 安裝的 JRE > 執行環境

在「Execution Environments」面板中選取 JavaSE-1.8，然後在「Compatible JRE」面板中選取描述為「完全相符」的環境。

- 在 Studio 的「喜好設定」面板中，導覽至 Java > 編譯器。
- 在 JDK 合規性下，從 編譯器合規性層級 下拉式功能表中選取 1.8。
- 按一下 套用並關閉。

註：配置 Studio 前，請參閱此 [Workday 資料中心總覽](#)。用戶的資料中心頁麵包含 IT 部門可用來配置防火牆並確保 Studio 可正確存取 Workday 的資訊。

下載並安裝 MacOS 版 Workday Studio

執行這項作業的原因和時機

Workday Studio 是建立在 Eclipse 開發平台之上。除了平台和第三方依存關係外，它還包含 150 多個 Eclipse 外掛程式。若要使用此功能，您需要 2 個版本的 Java™ 開發套件 (JDK)：

- Eclipse 需要與 Java 版本 17 相容的 JDK。
- Studio 執行時需要與 Java 版本 8 相容的 JDK。

安裝兩個 JDK，然後確保：

- 您的 Java 執行階段環境已設為 Java 8。
- 您的 JDK 編譯器法規遵循層級已設為 Java 1.8。

程序

1. 下載並安裝與 Java 版本 17 相容的 JDK。您可以從各種來源下載免費的開放原始碼實作 OpenJDK。

範例：

- 選取適合您作業系統的 [Zulu OpenJDK 17](#)。

註：

- Apple 系統上的使用者必須使用 Apple 系統上的 ARM JDK17。x86_64 JDK 將無法運作。
- Studio 需要 JDK 專為與 Java 17 相容，而最低要求為與 Java 17 不相容。

2. 下載並安裝與 Java 版本 8 相容的 JDK。同樣，您可以選擇免費的開放原始碼實作 OpenJDK。

範例：

- 選取適合您作業系統的 [Zulu OpenJDK 8](#)

註：

- Apple 系統上的使用者必須使用 Apple 系統上的 ARM JDK8。x86_64 JDK 將無法運作。
- Studio 需要與 Java 8 完全相容的 JDK，最低要求是與 Java 8 不相容。

3. 下載正確的 MacOS Studio 安裝程式：

- [Apple 芯片](#)
- [x86, 64 位元](#)

4. 若要啟動安裝，請開啟安裝 .jar 檔案。
5. 閱讀授權協議，然後選取「接受條款」核取方塊。按一下下一步。
6. 瀏覽至 JDK 17 安裝位置，然後按一下 下一步。範例：/Library/Java/JavaVirtualMachines/zulu-17.jdk/Contents/Home。
7. 如果 JVM 詳細資料正確無誤，請按一下 確認。否則，請按一下 返回。
8. 選取 報告設計 工具包，將其與 Workday Studio 一起安裝。您也可以稍後從 Studio 中安裝報告設計工具。
9. 瀏覽至您要安裝 Studio 的目錄。按一下下一步。
10. 當 Studio 顯示「安裝完成」訊息時，請按一下 下一步，然後按一下 完成。

註：使用 Java 17 的 Eclipse 無法使用使用 Java 8 的 Eclipse 建立的工作區。啟動時，您可以更新現有的工作區，使其與 Java 17 相容。或者，您可以建立全新的相容工作區。更新工作區後，您無法再將其與較早版本的 Studio 搭配使用。

11. 若要確保 Java 8 可做為 Studio 中的 Java 執行階段環境，請導覽至其「喜好設定」面板：
WorkdayStudio > 喜好設定 > Java > Installed JREs

如果缺少 Java 8，請按一下 新增，然後選取 標準虛擬機 做為 JRE 類型。指定您安裝 JDK 8 的目錄。

12. 確定已選取 Java 8 JRE 旁邊的核取方塊。
13. 檢查執行環境是否完全相符：Workday Studio > 喜好設定 > Java > Installed JRE > Execution Environments

在「Execution Environments」面板中選取 JavaSE-1.8，然後在「Compatible JRE」面板中選取描述為「完全相符」的環境。

14. 在 Studio 的「喜好設定」面板中，導覽至 Java > 編譯器。
15. 在 JDK 合規性下，從 編譯器合規性層級 下拉式功能表中選取 1.8。

16. 按一下 套用並關閉。

註：配置 Studio 前，請參閱此 [Workday 資料中心總覽](#)。用戶的資料中心頁麵包含 IT 部門可用來配置防火牆並確保 Studio 可正確存取 Workday 的資訊。

概念：Workday Studio

Workday 提供 2 種方式執行整合：

- 連接器：連接各種常用第三方服務的封裝整合。
- 企業介面建置器 (EIB)：無需任何程式設計即可建立整合的簡單工具。

Workday Studio 是第三種選擇，可讓您建立、擁有並支援 Workday 代您託管和執行的更複雜的整合。與連接器和 EIB 不同的是，Studio 不是 Workday 的一部分。這是您單獨下載的基於 Eclipse 的整合式開發環境 (IDE)。與用戶中的任何其他整合一樣，您可以啟動、排程和稽核您使用 Studio 建立的整合。您可以從 Studio 本身或從 Workday 應用程式啟動 Studio 建立的整合。

Studio 的目標使用者：

- 具有發展技能組合。
- 熟悉整合程式設計概念。
- 能熟練使用 IDE。

Studio 平台有 4 個主要元件：

- Studio IDE
- 整合、集合和雲端資料庫
- Workday 應用程式
- Cloud Runtime

Studio IDE

Studio 是以 Eclipse 為基礎的 IDE。您可以使用它來建構符合以下條件的整合流程：

- 從 Workday 存取資訊。
- 進行操作。
- (選用) 傳送至某處。

這些流程稱為程序集。您可以在 Studio 組合編輯器中新增和連接稱為元件和步驟的圖形元素，以建立組合。

Studio 包含許多動作的元件和步驟，包括拆分、轉換和路由。Workday 特定元件內建說明 Workday 如何呈現資訊。

您可以先在 Studio 中測試您的整合，然後再將其直接部署至 Workday 用戶。

Studio 支援在程序集專案中使用 Java 延伸。您可以提供自己的 Java 代碼，並將其配置為 Spring bean。此外，您可以在整合內封裝第三方 Java 資料庫。

整合、集合和雲端儲存庫

「雲端資料庫」是 Workday 對 Studio 整合儲存空間的簡稱。在部署之前，Studio 會將每個整合指派給一個集合。一個集合可以包含多個整合。整合可屬於多個集合。這表示您可以跨整合和用戶輕鬆重複使用組件和子組件。

任何時候只能執行特定集合的 1 個實例。如果您修改集合內的程序集專案並重新部署該專案，該專案會取代任何先前部署的版本。

Studio 部署到 Workday 雲端儲存庫的二進位檔案稱為 CLAR (CLOUD ARchive)。CLAR 包含：

- 所包含的整合專案中的所有成品，例如其 components.xml、XSLT 和文字結構描述檔案。
- 開發人員已新增至整合專案的任何已編譯 Java 代碼或第三方 Java 資料庫。

由於是二進位物件，CLAR 不會包含 Java 來源檔案等來源代碼成品。

儲存庫會以加密格式儲存 CLAR。但是，請勿將密碼等資訊直接儲存在您的整合中。請改為在 Workday 應用程式中配置敏感詳細資料，以便進行動態管理。

Workday 應用程式

Workday 使用稱為整合系統的抽象來建立整合模型。在 Workday 中，您可以針對每個整合指定：

- 哪些使用者有權啟動整合，以及執行整合時擁有的權限。
- 在執行整合時，如何以及何時通知不同的 Workday 使用者。

由於您是單獨設計和配置 Studio 整合，因此您可以重新部署整合，而不會影響其配置。

Workday 可讓整合在許多不同的情境中啟動。包括：

- 手動啟動整合
- 排定整合時間，以事件的形式自動執行。
- 在業務流程中自動啟動整合。
- 使用 Web 服務用戶端遠距觸發啟動，以呼叫 Workday Web 服務。

您可以使用 Workday 流程監控 報告來監控整合的執行情況。

Cloud Runtime

當整合啟動時，Workday 會：

- 產生整合事件。
- 從雲端儲存庫擷取整合。
- 指派一組稱為雲端執行階段受保護的處理資源。
- 在 Cloud Runtime 中執行整合。

如果 Workday 無法執行整合，則會將「流程監控」更新為整合啟動狀態 Failed。

如果 Workday 在執行過程中遇到問題，例如無法使用外部資源，它會終止 流程監控，並將整合啟動狀態更新為「」。Completed with errors。

註：整合開發人員有責任處理標示為「」的整合 Completed with errors。

Workday 會終止執行時間超過 2 小時的整合作業。開發人員提供的 Java 代碼通常無法讀取或寫入檔案系統或 Java 系統屬性。開發人員提供的 Java 代碼無法產生新的 Java 執行緒。

安全性與隱私權

Studio 會自動支援 2 種雲端環境：

- 沙箱
- 正式作業

您可以新增連至其他 Workday 環境的連線。您應自行負責確保將正確的整合部署到適當的環境。標準 Workday Web 服務安全原則會套用至使用 Studio 整合存取的所有資料。

您可以在 Studio 中離線建立組件，但必須具有適當的安全性配置，才能將組件與 Workday 整合。部署完成後，整合會使用執行者的憑證和權限啟動及執行。

Workday 無法深入了解您的整合功能，或其整合的設計用途。您有責任確保其運作符合預期。整合與 Workday 用戶的互動是透過 Workday 公開 API 進行，因此您會繼承所有相關聯的穩健性和安全性保證。

概念：整合開發生命週期

您可以視情況使用 Workday Studio 或 Workday 應用程式，在以下不同階段開發和監控 Studio 整合：

階段	說明
整合設計	在 Studio 中，開發構成整合的程序集。您也可以建立組合模組，以便重複使用及共用整合模式和配置。 每個程序集至少有 1 個 Workday-in 傳輸 (表示執行開始的位置)。每個 Workday-in 調動系統都可選擇定義： - 啟動參數，讓用戶端能夠將資訊做為輸入傳遞至整合。範例：薪資整合可能會指定其需要 2 個啟動參數：payPeriod 和 payGroup。這可讓整合使用者在計算薪資時指定月份和正職員工群組。 - 程序集在執行階段使用的 Workday 服務。Workday 為屬性對應、序列產生、交易記錄、文件擷取和交付提供通用整合服務。它還提供偵聽器服務、報告服務和自訂物件服務。
整合部署	在 Studio 中，連線至 Workday 雲端資料庫以部署您的整合。對於整合中定義的每個 Workday-in 傳輸，Workday 會註冊一個稱為整合系統的抽象概念。整合系統包含任何必要的啟動參數和服務。
整合配置	在 Workday 中，您可以配置整合的某些方面，包括權限和通知。
整合啟動	整合系統可讓 Workday 啟動並追蹤您的整合。但為了更輕鬆地進行開發和測試，您也可以從 Studio 啟動。
整合監控與疑難排解	在 Studio 或 Workday 中追蹤整合進度。在 Studio 中，您可以在「流程監控」上監控整合的執行情況。您也可以整合流程完成後檢視其合併報告。

概念：MediationContext 物件

在程序集執行期間，Workday 會建立一個名為「」的臨時支架物件。MediationContext。該 MediationContext 包含 3 種主要類型的資訊：

- MediationMessage：一種物件，內含文字或二進位資料訊息，Studio 最終會使用「傳出」傳送至外部目標。範例：HTTP-Out、FTP-Out。
- 變數：訊息資料的臨時持有者。
- 屬性：名稱-物件組的對應，組合元素可用來在訊息外儲存臨時資料。

程序集中的所有處理都涉及從「」讀取或寫入 MediationContext。

註：當您存取 MediationContext 使用 MVEL 的物件，則存取者為 context。範例：
`context.getErrorMessage()`。在 Java 中為 mediationContext。範例：
`mediationContext.getErrorMessage()`。

概念：整合系統和事件

整合系統

在 Workday 中，整合系統是與外部系統或 Workday 端點整合的抽象化。整合系統提供了 Workday 用來啟動和追蹤整合的架構。您可以使用整合系統來配置：

- 安全性：指定誰可以啟動整合，以及執行整合時所擁有的權限。
- 通知：指定誰會在整合執行時收到通知。
- 參數：您可以參數化整合的呼叫。例如，您可以配置整合所需的端點和使用者憑證，以便將資料上傳至外部 SFTP 網站。
- 觸發器：您可以配置要觸發的整合：
 - 手動
 - 包含在訂閱中
 - 使用 API
 - 在排程中。
 - 做為業務流程的一部分

Workday 將所有整合 (無論來源為何) 表示為整合系統。

當您使用 Workday Studio 開發整合併將其部署至 Workday 時，您會為該整合中的每個 Workday-in 傳輸自動註冊新的整合系統。

整合事件

當整合啟動時，Workday 會建立一個稱為整合事件的物件。此物件對每個呼叫都是唯一的，並包含稱為整合事件 ID 的 ID。Workday 會使用此唯一 ID 來追蹤所有正在執行的整合作業。您可以使用 Workday 中的 流程監控 來檢視整合事件。

Workday 會在建立整合事件物件的同時，建立整合事件要求訊息，並將其傳送至整合。此要求包含有關整合系統和整合事件 ID 的資訊。其中還包含啟動時提供的任何參數的值。然後，當從 Workday Web 服務要求資訊時，整合功能就可以使用此資訊。整合也可使用此資訊將更新資訊傳回至 Workday。例如：一條狀態訊息，告訴使用者已成功處理 100 筆正職員工記錄。Workday 會將這些更新新增至整合事件物件。

參照：介面基礎知識

Workday Studio 介面由多個檢視組成。若要顯示或隱藏檢視，請選取 視窗 > 顯示檢視 > 其他...，然後展開 Workday 資料夾。

Studio 中最常用的檢視如下：

檢視	說明
Project Explorer	Studio 的專案瀏覽器顯示儲存在目前工作區中的專案。每個資料夾都包含該專案的所有相關檔案，並參照該專案所屬的集合。
Cloud Explorer	顯示 Studio 可部署至的整合雲端伺服器，以及目前已部署的集合。您可以將人脈新增至多個環境。如果發生連線、驗證或授權錯誤，Studio 會在連線圖示上顯示錯誤或警告裝飾。
調色盤	顯示您可以新增的所有組合體元件。具有 3 個分頁子檢視： • 設計：顯示組合圖。 • 來源：顯示 XML 來源。

檢視	說明
	訊息建置器：圖形檢視，可簡化定義程序集元素訊息內容的流程，例如 寫入 步驟。
集合編輯器	集合的圖示。若要新增元件和步驟，請從 選用板 拖放。若要連接元件，請將滑鼠移至來源元件上方以顯示連線控點，然後將控點拖曳至目標元件。
屬性	顯示目前在組合圖中選取的組合元素的配置詳細資料。使用沿檢視左側顯示的分頁編輯屬性和參數。
流程監控	在您啟動整合時顯示組合流程的相關資訊。選取要在「流程詳細資料」檢視中顯示其詳細資料的流程。
結構描述資源管理器	可讓您探索 WSDL 和 XML 結構描述檔案的內容，並提供多項實用工具功能，用於開發與 Web 服務互動的程序集。
合併報告檢視器	可讓您分析整合中的組套件處理步驟。

整合設計

組套件

建立集合專案

執行這項作業的原因和時機

程序集定義了以訊息為基礎的整合流程，其中包含註解、傳輸、中介元件、中介步驟和錯誤處理程式。

程序

1. 選取「檔案 > 新的 > Workday 程序集專案」。
2. 在 專案名稱 欄位中，輸入您的專案名稱。
專案名稱不可包含空格。
3. 從「目標執行階段」清單中，選取 *Workday Runtime*。
4. 從 配置 清單中，選取下列其中一個配置選項：

選項	敘述
集合體	建立簡單組合專案。
程序集 (具有 Java 支援)	建立包含 Java 多面向的專案。
自訂	可讓您指定組合專案的專案多面向。

5. 按一下下一步。請輸入 集合名稱「」。
「集合名稱」欄位會自動填入您的專案名稱。
6. 按一下「完成」建立您的專案。

註：如果您在工作區之外建立集合專案，則可以在 Studio 中為其指定新名稱，而無需變更原始位置中的資料夾名稱。這也適用於您從工作區外部匯入的專案。為此，請選取「檔案 > 重新命名」。

結果

您的新組合專案會顯示在 Project Explorer 檢視中。其中包含一個 assembly.xml 檔案，您現在可以在「Assembly Editor」中開啟該檔案。

下一步

使用集合編輯器，您可以新增和連結傳輸、中介元件和中介步驟，以定義您的整合。

將步驟新增至元件

開始之前

建立程序集專案。

執行這項作業的原因和時機

當訊息經過整合時，步驟會在訊息上執行個別作業。您只能將步驟新增至 異步中介 和 同步中介 元件。

程序

1. 在集合編輯器中開啟集合。
2. 將 async-mediation 或 Sync-mediation 元件新增至程序集。
3. 將步驟從 選用板 拖曳到您的 異步中介 或 同步中介 元件中。請按照執行順序排列步驟。執行此任務時，請參考以下說明：

調色盤種類	說明
轉換	顯示可讓您在訊息經過整合時轉換這些中介步驟。
記錄與事件	顯示可讓您在訊息經過整合時記錄訊息的中介步驟。
訊息處理	顯示可讓您在訊息經過整合時控制訊息的中介步驟。
驗證	顯示可讓您在訊息經過整合時驗證訊息的中介步驟。
編碼	顯示可讓您在訊息經過整合時對訊息進行編碼的中介步驟。
其他	顯示可讓您在訊息經過整合時對這些訊息執行個別作業的雜項中介步驟。

4. 按一下儲存。

結果

Studio 會在組合圖上顯示元件內的新步驟。

下一步

配置新步驟的屬性。

新增子系元素

開始之前

建立程序集專案。

執行這項作業的原因和時機

某些集合元素支援子系元素。子元素為選用元素。

程序

1. 在集合編輯器中開啟集合。
2. 將支援子系元素的組件元素新增至組件圖表。
3. 將游標移至組合元素標題上方，可顯示其子系元素選項。
4. 按一下子系元素選項，將其新增至您的組合元素。

結果

Studio 會在您的組合元素上顯示子系元素。

下一步

配置子系元素的屬性。

連接集合元素

開始之前

建立程序集專案。

執行這項作業的原因和時機

程序集元素之間的連結會定義訊息在程序集中的路徑。您可以在集合編輯器中，使用 傳出連線 控點 (藍色方塊) 或 傳入連線 控點 (橙色方塊) 來連接集合元素。

程序

1. 在集合編輯器中開啟集合。
2. 將 2 個組件元素新增至組件圖表。
3. 將游標移至第一個組合元素的標題上方，以顯示其節點控點。將傳出連線 控點 (藍色方塊) 從第一個集合元素拖曳至第二個集合元素。
4. 或者，將游標移至第二個組合元素的標題上，然後 將傳入連線 控點 (橙色方塊) 拖曳至第一個組合元素。

結果

第一個組合元素的輸出訊息會變成第二個組合元素的輸入。

配置集合元素屬性

開始之前

將集合元素新增至集合圖表。

執行這項作業的原因和時機

Workday Studio 選用板 中的每個元素都有一個「屬性」檢視，您可以在其中配置其功能。

程序

1. 在組合元素上按一下滑鼠右鍵，然後選取 顯示屬性檢視。
2. 必填屬性的圖示左上角會有星號。執行此任務時，請參考以下說明：

屬性圖示	屬性說明
	值在程序集中必須是唯一的。
	集合元素的輸入
	組合體的輸出。
	MVEL 運算式
	實際應用程式語言清單範本
	布林值
	選項
	數值。
	屬性內容值
	Java 類別實作。
	參照 Java 類別實作的 Spring bean。

註：當屬性內容支援 MVEL 時，相關的 MVEL 圖示會新增到屬性內容本身的圖示上。範例：

3. 若要比較屬性檢視：
 - a) 在組合圖中選取元件。
 - b) 按住 Ctrl 鍵的同時選取第二個元件。
 - c) 在任一元件上按一下滑鼠右鍵，然後選取「比較屬性」。第一個元件的 屬性 檢視會顯示在左側。第二個元件的「屬性」檢視會顯示在右側，且已釘選至該選取項目。當您選取不同的元件時，左側的檢視會變更。
4. 在具有 MVEL 功能的屬性欄位中，按 Ctrl+空格鍵可存取內容小幫手，這是一組內容相關的代碼完成選項。

註：內容輔助僅適用於包含 Java 多面向的組合專案。

管理收款和專案

開始之前

建立程序集專案。

執行這項作業的原因和時機

在 Workday Studio 中，集合可讓您封裝和部署多個專案。專案一律至少屬於 1 個集合，且可同時屬於多個集合。

程序

1. 按一下 Workday > 管理收款。

如果您在左側窗格中展開集合，則 Studio 會顯示其中包含的專案。不屬於該集合的專案會顯示在右側的視窗窗格中。

2. 執行此任務時，請參考以下說明：

按鈕	說明
新產品系列	將新的集合新增至您的工作區。
重新命名集合	重新命名選取的集合。
移除收款	將選取的集合從您的工作區移除。
新增至集合	將選取的專案新增至集合。
將所有整合新增至集合	將所有專案新增至選取的集合。
從集合中移除整合	將選取的專案從集合中移除。

註：請確保集合中的所有專案都具有相同的程序集版本。否則，Studio 會顯示錯誤標記。您可以在程序集編輯器的「屬性」檢視中配置程序集的「版本」屬性值。

3. 按一下 完成。

宣告名稱空間字首

開始之前

建立程序集專案。

執行這項作業的原因和時機

在 Workday Studio 中，名稱空間字首是包含預先定義 URI 值的縮寫。您可以聲明名稱空間字首，以參照程序集中的特定 URI 值。

程序

1. 在集合編輯器中開啟集合。
2. 以滑鼠右鍵按一下組合圖的背景，然後選取「顯示屬性檢視」。
3. 按一下 命名空間字首 分頁。
4. 按一下 新增，新增名稱空間字首。執行此任務時，請參考以下說明：

選項	敘述
字首	在此輸入字首名稱。
名稱空間	在此輸入名稱空間 URL。

5. 若要編輯新的名稱空間字首，請按一下 編輯。
6. 若要移除新的名稱空間字首，請按一下「移除」。

結果

程序集的名稱空間字首會顯示在 XML 結構描述中。

建立路徑變數

開始之前

建立程序集專案。

執行這項作業的原因和時機

路徑變數會指定資源檔案在工作區資料夾中的位置。當您在組合專案中連結平台資源時，可以使用工作區路徑變數。

程序

1. 選取 Windows > 喜好設定的 > 一般 > 工作區 > 連結的資源。
2. 按一下 新增。

執行此任務時，請參考以下說明：

選項	敘述
名稱	輸入路徑變數的名稱。
地點	輸入路徑變數的檔案位置。

3. 按一下 確定。

結果

新的路徑變數現在會指定資源檔案在工作區資料夾中的位置。

概念：集合調色板

調色盤 會顯示 Studio 中所有可用的程序集元素。若要將集合元素新增至集合圖表，請將其從 選用板 拖曳到集合編輯器上。「調色盤」列出以下種類下的集合元素：

調色盤種類	說明
我的最愛	顯示您常用的集合元素。按一下集合元素旁邊的空白星號圖示，即可將其新增至此種類。按一下元素旁邊以醒目提示標出的星號圖示，即可從此種類中移除元素。
註解	顯示可讓您組織、整理和標記組合圖的註解。
模組庫	顯示模組組件元素，方便您為整合配置 Workday Web 服務呼叫和全域錯誤處理常式。
傳輸中	顯示從 Workday 或本機資源接收訊息的在途程序集元素。
外傳	顯示可傳送訊息至資源的外運裝配件元素。
元件	顯示可讓您在訊息經過整合時操縱和轉換訊息的中介元件。
通用元件	顯示可用來建置整合的預先封裝子組合體。
工作區元件	顯示您已建立並儲存至工作區的自訂子裝配件。
錯誤處理程式	顯示可讓您擷取整合中錯誤的錯誤處理常式。
轉換	顯示可讓您在訊息經過整合時轉換這些中介步驟。

調色盤種類	說明
記錄與事件	顯示可讓您在訊息經過整合時記錄訊息的中介步驟。
訊息處理	顯示可讓您在訊息經過整合時控制訊息的中介步驟。
驗證	顯示可讓您在訊息經過整合時驗證訊息的中介步驟。
編碼	顯示可讓您在訊息經過整合時對訊息進行編碼的中介步驟。
其他	顯示中介步驟，方便您在訊息經過整合時對這些步驟執行個別作業。
連結	顯示可讓您定義透過整合的訊息路徑的「傳送至」和「傳送回覆至」連線。

若要手動篩選 調色盤上顯示的集合元素，請使用 類型篩選文字 欄位。Studio 還在 類型篩選條件文字 欄位下方提供以下預先配置的種類篩選條件：

種類篩選條件	說明
還原	清除先前的篩選條件選擇。
傳輸	僅顯示傳輸 調色盤 種類。
元件	僅顯示中介元件 選用板 種類。
步驟	僅顯示中介步驟 調色盤 種類。
處理程式和連線	僅顯示錯誤處理常式和連線 選用板 種類。
篩選種類	顯示「調色盤種類篩選條件」視窗。
全部展開	展開所有 調色盤 種類。
全部收合	收合所有 調色盤 種類。

Studio 會自動在程序集編輯器旁顯示 調色板。若要在個別檢視中顯示 調色盤，請選取 視窗 > 顯示檢視 > 其他 > 一般 > 調色盤，然後按一下 確定。您也可以調色盤上按一下滑鼠右鍵，然後選取「版面配置」，即可變更 調色盤 的版面配置。

relatedlinks

相關參考資料

[Workday 31 功能更新公告：Workday Studio](#)

概念：元件和步驟

您只能將步驟新增至 異步中介 和 同步中介 元件。

Workday Studio 提供的步驟可支援轉換、驗證和記錄等作業。您也可以使用 Spring bean 撰寫自訂步驟。

async-mediation 元件支援 1 個處理鏈，僅處理要求訊息。同步中介 元件支援 2 個處理鏈，即處理要求訊息和要求-回應訊息。

步驟具有「輸入」和「輸出」屬性，可指定步驟從何處取得資料做為輸入，以及從何處導向輸出。有 5 個可能的值：

選項	說明
訊息	整封訊息，包括根部分和附件。當您指定此選項時，Studio 會建立新訊息，有效移除所有附件。
<i>soapbody</i>	訊息根部分的 SOAP 內文內容。
<i>attachment</i>	訊息的 MIME 附件。請指定您要存取的附件的索引值。您也可以指定索引值為 -1，將新附件新增至訊息。
<i>rootpart</i>	MIME 訊息的根目錄部分的內容，如果不是 MIME 訊息，則是訊息本身。當為輸出屬性內容指定此值時，系統會保留原始訊息中的所有附件。
變數	中的變數 <code>MediationContext</code> 。指定要使用的變數名稱。

概念：集合註解

註解可讓您對組合圖進行結構、組織和標記。Studio 在 調色盤的 註解 區段下提供以下選項：

- 矩形
- 橢圓
- 附註
- 文字
- 泳道

包括：

- 使用「矩形」、「橢圓」、「附註」和「文字」註解，將一般說明文字新增至您的集合。
- 使用「顯示/隱藏註解」按鈕可顯示或隱藏「矩形」、「橢圓」、「附註」和「文字」註解。只有當「顯示註解」為作用中時，您才能將新的註解新增至組合圖。
- 使用「矩形」、「橢圓」、「附註」和「文字」註解上的圖釘圖示，將其釘選至集合。即使在「隱藏註解」啟用時，釘選的註解也會一律顯示在您的圖表上。

泳道

泳道可讓您將集合元件編排成群組。

包括：

- 按一下 顯示/隱藏泳道說明 按鈕，即可顯示或隱藏泳道說明。
- 按一下 展開所有泳道和收合所有 泳道按鈕，即可展開和收 合所有泳道。
- 按一下標頭中的加號和減號圖示，即可展開和收合個別泳道。「泳道」狀態為持續顯示。當您開啟集合時，泳道會以儲存集合時的收合或可見狀態顯示。

泳道可內嵌。

`relatedlinks`

相關參考資料

[Workday 31 功能更新公告：Workday Studio](#)

概念：組合專案的相關資源檔案

連結的資源是儲存在工作區中的檔案，您可在多個程序集專案中連結到這些檔案。

若要檢視專案的連結資源清單，請在「Project Explorer」檢視中的專案上按一下滑鼠右鍵。選取「屬性 > Workday > 連結的資源」，顯示「連結的資源」視窗。

若要從專案中移除連結的資源檔案，請在「Project Explorer」檢視中的資源檔案上按一下滑鼠右鍵，然後選取「移除連結的資源」。

若要將連結的資源複製到專案，請在 Project Explorer 檢視中的資源檔案上按一下滑鼠右鍵，然後選取 將連結的資源轉換為要復制的連結資源。

概念：快速大綱檢視

「快速大綱」檢視提供了一種找出集合中各種元素的簡單方法。若要存取「快速大綱」檢視，請在 集合編輯器 具有焦點時按 Ctrl+O。

註：您無法從「視窗 > 顯示檢視」功能表存取 快速大綱 檢視。

「快速大綱」檢視可讓您透過以下方式搜尋程序集：

搜尋類型	說明
MVEL 運算式	搜尋 MVEL 運算式和範本屬性。使用此檢視可搜尋出現的文字值。
ID	搜尋組合體元件 ID。
變數	搜尋變數名稱。Studio 會將變數分為 2 組：針對存取步驟定義的變數，以及針對寫入步驟定義的變數。

若要選取搜尋類型，請按一下檢視頂端搜尋欄位下方的相關按鈕。

若要進行搜尋，請在搜尋欄位中輸入文字。當您在搜尋結果清單中選取 MVEL 運算式、元件 ID 或變數時，Studio 會跳至組件圖中的相關元件。

參照：組合屬性

屬性	XML 屬性內容名稱	說明
ID	id	程序集的唯一 ID。
版本	version	程序集版本號。
稽核記錄	audit-log	程序集的訊息記錄設定。選項包括： - True：啟用並儲存所有記錄。 - False：停用所有記錄。 - 發生錯誤時：啟用所有記錄，且僅儲存錯誤記錄。
回調處理常式 Bean	callback-handler-bean	處理程序集的異步通訊的 Spring bean。
豆類	beans	程序集中的 Spring bean。
命名空間字首	xmlns	程序集中的名稱空間字首。

參照：泳道屬性

通用分頁

屬性	說明
名稱	泳道的名稱。顯示在泳道的標頭中。
說明	泳道的說明。在您選取「顯示泳道說明」時顯示。

屬性	說明
方向	泳道的方向。選項包括： • 水平 • 垂直
對齊方式	泳道中元件的對齊方式。選項包括： • 開始 • MIDDLE • END
標籤對齊方式	泳道標籤的對齊方式。選項包括： • LEFT • CENTER • 右
上框線色彩	泳道上框線的色彩。
線條色彩	泳道側邊和底邊框線的顏色。
填滿色彩	泳道的背景色彩。
字型	泳道標籤的字型。
線型	泳道側邊和底邊的樣式。選項包括： • 實線 • DASH • DOT • DASHDOT • DASHDOTDOT

參照：連線行屬性

通用分頁

屬性	說明
明細類型	此線類型用於在組合圖表上顯示「發送至」和「發送回覆」連結方。選項包括： • straight • 彎曲

裝配件傳輸

在 Studio 中使用列舉做為啟動參數

開始之前

建立包含 Workday-In 傳輸的集合專案。

執行這項作業的原因和時機

在 Workday Studio 中，您可以使用列舉做為整合啟動參數。這會限制您在啟動整合時可以在「整合條件」分頁中選取的值。您可以在 Studio 中定義列舉，或參照用戶中現有的列舉定義。

程序

1. 在集合編輯器中開啟集合。
2. 在 Workday-in 傳輸的「屬性」檢視的「啟動參數」分頁上，按一下 新增列舉類型參數。
3. 按一下 編輯類型。
4. 配置列舉時，請參考以下說明：

選項	敘述
列舉參照	請輸入用戶中現有列舉定義的 類型名稱 。
列舉定義	請為列舉定義輸入不重複的 類型名稱 。
	註：當您輸入現有列舉定義的 類型名稱 時，Workday 會覆寫該定義。
	新增列舉允許的值。

下一步

部署您的整合。

[relatedlinks](#)

[相關任務](#)

[將整合部署至 Workday 頁 247](#)

概念：集合傳輸

Studio 在 調色盤上提供了 2 種類型的程序集傳輸：

- 傳輸中
- 外傳

In-transports 會接收來自 Workday 或本機資源的訊息，並將這些訊息轉送至集合中定義的其他元素，以便處理。傳輸中一律表示處理鏈的開始。

Out-transport 會與 HTTP 端點、Workday 應用程式或 FTP 網站等資源交換訊息。包含 本機傳出 傳輸的程序集，可與包含 本機輸入 傳輸的子程序集通訊。

Studio 支援 調色盤上的這些傳輸中程序集元素：

名稱	函數	附註
workday-in	定義組合執行的起始點。	• 偵聽 Workday 應徵要求訊息。 • 支援整合的啟動參數和服務定義。 • 同步 (要求-回應)。
local-in	定義組合體或子組合體的進入點。	• 與包含 本機傳出 傳輸的程序集交換訊息。 • 支援參數和輸出參數。 • 同步 (要求-回應)。

Studio 支援 調色盤上的以下外傳組套件元素：

名稱	函數	附註
workday-out-rest	將 Workday 應用程式 REST 要求訊息傳送至 HTTP URL。	• 同步 (要求-回應)。
workday-out-soap	將 Workday 應用程式 SOAP 要求訊息傳送至 HTTP URL。	• 同步 (要求-回應)。

名稱	函數	附註
local-out	與包含 本地輸入 傳輸功能的組 合件及子組件交換訊息。	• 同步 (要求-回應)。
http-out	將 HTTP 要求訊息傳送至 HTTP URL。	• 支援以下 HTTP 子系元素 ： • http-basic-auth • http-custom-auth • https-properties • 同步 (要求-回應)。
ftp-out	將訊息傳送至 FTP 伺服器。	• 非同步 (僅限要求)。
ftps-out	使用 FTP over SSL (FTPS) 通訊 協定，將訊息安全地傳送至 FTP 伺服器。	• 非同步 (僅限要求)。
sftp-out	使用 FTP over SSH (SFTP) 通 訊協定，將訊息安全地傳送至 FTP 伺服器。	• 非同步 (僅限要求)。
xmpp-out	將文字訊息寫入 Jabber 伺服 器。範例：GoogleGmail 伺服 器。	• 非同步 (僅限要求)。
custom-out	建立可自訂的出貨單。	• 同步性由 Spring bean 實 作定義。
電子郵件寄出	將電子郵件寫入 SMTP (郵件) 伺 服器。	• Studio 會檢查以下屬性中 是否有前置和尾端空白： • 收件者 • 主旨 • 開始 • 密件抄送 • Cc • 主機 • 連接埠 • 使用者 • 密碼 • 回覆 • 非同步 (僅限要求)。

概念：列舉

列舉將屬性內容值限制為一組常值，例如成本中心代碼。在設計整合時，您可以使用列舉來定義一組常數，以做為常值。這可讓您限制整合使用的資料範圍。

您可以使用以下任務在 Workday 中建立和維護列舉定義：

- 建立整合列舉定義
- 編輯整合列舉定義
- 刪除整合列舉定義
- 檢視整合列舉定義

您可以在下列元素中使用列舉：

- 啟動參數
- 整合屬性內容
- 整合對應

您可以指派列舉：

- 在 Studio 中
- 在 Workday 中使用「建立整合通用服務」任務。

您可以使用下列其中一種方法，在 Studio 整合中指派列舉：

- 在 Studio 中建立列舉定義，並在啟動 Studio 整合時將其部署至您的用戶。
- 建立對您用戶中現有列舉定義的列舉參照。

您可以選取 1 個整合來提供定義，並在其他整合中參照該定義，即可在多個整合中使用相同的列舉定義。

relatedlinks

相關任務

[建立屬性對應服務](#) 頁 226

參照：Workday-In 傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	Workday-in 調動在集合內的唯一 ID。
路由至	routes-to	Workday-in 傳輸系統將訊息導向的程序集元素。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	指定個別程序集的替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
整合系統名稱	integration-system-name	您的整合系統名稱。

啟動參數分頁

啟動參數	說明
簡單類型參數	使用字串常值的啟動參數
參照類型參數	使用 Workday CRF 和 Web 服務的參照啟動參數。
列舉類型參數	啟動使用 Workday CRF 和列舉定義的參數

服務分頁

整合服務	說明
屬性對應服務	整合中使用的整合屬性內容和整合對應。
交付服務	將文件從您的 Workday 用戶傳送至目標端點。
偵聽器服務	讓您的整合功能透過 Workday 用戶接收第三方用戶端的啟動訊息。
擷取服務	從遠端用戶端擷取文件，並將其儲存在您的 Workday 用戶中。
序列產生器服務	每次執行整合時，系統都會為文件產生唯一且有序的檔案名稱。
異動記錄服務	顯示您的 Workday 用戶中整合系統已訂閱的異動事件記錄。
報告服務	指定您在整合中參照的任何自訂報告的別名。
自訂物件服務	指定您在整合中參照的任何自訂物件的別名。
服務參考	參照在同一個集合內的另一個 Workday-in 傳輸中配置的任何整合服務。

參照：使用當地文字輸入傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	本機輸入 傳輸在程序集中的唯一 ID。
路由至	routes-to	以 當地文字輸入的 傳輸功能將訊息導向的程序集元素。
存取	access	指定您的子裝配件是 公開 還是 私人。 調色盤 僅顯示屬於 工作區元件類型的公開子裝配件。
使用全域錯誤處理程式	use-global-error-handlers	決定是否在子裝配件層級處理子裝配件內發生的錯誤。預設值為 false，表示錯誤會由呼叫子組合件的程序集處理。
圖示	icon	指定子裝配件的自訂圖示。支援 24x24 像素的 PNG 或 GIF 圖示。
工具提示	tooltip	子裝配件的工具提示。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	「儲存訊息」不適用於本機輸入傳輸。Studio 會忽略本機輸入傳輸的此設定。
傳輸類別	transport-class	指定個別程序集的替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。

參數分頁

使用本機輸入的傳輸參數指定任何 MediationContext 可設定的屬性

屬性	XML 屬性內容名稱	說明
名稱	name	輸入當地文字 的參數名稱。
文件	documentation	輸入 當地文字 參數的文字說明。
必填	required	決定您的 以當地文字輸入的 傳輸方式是否需要您的參數。
類型	type	指定您的參數類型。選項包括： - 字串 <i>integer</i> - 長 - 加倍 <i>float</i> - 布林值
預設值	default	參數的預設值。
驗證	validation	布林 MVEL 運算式，評估結果必須為 true，否則 Studio 會引發錯誤。可讓子組合件開發人員對參數值施加限制。

輸出參數分頁

本端輸入 傳輸的輸出參數指定了任何 MediationContext 子組合體擷取的屬性。

屬性	XML 屬性內容名稱	說明
名稱	name	出參數的名稱。
文件	documentation	出參數的文字說明。

參照：Workday-Out-Rest 傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	Workday-out-REST 交通方式在集合中的唯一 ID。
將回覆送交至	routes-response-to	Workday-out-REST 傳輸的回應訊息目的地。
執行時間	execute-when	一個條件，用於判斷是否執行 Workday-out-REST 交通規則。
額外路徑	extra-path	指定 Workday 用戶中 Studio 用來傳送 REST 要求的路徑。Studio 會自動為此路徑設定並附加 Workday 服務端點。範例：以下路徑中的端點會從名為 ACME 的用戶中呼叫 Workday Business Site Directory 報告，並以 JSON 格式要求該報告： systemreport2/ACME/Business_Site_Directory?format=json

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	指定個別程序集的替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
方法	method	可讓您將 REST 方法新增至 Workday-out-REST 傳輸。範例：put, get, post。
失敗訊息	failure-message	自訂訊息，用於取代您在呼叫 Workday-out-REST 端點時傳回的任何例外訊息。

參照：Workday-Out-Soap 傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	Workday-out-soapTransport 在程序集中的唯一 ID。
將回覆送交至	routes-response-to	Workday-out-soap 傳輸的回應訊息目的地。
執行時間	execute-when	一個決定是否執行 Workday-out-soap 傳輸的條件。
應用程式	application	Workday 應用程式的名稱，傳輸系統會將內容傳送到該應用程式。Studio 會使用您在此欄位中輸入的名稱來建立 URI，以指定 Workday 應用程式的位置。
版本	version	SOAP 要求呼叫的 Web 服務版本。 註：當您在 XSLT 要求中指定 Web 服務版本時，將優先使用該版本。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
失敗訊息	failure-message	自訂訊息，用於取代您在呼叫 Workday-out-soap 端點時傳回的任何例外訊息。
取代為 SOAP 錯誤	replace-with-soap-fault	以 SOAP 錯誤取代失敗的 Workday-out-soap 訊息。SOAP 錯誤包含擷取的 faultcode 和 faultstring 屬性值

參照：本端傳出傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	程序集內 本端傳出 傳輸的唯一 ID。
將回覆送交至	routes-response-to	本地傳出 傳輸的回應訊息目的地。
執行時間	execute-when	決定是否執行 本端傳出 傳輸的條件。
端點	endpoint	本地傳出 傳輸系統將訊息導向的目標端點地址。 端點位址遵循以下格式 application-name/ component-ID，其中： • application-name 是包含目標程序集的專案名稱。 • component-ID 是目標元件的 ID。 範例：<cc:local-out id="local" endpoint="myassembly/mediation1"/>

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存 步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要建立由 本地傳出 傳輸器處理的要求訊息的副本，如

屬性	XML 屬性內容名稱	說明
		此任何修改訊息的子組物件都不會影響其他子組合。
傳播中止	propagate-abort	指定中止標幟是否會從子組物件傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組物件後，是否重設所有 本地輸出 屬性。

參數分頁

屬性	XML 屬性內容名稱	說明
名稱	name	參數子系元素的名稱。
值	value	參數子元素的值。

參照：HTTP 傳出傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	程序集內 http-out 傳輸的唯一 ID。
將回覆送交至	routes-response-to	http-out 傳輸的回應訊息目的地。 註: http-out 傳輸會填入 MediationContext 屬性 http.response.status 並使用遠端伺服器傳回的標準 HTTP 回應代碼。範例：202 (接受)、403 (禁止)。程序集可使用此屬性中的值來決定不同回應的處理方式。
執行時間	execute-when	決定是否執行 http-out 傳輸的條件。
端點	endpoint	http-out 傳輸裝置將要求訊息導向的目標 HTTP URL 位址。 範例：http://myhost.example.org/MyService
Http 方法	http-method	http-out 傳輸支援的 HTTP 要求方法。範例：DELETE, GET, POST, PUT, 和 PATCH。 註: 若要使用 PATCH 方法，您必須設定 wd.http.client 屬性 apache。如果您不希望下游傳

屬性	XML 屬性內容名稱	說明
		輸同時使用 Apache HTTP 用戶端，請重設 wd.http.client 屬性 null。 如果您未為此屬性指定方法，則 Studio 會使用 POST 方法

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
連線逾時	connect-timeout	指定與 HTTP 伺服器建立連線的逾時期間 (以毫秒為單位)。如果您未指定值，則階段作業會在 300000 毫秒 (5 分鐘) 後逾時。
回覆逾時	response-timeout	指定等候 HTTP 伺服器回應的逾時期間 (以毫秒為單位)。如果您未指定值，則階段作業會在 21600000 毫秒 (6 小時) 後逾時。 如果 HTTP 連線無回應的狀態超過此期間，則 http-out 傳輸失敗。
錯誤為回覆	error-as-response	可讓您將錯誤回應訊息從伺服器複製到 http-out 傳輸上的訊息。
關閉連結	close-connection	新增 Connection: close 標頭附加到傳出的 http-out 傳輸訊息。
串流	streaming	啟用 HTTP 串流 PUT 和 POST HTTP 方法
自動處理中	auto-inproc	指定是否要對 http-out 要求訊息使用進行中的傳輸通訊。讓您能夠直接傳送訊息至位於同一伺服

屬性	XML 屬性內容名稱	說明
		器內的程序集端點，而無需使用本機主機的網路 I/O 功能。
重試次數	retries	指定發生傳輸例外時 http-out 傳輸要重試的次數。
重試延遲	retry-delay	指定每次重試嘗試之間的延遲期間 (以毫秒為單位)。 註: Workday 符合以下要求：Retry-After 或 RateLimit-Reset 外部端點的 429 太多要求回應中的標頭值，但前提是該值小於預設的最長 5 分鐘，或 retry-delay 此處指定的值。 如果缺少回應標頭值，Workday 會使用 retry-delay 值。 如果兩者 retry-delay 缺少標頭值時，Workday 會使用 5 分鐘的預設值。 如果標頭值大於 retry-delay 值和預設最大值
記錄重試次數	log-retries	讓您將 HTTP 重試嘗試記錄到伺服器記錄檔。
接受 Gzip	accept-gzip	指定是否包含 HTTP accept-encoding=gzip http-out 要求標頭中的欄位，此欄位表示是否接受回應中的 gzip 內容。
Gzip 內容	gzip-content	指定是否包含 HTTP content-encoding=gzip http-out 要求標頭中的欄位，此欄位表示是否要將傳出要求中的訊息內容壓縮壓縮檔。
接受	accept	指定 accept 隨 HTTP 要求傳送的標頭值，此值定義 HTTP 伺服器回應中可接受的 1 或多種內容類型。

註：對於 Amazon S3 傳輸，請使用以下金鑰將標籤新增至您的 IAM 使用者。 *Workday-type* 且值為 整合。此標記區分大小寫，且不會忽略空格。

參照：FTP 傳出傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	ftp 輸出 元件在程序集中的唯一 ID。
將回覆送交至	routes-response-to	ftp-out 傳輸的回應訊息目的地。

屬性	XML 屬性內容名稱	說明
執行時間	execute-when	決定是否執行傳輸的條件。
端點	endpoint	指定要寫入檔案的主機 (也可選擇傳輸埠) 和 FTP 目錄。 遵循格式 ftp://FTP-directory, 其中 FTP-directory 是目標位置。 範例: ftp://ftpserver.example.org/outputDir 註: 您指定的目錄位置必須已存在於 FTP 伺服器上。
二進制檔案	binary-file	指定是否要以二進位格式傳輸輸出檔案。
輸入檔案模式	input-file-pattern	檔案名稱模式, 用於比對特定類型的檔案。 範例: 指定 *.xls, 僅選取 Excel 檔案。
輸入內容類型	input-content-type	執行階段包含的媒體類型: content-type 訊息標頭中的值媒體類型至少包含 2 個部分: 類型、子類型和 1 或多個選用參數。該 text 子類型有選填項目 charset 參數, 指定字元編碼。 範例: text/html; charset=UTF-8 如果是非 XML 文字檔格式, 且檔案中並未明確指出媒體類型和字元集, 此屬性就特別有用。
輸出檔案模式	output-file-pattern	指定產生輸出檔案名稱的方式。模式可包含常值字串以及下列一或多個標記: • \${INFILE}, 取代為輸入檔案的名稱。 • \${UUID}, 取代為通用唯一 ID。 • \${EXT}, 放置在模式的結尾, 根據寫入組件元件輸出的資料內容類型, 以不同的副檔名取代。
方法	method	指定轉移方法: • put 「」: 將檔案寫入包含遠端目錄的 FTP 端點。 • list: 列出 FTP 端點上的所有檔案和目錄。將「輸入檔案模式」屬性搭配使用 list, 將模式套

屬性	XML 屬性內容名稱	說明
		用至要傳回的清單。您可以存取「」的結果 list 方法使用 MVELftp 物件。範例：ftp.list() 會傳回一個字串清單，其中包含最後擷取的清單。 • get：擷取第一個符合「輸入檔案模式」屬性值的檔案。 • delete：刪除找到的第一個符合「輸入檔案模式」屬性值的檔案。 註：刪除後，檔案不會自動從 FTP 網站刪除。get 已執行「」。如果您要刪除檔案，必須呼叫個別的方法。範例：您可以包含單獨的 ftp-out 專門傳送以刪除檔案。
密碼	password	與用來存取 FTP 伺服器的使用者名稱相關聯的密碼。
使用者名稱	username	用於存取 FTP 伺服器的使用者名稱。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
被動模式	passive-mode	指定傳輸是否支援被動模式下的 FTP。 在被動模式下，FTP 用戶端會起始 FTP 傳輸所需的兩個連線。用戶端會透過指令連接埠聯絡伺服器並發出 PASV 命令。伺服器

屬性	XML 屬性內容名稱	說明
		隨後會回覆，指出其正在偵聽哪個通訊埠以進行資料連線。用戶端接著起始資料連線，從其資料通訊埠連至指定的伺服器資料通訊埠。最後，伺服器會將 ACK 傳回至用戶端的資料通訊埠。在作用中模式下，用戶端會起始指令連線，而伺服器會起始資料連線。
逾時	timeout	指定用戶端與傳輸系統正在偵聽的 FTP 伺服器之間的 FTP 階段作業逾時期間 (以毫秒為單位)。如果您未指定值，則階段作業會在 300000 毫秒 (5 分鐘) 後逾時。
MIME 類型對應檔案	mime-types-map-file	MIME 類型與副檔名之間對應的對應檔案位置。使用此屬性可： - 當「方法」屬性設定為時，將 MIME 類型對應至檔案副檔名 put。 - 當「方法」屬性設定為時，將檔案副檔名對應至 MIME 類型 get。 此對應的檔案為 ccx/conf/mime.types。您可以提供您自己的此檔案版本。 此傳輸可使用 \${EXT} 放在結尾的標記 output-file-pattern 值。
臨時檔案名稱	temp-file-name	指定臨時檔案名稱。當「方法」屬性設為 put，檔案會使用此臨時檔案名上傳。FTP 完成後，檔案會重新命名為輸出檔案模式屬性指定的名稱。
關閉連結	close-connection	指定是否在 FTP 動作完成時進行中斷連接。
額度大小	pool-size	此元件在個別主機上可支援的同時連線和階段作業數上限。如果嘗試連線至同一主機的執行緒數超過指定數量，系統會強制部分執行緒等候存取池。 若要使分配池大小不受限制，請指定負數。
偵錯	debug	啟用傳輸中的 JSCAPE 偵錯模式。

參照：FTPS 傳出傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	ftps-out 元件在程序集中的唯一 ID。
將回覆送交至	routes-response-to	ftps-out 傳輸的回應訊息目的地。
執行時間	execute-when	決定是否執行傳輸的條件。
端點	endpoint	指定要寫入檔案的主機 (也可選擇傳輸埠) 和 FTP 目錄。 遵循格式 <code>ftps://FTPS-directory</code>，其中 <code>FTPS-directory</code> 是目標位置。 範例：<code>ftps://ftpserver.example.org/outputDir</code> 註：伺服器上必須已有您指定的目錄位置。
二進制檔案	binary-file	指定是否要以二進位物件的形式傳輸輸出檔案。
明確 SSL	explicit-ssl	指定連線類型：顯式 SSL (AUTH TLS) 或隱式 SSL (AUTH SSL)。設為 <code>true</code> 以使用明確的 SLL。 FTP 用戶端可以選擇像一般 FTP 階段作業一樣在未加密的情況下繼續通訊，或發出 AUTH 命令，切換為 FTPS 模式。
輸入檔案模式	input-file-pattern	檔案名稱模式，用於比對特定類型的檔案。 範例：指定 <code>*.xls</code>，僅選取 Excel 檔案。
輸入內容類型	input-content-type	執行階段包含的媒體類型：<code>content-type</code> 訊息標頭中的值媒體類型至少包含 2 個部分：類型、子類型和 1 或多個選用參數。該 <code>text</code> 子類型有選填項目 <code>charset</code> 參數，指定字元編碼。 範例：<code>text/html; charset=UTF-8</code> 如果是非 XML 文字檔格式，且檔案中並未明確指出媒體類型和字元集，此屬性就特別有用。

屬性	XML 屬性內容名稱	說明
輸出檔案模式	output-file-pattern	指定產生輸出檔案名稱的方式。模式可包含常值字串以及下列一或多個標記： • <code>\${INFILE}</code>，取代為輸入檔案的名稱。 • <code>\${UUID}</code>，取代為通用唯一 ID。 • <code>\${EXT}</code>，放置在模式的結尾，根據寫入組合件元件輸出的資料內容類型，以不同的副檔名取代。
方法	method	指定轉移方法： • <code>put「」</code>：將檔案寫入包含遠端目錄的 FTP 端點。 • <code>list</code>：列出 FTP 端點上的所有檔案和目錄。將「輸入檔案模式」屬性搭配使用 <code>list</code>，將模式套用至要傳回的清單。您可以存取「」的結果 <code>list</code> 方法使用 <code>MVELftp</code> 物件。範例：<code>ftp.list()</code> 會傳回一個字串清單，其中包含最後擷取的清單。 • <code>get</code>：擷取第一個符合「輸入檔案模式」屬性值的檔案。 • <code>delete</code>：刪除找到的第一個符合「輸入檔案模式」屬性值的檔案。 註：刪除後，檔案不會自動從 FTP 網站刪除。<code>get</code> 已執行「」。如果您要刪除檔案，必須呼叫個別的方法。範例：您可以包含單獨的 <code>ftp-out</code> 專門傳送以刪除檔案。
密碼	password	與用來存取 FTP 伺服器的使用者名稱相關聯的密碼。
使用者名稱	username	用於存取 FTP 伺服器的使用者名稱。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。

