

資訊中心

8.0 版 此資訊正在進行開發，而且可能隨時變更；*ibm.com* 會提供已發行的版本。

附註

在使用此資訊與其支援的產品之前，請先閱讀第 537 頁的『注意事項』中的資訊。

目錄

第 1 章 關於 IBM Rational Build Forge 1

Rational Build Forge 簡介	1
Build Forge 產品版本	1
程序自動化	2
Build Forge 概念和物件	3

第 2 章 Build Forge 入門 7

存取和使用主控台	7
過濾和排序清單	10
建立 Hello World 專案	10
設定伺服器	11
定義專案	13
執行工作	13
檢視工作結果	14
專案範例	14

第 3 章 IBM Rational Build Forge 文件注意事項 17

第 4 章 需求 21

硬體、軟體與資料庫需求	21
資料庫需求	21
Installation Manager 需求	21
產品版本的授權需求	21
在安裝期間指定授權檔	22
配置 Build Forge 的 Rational License Server	22
取得授權碼並設定 Rational License Server	24
IPv6 支援的網路需求	25
國家語言支援需求	26
「管理主控台」和「代理程式」的語言設定	26
資料庫主機的國際資料支援	27
在「管理主控台」中變更使用者語言喜好設定	27
判斷 UNIX/Linux 主機的語言/字集	28
確定 Windows 主機的語言字碼頁	28

第 5 章 安裝規劃 29

元件	29
部署類型	30
安裝方法	30
配置選項	30
與其他產品整合	31

第 6 章 前置安裝設定 33

國際資料設定	33
資料庫備份（必要）	34
資料庫設定	34
DB2 設定	35
設定 Microsoft SQL Server	38
MySQL 設定	43
Oracle 設定	46
安全設定	52

第 7 章 安裝管理主控台 55

利用啟動程式來啟動 Installation Manager	55
從產品 DVD 執行啟動程式	56
從下載的套件執行啟動程式	56
Installation Manager 中的安裝步驟	57
後置安裝核對清單	61
增加 Linux 的檔案控點數目	62

第 8 章 替代安裝方法 63

使用您自己的元件安裝	63
必備條件	63
國際資料支援	63
資料庫安裝與配置	64
Apache HTTP Server 安裝與配置	65
PHP 安裝與配置	67
Apache Tomcat 安裝及配置	70
手動安裝 Installation Manager	72
啟動 Installation Manager	72
指定儲存庫 URL	72
執行無聲自動安裝產品元件	73
以 Installation Manager 建立回應檔	73
以無聲自動模式安裝及執行 Installation Manager	74
執行產品元件的無聲自動升級	75
利用 Installation Manager 來已建立更新回應檔	75
以無聲自動模式來執行更新安裝	76
在 VMware 上安裝 Build Forge 系統	76
在 System z 上的 Linux 中安裝管理主控台	77

第 9 章 在「管理主控台」中配置其他特性 83

Build Forge 配置檔 (buildforge.conf)	83
更新 buildforge.conf 檔	83
Buildforge.conf 參照	84
配置「管理主控台」以使用替代埠	85
配置備用功能	85
關於備用功能	85
安裝冗餘系統	86
使用備用功能	87
啟用 IPv6 網路支援	87
修改 httpd.conf	87
安全特性	88
實作單一登入	88
啟用 SSL 和 HTTPS	101
啟用密碼加密	119

第 10 章 安裝代理程式 131

在 Windows 平台上安裝代理程式	131
使用者模式代理程式	132
在 Windows 作業系統上執行代理程式的無聲自動安裝	132
將代理程式安裝在 UNIX 和 Linux 系統上	133
在 System i 平台上安裝代理程式	134