屬性	XML 屬性內容名稱	說明
		若要儲存訊息，請使用 Studio 的儲存 步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
被動模式	passive-mode	指定傳輸是否支援被動模式下的 FTP。 在被動模式下，FTP 用戶端會起始 FTP 傳輸所需的兩個連線。用戶端會透過指令連接埠聯絡伺服器並發出 PASV 命令。伺服器隨後會回覆，指出其正在偵聽哪個通訊埠以進行資料連線。用戶端接著起始資料連線，從其資料通訊埠連至指定的伺服器資料通訊埠。最後，伺服器會將 ACK 傳回至用戶端的資料通訊埠。在作用中模式下，用戶端會起始指令連線，而伺服器會起始資料連線。
逾時	timeout	指定用戶端與傳輸系統正在偵聽的 FTP 伺服器之間的 FTP 階段作業逾時期間 (以毫秒為單位)。 如果您未指定值，則階段作業會在 300000 毫秒 (5 分鐘) 後逾時。
MIME 類型對應檔案	mime-types-map-file	指定對應檔案的位置，以便在 MIME 類型與檔案副檔名之間進行對應。使用此屬性可： - 當「方法」屬性設定為時，將 MIME 類型對應至檔案副檔名 put。 - 當「方法」屬性設定為時，將檔案副檔名對應至 MIME 類型 get。 此對應的檔案為 ccx/conf/mime.types。您可以提供您自己的此檔案版本。

屬性	XML 屬性內容名稱	說明
		此傳輸可使用 <code>\${EXT}</code> 放在結尾的標記 <code>output-file-pattern</code> 值。
臨時檔案名稱	<code>temp-file-name</code>	指定臨時檔案名稱。當「方法」屬性設為 <code>put</code> ，檔案會使用此臨時檔案名上傳。FTP 完成後，檔案會重新命名為輸出檔案模式屬性指定的名稱。
關閉連結	<code>close-connection</code>	指定是否在 FTP 動作完成時進行中斷連接。
額度大小	<code>pool-size</code>	此元件在個別主機上可支援的同時連線和階段作業數上限。如果嘗試連線至同一主機的執行緒數超過指定數量，系統會強制部分執行緒等候存取池。 若要使分配池大小不受限制，請指定負數。
偵錯	<code>debug</code>	啟用傳輸中的 JSCAPE 偵錯模式。

參照：SFTP 傳出傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	<code>id</code>	sftp-out 元件在程序集中的唯一 ID。
將回覆送交至	<code>routes-response-to</code>	sftp-out 傳輸的回應訊息目的地。
執行時間	<code>execute-when</code>	指定一個條件，決定是否執行傳輸。
端點	<code>endpoint</code>	指定要寫入檔案的主機 (也可選擇傳輸埠) 和 FTP 目錄。 遵循格式 <code>sftp://SFTP-directory</code> ，其中 <code>SFTP-directory</code> 是目標位置。 範例： <code>sftp://sftpserver.example.org/outputDir</code> 註：伺服器上必須已有您指定的目錄位置。
雙重驗證	<code>dual-authentication</code>	指定是否同時使用使用者名稱/密碼組合和私人金鑰強制執行雙重驗證。預設值為 <code>false</code> ，這表示使用者可以使用私人金鑰或使用

屬性	XML 屬性內容名稱	說明
		者名稱/密碼組合進行驗證，但不能同時使用兩者。 如果屬性設為 false 且使用者輸入私人金鑰 (透過配置 sftp-out 元件的私人金鑰屬性)，則驗證會自動使用該金鑰值和使用者的名稱，而忽略密碼欄位。如果屬性設為 false 且未配置私人金鑰，則驗證會使用使用者名稱/密碼組合。 如果屬性設為 true 則驗證會同時使用私人金鑰值和使用者的名稱/密碼組合。
主機金鑰指紋	host-key-fingerprint	FTP 伺服器將用於 SSH 通訊的加密金鑰。
使用者名稱	username	用於存取 FTP 伺服器的使用者名稱。
密碼	password	與用來存取 FTP 伺服器的使用者名稱相關聯的密碼。
二進制檔案	binary-file	指定是否要以二進位格式傳輸輸出檔案。
明確 SSL	explicit-ssl	連線類型：顯式 SSL (AUTH TLS) 或隱式 SSL (AUTH SSL)。設為 true 使用明確的 SSL。
輸入檔案模式	input-file-pattern	檔案名稱模式，用於比對特定類型的檔案。 範例：指定 *.xls，僅選取 Excel 檔案。
輸入內容類型	input-content-type	執行階段包含的媒體類型：content-type 訊息標頭中的值媒體類型至少包含 2 個部分：類型、子類型和 1 或多個選用參數。該 text 子類型有選填項目 charset 參數，指定字元編碼。 範例：text/html; charset=UTF-8 如果是非 XML 文字檔格式，且檔案中並未明確指出媒體類型和字元集，此屬性就特別有用。
輸出檔案模式	output-file-pattern	指定產生輸出檔案名稱的方式。模式可包含常值字串以及下列一或多個標記： • \${INFILE}，取代為輸入檔案的名稱。 • \${UUID}，取代為通用唯一 ID。

屬性	XML 屬性內容名稱	說明
		• <code>\${EXT}</code>，放置在模式的結尾，根據寫入組件元件輸出的資料內容類型，以不同的副檔名取代。
方法	<code>method</code>	指定轉移方法： • <code>put「」</code>：將檔案寫入包含遠端目錄的 FTP 端點。 • <code>list</code>：列出 FTP 端點上的所有檔案。將「輸入檔案模式」屬性搭配使用 <code>list</code>，將模式套用至要傳回的清單。您可以存取「」的結果 <code>list</code> 方法使用 <code>MVELftp</code> 物件。範例：<code>ftp.list()</code> 會傳回一個字串清單，其中包含最後擷取的清單。 • <code>get</code>：擷取第一個符合「輸入檔案模式」屬性值的檔案。 • <code>delete</code>：刪除找到的第一個符合「輸入檔案模式」屬性值的檔案。 註：刪除後，檔案不會自動從 FTP 網站刪除。<code>get</code> 已執行「」。如果您要刪除檔案，必須呼叫個別的方法。範例：您可以包含單獨的 <code>ftp-out</code> 專門傳送以刪除檔案。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	<code>store-message</code>	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	<code>transport-class</code>	視需要逐一指定替代實作類別。 您可透過編輯 <code>WEB-INF/classes/spring/assembly-bean.xml</code> 檔案，全域變更實作類別。
忽略動態端點	<code>ignore-dynamic-endpoints</code>	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。

屬性	XML 屬性內容名稱	說明
		請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
逾時	timeout	指定用戶端與傳輸系統正在偵聽的 FTP 伺服器之間的 FTP 階段作業逾時期間 (以毫秒為單位)。 如果您未指定值，則階段作業會在 300000 毫秒 (5 分鐘) 後逾時。
MIME 類型對應檔案	mime-types-map-file	指定對應檔案的位置，以便在 MIME 類型與檔案副檔名之間進行對應。使用此屬性可： - 當「方法」屬性設定為時，將 MIME 類型對應至檔案副檔名 put。 - 當「方法」屬性設定為時，將檔案副檔名對應至 MIME 類型 get。 此對應的檔案為 ccx/conf/mime.types。您可以提供您自己的此檔案版本。 此傳輸可使用 \${EXT} 放在結尾的標記 output-file-pattern 值。
臨時檔案名稱	temp-file-name	指定臨時檔案名稱。當「方法」屬性設為 put，檔案會使用此臨時檔案名上傳。FTP 完成後，檔案會重新命名為輸出檔案模式屬性指定的名稱。
關閉連結	close-connection	指定是否在 FTP 動作完成時進行中斷連接。
額度大小	pool-size	此元件在個別主機上可支援的同時連線和階段作業數上限。如果嘗試連線至同一主機的執行緒數超過指定數量，系統會強制部分執行緒等候存取池。 若要使分配池大小不受限制，請指定負數。
偵錯	debug	啟用傳輸中的 JSCAPE 偵錯模式。

參照：XMPP 傳出傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	程序集內 xmpp-out 元件的唯一 ID。
將回覆送交至	routes-response-to	ftp-out 傳輸的回應訊息目的地。
執行時間	execute-when	決定是否執行傳輸的條件。
端點	endpoint	指定即時訊息的收件者。範例： xmpp:logan.mcneil@gmail.com
Domain (網域)	domain	註冊使用者名稱的網域。範例： gmail.com
伺服器	server	XMPP/Jabber 伺服器的主機或 IP 位址。
連接埠	port	XMPP 傳輸裝置用來偵聽訊息的連接埠號。大多數 XMPP 伺服器使用的連接埠是 5222。
使用者名稱	username	用於存取 FTP 伺服器的使用者名稱。
密碼	password	與用來存取 FTP 伺服器的使用者名稱相關聯的密碼。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。

參照：自訂傳出傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	自訂移出 元件在組合體中的唯一 ID。
將回覆送交至	routes-response-to	自訂傳出 傳輸的回應訊息目的地。
執行時間	execute-when	決定是否執行傳輸的條件。
端點	endpoint	自訂傳出 傳輸系統會將訊息傳送到端點。由 自訂輸出的 Spring bean 實作使用。
輸入	input	指定集合元件從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定集合元件將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。

屬性	XML 屬性內容名稱	說明
Spring Bean	spring-bean	指定實作傳輸介麵類別的 Spring bean。
方法名稱	method-name	自訂 Java 類別中的 Java 方法名稱。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存 步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。

參照：電子郵件傳出傳輸屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	集合內 電子郵件寄出 元件的唯一 ID。
將回覆送交至	routes-response-to	電子郵件傳出 傳輸的回應訊息目的地。
執行時間	execute-when	決定是否執行傳輸的條件。
收件者	to	目標收件者的主要電子郵件地址。遵循格式 mailto:[emailaddress]。 如果您只想使用「密件抄送」地址，請在欄位中指定一個空白。mailto: 值。
主旨	subject	電子郵件的主旨行。
開始	from	應顯示在已發送電子郵件的「寄件者」欄位中的電子郵件地址。
SSL	ssl	指定連線是否為 SSL。

屬性	XML 屬性內容名稱	說明
開始	starttls	指定是否要在開始 SMTP 階段作業之前，發出 STARTTLS 命令，立即啟用與遠端 SMTP 伺服器的 SSL 協商。SMTP 伺服器以下列其中一個回覆代碼做為回應： • 220 準備開始。 • TLS 501 語法錯誤 (不允許參數)。 • 454 TLS 由於暫時性原因而無法使用。 註: Ssl 和 starttls 屬性互斥。
主機	host	傳輸系統用來傳送訊息的 SMTP 郵件伺服器的名稱或 IP 位址。 如果您使用的是 OAuth，請將此值設為 localhost。
連接埠	port	執行郵件伺服器的連接埠號。預設值為 25，即標準 SMTP 連接埠。例外： • 如果 SSL 屬性設為 true 且未指定連接埠，則預設為 465。 • 如果 Starttls 屬性設為 true 且未指定連接埠，則預設為 587。 註:
使用者		用於存取電子郵件伺服器的使用者名稱。
密碼	password	在「使用者」屬性中指定的使用者名稱的密碼。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。

屬性	XML 屬性內容名稱	說明
		請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
密件抄送	bcc	指定密件副本的電子郵件收件者 (以分號分隔)。範例： logan.mcneil@workday.com; jake.lee@workday.com。
Cc	cc	使用分號分隔符號指定複寫電子郵件收件者。範例： logan.mcneil@workday.com; jake.lee@workday.com。
回覆	reply-to	使用者回覆電子郵件時，系統會自動插入「收件人」欄位中的電子郵件地址。對於大多數電子郵件，「寄件者」欄位中的地址就足夠了，因此不需要定義「回覆」屬性。指定多個電子郵件地址時請使用分號分隔符號。範例： logan.mcneil@workday.com; jake.lee@workday.com。
逾時	timeout	指定用戶端與傳輸正在偵聽的 FTP 伺服器之間的 FTP 階段作業逾時期間 (以毫秒為單位)。如果您未指定值，則階段作業會在 300000 毫秒 (5 分鐘) 後逾時。
暫時性例外	transient-exceptions	暫時性例外是在重試時不做任何修改即可成功的例外。「非暫時性例外」是指在問題的根本原因得到更正之前，重試時仍會失敗的例外情況。 根據預設，Workday 只會處理 <code>java.net.ConnectException</code> 和 <code>java.net.SocketException</code> 錯誤列為暫時性例外。 請將其他暫時性例外指定為逗號、空格或逗號與空格分隔的類別名稱列舉。範例： <code>java.net.UnknownHostException</code> <code>java.net.ConnectException</code> <code>java.net.SocketException</code> <code>java.net.UnknownHostException,j</code> <code>java.net.UnknownHostException, java.net.ConnectException, java.net.SocketException</code> <code>@{new</code> <code>java.net.UnknownHostException(</code>

屬性	XML 屬性內容名稱	說明
		註：如果您指定其他暫時性例外，則還必須包含預設例外 <code>java.net.ConnectException</code> 和 <code>java.net.SocketException</code> 。否則 Workday 會覆寫這些內容。

SMTP 標頭分頁

屬性	XML 屬性內容名稱	說明
名稱	name	自訂 SMTP 標頭的名稱。 若要在任何現有的列下方新增自訂標頭，請按一下 新增，然後提供 名稱 和 值。 若要在現有列上方新增標頭，請將游標移到該列，然後在 新增 下拉式功能表上按一下 插入。 使用 SMTP 標頭 分頁右側的按鈕可刪除列或重新排序。
值	value	自訂 SMTP 標頭的值。

OAuth 分頁

若要將 Microsoft Outlook OAuth 驗證搭配 Studio 的電子郵件寄出元件使用，請在 Microsoft Azure 中將 Studio 註冊為應用程式，提供全網域委派，並授予管理者同意以下權限：Mail.Send 權限，並取得各種值。詳情請參閱 <https://learn.microsoft.com/en-gb/azure/active-directory/develop/v2-oauth2-client-creds-grant-flow>。

若要將 Google Gmail OAuth 驗證與 Studio 的電子郵件寄出元件搭配使用，請在 Google Cloud 註冊服務帳戶、提供全網域的委派，並取得各種值的 JSON 說明。詳情請參閱 <https://support.google.com/cloud/answer/6158849?hl=en>。

屬性	XML 屬性內容名稱	說明
提供者	provider	電子郵件服務業者的名稱，兩者擇一 MICROSOFT_OUTLOOK 或 GOOGLE_GMAIL。
名稱	name	您的 Workday 強制定義名稱，與從 Microsoft Azure 入口網站或 Google 雲端主控台取得的各種值相關聯。 若為 Microsoft Outlook，請輸入以下名稱： • outlook.client.id • outlook.client.secret • outlook.tenant.id • outlook.user.id 對於 GoogleGmail，輸入：

屬性	XML 屬性內容名稱	說明														
		- gmail.service.account.key.json - gmail.delegated.user														
值	value	從 Microsoft 或 Google 取得且與上述名稱相關聯的值。 若為 Microsoft Outlook，請建立以下名稱/值組合： <table><tr><th>名稱</th><th>值</th></tr><tr><td>outlook.client.id</td><td>從 Microsoft Azure 入口網站上已註冊應用程式的「總覽」頁面：應用程式 (用戶端) ID</td></tr><tr><td>outlook.client.secret</td><td>從 Microsoft Azure 入口網站上已註冊應用程式的「憑證和密鑰」頁面：用戶端密鑰 ID</td></tr><tr><td>outlook.tenant.id</td><td>從 Microsoft Azure 入口網站上已註冊應用程式的「總覽」頁面：目錄 (用戶) ID</td></tr><tr><td>outlook.user.id</td><td>傳送電子郵件的帳戶 Outlook 位址。也在「通用」分頁的「來源」欄位中輸入。</td></tr></table> 對於 GoogleGmail，請建立以下名稱/值組合： <table><tr><th>名稱</th><th>值</th></tr><tr><td>gmail.service.account.key.json</td><td>從 Google 雲端主控台「服務帳戶」頁面的</td></tr></table>	名稱	值	outlook.client.id	從 Microsoft Azure 入口網站上已註冊應用程式的「總覽」頁面：應用程式 (用戶端) ID	outlook.client.secret	從 Microsoft Azure 入口網站上已註冊應用程式的「憑證和密鑰」頁面：用戶端密鑰 ID	outlook.tenant.id	從 Microsoft Azure 入口網站上已註冊應用程式的「總覽」頁面：目錄 (用戶) ID	outlook.user.id	傳送電子郵件的帳戶 Outlook 位址。也在「通用」分頁的「來源」欄位中輸入。	名稱	值	gmail.service.account.key.json	從 Google 雲端主控台「服務帳戶」頁面的
名稱	值															
outlook.client.id	從 Microsoft Azure 入口網站上已註冊應用程式的「總覽」頁面：應用程式 (用戶端) ID															
outlook.client.secret	從 Microsoft Azure 入口網站上已註冊應用程式的「憑證和密鑰」頁面：用戶端密鑰 ID															
outlook.tenant.id	從 Microsoft Azure 入口網站上已註冊應用程式的「總覽」頁面：目錄 (用戶) ID															
outlook.user.id	傳送電子郵件的帳戶 Outlook 位址。也在「通用」分頁的「來源」欄位中輸入。															
名稱	值															
gmail.service.account.key.json	從 Google 雲端主控台「服務帳戶」頁面的															

屬性	XML 屬性內容名稱	說明	
		名稱	值
			「KEYS」區段下載的檔案中的 JSON 文字。
		gmail.delegated	用來傳送電子郵件的 Gmail 地址。必須是與 Google 服務帳戶相關聯的委託人。也在「通用」分頁的「來源」欄位中輸入。

透過 **SSL** 或使用 **STARTTLS** 的 Gmail

配置 電子郵件傳出 傳輸方式以使用透過 SSL 或 STARTTLS 的 Gmail 時，請確保：

- 使用者名稱 和 密碼 包含有效的 Gmail 憑證。
- 端點 為 `mailto:[name]@gmail.com`。
- 主機 為 `smtp.gmail.com`。
- `ssl` 或 `starttls` 設為 `true`。
- 連接埠 為：
 - 465 (如果 `ssl` 設定為 `true`。
 - 587 (如果 `starttls` 設為 `true`。
- 逾時時間 為 30000 (30 秒)。

Microsoft Office 365 over SSL 或 STARTTLS

配置 電子郵件傳出 傳輸以使用 Microsoft Office 365 over SSL 或 STARTTLS 時，請確保：

- `user` 和 `password` 包含有效的 Office 365 使用者憑證。
- 端點 為 `mailto:[name]@[domain-name.top-level-domain]`。
- 主機 為 `smtp.office365.com`。
- `ssl` 或 `starttls` 設為 `true`。
- 連接 以下任一項：
 - 465 (如果 `ssl` 設定為 `true`。
 - 587 (如果 `starttls` 設為 `True`)
- 逾時時間 為 30000 (30 秒)。

配置非二進位格式的訊息內文和附件

許多電子郵件應用程式在接收傳入訊息時，會比對 MIME 訊息中的第一個文字部分，並將其當做訊息文字使用。將根訊息部分的 MIME 類型設為 `text/plain`，將訊息的 輸入 和 輸出 屬性設為 訊息，然後 將輸出 MIME 類型設為 文字/純文字。

使用 訊息建置器，透過 文字 元素定義電子郵件內文的訊息內容。

將附件的「輸出」屬性設為「」，將「輸出」附件索引設為 0，輸出 MIME 類型為「純文字/純文字」。

同樣，請使用 訊息建置器，透過 文字 元素定義附件的訊息內容。

若要避免某些電子郵件應用程式在訊息內文中顯示文字附件，請將附件 MIME 類型設為 *application/octet-stream*，即使附件實際上是文字也是如此。為此，請在 評估 步驟內使用 MVEL 運算式。範例：
`parts[1].setMimeType('application/octet-stream')`

若要設定附件檔案名稱，請使用 MVEL 運算式來設定其附件檔案名稱。Content-Disposition 頁首。範例：`parts[1].setHeader('Content-Disposition', 'attachment; filename=payload.txt')`。

最後，配置 電子郵件寄出 步驟。

配置二進位格式的訊息內文和附件

若要将二進位檔案插入 MediationContext，請為從 FTP 或 SFTP 傳輸擷取檔案的 Workday-in 定義擷取服務。

建立訊息內文，然後使用 評估 步驟將擷取的檔案複製到變數。您可以使用 `da` 文件存取器變數和 `toVar` 方法來達成此目的。範例：`da.toVar(Rising_2011_Logo.jpg', 'wd.retrieve.variable')`。

使用 複製 步驟，將檔案從 `wd.retrieve.variable` 至附件。為此，請將 輸入 屬性設為「變數」並指定 `wd.retrieve.variable` 做為輸入變數。將 輸出 屬性設為 附件「」，指定的 輸出 附件索引為 0 和「*image/jpeg*」的 輸出 MIME 類型。

最後，配置 電子郵件寄出 步驟。

HTTP 傳出傳輸子元素

概念：HTTP-Out 傳輸子元素

http-outTransport 支援下列子系元素：

子元素	說明
http-basic-auth	支援 http-out 傳輸的基本驗證。
http-custom-auth	支援對您的 http-out 傳輸進行自訂驗證。
https-properties	在您的 http-out 傳輸上支援 HTTPS 屬性。

參照：HTTP-基本驗證屬性

通用分頁

屬性	XML 屬性內容名稱	說明
密碼	password	用於驗證的密碼。
使用者名稱	username	用於驗證的使用者名稱。
領域	realm	回叫處理常式 Bean 使用的領域名稱。
回調處理程式 Bean	callback-handler-bean	實現 <code>javax.security.auth.callback.Callback</code> 並處理回呼類型 <code>com.capeclear.transport.impl.http.a</code> 提供使用者名稱和密碼詳細資料以供驗證。

屬性	XML 屬性內容名稱	說明
		如果您沒有為 http-basic-auth 子元素指定密碼詳細資料，則需要改用回呼處理常式 bean。範例：<http-basic-auth username="me" callback-handler-bean="myCallBackHandlerBean"/>。 如果您沒有為 http-basic-auth 子元素指定使用者名稱和密碼詳細資料，則需要改用回呼處理常式 bean。範例：<http-basic-auth callback-handler-bean="myCallBackHandlerBean"/>。 如果您為 http-basic-auth 子元素指定使用者名稱和密碼詳細資料，則不需要回呼處理常式 Bean。範例：<http-basic-auth username="me" password="mypassword"/>。

參照：HTTP 自訂驗證屬性

通用分頁

屬性	XML 屬性內容名稱	說明
密碼	password	用於驗證的密碼。
使用者名稱	username	用於驗證的使用者名稱。
領域	realm	回叫處理常式 Bean 使用的領域名稱。
回調處理程式 Bean	callback-handler-bean	實現 javax.security.auth.callback.Callback 並處理回呼類型 com.capeclear.transport.impl.http.a 提供使用者名稱和密碼詳細資料以供驗證。 如果您沒有為 http-custom-auth 子元素指定密碼詳細資料，則需要改用回呼處理常式 Bean。範例：<http-custom-auth username="me" callback-handler-bean="myCallBackHandlerBean"/>。 如果您沒有為 http-custom-auth 子元素指定使用者名稱和密碼詳細資料，則需要

屬性	XML 屬性內容名稱	說明
		改用回呼處理常式 bean。 範例：<http-custom-auth callback-handler-bean="myCallbackHandlerBean"/>。 如果您為 http-custom-auth 子元素指定使用者名稱和密碼詳細資料，則不需要回呼處理常式 Bean。範例：<http-custom-auth username="me" password="mypassword"/>。
身分驗證器 Bean	authenticator-bean	執行自訂身分驗證的 Spring Bean。

參照：HTTPS 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
接受主機	accept-hosts	Java 規則運算式字串，用於驗證 SSL 憑證中指定的主機名稱。
金鑰密碼	key-password	存取金鑰庫內適當金鑰時使用的密碼。
金鑰庫檔案	key-file	連線至 HTTPS 端點時使用的金鑰庫檔案位置。
金鑰庫密碼	keystore-password	存取金鑰庫時要使用的密碼。
Proxy 主機	proxy-host	代理伺服器的主機名稱，您可用來代理此傳輸的 HTTPS 要求。
Proxy 連接埠	proxy-port	代理伺服器的連接埠，用於代理此傳輸的 HTTPS 要求。
信任庫檔案	truststore-file	連線至 HTTPS 端點時使用的信任庫檔案位置。
信任庫密碼	truststore-password	存取信任庫時要使用的密碼。

FTPS-Out 傳輸子元素

概念：FTPS-Out 傳輸子元素

ftps-out 傳輸支援下列子系元素：

子元素	說明	附註
proxy-properties	Proxy 驗證屬性	
防火牆屬性	防火牆連線屬性	ftps-out 傳輸的預設適用於防火牆，因為連線是被動

子元素	說明	附註
		的。但是，某些防火牆可能會要求您將所有連線路由至這些防火牆。
client-key-properties	用戶端金鑰驗證屬性	
伺服器金鑰屬性	伺服器金鑰驗證屬性	

參照：Proxy 屬性屬性

通用分頁

屬性	XML 屬性內容名稱	說明
Proxy 主機	proxy-host	連至安全 FTP 網站時使用的 Proxy 伺服器主機名稱。
Proxy 密碼	proxy-password	代理伺服器連線至安全 FTP 網站時所用的密碼。
Proxy 連接埠	proxy-port	連至安全 FTP 網站時使用的 Proxy 伺服器連接埠號碼。
Proxy 類型	proxy-type	連至安全 FTP 網站時使用的 Proxy 伺服器 Proxy 類型。預設類型為 http。
Proxy 使用者名稱	proxy-username	連至安全 FTP 網站時使用的 Proxy 伺服器使用者名稱。

參照：防火牆-屬性內容

通用分頁

屬性	XML 屬性內容名稱	說明
防火牆主機	firewall-host	建立安全 FTP 網站連線時使用的防火牆主機名稱或 IP 位址。
防火牆連接埠	firewall-port	防火牆的連接埠號碼。

參照：用戶端金鑰屬性

通用分頁

屬性	XML 屬性內容名稱	說明
密碼	passphrase	使用安全 FTP 網站驗證時使用的私人金鑰密碼。
私人金鑰	private-key	與用於 SFTP 驗證的使用者名稱相關聯的私人金鑰。您可以透過 2 種方式指定： 使用附件通訊協定和 Workday ID 的 URI。例如，

屬性	XML 屬性內容名稱	說明
		attachment:privatekey/{WID}，其中 WID 是 32 個字元的 Workday ID。 相對路徑位置範例：private_key.txt，其中私人密鑰.txt 是位於 ws/WSAR-INF 資料夾中的檔案。

參照：伺服器金鑰屬性

通用分頁

屬性	XML 屬性內容名稱	說明
密碼	passphrase	使用安全 FTP 網站驗證時使用的私人金鑰密碼。
私人金鑰	private-key	與用於 SFTP 驗證的使用者名稱相關聯的私人金鑰。您可以透過 2 種方式指定： - 使用附件通訊協定和 Workday ID 的 URI。例如，attachment:privatekey/{WID}，其中 WID 是 32 個字元的 Workday ID。 - 相對路徑位置範例：private_key.txt，其中私人密鑰.txt 是位於 ws/WSAR-INF 資料夾中的檔案。

SFTP-Out 子系元素

概念：SFTP-Out 子系元素

sftp-out 傳輸支援下列子系元素：

子元素	說明	附註
proxy-properties	Proxy 驗證屬性	
private-key-properties	私人金鑰驗證屬性	

參照：Proxy 屬性屬性

通用分頁

屬性	XML 屬性內容名稱	說明
Proxy 主機	proxy-host	連至安全 FTP 網站時使用的 Proxy 伺服器主機名稱。

屬性	XML 屬性內容名稱	說明
Proxy 密碼	proxy-password	代理伺服器連線至安全 FTP 網站時所用的密碼。
Proxy 連接埠	proxy-port	連至安全 FTP 網站時使用的 Proxy 伺服器連接埠號碼。
Proxy 類型	proxy-type	連至安全 FTP 網站時使用的 Proxy 伺服器 Proxy 類型。預設類型為 http。
Proxy 使用者名稱	proxy-username	連至安全 FTP 網站時使用的 Proxy 伺服器使用者名稱。

參照：私人金鑰屬性

通用分頁

屬性	XML 屬性內容名稱	說明
密碼	passphrase	使用安全 FTP 網站驗證時使用的私人金鑰密碼。
私人金鑰	private-key	與用於 SFTP 驗證的使用者名稱相關聯的私人金鑰。您可以將其指定為相對路徑位置。範例：private_key.txt，其中私人金鑰檔案位於 ws/WSAR-INF 資料夾。

元件

概念：元件

元件可讓您在訊息經過您的整合時，操縱和轉換訊息。Workday Studio 提供對下列部件的支援：組套件 調色板的 元件 區段：

中介組成	說明	附註
異步中介	支援 1 個處理鏈，您可以填入在程序集中定義要求訊息的步驟。	
同步中介	支援 2 個處理鏈，您可以填入步驟，在程序集中定義要求訊息和要求-回應訊息。	
custom-mediation	建立同時支援要求訊息和要求/回應訊息的可自訂元件。	
路線	實行路由策略和子路由，以引導訊息在您的程序集中流動。	
拆分器	實作拆分策略和子路由，以引導訊息在您的程序集中流動。	向下游 匯總器 元件發送信號。
aggregator	將訊息串連成批次，您可將這些批次導向已配置的目的地。	偵測上游 拆分 器元件。

參照：非同步中介元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	程序集內 異步中介 元件的唯一 ID。
路由至	routes-to	異步中介 會將訊息導向哪個程序集元素。
執行時間	execute-when	一個條件，決定是否執行 異步中介 元件。

進階分頁

屬性	XML 屬性內容名稱	說明
發生錯誤後繼續	continue-after-error	指出當錯誤處理程式清除錯誤時，Workday 的行為方式。 預設值為「倒帶」，表示從組合體中的上一個元件繼續處理。 復原 設定可指定當發生錯誤時，Workday 會跳到中介的下一個步驟。
處理下游錯誤	handle-downstream-errors	是否要處理下游錯誤。 如果設為 False，則本機錯誤處理常式僅處理髮生在 async-mediation 元件內的錯誤。全域錯誤處理程式會改為處理下游錯誤。 如果設為 True，則 async-mediation 元件的本機錯誤處理常式可處理下游錯誤。

參照：同步中介元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	程序集中 同步中介 元件的唯一 ID。
路由至	routes-to	同步中介將 訊息導向的程序集元素。
將回覆送交至	routes-response-to	用於 同步中介 元件的回應訊息目的地。
執行時間	execute-when	決定是否執行 同步中介 元件的條件。

進階分頁

屬性	XML 屬性內容名稱	說明
發生錯誤後繼續	continue-after-error	指出當錯誤處理程式清除錯誤時，Workday 的行為方式。 預設值為「倒帶」，表示從組合體中的上一個元件繼續處理。 復原 設定可指定當發生錯誤時，Workday 會跳到中介的下一個步驟。
處理下游錯誤	handle-downstream-errors	是否要處理下游錯誤。 如果設為 False，則本機錯誤處理常式只會處理在 同步中介 元件內發生的錯誤。全域錯誤處理程式會改為處理下游錯誤。 如果設為 True，則 同步中介 元件的本機錯誤處理常式可處理下游錯誤。

參照：自訂中介元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID		自訂中介 元件在程序集中的唯一 ID。
路由至	routes-to	自訂中介 將訊息導向的程序集元素。
將回覆送交至	routes-response-to	自訂中介 元件的回應訊息目的地。
執行時間	execute-when	決定是否執行 自訂中介 元件的條件。
Spring Bean	ref	由 custom-mediation實作的自訂 Spring bean Java 類別。
要求方式	request-method	要求處理方法的名稱。
回覆方法	response-method	用於處理回應的方法名稱。

參照：路由元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	路線 元件在裝配體中的唯一 ID。

參照：拆分器元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	拆分器 元件在組合體中的唯一 ID。
沒有拆分訊息錯誤	no-split-error-message	指定當 拆分器 元件未為整批輸入訊息產生拆分訊息時，Workday 是否會引發錯誤。
拆分截止日期	split-until	指定拆分器元件何時停止處理大量輸入訊息。

參照：匯總器元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	集合體中 匯總器 元件的唯一 ID。
路由至	routes-to	匯總器 將訊息導向的程序集元素。
核對時間	collate-when	一個 MVEL 運算式，用於決定是否整理傳入的訊息。
在最後一則訊息上強制執行批次	force-batch-on-last-message	在 拆分器 元件收到最後一條訊息後強制執行批次作業。
強制執行批次的時間	force-batch-when	一個 MVEL 運算式，用於決定是否將傳入訊息進行批次處理。

進階分頁

屬性	XML 屬性內容名稱	說明
允許外部控制	allow-external-control	讓上游元件能夠控制 匯總器 元件何時進行對照、何時執行批次，或兩者皆執行。

路由元件子系元素

概念：路由元件子系元素

路線 元件支援這些子系元素。

子元素	說明	附註
custom-strategy	可自訂的路由策略，您可將其配置為 Spring bean。	

子元素	說明	附註
regex-strategy	實作規則運算式，將訊息導向子路由。	實作 choose-route 子元素，以指出您的目標 子路由「」。
mvel-strategy	實作 MVEL 運算式，將訊息導向子路由。	實作 choose-route 子元素，以指出您的目標 子路由「」。
round-robin-strategy	將訊息按循環順序導向子路由。	
xPath-strategy	實作 XPath 運算式，將訊息導向子路由。	實作 choose-route 子元素，以指出您的目標 子路由「」。
failover-strategy	實作故障轉移機制，將訊息導向子路由。如果 子路由 子系元素發生故障，則會以下一個 子路由取代該元素。	
all-strategy	將訊息依序導向每個子路由。	
文件迭代器	重複執执行程序集從 Workday 擷取的文件。	
循環策略	實作循環機制，將訊息導向子路由。	
choose-route	指出您的策略要將訊息導向哪個目標 子路由 子系元素。	支援 XPath、MVEL 或規則運算式。
子路由	接收路由策略傳來的訊息，並將訊息導向集合中的其他元件和傳輸。	

參照：自訂策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
Spring Bean	ref	實作您的 自訂策略 介麵類別的 Spring bean。
可重複	repeatable	啟用循環。 如果設為 True，則程序集會實作一個循環，重複呼叫您的 custom-strategy。 如果設為 False，則程序集不會為您的 custom-strategy實作循環。
重複限制	repeat-limit	限制循環呼叫您的 custom-strategy次數。
處理錯誤	handle-error	指出您的父系 路由 元件是否會處理 子路由的例外。

參照：Regex-策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
篩選	filter	表示當您在 choose-route 子元素上實作無效的規則運算式時，Studio 是否會顯示錯誤訊息。

參照：最佳實際教學應用程式 - 策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
篩選	filter	指出當您在 choose-route 子元素上實作無效的 MVEL 運算式時，Studio 是否會顯示錯誤訊息。

參照：循環策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
複製要求	clone-request	表示 循環策略 是否會將原始中介訊息的副本傳送至您的每個子路由 子系元素。
故障切換	failover	表示當「子路由」失敗時，循環策略 是否會循環執行您的 子路由 元素。
逾時	timeout	毫秒值，指出失敗的子路由會從路由排除多久。

參照：XPath-策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
篩選	filter	表示當您在 choose-route 子元素上實作無效的XPath 運算式時，Studio 是否會顯示錯誤訊息。
名稱空間	namespaces	XPath 運算式使用的名稱空間，格式為 prefix namespace prefix2 namespace2。覆寫 component.xml 檔案中的名稱空間對應。
XPath 版本	xpath-version	指定是否使用 XPath 版本 1.0、2.0 或 3.0。對於從 Workday 11 開始的組套件

屬性	XML 屬性內容名稱	說明
		版本，請輸入值 1。請針對 Workday 12 起的組套件版本輸入 2 或 3。 您可以讓 Workday 使用可插入系統屬性值的 MVEL 範本動態設定版本。範例： <code>@{props.get('the_version')}</code> 。

參照：故障轉移策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
清除錯誤	<code>clear-fault</code>	表示 故障轉移策略 是否會清除子路由子系元素產生的例外。
複製要求	<code>clone-request</code>	表示 failover-strategy 是否會將原始中介訊息的副本傳送至您的每個子路由子系元素。

參照：所有策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
複製要求	<code>clone-request</code>	表示 整體策略 是否會將原始中介訊息的副本傳送至您的每個子路由子系元素。
忽略錯誤	<code>ignore-errors</code>	指出 整體策略 是否忽略子路由子系元素產生的錯誤。

參照：文件迭代器屬性

通用分頁

屬性	XML 屬性內容名稱	說明
變數名稱	<code>variable-name</code>	變數的名稱，文件迭代器會在其中儲存您擷取的文件。預設值為 <code>wd.retrieve.variable</code> 。
標籤	<code>labels</code>	指定附加到您擷取的文件的一或多個標籤。
排序依據	<code>sort-by</code>	擷取和呈現文件的排序順序。選項包括： • <code>FILENAME_ASCENDING</code> • <code>FILENAME_DESCENDING</code> • <code>NONE</code>

參照：循環策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
條件	condition	產生布林值的 MVEL 運算式。循環策略 會執行，直到運算式計算結果為 False。
增量	increment	每個循環小數次的增量值。
起始	init	循環的起始值。
重複限制	repeat-limit	循環策略實作的循環次數上限。 路線策略最多可重複重複 200,000 次。如果多次呼叫相同的策略，則不會重設循環計數器。從上一個值繼續。您可以使用自訂 Java Bean 或自訂拆分器來避免此限制。

參照：選擇路由屬性

通用分頁

屬性	XML 屬性內容名稱	說明
運算式	expression	用來比對傳入訊息值的運算式。
路由	route	比對運算式值的目標 子路由 子系元素。

參照：子路由屬性

通用分頁

屬性	XML 屬性內容名稱	說明
名稱	name	子路由 在裝配體中的唯一名稱。
路由至	routes-to	子路由 將訊息導向的程序集元素。

拆分器元件子系元素

概念：拆分器元件子系元素

拆分器 元件支援這些子系元素。

子元素	說明	附註
custom-splitter	可自訂的拆分器策略，您可將其配置為 Spring bean。	

子元素	說明	附註
standard-splitter	從大量訊息中擷取資料、將資料拆分為多個權杖，並將權杖化資料導向您的目標子路由。	支援以下子系元素： - 標頭結尾與 - 標頭固定明細 - content-fixed-lines - content-starts-ends - content-starts-with - 頁尾固定行 - 頁尾開頭為
xPath-拆分器	實作 XPath 運算式以從傳入的大量訊息中擷取資料。	
xml-流-拆分器	實作 XPath 運算式以從傳入的大量訊息中擷取資料。支援 XML 串流。	
json-splitter	從大量輸入訊息中擷取 JSON。	
unzip-splitter	從 ZIP 和 TAR 檔案中擷取資料、將資料拆分為個別訊息，並將訊息導向其目標子路由。	
mtable-splitter	從「」中擷取資料 mtable 物件，並將資料拆分成個別的列。	
子路由	接收來自拆分策略的訊息，並將這些訊息導向集合中的其他元件和傳輸。	

參照：自訂拆分屬性

通用分頁

屬性	XML 屬性內容名稱	說明
Spring Bean	ref	實作您的 自訂拆分器 介麵類別的 Spring bean。

參照：標準拆分屬性

通用分頁

屬性	XML 屬性內容名稱	說明
輸出 MIME 類型	output-mimetype	符號化輸出資料的 MIME 類型。
明細代碼分隔符號 Regexp	line-token-delimiter-regexp	從大量訊息中擷取權杖化資料的分隔規則運算式。
多行分隔符號	multiple-lines-separator	串連跨越多行的標記的字串。

參照：XPath 拆分器屬性

通用分頁

屬性	XML 屬性內容名稱	說明
名稱空間	namespaces	XPath 運算式使用的名稱空間。
XPath	xpath	決定輸入訊息在何處拆分的 XPath 運算式。
XPath 版本	xpath-version	指定是否使用 XPath 版本 1.0、2.0 或 3.0。對於從 Workday 11 開始的組套件版本，請輸入值 1。請針對 Workday 12 起的組套件版本輸入 2 或 3。 您可以讓 Workday 使用可插入屬性值的 MVEL 範本動態設定版本。範例： @{props.get('the_version')}}。

參照：XML 串流拆分器屬性

通用分頁

屬性	XML 屬性內容名稱	說明
名稱空間	namespaces	XPath 運算式使用的名稱空間。
XPath	xpath	從大量輸入訊息中擷取 XML 的 XPath 運算式。
XPath 版本	xpath-version	指定是否使用 XPath 版本 1.0、2.0 或 3.0。對於從 Workday 11 開始的組套件版本，請輸入值 1。請針對 Workday 12 起的組套件版本輸入 2 或 3。 您可以讓 Workday 使用可插入系統屬性值的 MVEL 範本動態設定版本。範例： @{props.get('the_version')}}。

參照：JSON 拆分器屬性

通用分頁

屬性	XML 屬性內容名稱	說明
Json 路徑	json-path	從大量輸入訊息中擷取 JSON 的運算式。

參照：解壓縮拆分器屬性

通用分頁

屬性	XML 屬性內容名稱	說明
檔案模式	file-pattern	從 ZIP 和 TAR 輸入檔案中擷取資料的規則運算式。
格式	format	輸入檔案的格式。選項包括： • zip • tar

參照：Mtable 拆分器屬性

通用分頁

屬性	XML 屬性內容名稱	說明
清單名稱	mtable-name	的名稱 MediationContext 屬性內容包含 mtable。
列名稱	row-name	的名稱 MediationContext 儲存迭代列的屬性
列格式	row-format	決定重複列的類別。支援的值包括： • array • map

標準拆分子元素

概念：標準拆分子元素

標準拆分器 元件支援這些子系元素。

子元素	說明
標頭結尾與	表示文件中標頭區段的結束。
標頭固定明細	指出文件中標頭區段所佔用的行數。
content-fixed-lines	指出文件中資料記錄內容區段所佔用的行數。
content-starts-ends	指出文件中資料記錄內容區段的起點和終點。
content-starts-with	指出文件中資料記錄內容區段的起點。
頁尾固定行	指出文件中頁尾區段所佔用的行數。
頁尾開頭為	指出文件中頁尾區段的起點。

參照：標頭結尾包含屬性

通用分頁

屬性	XML 屬性內容名稱	說明
包含在標頭中	include-in-header	決定規則運算式的相符行是否為標頭的一部分。
Regex	regex	識別標頭結尾的規則運算式。

參照：標頭固定明細屬性

通用分頁

屬性	XML 屬性內容名稱	說明
明細數	lines-count	標頭所佔的明細數。

參照：內容固定線屬性

通用分頁

屬性	XML 屬性內容名稱	說明
忽略明細 Regex	ignore-lines-regex	規則運算式，用於從資料記錄內容區段中排除相符的行。
明細數	lines-count	資料記錄內容區段所佔用的行數。

參照：內容-開始-結束屬性

通用分頁

屬性	XML 屬性內容名稱	說明
結束 Regex	end-regex	識別資料記錄內容區段結尾的規則運算式。
開始 Regex	start-regex	識別資料記錄內容區段開始的規則運算式。

參照：內容開始於屬性

通用分頁

屬性	XML 屬性內容名稱	說明
Regex	regex	識別資料記錄內容區段開始的規則運算式。

參照：頁尾固定線屬性

通用分頁

屬性	XML 屬性內容名稱	說明
忽略明細 Regexp	ignore-lines-regexp	規則運算式，用於從頁尾區段排除相符的行。
明細數	lines-count	頁尾區段所佔的行數。
Trim Eof Ignore Lines	trim-eof-ignore-lines	決定規則運算式的相符行是否會從頁尾剪掉。

參照：頁尾開始位置屬性

通用分頁

屬性	XML 屬性內容名稱	說明
Regexp	regexp	識別資料記錄內容區段開始的規則運算式。

匯總器組成要素子系元素

概念：匯總器組成要素子系元素

匯總器 元件支援這些子系元素。

子元素	說明
custom-batch-strategy	可自訂的批次處理策略，您可將其配置為 Spring bean。
size-batch-strategy	依據指定的批次大小值，將傳入的訊息批次處理。
time-batch-strategy	依據指定時段值批次處理傳入訊息。
custom-collater	可自訂的整理策略，您可將其配置為 Spring bean。
message-content-collater	匯總的訊息內容 MediationMessage。
xml-message-content-collater	匯總 XML 訊息串流的 XML 文件內容。
json-collater	將 JSON 文件匯總為單一輸出。
zip-file-collater	將多個檔案匯總為單一 ZIP 檔案。
mtable-collater	匯總 mtable 列變成新的 mtable 實例。

參照：自訂批次策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
Spring Bean	ref	實作您的 custom-batch-strategy 介麵類別的 Spring bean。

參照：大小-批次-策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
批次大小	batch-size	批次串連的訊息數上限。

參照：時間批次策略屬性

通用分頁

屬性	XML 屬性內容名稱	說明
時段	time-period	批次匯總的持續時間。
時間單位	time-unit	「時段」屬性所實行的時間單位。預設值為 秒。

參照：自訂整理來源屬性

通用分頁

屬性	XML 屬性內容名稱	說明
輸出	output	指定 custom- collater 子元素將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。

屬性	XML 屬性內容名稱	說明
		變數：將輸出另存為具名變數 <code>MediationContext</code>。
Spring Bean	<code>ref</code>	實作您的 <code>custom-collater</code> 介麵類別的 <code>Spring bean</code> 。

參照：訊息內容整理程序屬性

通用分頁

屬性	XML 屬性內容名稱	說明
輸出	<code>output</code>	指定 <code>message-content-collater</code> 子元素將輸出導向何處。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。此值會建立新訊息，有效地移除所有附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息的 MIME 附件。請指定附件索引。 <code>rootpart</code>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。
標頭文字	<code>header-text</code>	在輸出訊息中新增標頭。
頁尾文字		在輸出訊息中新增頁尾。
分隔符號	<code>separator</code>	串連批次中的訊息時使用的字串。

參照：XML-Message-Content-Collater 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
輸出	<code>output</code>	指定 <code>xml-message-content-collater</code> 子元素會將輸出導向何處。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。此值會建立新訊息，有效地移除所有附件。

屬性	XML 屬性內容名稱	說明
		<code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息的 MIME 附件。請指定附件索引。 <code>rootpart</code>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。
標頭文字	<code>header-text</code>	在輸出訊息中新增標頭。
頁尾文字		在輸出訊息中新增頁尾。
分隔符號	<code>separator</code>	串連批次中的訊息時使用的字串。
名稱空間	<code>namespaces</code>	XPath 運算式使用的名稱空間。
XPath	<code>xpath</code>	一個 XPath 運算式，用於匯總相符的 XML 訊息。如果未指定，則 <code>xml-message-content-collater</code> 會將所有 XML 訊息內容匯總至 1 個輸出檔案。

參照：JSON 匯總屬性

通用分頁

屬性	XML 屬性內容名稱	說明
輸出	<code>output</code>	指定 <code>json-collater</code> 子元素將輸出導向何處。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。此值會建立新訊息，有效地移除所有附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息的 MIME 附件。請指定附件索引。 <code>rootpart</code>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。

參照：Zip-File-Collater 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
輸出	output	指定 zip-file- collater 子元素將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
格式	format	輸出檔案的格式。選項包括： - zip - tar
訊息實體名稱	message-entity-name	新增至輸出檔案的每條訊息的名稱。

參照：Mtable-Collater 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
輸出	output	Workday 會忽略 mtable- collater 子元素的此設定。
清單名稱	mtable-name	的名稱 <i>MediationContext</i> 儲存輸出的屬性 mtable。
列名稱	row-name	的名稱 <i>MediationContext</i> 儲存匯總資料的屬性 mtable 列。
欄名稱	column-names	定義輸出的欄清單 (以逗號分隔) mtable。
索引欄	index-column	輸出的索引欄名稱 mtable。
允許空列	allow-null-row	決定您的輸出是否 mtable 支援空列值。

通用元件

概念：通用元件

Workday Studio 提供一系列預先封裝的子組合，可用來建立整合。這些子裝配件可從 選用板的 通用元件 分頁取得。當您新增至組合圖表時，Studio 會插入 local-out 將 XML 來源中的元素與 Workday 中的適當端點搭配使用。

也會顯示您可搭配該子裝配件使用的參數，以及已選取的必要參數。您可以隨時在子裝配件體的 屬性 檢視中調整參數。

Studio 的 調色盤中包含以下 通用元件：

名稱	函數	附註
Ftp	路由至 ftp-out 傳輸。	- 為所有 FTP 傳輸提供單一入口點。 - 使用 wd.ftp.endpoint 參數值，以決定已實作的 FTP 傳輸類型。範例： ftp://, sftp://, 或 ftps://。 - 顯示 Workday 中常用的傳輸設定，例如使用者名稱和密碼驗證。隱藏更進階的公開金鑰驗證。 註: 已無法再從 Studio 選用板使用 FTP 子裝配件。在新的整合中，您應使用 ftp-out 或 sftp-out 元件，將文件傳送至 FTP 伺服器。這些元件可提供更高的安全性、更佳的效能和更強大的可靠性。也可與更高版本的 JSCAPE 相容。FTP 子組合件在現有整合中仍可運作，但 Workday 建議您在實際可行的情況下將其取代。
GetEventConfigurations	擷取附加至特定整合事件的服務配置。	
GetEventDocuments	擷取附加至特定整合事件的文件。	
GetIntegrationEvent	依據 Workday 參考 ID 擷取整合事件。	
GetIntegrationSystems	從 Workday 擷取整合系統的配置，並進行剖析。	
PagedGet	實作任何所需的分頁邏輯。paged-getWeb 服務作業，可逐頁擷取資料	- 使用 is.paged.get.process.endpoint 參數，以將回應資料的每一頁傳送至另一個子組合件進行處理。 - 使用 is.paged.get.aggregate.header, is.paged.get.aggregate.footer ,

名稱	函數	附註
		<code>is.paged.get.aggregate.xpath</code> 參數，以匯總回應資料的所有頁面。 - 匯總 XML 資料可做為「」中的訊息 <code>MediationContext</code> 當子裝配件退回時 - 如果 Web 服務預期會傳回大量頁面，請使用按頁面處理的方法。
PagedGetLocalPaging	實作任何所需的分頁邏輯。 <code>paged-get</code> 本機 Web 服務作業，可逐頁擷取資料。	- 允許在 ESB 伺服器本機執行分頁流程，而不是由 Workday 遠距執行。 - 使用單一查詢傳回所有 Workday ID。後續呼叫會擷取每個 Workday ID 的資料。
PdfPrintStep	將自訂報告資料從 <code>workday-out-rest</code> 使用 BIRT 報告設計將要求轉換為 PDF 格式。	請為每個支援 MVEL 的參數指定常值字串值，請將其放在單引號中。
PutIntegrationEvent	在 Workday 中更新整合系統的整合事件狀態。	- 使用 <code>Put_Integration_Event</code> Web 服務作業 - 將從 Web 服務作業傳回的事件回應 Workday ID 放入 <code>MediationContext</code> 屬性 <code>integrations.event.response.wi</code>
PutIntegrationMessage	將更新整合訊息傳送至 Workday。	- 也可以將參照形式的文件新增至 Workday 文件儲存區。 - 使用 <code>Put_Integration_Message</code> Web 服務作業 - Workday 將 <code>PutIntegrationMessage</code> 子組件在單一整合中的呼叫次數限制為 500 個。後續的整合訊息會記錄在記錄檔案中，無法在整合事件中檢視。(重新導向不適用於有附件的訊息或整合事件狀態變更。)
SalesforceConnector	將已驗證的訊息傳送至 <code>salesforce.com</code> 。	
PrismConnector	將 CSV 資料傳送至 Prism Analytics。	

參照：FTP 子組合屬性

通用分頁

註：已無法再從 Studio 選用板使用 FTP 子裝配件。在新的整合中，您應使用 ftp-out 或 sftp-out 元件，將文件傳送至 FTP 伺服器。這些元件可提供更高的安全性、更佳的效能和更強大的可靠性。也可與更高版本的 JSCAPE 相容。FTP 子組合件在現有整合中仍可運作，但 Workday 建議您在實際可行的情況下將其取代。

屬性	XML 屬性內容名稱	說明
ID	id	Ftp 子 組合件在組合體中的唯一 ID。
將回覆送交至	routes-response-to	Ftp 子 組合件的回應訊息目的地。
執行時間	execute-when	決定 Ftp 子 組合件是否執行的條件。
端點	endpoint	子組合件的目標端點地址。Ftp 元件支援 vm:// 端點地址範例： vm://wcc/Ftp。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存 步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組合件。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組合件傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。

屬性	XML 屬性內容名稱	說明
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組 合件後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
wd.ftp.endpoint	端點配置，用來決定要使用哪個 ftp-out 傳輸。端點必須以 ftp://, sftp://, 或 ftps://。
wd.ftp.username	用於存取 FTP 伺服器的使用者名稱。
wd.ftp.password	與用來存取 FTP 伺服器的使用者名稱相關聯的密碼。
wd.ftp.method	FTP 子組件在連至端點時實行的方法或訊息動作。請在每個動作加上單引號。有效的選項包括： 'put'：FTP 元件會傳送訊息至 FTP 伺服器，且訊息保持不變。 'get'：程序集會以從 FTP 伺服器接收的資料，取代訊息的根節點部分。 'list'：訊息保持不變。Workday 新增名為「」的環境定義屬性。wd.ftp.files.list，其中包含 List<String> 物件實例 註：使用 'list' 方法，檔案模式為 *.txt 或 *.xml 產生例外。 'delete'：Workday 要求從 FTP 伺服器刪除檔案。訊息將保持不變。 'mget' (Multiget)：Workday 會下載所有符合輸入模式的檔案，並將訊息的根目錄部分取代為包含這些檔案的 ZIP 檔案。 'mdelete' (多次刪除)：Workday 會刪除所有符合輸入模式的檔案。訊息將保持不變。
wd.ftp.passive.mode	指定在 FTP 連線期間是否使用被動模式。
wd.ftp.file.pattern	指定執行 FTP 作業時使用的檔案模式。範例： *.xml *.txt 適用於大多數方法以外的 list、Workday 支援萬用字元，並在用戶端執行篩選。針對 list 方法時，Workday 會依據規則運算式語法使用伺服器端篩選。範例： - wd.ftp.method 是 'get' 和 wd.ftp.file.pattern 是 *.xml：Workday 會取得 FTP 伺服器上指定目錄的第一個 .xml 檔案。 - wd.ftp.method 是 'mget' 和 wd.ftp.file.pattern 是 *.xml：Workday 會取得 FTP 伺服器上指

參數	說明
	定目錄的所有 .xml 檔案，並以 ZIP 格式傳回這些檔案。 wd.ftp.method 是 'list' 和 wd.ftp.file.pattern 是 ^.*\.(xml)\$：Workday 會列出 FTP 伺服器上指定目錄的所有 .xml 檔案。
wd.ftp.temp.file.name	指定執行 FTP 時是否應使用臨時檔案 put 訊息動作
wd.ftp.timeout	訊息動作的逾時時間 (以毫秒為單位)。
wd.ftp.directory	相對或完全限定目錄路徑。
wd.ftp.private.key	與用於 SFTP 驗證的使用者名稱相關聯的私人金鑰。已使用附件通訊協定和 Workday ID 的 URI 格式。範例：attachment:privatekey/{WID}，其中 WID 對應於 32 個字元的 Workday ID。
wd.ftp.max.attempts	嘗試存取 FTP 伺服器的次數上限。
wd.ftp.debug	指定 Workday 是否會將偵錯記錄新增至記錄。
wd.ftp.proxy.host	FTP Proxy 伺服器的主機名稱。
wd.ftp.proxy.port	FTP Proxy 伺服器連接埠。
wd.ftp.proxy.username	用於存取 FTP Proxy 伺服器的使用者名稱。
wd.ftp.proxy.password	FTP Proxy 伺服器的密碼。
wd.ftp.proxy.type	Proxy 伺服器的類型。
wd.ftp.input.content.type	指定使用 FTP 擷取的檔案內容類型 get。範例：text/plain; charset=Big5。

參照：GetEventConfigurations 子組合屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	GetEventConfigurations 子組件在程序集中的唯一 ID。
將回覆送交至	routes-response-to	GetEventConfigurations 子組件的回應訊息目的地。
執行時間	execute-when	決定 GetEventConfigurations 子組件是否執行的條件。
端點	endpoint	指定子組合的目標端點地址。GetEventConfigurations 元件支援 vm:// 端點地址範例：vm://wcc/GetEventConfigurations。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存 步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組合併。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組合併傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組合併後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
ie.event.wid	您要擷取其服務配置的整合事件的 Workday ID。自動填入 lp.isSet() ? lp.getIntegrationEventWID() : null。
is.event.disable.msg.storage	設定為時關閉訊息儲存區 true。
ie.event.wws.version	整合事件 Web 服務版本。

參照：GetEventDocuments 子組合屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	GetEventDocuments 子組合作在程序集中的唯一 ID。
將回覆送交至	routes-response-to	GetEventDocuments 子組合件的回應訊息目的地。
執行時間	execute-when	決定 GetEventDocuments 子組合件是否執行的條件。
端點	endpoint	子組合件的目標端點地址。GetEventDocuments 元件支援 vm:// 端點地址範例：vm://wcc/GetEventDocuments。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組合件。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組合件傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。

屬性	XML 屬性內容名稱	說明
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組件後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
ie.event.wid	您要擷取其服務配置的整合事件的 Workday ID。自動填入 lp.isSet() ? lp.getIntegrationEventWID() : null。這表示如果已在中介環境定義中指定啟動文件，則參數值會預設為整合事件 WID，否則會預設為空值。

參照：GetIntegrationEvent 子組件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	組件內 GetIntegrationEvent 子組件的唯一 ID。
將回覆送交至	routes-response-to	GetIntegrationEvent 子組件的回應訊息目的地。
執行時間	execute-when	決定 GetIntegrationEvent 子組件是否執行的條件。
端點	endpoint	子組件的目標端點地址。GetIntegrationEvent 元件支援 vm:// 端點地址範例：vm://wcc/GetIntegrationEvent。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存 步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。

屬性	XML 屬性內容名稱	說明
		請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組合併。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組合併傳播至呼叫程序集。 子裝配件在標示 MediationException 已中止。
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組合併後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
ie.event.wid	您要擷取其服務配置的整合事件的 Workday ID。

參照：GetIntegrationSystems 子組合體屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	GetIntegrationSystems 子程序集在程序集中的唯一 ID。
將回覆送交至	routes-response-to	GetIntegrationSystems 子組合體的回應訊息目的地。
執行時間	execute-when	決定 GetIntegrationSystems 子組合體是否執行的條件。
端點	endpoint	子組合體的目標端點地址。GetIntegrationSystems 元件支援 vm:// 端點地址範例：vm://wcc/GetIntegrationSystems。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存 步驟。

屬性	XML 屬性內容名稱	說明
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組套件。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組套件傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。
取消設定屬性	unset-properties	指定在本地輸出傳輸呼叫子組套件後，是否重設所有本地輸出屬性。

參數分頁

參數	說明
is.system.wid	您要擷取並剖析其服務配置的整合系統的 Workday ID。自動填入 lp.getIntegrationSystemRefWID()。

參照：PagedGet 子組套件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	PagedGet 子程序集在程序集中的唯一 ID。
將回覆送交至	routes-response-to	PagedGet 子組合件的回應訊息目的地。
執行時間	execute-when	決定是否執行 PagedGet 子組合件的條件。
端點	endpoint	子組合件的目標端點地址。PagedGet 元件支援 vm://

屬性	XML 屬性內容名稱	說明
		端點地址範例：vm://wcc/PagedGet。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組合併。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組合併傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。
取消設定屬性	unset-properties	指定在本地輸出傳輸呼叫子組合併後，是否重設所有本地輸出屬性。

參數分頁

參數	說明
is.paged.get.request.current.page.xpath	要求文件中的 XPath 位置 PagedGet 組成要素可設定要求的頁面數。
is.paged.get.response.current.page.xpath	XPath 運算式，用於從 Web 服務回應訊息中擷取目前的頁面值。
is.paged.get.response.total.pages.xpath	XPath 運算式，用於從 Web 服務回應訊息中擷取總頁數的值。

參數	說明
<code>is.paged.get.response.total.results.xpath</code>	XPath 運算式，用於從 Web 服務回應訊息中擷取總計結果頁面值。
<code>is.paged.get.process.endpoint</code>	處理每個結果頁面的子組合件的 本地輸入 資訊端點。範例：<code>vm://payrollInterface/ProcessPayeesLocalIn</code>。 註：當您設定此參數時，Workday 會忽略匯總參數。 這些 <code>MediationContext</code> 屬性在處理子裝配件中可用： <code>is.paged.get.request.current.page.xpath</code> <code>is.paged.get.response.current.page.xpath</code> <code>is.paged.get.response.total.pages.xpath</code> <code>is.paged.get.response.total.results.xpath</code>
<code>is.paged.get.aggregate.header</code>	匯總回應的標頭值，採用 XML 片段形式。範例： <code><AllData></code> 。
<code>is.paged.get.aggregate.footer</code>	匯總回應的標頭值，採用 XML 片段形式。範例： <code></AllData></code> 。
<code>is.paged.get.aggregate.xpath</code>	匯總時要套用至回應訊息的 XPath 運算式。
<code>is.paged.get.namespaces</code>	指定在文件中無法使用的名稱空間。
<code>is.paged.get.application</code>	您要呼叫的 Web 服務所屬的應用程式群組。範例： <code>Human_Resources</code> 。
<code>is.paged.get.version</code>	您正在呼叫的公開 Web 服務版本。最佳做法是使用您建置整合時的最新 WWS 版本。請勿使用會隨著整合執行而變更的動態最新版本。 對於版本低於 2016.45 的程序集，此參數會自動填入 <code>util.assemblyVersionAsWWSVersion</code>。 對於版本為 2016.45 或更高版本的程序集，您必須明確提供 Web 服務版本。範例：<code>v27.2</code>。
<code>is.paged.get.page.zero</code>	指定是否應為總頁數為零的個案產生回應頁面。
<code>is.paged.get.parallel</code>	指定是否並行擷取 Web 服務回應，以啟用 <code>is.paged.get.process.endpoint</code> 參數，以處理每個分頁的回應。自動填入 <code>false</code>，表示會匯總 Web 服務回應，而不是並行擷取。 Workday 會使用獨有的環境定義處理並行回應，這些環境定義可讓您以唯讀的方式存取在輸入 <code>PagedGet</code> 前設定的屬性值和變數。當子裝配件處理完每個頁面時，您會遺失在處理期間設定的任何屬性或變數。<code>PagedGet</code> 仍會設定 <code>is.paged.get.total.pages</code> 和 <code>is.paged.get.total.results</code> 呼叫的屬性 <code>MediationContext</code>。該 <code>is.paged.get.current.page</code> 屬性是在並行分頁環境定義中設定。

參數	說明
	Workday 在相同的環境定義中處理非並行回應，並處理子裝配件中的頁面。這表示您可以在子裝配件中設定屬性和變數，並檢查先前頁面的狀態。
<code>is.paged.get.store.requests</code>	指定是否要儲存訊息儲存區中每個資料頁面的 Web 服務要求訊息。
<code>is.paged.get.overall.timeout.seconds</code>	整體逾時 (秒)。預設值為 6 小時 (如果未設定)。您可以設定的最長逾時時間為 30 小時。
<code>is.paged.get.timeout.seconds</code>	處理每個頁面的逾時時間。預設值為 6 小時 (如果未設定)。不可超過整體逾時值。

簽出參數

在處理 PagedGet 子組合件的過程中，它會傳遞以下輸出參數值，以供下游元件使用：

輸出參數	說明
<code>is.paged.get.last.page</code>	Web 服務作業傳回的最後一頁。
<code>is.paged.get.current.page</code>	目前正在處理的頁面編號。
<code>is.paged.get.total.pages</code>	要處理的總頁數。
<code>is.paged.get.total.results</code>	Web 服務作業傳回的結果總數。

參照：PagedGetLocalPaging 子組合屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	<code>id</code>	PagedGetLocalPaging 子組合件在程序集中的唯一 ID。
將回覆送交至	<code>routes-response-to</code>	PagedGetLocalPaging 子組合件的回應訊息目的地。
執行時間	<code>execute-when</code>	一個條件，用於判斷是否執行 PagedGetLocalPaging 子組合件。
端點	<code>endpoint</code>	子組合件的目標端點地址。PagedGetLocalPaging 元件支援 <code>vm://</code> 端點地址範例： <code>vm://wcc/PagedGet</code> 。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	<code>store-message</code>	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	<code>transport-class</code>	視需要逐一指定替代實作類別。

屬性	XML 屬性內容名稱	說明
		您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組零件。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組零件傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。
取消設定屬性	unset-properties	指定在本地輸出傳輸呼叫子組零件後，是否重設所有本地輸出屬性。

參數分頁

參數	說明
is.paged.get.process.endpoint	指定將處理每個結果頁面的子組合件的本地輸入資訊端點。範例：vm://payrollInterface/ProcessPayeesLocalIn。 註：當您設定此參數時，Workday 會忽略匯總參數。 這些 MediationContext 屬性在處理子裝配件中可用： - is.paged.get.request.current.page.xpath - is.paged.get.response.current.page.xpath - is.paged.get.response.total.pages.xpath - is.paged.get.response.total.results.xpath
is.paged.get.page.size	每頁資料中要擷取的記錄數。自動填入 100。
is.paged.get.response.reference.wids.xpath	XPath 運算式，用於在「get all WIDs」呼叫回應訊息中尋找 WID。 範例：//wd:Payee_Reference/wd:ID[@wd:type='WID']。
is.paged.get.request.reference.wids.xpath	XPath 運算式，用於尋找在取得每個資料頁面時會指出 WID 的要求訊息的父系元素。

參數	說明
<code>is.paged.get.request.reference.wids.element</code>	要在每個資料頁面的要求中使用 WID 的元素名稱。
<code>is.paged.get.application</code>	您要呼叫的 Web 服務所屬的應用程式群組。範例：Human_Resources。
<code>is.paged.get.version</code>	您正在呼叫的公開 Web 服務版本。最佳做法是使用您建置整合時的最新 WWS 版本。請勿使用會隨著整合執行而變更的動態最新版本。 對於版本低於 2016.45 的程序集，此參數會自動填入 <code>util.assemblyVersionAsWSVersion</code> 。 對於版本為 2016.45 或更高版本的程序集，您必須明確提供 Web 服務版本。範例：v27.2。
<code>is.paged.get.xpath.ns.prefix</code>	所有 XPath 運算式中使用的 Workday XML 名稱空間的字首。自動填入 <code>wd</code> 。

參照：PdfPrintStep 子組合屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	<code>id</code>	PdfPrintStep 子組件在組件內的唯一 ID。
將回覆送交至	<code>routes-response-to</code>	PdfPrintStep 子組件的回應訊息目的地。
執行時間	<code>execute-when</code>	一個條件，決定是否執行 PdfPrintStep 子組件。
端點	<code>endpoint</code>	子組件的目標端點地址。PdfPrintStep 元件支援 <code>vm://</code> 端點地址範例： <code>vm://wcc/PdfPrintStep</code> 。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	<code>store-message</code>	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	<code>transport-class</code>	視需要逐一指定替代實作類別。 您可透過編輯 <code>WEB-INF/classes/spring/assembly-bean.xml</code> 檔案，全域變更實作類別。
忽略動態端點	<code>ignore-dynamic-endpoints</code>	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。

屬性	XML 屬性內容名稱	說明
		請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組合併。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組合併傳播至呼叫程序集。 子裝配件在標示 MediationException 已中止。
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組合併後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
pdf.report.design.variable	包含報告設計的整合變數的名稱，用於定義列印自訂報告的版面配置。範例： 'test.report.design'。 報告設計定義包含在副檔名為 .rptdesign 的檔案中。
pdf.workday.report.variable	包含原始 Workday 報告資料的整合變數的名稱。範例：'test.workday.report'。
pdf.timezone	用於格式設定的 Java 時區。範例： 'US/Eastern' 'GMT' 'CET'
pdf.language	用於格式設定的報告語言。範例： 'en_US' 'fr_FR' 'de_DE'
pdf.locale	用於格式設定的報告地區設定。範例： 'en_US' 'fr_FR' 'de_DE'
pdf.business.form.layout.wid	所使用的業務表單版面配置的 Workday ID。允許擷取在版面配置中定義為權杖的影像。
pdf.report.design.wid	正在使用的報告設計的 Workday ID。啟用翻譯。
pdf.apply.report.design.per.row	指定 Workday 是將報告設計套用至個別報告列，而不是整個報告。

relatedlinks
相關參考資料

[下一個層級：在 Studio 中使用 BIRT 整合](#)

參照：PutIntegrationEvent 子組件屬性

通用表格

屬性	XML 屬性內容名稱	說明
ID	id	組件內 PutIntegrationEvent 子組件的唯一 ID。
將回覆送交至	routes-response-to	PutIntegrationEvent 子組件的回應訊息目的地。
執行時間	execute-when	決定是否執行 PutIntegrationEvent 子組件的條件。
端點	endpoint	子組件的目標端點地址。PutIntegrationEvent 元件支援 vm:// 端點地址範例：vm://wcc/PutIntegrationEvent。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set-Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組件。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組件傳播至呼叫程序集。

屬性	XML 屬性內容名稱	說明
		子裝配件在標示 <code>MediationContext</code> 已中止。
取消設定屬性	<code>unset-properties</code>	指定在 本地輸出 傳輸呼叫子組件後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
<code>is.event.wid</code>	Workday 會將訊息連結到的整合事件的 Workday ID。 如果您未指定此值，且此值不在 <code>lp MVEL</code> 變數時，Workday 會使用 <code>is.system.wid</code> 參數，將訊息連結至整合系統。自動填入 <code>lp.isSet() ? lp.getIntegrationEventWID() : null</code> 。
<code>is.system.wid</code>	整合系統的 Workday ID。 自動填入 <code>lp.isSet() ? lp.getIntegrationSystemRefWID() : null</code> 。 只有在整合事件 ID 不可用時，Workday 才會使用此參數。
<code>is.process.desc</code>	整合事件說明。Workday 會在其「流程監控」中顯示此值。建立新的整合事件時為必填。
<code>is.event.initiated</code>	整合事件的起始日期和時間。如果您輸入字串值 'current'，系統會將目前的日期和時間傳送至 Workday。
<code>is.event.completed</code>	整合事件完成的日期和時間。如果您輸入字串值 'current'，系統會將目前的日期和時間傳送至 Workday。
<code>is.response.msg</code>	最近的整合事件訊息或活動。自由格式文字欄位
<code>is.event.member.wids</code>	以逗號分隔的 Workday ID 清單，代表連結至此整合事件的成員。
<code>is.percent.complete</code>	Workday 用來維護完成百分比進度列的值。值可以是任何可轉換為 Java <code>BigDecimal</code> 物件的類型。必須是 0 到 100 之間的值。

參照：PutIntegrationMessage 子組件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	<code>id</code>	組件內 <code>PutIntegrationMessage</code> 子組件的唯一 ID。
將回覆送交至	<code>routes-response-to</code>	<code>PutIntegrationMessage</code> 子組件的回應訊息目的地。

屬性	XML 屬性內容名稱	說明
執行時間	execute-when	決定是否執行 PutIntegrationMessage 子組合件的條件。
端點	endpoint	子組合件的目標端點地址。PutIntegrationMessage 元件支援 vm:// 端點地址範例：vm://wcc/PutIntegrationMessage。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存 步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組合件。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組合件傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組合件後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
is.event.wid	Workday 會將訊息連結到的整合事件的 Workday ID。

參數	說明
	如果您未指定此值，且此值不在 lp MVEL 變數時，Workday 會使用 is.system.wid 參數，將訊息連結至整合系統。自動填入 lp.isSet() ? lp.getIntegrationEventWID() : null。
is.system.wid	整合系統的 Workday ID。 自動填入 lp.isSet() ? lp.getIntegrationSystemRefWID() : null。 只有在整合事件 ID 不可用時，Workday 才會使用此參數。
is.message.severity	指定訊息嚴重程度層級：'CRITICAL', 'DEBUG', 'ERROR', 'INFO'，或 'WARNING'。自動填入 'INFO'。
is.message.summary	傳送至 Workday 的訊息摘要。
is.message.detail	傳送至 Workday 的訊息詳細資料。
is.message.detail.richtext	決定是否將訊息詳細資料設為 RTF 格式。自動填入 true。
is.document.variable.name	中的變數名稱 MediationContext，包含組件儲存步驟的 Atom 結構描述輸出。變數內容用於新增整合事件的參照。 該 store 用來建立文件的步驟必須有如下的結構描述屬性內容設定：schema="http://www.w3.org/2005/Atom"。
is.document.file.name	文件的檔案名稱。如果您未提供，Workday 會使用 Atom 文件中的文件標題。
is.document.owner.wid	文件擁有者的 Workday ID。若未提供，Workday 會使用目前整合系統使用者的 ID。
is.document.labels	以逗號分隔的標籤清單，當文件在整合事件 ID 或相關聯的業務流程範圍內時，系統會套用到文件。
is.document.deliverable	指定新增至整合事件的文件是否自動傳送給客戶或外部提供者。自動填入 'true'。
is.document.delivery.services	以逗號分隔的整合服務名稱清單，用於標記要附加至整合事件的文件。預設情況下，Workday 會使用與整合系統相關聯的所有發送服務。
is.document.retrieved	指定是否將文件視為已由擷取服務擷取。自動填入 'false'。
is.message.targets	此訊息的目標物件參照清單 (逗號分隔)。這些目標會變成 Workday 應用程式「檢視整合訊息」分頁中的物件連結。
is.message.targets.type	目標參照類型。自動填入 'WID'。
is.message.secured.instance.refs	此訊息的安全實例參照清單 (逗號分隔)。
is.message.secured.instance.refs.type	啟用後，系統會覆寫要覆寫的安全實例參照的預設類型。自動填入 'WID'。

參數	說明
<code>is.message.storage.enabled</code>	決定是否儲存訊息。自動填入 'false'。

參照：SalesforceConnector 子組件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	SalesforceConnector 子組件在集合內的唯一 ID。
將回覆送交至	routes-response-to	SalesforceConnector 子組合的回應訊息目的地。
執行時間	execute-when	決定 SalesforceConnector 子組件是否執行的條件。
端點	endpoint	子組合的目標端點地址。SalesforceConnector 元件支援 <code>vm://</code> 端點地址範例： <code>vm://wcc/SalesforceConnector</code> 。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存步驟。
傳輸類別	transport-class	視需要逐一指定替代實作類別。 您可透過編輯 <code>WEB-INF/classes/spring/assembly-bean.xml</code> 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 <code>set-Dynamic-endpoint</code> 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 <code>True</code> 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組件。設為 <code>False</code> 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組件傳播至呼叫程序集。

屬性	XML 屬性內容名稱	說明
		子裝配件在標示 MediationContext 已中止。
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組件後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
wd.sf.auth.username	您的 salesforce.com 開發人員帳戶使用者名稱。
wd.sf.auth.password	您的 salesforce.com 開發人員帳戶密碼。
wd.sf.msg.varname	Workday 傳送至 salesforce.com 端點的 XML 訊息。
wd.sf.auth.token	您的 salesforce.com 開發人員帳戶安全權杖。
wd.sf.auth.endpoint	選用的 salesforce.com 端點。預設端點已配置為子裝配件的一部分。http-out 運輸。
wd.sf.api.custom.version	選用的 salesforce.com API 版本。如果您未提供值，則會使用端點 API 版本。

PrismAnalytics 子裝配件

步驟：將 CSV 資料傳送至 Prism Analytics

開始之前

安全性：「系統」功能區中的「安全性管理」網域。

執行這項作業的原因和時機

Workday Studio 的 PrismAnalytics 元件可讓您將 CSV 資料傳送至 Workday Prism Analytics。在您配置程序集以使用 PrismAnalytics 之前，您必須在 Workday 中註冊 Prism API 用戶端並取得 REST API 端點。您還必須建立來源 CSV 資料的 JSON 說明。

程序

1. [註冊 Workday Prism Analytics API 用戶端](#) 頁 102.
在 Workday 中，註冊整合 API 用戶端，並將 Prism Analytics 設為作用域。取得 PrismAnalytics 元件所需做為參數的 Client ID、Client Secret 和 Refresh Token 值。
2. [取得 Workday REST API 端點](#) 頁 102.
在 Workday 中，取得 PrismAnalytics 元件所需做為參數的 Workday REST API 端點。
3. [說明來源 CSV 資料](#) 頁 103.
以 JSON 文字描述來源 CSV 資料，以便做為變數傳遞至 PrismAnalytics 元件。
4. [配置具有 PrismAnalytics 子裝配件的裝配件](#) 頁 108.
在 Workday Studio 中，建立程序集，以擷取 CSV 資料，並將其傳遞至正確配置的 PrismAnalytics 元件。

relatedlinks

相關參考資料

[Workday 31 功能更新公告：Workday Studio](#)

註冊 Workday Prism Analytics API 用戶端

開始之前

安全性：「系統」功能區中的「安全性管理」網域。

執行這項作業的原因和時機

若要用 Workday Studio 的 PrismAnalytics 元件，您必須在 Workday 中註冊 API 用戶端，並將 Prism Analytics 設為作用域。此元件需要 Workday 在註冊過程中產生的用戶端 ID 和用戶端密鑰。還需要重新整理權杖值。

程序

1. 存取「為整合註冊 API 用戶端」任務。執行此任務時，請參考以下資訊：

選項	敘述
不會到期的重新整理權杖	防止重新整理權杖逾時。已由 Workday 自動選取。不清除。
範圍 (功能區)	選取 Prism 分析。
包含 Workday 自有的範圍	提供以下權限：存取不在任何功能區中的核心 Workday 網域。選取此核取方塊。

2. 複製用戶端 ID 和用戶端密鑰的值。Studio 的 PrismAnalytics 元件在執行 `wd.prism.component.client.id` 和 `wd.prism.component.client.secret` 參數
3. 從相關動作功能表中，選取「API 用戶端 > - 管理整合的重新整理權杖」。
4. 選取要與重新整理權杖相關聯的 Workday 帳戶。帳戶必須有權在 Workday 中建立 Prism 表。
註：您無法以整合系統使用者的身分執行這些步驟。
5. 選取產生新的重新整理權杖。
6. 複製重新整理權杖值。PrismAnalytics 元件在執行時必須用到 `wd.prism.component.refresh.token` 參數

取得 Workday REST API 端點

開始之前

安全性：「系統」功能區中的「安全性管理」網域。

執行這項作業的原因和時機

Studio 的 PrismAnalytics 元件需要 Workday REST API 端點。您可以在 Workday 中找到此端點。

程序

1. 存取「檢視 API 用戶端」報告。
2. 複製 Workday REST API 端點值。
範例：`https://i-07d55d20e81e2d191.workdaysuv.com/ccx/api/v1/super`。
3. 刪除 `https://` 在開頭和 `/ccx/api/v1/<tenant_name>` 放在最後。
範例：`i-07d55d20e81e2d191.workdaysuv.com`。
4. 複製文字。Studio 的 PrismAnalytics 元件在執行其作業時需要該元件。
`wd.prism.component.api.hostname` 參數

說明來源 CSV 資料

執行這項作業的原因和時機

Workday Studio 的 PrismAnalytics 元件需要來源 CSV 資料的說明。您可以在所選工具中建立 JSON 文字，稍後在 Studio 中使用。例如，將其貼到 撰寫 步驟的 訊息建置器 分頁上的文字訊息中。

程序

1. 建立包含 2 個主要節點的 JSON 文字：
 - `parseOptions`：CSV 資料格式的相關資訊，包括字元集和分隔符號。
 - `fields`：CSV 資料中各欄位的相關資訊，包括資料類型和格式設定。
2. 在 `parseOptions` 節點，您可以定義以下欄位：

欄位名稱	類型	必填	說明
<code>charsetName</code>	物件	是	CSV 資料檔案的字元集。範例： <pre>"charset": { "id": "Encoding=UTF-8" }</pre> 目前，Workday 僅支援 UTF-8 編碼格式。Workday 最終會以 Unicode 格式儲存資料。沒有 Unicode 等量字元會儲存為 Unicode 取代字元，即黑色菱形背景上的問號。
<code>fieldsEnclosedBy</code>	String	否	在 CSV 資料中用來括住欄位的字元。預設值為雙引號。如果欄位內使用了雙引號，請在該欄位前面加上另一個雙引號進行轉義。範例： <pre>"fieldsEnclosedBy": "\\"</pre>
<code>fieldsDelimitedBy</code>	String	是	CSV 資料中用於分隔欄位值的字元。範例： <pre>"fieldsDelimitedBy": ","</pre>

欄位名稱	類型	必填	說明
headerLinesToIgnore	數字	否	解析器忽略的標頭行數。設為 0 CSV 資料沒有標頭時範例： <pre>"headerLinesToIgnore": 1</pre>
type	物件	是	輸入檔案的類型。目前，Workday 僅支援分隔格式檔案。範例： <pre>"type": { "id": "Schema_File_Type=Delimit" }</pre> 註：此值區分大小寫。請注意「分隔符號」中的大寫「D」。

3. 在 fields 節點，您可以定義以下欄位：

欄位名稱	類型	必填	說明
ordinal	數字	是	欄位在 CSV 資料集中的順序 (索引)。序數值： - 必須以 1 開頭。 - 必須是連續的。 - 無法略過任何數字。 - 上限為 1000。
name	String	是	CSV 資料集欄位的名稱。欄位名稱： - 在資料集中不能重複。 - 最多可包含 255 個字元。 - 只能包含英數字元和底線字元。 - 必須以字母開頭。 - 結尾不能使用底線。 - 開頭不可使用 WPA_。

欄位名稱	類型	必填	說明
description	String	否	欄位的說明。最多 1000 個字元。範例： <pre>"description": "The worker's full name."</pre>
type	物件	是	欄位的類型。選項包括： • 文字 • 數值 • 布林運算子 • 日期 範例： <pre>"type": { "id": "Schema_Field_Type=Da</pre>
parseFormat	String	是 (適用於日期類型)	日期的格式。範例： <pre>"parseFormat": "dd-MM-yy HH24:mm:ss"</pre>
precision	數字	是 (適用於數值類型)	數值中的最大位數。包含小數點左右的所有數字，但不包含小數點本身。最大值為 38。範例： <pre>"precision": 38</pre>
scale	數字	是 (適用於數值類型)	數值中小數點右側的位數。必須小於 precision 值。範例： <pre>"scale": 2</pre>
defaultValue	文字	否	欄位的預設值 (如果在上傳檔案中未提供任何欄位值) 使用雙引號表示預設文字值。範例： "N/A"。 預設數值必須與指定的 precision 和 scale。 預設日期值必須採用以下格式： yyyy-MM-dd'T'HH:mm:ss'Z'。

欄位名稱	類型	必填	說明
required	布林運算子	否	指定欄位是否必須包含值。設為 True 若為必填欄位，則設為 False。如果欄位為必填，則在任何插入、更新和更新插入資料作業中，其值都不能為空值。
externalID	布林運算子	「是」(以利用更新、更新插入和刪除作業)。	指定欄位是否代表外部 ID。設為 True 將欄位指定為外部 ID，否則設為 False。 由於外部 ID 欄位不可為空值，因此您應指定任何外部 ID 欄位為必填欄位。您只能指定 1 個欄位做為外部 ID。 如果您要使用更新、插入和刪除功能，請在新表格中指定外部 ID。 若要對現有表格使用更新、插入或刪除功能，您必須對其進行修改，使其具有一個指定為外部 ID 的欄位。Prism 公開 API 不支援修改包含資料的表格。在這種情況下，請在 Workday 中進行修改。

範例

您的 CSV 來源檔案對此表格進行說明：

正職員工	正職員工 ID	職稱	僱用日期	年薪
Logan McNeil	21001	人力資源副總裁	1/1/2010	213298.00
奧立華	21003	首席資訊官	1/1/2010	394935.00
馬克西米利安·施耐德	21004	首席營運官	1/1/2010	255373.00
Teresa Serrano	21005	會計長	1/1/2010	291785.00

您可以使用 JSON 進行說明，如下所示：

```
{
  "parseOptions": {
    "fieldsDelimitedBy": ", ",

```

```

"fieldsEnclosedBy": "\"",
"headerLinesToIgnore": 1,
"charset": {
  "id": "Encoding=UTF-8"
},
"type": {
  "id": "Schema_File_Type=Delimited"
}
},
"fields": [
  {
    "ordinal": 1,
    "name": "Employee",
    "description": "Employee",
    "precision": 255,
    "scale": 0,
    "parseFormat": null,
    "type": {
      "id": "Schema_Field_Type=Text"
    }
  },
  {
    "ordinal": 2,
    "name": "Employee_ID",
    "description": "Employee ID",
    "precision": 38,
    "scale": 0,
    "parseFormat": null,
    "type": {
      "id": "Schema_Field_Type=Numeric"
    }
  },
  {
    "ordinal": 3,
    "name": "Job_Title",
    "description": "Job Title",
    "precision": 255,
    "scale": 0,
    "parseFormat": null,
    "type": {
      "id": "Schema_Field_Type=Text"
    }
  },
  {
    "ordinal": 4,
    "name": "Hire_Date",
    "description": "Hire Date",
    "precision": 0,
    "scale": 0,
    "parseFormat": "MM/dd/yy",
    "type": {
      "id": "Schema_Field_Type=Date"
    }
  },
  {
    "ordinal": 5,
    "name": "Annual_Salary",
    "description": "Annual Salary",
    "precision": 38,
    "scale": 2,
    "type": {
      "id": "Schema_Field_Type=Numeric"
    }
  }
]

```

```

    },
    "schemaVersion": {
      "id": "Schema_Version=1.0"
    }
  }
}

```

relatedlinks

相關參考資料

[參照：表格和資料集中外部資料支援的日期格式](#)

配置具有 **PrismAnalytics** 子裝配件的裝配件

執行這項作業的原因和時機

要將 CSV 資料傳送至 Prism Analytics，您的程序集必須包含符合以下條件的 PrismAnalytics 元件：

- 接收 CSV 資料做為輸入，無論是在訊息的根部分中，還是做為變數。
- 接收包含 CSV 資料的 JSON 說明的變數。
- 有多個參數配置正確。

程序

1. 配置您的程序集，以取得一個或多個來源 CSV 檔案。根據您用來取得 CSV 資料的方法，您可能需要提供變數名稱或設定 MIME 類型。

執行此任務時，請參考以下說明：

CSV 取得方式	附註
sftp-out 元件	• 在元件的「輸入檔案模式」屬性中指定 CSV 檔案名稱。您可以使用萬用字元。 • 使用元件的 get 方法來擷取單一檔案或其 mget 方法來擷取多個檔案。 • Studio 會將 CSV 資料新增至訊息根部分。 • 不需要變數名稱。 • Studio 會自動正確設定 MIME 類型。
擷取或偵聽服務	• Studio 會將 CSV 資料新增至變數。 • 使用「」為變數命名 wd.prism.component.csv.input.varname 參數 • Studio 會自動正確設定 MIME 類型。
任何其他方法	• Studio 會將 CSV 資料新增至訊息根部分或變數。 • 如果適用，請使用 wd.prism.component.csv.input.varname 參數 • 將根部分或變數的 MIME 類型設定為： • text/csv • 純文字/csv • 應用程式/zip

2. 在寫入步驟的「訊息建置器」分頁上建立文字訊息，並新增說明資料的 JSON 文字。輸出變數。範例：temp.schema。

註：您可以將 JSON 文字直接新增至訊息建置器。或者，您可以將其寫入您選擇的工具中並貼上。

3. 配置 PrismAnalytics 元件的參數：

參數	說明
<code>wd.prism.component.schema.varname</code>	包含描述資料的 JSON 結構描述的變數。
<code>wd.prism.component.table.name</code>	Prism 表的名稱。表格名稱： - 在資料目錄中不能重複。 - 最多可包含 255 個字元。 - 只能包含英數字元和底線字元。 - 必須以字母開頭。 - 結尾不能使用底線。
<code>wd.prism.component.client.id</code>	Workday 指派給 API 用戶端的用戶端 ID。存取「檢視 API 用戶端」報告，然後檢視用於整合的 API 用戶端分頁。
<code>wd.prism.component.client.secret</code>	Workday 指派給 API 用戶端的用戶端密鑰。存取「檢視 API 用戶端」報告，然後檢視用於整合的 API 用戶端分頁。
<code>wd.prism.component.refresh.token</code>	Workday 指派給 API 用戶端的重新整理權杖。存取「檢視 API 用戶端」報告，然後檢視用於整合的 API 用戶端分頁。
<code>wd.prism.component.operation</code>	對 Prism 表執行的作業。「取代」是目前唯一受支援的選項。其會建立表格或取代其中一個同名的表格。
<code>wd.prism.component.csv.input.varname</code>	(選填)包含 CSV 輸入的變數。若您要將 CSV 資料新增至訊息根部分，則不需要。 註：當您將 PrismAnalytics 元件新增至集合時，Studio 不會自動選取包含此參數。如有需要，請手動選取，或稍後在元件的「參數」分頁中新增。
<code>wd.prism.component.api.hostname</code>	Workday REST API 端點。存取「檢視 API 用戶端」報告。移除 <code>https://</code> 在開頭和 <code>/ccx/api/v1/<tenant-name></code> 放在最後。 範例： <code>i-09e55d32d61e2e287.workdaysuv.com</code> 。
<code>wd.prism.component.table.description</code>	(選填)表格的說明。最多可包含 255 個字元。
<code>wd.prism.component.table.description.displayname</code>	(選填)表格的顯示名稱。最多可包含 255 個字元。
<code>wd.prism.component.table.field.varname</code>	(選填)定義 Prism 表。將定義寫入變數，然後使用 <code>vars['variable-name']</code> 將其傳遞至此參數。

下一步

部署並啟動整合。

參照：**PrismAnalytics** 子裝配件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
ID	id	PrismAnalytics 子裝配件的唯一 ID。
將回覆送交至	routes-response-to	子裝配件在組套件內的回應訊息目的地。PrismAnalytics 子組套件
執行時間	execute-when	決定 PrismAnalytics 子裝配件是否執行的條件。
端點	endpoint	子組合件的目標端點地址。PrismAnalytics 元件支援 vm:// 端點地址範例：vm://wcc/PrismAnalytics。

進階分頁

屬性	XML 屬性內容名稱	說明
商店訊息	store-message	除非 Workday 支援團隊另有建議，否則請指定「無」。 若要儲存訊息，請使用 Studio 的儲存 步驟。
傳輸類別	transport	視需要逐一指定替代實作類別。 您可透過編輯 WEB-INF/classes/spring/assembly-bean.xml 檔案，全域變更實作類別。
忽略動態端點	ignore-dynamic-endpoints	指定是否要忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值，或使用該值覆寫傳輸的「端點」屬性中的值。 請使用 set- Dynamic-endpoint 步驟指定 WS-Addressing 詳細資料。
複製要求	clone-request	指定是否要在子裝配件之間連結處理。設為 True 可建立原始訊息的副本，這樣任何修改要求訊息的子組合體都不會影響其他組套件。設為 False 可傳遞原始訊息。
傳播中止	propagate-abort	指定中止標幟是否會從子組套件傳播至呼叫程序集。 子裝配件在標示 MediationContext 已中止。

屬性	XML 屬性內容名稱	說明
取消設定屬性	unset-properties	指定在 本地輸出 傳輸呼叫子組 合併後，是否重設所有 本地輸出 屬性。

參數分頁

參數	說明
<code>wd.prism.component.schema.varname</code>	包含描述資料的 JSON 結構描述的變數。
<code>wd.prism.component.table.name</code>	Prism 表格在資料目錄中的 ID。表格名稱： - 在資料目錄中不能重複。 - 最多可包含 255 個字元。 - 只能包含英數字元和底線字元。 - 必須以字母開頭。 - 結尾不能使用底線。 此元件無法用來更新 Prism 表格名稱，只能建立一個。
<code>wd.prism.component.client.id</code>	Workday 指派給 API 用戶端的用戶端 ID。存取「檢視 API 用戶端」報告，然後檢視用於整合的 API 用戶端分頁。
<code>wd.prism.component.client.secret</code>	Workday 指派給 API 用戶端的用戶端密鑰。
<code>wd.prism.component.refresh.token</code>	Workday 指派給 API 用戶端的重新整理權杖。存取「檢視 API 用戶端」報告，然後檢視用於整合的 API 用戶端分頁。
<code>wd.prism.component.operation</code>	對 Prism 資料集執行的作業。範例：Replace, Append。 Replace 建立表格或取代其中一個同名的表格。
<code>wd.prism.component.csv.input.varname</code>	(選填)包含 CSV 輸入的變數。若您要將 CSV 資料新增至訊息根部分，則不需要。 註：當您將 PrismAnalytics 元件新增至集合時，Studio 不會自動選取包含此參數。如有需要，請手動選取，或稍後在元件的「參數」分頁中新增。
<code>wd.prism.component.api.hostname</code>	Workday REST API 端點。存取「檢視 API 用戶端」報告。移除 <code>https://</code> 在開頭和 <code>/ccx/api/v1/<tenant-name></code> 放在最後。 範例： <code>i-09e55d32d61e2e287.workdaysuv.com</code> 。
<code>wd.prism.component.table.description</code>	(選填)表格的說明。最多可包含 255 個字元。
<code>wd.prism.component.table.display.name</code>	(選填)表格的顯示名稱。最多可包含 255 個字元。 此元件無法用來更新 Prism 表格顯示名稱，只能建立一個。
<code>wd.prism.component.table.field.varname</code>	(選填)定義 Prism 表。將定義寫入變數，然後使用 <code>vars['variable-name']</code> 將其傳遞至此參數。

錯誤處理程式元件

概念：錯誤處理程式元件

Studio 在 調色盤上提供了 3 種不同的 錯誤處理常式：

名稱	函數
傳送錯誤	可讓您使用標準元件和步驟處理錯誤。常見的模式是使用「send-error」，透過呼叫 PutIntegrationMessage 通用元件，將錯誤報告給整合事件。
log-error	將目前的錯誤記錄至輸出記錄。包含 Java 例外的錯誤訊息和堆疊追蹤。用於不需要個別顯示整合事件的錯誤，但您想在發生錯誤時記錄訊息。避免記錄大型訊息，因為這類訊息會降低效能，且可能遭到截斷，進而導致診斷資料遺失。
custom-error-handler	可讓您提供使用 Java 處理錯誤的 Spring bean。當錯誤處理需要複雜的 Java 代碼時使用。

您可以將錯誤處理常式新增為中介元件的子系元素，或新增為集合中的個別元素。新增至中介元件的錯誤處理常式為本端。程序集上的錯誤處理常式是全域的。

當 Workday 在 Studio 整合中發生錯誤時，會停止正常處理並展開已處理的元件鏈，尋找要呼叫的錯誤處理程式。其一律會先嘗試在本機處理錯誤。如果沒有可用的本機錯誤處理程式，職責會轉移給全域錯誤處理程式。如果您想要精細控制發生錯誤時的處理方式，或需要報告發生錯誤時整合作業的詳細環境定義，請使用本機錯誤處理程式。將全域錯誤處理程式視為失效保護。

如果每個錯誤處理常式已觸發，但仍有錯誤未標示為已處理，Workday 會處理訊息、從例外產生錯誤，並將程序集標示為已終止。整合說明為「已完成但有錯誤」。

顯著的預設行為：

- 本端錯誤處理程式只會處理髮生在其上層中介元件中的錯誤。但是，他們也可以處理下游組成要素中的錯誤。若要啟用此行為，請將包含錯誤處理常式的中介元件的「下游錯誤處理句柄」屬性設為 True。
- 呼叫錯誤處理程式時，Workday 會將錯誤標示為「已處理」。不過，如果您將錯誤處理程式的「重新引發錯誤」屬性設為 True，Workday 就不會清除錯誤。因此，會交由下一個範圍內的上游錯誤處理程式或全域錯誤處理程式處理。使用此模式可在本地報告錯誤的詳細資料，但要在程序集的更高層級處理整體失敗或完成。
- Workday 處理錯誤時，會從呼叫的錯誤處理常式重新開始處理訊息，並繼續沿回應路徑前往發生錯誤的元素。但是，您可以指定在發生錯誤的元素處繼續處理。若要啟用此行為，請將包含錯誤處理程式的中介元件 () 的「發生錯誤後繼續」屬性設為 復原。如果錯誤處理程式呼叫的代碼可更正錯誤條件，進而重試失敗的作業，請使用此行為。

您可以在「屬性」檢視中新增 MVEL 條件運算式，以更有效地控制錯誤處理常式。當條件運算式存在時，只有在條件運算式的計算結果皆為 True 時，Workday 才會呼叫錯誤處理常式。

您可以使用 context.setError 或 context.SetException MVEL 或 Java 代碼中的方法使用時 setError 方法中，您可以配置錯誤處理程式上的條件運算式可偵測到的錯誤 ID。當您在本機處理錯誤，但想要將錯誤類型的某些詳細資料傳遞至上游錯誤處理常式以供一般處理時，可使用此技巧。

您也可以使用 路由 元件來處理錯誤：

- failover-strategy 嘗試使用路由，如果發生錯誤，則會執行故障轉移路由，並將錯誤標示為「已處理」。
- custom-strategy 可讓您提供 Spring bean。java - 的 RoutingStrategy 介面，其中包含 isHandleError 方法使用此方法可指定由策略處理錯誤。

參照：自訂錯誤處理程式元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
錯誤處理程式 ID	id	程序集中的 custom-error-handler 元件的唯一 ID。
重新拋出錯誤	rethrow-error	指定 custom-error-handler 元件在完成處理後，是否將錯誤傳回任何上游錯誤處理常式。預設值為 False，表示 custom-error-handler 元件會單獨處理錯誤，然後將其標示為已處理。 您也可以呼叫「」，將錯誤標示為已處理。setErrorHandled 的方法 MediationContext 物件。範例： mediationContext.setErrorHandled(tr
Spring Bean	ref	實作以下 2 個介麵類別之一的 Spring bean 的 ID： - OnErrorHandler. - ConditionalOnErrorHandler，延伸至 OnErrorHandler 使用布林值方法 invokeHandleError(MediationContext)。如果此方法傳回 true，Workday 會呼叫 handleError(MediationContext context) 方法 註： - 如果您實施 ConditionalOnErrorHandler 介面時，您會覆寫您定義的任何條件運算式。

運算式分頁

屬性	XML 屬性內容名稱	說明
條件運算式	condition-expression	可讓您直接輸入 MVEL 條件運算式。 如果所有運算式的計算結果都為 true，或者如果您未提供運算式，則會呼叫錯誤處理常式。否則，不會叫用錯誤處理程式，錯誤會依處理鏈繼續進行，且 Workday 會新增一個類型為「稽核記錄」的項目。ERROR_HANDLER_SKIPPED。

屬性	XML 屬性內容名稱	說明
		若要在任何現有的列下方新增運算式，請按一下 新增，然後輸入運算式。 若要在現有的列上方新增運算式，請將游標移至該列，然後在新增 下拉式功能表上按一下 插入。 使用 運算式 分頁右側的按鈕，刪除列或重新排序。

參照：記錄錯誤元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
錯誤處理程式 ID	id	程序集內 記錄錯誤 元件的唯一 ID。
重新拋出錯誤	rethrow-error	指定在處理完成後，log-error 元件是否會將錯誤傳回任何上游錯誤處理常式。預設值為 False，表示 日誌錯誤 元件會單獨處理錯誤，然後將其標示為已處理。 您也可以呼叫「」，將錯誤標示為已處理。setErrorHandled 的方法 MediationContext 物件。範例： mediationContext.setErrorHandled(tr
層級	level	Log4J 記錄層級。有 5 個選項： - 偵錯 - info - 警告 - 錯誤 - 嚴重

進階分頁

屬性	XML 屬性內容名稱	說明
種類	category	指定 Log4J 記錄器名稱的一部分。預設值為 DEFAULT。

運算式分頁

屬性	XML 屬性內容名稱	說明
條件運算式	condition-expression	可讓您直接輸入 MVEL 條件運算式。

屬性	XML 屬性內容名稱	說明
		如果所有運算式的計算結果都為 true，或者如果您未提供運算式，則會呼叫錯誤處理常式。否則，不會叫用錯誤處理程式，錯誤會依處理鏈繼續進行，且 Workday 會新增一個類型為「稽核記錄」的項目。ERROR_HANDLER_SKIPPED。 若要在任何現有的列下方新增運算式，請按一下 新增，然後輸入運算式。 若要在現有的列上方新增運算式，請將游標移至該列，然後在新增 下拉式功能表上按一下 插入。 使用 運算式 分頁右側的按鈕，刪除列或重新排序。

參照：傳送錯誤元件屬性

通用分頁

屬性	XML 屬性內容名稱	說明
錯誤處理程式 ID	id	集合中 傳送錯誤 元件的唯一 ID。
重新拋出錯誤	rethrow-error	指定 send-error 元件在完成處理後，是否將錯誤傳回任何上游錯誤處理常式。預設值為 False，表示「send-error」元件會單獨處理錯誤，然後將其標示為已處理。 您也可以呼叫「」，將錯誤標示為已處理。setErrorHandled 的方法 MediationContext 物件。最小化生產量範例： context.setErrorHandled(true)。
路由至	routes-to	程序集元素的 ID，send-error 組成要素會將訊息導向該元素。

運算式分頁

屬性	XML 屬性內容名稱	說明
條件運算式	condition-expression	可讓您直接輸入 MVEL 條件運算式。 如果所有運算式的計算結果都為 true，或者如果您未提供運算式，則會呼叫錯誤處理常式。否則，不會叫用錯誤處理

屬性	XML 屬性內容名稱	說明
		程式，錯誤會依處理鏈繼續進行，且 Workday 會新增一個類型為「稽核記錄」的項目。ERROR_HANDLER_SKIPPED。 若要在任何現有的列下方新增運算式，請按一下 新增，然後輸入運算式。 若要在現有的列上方新增運算式，請將游標移至該列，然後在新增 下拉式功能表上按一下 插入。 使用 運算式 分頁右側的按鈕，刪除列或重新排序。

轉換步驟

使用文字結構描述檔案將文字轉換為 XML

開始之前

建立包含 文字結構描述 步驟的程序集。取得將構成文字結構描述基礎的範例文字。

執行這項作業的原因和時機

文字結構描述 步驟可讓您使用文字結構描述將文字轉換為 XML。若要開發文字結構描述，您必須知道要轉換的資料結構。透過包含代表性資料的小型範例文件，此流程可大幅簡化。本主題假設您有這樣的範例。

您可以使用 文字架構 步驟將 XML 轉換為文字，或轉換分隔檔案的檔案，但在大多數情況下，分別使用 XTT 步驟和 csv-to-xml 步驟會更簡單。

程序

1. 在 文字結構 描述步驟的 屬性 檢視中，按一下 建立文字結構描述檔案。
2. 選取文字結構描述檔案的上層資料夾並提供檔案名稱。如果需要，請指定目標名稱空間 URI。
3. Studio 會開啟文字結構描述編輯器，並在 樹狀 目錄檢視中顯示新的文字結構描述。
4. 將範例文字剪下並貼到文字結構描述編輯器右上角的區域中，取代 initial-text。

註：若要從輸入檔案中移除終止換行符，請將此屬性內容新增至根結構描述元素：
ts:eofStrip="
"。

5. 開始說明範例文字。選取第一個欄位，按一下滑鼠右鍵並選取「建立欄位」。

註：如果您沒有要轉換的文字範例，但知道其結構，則可直接編輯文字結構描述的「來源」檢視。

6. 在「建立欄位精靈」的「身分」頁面上，提供欄位的基本說明。執行此任務時，請參考以下說明：

選項	敘述
名稱	欄位的名稱。
週期	選項包括： • 預設值：此欄位為必填欄位，且只會出現一次。 • 必填：與預設值相同。 • 選填：欄位為選填欄位，只能出現一次。

選項	敘述
	- 多個選填：欄位為選填，可出現多次。 - 多個必填：此欄位為必填欄位，可出現多次。
開始標記	規則運算式，用於指定導致產生元素的文字。請注意，符合此運算式的文字不會構成輸出值的一部分。如果您希望在輸出值中顯示相符的文字，請使用 <code>ts:format</code> 選項。
結束標記	規則運算式，指定用來標示元素結尾的文字。請注意，符合此運算式的文字不會構成輸出值的一部分。如果您希望在輸出值中顯示相符的文字，請使用 <code>ts:format</code> 選項。
截斷	指定是否要截斷固定長度欄位中的溢位資料，或拋出錯誤。預設值為 <code>true</code> 。
分隔符號	規則運算式，用於指定分隔重複欄位的字元。僅當您選取「多個選填項目」或「多個必填項目」選項時啟用。
省略	指定在文字至 XML 或 XML 至文字的轉換中，是否要省略輸出中的欄位。預設為無遺漏。

7. 在「建立欄位精靈」的「分隔符號」頁面上，說明範例文字中有多少字元可代表一個欄位：

選項	敘述
格式	識別欄位有效文字模式的規則運算式。使用此運算式來代替開始標記和結束標記，可確保輸出中顯示所有相符的文字。您也可以在處理多次出現的值時使用它，以便在序列結束時清楚顯示。
固定長度	指定固定長度欄位的長度。包含引號和逸出字元。
已報價	表示是否將字串值視為帶引號的字串。選項包括： - 預設值： - 一律： - 始終包含空白： - 選填： - 從不：
內邊距	識別固定長度欄位內任何填補文字的規則運算式。
偏好的填補	指定填補運算式可比對 1 個以上可能字串時要使用的值。
對齊	指定需要填補的固定長度欄位的對齊方式。

8. 在「建立欄位精靈」的「解釋」頁面上，說明範例文字中有多少字元可代表一個欄位：

選項	敘述
類型	欄位的資料類型。
預設值	欄位的預設值。

選項	敘述
日期格式	指定日期或日期時間的模式。支援 Java 允許的任何模式 SimpleDateFormat。
日期語言	指定解釋日期時要使用的日期語言。當日期格式使用月份名稱時，請確保以適當的語言處理名稱。預設值為 EN。
兩位數的年份	指定用於解釋 2 位數年份日期的值。文字結構描述剖析器會將在該值之前的 2 位數年份日期解釋為目前的世紀。範例：如果您輸入 18，剖析器會解釋 14 身分 2014 和 96 身分 1996。
數字格式	指定數字的模式。支援 Java 允許的任何模式 DecimalFormat。
小數點分隔符號	指定用於十進位數字的分隔符號。
分組分隔符號	指定數字的千分位分組分隔符號。

9. 對范例文字中的每個欄位重複步驟 6-9，在 樹狀 目錄檢視中新增元素。建立所有必填欄位定義後，請刪除其餘的欄位定義。Unparsed 子元素
10. 若要將所有已建立的元素合併為一個複雜類型，請選取元素並按一下滑鼠右鍵，然後選取「合併元素」。然後提供包裝元素的名稱和出現值。
11. 在包裝元素的 屬性 檢視中，從 結束標記 下拉清單中選取 /n。
12. 預設結構描述根元素名為 File。在元素 屬性 檢視的 結構描述 分頁上，提供更有意義的名稱。
13. 在最上層的「屬性」檢視中 schema 元素，則「根元素」屬性值為 tns:File。取代 File 並使用您為根元素選取的有意義的名稱。
14. 若要從產生的輸出中移除任何多餘的空格，請選取最上層的「屬性」檢視 schema 元素從 填補 下拉清單中選取 +。儲存結構描述。

概念：轉換步驟

轉換步驟可讓您執行個別作業，在訊息經過整合時重組訊息結構。Workday Studio 在 調色盤 上提供以下 轉換 步驟：

轉換步驟	函數	附註
xslt	使用 XSL 轉換 (XSLT) 樣式表轉換 XML 文件。	• 支援 XSLT 2.0。 • 與版本號不超過 2017.05 的程序集相容。
XSLT+	使用 XSL 轉換 (XSLT) 樣式表轉換 XML 文件。	• 支援 XSLT 3.0 串流功能。 • 與版本號 2017.05 之後的程序集相容。
fop	從 XML 檔案產生 PDF 和 RTF 文件。	• 呼叫 Apache 格式設定物件處理器 (FOP) 實作。 • 實作 Apache FOP XML 配置檔案，以註冊自訂字型。
text-excel	將 Excel 檔案轉換為逗號分隔值 (CSV) 文字格式的檔案。也會將 CSV 檔案轉換為 Excel 檔案。	

轉換步驟	函數	附註
textschemata	使用文字結構描述檔案，將文字轉換為 XML，或將 XML 轉換為文字。	
wrap-soap	將 XML 訊息包裝在 SOAP 信封中，以便轉寄至 SOAP 型 Web 服務。	
xml-to-java	使用 JAXB 將 XML 取消編組為 Java 物件。	• 實作 XPath 以從訊息中擷取 XML。 • 實作 JAXB 以將 XML 轉換為 Java 物件。 • 將轉換後的 Java 物件插入 MediationContext。
json-transformer	接受 JSON 文件做為輸入，並透過將值對應至 JSON 路徑來轉換文件。	
xml-to-json	將 XML 文件轉換為 JSON 格式。	
json-to-xml	將 JSON 文件轉換為 XML 格式。	
java-to-xml	使用 JAXB 將 Java 物件封送為 XML。	• 從「」中擷取 Java 物件 MediationContext。 • 實作 JAXB 以將 Java 物件轉換為 XML。 • 實作 XPath，將轉換後的 XML 插入訊息中。
xml-to-csv	將 XML 文件轉換為 CSV 格式，同時串流輸入和輸出文件，節省記憶體空間。	
csv-to-xml	將 CSV 文件轉換為 XML 格式，同時串流輸入和輸出文件，節省內存。	
字元轉換	將傳入訊息的文字內容執行字元轉換作業。	
mtable-建置器	建立並儲存 mtable 中的物件 MediationContext 屬性	• 支援以下子系元素： • csv-mtable-reader • json-mtable-reader
mtable-writer	擷取一個 mtable 物件 MediationContext 屬性 顯示某個項目的內容 mtable 輸出訊息中的 物件	• 支援以下子系元素： • csv-mtable-writer • json-mtable-writer
etv	根據 XML 文件中元素的附加屬性，轉換並驗證元素。	
xtt	根據 XML 中定義的屬性內容，將 XML 轉換為文字。	

參照：XSLT 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	xslt 中介步驟在程序集中的唯一 ID。
輸入	input	指定 xslt 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 xslt 中介步驟會將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
URL	url	xslt 中介步驟使用的 XSLT 檔案位置。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 xslt 中介步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 XSLT 步驟的條件。
使用以檔案做為依據的受管理資料	use-file-backed-managed-data	指定在達到資料限值時是否將所有輸出串流至本機磁碟。
Messages 屬性	messages-property	該 MediationContext 您打算在上配置的屬性 List 物件。
重複使用策略	reuse-strategy	指定 XSLT 轉換器重複使用策略。有 4 個選項： - 預設值：MediationContext 會自動決定您的重複使用策略。 - no-reuse：您的伺服器不會重複使用轉換器物件。 - 範本：您的伺服器會建置 Templates 物件，系統會根據該物件建立新的轉換器物件。 - 轉換器池：您的伺服器使用轉換器池。只有在轉換器物件不會快取值時，我們才建議使用此選項。範例：Saxon 會快取「」的結果 document() 來電。
安全處理	secure-processing	指定是否啟用可能危害安全性的處理。 如果設為 True，xslt 中介步驟會安全處理 XML，並對 XML 結構設定限制，以避免發生拒絕服務攻擊等狀況。 如果設為 False，則 xslt 中介步驟會忽略 XML 結構限制等安全問題。
Transformer 工廠	transformer-factory	XSLT 的類別名稱 TransformerFactory 由 xslt 中介步驟使用。 例如，「撒克遜文」的值為 net.sf.saxon.TransformerFactoryImpl

參照：XSLT+ 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	XSLT+ 中介步驟在程序集中的唯一 ID。
輸入	input	指定 XSLT+ 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 XSLT+ 中介步驟將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
URL	url	XSLT+ 中介步驟使用的 XSLT 檔案位置。
安全處理	secure-processing	指定是否啟用可能危害安全性的處理。

屬性	XML 屬性內容名稱	說明
		如果設為 True，XSLT+ 中介步驟會安全處理 XML，並對 XML 結構設定限制，以避免發生拒絕服務攻擊等狀況。 如果設為 False，XSLT+ 中介步驟會忽略 XML 結構限制等安全問題。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 XSLT+ 中介步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 XSLT+ 步驟的條件。
Messages 屬性	messages-property	該 MediationContext 您打算使用 來配置的屬性 List 物件。
結構描述驗證模式	schema-validation-mode	根據提供的結構描述，指定是否使用 Saxon XSLT 處理器驗證輸入和輸出。

結構描述 URL 分頁

屬性	XML 屬性內容名稱	說明
結構描述 URL	schema-url	XSLT+ 中介步驟用來驗證輸出文件的結構描述檔案所在位置。

參照：FOP 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集內 最後一個 中介步驟的唯一 ID。
輸入	input	指定 fop 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。

屬性	XML 屬性內容名稱	說明
		<i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 fop 中介步驟將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
作者	author	指定 author 文件中繼資料中的值
基本 URL	base-url	用於解析相對 URL 的基準 URL。此位置也用於相對字型位置 (如果沒有) <i>font-base-url</i> 已指定「」。可以是專案的相對路徑。WSAR-INF 或絕對 URL。範例： ./fop
建立日期	creation-date	指定 creation date 文件中繼資料中的值使用美國日期格式輸入值。範例：5/14/08 4:50 PM。
建立者	creator	指定 creator 文件中繼資料中的值
關鍵字	keywords	指定 keywords 文件中繼資料中的值
標題	title	指定 title 文件中繼資料中的值
使用者密碼	user-password	可讓您設定 PDF 和 RTF 文件的使用者密碼。此密碼可啟用以下文件權限：

屬性	XML 屬性內容名稱	說明
		• allow-copy-content • allow-edit-annotations • allow-edit-content • allow-print

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 匯款單 中介步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	一個條件，用於判斷是否執行 最後一個 步驟。
允許複製內容	allow-copy-content	指定在您使用存取文件時，是否可以複製和貼上文件內容。user-password。
允許編輯註解	allow-edit-annotations	指定當您使用存取文件時，是否可以在文件內編輯註解。user-password。
允許編輯內容	allow-edit-content	指定當您使用存取文件內容時，是否可以在文件中編輯內容。user-password。
允許列印	allow-print	指定在您存取文件時，是否可列印文件 user-password。
配置 Xml	configuration-xml	Apache FOP XML 配置檔案的路徑，該檔案必須位於您的本機磁碟上，而非 URL。此配置檔案是註冊自訂字型的唯一方式。
建立日期格式	creation-date-format	用於剖析 creation-date XML 屬性內容支援下列值： • full • long • medium • short
建立日期 - 地區設定的國家/地區	creation-date-locale-country	該 country 用於剖析的值 creation-date XML 屬性內容
建立日期地區設定語言	creation-date-locale-language	該 language 用於剖析的值 creation-date XML 屬性內容值必須是兩個字母的小寫 ISO-639 語言代碼。
建立時間格式	creation-time-format	用於剖析 creation-date XML 屬性內容支援下列值： • full • long

屬性	XML 屬性內容名稱	說明
		• medium • short
字型基本 URL	font-base-url	用於解析相對字型 URL 的基準 URL。範例： ./fop/fonts
連字號基本 URL	hyphenation-base-url	基準 URL，用於將相對 URL 解析為斷字模式檔案。
擁有者密碼	owner-password	可讓您設定 PDF 和 RTF 文件的擁有者密碼。此密碼可啟用以下文件權限： • allow-copy-content • allow-edit-annotations • allow-edit-content • allow-print
生產者	producer	指定 producer 文件中繼資料中的值
目標解析度	target-resolution	由位圖呈現器 (例如 TIFF 呈現器和 Apache Batik) 產生的位圖影像的輸出解析度 (DPI)。

參照：文字-Excel 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集中的 文字-Excel 中介步驟的唯一 ID。
輸入	input	指定 text-Excel 中介步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。您必須指定附件索引。 • <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。

屬性	XML 屬性內容名稱	說明
輸出	output	指定 Excel 文字 中介步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 • 附件：訊息的 MIME 附件。您必須指定附件索引。 • <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 <i>MediationContext</i>。
樣式	style	Excel 文字 中介步驟的轉換樣式。有 3 個選項： • XLSX/XLS 轉文字：將 XLSX 或 XLS 格式的 Excel 檔案轉換為 CSV 檔案。 • 文字轉 XLS：將 CSV 檔案轉換為 XLS 格式的 Excel 檔案。 • 文字轉 XLSX：將 CSV 檔案轉換為 XLSX 格式的 Excel 檔案。 註：此元件會在轉換為 XLSX 或從 XLSX 轉換時串流。
Excel 樣式	excel-style	指定 文字-Excel 中介步驟是轉換整個 Excel 工作簿還是單一 Excel 工作表。
Excel 工作表編號	excel-sheet-number	要在 Excel 檔案中轉換的目標工作表編號。 如果您將「」工作簿的「Excel 樣式」屬性配置為，Workday 會忽略 文字-Excel 中介步驟的 Excel 工作表編號 設定。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 Excel 文字 步驟說明。您可以使用此功能來

屬性	XML 屬性內容名稱	說明
		說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 Excel 文字 步驟步驟的條件。
使用值	usevalues	指定 Excel 文字 中介步驟是否使用試算表的基礎值，而不是螢幕上顯示的已設定格式的值。

relatedlinks

相關參考資料

[Workday 31 功能更新公告：Workday Studio](#)

參照：文字結構描述步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	文字架構 中介步驟在程序集中的唯一 ID。
輸入	input	指定 文字 結構描述中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 文字 結構描述中介步驟會將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。

屬性	XML 屬性內容名稱	說明
		- 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。
URL	<code>url</code>	文字結構 描述中介步驟使用的文字結構描述 XSD 檔案的位置，以相對路徑或絕對 URL 形式提供。您可以使用 MVEL 運算式，從 <code>MediationContext</code> 。範例： <code>@{parts[1].xpath('/script/location')}</code>
樣式	<code>style</code>	文字結構 描述中介步驟的轉換樣式。有 2 個選項： <code>text2xml</code>：將文字轉換為 XML。 <code>xml2text</code>：將 XML 轉換為文字。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 文字 結構描述中介步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<code>execute-when</code>	決定 文字結構 描述步驟是否執行的條件。
使用以檔案做為依據的受管理資料	<code>use-file-backed-managed-data</code>	指定在達到資料限值時是否將所有輸出串流至本機磁碟。

參照：自動換行步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	<code>id</code>	程序集中的 <code>lap-soap</code> 中介步驟的唯一 ID。
輸入	<code>input</code>	指定 <code>Wrap-soap</code> 中介步驟從何處取得資料做為輸入。有 5 個可能的值：

屬性	XML 屬性內容名稱	說明
		<i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 lap-soap 中介步驟將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 lap-soap 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	一個條件，用於判斷是否執行 lap-soap 步驟。
介面	interface	SOAP Body 元素的直屬子系。
SOAP 動作	soap-action	SOAP 動作標頭的值。您必須為 SOAP HTTP 要求指定此值。
SOAP 版本	soap-version	SOAP 信封的版本。

參照：XML 轉 Java 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	組合內 xml-to-java 中介步驟的唯一 ID。
輸入	input	指定 xml-to-java 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
屬性	property	的名稱 <i>MediationContext</i> xml-to-java 中介步驟會將 Java 物件插入其中的屬性。
方案	packages	xml-to-java 中介步驟使用 JAXB 將 XML 轉換為 Java 物件時所需的 Java 封裝清單 (以分號分隔)。
展開 JAXB 元素	unwrap-jaxbelement	指定 xml-to-java 中介步驟是否將 Java 物件儲存在 JAXB 元素內。
名稱空間	namespaces	XPath 運算式使用的名稱空間。
XPath	xpath	從訊息中擷取 XML 以將其轉換為 Java 物件的 XPath 運算式。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 xml-to-java 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 xml-to-java 步驟的條件。

屬性	XML 屬性內容名稱	說明
XPath 版本	xpath-version	xml-to-java 中介步驟使用的 XPath 版本。範例：版本 1.0、版本 2.0 或版本 3.0。

參照：JSON 轉換器屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集內 json-transformer 中介步驟的唯一 ID。
輸入	input	指定 json-transformer 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 MediationContext。
輸出	output	指定 json-transformer 中介步驟將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。

屬性	XML 屬性內容名稱	說明
		變數：將輸出另存為具名變數 <code>MediationContext</code>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 <code>json-transformer</code> 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<code>execute-when</code>	決定 <code>json-transformer</code> 步驟是否執行的條件。

對應分頁

屬性	XML 屬性內容名稱	說明
Json 路徑	<code>json-path</code>	指定 JSON 樹狀目錄中指派相關聯值的 位置。
未通過 - 無相符項目	<code>fail-no-match</code>	決定當指定的 JSON 路徑未傳回節點時會發生什麼： <code>true</code>：引發錯誤。 <code>fail</code>：忽略缺少的路徑。

參照：XML 轉 JSON 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	<code>id</code>	<code>xml 轉 json</code> 中介步驟在程序集中的唯一 ID。
輸入	<code>input</code>	指定 <code>xml-to-json</code> 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <code>rootpart</code>：MIME 訊息的根部分的内容。如果訊息不是 MIME，則會選取整封訊息。

屬性	XML 屬性內容名稱	說明
		變數：使用中的具名變數 <code>MediationContext</code>。
輸出	<code>output</code>	指定 <code>xml-to-json</code> 中介步驟將輸出導向何處。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。此值會建立新訊息，有效地移除所有附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <code>rootpart</code>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。
結構描述 URL	<code>schema-url</code>	指定相對 URL，以便 <code>xml-to-json</code> 中介步驟擷取 XML 結構描述檔案，做為程序集 Web 服務專案內的資源。
結構描述	<code>schema</code>	指定 <code>xml</code> 轉 <code>json</code> 步驟取得 XML 結構描述的位置。有 5 個可能的值： - 訊息 <code>soapbody</code> <code>attachment</code> <code>rootpart</code> - 變數
忽略屬性內容	<code>ignore-attributes</code>	指定將 XML 文件轉換為 JSON 時， <code>xml-to-json</code> 步驟是否忽略屬性內容。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 <code>xml</code> 轉 <code>json</code> 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<code>execute-when</code>	決定是否執行 <code>xml-to-json</code> 步驟的條件。

參照：JSON 轉 XML 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	json 轉 xml 中介步驟在程序集中的唯一 ID。
輸入	input	指定 json-to-xml 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 json 轉 xml 中介步驟會將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
根元素名稱	root-element-name	XML 文件最外層的換行標記名稱。
內嵌陣列名稱	nested-array-name	JSON 輸入中匿名內嵌陣列的名稱。

屬性	XML 屬性內容名稱	說明
內嵌物件名稱	nested-object-name	JSON 輸入中匿名內嵌物件的名稱。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 json-to-xml 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定 json-to-xml 步驟是否執行的條件。

參照：Java 轉 XML 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集內 java-to-xml 中介步驟的唯一 ID。
輸出	output	指定 java-to-xml 中介步驟將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。請指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
屬性	property	的名稱 <i>MediationContext</i> java-to-xml 中介步驟從中擷取 Java 物件的屬性。
方案	packages	java-to-xml 中介步驟使用 JAXB 將 Java 物件轉換為 XML 時所需的 Java 封裝清單 (以分號分隔)。
根元素名稱	root-element-name	用於將 Java 物件封送為 XML 的根元素名稱。

屬性	XML 屬性內容名稱	說明
根元素命名空間	root-element-namespace	用於將 Java 物件封送為 XML 的根元素的名稱空間。
名稱空間	namespaces	XPath 運算式使用的名稱空間。
XPath	xpath	將 XML 插入訊息的 XPath 運算式。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 java-to-xml 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定 java-to-xml 步驟是否執行的條件。
XPath 版本	xpath-version	指定是否使用 XPath 版本 1.0、2.0 或 3.0。對於從 Workday 11 開始的組件版本，請輸入值 1。請針對 Workday 12 起的組件版本輸入 2 或 3。 您可以讓 Workday 使用可插入系統屬性值的 MVEL 範本動態設定版本。範例： @{props.get('the_version')}。}

參照：XML 轉 CSV 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	xml 轉 csv 步驟在程序集中的唯一 ID。
輸入	input	指定 xml 轉 csv 步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。

屬性	XML 屬性內容名稱	說明
		變數：使用中的具名變數 <code>MediationContext</code>。
輸出	<code>output</code>	指定 xml 轉 csv 步驟會將輸出導向何處。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。此值會建立新訊息，有效地移除所有附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息的 MIME 附件。請指定附件索引。 <code>rootpart</code>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。
分隔符號	<code>separator</code>	要在 CSV 輸出檔案中每個資料輸入之後插入的分隔符號。
行分隔符號	<code>line-separator</code>	要在 CSV 輸出檔案中每個資料輸入之後插入的新行分隔符號。支援下列值： - LF 適用於 Linux 和 Mac OS X 及更高版本。 - CR 適用於 Mac OS 版本 9 或更低版本。 - CRLF 適用於 Microsoft Windows。
寫入標頭行	<code>writeHeaderLine</code>	指定 xml 轉 csv 步驟是否將 XML 文件中連續子元素的名稱以逗號分隔值的形式插入輸出文件的第一行。
格式	<code>format</code>	指定您的 CSV 輸出文件是否符合 RF-4180 規範。您可以輸入下列任一值： <code>rfc4180</code> - 簡單 如果輸入 <code>rfc4180</code>，則 CSV 輸出文件將符合 RF-4180 標準。 如果您輸入「簡單」或將此欄位保留空白，則 xml 轉 csv 步驟會： - 將單引號視為替代引號。 - 修剪值中的初始空白。

屬性	XML 屬性內容名稱	說明
		接受任何明細結尾 (CR、LF 或 CRLF)。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 xml 轉 csv 步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 xml 轉 csv 步驟的條件。

參照：CSV 轉 XML 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	CSV 轉 xml 步驟在程序集中的唯一 ID。
輸入	input	指定 csv 轉 xml 步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 MediationContext。
輸出	output	指定 csv 轉 xml 步驟會將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。 - 附件：訊息的 MIME 附件。請指定附件索引。

屬性	XML 屬性內容名稱	說明
		<i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
欄名稱	<code>colNames</code>	XML 輸出檔案中顯示的欄名稱。請以逗號分隔的名稱清單輸入此屬性的值。範例： <code>col1, col2, col3</code> 。
命名空間字首	<code>namespacePrefix</code>	XML 輸出檔案中顯示的名稱空間字首。
名稱空間 URI	<code>namespaceURI</code>	顯示在 XML 輸出檔案中根元素上的名稱空間 URI。 範例：如果您指定的 URI 為 <code>http://example.workday.com/SupplyChainManagement/DataTypes.xsd</code> 和「」名稱空間字首 <code>ns</code> ， <code>csv</code> 轉 <code>xml</code> 步驟會在您的 XML 輸出檔案中建立此根元素： <pre><ns:root xmlns:ns="http:// example.workday.com/ SupplyChainManagement/ DataTypes.xsd"></pre>
根名稱	<code>rootName</code>	XML 輸出檔案中顯示的根元素名稱。
列名稱	<code>rowName</code>	XML 輸出檔案中顯示的列名。請以逗號分隔的名稱清單輸入此屬性的值。範例： <code>row1, row2, row3</code> 。
分隔符號	<code>separator</code>	CSV 輸入檔案中使用的分隔符號。
格式	<code>format</code>	指定您的 CSV 輸入檔案是否符合 RF-4180 規範。您可以輸入下列任一值： <i>rfc4180</i> - 簡單 如果您輸入 <i>rfc4180</i> ， <code>csv-to-xml</code> 步驟會假設您的 CSV 輸入檔案符合 RF-4180 標準。

屬性	XML 屬性內容名稱	說明
		如果您輸入「簡單」或將欄位保留空白，則 csv-to-xml 步驟會： - 將單引號視為替代引號。 - 修剪值中的初始空白。 - 接受任何明細結尾 (CR、LF 或 CRLF)。
使用第一行做為標頭	useFirstLineAsHeader	指定 csv 轉 xml 步驟是否使用 CSV 輸入檔案的第一行來命名 XML 輸出檔案中顯示的欄。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 csv 轉 xml 步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 csv-to-xml 步驟的條件。

參照：字元轉換步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	字元轉換 中介步驟在程序集中的唯一 ID。
輸入	input	指定 字元轉換 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 MediationContext。
輸出	output	指定 字元轉換 中介步驟會將輸出導向何處。有 5 個可能的值：

屬性	XML 屬性內容名稱	說明
		<i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
移除強調色	<code>remove-accents</code>	指定 字元轉換 步驟是否會將輸出訊息中的重音字元取代為無重音字元。 範例：如果設為 <code>true</code> ，字元轉換 步驟會將 <code>é</code> 取代為 <code>e</code> 。
移除 Eols	<code>remove-eols</code>	指定 字元轉換 步驟是否移除輸出訊息中的行尾字元。
大寫	<code>uppercase</code>	指定 字元轉換 步驟是否將輸出訊息中的所有字元轉換為大寫。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 字元轉換 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<code>execute-when</code>	決定是否執行 字元轉換 步驟的條件。
取代控制	<code>replace-controls</code>	指定 字元轉換 步驟是否會在輸出訊息中將 ASCII 控制字元取代為空格。

參照：Mtable-建置器步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	<code>id</code>	<i>mtable</i> -建置器 中介步驟在程序集中的唯一 ID。

屬性	XML 屬性內容名稱	說明
輸入	input	指定 mtable-建置器 中介步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。您必須指定附件索引。 • <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。
清單名稱	mtable-name	的名稱 MediationContext 的屬性 mtable 物件。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 mtable-建置器 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 mtable-建置器 步驟的條件。

參照：Mtable-Writer 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	mtable-Writer 中介步驟在程序集中的唯一 ID。
輸出	output	指定 mtable-Writer 中介步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。

屬性	XML 屬性內容名稱	說明
		- 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
清單名稱	<i>mtable-name</i>	的名稱 <i>MediationContext</i> 的屬性 <i>mtable</i> 物件。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<i>description</i>	顯示在合併日誌中的 <i>mtable-Writer</i> 中介步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<i>execute-when</i>	決定是否執行 <i>mtable-Writer</i> 步驟的條件。

參照：ETV 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	<i>id</i>	程序集內 <i>Etv</i> 中介步驟的唯一 ID。
輸入	<i>input</i>	指定 <i>etv</i> 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。

屬性	XML 屬性內容名稱	說明
輸出	output	指定 etv 中介步驟將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 etv 中介步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	一種條件，用於判斷是否執行「etv」步驟。
附加訊息	append-messages	指定新訊息是取代現有訊息，還是附加到現有訊息。
訊息屬性	message-property	的名稱 <i>MediationContext</i> 儲存由 etv 中介步驟建立的訊息的屬性。

參照：XTT 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	xtt 中介步驟在程序集中的唯一 ID。
輸入	input	指定 xtt 中介步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。

屬性	XML 屬性內容名稱	說明
		• <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。您必須指定附件索引。 • <i>rootpart</i>：MIME 訊息的根部分的內容。如果訊息不是 MIME，則會選取整封訊息。 • 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 xtt 中介步驟會將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 • 附件：訊息的 MIME 附件。您必須指定附件索引。 • <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 <i>MediationContext</i>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 xtt 中介步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 xtt 步驟的條件。
附加訊息	append-messages	指定新訊息是取代現有訊息，還是附加到現有訊息。
訊息屬性	message-property	的名稱 <i>MediationContext</i> 儲存由 xtt 中介步驟建立的訊息的屬性。

Mtable-建置器子元素

概念：Mtable-建置器子系元素

mtable-建置器 步驟支援下列子系元素：

子元素	說明
csv-mtable-reader	將 CSV 內容剖析為 mtable 物件。
json-mtable-reader	將 JSON 內容剖析為 mtable 物件。

參照：CSV-Mtable-Reader 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
欄名稱	column-names	以逗號分隔的欄名稱清單，顯示在您的資料欄位上。mtable 物件。
包含標頭列	contains-header-row	指出您的 mtable 物件包含標頭列。
索引欄	index-column	上顯示的索引欄名稱 mtable 物件。
將欄名稱標準化	normalise-column-names	表示您的「」中的欄名稱 mtable 以空格顯示的物件。 如果設為 true，您的表上的欄名稱 mtable 以空格顯示的物件。 如果設為 false，您的表上的欄名稱 mtable 物件顯示時不含空格。
分隔符號	separator	中用於分隔欄位的分隔符號 mtable 物件。
使用標頭列	use-header-row	指出您的 mtable 物件使用標頭列來顯示欄名稱。

參照：JSON-Mtable-Reader 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
剖析目標	parse-target	您要剖析為「」的根 JSON 物件或陣列的名稱。mtable 物件。
忽略 ID 欄	ignore-id-column	指定是否要將 JSON 輸入資料中的 ID 欄顯示在 mtable 物件。

屬性	XML 屬性內容名稱	說明
		如果設為 true，Workday 會忽略您資料夾中 JSON 輸入資料中的 ID 欄。mtable 物件。 如果設為 false，Workday 會在您的「」中顯示 JSON 輸入資料中的 ID 欄 mtable 物件。

Mtable-Writer 子元素

概念：Mtable-Writer 子元素

mtable-Writer 步驟支援以下子系元素：

子元素	說明
csv-mtable-writer	顯示某個項目的內容 mtable CSV 格式的物件
json-mtable-writer	顯示某個項目的內容 mtable CSV 格式的物件

參照：CSV-Mtable-Writer 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
欄名稱	column-names	以逗號分隔的欄名稱清單，顯示在您的資料欄位上。mtable 物件。
行分隔符號	line-separator	您中的新行分隔符號 mtable 物件。支援下列值： - LF 適用於 Linux 和 Mac OS X。 - CR 適用於 Mac OS 版本 9 或更低版本。 - CRLF 適用於 Microsoft Windows。
分隔符號	separator	中用於分隔欄位的分隔符號 mtable 物件。
寫入欄名稱	write-column-names	指出您的 mtable 物件使用標頭列，在 MediationContext。

參照：JSON-Mtable-Writer 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
欄位名稱	field-name	JSON 欄位的名稱，此欄位會顯示 mtable 物件。

屬性	XML 屬性內容名稱	說明
將 Mtable 寫入為物件	write-mtable-as-object	表示單列 Mtable 物件是否顯示為： - JSON 陣列包含 1 個物件元素和多個欄位 (True)。 - 具有多個欄位的 JSON 物件 (False)。 預設設定為 False。

記錄和事件處理步驟

概念：記錄和事件處理步驟

記錄步驟可讓您建立自訂的記錄訊息，以便輸出至不同的目的地。Workday Studio 在 調色盤上提供以下 記錄和事件處理 步驟：

名稱	函數	附註
記錄	建立寫入 Workday 的自訂 Log4j 訊息。	插入 MVEL 運算式時按 Ctrl+ 空格鍵可存取內容輔助。
cloud-log	可讓您將整合流程的自訂訊息輸出至 HTML 或 CSV，傳送至您選擇的端點。	「雲端日誌」步驟會產生輸出檔案，但不會將其導向任何地方。請使用另一個步驟來處理檔案的儲存或轉移。範例：使用「儲存」步驟，將輸出檔案傳送給您的用戶。 - 雲端日誌 步驟產生的輸出檔案一律包含 5 欄： - 時間戳記 - 層級 - 參考 ID - 訊息摘要 - 訊息詳細資料 使用「額外欄」分頁來新增額外資訊。 輸出檔案的大小上限為 256 MB。如果輸出超過 256 MB，系統會通知您結果已截斷。

參照：記錄步驟屬性

訊息建置器分頁

訊息建置器 簡化了某些組合步驟的訊息內容定義。

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	記錄 步驟在程序集中的唯一 ID。
輸入	input	指定 記錄 步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
層級	level	Log4J 記錄層級。有 5 個選項： - 偵錯 - info - 警告 - 錯誤 - 嚴重

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 記錄 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 記錄 步驟的條件。
種類	category	記錄器名稱的一部分。若未指定值，Workday 會使用 DEFAULT。

您可以使用 `static-file` 子元素，以指定輸入檔案的字元編碼屬性。範例：

```
<cc:log-message>
  <cc:static-file character-encoding="UTF-16" input-
file="logfile.html"/>
</cc:log-message>
```

如果您未指定 character-encoding 值時，Workday 會使用中指定的字元集 output-mimetype 屬性如果此處也未指定字元集，Workday 會使用 UTF-8。

參照：雲端記錄步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集內 雲端記錄 步驟的唯一 ID。
變數名稱	variable-name	儲存輸出檔案的變數名稱。您可以為多個 雲端記錄 步驟指定相同的變數名稱，將其訊息串連成單一報告。您可以使用不同的變數名稱來產生個別的輸出檔案。
層級	level	記錄訊息的嚴重程度層級。有 6 個選項： - 偵錯 - info - 警告 - 錯誤 - 嚴重 - custom 如果您選取「自訂」，您必須在相鄰欄位中使用 MVEL 運算式來指定您的自訂屬性。
標頭	header	雲端日誌檔案標頭的文字。如果您將此欄位保留空白，則 Studio 會使用整合名稱做為標頭文字。 如果您的程序集有多個已定義標頭的 雲端記錄 步驟，則 Studio 會使用流程中第一個標頭的標頭。
訊息	message	記錄訊息的文字。
訊息詳細資料	message-details	記錄訊息的說明。
參考 ID	reference-id	Workday 物件的參考 ID。可用於疑難排解。
條件	condition	MVEL 布林運算式，指定 Workday 在何種環境下記錄訊息。 若未指定條件，Workday 一律會記錄訊息。
輸出檔案類型	output-file-type	雲端記錄 步驟產生的檔案類型。有 3 個選項： - HTML - CSV - XLSX

屬性	XML 屬性內容名稱	說明
		輸出檔案的大小上限為 256 MB。如果輸出超過 256 MB，Studio 會通知您並截斷結果。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	記錄訊息的說明。
執行時間	execute-when	決定是否執行 雲端記錄 步驟的條件。

額外欄分頁

屬性	XML 屬性內容名稱	說明
索引鍵	key	其他自訂訊息的 ID。在 1 個記錄變數中不重複。
標籤	label	其他自訂訊息的面向使用者的標籤。
值	value	其他自訂訊息的文字。

relatedlinks

相關參考資料

[Workday 31 功能更新公告：Workday Studio](#)

訊息處理步驟

概念：訊息處理步驟

訊息處理步驟可讓您在訊息經過整合時自訂其內容。Workday Studio 在 調色盤上提供以下 訊息處理 步驟：

名稱	函數	附註
複製	擷取傳入訊息的個別部分，並將其複製到傳出訊息的不同部分。	
JavaScript	實作 JavaScript 指令碼或函數，以操縱傳入的訊息。	
set-headers	操作中介訊息的標頭。	
寫入	可讓您建構自訂資料，並將其設為中介訊息的內容。	插入 MVEL 運算式時按 Ctrl+空格鍵可存取內容輔助。

參照：複製步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	複製 步驟在程序集中的唯一 ID。
輸入	input	指定 複製 步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸入 XPath	input-xpath	一個 XPath 運算式，用於從 中的訊息部分或變數擷取 XML 輸入資料的相符區段。 <i>MediationContext</i> 。
輸出	output	指定 複製 中介步驟會將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。您必須指定附件索引。 <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
輸出 XPath	output-xpath	XPath 運算式，用於取代「」中的訊息部分或變數中

屬性	XML 屬性內容名稱	說明
		的 XML 輸出資料的相符區段 MediationContext。
附加至輸出元素	append-to-output-element	指定 複製 步驟是否要將輸入 XML 元素附加至與 output-xpath 值。
名稱空間	namespaces	XPath 運算式使用的名稱空間，格式為 prefix namespace prefix2 namespace2。覆寫 component.xml 檔案中的名稱空間對應。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 複製 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 複製 步驟的條件。
串流 XPath	stream-xpath	指定 複製 步驟是否提供對簡單 XPath 運算式的串流支援。
XPath 版本	xpath-version	指定使用 XPath 1.0 版還是 2.0 版。對於 Workday 11 到的程序集版本，請輸入值 1，對於 Workday 12 之後的程序集版本，請輸入值 2。如果「Stream-xPath」的值設為 True，Workday 會忽略此屬性。 您可以讓 Workday 使用可插入系統屬性值的 MVEL 範本動態設定版本。範例： <code>@{props.get('the_version')}</code> 。}

參照：JavaScript 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	JavaScript 步驟在程序集中的唯一 ID。
輸入	input	指定 JavaScript 步驟從何處取得資料做為輸入。有 5 個可能的值： <code>message</code>：整封訊息，包括 rootpart 和附件。

屬性	XML 屬性內容名稱	說明
		• <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	<code>output</code>	指定 JavaScript 中介步驟會將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 • 附件：訊息的 MIME 附件。您必須指定附件索引。 • <i>rootpart</i>：MIME 訊息的根部分的内容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 <i>MediationContext</i>。
函數	<code>function</code>	包含 JavaScript 函數的檔案名稱，該函數必須位於專案的 <code>ws/WSAR-INF</code> 目錄中。 請為「函數」或「指令碼」屬性指定值，請勿同時指定兩者。您無法部署同時配置這兩個值的 JavaScript 步驟。
指令碼	<code>script</code>	包含 JavaScript 指令碼的檔案名稱，該指令碼必須位於專案的 <code>ws/WSAR-INF</code> 目錄中。 請為「函數」或「指令碼」屬性指定值，請勿同時指定兩者。您無法部署同時配置這兩個值的 JavaScript 步驟。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 JavaScript 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定 JavaScript 步驟是否執行的條件。

參照：設定標頭步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	集合標頭 步驟在程序集中的唯一 ID。
輸出	output	指定 set-headers 中介步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 • 附件：訊息的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 <i>MediationContext</i>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	set-headers 步驟的說明。顯示在合併記錄中。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行「set-headers」步驟的條件。
全部清除	clear-all	清除中介訊息的所有標頭。
全部清除 - 排除	clear-all-exclude	當「全部清除」設為「」時不會刪除的標頭名稱清單 <i>true</i> 。

標頭分頁

新增標頭

屬性	XML 屬性內容名稱	說明
名稱	name	您要新增至中介訊息的標頭名稱。
值	value	您要新增至中介訊息的標頭值。

移除標頭

屬性	XML 屬性內容名稱	說明
名稱	name	您要從中介訊息中移除的標頭名稱。

參照：寫入步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	寫入 步驟在程序集中的唯一 ID。
輸入	input	指定 寫入 步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 寫入 中介步驟會將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息的 MIME 附件。請指定附件索引。

屬性	XML 屬性內容名稱	說明
		<code>rootpart</code>：MIME 訊息的根部分的內容，如果不是 MIME 則為訊息本身。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 寫入 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<code>execute-when</code>	決定 寫入 步驟是否執行的條件。

驗證步驟

概念：驗證步驟

驗證中介步驟可讓您在訊息經過整合時對其進行評估。Workday Studio 在 調色盤上提供以下 驗證 步驟：

名稱	函數	附註
<code>eval</code>	可讓您直接評估運算式。	
驗證	驗證中介訊息，確保其符合 XML 結構描述或文件類型定義 (DTD)，或是包含格式正確的 XML。	- 驗證失敗時，取決於「篩選 條件」屬性的值，系統會產生錯誤或捨棄訊息。 - 您可以指定自訂錯誤訊息。 - 您可以提供僅用於驗證所選元素的結構描述或 DTD，並藉由提供 XPath 運算式，從中介訊息中選取這些元素。
<code>validate-exp</code>	使用一或多個 MVEL 運算式驗證傳入訊息。	MVEL 運算式定義為子元素序列。
<code>validate-xpath</code>	使用以布林值評估的 XPath 運算式驗證訊息。	如果傳回的 XPath 運算式不為空白，則視為驗證成功。

參照：評估步驟屬性

運算式分頁

屬性	XML 屬性內容名稱	說明
運算式	expression	可讓您直接輸入 MVEL 條件運算式。 若要在任何現有的列下方新增運算式，請按一下 新增，然後輸入運算式。 若要在現有的列上方新增運算式，請將游標移至該列，然後在新增 下拉式功能表上按一下 插入。 使用 運算式 分頁右側的按鈕，刪除列或重新排序。

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	組合體中 評估 步驟的唯一 ID。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 評估 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 評估 步驟的條件。
語言	language	指定您要使用的運算式語言。中英文文化語言是目前唯一支援的語言。

參照：驗證步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	驗證 步驟在程序集中的唯一 ID。
輸入	input	指定 驗證 步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊

屬性	XML 屬性內容名稱	說明
		息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息。如果訊息不是 MIME，則 Studio 會選取整封訊息。 • 變數：使用中的具名變數 <i>MediationContext</i>。
結構描述	schema	此步驟用來驗證訊息的結構描述檔案所在位置。如果您指定地點 (而非 MVEL 範本)，Workday 會使用匯總驗證程式。
名稱空間	namespaces	XPath 運算式使用的名稱空間，格式為 <i>prefix namespace prefix2 namespace2</i> 。覆寫 <i>component.xml</i> 檔案中的名稱空間對應。
XPath	xpath	選取元素進行驗證的 XPath 運算式。
失敗訊息	failure-message	自訂錯誤訊息，發生驗證錯誤時會覆寫預設錯誤訊息。只有當您將「篩選條件」屬性設為 <i>False</i> 時，才會顯示錯誤訊息。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 驗證 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 驗證 步驟的條件。
模式	mode	執行的驗證類型。選項包括： • 結構描述：依據 W3C XML 結構描述進行驗證。 • dtd：依據文件類型定義進行驗證。 • 格式正確：驗證 XML 的格式是否正確。
DTD	dtd	此步驟用來驗證訊息的 DTD 檔案位置。

屬性	XML 屬性內容名稱	說明
取代訊息	replace-message	表示驗證失敗時，是否要使用自訂錯誤訊息集做為「失敗訊息」屬性。
XPath 版本	xpath-version	指定使用 XPath 1.0 版還是 2.0 版。對於 Workday 11 到的程序集版本，請輸入值 1，對於 Workday 12 之後的程序集版本，請輸入值 2。如果「Stream-xPath」的值設為 True，Workday 會忽略此屬性。 您可以讓 Workday 使用可插入系統屬性值的 MVEL 範本動態設定版本。範例： <code>@{props.get('the_version')}</code> 。
篩選	filter	指定是否在驗證失敗時中斷中介訊息處理。預設值為 False，表示 Studio 會將驗證錯誤傳遞至元件內定義的錯誤處理程式。 若設為 True，則 Studio 會執行元件的回應步驟，但會捨棄訊息且不再呼叫任何步驟。這不會引發錯誤。

參照：驗證計算步驟屬性

運算式分頁

屬性	XML 屬性內容名稱	說明
值	expression	可讓您直接輸入 MVEL 條件運算式。 若要在任何現有的列下方新增運算式，請按一下 新增，然後輸入運算式。 若要在現有的列上方新增運算式，請將游標移至該列，然後在新增 下拉式功能表上按一下 插入。 使用 運算式 分頁右側的按鈕，刪除列或重新排序。
失敗訊息	failure-message	驗證運算式失敗時要新增至例外訊息的文字。
錯誤編號	error-number	驗證運算式失敗時要加到傳回的例外訊息中的錯誤編號。
取代	replace	表示是否要將 Workday 傳回的失敗訊息取代為「失敗訊息」屬性中指定的文字。

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集中的 validate-exp 步驟的唯一 ID。
篩選	filter	指定是否在驗證失敗時中斷中介訊息處理。預設值為 False，表示 Workday 會將驗證錯誤傳遞至元件內定義的錯誤處理程式。 若設為 True，Workday 會執行元件的回應步驟，但會捨棄訊息，且不再叫用任何步驟。這不會引發錯誤。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 validate-exp 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 validate-exp 步驟的條件。

參照：驗證 XPath 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	validate-xPath 步驟在程序集中的唯一 ID。
輸入	input	指定 validate-xPath 步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 rootpart 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 - 變數：使用中的具名變數 MediationContext。

屬性	XML 屬性內容名稱	說明
篩選	filter	指定是否在驗證失敗時中斷中介訊息處理。預設值為 False，表示 Workday 會將驗證錯誤傳遞至元件內定義的錯誤處理程式。 若設為 True，Workday 會執行元件的回應步驟，但會捨棄訊息，且不再叫用任何步驟。這不會引發錯誤。
名稱空間	namespaces	XPath 運算式使用的名稱空間，格式為 prefix namespace prefix2 namespace2。覆寫 component.xml 檔案中的名稱空間對應。
XPath	xpath	選取元素進行驗證的 XPath 運算式。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 validate-xPath 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 validate-xPath 步驟的條件。
XPath 版本	xpath-version	指定使用 XPath 1.0 版還是 2.0 版。對於 Workday 11 到的程序集版本，請輸入值 1，對於 Workday 12 之後的程序集版本，請輸入值 2。如果「Stream-xPath」的值設為 True，Workday 會忽略此屬性。 您可以讓 Workday 使用可插入系統屬性值的 MVEL 範本動態設定版本。範例： <code>@{props.get('the_version')}</code> 。

編碼步驟

概念：編碼步驟

編碼步驟可讓您對「」的內容進行編碼、解碼、壓縮和解壓縮。MediationContext。

Workday Studio 在 調色盤上提供以下 編碼 步驟：

名稱	函數	附註
pgp-encrypt	使用 PGP 擷取公開金鑰並加密傳出資料。	- 若要使用儲存在 Workday 中的金鑰加密訊息，請為您的整合配置發送服務。Studio 整合無法存取 Workday 中儲存的金鑰。 - 您也可以使用 file:// 通訊協定來參照本機 PGP 金鑰，然後將其與整合一起部署。但是，交付服務方法是 Studio 整合的最佳做法。
pgp-decrypt	擷取私人金鑰，並對使用 PGP 加密的傳入資料進行解密。	
base64-decode	將附件資料的 Base64 編碼解碼為二進位格式。	
base64-encode	以 Base64 編碼格式編碼二進位附件資料。	
zip	將根部分和附件從 MediationContext 壓縮成 ZIP 檔案。	
unzip	擷取 ZIP 檔案的內容。	如果 ZIP 包含超過 1 個檔案，則首選中介訊息之類的多部分結構。例如，如果您使用「訊息」的預設輸出選項，則 ZIP 中的第一個檔案會成為根部分，而 ZIP 中的其他檔案會成為附件。
壓縮	轉換中的單一檔案 MediationContext 轉換為 GZIP 或 BZIP2 檔案。	與 ZIP 步驟不同，此步驟不支援檔案封存。使用它來壓縮單一檔案。
解壓縮	擷取單一 GZIP 或 BZIP2 檔案的內容。	

概念：PGP 檔案名稱

做為一項安全措施，當明文檔案使用 PGP 加密時，加密檔案的檔名可能與原始檔案無關，也不會提示其內容。但是，不會捨棄原始檔案名稱。內嵌在加密檔案中，供收件者的解密系統稍後擷取。例如：您加密了名為 myPlaintext.txt 的檔案。產生的檔案名為 myEncryption.pgp。其加密內容的一部分是原始檔案名稱。當您的收件者解密檔案時，其系統會將產生的檔案命名為 myPlaintext.txt。

Workday 發送與擷取服務

Workday 發送和擷取服務會盡可能遵循典型的 PGP 檔案命名慣例。

Workday Delivery Service 會在傳出的 PGP 檔案中嵌入純文字檔名，以便在解密期間將其複原。

當 Workday 擷取服務解密附加至整合事件的傳入 PGP 檔案時，它會使用內嵌的純文字檔名來命名解密的檔案。

不過，有時傳入的 PGP 檔案並沒有內嵌的純文字檔名。如果是，Workday 擷取服務會根據 PGP 檔案的名稱來做解密檔案的名稱：

- 如果 PGP 檔案名稱以 .pgp 結尾，擷取服務會從解密的檔案名稱中移除該後綴。範例：PGP 檔案 myFile.csv.pgp 解密為 myFile.csv，PGP 檔案 myFile.pgp 解密為 myFile。
- 如果 PGP 檔案名稱以 .xlsx 結尾，擷取服務會保持不變。例如，PGP 檔案 myFile.xlsx 被解密為 myFile.xlsx。
- 在所有其他情況下，擷取服務會加上 .plaintext 做為後綴。範例：PGP 檔案 myFile.doc 解密為 myFile.plaintext。

Workday Studio

當您使用 Studio 的 pgp-加密 步驟加密訊息時，可以使用 解密檔名 步驟屬性，指定明文檔名。然後，Studio 會將該檔案名稱嵌入加密檔案中，以供後續以正常方式擷取。

同樣地，Studio 的 pgp-decrypt 步驟會使用任何內嵌的純文字檔名來命名已解密的檔案。您可以使用 解密的檔案名稱 步驟屬性，為未內嵌任何檔案名稱的情況提供檔案名稱。

疑難排解

當遠端加密檔案的當事人無意中將內嵌的純文字檔名設定為收件者系統要求的值以外的值時，就會發生問題。由於 PGP 檔案名稱與純文字檔案名稱完全無關，因此無法推導出後者。收件者沒有此情境中的選項。

範例：擷取服務配置為擷取及解密檔案。執行時，它會收集名為 ExternalStudents.pgp 的 PGP 加密檔案，但名為 _CONSOLE 的檔案會附加至整合事件，以供整合處理。在這種情況下，遠端系統已將 _CONSOLE 設為內嵌純文字檔名。結果是 Workday 整合系統將檔案識別為 _CONSOLE。

避免此類問題的唯一方法是確保 PGP 檔案的傳送者了解如何正確內嵌所需的純文字檔名。

參照：PGP 加密步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集內 pgp-加密 步驟的唯一 ID。
輸入	input	指定 pgp-加密 步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。
輸出	output	指定 pgp-加密 步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。此值

屬性	XML 屬性內容名稱	說明
		會建立新訊息，有效地移除所有附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <code>rootpart</code>：的內容 <code>rootpart</code> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。
ASCII Armored	<code>ascii-armored</code>	指出 pgp-加密 步驟是以 ASCII 保護格式 (Base64 的 PGP 版本) 還是純二進位格式產生 PGP 資料。符合 ASCII 碼的格式設定可產生電子郵件易讀的文字，包括頁首和頁尾。
憑證	<code>certificate</code>	用於加密的憑證，由下列其中一種 URL 類型提供： - 附件 URL，用於從用戶處擷取公開金鑰。必須遵循模式 <code>attachment:certificate/WIDvalue</code>，其中 <code>WIDvalue</code> 是儲存在用戶中的公開金鑰的 WID。例如，您可以在表單中使用整合附件來配置推薦信。<code>attachment:certificate/@{intsys.getAttributeReferencedEncryption Key', 'WID'})}</code> 其中 PGP Encryption Key 是整合屬性內容的名稱。 <code>mctx</code> URL，用於從由整合儲存的环境定義變數中擷取公開金鑰。範例：<code>mctx:vars/MyPublicKey</code> 其中 <code>MyPublicKey</code> 是環境定義變數的名稱。 - 儲存在程序集的 WSAR-INF 資料夾或其任一子資料夾中的公開金鑰的檔案名稱。範例：<code>PublicKey.asc</code>

屬性	XML 屬性內容名稱	說明
		或 /keys/ PublicKey.asc。
包含完整性檢查	containing-integrity-check	指出 pgp-加密 步驟是否產生訊息簽章，以供接收者進行完整性檢查。
解密的檔案名稱	decrypted-filename	解密檔案的名稱。範例： yourname.plaintext.mail。Workday 會在雜湊字串中包含指定的值。
Pgp26x 相容	pgp26x-compatible	指出 pgp-加密 步驟是否產生向後兼容舊 pgp26x 版本的加密資料。
私人金鑰密碼	private-key-passphrase	指定私人金鑰的密碼。
簽署私人金鑰	signing-private-key	用來以檔案名稱形式簽署訊息的私人金鑰，使用 file:// 通訊協定 註: Workday Studio 整合的最佳做法是使用傳送服務來執行加密，而不是使用 pgp-加密 步驟。
文字模式	textmode	指定 true 表示要加密的資料為文字格式，且包含 <CR><LF> 行結尾。 註: 如果您指定 true 對於不包含 <CR><LF> 行結尾的資料，Workday 會報告錯誤。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 pgp-加密 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 pgp-加密 步驟的條件。

relatedlinks

相關參考資料

[Workday 31 功能更新公告：Workday Studio](#)

參照：**PGP** 解密步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集內 pgp-decrypt 步驟的唯一 ID。

屬性	XML 屬性內容名稱	說明
輸入	input	指定 pgp-decrypt 步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。
輸出	output	指定 pgp-decrypt 步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 MediationContext。
私人金鑰	private-key	Workday 私人金鑰的檔案名稱或 URI。使用 file:// 協議，將其指定為檔案名稱。
密碼	passphrase	指定私人金鑰的密碼。
解密的檔案名稱	decrypted-filename	設定 content-disposition pgp-decrypt 步驟輸出中的 . 屬性，若輸出為變數，則為檔案名。但是，如果 pgp-加密 步驟指定了解密的檔案名稱 值，則優先使用該值。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 pgp-decrypt 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 pgp-decrypt 步驟的條件。

參照：Base64 解碼步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	組合體中 Base64 位元解碼 步驟的唯一 ID。
輸入	input	指定 base64 解碼 步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。
輸出	output	指定 base64 解碼 步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取

屬性	XML 屬性內容名稱	說明
		整封訊息。此值會保留原始訊息中的所有附件。 變數：將輸出另存為具名變數 <code>MediationContext</code>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 base64 解碼 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<code>execute-when</code>	決定是否執行 base64 解碼 步驟的條件。

參照：Base64 編碼步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	<code>id</code>	Base64 編碼 步驟在程序集中的唯一 ID。
輸入	<code>input</code>	指定 base64 編碼 步驟從何處取得資料做為輸入。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <code>rootpart</code>：的內容 <code>rootpart</code> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 - 變數：使用中的具名變數 <code>MediationContext</code>。
輸出	<code>output</code>	指定 base64 編碼 步驟將輸出導向何處。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。此值會建立新訊息，有效地移除所有附件。

屬性	XML 屬性內容名稱	說明
		<code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <code>rootpart</code>：的內容 <code>rootpart</code> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 base64 編碼 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<code>execute-when</code>	決定 Base64 編碼 步驟是否執行的條件。

參照：Zip 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	<code>id</code>	程序集內 zip 步驟的唯一 ID。
輸入	<code>input</code>	指定 zip 步驟從何處取得資料做為輸入。有 5 個可能的值： <code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <code>rootpart</code>：的內容 <code>rootpart</code> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 - 變數：使用中的具名變數 <code>MediationContext</code>。

屬性	XML 屬性內容名稱	說明
輸出	output	指定 zip 步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 MediationContext。
格式	format	壓縮檔案的格式。選項包括： • <i>zip</i> • 焦油 註： TAR 格式支援封存，但不支援壓縮。若要壓縮 TAR 檔案，請使用下列任一指令： • <i>gzip</i>，其會將 .tar 檔的副檔名取代為 .tar.gz 或 .tgz • <i>bzip2</i>，將副檔名 .tar 取代為 .tar.bz2 您可以分別使用 壓縮 和 解壓縮 步驟來壓縮和解壓縮在 zip 步驟中產生的 TAR 封存文件。
層級	level	壓縮層級，範圍為 0 到 9，其中 9 為最大壓縮值。預設值為 -1。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	zip 步驟的說明。顯示在合併記錄中。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定 zip 步驟是否執行的條件。

屬性	XML 屬性內容名稱	說明
備註	comment	內嵌在 ZIP 檔案中的備註。備註是最多 65,535 位元組的任意 ASCII 字串。

參照：解壓縮步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	程序集內 解壓縮 步驟的唯一 ID。
輸入	input	指定 解壓縮 步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。
輸出	output	指定 解壓縮 步驟會將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 MediationContext。
格式	format	壓縮檔案的格式。選項包括：

屬性	XML 屬性內容名稱	說明
		• <i>zip</i> • 焦油 註: TAR 格式支援封存，但不支援壓縮。若要壓縮 TAR 檔案，請使用下列任一指令： • <i>gzip</i>，其會將 <i>.tar</i> 檔的副檔名取代為 <i>.tar.gz</i> 或 <i>.tgz</i> • <i>bzip2</i>，將副檔名 <i>.tar</i> 取代為 <i>.tar.bz2</i> 您可以分別使用 壓縮 和解 壓縮 步驟來壓縮和解壓縮在 <i>zip</i> 步驟中產生的 TAR 封存文件。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<i>description</i>	解壓縮的說明 <i>MediationContext</i> 。
執行時間	<i>execute-when</i>	決定是否執行 解壓縮 步驟的條件。
檔案名稱	<i>filename</i>	內容解壓縮時建立的檔案名稱。

參照：壓縮步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	<i>id</i>	程序集內 壓縮 步驟的唯一 ID。
輸入	<i>input</i>	指定 壓縮 步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 <i>MediationContext</i>。

屬性	XML 屬性內容名稱	說明
輸出	output	指定 壓縮 步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 <i>MediationContext</i>。
格式	format	壓縮檔案的格式。選項為 <i>gzip</i> 和 <i>bzip2</i> 。預設格式為 <i>gzip</i> 。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 壓縮 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 壓縮 步驟的條件。
檔案名稱	filename	內容解壓縮時建立的檔案名稱。

參照：解壓縮步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	組合體中 解壓縮 步驟的唯一 ID。
輸入	input	指定 解壓縮 步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊

屬性	XML 屬性內容名稱	說明
		息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 解壓縮 步驟將輸出導向何處。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
格式	format	壓縮檔案的格式。選項為 <i>gzip</i> 和 <i>bzip2</i>。預設格式為 <i>gzip</i>。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 解壓縮 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 解壓縮 步驟的條件。
檔案名稱	filename	內容解壓縮時建立的檔案名稱。

其他步驟

概念：其他步驟

步驟可讓您在訊息經過整合時對這些訊息執行個別作業。Workday Studio 的 選用板提供以下 其他 步驟：

名稱	函數
custom	可讓您將使用 Spring bean 撰寫的自訂代碼套用至傳入訊息。
set-dynamic-endpoint	配置傳出的 WS-Addressing (WSA) 端點資訊。
商店	建立文件的永久副本，以供其他使用者使用和/或視為整合事件的附件。 儲存步驟的個別訊息大小上限為 1 GB (壓縮)。累計限制為 3 GB (壓縮)。Workday 會終止超出這些限制的整合作業。請盡可能使用拆分器或串流技術來拆分整合中的大型檔案。
擷取	擷取先前由 儲存 步驟儲存的文件的永久副本。
xmldiff	比較 2 個 XML 輸入文件。
enqueue-message	將訊息整理到佇列中。

參照：自訂步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	指定 自訂 步驟在程序集中的唯一 ID。
輸入	input	指定 自訂 步驟從何處取得資料做為輸入。有 5 個可能的值： <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 - 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 自訂 步驟會將輸出導向何處。有 5 個可能的值：

屬性	XML 屬性內容名稱	說明
		<i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <i>MediationContext</i>。
方法名稱	method-name	自訂 Java 類別中的 Java 方法名稱。
Spring Bean	ref	由 自訂 步驟實作的自訂 Spring bean Java 類別。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 自訂 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 自訂 步驟的條件。

參照：設定動態端點步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	集合內 動態端點設定 步驟的唯一 ID。
端點	endpoint	動態端點的位址。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 set-Dynamic-endpoint 步驟的說明。

屬性	XML 屬性內容名稱	說明
		您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	一個條件，決定是否執行 set-Dynamic-endpoint 步驟。

參照：儲存步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	商店 步驟在程序集中的唯一 ID。
輸入	input	指定 儲存 步驟從何處取得資料做為輸入。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。
輸出	output	指定 商店 步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 MediationContext。

屬性	XML 屬性內容名稱	說明
		儲存 步驟可將訊息的 MIME 類型與訊息本身一起儲存。範例：「<i>text/plain</i>」。當您使用 擷取 步驟擷取訊息時，Workday 會自動以儲存的 MIME 類型標示訊息，以便後續步驟可正確處理訊息。 儲存步驟的個別訊息大小上限為 1 GB (壓縮)。累計限制為 3 GB (壓縮)。Workday 會終止超出這些限制的整合作業。請盡可能使用拆分器或串流技術來拆分整合中的大型檔案。
收款	collection	儲存文件的集合名稱。
到期時間	expiresIn	您的文件在多長時間前到期。 以 XSD 期間格式指定值。 例如，若要表示 1 年 3 個月 5 天 8 小時 45 分 10 秒，請輸入 P1Y3M5DT8H45M10S。
摘要	summary	文件摘要顯示在您的 Workday 用戶中。
標題	title	文件的標題。
建立文件參考	createDocumentReference	指定 儲存 步驟是否自動呼叫 Workday 以建立文件參照。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 商店 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 商店 步驟的條件。
內容捨棄	contentDisposition	文件的內容捨棄標頭。 覆寫 標題 屬性值 (若有指定)。
Blob 儲存庫 URI	blobitoryURI	的 URI cc-blobitory 在 Workday 用戶中實作的服務 範例：https://machine-name.workday.com:8080/ccx/cc-blobitory。
到期日	expiresOn	文件到期的日期和時間。 覆寫 「Expires In」 屬性值 (若有指定)。

屬性	XML 屬性內容名稱	說明
		在中輸入此屬性的值 CCYY-MM-DDThh:mm:ssZ 格式 範例：若要表示 2009 年 12 月 31 日晚上 11:59，請輸入 2009-12-31T23:59:00Z。
結構描述	schema	參照您文件的結構描述 URI。有 2 個可能的值： • http://www.w3.org/2005/Atom • urn:com.workday/bsvc/blob
項目 ID	entryID	指定每個文件的項目 ID。在您的用戶中不能重複。若未指定項目 ID，Workday 會產生一個 ID。如果您指定的是已存在的文件，Workday 會覆寫可變更的文件。如果現有文件是不可變的，則無法重複使用項目 ID。 使用 REST API 更新文件時： • 對具有預先定義的項目 ID 的文件使用 PUT HTTP 方法。 • 當未指定項目 ID，或計算結果為空值或空字串時，請使用 POST HTTP 方法。
是附件	isAttachment	指定您的文件是否為 OMS 附件。
不可變	immutable	指定您是否可以在建立文件後進行修改。
指派擁有者	assignOwner	指定 Workday 是否將您指派為文件的擁有者。
文件參考說明	documentReferenceDescription	文件的說明。顯示在您的 Workday 用戶中。

參照：擷取步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	指定集合中 擷取 步驟的唯一 ID。
輸出	output	指定 擷取 步驟會將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。此值

屬性	XML 屬性內容名稱	說明
		會建立新訊息，有效地移除所有附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <code>rootpart</code>：的內容 <code>rootpart</code> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。 當您在業務流程中配置文件擷取服務時，Workday 會使用您在「自訂內容類型」欄位中輸入的字串，做為該服務所儲存文件的 MIME 類型。 當您使用 擷取 步驟 (直接或透過文件迭代器或文件存取器 Helper 物件) 擷取文件時，Workday 會自動使用儲存的 MIME 類型標示該文件，以便後續步驟可以正確處理文件。如果您未輸入「自訂內容類型」，Workday 會使用 UTF-8 編碼。為避免發生錯誤，對於非 UTF-8 編碼的資料，請一律輸入適當的 自訂內容類型。
收款	<code>collection</code>	儲存文件的集合名稱。
項目	<code>entry</code>	集合中文件的 ID。可擷取自 <code>MediationContext</code> 使用 MVEL 運算式 範例： <pre>@{parts[0].xpath('//blob:entry', 'blob urn:com.workday/bsvc/blob')}</pre>
密碼	<code>password</code>	存取文件的密碼憑證。
使用者名稱	<code>username</code>	存取文件的使用者名稱憑證。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 擷取 步驟說明。您可以使用此功能來說

屬性	XML 屬性內容名稱	說明
		明步驟的功能以及執行步驟的資料。
執行時間	execute-when	決定是否執行 擷取 步驟的條件。

參照：Xmldiff 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	指定 xmldiff 步驟在程序集中的唯一 ID。
第一個輸入	first-input	指定 xmldiff 步驟的第一個資料輸入來源。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。
第二輸入	second-input	指定 xmldiff 步驟的第二個資料輸入來源。有 5 個可能的值： • <i>message</i>：整封訊息，包括 rootpart 和附件。 • <i>soapbody</i>：「」中的 SOAP 內文 rootpart 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 rootpart 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 MediationContext。
輸出	output	指定 xmldiff 步驟會將輸出導向何處。有 5 個可能的值：

屬性	XML 屬性內容名稱	說明
		<code>message</code>：整封訊息，包括 <code>rootpart</code> 和附件。此值會建立新訊息，有效地移除所有附件。 <code>soapbody</code>：「」中的 SOAP 內文 <code>rootpart</code> 訊息的名稱。 - 附件：訊息內的 MIME 附件。請指定附件索引。 <code>rootpart</code>：的內容 <code>rootpart</code> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 - 變數：將輸出另存為具名變數 <code>MediationContext</code>。
複製到輸出	<code>copy-to-output</code>	指定 <code>xmldiff</code> 步驟是將輸出複製到中介訊息還是變數。 如果設為 <code>True</code>，則 <code>xmldiff</code> 步驟會將輸出複製到中介訊息。 如果設為 <code>False</code>，則 <code>xmldiff</code> 步驟會將輸出複製到變數。
忽略屬性內容	<code>ignore-attributes</code>	指定在比較 2 個 XML 文件時， <code>xmldiff</code> 步驟是否忽略屬性內容。
名稱空間	<code>namespaces</code>	由 <code>xmldiff</code> 步驟實作 XPath 運算式的名稱空間覆寫定義。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	<code>description</code>	顯示在合併記錄中的 <code>xmldiff</code> 步驟說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。
執行時間	<code>execute-when</code>	決定 <code>xmldiff</code> 步驟是否執行的條件。
XPath 版本	<code>xpath-version</code>	指定使用 XPath 1.0 版還是 2.0 版。對於 Workday 11 到的程序集版本，請輸入值 1，對於 Workday 12 之後的程序集版本，請輸入值 2。如果「Stream-xPath」的值設為 <code>True</code>，Workday 會忽略此屬性。 您可以讓 Workday 使用可插入系統屬性值的 MVEL 範

屬性	XML 屬性內容名稱	說明
		本動態設定版本。範例： <code>@{props.get('the_version')}</code> 。

定義分頁

屬性	XML 屬性內容名稱	說明
名稱	name	指定名稱 sub-diff 在查詢 xmldiff 步驟的結果時使用。
忽略刪除	ignore-deletions	指定 xmldiff 步驟是否忽略輸出中的刪除項目。
僅限不同的節點	result-document-different-nodes-only	指定在 xmldiff 步驟的輸出文件中顯示哪些節點。 如果設為 true，輸出文件中只會顯示最上層節點。 如果設為 false，所有子系節點都會顯示在輸出文件中。
包含的結果文件	result-document-inclusion	指定 xmldiff 步驟是否記錄輸出文件中節點之間的差異。有 3 個可能的值： - 一律：選取的節點及其子系一律顯示在結果文件中。 - 不同：只有在發現差異時，才會在結果文件中顯示選取的節點及其子系。 - 從不：選取的節點及其子係從不在結果文件中顯示。
XPath	xpath	由 xmldiff 步驟實作以在輸入文件中尋找節點的 XPath 運算式。

比對器 XPath 分頁

在 比對器 XPath 分頁上，您可以新增 XPath 運算式，以便在輸入文件中找出無順序的比對節點。

參照：Enqueue-Message 步驟屬性

通用分頁

屬性	XML 屬性內容名稱	說明
步驟 ID	id	enqueue-message 步驟在程序集中的唯一 ID。
輸入	input	指定 enqueue-message 步驟從何處取得資料做為輸入。有 5 個可能的值： message：整封訊息，包括 rootpart 和附件。

屬性	XML 屬性內容名稱	說明
		• <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。僅適用於輸入包含 SOAP 訊息時。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。 • 變數：使用中的具名變數 <i>MediationContext</i>。
輸出	output	指定 <i>enqueue-message</i> 步驟將輸出導向何處。有 5 個可能的值： • <i>message</i>：整封訊息，包括 <i>rootpart</i> 和附件。此值會建立新訊息，有效地移除所有附件。 • <i>soapbody</i>：「」中的 SOAP 內文 <i>rootpart</i> 訊息的名稱。 • 附件：訊息內的 MIME 附件。請指定附件索引。 • <i>rootpart</i>：的內容 <i>rootpart</i> 的 MIME 訊息如果訊息不是 MIME，Workday 會選取整封訊息。此值會保留原始訊息中的所有附件。 • 變數：將輸出另存為具名變數 <i>MediationContext</i>。
內容類型	contentType	覆寫輸出訊息的內容類型，預設為 <i>text/xml</i> 。
佇列名稱	queueName	<i>enqueue-message</i> 步驟會將輸出導向其中的佇列名稱。
檔案名稱	fileName	覆寫中介訊息標頭中的檔案名稱。

進階分頁

屬性	XML 屬性內容名稱	說明
說明	description	顯示在合併記錄中的 <i>enqueue-message</i> 步驟的說明。您可以使用此功能來說明步驟的功能以及執行步驟的資料。

屬性	XML 屬性內容名稱	說明
執行時間	execute-when	決定是否執行 enqueue-message 步驟的條件。

集合模組

建立集合模組

開始之前

建立程序集專案。

執行這項作業的原因和時機

Workday Studio 組合件模組提供用於建置組合件的參數化範本。它們可讓您建立和重複使用預先配置的元件集和步驟集，以執行特定任務。

程序

- 新增您要包含在集合模組中的元素，並視需要配置其屬性。您可以從功能完整的組合體建立模組，也可以從較少的元件和步驟集建立模組。使用者可在實作模組時調整參數。
- 儲存您的程序集。
- 在 專案資源管理器中，以滑鼠右鍵按一下組合專案，然後選取「建立組合模組...」。
- 提供模組的名稱和說明。說明會顯示為模組的工具提示。
- 選取模組的圖示。您可以使用預設圖示，也可以上傳您自己的圖示。
- Studio 會將模組參數整理到稱為「頁面」的集合中。參數必須屬於頁面。每個頁面都有名稱和說明。每個參數都有名稱、標籤 (顯示在「配置模組」精靈中)、說明、類型和預設值。參數類型共有 6 種：

類型	說明	範例
String	文字值。	"HelloWorld"
XMLRequest	要求訊息的 XML 文字。模組使用者可使用「建立 Literal XML」或「建立 Workday Web 服務要求」精靈來指定 Web 服務要求的值。	<?xml version="1.0" encoding="UTF-8"?> <env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <env:Body> <wd:Get_Workers_Request xmlns:wd="urn:com.workday/bsvc" wd:version="v23.2"> <wd:Request_References wd:Skip_Non_Existing_Instances="t <wd:Worker_Reference> <wd:ID wd:type="Employee_ID">abcdef</wd:ID> </wd:Worker_Reference> </

類型	說明	範例
		wd:Request_References> </ wd:Get_Workers_Request> </env:Body> </ env:Envelope>
提示	您可以使用類別報告欄位 (CRF) 來提示輸入特定類型 Workday 物件的啟動參數。若要指定 CRF 參考資料，請按一下 選取參考資料 ID。 註：若要指定 CRF 參照，您必須連線至 Workday 用戶。	fbbe44a681251000039acc2ff657000a
檔案	檔案名稱您可以將副檔名指定為 data 值。	*.xslt
列舉	以逗號或空格分隔的單一常值或多個值。	code1 code2 code3
布林運算子	布林值。	true

- 當您完成新增參數並儲存模組時，Studio 會在 XML 編輯器中開啟其 XMI 檔案。檔案包含 params 為您指定的每個參數新增一個元素。該 assemblyText 和 diagramText 元素包含封裝在內的程序集 XML 和圖表詳細資料。<![CDATA[]]> 標記。您可以直接編輯 XMI 檔案。
- 關閉 XMI 檔案。

將組套件模組新增至專案

開始之前

建立組合專案或開啟現有的組合專案。

執行這項作業的原因和時機

Workday Studio 組套件模組提供用於建置組合件的參數化範本。它們可讓您建立和重複使用預先配置的元件集和步驟集，以執行特定任務。

程序

- 將模組從「選用 板」的「模組」區段拖曳至程序集編輯器。Studio 會開啟「配置模組」精靈。
註：模組庫 只會顯示您在 視窗 > 喜好設定的 > Workday > 組合編輯 > 器模組 偏好設定頁面上選取的組合模組。
- 配置模組的參數
- 如果組套件模組包含類型為「」的參數 XMLRequest，按一下 建立 Literal XML 或 建立 Workday Web 服務要求，指定 Web 服務要求的值。執行此任務時，請參考以下資訊：

精靈	營運	附註
建立文字 XML	Get, Put	- 可用於任何 Web 服務。 - 從文字窗格下方的清單中，選取要求的 Web 服務名稱和 Web 服務版本值。

精靈	營運	附註
建立 Workday Web 服務要求	Get	- 僅用於 Workday Web 服務。 - Studio 會在您選取作業時自動填入 Web 服務名稱和版本的值。

4. Studio 會顯示模組所含組合元素的預覽。當您完成精靈時，Studio 會將這些元素新增至集合。

概念：集合模組

Workday Studio 組件模組提供用於建置組合件的參數化範本。它們可讓您建立和重複使用預先配置的元件集和步驟集，以執行特定任務。您可以從整個組件建立模組，也可以從其中的區段建立模組。

您使用模組建立的程序集與您從頭開始建立的程序集並沒有什麼不同。兩者的外觀和行為皆相同。

Studio 會將模組另存為 XML 中繼資料交換 (XMI) 格式的範本，副檔名為 xmi。從 Studio 調色盤上的 模組庫 存取您儲存的模組。若要讓其他開發人員存取您的模組，請將模組儲存在共用位置。

共用久經驗證的集合模組有助於加快整合開發並降低成本。

模組庫 只會顯示您在 視窗 > 喜好設定的 > Workday > 組合編輯 > 器模組 偏好設定頁面上選取的組合模組。

子裝配件

建立子裝配件

開始之前

建立程序集。

執行這項作業的原因和時機

「子裝配件」是一個處理鏈，其第一個組成要素是 本地輸入的 傳輸。您可以從同一個程序集內或從不同的程序集，使用 本端傳出 傳輸來呼叫它。

程序

1. 新增並配置定義子組合件開頭的 本機輸入 法。
 - a) 在您要轉換為子裝配件的處理鏈開頭新增 本機輸入 傳輸功能。
 - b) 在其「屬性」檢視的「通用」和「進階」分頁上，配置傳輸的屬性。您可以提供子裝配件的圖示，以及做為其工具提示的文字。
 - c) 在 參數 分頁上，定義子裝配件需要做為輸入的參數。您可以在呼叫子組合件的 本端傳出 傳輸上配置這些參數。Studio 會將其儲存為 MediationContext 屬性。
 - d) 在 簽出參數 分頁上，定義 MediationContext 子組合體設定的屬性。
 - e) 儲存集合。
2. 新增並配置呼叫子組合件的 本端傳出 傳輸。
 - a) 將 本機輸出 傳輸新增至呼叫子組合件的組合件。
 - b) 在其「屬性」檢視的「通用」和「進階」分頁上，配置其屬性。將子組合件的 本地 輸入法指定為 端點 屬性。使用格式 `vm://assembly-name/local-out-ID`，或按一下 選取使用 當地文字的子程序集。
 - c) 在 參數 分頁上，設定 MediationContext 屬性您可以按一下 建立 'set' 元素來建立新參數。您也可以按一下 選取參數，選取您在 本機輸入 傳輸中定義的任何參數，並為其指派值。
 註：在 選取參數 精靈中，星號表示必須有屬性。問號表示使用情況為動態。
 - d) 儲存集合。

概念：子裝配件

「子裝配件」是一個處理鏈，其第一個組成要素是本地輸入的傳輸。本地輸入 - 定義處理鏈的進入點，使其能夠由相同程序集或另一個程序集中的本端傳出傳輸調用。您可以在同一個程序集中包含多個子程序集和本機傳出傳輸。

註：您無法將子組件單獨部署至 Workday，因為這些組件一開始不是以 Workday-in 傳輸方式命名。

由於您可以隨意調用子組合，因此您可以使用子組合封裝處理邏輯以供重複使用。您可以將其視為程式設計子例程。本地輸入傳輸的預設 ID 為 SubRoutine。本地傳出傳輸的預設 ID 為 CallSubRoutine。

子裝配件具有參數。您可以在本地輸入傳輸上指定這些值，並在本地輸出值上為其指派值。這些參數可定義「」中的屬性 MediationContext。

當您儲存子裝配件時，Workday Studio 會自動將呼叫該裝配件的本機輸出傳輸，新增至調色板的「工作區元件」種類。如果您提供了子裝配件的本機輸入傳輸中的自訂圖示，此本地輸出傳輸功能將使用該自訂圖示。否則，系統會使用問號圖示。

調色盤一律顯示公開子裝配件。只有在集合中出現私人子裝配件時，才會顯示私人子裝配件。Studio 在調色盤的「通用元件」種類中提供了多個預先封裝的子裝配件。

若要找出本機-出碼參照的以本機輸入的傳輸方式，請以滑鼠右鍵按一下本機輸出，然後選取工作區中的申報。

若要找出本機輸入參照的本機-調出傳輸，請在本機輸入上按一下滑鼠右鍵，然後選取工作區中的參照。

relatedlinks

相關概念

[概念：通用元件](#) 頁 80

自訂集合元件

建立自訂 Java 類和 Spring Bean

開始之前

建立程序集專案。

執行這項作業的原因和時機

您可以從 Workday Studio 的調色板建立 4 個自訂組合元素：

- custom-mediation 元件
- 自訂步驟
- 自訂運出
- custom-error-handler

第五個自訂元素「自訂路由策略」可從「路由」元件的環境定義工具列使用。

程序

1. 在集合編輯器中開啟集合。
2. 將下列其中一個自訂元素新增至集合體：
 - custom-mediation 元件
 - 自訂步驟
 - 自訂運出
 - custom-error-handler
3. 在自訂元素的「屬性」檢視中配置其屬性。

4. 按一下「Spring Bean」欄位右側的「建立 Spring Bean 屬性和 Java 類別」。Studio 顯示的屬性會因自訂元素類型而略有不同。執行此任務時，請參考以下說明：

選項	敘述
ID	請提供唯一的 Spring bean ID。
範圍	- 選取 單一實例 可讓每個參照傳回相同的 Bean 實例。 - 選取「原型」，每個參照都會傳回不同的 Bean 實例。
方案	指定 Java 封裝。
名稱	請指定類別名稱。
輸入類型	選取要傳遞至 Java 方法的輸入類型。您不需要指定實作詳細資料，即可存取訊息部分並將其轉換為適當的類型。Workday 會檢查方法的第一個參數，並將訊息部分轉換為相同的類型。
輸出類型	選取方法的輸出 (傳回) 類型。
方法名稱	輸入您要在 Java 類別中定義的 Java 方法名稱。預設值為 process。
包含動態端點的引數	指定是否包含動態端點的引數。
要顯示在選用板中的註解 Java 類別	選取此核取方塊可在 調色盤上顯示自訂元素。此選項會將註解新增至產生的 Java 類別，該類別位於程序集專案資料夾的 src 資料夾中。
工具提示	輸入顯示為自訂元素說明的文字。

5. 儲存集合。

結果

Studio 會產生代表指定封裝內自訂元素的 Java 類別。也會在assembly.xml 檔案的底部新增對新自訂Spring bean 的參照。

下一步

將必要的代碼新增至自訂 Java 類別。

概念：自訂集合元素

您可以在 Workday Studio 中建立 5 種類型的自訂程序集元素：

自訂元素	程序
中介元件	請從「調色盤」的「元件」種類新增 自訂中介元件。
中介步驟	請從 調色盤上的「其他」種類新增 自訂 步驟。
傳出傳輸	請從 調色盤上的 傳出傳輸 種類新增 自訂傳出 傳出。
錯誤處理程式	從 選用板的 錯誤處理程式 種類中新增一個 custom-error-handler 元件。
路由策略	從 路由 元件的操作工具列中選取 建立自訂策略 。

在每種情況下，自訂元素都由 3 個部分組成：

- 基礎 Java 實作，定義元素使用的類別和方法。
- 說明 Workday 如何建立 Java 類別實例的 Spring Bean 配置。
- 組合體元件配置，用於將自訂元素置於圖形組合體檢視畫面上。

自訂程序集元素可在處理傳入資料時，以一般方式存取訊息部分。您也可以指定 Workday 將自訂元素的輸出導向其中的訊息部分。

若要共用自訂元素，請將 Java 封裝匯出為二進位 JAR。然後，其他開發人員可以將 JAR 新增至其專案的類別路徑。

自訂步驟

為自訂步驟撰寫 Java 實作時，請考量以下幾點：

- 必須聲明該方法 `public` 以便可以從類別外部呼叫。
- 方法必須為非靜態。
- 每個方法的名稱不可重複。
- 參數清單必須符合支援的組合。
- 此方法可引發任何類型的例外。
- 若未提供方法名稱，Workday 會選取參數清單與支援方法簽章相符的任何公開方法 (忽略 `getter` 和 `setter`)。如有任何不明確之處，Workday 會在部署期間報告錯誤，您可透過指定方法名稱來解決此錯誤。

自訂步驟必須提供符合下列其中一個簽章的方法：

簽名	用途	範例
<code>public return type method(input parameter type input) [throws any Exception]</code>	定義方法的傳回類型和輸入參數。	<code>public String addTimestamp(byte[] messageContent) throws IOException {</code>
<code>public void method(input parameter type input, output parameter type output) [throws any Exception]</code>	定義支援的輸入和輸出參數。	<code>public void addTimestamp(byte[] messageContent, OutputStream out) throws IOException {</code>

自訂步驟支援以下輸入、傳回和輸出參數類型：

輸入類型	傳回類型	輸出參數類型
• <code>java.lang.String</code> • <code>byte[]</code> • <code>java.io.InputStream</code> • <code>javax.activation.DataHandler</code> • <code>javax.activation.DataSource</code> • <code>org.w3c.dom.Element</code> • <code>org.w3c.dom.Document</code> • <code>javax.xml.transform.Source</code> • <code>javax.xml.transform.stream.StreamSource</code> • <code>javax.xml.transform.dom.DOMSource</code>	• <code>void</code> • <code>boolean</code> (僅限自訂步驟) • <code>java.lang.String</code> • <code>byte[]</code> • <code>java.io.InputStream</code> • <code>javax.activation.DataHandler</code> • <code>javax.activation.DataSource</code> • <code>org.w3c.dom.Element</code> • <code>org.w3c.dom.Document</code> • <code>javax.xml.transform.Source</code>	• <code>java.io.OutputStream</code> • <code>javax.xml.transform.Result</code>

如果自訂步驟傳回 `void` 類型或空值，則不會變更中介訊息。

概念：自訂元素封裝

Java 類別註解

如果您想在不同的專案中重複使用自訂組合元素，請使用註解讓元素顯示在 調色盤上。

為此，請在您建立 Java 類和 Spring bean 時選取「將 Java 類註解為顯示在選用板」核取方塊。Studio 會新增 @Component 產生的 Java 類別中的註解，該類別位於程序集專案的 src 資料夾中。範例：

```
import com.capeclear.assembly.annotation.Component;
import com.capeclear.assembly.annotation.ComponentMethod;
import com.capeclear.assembly.annotation.Property;
import static com.capeclear.assembly.annotation.Component.Type.*;

@Component(
    name = "TimeStamperCustomMediationStepImpl",
    type = mediation,
    tooltip = "This custom code step adds the current time on the first
line of the message.",
    scope = "singleton",
    smallIconPath = "icons/TimeStamperCustomMediationStepImpl_16.png",
    largeIconPath = "icons/TimeStamperCustomMediationStepImpl_24.png"
)
```

您可以手動編輯此資訊。

請注意，Studio 會將自訂元素的圖示儲存在專案的 src/icons 資料夾。有 2 個版本，16x16 和 24x24 像素。若要使用自訂圖示，請將 _16.png 和 _24.png 檔案取代為您自己的影像。

顯示自訂集合元素屬性

如果您的自訂 Java 代碼具有您希望使用者設定的屬性，您可以在自訂元素的「屬性」檢視的 Bean 分頁上顯示這些屬性。

為此，請提供遵循 Bean 模式的 getter 和 setter 方法。範例：針對屬性 abc，提供 getter 和 setter 方法，例如 getAbc 和 setAbc。

然後新增 @Property 註解 set 或 get 方法該 name 值會在「屬性」檢視中定義屬性的標籤，而 tooltip 值可定義當您將滑鼠移至標籤上方時 Studio 顯示的文字。您可以使用順序值來控制屬性列出的順序。首先顯示較低的值。一個 editelement 值會指定在屬性輸入欄位旁邊顯示的按鈕類型，如下所示：

```
// Displays a text box with a Select button to enable you to select a Spring
bean.
@property(name="Text", edit=Edit.BEAN_REF)

// Displays a text box with a Select button to enable you to select a Java
class.
@property(name="Text", edit=Edit.CLASS)

// Displays a text box with a dialog box that enables you to specify a cron
expression.
@property(name="Text", edit=Edit.CRON)

// Displays a text box without a button.
@property(name="Text", edit=Edit.DEFAULT)

// Displays a Browse button to enable you to specify a file.
@property(name="Text", edit=Edit.FILE)

// Displays a Browse button to enable you to specify a folder.
```

```
@Property(name="Text", edit=Edit.FOLDER)
```

Studio 會在包含正在編輯的 components.xml 檔案的專案的專案類別路徑中尋找自訂類別。要將您的自訂元件發布至多個組合專案，請將二進位格式的自訂項目打包為一個包含類別檔案和圖示影像的 JAR 檔案。其他使用者可將此 JAR 包含在專案的類別路徑中，以便在其組合編輯器中使用。

集合編輯器會在啟動時自動載入其在類別路徑中找到的任何自訂元件。在組合圖背景上按一下滑鼠右鍵，然後選取 **重新整理自訂項目**，重新載入任何最近新增的自訂元件。

Spring Bean 配置

概念：Spring Bean 配置

您可以在 Assembly Editor 的 Properties 檢視中，在 Beans 分頁上編輯 Spring bean 屬性。

Studio 會在檢視左側的最上層 bean 節點下方顯示程序集的所有 bean。與 bean 節點相關聯的屬性是全域屬性，會影響集合中的所有 bean。

各種選項會顯示在 Bean 上方。其可用性取決於您在 Bean 清單中的選擇。選項包括：

選項	說明
建立 'bean'	在組合物檔案中建立 Spring bean 元素。請選取最上層 Bean 節點，以使此選項可用。
建立「別名」	為組合物檔案中的 Spring bean 元素建立別名。請選取最上層 Bean 節點，以使此選項可用。
建立「屬性」	為選取的 Bean 建立屬性元素。
建立「constructor-arg」	建立 constructor-arg 選定 Bean 的元素
建立「ref」	建立 ref 元素，讓其他 Bean 參照選定的屬性。
建立「清單」	建立 list 選取屬性的元素，讓您能夠定義 List Java 集合的屬性。
建立「地圖」	建立 map 選取屬性的元素，讓您能夠定義對應 Java 集合的屬性。
建立「匯入」	建立最上層 import 組合體的 bean 定義內的元素，讓您指定內含一或多個 bean 定義的資源檔案。請選取最上層 Bean，以使此選項可用。
建立「值」	建立內嵌 value 選取屬性定義中的元素，讓您能夠以可讀字串的形式指定屬性。

參照：全域 Bean 屬性

屬性	XML 屬性內容名稱	說明
預設自動連線	default-autowire	檢查 BeanFactory 以自動解析 Bean。
預設自動電匯候選人	default-autowire-candidate	可讓您根據與 bean 名稱的模式比對，將自動連線限制為一組候選 bean。
預設依存關係檢查	default-dependency-check	指定 Studio 是否檢查未解析的 Bean 依存關係。

屬性	XML 屬性內容名稱	說明
預設破壞方式	default-destroy-method	為指定的析構回呼方法檢查每個 Bean。
預設起始方法	default-init-method	每個 Bean 的初始化回呼方法。
預設延遲起始	default-lazy-init	指定是在啟動時建立，還是在第一次提出要求時建立 Bean 實例。如果您不希望在使用 ApplicationContext，設為 True。
預設合併	default-merge	決定 Studio 是否在預設情況下合併集合。

參照：個別 **Bean** 屬性

屬性	XML 屬性內容名稱	說明
ID	id	組合內自訂 Bean 元素的唯一 ID。
Abstract	abstract	指出 Bean 是否只能做為將做為子系定義範本的父系 Bean 定義。預設值為 false。
自動連線	autowire	檢查 BeanFactory 以自動解析 Bean。
自動電匯候選人	autowire-candidate	可讓您根據與 bean 名稱的模式比對，將自動連線限制為一組候選 bean。
類別	class	實作類別名稱。
依存關係檢查	dependency-check	表示 Studio 是否會檢查未解析的 Bean 依存關係。
取決於	depends-on	在初始化使用此元素的 bean 之前，強制初始化一或多個 bean。若要表達對多個 Bean 的依存關係，請列出其名稱，以逗號、空格或分號分隔。
破壞方法	destroy-method	針對指定的析構回呼方法檢查 Bean。
工廠 Bean	factory-bean	目前容器或父系容器中的 Bean，用於保存實例方法 (其呼叫會建立物件)。
工廠方法	factory-method	實例化 Bean 時使用的工廠方法名稱。
起始方法	init-method	Bean 的通用初始化方法。
延遲起始	lazy-init	指出應在啟動時或首次提出要求時建立 Bean 實例。如果您不希望在使用 ApplicationContext，設為

屬性	XML 屬性內容名稱	說明
		True。預設情況下，此屬性的值來自容器層級設定。
名稱	name	Bean 的名稱。
父系	parent	父系 Bean。如果指定，則目前的 Bean 繼承其配置資料。
(主要)	primary	將目前的 bean 定義指定為主要的候選自動連線裝置。預設值為 false。
範圍	scope	從 Bean 定義建立的物件範圍。預設值為 singleton。

參照：別名屬性

屬性	XML 屬性內容名稱	說明
別名	alias	別名。
名稱	name	原始 Bean 的 ID。

參照：屬性屬性

屬性	XML 屬性內容名稱	說明
名稱	name	屬性名稱。
Spring Bean	spring-bean	容器中另一個 Bean 的 ID。
值	value	屬性值。

參照：構造函數- Arg 屬性

屬性	XML 屬性內容名稱	說明
索引	index	構造函數引數從零開始的索引。
Spring Bean	spring-bean	容器中另一個 Bean 的 ID。bean 定義中指定的建構函數引數會做為引數傳遞至參照 bean 的建構函數。
類型	type	構造函數引數的類型。
值	value	構造函數引數的值。

參照：參照屬性

屬性	XML 屬性內容名稱	說明
比恩	bean	透過在 名稱 或 ID 中指定其中一個值來參照目標 Bean。
Local	local	透過指定 ID 屬性內容來參照本機目標 Bean。啟用 XML 剖析器，以驗證同一檔案內的 XML ID 參照。

屬性	XML 屬性內容名稱	說明
父系	parent	參照目前容器的父系容器中的目標 Bean。指定 名稱 或 ID 中的其中一個值。

參照：清單屬性

屬性	XML 屬性內容名稱	說明
合併	merge	指定是否合併 Java List 集合。預設情況下，merge 屬性從容器層級取得值 default-merge 設定。
值類型	value-type	「」的 Java 類型 value 中的個元素。

參照：地圖屬性

屬性	XML 屬性內容名稱	說明
合併	merge	指定是否合併 Java 地圖集合。預設情況下，merge 屬性從容器層級取得值 default-merge 設定。
值類型	value-type	「」的 Java 類型 value 中的個元素。
金鑰類型	key-type	「」的 Java 類型 key 地圖中的元素

訊息建置器

根據結構描述或 WSDL 建立 XML 文字

開始之前

建立包含下列其中一個元素的集合：

- 寫入 步驟
- 記錄 步驟
- Alog-error 錯誤處理程式

執行這項作業的原因和時機

Workday Studio 的「文字建置器」可讓您建立符合所選結構描述的簡單 XML 文件，或建立符合所選 WSDL 的 SOAP 要求訊息。您可以將 WS-Security 標頭附加至 SOAP 要求訊息。

程序

1. 請將 文字 元素新增至下列其中 1 個集合元素：

- 寫入 步驟
- 記錄 步驟
- Alog-error 錯誤處理程式

2. 在右側窗格中，按一下 產生 XML 文字。
3. 在「文字建置器」精靈中，選取您要為其產生 XML 文字的結構描述或 WSDL 元素。執行此任務時，請參考以下說明：

選項	敘述
顯示訊息	預設情況下，Studio 會將 WSDL 元素顯示為訊息。若要將其顯示為介面定義 portTypes、作業、輸入訊息和輸出訊息，請取消選取此選項。
包裝 SOAP	將選取的 XML 定義封裝在 SOAP 信封中。
新增 WS-Security 標頭	新增 SOAP Header 元素新增至 SOAP 要求。SOAP 標頭包含您在「」中提供的 WS-Security 憑證。UsernameToken 元素僅在您選取 SOAP 換行時可用。
自動換行 XSLT	建立包裝在 XSL 樣式表中的 XML 常值。您可以將樣式表另存為 XSLT 檔案以供日後使用。
選擇詢問選項	在建立和復制 XML 常值或 SOAP 常值時，提示您從結構描述中選取特定選項。如果您未選取此選項，則 Studio 會使用特定集合中的第一個選擇。

結果

Studio 會顯示 XML 常值的預覽，供您編輯。

根據結構描述或 WSDL 建立 XPath 運算式

開始之前

建立包含下列其中一個步驟的集合體：

- java-to-xml
- 驗證
- xml-to-java

執行這項作業的原因和時機

在 Workday Studio 中，XPath 運算式建置器可讓您依據選取的結構描述或 WSDL 建立 XPath 運算式。

程序

1. 在下列其中 1 個步驟的「屬性」檢視中，按一下 XPath 屬性旁邊的 建立 XPath 運算式：
 - java-to-xml
 - 驗證
 - xml-to-java
2. 在 XPath 運算式建置器精靈的下方窗格中，選取 XPath 運算式將比對的結構描述或 WSDL 元素。執行此任務時，請參考以下說明：

選項	敘述
顯示訊息	預設情況下，Studio 會將 WSDL 元素顯示為訊息。若要將其顯示為介面定義 portTypes、作業、輸入訊息和輸出訊息，請取消選取此選項。

選項	敘述
SOAP 字首	預設情況下，如上方的文字窗格所示，Studio 會在您建立的 XPath 運算式加上此運算式做為前綴：env:Envelope/env:Body。若要移除字首，請取消選取此選項。

3. 如果需要，您可以在上方的文字窗格中手動編輯 XPath 運算式。範例：您可能想要使用謂詞來篩選 XPath 傳回的節點集。

概念：Workday Studio 訊息建置器

Workday Studio 的 訊息建置器 提供了圖形檢視，可簡化以下程序集元素的訊息內容定義：

- 寫入 步驟
- 記錄 步驟
- 記錄錯誤 錯誤處理程式

集合編輯器中的「訊息建置器」分頁提供了一個位置，您可以在其中定義集合中任何相關元素的訊息內容。訊息建置器 也可在個別 寫入、日誌和 記錄錯誤 元素的「屬性」檢視中以分頁的形式使用。

左側的 訊息建置器 窗格可讓您定義訊息內容類型。使用頂端的選項新增各種類型的訊息元素：

選項	說明
建立「message-content」	將中介訊息的內容新增至訊息。
建立「文字」	將字串常值新增至訊息。有 3 個選項： • 手動輸入文字，然後按一下 套用變更。 • 按一下 插入 MVEL 運算式，輸入 MVEL 運算式 (如有需要，可搭配純文字)，然後按一下 套用。 • 按一下 產生 XML 文字，然後選取定義的結構描述或 WSDL 元素。
建立「例外」	如果中介環境定義中存在例外，則在訊息中新增一個例外。僅用於 記錄錯誤 錯誤處理程式。
建立 'line-separator'	在訊息中新增換行字元。
建立「XPath」	將 XPath 運算式的結果新增至訊息。
建立「xslt」	將 XSL 轉換的結果新增至訊息。
建立「xml-流」	建立 記錄 或 寫入 步驟的內容。
建立「regex-match」	將規則運算式套用至訊息內容，並將所有相符的實例新增至中介訊息。
建立「靜態檔案」	指定一個輸入檔案，此檔案會填入訊息的完整內容。
Create 'message-store-id'	新增訊息的訊息儲存區 ID。只有當您設定 store-message 傳輸的屬性
建立「標頭」	將具名訊息標頭的值新增至訊息。
建立「標頭」	以名稱/值標頭組合的形式新增完整的標頭集，並以換行字元分隔。
建立 'customer-id'	新增以下帳戶中的客戶 ID：MediationContext 至訊息。若未設定客戶 ID，則會在訊息中顯示「空值」。

您可以在右側窗格中定義訊息元素的值。值窗格的內容因訊息元素類型而異。

若要重新排序或刪除訊息元素，請使用內容窗格右側的按鈕。其在 訊息建置器 中的順序會反映在 XML 中。

參照：訊息元素屬性

使用 訊息建置器 頂端的按鈕，將訊息元素新增至「寫」、「記錄」和「記錄錯誤」元素。

並非每個訊息元素都有屬性。

message-content

屬性	XML 屬性內容名稱	說明
限制	limit	訊息可包含的字元數上限。預設值零表示沒有字元數限制。

例外

屬性	XML 屬性內容名稱	說明
逸出 Xml	escape-xml	允許對 寫入 步驟中的例外情況進行 XML 轉義。

xPath

屬性	XML 屬性內容名稱	說明
名稱空間	namespaces	可讓您指定 XPath 運算式所需的任何名稱空間，格式為 <prefix1> <namespace1> <prefix2> <namespace2>。
XPath	xpath	用於從傳入訊息中擷取值的 XPath 運算式。
XPath 版本	xpath-version	Workday Studio 支援 XPath 1.0 和 2.0 版。對於 Workday 12 之後的程序集版本，預設值為 2，在較早版本中，預設值為 1。

xslt

屬性	XML 屬性內容名稱	說明
允許產生撒克遜位元碼		指定是否允許產生撒克遜位元碼。預設值為 false。
重複使用策略		指定 XSLT 轉換器重複使用策略： 預設值：如果您指定 url 屬性使用 mctx URL 通訊協定或 MVEL 運算式，則預設行為為 no-reuse。否則為 transformer-pool。 註：當您將 Transformer-factory 設為撒克遜

屬性	XML 屬性內容名稱	說明
		文時，預設行為為 templates。 - no-reuse：Workday 不會重複使用轉換器物件。 - 範本：Workday 會建置一個「範本」物件，並從中建立新的轉換器物件。這些轉換器不會重複使用。 - 轉換器池：Workday 使用轉換器池。如果轉換器物件會快取值，則不建議使用。
安全處理		指定轉換器是否僅執行 XSLT 標準處理。設為 True 可對 XML 結構施加限制。範例：忽略 JavaScript 或 Java 延伸處理。設為 False 可依據 XML 規範處理 XML。
Transformer 工廠		的類別名稱 TransformerFactory 您要使用的。預設值為 JAXP 工廠。還有 3 個其他選項： - net.sf.saxon.TransformerFactory 為撒克遜文。 - org.apache.xalan.processor.Tr 適用於 Xalan。 - net.sf.joost.trax.TransformerF 為 Joost。
URL		XSLT 檔案位置，可以是相對路徑或絕對 URL。支援 MVEL 運算式範本。 您可以使用 mctx (中介環境定義) URL 通訊協定直接從訊息內容載入 XSLT。範例：若要載入附件中包含的 XSLT 檔案，請使用 mctx:parts/1。

xml-stream

屬性	XML 屬性內容名稱	說明
名稱空間	namespaces	可讓您指定 XPath 運算式所需的任何名稱空間，格式為 <prefix1> <namespace1> <prefix2> <namespace2>。
XPath	xpath	擷取 XML 元素時，要套用到訊息的 XPath 運算式。Workday 使用 Streaming API for XML (StAX) 處理。此資料庫會讀取 XML 訊息，並將其轉換為事

屬性	XML 屬性內容名稱	說明
		件，而不是 DOM。其本身不會執行任何轉換。 XPath 運算式與您在「」中使用的簡化 XPath 子集相同。xml-stream-splitter 策略 Workday 支援限定元素或本地元素名稱和星號。不支援完整的 XPath 標準。 範例： • /order/item, /ns:order/ns:order • //item, //ns:item, /ns:order/item • /*/*, //* /*/*
XPath 版本	xpath-version	Workday Studio 支援 XPath 1.0 和 2.0 版。對於 Workday 12 之後的程序集版本，預設值為 2，在較早版本中，預設值為 1。
限制	limit	依據 XPath 運算式限制要比對的 XML 片段數。主要用於效能方面的考量。如果您知道只有 1 個符合的 XML 片段，請將限制設為 1 以避免在擷取結果後對文件進行提取剖析。預設值為 -1，這表示結果集不受限制。

regex-match

屬性	XML 屬性內容名稱	說明
分隔符號	delimiter	指定用於分隔 2 個以上實例的字串。預設值為一個空格。
限制	limit	指定要包含在結果集中的出現次數。預設值為零，表示所有出現次數。
規則運算式	regex	定義比對運算式。

靜態檔案

屬性	XML 屬性內容名稱	說明
字元編碼	character-encoding	指定字元編碼。預設值為 UTF-8。
輸入檔案	input-file	指定檔案。

標頭

屬性	XML 屬性內容名稱	說明
包含名稱	include-name	指定 Studio 是否插入標頭名稱。
名稱	name	標頭名稱。
未設定值	not-set-value	找不到標頭或為空白時發出的字串。

傳輸系結

概念：傳輸系結

傳輸綁定是可選的子元素，您可以在 Studio 中對轉出進行配置。傳輸系結可讓傳出傳輸以指定格式支援 Web 服務。

範例：當您想要呼叫 RESTful Web 服務時，可以在 http-out 傳輸上配置 REST-Binding 子元素。

您可以在以下轉出傳輸上配置傳輸系結：

- http-out
- ftp-out
- ftps-out
- sftp-out
- xmpp-out
- custom-out
- 電子郵件寄出

Studio 支援以下傳輸系結：

傳輸系結	說明	附註
REST-Binding	可讓您使用 HTTP 要求方法來呼叫 RESTful Web 服務。	支援以下 HTTP 方法： • GET • PUT • POST
wsdl-http-binding	可讓您使用 HTTP 要求方法和 WSDL 檔案來呼叫 Web 服務。	支援以下 HTTP 方法： • GET • PUT • POST
wsdl-soap-binding	可讓您使用 HTTP 要求方法來叫用 SOAP 型 Web 服務。	支援以下 HTTP 方法： • GET • PUT • POST 支援以下子系元素 • custom-Policy • ws-rm • ws-security • ws-ut

參照：WSDL-HTTP-Binding 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
端點名稱	endpoint-name	wsdl-http-binding 元素用來存取 Web 服務的端點。
介面名稱	interface-name	該 portType Web 服務的 (或介面) 名稱。
服務名稱	service-name	Web 服務的名稱。
WSDL 檔案	wsdl-name	說明 Web 服務的 WSDL 檔案。
序列化參數	serializeParams	目前未使用。

參照：WSDL-SOAP-Binding 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
端點名稱	endpoint-name	wsdl-soap-bound 元素用來存取 Web 服務的端點。
介面名稱	interface-name	該 portType Web 服務的 (或介面) 名稱。
服務名稱	service-name	Web 服務的名稱。
WSDL 檔案	wsdl-name	說明 Web 服務的 WSDL 檔案。
故障處理	fault-handling	指定 HTTP 500 回應代碼的 wsdl-soap-bound 元素錯誤處理流程。選項包括： • 忽略 • 錯誤 • 中止

WSDL-SOAP 系結子元素**概念：WSDL-SOAP-Binding 子元素**

wsdl-soap-bound 傳輸系結支援下列子系元素：

子元素	說明
custom-Policy	支援外傳的自訂 Web 服務安全原則。
ws-rm	支援傳出的訊息傳送安全原則。
ws-security	實作本機檔案，其中包含傳出傳輸中 SOAP 要求的 Web 服務安全原則。
ws-ut	啟用 Workday 整合伺服器，在程序集的傳出傳輸中擷取、驗證及傳播使用者名稱和密碼安全憑證。

參照：自訂原則屬性

通用分頁

屬性	XML 屬性內容名稱	說明
方法	method	對您的自訂安全原則呼叫的方法。
春季 bean	ref	實作您的自訂安全性原則的自訂 Spring bean。

參照：WS-Rm 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
檔案	file	說明訊息安全原則配置的檔案路徑。

參照：WS-Security 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
檔案	file	說明 Web 服務安全性配置的檔案路徑。

參照：WS-Ut 屬性

通用分頁

屬性	XML 屬性內容名稱	說明
使用者名稱	username	將使用者名稱插入 中出站 SOAP 訊息的安全標頭中：MediationContext。
密碼	password	將密碼插入 中出站 SOAP 訊息的安全標頭中：MediationContext。
密碼摘要演算法	password-digest-algorithm	將雜湊演算法套用至「」中傳出 SOAP 訊息安全標頭中的密碼 MediationContext。選項包括： • SHA-1 • SHA-256 • SHA-384 • SHA-512 • 無
Spring Bean	ref	一個選用的 Spring bean，用於驗證 中傳出 SOAP 訊息的安全

屬性	XML 屬性內容名稱	說明
		性標頭中的使用者名稱和密碼值。MediationContext。
Created	created	在「」中的出站 SOAP 訊息的安全標頭中建立時間戳記元素 MediationContext。
隨機數	nonce	在「」中的出站 SOAP 訊息的安全標頭中建立隨機數元素 MediationContext。

結構描述資源管理器

顯示 WSDL 和 XSD 檔案

執行這項作業的原因和時機

結構描述資源管理器 檢視以唯讀格式顯示 WSDL 和 XSD 檔案。您可以將這些檔案中的內容複製到剪貼簿，以便在 Studio 的其他文字編輯器檢視中重複使用這些資料。

程序

1. 選取「視窗 > - 顯示檢視 > 結構探索器」。
2. 在「架構探索器」檢視中，按一下 新增 WSDL 或 XSD。
3. 選取 WSDL 或 XSD 檔案。
4. 在「結構描述資源管理器」檢視中，展開 WSDL 或 XSD 檔案以檢視其結構。

結果

Studio 會在 結構描述探索者 檢視中顯示 WSDL 或 XSD 檔案。

下一步

將 WSDL 或 XSD 檔案中的內容複製到剪貼簿。

顯示 Workday Web 服務 WSDL

執行這項作業的原因和時機

結構描述資源管理器 檢視以唯讀格式顯示 WSDL 和 XSD 檔案。您可以將這些檔案中的內容複製到剪貼簿，以便在 Studio 的其他文字編輯器檢視中重複使用這些資料。

程序

1. 選取「視窗 > - 顯示檢視 > 結構探索器」。
2. 在「架構探索器」檢視中，按一下 新增 Workday Web 服務。
3. 選取並展開 Workday Web 服務。
4. 選取 wsdl。
5. 在 Schema Explorer 檢視中，展開 Workday Web 服務 WSDL 以檢視其結構。

結果

Studio 會在 Schema Explorer 檢視中顯示 Workday Web 服務 WSDL。

下一步

將 Workday Web 服務 WSDL 中的內容複製到剪貼簿。

顯示自訂 RaaS 報告結構描述

執行這項作業的原因和時機

「結構描述資源管理器」檢視以唯讀格式顯示 WSDL 和結構描述 (XSD) 檔案。您可以將這些檔案中的內容複製到剪貼簿，以便在 Studio 的其他文字編輯器檢視中重複使用這些資料。

程序

1. 選取「視窗 > - 顯示檢視 > 結構探索器」。
2. 在「結構探索器」檢視中，按一下「新增自訂 Raas 報告結構描述」。
3. 在「新增自訂 RaaS 報告」視窗中，請參考以下說明：

選項	敘述
環境	包含自訂報告的 Workday 環境
報告名稱	自訂報告的名稱。

4. 在「選取報告」視窗中，選取自訂報告。
5. 按一下 完成。
6. 在「結構探索器」檢視中，展開自訂報告結構描述以檢視其結構。

結果

Studio 會在 結構描述探索器 檢視中顯示自訂報告結構描述。

下一步

將自訂報告結構描述中的內容複製到剪貼簿。

概念：結構描述探索器

結構描述資源管理器 檢視以唯讀格式顯示 WSDL 和 XSD 檔案。您可以將這些檔案中的內容複製到剪貼簿，以便在 Studio 的其他文字編輯器檢視中重複使用這些資料。

您可以在 類型篩選條件文字 欄位中輸入關鍵字，篩選「結構描述探索者」檢視的內容。

預設情況下，「結構描述探索者」檢視會顯示在畫面的右上角。您可以在 Studio 使用者介面上，將 結構描述探索者 檢視拖放到您偏好的位置。

MVEL

概念：MVEL 基礎知識

語法

MVEL 是 Java 型應用程式的一種運算式語言。其語法與 Java 類似，但新增了一些功能，可提高比對內容的效率。

注意：

- 中小型虛擬教學項目使用 Java 名稱空間和類別，但兩者本身都無法聲明。
- 與 Java 不同的是，MVEL 是動態類型的，具有可選的類型。

- 中部實際值水準運算式可以是單行陳述式，也可以是多行指令碼。若為多行指令碼，請使用分號分隔每個陳述式。
- 最小平均價值支出 (MVEL) 依據「最後輸出價值」的原則運作。它支援 return 關鍵字，但通常不需要。
- 最佳作業作業限位 (mover) 支援運算子優先順序規則，並使用括弧來控制執行順序。
- 您可以將 MVEL 字串常值放在單引號或雙引號中。
- 分號是 MVEL 中的保留字元。無法在運算式中使用。使用 (char) 59。範例：
props['string']='a;b' 引發錯誤。此代碼不會：

```
props['string']= (char) 59
props['string_new']='a'props['string']'b'
```

MVEL 運算式可包含：

- 布林運算式
- 方法呼叫
- 屬性運算式
- 變數指派

簡單 MVEL 屬性運算式的範例：

```
context.errorMessage
```

其會擷取屬性 errorMessage 從變數或環境定義物件 context。

MVEL 支援所有常見的 Java 比較運算子、數學運算子和邏輯運算子。下表匯總了其他重要的 MVEL 運算子：

運算子	說明	範例
new	實例化物件。	new String("foo")
with	在單一物件實例上執行多個作業。	with (value) { name = 'Foo', age = 50, sex = Sex.MALE }
assert	判斷值為 True 或執行失敗並產生 AssertionError。	assert foo != null
contains	判斷左側的值是否包含右側的值。	var contains "Foo"
is 或 instance of	判斷左側的值是否是右側類別的成員。	var instanceof String
strsim	比較字串並傳回以百分比表示的相似度。	"foobol" strsim "foobar"
soundslike	在 2 個字串之間執行 roundex 比較。	"foobar" soundslike "fubar"
&	執行位元 AND 運算。	foo & 5
	執行位元 OR 運算。	foo 5
^	執行位元 XOR 運算。	foo ^ 5
+	用於串連 2 個字串的重載運算子。	"foo" + "bar"
#	以字串的形式串連 2 個常值。	8 # 9

運算子	說明	範例
<code>in</code>	檢查集合內的物件。	<code>(foo in list)</code>
<code>=</code>	將值指派給變數。	<code>var = "foobar"</code>

值測試

在 Java 中，運算子 `==` 檢查參照是否指向同一個物件。但是，在 MVEL 中，比較運算子 `==` 檢查值是否相等。MVEL 運算式 `foo == 'bar'` 相當於 Java 的 `foo.equals("bar")`。

MVEL 布林運算式的範例：

```
user.name == 'John Smith'
```

傳回 `true` 如果 `name` 變數或環境定義物件的屬性 `user` 是 John Smith。

MVEL 使用類型強制轉換。如果您比較 2 個不同類型的值，則 MVEL 會嘗試將右側的值強制轉換為左側的類型。如果無法這麼做，系統會嘗試將左側的值強制轉換為右側的類型。範例：

```
"123" == 123;
```

此運算式傳回 `true` 因為 MVEL 會在執行比較前，將右側的值強制轉換為字串。

清單、對應和陣列

請在方括號內加上最小化作業最低薪資清單。使用逗號分隔項目。範例：

```
["Paris", "London", "Berlin"]
```

請在方括號中加上 MVEL 對應。使用冒號分隔索引鍵和值。範例：

```
["Foo" : "Bar", "Bar" : "Foo"]
```

請將 MVEL 陣列放在大括號中。使用逗號分隔項目。範例：

```
{"Bill", "Bob", "Ben"}
```

屬性瀏覽

MVEL 提供了單一、統一的語法，可用來存取屬性、靜態欄位、清單、對應和陣列。

在 Java 中，您可以使用以下陳述式從物件存取屬性：

```
context.getException().getCause()
```

在 MVEL 中，您可以使用以下運算式來存取相同的屬性：

```
context.exception.cause
```

以相同方式存取清單和陣列。範例：

```
parts[2]
```

存取對應圖的方式與存取陣列的方式相同，但可以將任何物件做為索引值傳遞。範例：

```
props["foobar"]
```

如果對應使用字串做為索引鍵，您可以將該對應本身視為虛擬物件。範例：

```
props.foobar
```

出於編入索引的目的，MVEL 會將字串視為陣列。

此範例傳回 B：

```
foo = "Berlin";  
foo[0];
```

指派

您可以在 MVEL 運算式中指派變數，以便在運算式中使用或在執行階段擷取。

由於 MVEL 是動態類型語言，因此您在聲明新變數時不需要指定類型。但您可以這麼做。

以下兩個範例在 MVEL 中皆適用：

```
str = "Some string";  
String str = "Some string";
```

將值指派給類型化變數時，MVEL 會嘗試執行自動類型轉換。範例：

```
String num = 1;
```

您可以將動態類型變數的值轉換為另一種類型。範例：

```
String num = 1;
```

區段：輸入和強制轉換

中虛擬實際語言學習清單使用動態類型，以及選用的靜態類型。本機 Java 物件是靜態類型的。為與這些元素互動，MVEL 有時必須執行類型強制轉換，這可能會影響效能。

中部實際值清單 (MVEL) 會自動將類型強制轉換為最適合目前值的類型。這表示您在處理數字時必須謹慎。浮點運算中可能會發生錯誤，因為 MVEL 會自動放棄無限精度 `java.math.BigDecimal` 至 `java.lang.Double` 和 `java.lang.Long`。當儲存值的類型不是您預期的類型時，這可能會導致問題。一般而言，您應避免對 MVEL 中的財務值執行非整數算術。

MVEL 的類型強制轉換錶表示您可以在不調整輸入的情況下呼叫 Java 方法。解釋器或編譯器會分析您要傳遞給方法的類型，並決定需要何種強制轉換。

如果陣列類型發生衝突，則 MVEL 會嘗試將整個陣列強制轉換為所需的輸入類型。

MVEL 會對靜態類型變數執行任何必要的強制轉換，前提是指派的值是可強制轉換的。如果不是，則會發生例外狀況。

概念：MVEL 和 Studio

在 Workday Studio 中，透過 MVEL 功能，您可以檢查和操縱集合中的訊息部分和元件參數。您可以使用 MVEL 來：

- 從訊息中擷取資訊。
- 設定數字和日期的格式
- 取代字串中的特定字元。

- 擷取整合屬性內容或啟動參數。

您應避免使用 MVEL 來處理大量的資料字串。範例：將訊息轉換為字串可能會產生效能和可擴充性問題。

在支援 MVEL 的欄位中，您可以按 Ctrl+Space 鍵，顯示內容輔助快顯視窗。

中部實際值水準運算式可以是單行陳述式，也可以是多行指令碼。若為多行文字，請使用分號分隔每一行。

註：Studio 會將多行運算式視為一個整體進行評估，而不是依序進行。

具有 MVEL 功能的欄位會展開以容納新的明細。顯示 6 行後即停止展開並變成可捲動的狀態。Studio 會將語法填色套用至 MVEL 運算式。

執行時程中的運算子優先順序 (MVEL) 會實作運算子優先順序。您可以使用方括號來控制 MVEL 運算式的執行順序。

註：由於這些資料是在記憶體中處理，因此 XPath 方法有 100 萬字元的限制，例如 xpath, xpathB, 和 xpathF。如果您的查詢超出此限制，您將不會收到錯誤訊息。您應使用串流 API，例如 寫入 步驟的 xml-stream 元素 (如果您需要處理大型訊息)

參照：Studio Helper 物件

Studio 提供了許多 Helper 物件，讓您能夠使用 MVEL 來存取整合的執行階段屬性。您可以使用 Helper 物件對執行階段值進行簡單變更，或將這些值擷取並傳遞至集中的元件。

小幫手物件	目的
context	提供以下權限：存取 MediationContext 物件 - 即整合到執行階段的動態整合。該 MediationContext 物件連結至目前訊息、中介屬性、中介變數和目前錯誤資訊。 範例：此運算式擷取 屬性值 myProperty 自 MediationContext： <pre>context.getProperty('myProperty')</pre> 範例：此運算式擷取「」屬性 someProperty 自 MediationContext 如果屬性值設為 someValue： <pre>context.getProperty('someProperty').equals('someValue')</pre>
da	Document Accessor Helper 物件。提供以下權限：存取此整合執行前附加至整合事件的文件清單。有效地將 Get Event Documents Web 服務呼叫與擷取 Web 服務回應中參照的文件相結合。 該 da 小幫手物件有一些限制： • 必須明確包含處理可用附件數量超出預期的邏輯。 • 無法存取附件的檔案類型。 • 無法存取附件的字元編碼。 文件迭代 器路由策略會自動處理非預期的附件數量，而且它可以存取 MIME 類型和字元編碼。對於整合而言，使用該策略 (而不是此 Helper 物件) 來擷取輸入文件通常更為適當。

小幫手物件	目的
	範例：此運算式會從檔案名為 eib_transform_output.txt 的整合事件中擷取文件，並新增一個名為「」的變數。var1： <pre>da.toVar('eib_transform_output.txt', 'var1')</pre> 範例：以下運算式會擷取具有指定標籤的文件： <pre>da.getDocsMatching('label')</pre>
ftp	提供以下權限：存取向 ftp、ftps 或 sftp 伺服器發出 LIST 要求時傳回的檔案清單。 範例：以下運算式會擷取 URL 的檔案清單： <pre>ftp.list('URL')</pre>
intsys	提供對 Get Integration Systems Web 服務呼叫傳回的整合系統配置的便捷高效能存取。 範例：此運算式擷取名為「」的整合屬性內容 username： <pre>intsys.getAttribute('username')</pre> 範例：如果具有指定名稱的整合服務存在，則此運算式顯示為 True： <pre>intsys.isServiceEnabled('name')</pre> 註：如果您使用 intsys 若要存取屬性內容，您必須先在 workday-in 整合系統的傳輸
lp	提供以下權限：存取整合啟動事件中的啟動參數和其他資訊。 範例：以下運算式擷取具有指定名稱的啟動參數值： <pre>lp.getSimpleData('name')</pre> 範例：如果具有指定標籤的參數存在，則此運算式顯示為 True： <pre>lp.exists('label') == true</pre> 註：使用前 lp 變數，您需要將啟動參數 XML 指派給變數 wd.launchparameters。
message	提供以下權限：存取目前訊息及其標頭資訊。使用此物件為訊息指派新內容。 範例：此運算式會從以下位置以文字的形式擷取訊息的詞根部分：MediationMessage 介面： <pre>message.rootPartAsText</pre>

小幫手物件	目的
	範例：此運算式可擷取 MediationContext 變數： <pre>message.variables</pre>
mtable	提供以下權限：存取 MTableAdapter 介面，此介面定義 mtable 實例。 範例：此運算式可將 mtable： <pre>mtable['my_property'].set(0, 'Data_Column', 'my_test_data')</pre>
parts	提供以下權限：存取目前訊息的個別部分。Workday 整合中的大多數訊息都有一個部分，即根部分。使用 parts[0] 以存取根部部分。 範例：此運算式擷取標頭 someHeader 從根部分： <pre>parts[0].getHeader('someHeader')</pre> 範例：此運算式實作 xpathB 評估 XPath 布林運算式的方法： <pre>parts[0].xpathB('/a/b')</pre>
props	提供以下權限：MediationContext 屬性 範例：此運算式擷取 屬性值 myProperty 自 MediationContext： <pre>props.myProperty</pre> 範例：此運算式擷取 屬性值 bean.property 自 MediationContext： <pre>props['bean.property']</pre>
spring	提供以下權限：存取程序集的 Spring 配置中定義的建立和擷取 Bean。 範例：此運算式會擷取 name 名為「」的 Bean 的屬性內容 Fred 在 Spring 環境定義中： <pre>spring.getBean('fred').name</pre> 範例：此運算式存取 Spring 應用程式環境定義： <pre>spring.parent ApplicationContext</pre>
util	一組實用工具函數，用於執行常見操作，例如轉義值以插入 XML 或建立 HTTPS URL。

小幫手物件	目的
	範例：此運算式會檢查 <code>MediationContext</code> 是拆分器元件將產生的最後一條訊息： <pre>props['is.last.record'] = util.isLastMessageInBatch()</pre> 部分 <code>util</code> 函數有嚴重的限制。範例： <code>xpathToCommaDelimString</code> 函數針對指定 XML 字串執行 XPath 運算式，會傳回節點序列的字串值，產生以逗號分隔的清單。然後，您可以使用 MVEL 中的 Java 字串處理函數來拆分和操作逗號分隔的清單。但是，該函數有下列限制： • 僅支援 XPath 1。 • 無法產生 CSV 檔案，因為其無法正確轉義換行符、逗號和引號字元。 • 無法可靠分隔包含逗號的資料。 • 其要求將 XML 做為字串，這鼓勵了不可縮放的做法，例如將目前的環境定義訊息或變數轉換為字串。 註：為確保穩定性和向後兼容性，Workday 強烈建議使用明確的版本化 Web 服務要求。使用 <code>MVELUtilHelper.getLatestWWSVersion()</code> 在整合中加入並非最佳做法，而且可能會導致整合失敗或產生非預期的結果。
vars	提供以下權限： <code>MediationContext</code> 變數 範例：此運算式會擷取變數的長度 <code>myVariable</code>： <pre>vars['myVariable'].length</pre> 範例：此運算式會擷取變數的長度 <code>myVariable</code>： <pre>vars['myVariable'].length</pre>
xmldiff	提供以下權限：以程式方式存取使用 <code>xmldiff</code> 元件進行文件比較的結果。 範例：此運算式會檢查 2 個文件之間是否存在差異： <pre>xmldiff.isDifferent() == true</pre> 範例：以下運算式會擷取 2 個文件之間的差異清單： <pre>xmldiff.changes</pre>

參照：文件存取器 **Helper** 物件

實用函數

名稱	引數	傳回值類型	附註
getData	整數索引	來源	擷取指定從零開始的索引處的文件。與 <code>message.setMessage</code> 輔助函數，以節省記憶體的方式將擷取的資料填入目前的訊息。 範例：以下代碼會使用從第一個附件擷取的 UTF-8 XML 訊息填入目前的訊息： <pre>message.setMessage(da.getData(0, 'xml'))</pre> 如果沒有具有指定索引的整合附件，則會導致錯誤。
	字串檔案名稱	來源	擷取具有指定檔案名稱的文件。與 <code>message.setMessage</code> 輔助函數，以節省記憶體的方式將擷取的資料填入目前的訊息。 範例：以下代碼會在目前的訊息中填入名為 <code>input.xml</code> 的 UTF-8 XML 訊息附件： <pre>message.setMessage(da.getData('input.xml'))</pre> 如果沒有指定檔案名稱的整合附件，則會發生錯誤。
getDocumentIndexes	List<String> 標籤	List<Integer>	傳回一個 Java 整數清單，列出標有所有指定標籤的文件索引。 請注意，如果其中一個標籤為「*」，則表示比對任何標籤，且函數會傳回至少包含 1 個標籤的文件的所有索引。
getDocsMatching	List<String>	List<Source>	擷取標有所有指定標籤的文件。 註：Workday 會將小於 100 KB 的文件儲存在記憶體中，將較大的文

名稱	引數	傳回值類型	附註
			件儲存在磁碟中。請注意執行階段記憶體和磁碟空間限制。
	字串標籤	來源	擷取標有指定標籤的文件。 註：此處的「標籤」是指整合附件標籤。與文件標記同義。
	String[] 標籤	來源	擷取字串陣列中標有所有標籤的文件。
getFileName	整數索引	String	擷取具有指定索引的檔案名稱。
	字串標籤	List<String>	擷取標有指定標籤的所有文件名的 Java 清單。
	List<String> 標籤	List<String>	擷取標有指定標籤的所有文件名的 Java 清單。
	字串標籤，整數索引	String	擷取所有標有指定標籤的文件中的第 n 個檔案名稱。 功能上與以下代碼相同： <pre>da.getFileNames(label).get(</pre> 在大多數情況下，您必須使用 <code>da.getFileNames(label)</code> ，找出相符檔案名的數目後才能使用索引。唯一的例外是您需要第一個 (零索引) 值的有限情況。
	List<String> 標籤，整數索引	String	從所有標有指定標籤的文件中擷取第 n 個檔案名稱。 功能上與以下代碼相同： <pre>da.getFileNames(labels).get</pre> 在大多數情況下，您必須使用 <code>da.getFileNames(labels)</code> ，找出相符檔案名的數目後才能使用索引。唯一的例外是您需要第一

名稱	引數	傳回值類型	附註
			個 (零索引) 值的有限情況。
	String[] 標籤，整數索引		從所有標有指定標籤的文件中擷取第 n 個檔案名稱。 功能上與以下代碼相同： <pre>da.getFileNames(java.util.A</pre> 在大多數情況下，您必須使用 <code>da.getFileNames(labels)</code>，找出相符檔案名的數目後才能使用索引。唯一的例外是您需要第一個 (零索引) 值的有限情況。
hasFile	字串檔案名稱	布林運算子	判斷是否有指定檔案名稱的整合附件。
	字串標籤、字串檔案名稱	布林運算子	判斷是否有指定檔案名稱且標有指定標籤的整合附件。
	List<String> 標籤，字串檔案名稱	布林運算子	判斷是否有具有指定檔案名稱且標有清單中所有標籤的整合附件。
sort	字串順序	無	依檔案名稱將附加的文件排序。有效的順序值為： "FILENAME_ASCENDING" "FILENAME_DESCENDING" 沒有檔案名稱的附件放在最後。
size	無	整數	傳回附加文件的數目。
	List<String> 標籤	整數	傳回標記有所有指定標籤的文件數。
	字串標籤	整數	傳回標記有指定標籤的文件數。
	String[] 標籤	整數	傳回字串陣列中標有所有標籤的文件數。
toVar	整數索引、字串 var_name	MessageVariable	擷取指定索引處的文件，並以節省記憶體的方式放入指定的程序集變數中。

名稱	引數	傳回值類型	附註
			範例： <pre>da.toVar(0, 'myInputDocument')</pre>
	String filename, String var_name	MessageVariable	擷取具有指定檔案名稱的文件，並以節省記憶體的方式將其放入指定的程序集變數中。 範例： <pre>da.toVar('WorkerData.xml',</pre>

參照：啟動參數 **Helper** 物件

有用的屬性

名稱	附註
integrationEventWID	傳回與目前整合啟動相關的整合事件 WID。在進行 Web 服務呼叫時很有用，例如 Get_Event_Documents, Put_Integration_Message，或 Put_Integration_Event。 自動為 wcc://GetEventDocuments、wcc://PutIntegrationMessage 和 wcc://PutIntegrationEvent 通用元件提供預設參數。
integrationSystem	傳回 Integration_System_ID 適用於目前的整合系統。在為事件訂閱或欄位覆寫要求填入 Web 服務呼叫時很有用。
integrationSystemRefWID	傳回目前整合系統的 WID。的完全等價的替代方案 integrationSystem 屬性
parentEventWID	傳回目前啟動事件的上層事件的 WID。
sentOn	傳回採用 ISO-8601 日期和時間格式的整合啟動日期和時間。
sentOnAsDate	以「」的格式傳回整合啟動的日期和時間。 java.util.Date 物件。

實用函數

註：

其中一些函數僅依名稱指定啟動參數。另一些方法則指定啟動參數的名稱和提供該參數的整合服務名稱。當整合系統上配置的多個整合服務中存在同名的啟動參數時，後者非常有用。大多數整合不需要容納在多個服務中重複的啟動參數名稱，因此不太經常需要這些函數。

名稱	引數	傳回值類型	附註
exists	字串名稱	布林運算子	如果具名的啟動參數有非空值，則傳回 True。

名稱	引數	傳回值類型	附註
	String service_name, String name	布林運算子	若指定整合服務中的 具名啟動參數有非空 值，則傳回 True。 僅當整合系統上啟用的 多個整合服務中存在同 名的啟動參數時，建議 使用此版本而非單一引 數版本。
getDate	字串名稱	String	傳回執行階段參數的 字串格式日期值。格 式為 YYYY-MM-DDZ 或 YYYY-MM-DD+/- offset。
	String service_name, String name	String	傳回指定服務的執行階 段參數的字串格式日期 值。格式為 YYYY-MM- DDZ 或 YYYY-MM-DD +/-offset。
getParameterData	字串名稱	org.w3c.dom.Element	傳回 XML 元素 Integration_Runtime_Paramete 中包含「」中的 具名啟動參數 Launch_Integration_Event 啟動時傳送至整合的訊 息。
	String service_name, String name	org.w3c.dom.Element	傳回 XML 元素 Integration_Runtime_Paramete 中包含「」中指定服 務的具名啟動參數 Launch_Integration_Event 啟動時傳送至整合的訊 息。
getReferenceData	字串名稱、字串類型	String	針對具有指定名稱的啟 動參數，傳回指定類型 的參考 ID。 範例：以下代碼會為名 為「正職員工」的啟動 參數傳回正職員工 ID： <pre>lp.getReferenceData('Employ</pre>
getReferenceDataList	字串名稱、字串類型	List<String>	傳回一個 Java 清 單，其中包含多重實例 啟動參數中的零個或多 個參考 ID。
	String service_name、String name、String 類型	List<String>	傳回一個 Java 清 單，其中包含指定服務 的多重實例啟動參數中 的零個或多個參考 ID。

名稱	引數	傳回值類型	附註
getSequencedValue	字串名稱	String	傳回指定名稱的序列產生器的目前值。
	String service_name, String name	String	傳回在指定服務上聲明的指定序列產生器的目前值。

參照：效用 **Helper** 物件

實用函數

名稱	引數	傳回值類型	附註
cleanString	字串值	String	用於對必須在 XML 文件中進行轉義的任何輸入字元進行轉義。 範例： <pre><Value>@{util.cleanString(p root/row/ a_value')}}</ Value></pre>
getExtForContentType	字串 MIME 類型	String	啟用整合功能，根據指定的 MIME 類型取得副檔名，例如 .txt 或 .xml。
getFilenameFromContentDisposition	字串 content disposition	String	從內容捨棄字串擷取檔案名稱。某些程序集元件 (例如文件迭代器策略) 會使用檔案名稱資訊填入內容捨棄訊息標頭。 範例：此代碼會傳回值 testfile.pdf： <pre>util.getFilenameFromContent filename=testfile.pdf')</pre>
readFileToVar	String var_name, String filename		將非文字檔檔案讀入變數。這是唯一可載入非文字文件的預定義方式。請注意，寫入步驟的靜態檔案屬性只能將文字文件載入訊息或變數中。指定的檔案是相對於 WSAR-INF 目錄。 範例：以下代碼會將影像檔案 icon-32.png 從

名稱	引數	傳回值類型	附註
			WSAR-INF 目錄讀取到環境定義變數 v1 中： <pre>util.readFileToVar('v1', 'i</pre>
readFileToVar	字串 var_name、字串檔案名稱、字串 MIME 類型		將非文字檔案讀入您指定 MIME 類型的變數。這是唯一可載入非文字文件的預定義方式。請注意，寫入步驟的靜態檔案屬性只能將文字文件載入訊息或變數中。指定的檔案是相對於 WSAR-INF 目錄。 範例：以下代碼會將影像檔案 icon-32.png 從 WSAR-INF 目錄讀取到 MIME 類型為「image/png」的環境定義變數 v1 中： <pre>util.readFileToVar('v1', 'i png')</pre>

參照：範例 **Studio MVEL** 運算式

變數	範例運算式	說明
context	context.getProperty('myProperty')	擷取屬性值 myProperty 自 MediationContext。
	context.getProperty('someProperty').equals('someValue')	擷取屬性 someProperty 自 MediationContext 如果屬性值設為 someValue。
da	da.toVar('eib_transform_output.txt', 'var1')	從檔案名為 eib_transform_output.txt

變數	範例運算式	說明
		的整合事件中擷取文件，並新增名為「」的變數。var1。
	<code>da.getDocsMatching('label')</code>	擷取具有指定標籤的文件。
ec	<code>ec.getAttribute('attributeName')</code>	擷取屬性內容名稱。
	<code>ec.integrationMapLookup('map', 'key')</code>	擷取整合對應值。
env	<code>env['cc.hostname']</code>	擷取 cc.hostname 環境屬性設定為 -Dcc.hostname 在指令行中。
	<code>env.containsKey('key')</code>	如果環境屬性包含指定的索引鍵，則顯示為 True。
ftp	<code>ftp.list('URL')</code>	擷取 URL 的檔案清單。
intsys	<code>intsys.getAttribute('username')</code>	擷取名為「」的整合屬性內容 username。
	<code>intsys.isServiceEnabled('name')</code>	如果具有指定名稱的整合服務存在，則顯示為 True。
lp	<code>lp.getSimpleData('name')</code>	擷取指定名稱的啟動參數值。
	<code>lp.exists('label') == true</code>	如果具有指定標籤的參數存在，則顯示 True。
message	<code>message.rootPartAsText</code>	從以下位置以文字形式擷取訊息的根目錄部分： MediationMessage 介面
	<code>message.variables</code>	擷取 MediationContext 變數
mtable	<code>mtable['my_property'].set(0, 'Data_Column', 'my_test_data')</code>	將 mtable。
	<code>mtable['my_property'].get(0, 'Data_Column')</code>	從以下項目擷取資料： mtable。
parts	<code>parts[0].getHeader('someHeader')</code>	擷取標頭 someHeader 從根部聲部
	<code>parts[0].xpathB('/a/b')</code>	實作 xpathB 方法來評估 XPath 布林運算式。
props	<code>props.myProperty</code>	擷取屬性值 myProperty 自 MediationContext。
	<code>props['bean.property']</code>	擷取屬性值 bean.property 自 MediationContext。
spring	<code>spring.getBean('fred').name</code>	擷取 name 名為「」的 Bean 的屬性內容 Fred 在 Spring 環境定義中。
	<code>spring.parent</code>	存取 Spring 應用程式環境定義。

參照： **Studio MVEL 自動轉換語法範例**

函數	範例運算式	說明
將字串自動轉換為變數。	<pre>vars['fred'] = '<hello>world</ hello>'</pre>	建立名為「」的新變數 fred 與內容 <hello>world</hello>。 此方法僅能用來建立小型訊息。使用運算式建立大型字串可能會導致效能降低。 在大多數情況下，您應使用 write 建立變數內容的步驟。
請為變數提供內容和 MIME 類型資訊。	<pre>vars['fred'] = ['Hello World', 'text/plain']</pre>	建立名為「」的新變數 fred 與內容 Hello World 和 MIME 類型 text/plain。 此方法僅能用來建立小型訊息。使用運算式建立大型字串可能會導致效能降低。 在大多數情況下，您應使用 write 此步驟可為變數建立內容並配置 MIME 類型。
從附件複製到變數	<pre>vars['fred'] = parts[1]</pre>	建立新變數 fred 從目前訊息的第一個附件開始。 在大多數情況下，您應使用 copy 步驟，從附件取得輸入。
從訊息複製到變數	<pre>vars['fred'] = message</pre>	建立新變數 fred 從整封訊息中刪除 在大多數情況下，您應使用 copy 步驟，從目前訊息中取得輸入。
將內容和 MIME 類型資訊提供給值為運算式的變數。	<pre>vars['fred'] = [23 + 5, 'text/plain']</pre>	建立名為「」的新變數 fred 含內容 28 和 MIME 類型 text/plain。
從訊息根部分中的 XPath 片段建立變數。	<pre>vars['fred'] = parts[0].xpathF('/ dink/donk')</pre>	建立名為「」的新變數 fred 包含目前訊息根部分的 XPath 片段。 此方法僅能用來建立小型訊息。使用 XPath 運算式建立大型字串可能會導致效能降低。
取消變數。	<pre>vars['fred'] = null</pre>	移除名為「」的變數 fred。
將內容指派給根部分。	<pre>parts[0] = '<hello>world</ hello>'</pre>	指派內容 <hello>world</hello> 移至根部聲部。 此方法僅能用來建立小型訊息。使用運算式建立大型字串可能會導致效能降低。

函數	範例運算式	說明
		在大多數情況下，您應使用 write 此步驟可將內容指派給附件並配置 MIME 類型。
將內容指派給第一個附件。	<pre>parts[1] = ['Hello World', 'text/plain']</pre>	指派內容 Hello World MIME 類型為 text/plain 到第一個附件。 此方法僅能用來建立小型訊息。使用運算式建立大型字串可能會導致效能降低。 在大多數情況下，您應使用 write 此步驟可將內容指派給附件並配置 MIME 類型。
將內容指派給根附件，並指定 MIME 類型。	<pre>parts[0] = [23 + 5, 'text/plain']</pre>	指派具有內容的根部件 28 和 MIME 類型 text/plain。
將變數中的內容指派給訊息部分。	<pre>parts[0] = vars['fred']</pre>	指派變數的內容 fred 新增至訊息部分。 在大多數情況下，您應使用 copy 此步驟可從變數取得輸入，並將其指派給訊息部分。
移除根部。	<pre>parts[0] = null</pre>	移除根部分。
複製附件。	<pre>parts[2] = parts[1]</pre>	將第一個附件複製到第二個附件中。 在大多數情況下，您應使用 copy 此步驟可從一個附件取得輸入，並將其指派給另一個附件。
使用預設 MIME 類型建立新訊息。	<pre>context.message = '<hello>world</hello>'</pre>	建立新訊息 <hello>world</hello> 預設 MIME 類型為 text/xml。 在大多數情況下，您應使用 write 步驟以建立新訊息。
建立 MIME 類型為「」的新訊息 text/plain。	<pre>context.message = ['Hello World', 'text/plain']</pre>	建立新訊息 Hello World MIME 類型為 text/plain。 在大多數情況下，您應使用 write 步驟以建立新訊息。
從變數內容建立新訊息。	<pre>context.message = vars['fred']</pre>	建立與變數具有相同內容和 MIME 類型的新訊息 fred。 在大多數情況下，您應使用 copy 此步驟可從變數建立新訊息。
從目前訊息的根目錄部分建立新訊息。	<pre>context.message = parts[0]</pre>	從目前訊息的根目錄部分建立新訊息。

函數	範例運算式	說明
		在大多數情況下，您應使用 copy 從訊息部分建立新訊息的步驟。

參照：實際應用程式計量檢視方法

方法	說明	範例代碼	範例輸出
<code>getClass().getName()</code>	擷取屬性或變數的類型。	<pre>props['p.myProperty'] = java.lang.Boolean props['p.myProperty'].getClass().getName()</pre>	
<code>getEndpoint()</code>	使用 SOAP 要求時擷取 RaaS 報告的端點。	<pre>@{intsys.reportService Alias'}}</pre>	<pre>customreport2/Report <tenant_name>/ <username>/ report_name_in_XML_format</pre>
<code>getExtrapath()</code>	擷取 RaaS 報告的額外路徑。	<pre>intsys.reportService workday-in Report Service Alias>')</pre>	<pre>customreport2/your <tenant_name>/ <username>/ <Custom Report Name></pre>
<code>indexOf()</code>	擷取特定字元在輸入字串內的位置。	<pre>props['input.document'] = 24 = 'FoundationDataCollection/4d196113' props['index.of.separator'] = props['input.document.reference'].indexOf("/")</pre>	
<code>leftPad()</code>	在輸入字串的開頭新增字元。	<pre>org.apache.commons.lang3 10, '0')</pre>	<pre>0000001234 leftPad('1234', 10, '0')</pre>
<code>listToCommaDelimitedString()</code>	將陣列值的清單轉換為字串，並以逗號分隔。	<pre>props[p.workers] = [1234, 7890] props['p.workers'] = lp.getWIDs('Pay Group Members') props['p.workers.ids'] = props['p.workers'] ! = empty ? util.listToCommaDelimitedString(props['p.workers'])</pre>	<pre>1234, 7890</pre>
<code>replaceAll()</code>	取代輸入字串內所有指定字元的實例。	<pre>props['p.employee.id'] = 12345 = '12345pdf_a.pdf'</pre>	

方法	說明	範例代碼	範例輸出
		<pre>props['p.current.id'] = props['p.employee.id'].replaceAll('[^0-9]', '')</pre>	
substring()	擷取輸入字串的一部分。	<pre>props['p.test.string'] = 'test' props['p.test.substring'] = props['p.test.string'].substring(0,2)</pre>	
toUpperCase()	將字串轉換為大寫。	<pre>props['p.test.string'] = 'workday Studio' props['p.test.string'].toUpperCase()</pre>	WORKDAY STUDIO
URLEncoder ()	擷取僅包含有效 URL 字元的字串。	<pre>props['p.report.parameters'] = 'Full_Name=' # 'Logan McNeil' props['REST.URL'] = java.net.URLEncoder.encode(props['p.report.parameters'], 'UTF-8')</pre>	Full_Name=Logan%20McNeil

Studio 整合服務

建立屬性對應服務

開始之前

建立程序集專案。

執行這項作業的原因和時機

屬性內容對應服務可讓您定義整合的屬性內容和對應。屬性內容是定義整合的可配置值。您可以建立以下類型的屬性內容：

- 簡單。
- 參照
- 列舉

對應會調節內部 Workday 值與在整合中定義的外部屬性內容值。您可以建立以下類型的對應圖：

- 簡單。
- 類別報告欄位
- 列舉

程序

1. 在集合編輯器中開啟集合。
2. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
3. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立 attribute-map-service。
4. 在 服務名稱 視窗中，輸入屬性對應服務的名稱。

5. 在 整合屬性內容 分頁上，請參考以下說明：

選項	敘述
名稱	屬性內容的名稱。在所有整合系統中必須是唯一的。
類型	屬性內容類型定義。
新增簡單類型屬性內容	新增使用字串常值的屬性內容類型。
新增參照類型屬性內容	新增使用 Workday 類別報告欄位 (CRF) 和 Web 服務參照的屬性內容類型。
新增列舉類型屬性內容	選取 編輯類型 以配置列舉。包括： - 請輸入「類型名稱」，以使用指向您用戶中現有列舉定義的 列舉參照。 - 請在 列舉定義 分頁中輸入唯一的 類型名稱，並指定允許的值，即可建立新的列舉。 註：當您在「列舉定義」分頁中輸入現有定義的 類型名稱 時，Workday 會覆寫定義。

6. 在 整合對應 分頁上，選取 **新增** 以新增對應。

7. 配置「內部類型」和「外部類型」時，請參考以下說明：

選項	敘述
簡單類型	您可以選取以下簡單類型： - 布林運算子 - 日期 - 日期時間 - 數字 - 文字
類別報告欄位	選取 瀏覽 CRF 。
列舉	配置您的 列舉參照 或 列舉定義 。

結果

Studio 會將屬性地圖服務儲存在整合的 Workday-in 傳輸中。您可以使用 MVEL 從服務存取資料 intsys 運算式。

下一步

部署您的整合。

[relatedlinks](#)

[相關概念](#)

[概念：列舉](#)

[相關任務](#)

[將整合部署至 Workday 頁 247](#)

[在福利整合中設定屬性和對應](#)

建立交付服務

開始之前

建立程序集專案。

執行這項作業的原因和時機

交付服務可讓您將文件從 Workday 用戶傳送至目標端點。

程序

1. 在集合編輯器中開啟集合。
2. 在 調色盤上，展開 傳輸中 分頁。
3. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
4. 在集合編輯器中，按一下 Workday-in 傳輸方式。
5. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立delivery-service。
6. 在「服務名稱」視窗中，輸入交付服務的名稱。
7. 按一下儲存。

結果

Studio 會將傳送服務儲存在整合的 Workday-in 傳輸中。

下一步

在 Workday 的整合系統上部署整合併配置發送服務。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[設定整合交付](#)

建立偵聽器服務

開始之前

建立程序集專案。

執行這項作業的原因和時機

偵聽器服務可讓您的整合透過 Workday 用戶接收第三方用戶端的啟動訊息。

程序

1. 在集合編輯器中開啟集合。
2. 在 調色盤上，展開 傳輸中 分頁。
3. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
4. 在集合編輯器中，按一下 Workday-in 傳輸方式。
5. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立偵聽器服務。
6. 在「服務名稱」視窗中，輸入偵聽器服務的名稱。
7. 按一下儲存。

結果

Studio 會將偵聽器服務儲存在整合的 Workday-in 傳輸中。

下一步

在 Workday 中為您的整合系統部署整合併配置偵聽器服務。

註：只有在偵聽器服務沒有啟動參數的情況下，您才能部署使用偵聽器服務的整合。

relatedlinks

相關概念

[概念：偵聽器服務 API](#)

相關任務

[將整合部署至 Workday 頁 247](#)

建立擷取服務

開始之前

建立程序集專案。

執行這項作業的原因和時機

擷取服務可讓您的整合功能從遠端用戶端擷取文件，並將其儲存在您的 Workday 用戶中。

程序

1. 在集合編輯器中開啟集合。
2. 在 調色盤上，展開 傳輸中 分頁。
3. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
4. 在集合編輯器中，按一下 Workday-in 傳輸方式。
5. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立擷取服務。
6. 在「服務名稱」視窗中，輸入擷取服務的名稱。
7. 按一下儲存。

結果

Studio 會將擷取服務儲存在整合的 Workday-in 傳輸中。

下一步

在 Workday 的整合系統上部署整合併配置擷取服務。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[設定整合擷取](#)

建立序列產生器服務

開始之前

建立程序集專案。

執行這項作業的原因和時機

序列產生器服務可讓您在每次執行整合時，為文件產生唯一且已排序的檔案名稱。在 Studio 中，您可以在 Workday-in 傳輸上配置序列產生器服務。在啟動整合之前，您還必須在 Workday 用戶中配置序列產生器服務。

程序

1. 在集合編輯器中開啟集合。
2. 在集合編輯器中，按一下 Workday-in 傳輸方式。
3. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立序列產生器服務。
4. 在「服務名稱」視窗中，輸入序列產生器服務的名稱。
5. 在序列產生器服務的「屬性」檢視中，按一下 新增 以新增序列器。
6. 在 序列器 欄位上，輸入序列器的名稱。
7. 在「Project Explorer」檢視中，在您的組合專案上按一下滑鼠右鍵，然後選取「部署至 Workday」。
8. 在 Cloud Explorer 檢視中，導覽至您部署的程序集專案。
9. 在已部署的組合專案的整合節點上按一下滑鼠右鍵。
10. 選取 Workday：整合系統。
11. 在 Workday 分頁上，請參考以下說明：

選項	敘述
使用者名稱	您的 Workday 用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

12. 在 Workday 用戶的「自訂整合服務」區段的「整合 服務」欄中，找到序列產生器服務的名稱，然後選取「相關動作」和「預覽」。
13. 選取 整合序列產生器服務 > ，配置整合序列產生器。
14. 在 序列 ID 增量依據 欄位中，輸入值 1。
15. 在 序列 ID 格式 欄位中，輸入值 file[Seq].csv。
16. 按一下確定。
17. 按一下完成。

結果

Studio 會將序列產生器服務儲存在整合的 Workday-in 傳輸中。Workday 會將序列產生器設定儲存在您的整合系統中。

下一步

啟動您的整合。

relatedlinks

相關任務

[從 Studio 啟動整合](#) 頁 252

[設定整合序列產生器](#)

建立異動記錄服務

開始之前

建立程序集專案。

執行這項作業的原因和時機

交易記錄服務可讓您顯示您的整合已在 Workday 用戶中訂閱的交易事件記錄。

程序

1. 在集合編輯器中開啟集合。
2. 在 調色盤上，展開 傳輸中 分頁。
3. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
4. 在集合編輯器中，按一下 Workday-in 傳輸方式。
5. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立交易記錄服務。
6. 在「服務名稱」視窗中，輸入交易記錄服務的名稱。
7. 按一下儲存。

結果

Studio 會將異動記錄服務儲存在整合的 Workday-in 傳輸中。

下一步

在 Workday 的整合系統上部署整合併配置交易記錄服務。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[步驟：設定目錄服務整合](#)

建立 Report Services

開始之前

建立程序集專案。

執行這項作業的原因和時機

報告服務會為您在整合中參照的任何自訂報告指定別名。

程序

1. 在集合編輯器中開啟集合。
2. 在 調色盤上，展開 傳輸中 分頁。
3. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
4. 在集合編輯器中，按一下 Workday-in 傳輸方式。
5. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立報告服務。
6. 在「服務名稱」視窗中，輸入報告服務的名稱。
7. 在報告服務的 屬性 檢視中，按一下 新增項目 以新增別名。執行此任務時，請參考以下說明：

選項	敘述
別名	自訂報告的別名。
說明	自訂報告的文字說明。
報告參照	自訂報告在 Workday 用戶中顯示的名稱。

8. 按一下 選取 RaaS 報告。執行此任務時，請參考以下說明：

選項	敘述
環境	包含自訂報告的 Workday 環境
報告名稱	自訂報告的名稱。

9. 按一下下一步。
 10. 選取您的自訂報告。
 11. 按一下 完成。
 12. 按一下儲存。

結果

Studio 會將報告服務儲存在整合的 Workday-in 傳輸中。

下一步

在 Workday 中為您的整合系統部署整合併配置報告服務。

relatedlinks

相關概念

[概念：自訂報告](#)

相關任務

[將整合部署至 Workday 頁 247](#)

建立自訂物件服務

開始之前

建立程序集專案。

執行這項作業的原因和時機

自訂物件服務會為您在整合中參照的任何自訂物件指定別名。

程序

1. 在集合編輯器中開啟集合。
2. 在 調色盤上，展開 傳輸中 分頁。
3. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
4. 在集合編輯器中，按一下 Workday-in 傳輸方式。
5. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立 custom-object-service。
6. 在「服務名稱」視窗中，輸入自訂物件服務的名稱。
7. 在自訂物件服務的「屬性」檢視中，按一下 新增項目 以新增別名。執行此任務時，請參考以下說明：

選項	敘述
別名	自訂物件的別名。
說明	自訂物件的文字說明。
自訂物件參照	自訂物件在 Workday 用戶中顯示的名稱。

8. 按一下 選取自訂物件。
9. 在「選取自訂物件」視窗中，指定您的 Workday 用戶環境並瀏覽至您的自訂物件。
10. 按一下確定。

11. 按一下儲存。

結果

Studio 會在整合的 Workday-in 傳輸中儲存自訂物件服務。

下一步

在 Workday 的整合系統上部署整合併配置自訂物件服務。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[設定整合自訂物件](#)

建立資料起始服務

開始之前

建立程序集專案。

執行這項作業的原因和時機

資料初始化服務可讓您在 Studio 整合中更快速、更有效率地存取 Workday 資料。

程序

1. 在集合編輯器中開啟集合。
2. 在 調色盤上，展開 傳輸中 分頁。
3. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
4. 在集合編輯器中，按一下 Workday-in 傳輸方式。
5. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立資料初始化服務。
6. 在「服務名稱」視窗中，輸入資料初始化服務的名稱。
7. 按一下儲存。

結果

Studio 會將資料初始化服務儲存在整合的 Workday-in 傳輸中。

下一步

在 Workday 中為您的整合系統部署整合併配置資料初始化服務。

建立服務參考

開始之前

建立程序集專案。

執行這項作業的原因和時機

服務參照可讓您參照同一集合內另一個 Workday-in 傳輸中配置的任何整合服務。

程序

1. 在集合編輯器中開啟集合。
2. 在 調色盤上，展開 傳輸中 分頁。

3. 將 Workday-inTransport 從 選用板 拖曳至程序集編輯器。
4. 在集合編輯器中，按一下 Workday-in 傳輸方式。
5. 在 Workday-in 傳輸的「屬性」檢視的「服務」分頁上，按一下 建立 service-reference。
6. 在「服務資歷」視窗中，選取服務資歷。
7. 按一下確定。
8. 按一下儲存。

結果

Studio 會將服務參考儲存在整合的 Workday-in 傳輸中。

下一步

部署您的整合。

[relatedlinks](#)

[相關任務](#)

[將整合部署至 Workday 頁 247](#)

新增對 Workday-Out-Rest 傳輸中自訂報告的參照

開始之前

建立包含 Workday-out-REST 傳輸的集合專案。

執行這項作業的原因和時機

Workday-out-REST 傳輸可讓您將 REST 要求訊息傳送至 HTTP URL。您可以新增對自訂報告的參照，做為 Workday-out-REST 交通上的額外路徑值。

程序

1. 在集合編輯器中開啟集合。
2. 在集合編輯器中，按一下 Workday-out-REST 交通方式。
3. 在 Workday-out-REST 傳輸的「屬性」檢視的「常用」分頁上，按一下 選取 RaaS 報告。
4. 在「新增自訂 RaaS 報告」視窗中，請參考以下說明：

選項	敘述
環境	包含自訂報告的 Workday 環境
報告名稱	自訂報告的名稱。

5. 在「選取報告」視窗中，選取您的自訂報告。
6. 在「報告服務配置」視窗中，請參考以下說明：

選項	敘述
報告別名	自訂報告的別名。
說明	自訂報告的文字說明。

7. 按一下 完成。

結果

Studio 會在整合的 Workday-out-REST 傳輸中新增對自訂報告的參照。

下一步

啟動您的整合。

relatedlinks

相關任務

從 [Studio 啟動整合](#) 頁 252

新增對 **Workday-Out-Rest** 傳輸中自訂物件的參照

開始之前

建立包含 Workday-out-REST 傳輸的集合專案。

執行這項作業的原因和時機

Workday-out-REST 傳輸可讓您將 REST 要求訊息傳送至 HTTP URL。您可以新增對自訂物件的參照，做為 Workday-out-REST 傳輸上的額外路徑值。

程序

1. 在集合編輯器中開啟集合。
2. 在集合編輯器中，按一下 Workday-out-REST 交通方式。
3. 在 Workday-out-REST 傳輸的「屬性」檢視的「常用」分頁上，按一下 選取自訂物件。
4. 在「新增自訂物件」視窗中，請參考以下說明：

選項	敘述
環境	包含自訂物件的 Workday 環境
HTTP 方法	選項包括： • POST • PUT • GET • DELETE

5. 在 選取自訂物件 視窗中，請參考以下說明：

選項	敘述
選取現有的自訂物件別名	顯示現有自訂物件別名的清單。
新建自訂物件別名	可讓您建立新的自訂物件別名。

如果您選取了 PUT、GET 或 DELETE HTTP 方法，請跳至步驟 7。

6. 如果您選取 POST HTTP 方法，則 Studio 會顯示「POST 查詢參數」視窗。執行此任務時，請參考以下說明：

選項	敘述
updateIfExists	更新現有的自訂物件。
大量	建立或更新自訂物件的集合。

7. 如果您選取了 PUT、GET 或 DELETE HTTP 方法，請從「自訂物件 ID 參照選取」視窗選取自訂物件 ID 參照。
8. 按一下 完成。

結果

Studio 會在整合的 Workday-out-REST 傳輸中新增對自訂物件的參照。

下一步

啟動您的整合。

relatedlinks

相關任務

從 [Studio 啟動整合](#) 頁 252

CLAR 檔案

產生載貨清單檔案

執行這項作業的原因和時機

載貨清單檔案可讓您將程序集集合匯出為 CLAR 檔案。您必須將清單檔案儲存至程序集集合中的專案資料夾。載貨清單檔案包含程序集集合的中繼資料。

程序

1. 在 Studio 功能表列上，選取「Workday > 解決方案 > 產生 Workday 清單」。
2. 按兩下清單檔案的目標專案資料夾。

結果

Studio 會將清單檔案儲存至專案資料夾。

下一步

將程序集集合匯出為 CLAR 檔案。

匯入 CLAR 檔案

開始之前

將 CLAR 檔案儲存至您的檔案系統。

執行這項作業的原因和時機

您可以匯入 CLAR 檔案，將集合體新增至工作區。

程序

1. 在 Studio 功能表列上，選取「檔案 > 匯入...」。
2. 選取 CLAR 檔案 匯入精靈。
3. 按一下 瀏覽 並選取要匯入的 CLAR 檔案。
4. 按一下 下一步 ，顯示 CLAR 檔案中的集合和專案。
5. 選取要從 CLAR 檔案匯入的集合和專案。

Studio 會自動選取 CLAR 檔案中的每個集合和專案。使用核取方塊修改您的選擇。Studio 會在有名稱衝突的集合和專案上顯示裝飾器。

6. 編輯所有有衝突的專案名稱。
 - a) 請選取名稱有衝突的專案。
 - b) 按一下 編輯...
 - c) 在「專案名稱」欄位中，將專案重新命名。

7. 請編輯所有有衝突的集合名稱。
 - a) 選取名稱衝突的集合。
 - b) 按一下 編輯...
 - c) 在「集合名稱」欄位中，重新命名集合。
8. 按一下 完成。

結果

Studio 會將 CLAR 檔案的內容匯入至您的工作區。

匯出 CLAR 檔案

開始之前

- 建立程序集專案。
- 產生清單檔案。

執行這項作業的原因和時機

您可以從工作區將組件匯出為 CLAR 檔案。您可以使用 CLAR 檔案與其他 Studio 整合開發人員共用您的程序集。範例：將解決方案上傳至 Workday Community 上的「提供的解決方案」頁面。

程序

1. 在 Studio 功能表列上，選取「檔案 > 匯出...」。
2. 選取 CLAR 檔案 匯出精靈。
3. 在「CLAR 檔案規格」視窗中，選取要匯出為 CLAR 檔案的集合。
4. 在「Workday 清單規格」視窗中，選取「使用集合中現有的清單」。
5. 在「匯出目的地」視窗中，指定 CLAR 檔案的目標目的地。
6. 按一下 完成。

結果

Studio 會將程序集匯出為 CLAR 檔案。

relatedlinks

相關參考資料

[所有提供的解決方案](#)

概念：CLAR 檔案

CLAR 檔案

CLAR (雲端封存) 檔案是儲存程序集集合、專案和中繼資料的封裝檔案。CLAR 檔案可讓您透過工作區匯入和匯出組件。您可以使用 CLAR 檔案與其他 Studio 整合開發人員共用您的程序集。

載貨清單檔案

載貨清單檔案可讓您將程序集集合匯出為 CLAR 檔案。您必須將清單檔案儲存至程序集集合中的專案資料夾。載貨清單檔案包含程序集集合的中繼資料。

提供的解決方案

「提供的解決方案」頁面是一個線上儲存庫，可讓您存取和共用 CLAR 檔案。您可以在 Workday Community 中存取提供的解決方案頁面。

JAXB 專案

建立 JAXB 專案

開始之前

將 XML 結構描述檔案儲存至您的檔案系統。

執行這項作業的原因和時機

JAXB 是一個綁定框架，可讓您在 XML 和 Java 格式之間轉換資料。在 Studio 中，您可以建立 JAXB 專案，將 XML 結構描述檔案轉換為 Java 類別。建立 JAXB 專案後，您可以將其新增為組合件專案中的依存關係，藉此在整合中實作其 Java 類。

程序

1. 選取「> 將新的 > Workday 結構描述轉換為 Java (JAXB) 專案」。
2. 在「專案配置詳細資料」視窗中，請參考以下說明：

選項	敘述
專案名稱	專案的名稱。
目標執行階段	專案的目標執行階段伺服器。自動填入 <i>Workday Runtime</i> 。
配置	專案的多面向配置。自動填入 <i>Workday 結構描述轉 Java (JAXB) 專案</i> 。

3. 在「來源資料夾」視窗中，請參考以下說明：

選項	敘述
世代目錄	JAXB 產生的 Java 類別的目錄資料夾。自動填入為 <i>jaxb-gen</i> 。
將產生的資源標示為衍生的資源	將 JAXB 產生的 Java 類別標示為衍生資源。Studio 會自動選取此核取方塊。

4. 在 JAXB 結構描述選擇和產生設定 視窗中，請參考以下說明：

選項	敘述
結構描述檔案	XML 結構描述檔案的位置。Studio 會顯示以下選項： • 工作區 • 外部 • URL
JAXB 外部系統檔案 (選用)	XML 結構描述的任何系統檔案所在位置。Studio 會顯示以下選項： • 工作區 • 外部
為未對應的結構描述產生預設系統檔案	為您的 XML 結構描述建立通用系統檔案。
專案內的地點	通用系統檔案的目錄資料夾。自動填入 <i>jaxb-resources/bindings.xjb</i> 。
選擇管理結構描述和外部 JAXB 系統檔案的方式	選取 在專案中建立外部結構描述/系統檔案的連結，以在新的 JAXB 專案中建立指向您的結構

選項	敘述
	描述和系結檔案的連結。Studio 會自動選取此選項。 選取 立即將結構描述/系結檔案複製到專案，將您的結構描述和系結檔案複製到新的 JAXB 專案中。
編譯器選項	Studio 會顯示以下選項： • 指定目標封裝 (覆寫所有其他封裝定義) • 對結構描述套用較寬鬆的驗證 • 隱藏封裝層級註解 • 自動名稱衝突解決方案

5. 按一下 完成。

結果

Studio 會在 Project Explorer 檢視中顯示 JAXB 專案。

下一步

將 JAXB 專案新增為組合件專案中的依存關係。

將 JAXB 專案新增為程序集專案中的依存關係

開始之前

- 建立程序集專案。
- 建立 JAXB 專案。

執行這項作業的原因和時機

JAXB 是一個綁定框架，可讓您在 XML 和 Java 格式之間轉換資料。在 Studio 中，您可以在程序集專案中將 JAXB 專案新增為依存關係，藉此在整合中實作 JAXB Java 類別。

程序

1. 在「Project Explorer」檢視中，在組合專案上按一下滑鼠右鍵。
2. 選取 建置路徑 > 配置建置路徑...
3. 在 Java 建置路徑 視窗上，選取 專案 分頁。
4. 按一下 新增...
5. 選取要新增至組合件專案的 JAXB 專案。
6. 按一下確定。
7. 按一下 套用並關閉。

結果

Studio 會將 JAXB 專案新增為組合體專案中的依存關係。您現在可以在該組合件專案內儲存的整合中實作 JAXB Java 類別。

下一步

部署您的整合。

概念：Studio 中的 JAXB 支援

JAXB 是一個綁定框架，可讓您在 XML 和 Java 格式之間轉換資料。

JAXB 專案

在 Studio 中，您可以建立 JAXB 專案，將 XML 結構描述檔案轉換為 Java 類別。建立 JAXB 專案後，您可以將其新增為組件專案中的依存關係，藉此在整合中實作其 Java 類。

JAXB 中介步驟

在組合專案中，您可以在以下中介步驟的「屬性」檢視中調用 JAXB：

- java-to-xml
- xml-to-java

Studio Ivy 外掛程式

配置 Ivy 外掛程式

執行這項作業的原因和時機

Workday Studio 支援可讓您管理專案依存關係的 ApacheIvy™ 外掛程式。您可以在 Studio 中將 Ivy 外掛程式配置為喜好設定。

程序

1. 按一下 視窗 > 喜好設定 > Workday > Ivy。
2. 在「配置 Ivy 引擎」視窗中，請參考以下說明：

選項	敘述
ivy 設定 URL	ivysettings.xml 檔案的 URL。
常春藤屬性	ivyproperties.xml 檔案的 URL。
從主目錄載入 ivy.properties 檔案 (如果有)	自動從您的起始目錄載入 ivy.properties 檔案。

結果

Studio 會將您的 Ivy 外掛程式喜好設定套用至組合專案。

在 Workday Studio 專案中新增 Workday Ivy 屬性

開始之前

- 配置 Workday Studio 的 ApacheIvy™ 外掛程式。
- 建立程序集專案。

執行這項作業的原因和時機

Workday Studio 支援 Apache Ivy 外掛程式，讓您透過配置具有 Workday Ivy 屬性的專案，藉此管理專案依存關係。

程序

1. 在「Project Explorer」檢視中，在組合專案上按一下滑鼠右鍵。

2. 選取「配置」，> 新增 Workday Ivy (性質)。
3. 展開集合專案資料夾。
4. 開啟 ivy.xml 檔案
5. 展開 資訊 節點。執行此任務時，請參考以下說明：

選項	敘述
模組	模組名稱。
在此新增您的內容	插入父系 <dependencies> 元素，包含一或多個子系 <dependency> 元素

結果

Studio 會在專案中新增 Workday Ivy 屬性。

為具有 Ivy 定義的專案解除依存關係

開始之前

- 配置 Workday Studio 的 ApacheIvy™ 外掛程式。
- 配置具有 Workday Ivy 屬性的組合專案。

執行這項作業的原因和時機

Workday Studio 支援 Apache Ivy 外掛程式，讓您透過配置具有 Workday Ivy 屬性的專案，藉此管理專案依存關係。如果您的專案顯示與缺少依存關係相關的建置錯誤，您可以解決這些依存關係問題。

程序

1. 在「專案資源管理器」檢視中，以滑鼠右鍵按一下屬於 Workday Ivy 的組合專案。
2. 選取「解決」。

結果

主控台 檢視會顯示解決方法報告。

將相依 JAR 新增至常春藤項目

開始之前

- 配置 Workday Studio 的 ApacheIvy™ 外掛程式。
- 配置具有 Workday Ivy 屬性的組合專案。

執行這項作業的原因和時機

Workday Studio 支援 Apache Ivy 外掛程式，讓您透過配置具有 Workday Ivy 屬性的專案，藉此管理專案依存關係。如果 Studio 專案屬於 Ivy 項目，其項目中包含依存關係元素，ivy.xml 檔，您可以將專案配置為包含任何相依的 JAR 檔案。

程序

1. 在「專案資源管理器」檢視中，以滑鼠右鍵按一下屬於 Workday Ivy 的組合專案。
2. 選取 屬性「」。
3. 在「Java 建置路徑」視窗中，按一下 資料庫。
4. 按一下 Ivy-Workday IDE。
5. 按一下 新增 JAR。

6. 在「JAR 選取」視窗上，選取要新增至專案的 JAR 檔案，然後按一下「確定」。
7. 在「Java 建置路徑」視窗上，按一下 套用並關閉。

結果

Studio 會將相依的 JAR 檔案新增至您的專案。

下一步

部署您的專案。

從 Studio 專案中移除 Workday Ivy 屬性

開始之前

- 配置 Workday Studio 的 ApacheIvy™ 外掛程式。
- 配置具有 Workday Ivy 屬性的組合專案。

執行這項作業的原因和時機

Workday Studio 支援 Apache Ivy 外掛程式，讓您透過配置具有 Workday Ivy 屬性的專案，藉此管理專案依存關係。您可以視需要在專案中新增或移除 Workday Ivy 類型。

程序

1. 在「專案資源管理器」檢視中，選取程序集專案資料夾。
2. 選取 Workday > Ivy > (移除 Workday Ivy (性質))。

結果

Studio 會從您的專案中移除 Workday Ivy 的性質。

下一步

部署您的專案。

整合部署

新增人脈

執行這項作業的原因和時機

在 Studio 中，您可以建立和管理與 Workday 的連線。您可以在部署和啟動程序集時使用這些連線。

程序

1. 選取 Windows > 喜好設定 > Workday > 連線。
2. 按一下 新增...。
3. 在「新增連線」視窗中，請參考以下說明：

選項	敘述
名稱	連線的名稱。
URL	連線的 URL。

選項	敘述
群組	將連線新增至群組。

4. 在「憑證」區段中，指定用戶並選取驗證方法：Basic Auth 或 OAuth 2.0。

註：若要使用 OAuth 2.0 授權，您必須先註冊 API 用戶端。該用戶端上已註冊 Workday 多重要素驗證的使用者從 Studio 連線至 Workday 時，必須使用另一種驗證方法。

選項	敘述
基本驗證	指定連線的 使用者名稱 和 密碼。
OAuth 2.0	指定連線的下列詳細資料： • 配股類型 • 用戶端 ID • 授權端點 • 權杖端點 • 存取權杖

5. 按一下 測試連線。

Studio 會確認是否可存取連線。

結果

人脈會顯示在「人脈」喜好設定頁面上。

下一步

建置程序集。

relatedlinks

相關任務

[在 Workday Studio 中啟用多重要素驗證](#) 頁 250

新增人脈群組

執行這項作業的原因和時機

在 Studio 中，您可以建立和管理與 Workday 的連線。您可以將這些人脈整理成群組。您可以在部署和啟動程序集時使用這些連線。

程序

1. 選取 Windows > 喜好設定 > Workday > 連線。
2. 按一下 新增群組....
3. 在 新增群組 視窗中，請參考以下說明：

選項	敘述
名稱	連線群組的名稱。
已篩選	從 Cloud Explorer 檢視和「部署至 Workday」頁面中篩選出連線群組。

結果

人脈群組會顯示在「人脈」喜好設定頁面上。

下一步

建置程序集。

編輯人脈

開始之前

新增 Workday 連線。

執行這項作業的原因和時機

在 Studio 中，您可以建立和管理與 Workday 的連線。您可以在部署和啟動程序集時使用這些連線。

程序

1. 選取 Windows > 喜好設定 > Workday > 連線。
2. 選取連線。
3. 按一下 編輯環境 URL。執行此任務時，請參考以下說明：

選項	敘述
主機	連線的主機 ID。
URL	連線的 URL。

結果

人脈會顯示在「人脈」喜好設定頁面上。

下一步

建置程序集。

編輯人脈群組

開始之前

建立 Workday 人脈群組。

執行這項作業的原因和時機

在 Studio 中，您可以建立和管理與 Workday 的連線。您可以將這些人脈整理成群組。您可以在部署和啟動程序集時使用這些連線。

程序

1. 選取 Windows > 喜好設定 > Workday > 連線。
2. 選取連線群組。
3. 執行此任務時，請參考以下說明：

選項	敘述
名稱	連線群組的名稱。
已篩選	從 Cloud Explorer 檢視和「部署至 Workday」頁面中篩選出連線群組。

結果

人脈群組會顯示在「人脈」喜好設定頁面上。

下一步

建置程序集。

移除人脈

開始之前

新增 Workday 連線。

執行這項作業的原因和時機

在 Studio 中，您可以建立和管理與 Workday 的連線。您可以在部署和啟動程序集時使用這些連線。

程序

1. 選取 Windows > 喜好設定 > Workday > 連線。
2. 選取連線。
3. 按一下 移除...

結果

Studio 會從「人脈」喜好設定頁面中移除該人脈。

下一步

建置程序集。

匯入連結方

開始之前

存取包含 Workday 連線詳細資料的 XML 檔案。

執行這項作業的原因和時機

在 Studio 中，您可以建立和管理與 Workday 的連線。您可以在部署和啟動程序集時使用這些連線。

程序

1. 選取 Windows > 喜好設定 > Workday > 連線。
2. 按一下 匯入...
3. 按一下 瀏覽...
4. 選取 Workday 連線 XML 檔案。

結果

Studio 會在「人脈」喜好設定頁面上顯示人脈。

下一步
建置程序集。

匯出連結方

開始之前
新增 Workday 連線。

執行這項作業的原因和時機
在 Studio 中，您可以建立和管理與 Workday 的連線。您可以在部署和啟動程序集時使用這些連線。

- 程序
- 1. 選取 Windows > 喜好設定 > Workday > 連線。
 - 2. 按一下 匯出...。
 - 3. 選取連線。
 - 4. 按一下 瀏覽...
 - 5. 請選取 Workday 連線 XML 檔案的目標位置。

結果
Studio 會將連線匯出至目標位置。

下一步
建置程序集。

連線至 Workday

開始之前
新增 Workday 連線。

執行這項作業的原因和時機
在 Studio 中，您可以建立和管理與 Workday 的連線。您可以在部署和啟動程序集時使用這些連線。

- 程序
- 1. 從 Studio 喜好設定 功能表中，選取 「Workday > 人脈」。
 - 2. 選取連線。執行此任務時，請參考以下說明：

選項	敘述
用戶	請指定您的用戶名稱。
使用者名稱	輸入用戶的使用者名稱憑證。
密碼	輸入用戶的密碼憑證。

- 3. 按一下 測試連線。
Studio 會確認是否可存取您的用戶。
- 4. 按一下 套用並關閉。

結果

Studio 會連線至您的 Workday 用戶。

下一步

將組套件部署至 Workday。

將整合部署至 Workday

開始之前

- 建立程序集專案。
- 安全性：「整合建置」和「整合」，請在「整合」功能區中配置 網域。

執行這項作業的原因和時機

在 Studio 中，您可以將整合部署至 Workday。部署整合後，您可以在 Workday 用戶中啟動該整合。

程序

1. 在「Project Explorer」檢視中，在組合專案上按一下滑鼠右鍵。
2. 選取 部署至 Workday。
3. 在「部署至 Workday」視窗中，請參考以下說明：

選項	敘述
Workday 環境	請選取要部署整合的目標 Workday 環境。
部署中的集合	選取要部署至 Workday 的集合。

4. 按一下 完成。

結果

Studio 會將您的整合部署至 Workday。

下一步

啟動您的整合。

在 Workday 中檢視已部署的整合

開始之前

- 將整合部署至 Workday。
- 安全性：「整合建置」和「整合」，請在「整合」功能區中配置 網域。

執行這項作業的原因和時機

在 Studio 中，您可以將整合部署至 Workday。登入 Workday 用戶後，即可檢視已部署的整合。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。

4. 選取 Workday：整合系統。
5. 在 Workday 分頁上，登入您的 Workday 用戶。執行此任務時，請參考以下說明：

選項	敘述
使用者名稱	輸入 Workday 用戶的使用者名稱憑證。
密碼	輸入用戶的密碼憑證。

結果

您部署的整合會顯示在 Studio 的 Workday 分頁上。

下一步

啟動您的整合。

將 Studio 配置為使用 HTTPS Proxy Server

執行這項作業的原因和時機

如果您需要使用 443 以外的連接埠將 Workday Studio 連至培訓或測試環境，而您組織的防火牆預設不允許此動作，您可以使用網路中的 Proxy 伺服器進行連線。

程序

1. 選取 Windows > 喜好設定 -> 一般 > 網路連線。
2. 從 作用中的提供者 下拉式功能表中，選取 手動。
註：當您選取「手動」時，Proxy 項目 表格中的所有 3 個 結構 描述 (HTTP、HTTPS 和 SOCK) 對應的核取方塊都會勾選。您無法個別清除這些選擇。
3. 在 HTTPS 結構描述旁，輸入 Proxy 伺服器的 主機 IP 位址和 連接埠 號碼。
4. 按一下 套用並關閉。

概念：Studio 連結

在 Studio 中，您可以建立和管理與 Workday 的連線。您可以在部署和啟動程序集時使用這些連線。

註：所有連線名稱都必須是唯一的。Studio 提供 沙箱環境 和 正式 作業環境 做為預設連線。您無法重新命名這些連線。出於安全原因，在您退出 Studio 階段作業後，Studio 不會將使用者名稱和密碼詳細資料儲存至您的工作區。

概念：Cloud Explorer

Cloud Explorer 檢視會顯示您的雲端伺服器連線。您可以展開每個雲端伺服器節點來檢視其集合。您可以展開每個集合節點，以檢視可供啟動的整合。

註：Cloud Explorer 檢視最多可顯示 999 個集合。

Cloud Explorer 檢視使用的日期格式設定與您的本機作業系統相同。

參照：連線錯誤

連線錯誤	附註
您的使用者登入憑證無法驗證。	請檢查您的憑證，然後再試一次。請記住，用戶名稱區分大小寫。
您的用戶無法使用。	使用 Web 瀏覽器檢查您的用戶是否在線。請以下列表單輸入用戶 URL： <pre>https://host-name/tenant-name/ccx/cc-cloud-repo/collections</pre> 範例： <pre>https://wd5.myworkday.com/gms/ccx/cc-cloud-repo/collections</pre> 請使用您嘗試在 Studio 中使用的憑證，但使用者和用戶名稱中必須加上 @ 字元。範例： <pre>lmcneil@gms</pre> 如果連線成功，瀏覽器會顯示您用戶中部署的所有 Studio 整合集合。接下來，請檢查瀏覽器的連線喜好設定，查看瀏覽器是建立直接連線還是經由 Proxy 伺服器連線。如果配置為使用 Proxy 伺服器，則您需要將 Studio 配置為也使用該 Proxy。
部署權限遭拒。	要將整合部署至 Workday，您必須是「整合建置」網域中具有「修改」權限的安全群組成員。請聯絡您的 Workday 系統管理者。
已拒絕啟動權限。	若要從 Studio 啟動整合，您必須是以下其中一個網域中具有「更新」權限的安全群組的成員： • 整合事件 • 整合建置 • 整合偵錯 請聯絡您的 Workday 系統管理者。
部署權限遭拒 (整合包含交付或擷取服務)。	若要部署包含交付或擷取服務的整合，您需要其他權限。您必須是安全群組的成員，且該安全群組必須在下列其中一個網域中擁有「更新」權限： • Business Process Administration • Manage: Business Process Definitions • iLoad Web 服務 請聯絡您的 Workday 系統管理者。
已拒絕存取「我的報告」。	Workday 會使用可配置的安全性來控制以下權限：存取「我的報告」整合文件。您用來存取整合文件的 Workday 帳戶沒有必要的權限。請聯絡您的 Workday 系統管理者，以要求必要的權限。
SOAP 錯誤：發生驗證錯誤。	您對 Web 服務作業的 SOAP 要求無效。確認您的要求包含有效的資料元素和值。

連線錯誤	附註
	使用 Schema Explorer 中的「新增最新 Workday Web 服務」精靈，將有效的 WSDL 匯入 Workday Studio 環境中。然後，使用 建立和復制 SOAP 常值 或在 Web 服務測試器中開啟 SOAP 封裝的要求 功能表來建立有效的 SOAP 要求。
您的 IP 位址無法驗證。	您的 IP 位址無權存取 Workday 環境。請聯絡您的系統管理者。

Workday Studio 中的多重要素驗證

在 Workday Studio 中啟用多重要素驗證

執行這項作業的原因和時機

您可以將 Workday Studio 配置為在連線至用戶時使用 OAuth 2.0 授權。為此，您必須註冊 API 用戶端。用戶端使用者若在某個用戶中註冊了 Workday 多重要素驗證，則從 Studio 連線至該用戶時，必須使用另一種驗證方法。他們無法使用基本驗證。

程序

1. 在您的用戶中設定多重要素驗證，並指定需要哪些帳戶才能使用多重驗證。
請參閱「[步驟：使用身分驗證應用程式設定多重要素驗證](#)」。
請參閱「[步驟：使用 Duo Security 設定多重要素驗證](#)」。
請參閱「[步驟：使用透過電子郵件傳送的單次通行碼設定多重要素驗證](#)」。
請參閱「[步驟：使用 SMS 簡訊單次通行碼設定多重要素驗證](#)」。
2. 註冊啟用 OAuth 2.0 的 API 用戶端。
配置以下設定：

選項	值
用戶端授予類型	授權授予代碼
支援代碼交換驗證金鑰 (PKCE)	已啟用
存取權杖類型	Bearer
重新導向 URI	請輸入有效的 HTTPS URL。
範圍	在「自訂物件」清單中，選取您要存取的每個 Web 服務的功能區。 此外，請選取以下功能區： • 實作 • 整合 • 系統 • 用戶不可配置 註：如果您省略任何相關的 Web 服務，驗證將失敗。

選項	值
包含 Workday 自有的範圍	已啟用

註：請勿選取「強制執行 60 分鐘存取權杖到期」核取方塊。這麼做會在該期間過後停用 OAuth 2.0 驗證。OAuth 2.0 驗證也受制於為全用戶或個別帳戶配置的正常 Workday 階段作業逾時。

請參閱「[註冊 API 用戶端](#)」。

- 複製「註冊 API 用戶端」任務中提供的 用戶端 ID、授權端點和 權杖端點。
- 從 Workday Studio 的「喜好設定」功能表，瀏覽至「Workday > 人脈」。
- 新增用戶連線，然後選取 OAuth 2.0 憑證分頁。
- 貼上 用戶端 ID、授權端點和 權杖端點。
- 按一下 取得存取權杖。
如果您收到「內部伺服器錯誤」訊息，最可能的原因是用戶端 ID 不正確。
- 輸入用戶的使用者名稱和密碼。
如果您已登記 Workday 多重要素驗證，系統會提示您提供另一種驗證方法。
- 測試連線。
- 按一下 套用並關閉。

relatedlinks

相關參考資料

[2021R2 功能更新公告：Workday Studio 中的多重要素驗證](#)

整合配置

將整合系統使用者帳戶指派給整合

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合安全性」網域。

執行這項作業的原因和時機

Workday Studio 整合需要整合系統使用者 (ISU) 帳戶，才能進行驗證和 Web 服務呼叫。

程序

- 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
- 選取並展開集合節點。
- 在整合節點上按一下滑鼠右鍵。
- 選取「指派整合系統使用者」。
- 按一下 整合系統使用者。
- 從右側視窗窗格中選取 ISU。

結果

您可以以 ISU 的身分啟動整合。

下一步

啟動您的整合。

概念：整合系統安全性

在 Workday 中，您可以加入安全群組，以便在用戶中執行特定任務。在 Studio 中，當您連線至 Workday 用戶時，測試連線視窗會顯示 3 個安全群組：

- 部署
- 啟動
- 業務流程

在 Studio 中，您至少必須是其中一個安全群組的成員，才能執行整合。

relatedlinks

相關概念

[概念：可配置的安全性](#)

整合啟動

從 Studio 啟動整合

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

Studio 可讓您在 Workday 用戶中啟動整合。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。

結果

您的整合會在您的 Workday 用戶中啟動。在 Studio 中，流程監控分頁會顯示您的整合狀態。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

複製啟動 URL

開始之前

將整合部署至 Workday。

執行這項作業的原因和時機

Studio 可讓您在 Workday 用戶中復制整合的啟動 URL。您可以使用這些 URL 從具有基本驗證權限的第三方 REST 用戶端啟動整合。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取「複製啟動 URL」。
5. 在 啟動 URL 視窗上，反白選取啟動 URL，然後按 Ctrl + C。

結果

Studio 會將整合的啟動 URL 複製到剪貼簿。

下一步

在來自具有基本驗證權限的第三方 REST 用戶端的 REST 要求中，使用啟動 URL。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

建立啟動配置

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

Studio 可讓您在 Workday 用戶中啟動整合。您可以為整合建立啟動配置。

程序

1. 選取 執行 > 執行配置。
2. 在左側導覽窗格中，以滑鼠右鍵按一下 整合。
3. 按一下 新增。
4. 在右側導覽窗格中，請考慮：

選項	敘述
名稱	啟動配置的名稱。
啟動時間	用於啟動整合的目標 Workday 環境。

選項	敘述
收款	要在整合中啟動的集合。
專案	要在整合中啟動的專案。
Workday-In	整合中 Workday-In 傳輸的名稱。
啟動參數	整合的啟動參數。

- 按一下套用。
- 按一下 執行。

結果

Studio 會使用您新的啟動配置詳細資料執行整合。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

編輯啟動配置

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

Studio 可讓您在 Workday 用戶中啟動整合。您可以編輯整合的啟動配置。

程序

- 選取 執行 > 執行配置。
- 在左側導覽窗格中，展開 整合 節點。
- 選取啟動配置節點。
- 在右側導覽窗格中，請考慮：

選項	敘述
名稱	啟動配置的名稱。
啟動時間	用於啟動整合的目標 Workday 環境。
收款	要在整合中啟動的集合。
專案	要在整合中啟動的專案。
Workday-In	整合中 Workday-In 傳輸的名稱。
啟動參數	整合的啟動參數。

- 按一下套用。
- 按一下 執行。

結果

Studio 會使用您新的啟動配置詳細資料執行整合。

relatedlinks

相關任務

將整合部署至 Workday 頁 247

整合監控與疑難排解

流程監控

概念：整合事件 ID

Workday 會為所有整合事件指派 ID，讓您追蹤整合事件從啟動到完成的整個過程。當您從 Workday Studio 啟動整合時，整合事件 ID 會顯示在「流程監控」檢視的 ID 欄中。

參照：流程監控工具列按鈕

按鈕	說明
重新整理	重新整理「流程監控」檢視。
刪除流程	從「流程監控」檢視中刪除流程。
將流程 ID 複製到剪貼簿	將整合事件 ID 複製到剪貼簿。
取消流程	取消流程。
檢視合併報告	在合併報告檢視器中開啟合併報告。
檢視記錄檔案	在「檢視記錄檔案」視窗中顯示流程的記錄檔案。
檢視要求訊息	在「檢視要求訊息」視窗中顯示流程的要求訊息。
匯入合併報告	開啟「匯入合併報告」視窗。
匯入記錄檔	開啟「匯入記錄檔案」視窗。
匯入要求訊息	開啟「匯入要求訊息」視窗。
啟用自動重新整理	啟用「流程監控」檢視的自動重新整理功能。
Workday：流程監控	在 Workday 分頁上開啟 Workday 流程監控。
Workday	開啟 Workday 檢視並顯示您的整合的整合伺服器 ESB 呼叫。
檢視功能表	顯示「流程監控」檢視的工具列選項。

合併報告檢視器

載入整合的合併報告

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

在 Studio 中，合併報告檢視器 分頁會顯示與整合中每個步驟的計時、內存使用和呼叫次數相關的效能資訊。合併報告可讓您分析整合的處理流程。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。
6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
7. 選取「檢視合併報告」。

結果

您的合併報告會顯示在 合併報告檢視器 分頁上。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

將合併報告儲存至您的工作區

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

合併報告可讓您分析整合的處理流程。您可以將合併報告儲存至工作區。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。
6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
7. 選取「檢視合併報告」。
8. 在 合併報告檢視器 分頁上的合併報告上按一下滑鼠右鍵。
9. 選取 另存新檔...
10. 在「儲存合併報告」視窗中，選取合併報告的上層資料夾。
11. 按一下 完成。

結果

Studio 會將合併報告儲存至您的工作區。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

使用合併報告檢視 **Workday** 整合事件詳細資料

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

合併報告可讓您分析整合的處理流程。您可以在 Studio 中使用合併報告檢視 Workday 整合事件詳細資料。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。
6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
7. 選取「檢視合併報告」。
8. 在 合併報告檢視器 分頁上，選取「在 Workday 中檢視」。
9. 在 Workday 分頁上，登入您的 Workday 用戶。執行此任務時，請參考以下說明：

選項	敘述
使用者名稱	您的 Workday 用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

結果

您的 Workday 整合事件詳細資料會顯示在 Studio 的 Workday 分頁上。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

使用合併報告檢視整合事件的要求訊息

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

合併報告可讓您分析整合的處理流程。您可以使用 Studio 中的合併報告，檢視整合事件的要求訊息。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。
6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
7. 選取「檢視合併報告」。
8. 在 合併報告檢視器 分頁上，選取 檢視要求訊息。

結果

整合事件的要求訊息會顯示在「檢視要求訊息」視窗中。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

使用合併報告檢視器事件記錄監控整合

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

合併報告可讓您分析整合的處理流程。您可以使用 Studio 中 合併報告檢視器 分頁中的事件記錄來監控您的整合。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。
6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
7. 選取「檢視合併報告」。
8. 在 合併報告檢視器 分頁上，選取 事件記錄。

結果

整合的事件記錄會顯示在 合併報告檢視器 分頁上。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

使用合併報告檢視器設定檔記錄監控整合

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

合併報告可讓您分析整合的處理流程。您可以在 Studio 的 合併報告檢視器 分頁中，使用設定檔記錄來監控您的整合。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。
6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
7. 選取「檢視合併報告」。
8. 在 合併報告檢視器 分頁上，選取 基本資料記錄。

結果

整合的設定檔記錄會顯示在 合併報告檢視器 分頁上。使用的時區為太平洋標準時間。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

重播合併報告的程序集執行

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

合併報告可讓您分析整合的處理流程。您可以使用 Studio 中的合併報告，為您的整合執行程序集。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。
6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
7. 選取「檢視合併報告」。
8. 在 合併報告檢視器 分頁上，選取「重播」。

結果

整合的程序集執行會顯示在程序集編輯器上。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

將合併報告稽核事件連結至程序集元素

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

合併報告可讓您分析整合的處理流程。您可以在 Studio 中將合併報告稽核事件連結至程序集元素。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。
3. 在整合節點上按一下滑鼠右鍵。
4. 選取 啟動整合。
5. 按一下 啟動。
6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
7. 選取「檢視合併報告」。
8. 在 合併報告檢視器 分頁上，選取 使用編輯器連結。

結果

Studio 會將您的合併報告稽核事件連結至整合中的程序集元素。當您在 合併報告檢視器 分頁上選取節點時，Studio 會在組合圖表上以醒目提示標出對應的組合元素。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

在組合圖上顯示合併報告註解

開始之前

- 將整合部署至 Workday。
- 安全性：「整合」功能區中的「整合事件」網域。

執行這項作業的原因和時機

合併報告可讓您分析整合的處理流程。您可以在 Studio 的組合圖中顯示合併報告註解。

程序

1. 在 Cloud Explorer 檢視中，選取並展開雲端伺服器節點。
2. 選取並展開集合節點。

- 3. 在整合節點上按一下滑鼠右鍵。
- 4. 選取 啟動整合。
- 5. 按一下 啟動。
- 6. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
- 7. 選取「檢視合併報告」。
- 8. 在 合併報告檢視器 分頁上，選取 註解。

結果

Studio 會在組合編輯器的組合圖表上顯示合併報告註解。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

概念：合併報告檢視器

在 Workday Studio 中，「合併報告檢視器」分頁會顯示與整合中每個步驟的計時、內存使用和呼叫次數相關的效能資訊。「合併報告檢視器」分頁也會匯總 Workday Web 服務呼叫並以醒目提示標出整合錯誤。

您可以將合併報告連結至程序集。您也可以從圖表中的每個組合元素上顯示合併報告註解，以便找出整合中的問題和錯誤。

「合併報告檢視器」分頁會顯示為 Workday Studio 預設觀點的一部分。如果您將其關閉，可以選取 視窗 - > 顯示 > 合併報告檢視器 -將其重新開啟。

您可以從「合併報告檢視器」分頁載入合併報告，也可以從所選整合流程的「流程監控」檢視載入合併報告。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

參照：合併報告錯誤訊息

錯誤訊息	範例	說明
java.lang.Exception : Win on E2 : XSLT 作業失敗	<pre>com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete, ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. The Job with referenceId <referenceid> has failed as it exceeded the maximum number of retry attempts, 1 Root Cause: java.lang.Exception:</pre>	當您的作業超過重試嘗試次數上限時顯示。

錯誤訊息	範例	說明
	Win on E2: XSLT job failed.	
HttpAuthenticationRequiredException 未經授權	com.capeclear.mediation.MediationException: RetrieveStep id=null encountered a problem sending data to endpoint <endpoint>, reason:HttpAuthenticationRequiredException: Unauthorized.	當您沒有足夠的權限存取附件時顯示。
com.workday.esb.blobitory.Blobitory 用戶網域非作用中	com.capeclear.mediation.MediationException: application=bpelEnterpriseInterfaceBuilder <version-number> - Failed to store resource in collection eib_xslt_collection in workspace wdoms.	用戶配置錯誤。請聯絡支援和技術營運，確認用戶狀態為作用中。
java.lang.OutOfMemoryError : Java 堆空間。	com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete, ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. Integration Failed. com.capeclear.mediation.MediationException: Xslt step id=transformEIB failed to apply stylesheet mctx:vars/eib.xslt.	當您的整合包含編寫不佳的 XSLT 時顯示。
java.xml.transform.TransformConfig	com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete, ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. Integration Failed. com.capeclear.mediation.MediationException: Xslt step id=transformEIB stylesheet mctx:vars/ eib.xslt located at mctx:vars/eib.xslt is invalid or missing, reason: javax.xml.transform.TransformerConfigurationException: Failed to compile stylesheet. 1 error detected. Error reported by	當您的整合包含無效的 XSLT 時顯示。

錯誤訊息	範例	說明
	<pre>XML parser org.xml.sax.SAXParseException; systemId: mctx:vars/ eib.xslt; lineNumber: 960; columnNumber: 5; The markup in the document following the root element must be well-formed. Root Cause: javax.xml.transform.TransformerConfigurationException: Failed to compile stylesheet. 1 error detected.</pre>	
com.workday.services.server.capeclear.MediationException	<pre>com.capeclear.mediation.MediationException: application=bpelACHTransformOutbound-21.0.06.267 - Custom step id=aggregateElement, ref=DocumentElementProcessor - Error while invoking bean, reason: document processor step error Root Cause: com.workday.services.server.capeclear.bean.dep.ProcessingExc error transforming SOAP request element, reason:error transforming request element, reason:javax.xml.transform.TransformerException: org.w3c.dom.DOMException: HIERARCHY_REQUEST_ERR: An attempt was made to insert a node where it is not permitted.</pre>	當您運作無效的 XSLT 處理格式以建立 SOAPElementProcessor Transformer 實例
java.io.FileNotFoundException	<pre>com.capeclear.mediation.MediationException: application=<application- name> - Http Out transport id=retrieveSchemaFromOMS encountered error sending to endpoint <endpoint> Root Cause: java.io.FileNotFoundException: <keystore-file- location> (No such file or directory).</pre>	當 Workday ESB 伺服器中缺少某個金鑰庫檔案時顯示。
org.jaxen.UnresolvableException	<pre>com.capeclear.mediation.MediationException: Error eval step id=ValidatePaygroup</pre>	當您的程序集中缺少名稱空間字自定義時顯示。

錯誤訊息	範例	說明
	<pre> expression=props['myprop.empl'] = vars['input.picof'].xpath('/ picof:Employee/ picof:Status/ picof:Pay_Group') failed to evaluate. Reason: unable to resolve property: vars['input.picof'].xpath('/ picof:Employee/ picof:Status/ picof:Pay_Group') Root Cause: org.jaxen.UnresolvableException: Cannot resolve namespace prefix 'picof'. </pre>	
BadRequestException	<pre> com.capeclear.mediation.MediationException: Workday Out transport id=CallGetPos encountered problem sending to endpoint 'https://impl- cc.workday.com/ccx/ service/mycompany/ Staffing/v17' with operation 'Get_Positions_Request'. Details: Code: SOAP- ENV:Client.validationError, Reason: Validation error occurred. Invalid ID value. '?' is not a valid ID value for type = 'Position_ID' Root Cause: BadRequestException : Error code : unknown. </pre>	當您的整合產生無效的 SOAP 要求訊息時顯示。
java.net.SocketTimeoutException	<pre> com.capeclear.mediation.MediationException: Transport error in email out transport id=emFailureRecords with endpoint=mailto: reason:com.capeclear.capecconnect.transport.TransportException Transport messaging error: Unable to send message : Exception reading response Root Cause: java.net.SocketTimeoutException: Read timed out. </pre>	當您的整合包含無效的 SMTP 詳細資料時顯示。

relatedlinks

相關任務

[將整合部署至 Workday 頁 247](#)

[從 Studio 啟動整合 頁 252](#)

單元集合測試

建立程序集專案的 AUnit 測試框架

執行這項作業的原因和時機

AUnit 是一個 Workday Studio 測試框架，用於開發和執行程序集的單元測試。單元測試主旨可讓您為程序集建立測試用例。

程序

1. 選取「檔案 > 新的 > Workday 程序集專案」。
2. 在 專案名稱 欄位上，輸入專案的名稱。
專案名稱不可包含空格。
3. 在「目標執行階段」欄位上，選取 *Workday Runtime*。
4. 在「配置」欄位上，選取「程序集 (具有 Java 支援)」。
5. 按一下 完成。
6. 選取「檔案 > 新增 > 其他 > 類別」。
7. 在「新增 Java 類別」功能表上，請參考以下說明：

選項	敘述
方案	輸入方案名稱。
名稱	輸入新 Java 類別的名稱。
超類別	按一下 瀏覽，然後選取 <code>com.workday.aunit.AssemblyTestCase</code> 。

8. 按一下 完成。
9. 在「Project Explorer」檢視中，在專案中的 src 資料夾上按一下滑鼠右鍵。
10. 選取 新增 JUnit 資料庫：。

結果

Studio 會在 Java 編輯器上顯示您的 AUnit 測試框架。

下一步

為您的程序集執行單元測試。

概念：A 單元測試

AUnit 是一個 Workday Studio 測試框架，可讓您開發和執行程序集的單元測試。您可以使用在程序集專案中建立的測試主旨，以 Java 開發 AUnit 測試。AUnit 測試框架支援以下功能：

- 動作
- 註解
- 判斷提示

您可以使用這些功能來建立程序集的測試用例。

範例：

```
package com.workday.training.aunit;

import java.io.InputStream;

import com.workday.aunit.AssemblyTestCase;
import com.workday.aunit.actions.Action;
import com.workday.aunit.actions.StandardAction;
import com.workday.aunit.annotations.AssemblyTest;
import com.workday.aunit.annotations.AssertAfter;
import com.workday.aunit.annotations.AtComponent;
import com.workday.aunit.annotations.UnitTest;

@AssemblyTest(project = "AUnitTrainer", displayLabel = "Citizens of the
Cloud")
public class SimpleMediationTest extends AssemblyTestCase {

    @UnitTest(startComponent = "Simple-Mediation")
    public void testSimpleMediation() throws Exception {
        setMessagePart(0, "test/simple-input.xml", "text/xml");
    }

    // Using the @AssertAfter annotation and the assertTrue assertion.    //
    @AssertAfter(id = "Simple-Mediation", step="transform")
    public void verifySimpleMediation() throws Exception {
        assertTrue(compare(
            xml",
            (InputStream) getMediationContext().getMessage().getMessagePart(0,
            InputStream.class),
            "text/xml", Comparator.dom));
    }

    @AtComponent(id="Next-Thing")
    public Action handleAfterMediation() throws Exception {
        return new StandardAction(Action.Type.terminate);
    }
}
```

註解

所有 AUnit 測試都必須包含 @AssemblyTest 註解此註解可識別您的邏輯單元測試，並使您能夠將該測試與您的程序集專案建立關聯。範例：

```
@AssemblyTest(project = "AUnitTrainer")
```

測試方法上的註解定義了程序集處理鏈中單元測試的起點。範例：

```
@UnitTest(startComponent = "Simple-Mediation")
```

判斷提示

判斷提示定義了布林測試條件。如果判斷提示失敗，AUnit 會中止程序集執行，並將測試結果顯示為失敗。如果發生非預期的例外狀況，AUnit 會中止程序集執行，並將測試結果顯示為錯誤。

範例：以下範例使用 @AssertAfter 註解和 assertTrue 判斷提示，以在中介執行後檢查根訊息部分是否包含預期的 XML。

```
@AssertAfter(id = "Simple-Mediation", step="transform")
public void verifySimpleMediation() throws Exception {
    assertTrue(compare(
        getTestResourceInputStream("test/simple-expected.xml"), "text/xml",
```

```

        (InputStream) getMediationContext().getMessage().getMessagePart(0,
        InputStream.class),
        "text/xml", Comparator.dom));
    }

```

動作

動作定義了程序集執行至元件或退出點時的 AUnit 測試行為。範例：

```

@AtComponent(id="Next-Thing")
public Action
    handleAfterMediation() throws Exception {
        return new StandardAction(Action.Type.terminate); }

```

參照：功能單元測試註解

註解	說明	範例
@AssemblyTest	將測試識別為 AUnit 測試，並將其與程序集建立關聯。	@AssemblyTest(project = "AUnitTrainer", displayLabel = "Citizens of the Cloud")
@UnitTest	識別個別測試用例。	@UnitTest(startComponent = "Simple-Mediation") public void testSimpleMediation() throws Exception { setMessagePart(0, "test/simple-input.xml", "text/xml"); }
@AssertBefore	在執行具名的程序集元件或步驟之前執行判斷提示。	@AssertBefore(id="Store-Output") public void verifySummaryDoc() throws Exception { assertEquals("10", evaluateExpression("parts[0].xpath(summary/ wd:WorkerSummary)')))); }
@AssertAfter	在執行指定的程序集元件或步驟後執行判斷提示。	@AssertAfter(id = "Simple-Mediation", step="transform") public void verifySimpleMediation() throws Exception { assertTrue(compare(getTestResource("simple-expected.xml"), "text/xml", (InputStream) getMediationContext().getMessage().getPart(0, InputStream.class), "text/xml", Comparator.dom)); } @AssertAfter(id="Process", step="transform")

註解	說明	範例
		<pre>public void verifySummary() throws Exception { assertEquals("Ace Developer", evaluateExpression("parts[0].xpath wd:WorkerSummary/ wd:Name'")); assertEquals("2010-07-27-07:00", evaluateExpression("parts[0].xpath wd:WorkerSummary/ wd:HireDate'")); }</pre>
@AssertAfterCompletion	在組合完成後執行判斷提示。	<pre>@AssertAfterCompletion public void verifyErrorHandling() throws Exception { assertEquals(3, evaluateExpression("props['error.c } }</pre>
@AtComponent	定義測試在何處截取程序集以執行驗證，並定義指定元件的測試行為。	<pre>@AtComponent(id="Get- Workers") public Action handleGetWorkersInvocation() throws Exception { setMessagePart(0, "test/fromworkdayin/ workdayin_workers_response.xml", "text/xml"); return new StandardAction(Action.Type.mock); }</pre>

參照：功能單元測試動作

動作	說明	範例
continue_to_next	繼續對集合中的下一個元素執行 AUnit 測試。	<pre>return new StandardAction(Action.Type.continu</pre>
invoke	呼叫由集合的傳出所指定的端點。	<pre>return new StandardAction(Action.Type.invoke)</pre>
mock	指定包含回應訊息的輸入檔案，以模擬傳出傳回的回應訊息。	<pre>return new StandardAction(Action.Type.mock);</pre>
mock_exception	透過指定此類的實例，模擬輸出傳輸所引發的例外狀況： com.capeclear.capecconnect.transport.BackRequestException(msg)	<pre>new StandardAction(Action.Type.mock_ex new BackRequestException("String message here..."));</pre>
terminate	終止測試。	<pre>return new StandardAction(Action.Type.termina</pre>

Web 服務測試

使用 WSDL 檔案測試 Web 服務

執行這項作業的原因和時機

您可以透過 Workday Studio 的 Web 服務測試工具，透過 WSDL 檔案中所述的任何作業，傳送範例要求訊息。

程序

1. 在 Studio 工具列上，按一下「在測試儀中測試 Web 服務」。
2. 執行此任務時，請參考以下說明：

選項	敘述
檔案	顯示所選 WSDL 檔案的位置。
工作區	從您的工作區選取 WSDL 檔案。
外部	從外部位置選取 WSDL 檔案。
URL	從 URL 中選取 WSDL 檔案。
起始沙箱環境的回應端點和驗證詳細資料	指定是否： • 測試人員的「地點」欄位已預先填入 Web 服務端點。 • WS-Security 使用者名稱權杖已預先填入您的沙箱使用者登入詳細資料。 預設會選取此選項。
完成	當您選取 WSDL 檔案時，按一下。或者，直接按一下而不指定檔案，以開啟 Web 服務測試工具並輸入您選擇的位置和訊息。

3. 從您的工作區、外部位置或 URL 選取 WSDL 檔案。按一下 完成。
4. Studio 會開啟 Web 服務測試工具。執行此任務時，請參考以下說明：

選項	敘述
地點	Web 服務端點。
PortType	Web 服務公開的 WSDL 作業及其相關聯的系結。
作業	WSDL 檔案中的作業當您選取一個訊息時，Studio 會在「要求」窗格中產生範例訊息。
SOAP 動作	HTTP 要求中的 SOAPAction 標頭。
要求	Studio 傳送至 Web 服務的要求訊息。
回覆	Studio 從 Web 服務接收的回應訊息。

5. 根據需要在 要求 窗格中修改 XML。若要新增附件，請按一下「新增附件」，然後按一下「編輯」指定檔案位置、附件的名稱和 MIME 類型。
6. 按一下 傳送要求，將要求訊息傳送至位於「位置」欄位中指定 URL 的 Web 服務。Studio 會在「回覆」窗格中顯示回覆。

使用 XSD 檔案測試 Web 服務

執行這項作業的原因和時機

您可以透過 Workday Studio 的 Web 服務測試工具，使用 XSD 檔案中所述的任何元素，傳送範例要求訊息。

程序

1. 在 Studio 工具列上，按一下「在測試儀中測試 Web 服務」。
2. 執行此任務時，請參考以下說明：

選項	敘述
檔案	顯示所選 XSD 檔案的位置。
工作區	從您的「工作區」選取 XSD 檔案。
外部	從外部位置選取 XSD 檔案。
URL	從 URL 中選取 XSD 檔案。
起始沙箱環境的回應端點和驗證詳細資料	指定是否： • 測試人員的「地點」欄位已預先填入 Web 服務端點。 • WS-Security 使用者名稱權杖已預先填入您的沙箱使用者登入詳細資料。 預設會選取此選項。
完成	選取 XSD 檔案後，按一下。或者，直接按一下而不指定檔案，以開啟 Web 服務測試工具並輸入您選擇的位置和訊息。

3. 從您的工作區、外部位置或 URL 選取 XSD 檔案。按一下 完成。
4. Studio 會開啟 Web 服務測試工具。執行此任務時，請參考以下說明：

選項	敘述
地點	Web 服務端點。由於該地點的值在 XSD 中不可用，因此您必須手動輸入地點。
元素	XSD 檔案中的元素定義。當您選取一個訊息時，Studio 會在「要求」窗格中產生範例訊息。

5. 根據需要在 要求 窗格中修改 XML。若要新增附件，請按一下「新增附件」，然後按一下「編輯」指定檔案位置、附件的名稱和 MIME 類型。
6. 按一下 傳送要求，將要求訊息傳送至位於「位置」欄位中指定 URL 的 Web 服務。Studio 會在「回覆」窗格中顯示回覆。

從 Schema Explorer 測試個別 Web 服務

開始之前

在 Schema Explorer 中顯示 Workday Web 服務 WSDL 或 XSD。

執行這項作業的原因和時機

您可以使用 Workday Studio Web 服務測試工具，從架構探索者建立範例 SOAP 或 XML 要求訊息，以測試個別的 Web 服務。

程序

1. 若要為特定 Web 服務產生範例要求訊息，請在 結構描述探索器 檢視中的 WSDL 或 XSD 元素上按一下滑鼠右鍵。請選取「在 Web 服務測試器中開啟要求」或「在 Web 服務測試器中開啟 SOAP 封裝的要求」。
2. Studio 會開啟 WS 測試人員 檢視。目標 區段會顯示 Web 服務端點的 位置 。
3. 從「內容類型」下拉清單中，選取訊息的媒體類型和子類型：

選項	敘述
text/plain; charset=utf-8	指定純文字，其中不包含格式設定命令或指令。
text/xml; charset=utf-8	指定 XML 文字。
application/octet-stream	指定了無法辨識的類型。例如，若您要將二進位檔案附加至訊息，請選取此選項。

4. 根據需要在 要求 窗格中修改 XML。若要新增附件，請按一下「新增附件」，然後按一下「編輯」指定檔案位置、附件的 名稱 和 MIME 類型。
5. 按一下 傳送要求 ，將要求訊息傳送至位於「位置」欄位中指定 URL 的 Web 服務。Studio 會在「回覆」窗格中顯示回覆。

relatedlinks

相關任務

[顯示 Workday Web 服務 WSDL 頁 206](#)

使用要求建置器精靈測試 Get 作業和要求訊息

開始之前

在 Schema Explorer中顯示 Workday Web 服務 WSDL 或 XSD。

執行這項作業的原因和時機

您可以使用 Workday Web 服務要求建置器精靈，簡化許多 Get 作業和要求訊息的測試。符合資格的作業和要求訊息的圖示左上角會顯示箭頭裝飾符號。

程序

1. 在「架構探索器」檢視中的相關「取得」節點上按一下滑鼠右鍵，然後選取「Workday Web 服務建置器精靈」。
2. 選取您是否要：
 - 請使用參考 ID 擷取特定資料。
 - 從 Web 服務擷取所有可用的記錄。
3. 如果您選擇使用參考 ID，請輸入此 ID。
4. 請選取您要納入的應變動作群組。
5. 選取您是否要：
 - 將範例要求複製到剪貼簿。
 - 在新的 WS 測試人員 檢視中開啟範例要求。
 - 取代目前 WS 測試人員 檢視的內容。
6. 按一下 完成。

配置 Web 服務測試人員驗證

開始之前

建立 Web 服務測試。

執行這項作業的原因和時機

您可以在 Workday Studio 的 Web 服務測試工具中配置多個安全選項。「驗證」選項可讓您指定是將 WS-Security 或基本驗證套用至 SOAP 要求訊息，還是要呼叫內部 cc-authenticate 服務。

程序

1. 在 WS 測試人員 檢視中，按一下 傳送選項。
2. 選取 驗證 核取方塊。執行此任務時，請參考以下說明：

選項	敘述
WS-Security UsernameToken	可讓您指定要搭配 WS-Security 使用的使用者名稱和密碼，而不必選取 WS-Security 配置檔案。Studio 會插入包含您在其產生的要求訊息的 SOAP 標頭區段中指定的使用者名稱和密碼的 XML 片段。
驗證服務	Web 服務測試工具會呼叫內部 cc-authenticate 使用您指定的使用者名稱和密碼提供服務。如果使用者名稱和密碼在伺服器上有效，Workday 會傳回階段作業票證，Web 服務測試人員會在傳送要求時將其新增至 SOAP 標頭。伺服器處理中包含攔截程式，可檢查提供的階段作業票證是否有效。 預設情況下，與 cc-authenticate 服務只進行一次即可重複使用。如果您要停止再啟動伺服器，請在每次傳送前選取重新連線至 cc-authenticate 服務，以便為每個要求建立新的連線。
基本驗證 (先佔式)	可讓您指定用於搶先式 HTTP 基本驗證的使用者名稱和密碼。

配置 Web 服務測試工具 X.509 驗證

開始之前

建立 Web 服務測試。配置整合系統使用者，其使用者名稱應與金鑰儲存區別名相對應，但不含 @tenant-name 後綴將 Web 服務的適當使用者型群組指派給整合系統使用者。

執行這項作業的原因和時機

您可以在 Workday Studio 的 Web 服務測試工具中配置多個安全選項。X.509 驗證 選項可讓您選取與整合系統使用者及相關聯的私人和公開金鑰組相對應的金鑰儲存區別名。

程序

1. 在 WS 測試人員 檢視中，按一下 傳送選項。
2. 選取 X.509 驗證 核取方塊。
3. 從 使用者名稱 下拉清單中選取使用者名稱。每個使用者名稱都與一個金鑰儲存區別名相關聯。

4. (選填)按一下 **管理金鑰** 以管理現有的 X.509 金鑰或產生新的金鑰對。Studio 在每個階段作業都會要求提供一次金鑰儲存區密碼。執行此任務時，請參考以下說明：

選項	敘述
產生 X.509 金鑰組	可讓您產生具有相關聯數位憑證的 X.509 金鑰組。
從金鑰儲存區移除金鑰	移除選取的金鑰組。
檢視公開金鑰 (憑證)	顯示所選金鑰組的公開金鑰。
匯入金鑰組	您可以： - 從 Java 金鑰儲存區匯入私人金鑰和 X.509 憑證。 - 匯入 OpenSSL PEM 格式的私人金鑰和 X.509 憑證。
重新命名別名	可讓您為選取的金鑰組重新命名金鑰儲存區別名。

配置 Web Service Tester Advanced WS-Security

開始之前

建立 Web 服務測試。

執行這項作業的原因和時機

您可以在 Workday Studio 的 Web 服務測試工具中配置多個安全選項。進階 WS-Security 配置 選項可讓 Web 服務測試人員讀取安全性配置檔案，該檔案指定了一系列的要求安全步驟和另一套回應安全步驟。其會對傳出的要求訊息套用安全步驟，並從傳入的回應訊息中移除安全編碼。伺服器會在接收時移除安全性編碼，並在回應時新增安全性編碼。

程序

1. 在 WS 測試人員 檢視中，按一下 傳送選項。
2. 選取 進階 WS-Security 配置 核取方塊。
3. 提供 工作區 或 外部 位置的 配置檔案 URL。
4. (選填)如果安全性配置檔案具有金鑰儲存位置等值的相對位置，請提供「配置基本 URL」。Studio 會將此 URL 附加到相對位置。
5. (選填)選取 「每次傳送前請重新載入安全性配置資訊」核取 方塊，即可為每個要求重新載入檔案。當您要變更配置檔案時，此選項會很有用。

概念：Web 服務測試

Workday Studio Web 服務測試工具可讓您根據 Web 服務說明 (WSDL) 或結構描述定義 (XSD)，建立範例 SOAP 或 XML 要求訊息。此工具也會顯示 Web 服務傳回的任何回應訊息。支援 HTTP 和 HTTPS。

下表匯總了啟動 Web 服務測試工具的 3 種主要方式：

方法	附註
從 Studio 功能表列	可讓您從工作區、URL 或外部檔案位置，選取要測試的 WSDL 或 XSD 檔案。您也可以使用此選項，在不參照任何特定檔案的情況下開啟 Web 服務測試工具。

方法	附註
從 結構探索器 檢視	可讓您為特定 WSDL 作業或結構描述元素產生 XML 要求 (非 SOAP 或 SOAP 封裝格式)。
使用 Workday Web 服務要求建置器 精靈	簡化特定 Get 作業和 Get 要求訊息的測試。

參照：Web 服務測試工具列按鈕

一般

按鈕	說明
傳送要求	呼叫 Web 服務。回應訊息會顯示在「回應」窗格中。
傳送選項	開啟「傳送選項」對話方塊，讓您配置安全選項。
中止要求	中止目前的要求。
重新載入 WSDL	重新載入驅動 Web 服務測試工具的 WSDL。適用於 Web 服務已變更的情況。
透過喜好設定選擇 XML 字型	可讓您設定顯示在「要求」和「回應」窗格中的 XML 字型屬性。
要求/回應記錄	提供以下權限：存取先前傳送的要求訊息清單，以便選取並重新傳送這些要求訊息。
僅限申請	僅顯示「要求」窗格。
僅限回覆	僅顯示「回覆」窗格。
水平版面配置	水平排列版面配置。
垂直版面配置	垂直排列版面配置。
說明	顯示 Web 服務測試工具的內容相關說明。

要求窗格

按鈕	說明
儲存要求	將要求訊息另存為 XML 檔案。
載入要求	載入要求訊息。
格式要求	將標準縮排套用至要求訊息。文字可以超出面板寬度。
還原原始或上次傳送的要求	對於未傳送的訊息，將文字還原為原始的未格式化且未包裝的狀態。覆寫任何手動編輯內容。對於已傳送的訊息，將文字還原為上次傳送動作之前的狀態。
新增附件	可讓您新增或移除檔案附件。若要在要求訊息中新增附件，請按一下 編輯，從工作區或外部檔案位置選取檔案，然後指定 名稱 和 MIME 類型。 若要從要求訊息中移除附件，請按一下「移除」。

按鈕	說明
	若要儲存回覆訊息中包含的附件，請按一下「回覆」窗格下方的開啟。
增加字型	增加「要求」窗格中的文字大小。
縮小字型	縮小「要求」窗格中的文字大小。

回覆窗格

按鈕	說明
儲存回覆	將回應訊息另存為 XML 檔案。
載入回覆	載入回應訊息。
設定回覆格式	對回覆訊息套用標準縮排。文字可以超出面板寬度。
包裝回覆	套用縮排，但也會在面板寬度的邊界內設定文字的格式。
還原回覆	對於回覆訊息，將文字還原為其原始的未格式化且未包裝的狀態。對於範例回覆，從文字窗格中移除載入的檔案。復原任何手動編輯。
回覆範例	針對從 WSDL 產生的 SOAP 要求訊息產生範例回應。
增加字型	增加「回覆」窗格中的文字大小。
縮小字型	縮小「回覆」窗格中的文字大小。

程序集偵錯

偵錯程序集

開始之前

- 建立程序集專案。
- 安全性：「整合」功能區中的「整合調試」網域。

執行這項作業的原因和時機

Workday Studio 可讓您偵測和偵錯程序集中的錯誤。您可以對已部署至 Workday 伺服器環境的程序集進行偵錯。如果組件包含參照靜態檔案的元件或步驟，請先確定參照的檔案路徑中沒有空格，然後再進行偵錯。

程序

1. 在「Project Explorer」檢視中，在組合專案上按一下滑鼠右鍵。
2. 選取 部署至 Workday。
3. 在「部署至 Workday」視窗中，請參考以下說明：

選項	敘述
Workday 環境	請選取要部署整合的目標 Workday 環境。

選項	敘述
部署中的集合	請選取集合。

- 按一下 完成。
- 在 Cloud Explorer 檢視中，選取並展開集合專案部署目標 Workday 環境的雲端伺服器節點。
- 選取並展開您部署至 Workday 的集合。
- 在集合中的整合節點上按一下滑鼠右鍵。
- 選取「調試整合」。在「調試整合」視窗中，請參考以下說明：

選項	敘述
啟動時間	用於啟動偵錯階段作業的目標 Workday 環境。
收款	您要偵錯的集合名稱。
專案	您要偵錯的專案名稱。
Workday-In	您正在偵錯的專案中 Workday-In 傳輸的名稱。
啟動參數	您正在偵錯的專案的啟動參數。 若要使用替代啟動參數進行偵錯，請按一下 選取常值，然後選取 CRF 傳回的 Workday ID。 如果您知道要將某個特定的 Workday ID 做為啟動參數，請按一下 編輯常值，並在「值」欄位中輸入該值。

- 按一下「調試」。

註：如果您在偵錯執行階段開始啟動時遇到問題，請考慮調整「階段作業重新連線重試次數」和「啟動等待迭代次數上限」的值。

階段作業重新連線重試次數 決定了 Studio 嘗試將偵錯階段作業連至偵錯執行階段的次數。「啟動等待迭代次數上限」決定 Studio 檢查調試執行階段是否已準備好接收要求的次數。請嘗試以 10 為單位遞增。

您可以在「調試配置」對話方塊中為現有的啟動配置調整這些值。

若要針對未來的啟動配置進行調整，請選取「視窗 > 喜好設定」> Workday > 程序集「偵錯 > 偵錯執行階段」。

結果

偵錯階段作業的結果會顯示在「主控台」檢視中。

概念：程序集偵錯器

Workday Studio 程序集偵錯工具是用於在 Workday 伺服器環境中偵錯程序集的工具。程序集偵錯工具可讓您：

- 檢查程序集訊息執行階段環境定義。
- 暫停組合件串連。
- 測試 MVEL 運算式。

如果您的程序集包含參照靜態檔案的元件或步驟，請先確定參照的檔案路徑中沒有空格，然後再進行偵錯。

註：您無法在正式環境用戶中偵錯程序集。請使用非正式環境進行偵錯。

設定斷點

斷點可讓您在程序集中的特定點暫停偵錯流程。若要在組合元素上設定斷點，請在元素上按一下滑鼠右鍵，然後選取「切換組合元件斷點」。在集合編輯器主檢視中，元素右上角會顯示一個藍色圓圈，表示有新的中斷點。

Studio 管理

Studio 喜好設定

概念：集合編輯器喜好設定

您可以使用 集合編輯器 喜好設定頁面，指定集合圖表的喜好設定。若要瀏覽至喜好設定頁面，請選取 視窗 > 喜好設定 > Workday > 組合編輯器。選取下列其中一個節點以指定相關的組合編輯器喜好設定：

- 色彩
- 字型
- 移轉
- 模組

色彩

「圖表色彩喜好設定」頁面可讓您為組合圖中的元件指定色彩喜好設定。

設定	說明
背景色彩	指定組合體元件的背景色彩。
線條色彩	指定組合體元件的線條色彩。
文字色彩	指定組合體元件的字型色彩。
步驟背景色彩	指定組合步驟的背景色彩。
標頭背景色彩	指定組合體元件標頭的背景色彩。

字型

「圖表字型喜好設定」頁面可讓您為組合圖中的元件指定字型喜好設定。

設定	說明
預設值	指定組合圖元件 ID 值的標準字型類型、樣式和大小。
步驟	指定組合步驟 ID 值的標準字型類型、樣式和大小。
條件式工具提示標頭	指定組合圖中顯示的工具提示標頭文字的標準字型類型、樣式和大小。
條件式工具提示內容	指定組合圖中顯示的工具提示內容的標準字型類型、樣式和大小。

移轉

在「移轉」頁面上，您可以開啟或關閉使用早期 Workday Studio 版本建立的舊版程序集專案的移轉功能。

如果已啟用移轉功能，Workday Studio 會在您開啟 assembly.xml 檔案時，自動將其組合配置移轉至最新版的 Studio。

若要關閉自動移轉，請先在程序集編輯器中清除啟用舊版程序集移轉核取方塊，然後再開啟或儲存 Assembly.xml 檔案。

模組

在您開始開發組合模組範本之前，請先在您的工作區或外部位置 (例如 Workday Studio 安裝中的資料夾) 建立用來儲存組合模組的資料夾。

集合編輯器的集合工具選用板中的「模組庫」僅會顯示 視窗 > 喜好設定 > Workday > 集合編輯器 > 模組頁面中指定的從模組資料夾載入的集合模組。按一下 瀏覽工作區 或 瀏覽檔案系統，找出模組資料夾並指定此資料夾的路徑。Studio 會顯示「模組」表格中指定資料夾內的所有組件模組 XMI (XML 中繼資料交換) 檔案。選取您要載入 模組庫中的每個組件模組旁的核取方塊。Studio 會執行範本驗證，且僅載入有效的 XMI 檔案。

在設計時，組件模組 精靈會提示您將模組 XMI 檔案儲存至工作區的資料夾。如果願意，您可以稍後將組件模組 XMI 檔案從工作區資料夾複製到您在此處指定的模組資料夾 (如果與工作區資料夾不同)。

概念：人脈喜好設定

「連線」喜好設定頁面可讓您設定整合雲端連線憑證。若要瀏覽「人脈」喜好設定頁面，請選取 視窗 > 喜好設定 > Workday > 人脈。您可以輸入 沙箱 環境和 正式作業 環境的憑證。您也可以將人脈新增至其他環境，以及將群組新增至您的人脈。

概念：部署喜好設定

在「部署」喜好設定頁面上，您可以指定部署至 Workday 環境時，是否要將專案來源代碼包含在 CLAR 中。為此，請選取 在部署的 CLAR 中包含來源代碼 核取方塊。

若要從 CLAR 中排除來源代碼，請清除此核取方塊。

如果您選取 在部署的 CLAR 中包含來源代碼 選項，則可以將來源匯入工作區中的專案中。為此，請在 Cloud Explorer 中以滑鼠右鍵按一下集合，然後選取「匯入來源」。

概念：HTTPS 喜好設定

在 HTTPS 喜好設定窗格上，您可以指定是否要在 Studio 中啟用信任庫憑證管理。如果您選取「在 Studio 中啟用信任庫憑證管理」核取方塊，當 Studio 嘗試透過 HTTPS 連線至 Workday 時，將會顯示訊息，提示您要信任用於沙箱環境或正式環境的伺服器憑證。您信任的憑證會顯示在「信任庫憑證」窗格中，您可以在此重新載入、移除所有憑證，或刪除選取的憑證。若要開啟「信任庫憑證」窗格，請展開左側導覽窗格中的 Workday > HTTPS 連結。

概念：Ivy 喜好設定

Workday Studio 提供了 Ivy 外掛程式，可讓您透過 Apache Ivy 管理專案依存關係。若要讓 Ivy 外掛程式能夠解析資源，您可以指定要在專案中使用的設定和屬性檔案。

您可以在 Workday > Ivy 喜好設定頁面上配置以下設定：

- Ivy 設定 URL：按一下 瀏覽...，然後選取 ivysettings.xml 檔案或指定 URL。設定檔案定義了 Ivy 引擎在解析資源時可載入的變數。
- Ivy 屬性檔案：按一下 瀏覽...，然後選取設定檔案可參照的 .properties 檔案。屬性檔案包含 Ivy 做為變數載入的屬性，格式為：

name=value

選取 從主目錄載入 ivy.properties 檔案 (如果有) 核取方塊，即可自動從您的主目錄載入 ivy.properties 檔案。

概念：連結喜好設定

「連結」喜好設定頁面會指定是否要對工作區中的連結資源，使用絕對路徑的路徑變數。若要導覽至「連結」喜好設定頁面，請選取 視窗 > 喜好設定 > Workday > 連結。「將 PATH 變數用於連結的資源絕對路徑」設定會自動啟用。若要停用此設定，請清除此核取方塊。

概念：移動設備清單 (MEL) 驗證喜好設定

您可以在「中小型機器學習清單驗證」喜好設定頁面配置中小型機器學習清單代碼的驗證。啟用 MVEL 驗證設定會自動啟用。若要停用此設定，請清除此核取方塊。

概念：套裝整合喜好設定

「套裝整合」喜好設定頁面僅供內部 Studio 開發人員用於套裝整合部署用途。內部 Studio 開發人員 (請參閱您的內部 維客) 以取得封裝整合支援的更多詳細資料。

概念：執行階段喜好設定

Workday Studio 會自動檢查 Workday 執行階段是否有更新。如果有任何可用的更新，Studio 會在背景中下載更新，並在安裝時通知您。您可以在 視窗 > 喜好設定的 > Workday > Runtime 頁面上配置下載、安裝和通知喜好設定。

概念：伺服器喜好設定

在「伺服器」喜好設定頁面上，當 Workday 集合從「伺服器」設定中移除時，您可以對其進行調整。此設定會自動停用，這表示如果集合已部署至所選的 Workday 環境，而您從「已配置」清單中移除該集合，則 Studio 不會取消部署該集合。若要啟用此設定，請選取此核取方塊。當您在 部署至 Workday 精靈上按一下 完成 時，若有任何集合在您的工作區中，但已不在「已配置」清單中，則 Studio 便會取消部署這些集合。

概念：啟動畫面登入喜好設定

啟動時，Workday Studio 會顯示起始畫面 (選用)，您可在其中輸入 Workday 環境的連線詳細資料。如果願意，您可以稍後在 人脈詳細資料中輸入詳細資料。若要關閉啟動時的初始畫面，請選取「啟動時不要顯示登入」核取方塊，然後按一下「套用」以儲存變更。

概念：狀態列喜好設定

「狀態欄」喜好設定頁面可讓您指定要儲存在「狀態欄」記錄清單中的狀態欄項目數上限，該清單顯示在 Studio 工作區的右下角。在「狀態欄記錄項目數上限」欄位中，輸入您要儲存的項目數上限。

概念：更新喜好設定

Workday Studio 支援透過 Workday Content Delivery Network (CDN) 為低頻寬領域的整合者提供更新。您可以在「視窗 > 喜好設定的 > Workday > 安裝/更新」頁面上啟用對 Workday CDN 的存取。

註：使用 CDN 需要在連接埠 443 上未受限制的出站存取權。

概念：XPath Explorer 喜好設定

XPath Explorer 喜好設定頁面可讓您在為「評估 XPath 至 XML 片段」選項產生 XML 片段時，指定要使用的名稱空間-字首對應。

若要新增對應，請按一下 新增...。隨即顯示「新增名稱空間字首對應」欄位，可讓您為所需的對應輸入字首和名稱空間值。

若要移除不需要的對應，請將其選取並按一下 移除。若要重新排列對應順序，請選取對應，並視需要按一下 向上 或 向下 鍵。

參照：程序集偵錯執行階段喜好設定

設定	說明
主機名	啟動偵錯階段作業的主機名稱。
HTTP 連接埠	將偵錯階段作業連至 Workday 的連接埠號碼。

設定	說明
階段作業連線重試次數	Studio 嘗試將偵錯階段作業連至 Workday 的次數。
最長啟動等待小版本	Studio 檢查 Workday 是否已準備好接收要求的次數。
接受所有憑證	指定 Studio 是否接受未簽署的憑證。
啟用 HTTPS Proxy 設定以進行偵錯	啟用偵錯階段作業的 HTTPS Proxy 伺服器設定。
從 Eclipse 複製 Proxy 設定	指定是否要從 Eclipse 複製偵錯階段作業的 Proxy 設定。
Proxy 主機	Proxy 伺服器主機名稱。
Proxy 連接埠	Proxy 伺服器偵聽的通訊埠。
Proxy 使用者名稱	Proxy 伺服器的使用者名稱。
Proxy 密碼	Proxy 伺服器的密碼。
等候前面的業務流程步驟逾時，例如擷取 (秒)	在恢復偵錯階段作業之前，Studio 會分配給相關業務流程整合的逾時期間。

參照：程序集偵錯檢視喜好設定

設定	說明
自動設定 XML 根部分的格式	如果您希望 Studio 自動為訊息的根部部分設定顯示的 XML 格式，請勾選此方塊。
線程暫停時，將程序集偵錯檢視置於最前面	如果您希望 Studio 在偵錯流程期間有執行緒暫停時，將「程序集偵錯」檢視置於最前面，請勾選此方塊。
當執行緒在錯誤處理程式上暫停時，將「例外」分頁置於最前面	偵錯流程期間，若有執行緒掛起，如果您希望 Studio 將「程序集偵錯」檢視中的「例外」分頁置於最前面，請勾選此方塊。
自動擷取長度小於 (KB) 的根部分	指定自動下載閾值，即 程序集偵錯 檢視可自動顯示的訊息根部分的檔案大小上限。 若要檢視超過自動下載限額的訊息根部分，您必須下載訊息，或將其另存為工作區或檔案系統中的文件。為此，請使用「程序集偵錯」檢視工具列按鈕。
當下載到程序集偵錯檢視的訊息長度超過 時，截斷訊息 (KB)	下載後在「程序集偵錯」檢視中顯示時，超過指定檔案大小的訊息會截斷。

參照：Cloud Explorer 喜好設定

設定	說明
刪除集合時要求確認	如果啟用此功能，系統會提示您確認選擇，然後才能從雲端刪除集合。
在 Cloud Explorer 中顯示註解	如果啟用此功能，註解會顯示在 Cloud Explorer 的節點上。

設定	說明
顯示 Workday 連接器內容	如果啟用此功能，Cloud Explorer 會顯示 Workday-in 傳輸的內容，例如啟動參數、屬性內容和對應。

參照：合併報告檢視器喜好設定

基本資料記錄欄	說明
呼叫	元件在整合事件期間收到的呼叫次數。
平均時間	元件的平均處理時間。
最長時間	組成要素的最長處理時間。
最短時間	組成要素的最短處理時間。
標準偏差時間	與組成要素平均處理時間的標準差。
總時間	組成要素的總處理時間。
平均徵才需求大小	元件的平均要求大小。
最大申請大小	元件的要求大小上限。
最小徵才需求大小	組成要素的要求大小下限。
標準偏差申請大小	與組成要素平均要求大小的標準差。
徵才需求總大小	組成要素的要求總大小。
平均報告人數	組成要素的平均回應大小。
人數上限	元件的回應大小上限。
人數下限	組成要素的最小回應大小。
標準偏差或大小	與組成要素平均回應大小的標準差。
響應總人數	組成要素的總回應大小。
最大記憶體	元件的最大記憶體用量。
最小記憶體	元件的最低記憶體用量。
平均記憶體	元件的平均記憶體使用率。
可用記憶體	元件的可用記憶體用量。

參照：Java 專案設定喜好設定

設定	說明
強制要求僅兼容目前工作台的預設 JRE	預設情況下，Studio 產生的任何 Java 代碼都與任何已安裝的 JRE 相容。但是，透過選取此核取方塊，您可以強制 Java Web 服務專案僅與特定的 JRE 相容，即預設的 Workbench JRE。
遵循參照專案中類別路徑項目的匯出設定	選取此核取方塊可在匯出 WSAR 或部署 Web 服務專案時，僅包含從參照專案中匯出的資源。

參照：通知喜好設定

設定	說明
啟用通知	Studio 會自動啟用此設定。清除此核取方塊可停用通知在 Studio 畫面上顯示。
淡出延遲	請輸入一個值 (以毫秒為單位)，以指定通知在淡出前保持顯示的時間長度。

參照：流程監控喜好設定

設定	說明
啟用自動重新整理	預設會啟用「流程監控」檢視的自動重新整理功能。若要停用此功能，請清除此核取方塊。
重新整理間隔 (秒)	請輸入值 (以秒為單位)，以指定自動重新整理之間的時間長度。使用右側的捲動按鈕來增加或減少值。
要重新整理的上次作用中流程數	請輸入您要重新整理的作用中流程數目。重新整理的流程是按時間倒序排列的目前和先前流程。
自動重新整理失敗時顯示錯誤對話方塊	啟用在自動重新整理功能失敗時顯示錯誤訊息。Studio 會自動啟用此功能。若要停用此功能，請清除此核取方塊。
在啟動整合後顯示檢視	當您啟動整合時，在前景顯示「流程監控」。若要停用此功能，請清除此核取方塊。

參照：Web 服務測試人員喜好設定

設定	說明
要求和回應 XML 顯示	指定用於顯示要求和回應 XML 資料的字型大小和樣式。當您變更字型詳細資料時，開啟的編輯器會立即更新。
顯示的訊息不會大於：	指定要在 Web 服務測試人員的「回應」面板中顯示的訊息大小上限。標準值為 1024 KB。任何大於指定值的訊息都會儲存為附件。您可以在本機儲存附件。

Studio 更新

更新 Studio

執行這項作業的原因和時機

就像所有客戶都使用的 Workday 應用程式只有 1 個版本一樣，Workday Studio 也只有 1 個有效的版本。您的 Studio 安裝版本必須為最新版本，才能與沙箱環境或正式環境互動。

程序

1. 如果有可用的更新，當您在啟動時或之後輸入用戶憑證時，Studio 都會通知您。請確認您要更新，然後按照「更新 Workday Studio」精靈中的步驟操作。

註：請勿獨立於 Studio 更新排程來更新基礎 Eclipse 平台。這麼做可能會導致整合失敗。

2. 如果您選擇不立即更新，則 Studio 會在狀態列中顯示紅色箭頭做為提醒。當更新保持未安裝狀態時，您仍然可以在本機開發程序集。但是，您無法連線至沙箱或正式環境用戶。若要稍後更新，請選取「Workday > 更新 Workday Studio」。

註：Studio 包含 Workday Runtime 的輕量級版本，可讓您測試和偵錯您的整合。為確保您始終可以存取正式環境用戶中可用的相同執行階段，Studio 會在啟動時檢查執行階段更新。如果有可用的，則會在背景中下載。若要調整此行為，請選取「視窗 > 喜好設定 > 執行階段 > 自動更新」。

Workday 程序集執行階段變更

概念：程序集版本

程序集版本管理可讓 Workday 企業服務母線 (ESB) 伺服器 and Studio 開發團隊隨著時間的推移在 Workday Runtime 程式設計模型內發展程序集，而不會影響現有使用者。例如，程序集版本管理可讓 Workday：

- 已棄用的功能
- 推出新功能
- 修改現有功能的語義

所有程序集都有一個 version 屬性內容。預設情況下，當您新建組合專案時，Workday Studio 會設定 version 的屬性內容值 assembly 將 Assembly.xml 檔案中的元素複製到最新版本的 Workday Runtime。格式為 year.drop-number，其中「Drop-number」對應於週數。範例：2016.11。

Studio 會在 Project Explorer 中的「Assembly」資料夾旁顯示此數字。

對現有的執行階段功能進行變更時，Workday 會確保其向後兼容。這表示所有現有的程序集會依照其版本號碼如先前一般運作。但是，預設情況下，新程序集的行為與目前的版本號碼相對應。通常，當 Workday 引入變更時，我們會在組合步驟或元件上定義新的屬性。較舊的程序集沒有此屬性內容，因此我們會根據版本號碼引入規則，為較舊的程序集預設值。我們還引入了一項規則，可為較新的程序集預設不同的值。身為開發人員，如果您想要採用舊版程序集中的新功能，您有多種選擇。包括：

- 保留舊版本並接受 Workday Runtime 指派的預設值。
- 保留舊版本，並依據您的需求為新屬性設定明確的值。
- 移轉至較新的版本並接受新的預設值。
- 移轉至較新的版本，並為新屬性設定明確的值。

參照：結構描述層級的版本變更

此表格匯總了每次版本更新的結構描述層級變更。

版本	組成	屬性內容/屬性	變更說明
8.0	a 同步中介和同步中介元件	處理下游錯誤	預設值已變更自 true 至 false 值為 false 表示此元件中的本機錯誤處理常式僅處理髮生於此元件步驟中的錯誤。 值為 true 表示此元件中的本機錯誤處理常式可處理下游錯誤。

版本	組成	屬性內容/屬性	變更說明
9.0	寫入	output-mime-type	預設值已變更自 text/plain 至 text/xml，與其他步驟一致。
11.0	xmldiff 步驟	複製到輸出	預設值已變更為 true。 值為 true 表示此步驟會將輸出複製到中介訊息。 值為 false 表示步驟會將輸出複製到變數。
12.0	http-in 傳輸	安全	預設值已變更自 false 至 null，表示除非另有指定，端點可透過 HTTP 和 HTTPS 使用。
	http-out 傳輸	error-as-response	預設值已變更自 false 至 true。 值為 true 表示當執行階段在傳輸中遇到錯誤時，會在目前的訊息中設定錯誤。先前，只能透過 BadRequestException 於 MediationContext。
	local-out、http-out、ftp-out、ftps-out、sftp-out、xmpp-out、custom-out、email-out 傳輸	ignore-dynamic-endpoints	預設值已變更自 false 至 true。 值為 true 表示傳輸裝置會忽略傳入訊息標頭中的動態 WS-Addressing (WSA) 值。 值為 false 表示傳輸會使用 WSA 值覆寫其「端點」屬性中的值。
13.0	商店	結構描述	預設值已變更自 urn:com.workday/bsvc/blob 至 [http://www.w3.org/2005/Atom]。 將文件新增至目前的整合事件時，PutIntegrationMessage 子組元件可直接使用 Atom。
	a 同步中介和同步中介元件	continue-after-error	預設值已變更自 recover 至 rewind。

版本	組成	屬性內容/屬性	變更說明
			值為 <code>rewind</code> 表示清除錯誤後，會從集合中的上一個元件繼續處理。 先前的預設值 <code>recover</code>，現已棄用。
	本地出站傳輸	<code>unset-properties</code>	預設值已變更自 <code>false</code> 至 <code>true</code>。 值為 <code>true</code> 表示在呼叫子組合件後重設傳輸的屬性內容。
	發生錯誤時錯誤處理常式元件	<code>rethrow-error</code>	預設值已變更自 <code>true</code> 至 <code>false</code>。 值為 <code>false</code> 表示執行時會在錯誤處理程式觸發時清除錯誤。 值為 <code>true</code> 表示錯誤處理常式必須自行清除錯誤。
14.0	<code>workday-out-soap</code>	<code>replace-with-soap-fault</code>	預設值已變更自 <code>false</code> 至 <code>true</code>。 值為 <code>true</code> 表示傳回傳回時，傳輸會以 SOAP 錯誤取代目前的訊息。
	<code>local-in</code>	存取	預設值已變更自 <code>public</code> 至 <code>private</code>。 值為 <code>private</code> 僅表示 <code>local-out</code> 可以在同一個集合中呼叫傳輸。 值為 <code>public</code> 表示任何程序集都可以呼叫傳輸。
	匯總器元件的訊息內容整理策略。	<code>output-mimetype</code>	預設值已變更自 <code>application/octet-stream</code> 至 <code>text/plain</code>。 實作 AggregationCollater 介面仍預設為 <code>application/octet-stream</code>。
18.0	商店	不可變	新屬性，指定是否可以直接修改 Blob 參照的二進位資料。預設為 <code>true</code>。 值為 <code>true</code> 表示 Blob 不可變，無法修改。

版本	組成	屬性內容/屬性	變更說明
			值為 false 表示 Blob 可變且可修改。 註：在 Workday 18 中，伺服器執行階段會使用舊版本號碼移轉現有程序集中的所有商店步驟，以使用這個新的預設設定，讓 Blob 成為不可變。如果您的程序集需要更新 Blob，請務必將「不可變」設為 false。
2016.45 註：在第 27 版之後，Workday 在表單中引入了新的命名方案。 year.week	PagedGet	is.paged.get.version	預設值已變更自 util.assemblyVersionAsWSVer 至 null。
2019.30	PagedGet	As_Of_Effective_Date As_Of_Entry_DateTime	如果未提供這些屬性的值，則會將其預設為目前的日期與時間 (PST)。
2020.09	ftp-out、ftps-out、sftp-out		Workday 現在使用 JSCAPE 9.3 版，可支援更廣泛的加密密碼。
2020.09	PrismAnalytics	wd.prism.component.api.hostname	Studio 現在可呼叫第 2 版的 Prism Analytics REST API。 元件名稱已從 PrismConnector 變更為 PrismAnalytics。

relatedlinks

相關參考資料

常見問題：整合的加密、憑證和密碼

參照：XML 層級的版本變更

下表匯總了每個版本更新的 XML 層級變更。

版本	先前的 XML 元素/屬性內容	取代為	變更的意義/原因
11.0 (這些變更均涉及 Workday-in 傳輸元件中整合系統定義的變更。)	simple-type-reference	simple-type	項目類型為聲明對象，而非參照對象。
	type-reference	class-report-field	新的元素名稱可更清楚地表達其含義。
	service	service-reference	未定義此項目，已參照此項目。已移除

版本	先前的 XML 元素/屬性內容	取代為	變更的意義/原因
			defined-by 屬性內容，已不再使用「」。
	integration-system/@template	已移除	不再適用。
	param/@required	param/@option	已變更布林值 required 的屬性內容 option 值為 的屬性內容 required，其中為 True。
	cloud/name	cloud/@name	不那麼冗長。
	integration/name	in-connector/@id	不那麼冗長。
	in-connector/id	private-key	private-key
12.0	key-file	private-key	影響 sftp-out 和 ftps-out 傳輸已重新命名屬性內容，以更好地反映其含義。
	public-key-properties	private-key-properties	已重新命名以更好地反映其含義。
	aggregator/@spawn-aggregated-processing	已移除	此功能可讓匯總者建立新任務或討論串，以處理匯總訊息。此功能並未用於 Workday 整合，且不適用於公開的 Workday Studio，因此已將其移除。
	store/@compress store/@encrypt	已移除	先前允許對商店步驟進行精細控制。但是，Workday 的整合政策是一律壓縮和加密，因此已將其移除。
	ftp-out/@tmp-directory sftp-out/@tmp-directory ftps-out/@tmp-directory	已移除	先前，這些傳輸會管理自己的臨時檔案，但經過重新設計以使用通用的 FileBackedManagedData (FBDM) 功能，因此已移除這些選項。
14.0	delivery-service/child::*	已移除	派遞服務的子元素定義了允許的傳輸。此類限制不應在設計時定義。這是執行階段配置任務，因此已從設計階段開發階段移除。

參照：Saxon XSLT 處理器變更

下表匯總了 Workday 對各種版本的 Saxon 和 XSLT 的支援。請注意，在第 27 版之後，Workday 在表單中引入了新的命名方案。year.week。

Workday 版本	支援的撒克遜版本	支援的 XSLT 版本	附註
<20	9.1	1.0, (2.0)	對 XSLT 2.0 版的有限支援。
20-2017.05	9.4	2.0, (3.0)	對 XSLT 3.0 版的有限支援。
2017.05-	9.7	3.0	支援 XSLT 串流。

註：如果您建立的集合中包含不同版本的程序集，則 Studio 會顯示警告訊息。這是因為不同版本的程序集需要的 Saxon 版本可能會衝突，從而導致執行階段的類別載入錯誤。最佳做法是對所有整合使用最新的程序集版本。

請注意以下事項：

- MVEL 中可用於存取訊息部分和變數的 XPath 方法支援 XPath 3.0。
- 該 mctx URL 通訊協定允許 XSL 開發人員寫入訊息部分和變數，例如：

```
<xsl:result-document href="mctx:vars/var1">...</xsl:result-document>
```

- 雖然 XQuery 不提供直接程序集支援，但您可以新增自訂步驟來呼叫 XQuery 指令碼。
- 在 2022.46 版或更早版本的程序集中，只有在符合以下條件時，Saxon 才會設為預設轉換工具：use.saxon.transformer 屬性設為 true 於中介環境定義中。在 2022.47 版或更高版本的程序集中，如果該屬性設為「」，則該撒克遜文會設為預設值 true 或未顯示。

註：XSLT 2.0 和 XSLT 3.0 是強類型。例如，不允許 xs:string 要評估「」在 XPath 運算式中的相等性。以下運算式，其中 stringParam 是 XSL 參數，則會導致撒克遜執行階段錯誤：

```
$stringParam = true()
```

XSL 編譯器在部署時不會遇到此問題，因為它無法預先預測參數的類型。

此強類型問題不會影響您現有的 XSL 文件和 Workday Studio 整合，除非您選擇啟用 XSLT 3.0，方法是將 XSL 文件中的 version 屬性值變更為：3.0。如果您遇到這些執行階段錯誤 (由您與整合一起部署的 XSLT 所導致)，您必須採取行動加以修正。

註：在列印業務表單的舊版 XSL 中，發現以下與 Saxon 升級相關的錯誤：

```
Failed to compile stylesheet.Xerrors detected. Invalid value for @case-order. Value must be one of (lower-first|upper-first).
```

如果您遇到此錯誤，請檢查 XSL 的排序元素是否包含以下屬性內容設定：

```
case-order="#default"
```

這個選填 case-order 「」屬性會決定排序時先列出大寫還是小寫字母。預設會先列出大寫字母。Saxon 剖析器僅接受大小寫順序的下列值：

- upper-first
- lower-first

若要修正此錯誤，請執行下列其中一項操作：

- 請從排序元素中移除現有的個案順序屬性內容，因為這是選用設定。
- 將個案順序值明確設為 upper-first。
- 將個案順序值明確設為 lower-first。

範例程序集

範例：執行 Hello Cloud 範例程序集

執行這項作業的原因和時機

Workday Studio 可讓您在 Workday 中部署和啟動整合。此範例示範如何執行基本的範例整合。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

程序

1. 在 Studio 工具列上，選取 協助 > /歡迎 > 範例 > Hello Cloud。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在「Project Explorer」檢視中，展開 HelloCloud 資料夾，然後連按兩下 Assembly 資料夾。
4. 在組合體編輯器工具列上，按一下 顯示註解。
5. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
6. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

7. 按一下 套用並關閉。
8. 在「專案資源管理器」檢視中，在 HelloCloud 資料夾上按一下滑鼠右鍵，然後選取 部署至 Workday。
9. 在「部署至 Workday」視窗中，選取 沙箱 環境。按一下 完成。
10. 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > HelloCloudCollection > HelloCloud。
11. 在 StartHere 整合節點上按一下滑鼠右鍵，然後選取「啟動整合」。
12. 按一下 啟動。

結果

整合會在您的 Workday 用戶中啟動。Studio 中的「流程監控」檢視會顯示整合的狀態。

下一步

登入 Workday 並存取「流程監控」任務，即可顯示範例組合件的整合事件詳細資料。

範例：執行 Hello Workday Web 服務範例程序集

執行這項作業的原因和時機

Workday Studio 可讓您連線至 Workday Web 服務。此範例示範 SOAP Web 服務要求訊息 Get_Workers_Request。使用 XSLT 3.0 串流處理要求。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

程序

1. 在 Studio 工具列上，選取 Workday Web 服務，> 歡迎 > 範例，> 歡迎使用。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在「Project Explorer」檢視中，展開 HelloWWS 資料夾，然後按兩下 Assembly 資料夾。
4. 在組合體編輯器工具列上，按一下 顯示註解。
5. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
6. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

7. 按一下 套用並關閉。
8. 在「Project Explorer」檢視中，以滑鼠右鍵按一下 HelloWWS 資料夾，然後選取「部署至 Workday」。
9. 在「部署至 Workday」視窗中，選取 沙箱 環境，然後按一下 完成。
10. 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > HelloWWSCollection > HelloWWS。
11. 在 HelloWWS_EntryPoint 整合節點上按一下滑鼠右鍵，然後選取「啟動整合」。
12. 按一下 啟動。

結果

整合會在您的 Workday 用戶中啟動。Studio 中的「流程監控」檢視會顯示整合的狀態。

下一步

登入 Workday 並存取「流程監控」任務，即可顯示範例組合件的整合事件詳細資料。

範例：執行常見使用模式範例程序集

執行這項作業的原因和時機

此範例示範一些常見的整合設計模式。其中包含多個單獨的處理鏈，每個處理鏈說明不同的模式。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

程序

1. 在 Studio 工具列上，選取 協助 > /歡迎 > 範例 > 常見使用模式。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在「Project Explorer」檢視中，展開「CommonUsagePatterns」資料夾，然後按兩下「Assembly」資料夾。
4. 在組合體編輯器工具列上，按一下 顯示註解。
5. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
6. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

7. 按一下 套用並關閉。
8. 在「專案資源管理器」檢視中，在「CommonUsagePatterns」資料夾上按一下滑鼠右鍵，然後選取 部署至 Workday。
9. 在「部署至 Workday」視窗中，選取 沙箱 環境，然後按一下 完成。
10. 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > CommonUsagePatterns , > CommonUsagePatterns。
11. 在條件處理 整合節點上按一下滑鼠右鍵，然後選取「啟動整合」。
12. 按一下 啟動。
13. 啟動以下每個整合節點：
 - 正在增強
 - 全域錯誤處理
 - 本地錯誤處理
 - 循環播放
 - 要求回應處理鏈
 - 回應鍊式
 - 路由
 - 可擴充的訊息處理
 - 簡單處理鏈
 - 子例程
 - 變數

Request-Response-Processing-Chain 和 Response-Chaining 整合有一個額外的啟動參數：Employee_ID。按一下 選取常值 以進行配置。

結果

整合會在您的 Workday 用戶中啟動。Studio 中的「流程監控」檢視會顯示整合的狀態。

下一步

登入 Workday 並存取「流程監控」任務，即可顯示範例組合件的整合事件詳細資料。

範例：執行偵錯範例程序集

執行這項作業的原因和時機

Workday Studio 可讓您偵測和偵錯整合中的錯誤。此範例示範如何偵錯範例整合。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合偵錯
- 整合事件

程序

1. 在 Studio 工具列上，選取「說明 > /歡迎 > 範例程序 > 集偵錯工具」。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在「專案資源管理器」檢視中，展開「AssemblyDebug」資料夾，然後按兩下「Assembly」資料夾。
4. 在集合編輯器上，以滑鼠右鍵按一下每個集合元素，然後選取「切換集合斷點」：
 - create-get-workers-request
 - store-summary-doc
5. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
6. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

7. 按一下 套用並關閉。
8. 在「Project Explorer」檢視中，以滑鼠右鍵按一下「AssemblyDebug 資料夾」，然後選取「部署至 Workday」。
9. 在「部署至 Workday」視窗中，選取 沙箱 環境，然後按一下 完成。
10. 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > 程序集偵錯 > 程序集偵錯。
11. 在「從這裡開始」整合節點上按一下滑鼠右鍵，然後選取「調試整合」。
12. 按一下「調試」。
13. 在「確認觀點切換」視窗中，按一下「是」。
14. 在 程序集偵錯 檢視中，選取 訊息根部分 分頁，然後 在工作區中按一下將訊息或附件儲存至文件。
15. 在「儲存」視窗中，輸入以下值：

選項	敘述
上層資料夾	AssemblyDebug
檔案名稱	rootpart.xml

16. 在「調試」檢視中，在「建立要求」- create-get-workers-request 斷點上按一下滑鼠右鍵，然後選取「恢復」。
17. 在「程序集偵錯」檢視中，選取訊息根部分分頁，然後在工作區中按一下將訊息或附件儲存至文件。
18. 在「儲存」視窗中，輸入以下值：

選項	敘述
上層資料夾	AssemblyDebug
檔案名稱	rootpart1.xml

19. 在「偵錯」檢視中，以滑鼠右鍵按一下 Store - store-summary-doc 斷點，然後選取「終止」。

結果

Studio 會將 XML 斷點檔案儲存至「AssemblyDebug」專案資料夾。您可以使用這些檔案對集合中的錯誤進行疑難排解。

下一步

開啟程序集的 XML 斷點檔案。

範例：執行 Workday-In 功能範例程序集

執行這項作業的原因和時機

Workday-in 傳輸元件在 Workday 中呈現為整合系統。此範例示範如何在 Workday Studio 組合中使用 MVEL 運算式，以參照整合系統的服務和啟動參數。其中包含 3 個獨立的處理鏈。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

程序

1. 在 Studio 工具列上，選取 協助 > 歡迎 > 範例 > Workday-In 功能。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在 Project Explorer 檢視中，展開 WorkdayInFeatures 資料夾，然後連按兩下 Assembly 資料夾。
4. 在組合體編輯器工具列上，按一下 顯示註解。
5. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
6. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

7. 按一下 套用並關閉。
8. 在「專案資源管理器」檢視中，在 WorkdayInFeatures 資料夾上按一下滑鼠右鍵，然後選取 部署至 Workday。
9. 在「部署至 Workday」視窗中，選取 沙箱 環境，然後按一下 完成。

10. 對於每個單獨的整合，請參考以下說明：

整合	附註
AttributesAndMaps	部署後，您可以在 Workday 中調整屬性內容和對應值。 - 存取「檢視整合系統」報告並選取 <i>WorkdayInFeatures-attr-maps</i>。 - 從 SampleAttrMapService 整合服務的相關動作功能表中，選取 Integration Generic Service > Configure Integration Attributes 或 Integration Generic Service > Configure Integration Maps。 - 視需要調整屬性內容和對應值。 。
啟動參數	在 Studio 的 Cloud Explorer 檢視中，展開以下節點：沙箱區 > WorkdayInFeatures > WorkdayInFeatures。 - 在 referenceParam 欄位上，按一下 選取常值，然後選取正職員工。 - 輸入 password 在 passwordParam 欄位中
序列產生器	部署後，您可以在 Workday 中配置序列產生器服務： - 存取「檢視整合系統」報告並選取 <i>WorkdayInFeatures-sequences</i>。 - 從 SampleSeqGenService 整合服務的相關動作功能表中，選取 整合序列產生器服務， > 配置整合序列產生器。 - 在「增量」欄位中，輸入 1。 - 在 格式/語法 欄位中，輸入 <code>file[Seq].csv</code>。

11. 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > WorkdayInFeatures > WorkdayInFeatures。

12. 在 Workday-in 傳輸節點上按一下滑鼠右鍵，然後選取「啟動整合」。範例：AttributesAndMaps。

13. 按一下 啟動。

結果

整合會在您的 Workday 用戶中啟動。Studio 中的「流程監控」檢視會顯示整合的狀態。

下一步

登入 Workday 並存取「流程監控」任務，即可顯示範例組合件的整合事件詳細資料。

範例：執行錯誤處理範例程序集

執行這項作業的原因和時機

此範例示範了在 Workday Studio 中處理錯誤的多種方法。其中包含 6 個不同的處理鏈，每個處理鏈說明不同的模式。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

程序

1. 在 Studio 工具列上，選取「說明 > /歡迎 > 範例 > 錯誤處理」。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在「專案資源管理器」檢視中，展開 ErrorHandler101 資料夾，然後連按兩下 Assembly 資料夾。
4. 在組合體編輯器工具列上，按一下 顯示註解。
5. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
6. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

7. 按一下 套用並關閉。
8. 在「Project Explorer」檢視中，在 ErrorHandler 資料夾上按一下滑鼠右鍵，然後選取 部署至 Workday。
9. 在「部署至 Workday」視窗中，選取 沙箱 環境，然後按一下 完成。
10. 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > ErrorHandler > ErrorHandler。
11. 在 Workday-in 傳輸節點上按一下滑鼠右鍵，然後選取「啟動整合」。範例：「全有或全無」。
12. 按一下 啟動。

結果

整合會在您的 Workday 用戶中啟動。Studio 中的「流程監控」檢視會顯示整合的狀態。

下一步

登入 Workday 並存取「流程監控」任務，即可顯示範例組合件的整合事件詳細資料。

範例：執行 AUnit 範例測試

執行這項作業的原因和時機

AUnit 是一個 Workday Studio 測試架構，可讓您對程序集執行單元測試。此範例示範如何執行 AUnit 測試。其中包含 8 個獨立的處理鏈。

程序

1. 在 Studio 工具列上，選取「說明 /> 歡迎 > 範例 > A 單元測試」。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在「Project Explorer」檢視中，展開「AUnit 101」資料夾，然後連按兩下「Assembly」資料夾。
4. 在組合體編輯器工具列上，按一下 顯示註解。

5. 執行 單位 101 範例測試。

- 在「Project Explorer」檢視中，展開 src 資料夾。
- 在 com.workday.example.aunit 資料夾上按一下滑鼠右鍵。
- 選取 以 > 單位單元測試的身分執行測試。

結果

Studio 會在「主控台」檢視中顯示範例測試的結果。

下一步

導覽至 JUnit 檢視，以顯示結果的詳細摘要。

範例：執行 FOPStep 範例程序集

執行這項作業的原因和時機

Workday Studio 可讓您在整合輸出中產生 PDF 檔案。此範例示範如何產生基本的 PDF 文件。其中包含 3 個獨立的處理鏈。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

程序

- 在 Studio 工具列上，選取「說明 > 歡迎 > 範例 > FOP 步驟」。
- 在「專案名稱」視窗中，按一下 完成。
- 在「專案資源管理器」檢視中，展開 FOPStep 資料夾，然後連按兩下 Assembly 資料夾。
- 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
- 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

- 按一下 套用並關閉。
- 在「專案資源管理器」檢視中，在 FOPStep 資料夾上按一下滑鼠右鍵，然後選取 部署至 Workday。
- 在「部署至 Workday」視窗中，選取 沙箱 環境。按一下 完成。
- 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > FOPStep > FOPStep。
- 在 Simple-Start 整合節點上按一下滑鼠右鍵，然後選取「啟動整合」。
- 按一下 啟動。
- 在「流程監控」檢視中，以滑鼠右鍵按一下整合，然後選取 Workday：整合 ESB 呼叫。

13. 在 Workday 檢視中，輸入以下值：

選項	敘述
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

14. 在 檢視背景處理流程 視窗中，選取 輸出檔案 分頁。

15. 在「報告和其他輸出檔案」表格的「檔案」欄上，按兩下 PDF 檔案。

16. 啟動 Invoice-Start 和 Kanji-Start 整合節點。

結果

Workday 會顯示新產生的 PDF 檔案。

下一步

下載並儲存 PDF 檔案。

範例：針對員工範例程序集執行擷取組織資料

執行這項作業的原因和時機

Workday 可讓您使用組織類型將員工分組。此範例示範如何執行整合，從 Workday 擷取員工的組織類型和正職員工 ID。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

程序

1. 在 Studio 工具列上，選取「協助 > 歡迎 > 範例 > 擷取員工的組織資料」。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在「專案資源管理器」檢視中，展開 OrgDataForWorkerForOrgType 資料夾，然後連按兩下 Assembly 資料夾。
4. 在組合體編輯器工具列上，按一下 顯示註解。
5. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
6. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

7. 按一下 套用並關閉。
8. 在「專案資源管理器」檢視中，在 OrgDataForWorkerForOrgType 資料夾上按一下滑鼠右鍵，然後選取「部署至 Workday」。
9. 在「部署至 Workday」視窗中，選取 沙箱 環境。按一下 完成。

10. 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > OrgDataForWorker > OrgDataForWorkerForOrgType。
11. 在 StartGetOrgForOrgType 整合節點上按一下滑鼠右鍵，然後選取「啟動整合」。
12. 在 啟動整合 視窗中，導覽至 啟動參數 區段。
13. 在 組織類型 欄位上，按一下 選取常值。
14. 在「選取類別報告欄位」視窗上，按一下 下一步。
15. 在「CRF」視窗中，選取組織類型。執行此任務時，請參考以下說明：

選項	敘述
名稱	組織類型的名稱。
WID	組織類型的 Workday ID。

16. 按一下 完成。
17. 在 EmployeeID 欄位上，按一下 選取常值。
18. 在「選取類別報告欄位」視窗上，按一下 下一步。
19. 在「CRF」視窗中，選取正職員工。執行此任務時，請參考以下說明：

選項	敘述
名稱	正職員工的姓名。
WID	正職員工的 Workday ID。

20. 按一下 完成。
21. 按一下 啟動。
22. 在「流程監控」檢視中，以滑鼠右鍵按一下您的整合。
23. 選取 顯示詳細資料檢視。
24. 在「流程詳細 資料」檢視中，展開 OrgType 和 EmployeeID 啟動參數欄位。

結果

Studio 會在「流程詳細資料」檢視中顯示員工的組織類型和正職員工 ID 值。

範例：執行 PrismAnalytics 範例集合

執行這項作業的原因和時機

此範例示範如何使用 PrismAnalytics「」元件，將 CSV 資料傳送至 Workday Prism Analytics。集合專案中提供了來源 CSV 檔案。在此範例中，請將這些檔案複製到 SFTP 伺服器，以供範例組套件擷取。

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

程序

1. 註冊 Prism API 用戶端並複製 PrismAnalytics 元件需要。
 - a) 在 Workday 中，存取「註冊 API 用戶端以進行整合」任務。
 - b) 向 Prism Analytics 註冊 API 用戶端，做為其 範圍 (功能區)。選取 包含 Workday 自有範圍 核取方塊。
 - c) 複製 用戶端 ID 和 用戶端密鑰的 值。
 - d) 從相關動作功能表中，選取「API 用戶端 > - 管理整合的重新整理權杖」。
 - e) 選取要與重新整理權杖相關聯的 Workday 帳戶。帳戶必須具有建立 Prism 的權限 表格 在 Workday 中
 - f) 產生重新整理權杖並複製值。
2. 取得 Workday REST API 端點。
 - a) 存取「檢視 API 用戶端」報告。
 - b) 複製 Workday REST API 端點 值。刪除 https:// 在開頭和 /ccx/api/v1/<tenant_name> 放在最後。
3. 在 Studio 工具列上，選取「說明/> 歡迎 > 範例」> PrismAnalytics。
4. 在「專案名稱」視窗中，按一下 完成。
5. 在 Project Explorer 檢視中，展開 PrismAnalytics 資料夾，然後連按兩下 Assembly 資料夾。
6. 將提供的範例 CSV 檔案複製到您的 SFTP 伺服器。
 - a) 在 PrismAnalytics 專案資料夾中，展開 ws > WSAR-INF > PrismAnalytics Sample CSVs。
 - b) 將這些檔案複製到您的 SFTP 伺服器：
 - states.csv
 - states2.csv
 - states3.csv
 - states4.csv
 - states5.csv
7. 在組合體編輯器工具列上，按一下 顯示註解。
8. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
9. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。
10. 按一下 套用並關閉。
11. 在「Project Explorer」檢視中，按一下滑鼠右鍵 PrismAnalytics 資料夾，然後選取 部署至 Workday。
12. 在「部署至 Workday」視窗中，選取 沙箱 環境，然後按一下 完成。
13. 在 Cloud Explorer 檢視中，啟動此範例中的其中一個整合項目。
 - a) 展開 沙箱 節點。
 - b) 展開 PrismAnalytics收款 集合節點
 - c) 展開 PrismAnalytics 專案節點
 - d) 在 Workday-in 傳輸節點上按一下滑鼠右鍵，然後選取「啟動整合」。範例：SingleCSV。
14. 按一下 啟動。

結果

Workday Prism Analytics 會建立一個 表格 使用您提供的 CSV 資料

第三方薪資介面範例

步驟：執行薪資介面範例

執行這項作業的原因和時機

在此範例中，名為 PayData 擷取的整合會擷取薪資群組中與單次給付相關的資料。第二個 Studio 整合會循環瀏覽整合事件文件，並使用 XSLT 3.0 串流處理這些文件。

程序

1. [建立薪資介面安全群組](#) 頁 300.
建立可存取薪資介面範例所需網域的安全群組。
2. 存取「建立整合系統使用者」任務。
建立 PayData 擷取整合的整合系統使用者。範例：PayData 擷取 ISU。
從整合系統使用者的相關動作功能表中，選取「安全設定檔 > 指派整合系統安全群組」。在「要指派的整合系統安全群組」提示上，選取薪資介面安全性。
安全性：「整合」功能區中的「整合安全性」網域。
3. [建立 PayData 擷取整合系統](#) 頁 301.
在 Workday 中建立擷取薪資群組的單次給付資料的整合。
4. [測試 PayData 擷取整合系統](#) 頁 302.
5. [部署 Studio PayData 整合](#) 頁 303.
部署 Studio 整合，循環執行 PayData 擷取整合事件文件，並使用 XSLT 3.0 串流處理這些文件。
6. 存取「建立整合系統使用者」任務。
建立 PayData 整合的整合系統使用者。範例：PayData ISU。
安全性：「整合」功能區中的「整合安全性」網域。
7. [將整合系統使用者指派給 PayData 整合](#) 頁 304.
8. [連結 PayData 擷取與 PayData 整合](#) 頁 304.
9. [啟動連結的整合](#) 頁 305.

建立薪資介面安全群組

開始之前

安全性：「系統」功能區中的「安全性配置」網域。

執行這項作業的原因和時機

您必須先在 Workday 中建立安全群組，然後才能在 Studio 中執行薪資介面範例。

程序

1. 登入您的 Workday 測試環境。
2. 存取「建立安全群組」任務。執行此任務時，請參考以下說明：

選項	敘述
用戶定義的安全群組類型	整合系統安全群組 (未受限制)
名稱	薪資介面安全性

3. 存取「檢視安全群組」報告。

4. 從 安全群組 提示中選取 薪資介面安全性。
5. 從安全群組的相關動作功能表，選取「安全群組 > 的安全群組維護網域權限」。
6. 從 允許「取得」存取權的「網域安全原則」提示中選取以下網域：
 - 薪資介面
 - 管理：組織整合
 - 員工資料：公開員工報告
 - 員工資料：薪酬 - 依組織

Workday 會提醒您必須啟用安全原則變更。

7. 存取「啟用待處理的安全原則變更」任務並輸入備註。
8. 選取「確認」核取方塊。

結果

Workday 會將您的新安全群組新增至每個相關功能區的網域安全原則。

下一步

建立整合系統使用者，以執行 PayData 擷取 整合。

建立 PayData 擷取整合系統

開始之前

安全性：「整合建置」和「整合」，請在「整合」功能區中配置 網域。

執行這項作業的原因和時機

在 Workday 中建立擷取單次給付資料的整合項目。

程序

1. 登入您的 Workday 測試環境。
2. 存取「建立整合系統」任務。
3. 完成「建立整合系統」任務時，請參考以下說明：

選項	敘述
系統名稱	PayData Extract
新增 - 使用範本	薪資介面

4. 在「配置整合服務」頁面上，為以下整合服務選取 已啟用 核取方塊：
 - 薪資介面/PI 變更偵測啟動參數
 - 薪資介面/PI 薪資資料區段欄位
 - 薪資介面/PI 檔案名稱

Workday 會提醒您，您必須為某些整合屬性指派值，並為整合系統配置序列產生器。

5. 在 PayData 擷取 整合系統的相關動作功能表上，選取「編輯 > Workday 帳戶」。
6. 在「Workday 帳戶」提示中，輸入 PayData 擷取 整合系統的整合系統使用者名稱。
7. 在「編輯整合系統的帳戶」頁面上，按一下 確定。
8. 在 PayData 擷取 整合系統的相關動作功能表上，選取「整合系統 > 配置整合屬性」。完成任務時，請參考以下說明：

選項	敘述
版本	新增列並選取值 25。

選項	敘述
輸出文件標記	新增一行，選取「建立文件標記」，然後輸入文件標記名稱 <i>PICOF</i> 。
主要薪資整合	請新增一行。
薪資代管廠商	新增列並選取任何供應商。

9. 在 PayData 擷取 整合系統的相關動作功能表上，選取「整合系統」> 配置整合欄位屬性。

10. 在 欄位配置 區段中，針對下列欄位選取「包含在輸出中」核取方塊：

- 代碼
- 日期
- 收入項或扣除項
- 金額

11. 在 PayData 擷取 整合系統的相關動作功能表上，選取「整合系統 > 配置整合序列產生器」。

12. 在「配置整合序列產生器」頁面上，請參考以下說明：

選項	敘述
增量	1
格式/語法	Payrollinterface-paydata-[Seq].xml

結果

Workday 會配置 PayData 擷取 整合系統。

下一步

測試 PayData 擷取 整合系統。

測試 PayData 擷取整合系統

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

執行這項作業的原因和時機

您必須先在 Workday 中測試該薪資介面的整合系統，然後才能在 Studio 中執行此薪資介面範例。

程序

1. 登入您的 Workday 測試環境。
2. 存取「檢視整合系統」任務。
3. 在「整合系統」提示中，輸入 PayData 擷取。
4. 在 PayData 擷取 整合系統的相關動作功能表上，選取 整合 > 啟動/排程。
5. 在「排定整合」頁面上，在 值 欄中輸入以下值：

選項	敘述
薪資群組	請選取任何有單次給付的薪資群組。
薪資期間選擇選項	請使用目前日期的薪資期間。

選項	敘述
上次成功執行	選取的日期必須在該日期之後有新的單次給付。

6. 啟動整合併重新整理直到完成。

結果

PayData 擷取 整合系統會在 Workday 中啟動。

下一步

從 Studio 將 PayData 整合部署至 Workday。

部署 Studio PayData 整合

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置

執行這項作業的原因和時機

StudioPayData 整合會循環執行 PayData 擷取 整合事件文件，並使用 XSLT 3.0 串流處理這些文件。您必須先將其部署至 Workday，然後才能啟動。

程序

1. 在 Studio 工具列上，選取 協助 > /歡迎 > 薪資介面 > 範例。
2. 在「專案名稱」視窗中，按一下 完成。
3. 在「專案資源管理器」檢視中，展開 PayData 資料夾，然後按兩下 Assembly 資料夾。
4. 在組合體編輯器工具列上，按一下 顯示註解。
5. 在 Cloud Explorer 檢視中，按一下 連線詳細資料。
6. 選取 沙箱 環境，並在「連線」視窗輸入以下值：

選項	敘述
用戶	用戶名稱。
使用者名稱	用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

7. 按一下 套用並關閉。
8. 在「專案資源管理器」檢視中，在 PayData 資料夾上按一下滑鼠右鍵，然後選取 部署至 Workday。
9. 在「部署至 Workday」視窗中，選取 沙箱 環境。按一下 完成。

結果

Studio 會將 PayData 範例整合部署至 Workday。

下一步

建立整合系統使用者以執行 PayData 整合。

將整合系統使用者指派給 **PayData** 整合

開始之前

安全性：整合，請在「整合」功能區中配置 網域。

執行這項作業的原因和時機

將整合系統使用者指派給 Workday 中的 PayData 整合系統。

程序

1. 在 Cloud Explorer 檢視中，展開以下節點：沙箱 > PayData > PayData。
2. 在 StartHere 整合節點上按一下滑鼠右鍵。
3. 選取 Workday：整合系統。
4. 在 Workday 檢視中，請考慮：

選項	敘述
使用者名稱	您的 Workday 用戶的使用者名稱憑證。
密碼	用戶的密碼憑證。

5. 在 PayData 整合系統的相關動作功能表上，選取「編輯 > Workday 帳戶」。
6. 在 Workday 帳戶 提示上，輸入 PayData 整合系統的整合系統使用者名稱。
7. 在「編輯整合系統的帳戶」頁面上，按一下 確定。

結果

Workday 會將整合系統使用者指派給 PayData 整合系統。

下一步

連結 PayData 擷取 和 PayData 整合

連結 **PayData** 擷取與 **PayData** 整合

開始之前

安全性：整合，請在「整合」功能區中配置 網域。

執行這項作業的原因和時機

連結 Workday 中的 PayData 擷取和 PayData 整合。

程序

1. 登入您的 Workday 測試環境。
2. 存取「檢視整合系統」任務。
3. 在「整合系統」提示中，輸入 PayData 擷取。
4. 在 PayData 擷取 整合系統的相關動作功能表上，選取「業務流程的 > 建立、複製或連結定義」。
5. 按一下「確定」，建立新的業務流程定義。
6. 在「編輯 PayData 擷取的業務流程整合流程事件 (最上層)」頁面的「業務流程步驟」分頁上，按一下 新增列。在新列上，請考慮：

選項	敘述
順序	C

選項	敘述
類型	整合

Workday 會警告您尚未完成配置。

- 按一下 配置整合系統。
- 在「整合」提示上，選取 PayData。
- 在 配置整合步驟 頁面的 整合屬性內容 分頁上，請參考以下說明：

選項	敘述
值類型	在執行階段確定值
值	可交付文件

結果

Workday 會將 PayData 擷取 和 PayData 整合連結在一起，以提供所需的輸出。

下一步

在 Workday 中啟動連結的 PayData 擷取 和 PayData 整合。

啟動連結的整合

開始之前

安全性：「整合」功能區中的以下網域：

- 整合建置
- 整合配置
- 整合事件

執行這項作業的原因和時機

啟動已連結至 PayData 整合的 PayData 擷取 整合。

程序

- 登入您的 Workday 測試環境。
- 存取「檢視整合系統」任務。
- 在「整合系統」提示中，輸入 PayData 擷取。
- 在 PayData 擷取 整合系統的相關動作功能表上，選取 整合 > 啟動/排程。
- 在「排定整合」頁面上，在 值 欄中輸入以下值：

選項	敘述
薪資群組	請選取任何有單次給付的薪資群組。
薪資期間選擇選項	請使用目前日期的薪資期間。
上次成功執行	選取的日期必須在該日期之後有新的單次給付。

- 啟動整合併重新整理直到完成。
- 在 事件文件 區段上，記下 Workday 產生的 CSV 檔案。範例：payrollinterface-paydata-1.csv。