在 System z 平台上安裝及執行代理程式	135
在 z/OS 上對代理程式安裝作業進行疑難排解	137
代理程式更新及部署公用程式	138
支援的作業系統	139
安裝	139
執行 UNIX 或 Linux 的目標電腦連線功能設定	139
執行 Windows 的目標電腦連線功能設定	139
用法概觀	142
在 XML 檔中指定目標系統	142
執行 agentupdate 公用程式	146
agentupdate 公用程式參照	147
XML 檔案參數、指令行選項及環境變數的優先順序	148
透過日誌檔檢視公用程式動作	149
疑難排解	149
執行代理程式	154
在 Windows 上執行代理程式	154
在 UNIX、Linux 及 MacOS 上執行代理程式	154
在 System i 上執行代理程式	155
bfagent 參照	156
配置代理程式	156
找出代理程式配置檔	156
變更代理程式埠	157
配置不同的 Shell	157
在「網路共用檔案系統」上執行代理程式指令 (Windows)	157
配置代理程式之間的直接檔案傳輸	158
觸發變數和代理程式效能	159
bfagent.conf 參照	159
對代理程式進行疑難排解	168
測試主機名稱解析	168
測試連線	169
在 Windows 上進行代理程式疑難排解	170
在 UNIX、Linux 或 MacOS 上對代理程式進行疑難排解	170

第 11 章 後置安裝作業 173

啟動與停止引擎	173
在 Windows 上啟動和停止引擎	173
在 UNIX or Linux 系統上啟動和停止引擎	173
設定使用者	174
root 使用者	174
建立及編輯使用者	174
驗證安裝架構	176
伺服器鑑別	176
配置伺服器	176
建立測試專案	177
執行專案	177
日誌的位置與設定	178
一般問題的疑難排解	179

第 12 章 從舊版升級 181

升級必要條件	181
升級 7.1 版主控制台	181
執行更新安裝	181
升級到 7.1.3.1 版之後複製配接器	183

升級 7.0.2.x 版主控制台	184
升級代理程式	184

第 13 章 解除安裝產品元件 185

使用 Installation Manager 來解除安裝產品	185
在 Installation Manager 安裝失敗時手動解除安裝產品	186
解除安裝後清除 Build Forge 檔案	186
使用 Installation Manager 來重新安裝產品並使用現有的 DB2 Express	186
解除安裝 Windows Build Forge 代理程式	187
解除安裝 UNIX 或 Linux Build Forge 代理程式	187

第 14 章 版本注意事項 191

版本注意事項 - IBM Rational Build Forge 8.0 版	191
---	-----

第 15 章 「儀表板」標籤 195

第 16 章 管理 197

關於管理	197
存取群組	197
「存取」概觀	199
「存取」範例：讓群組有能力執行工作	200
團隊和專案安全計劃	200
管理存取內容	200
使用者	201
使用者內容	201
管理使用者	203
利用使用者類型控制存取權	203
帳戶鎖定	204
root 使用者	204
API 使用者	205
許可權	205
許可權練習	206
LDAP 整合	206
關於 LDAP 整合	206
LDAP 網域內容	207
作業	210
系統配置設定	212
其他系統設定	221
審核記載	222
訊息	225
翻譯的訊息	226
訂閱工作狀態的 RSS 資料資訊來源	227
過濾系統訊息的 RSS 資料資訊來源	227
「安全」畫面	228
「金鑰儲存庫」畫面	229
SSL 畫面	230
SSO 畫面	232
管理授權	232
輸入新授權碼	233
記號授權日誌	233
管理 Perl 引擎	233
暫停 Perl 引擎	233
啟動和停止 Perl 引擎	234
管理資料庫	235

刪除資料庫日誌檔	235
錯誤訊息	235
沒有作用中的步驟	235
授權碼無效或者Build Forge 授權碼毀損或遺漏。	235
需要資料庫授權	235
第 17 章 伺服器	237
關於伺服器資源	237
關於「伺服器」畫面	237
建立伺服器	238
測試伺服器	239
變更伺服器	240
啟用及停用伺服器	240
限制伺服器上的並行工作	240
重設工作計數	241
將伺服器上的工作計數重設為零	241
將所有伺服器上的工作計數重設為零	241
伺服器狀態的 RSS 資料資訊來源	241
伺服器鑑別	242
關於伺服器鑑別	242
建立伺服器鑑別	243
置換伺服器鑑別	243
容許使用受限的伺服器鑑別	244
選取元	244
關於選取元	244
選取元設定實務	245
選取元的變數類型	246
選取元變數的比較規則	246
合格伺服器資源的選取元評量方式	247
收集器	247
關於收集器	247
關於「收集器」畫面	248
建立收集器	248
資訊清單和動態選取伺服器	249
檢視資訊清單	250
手動重新整理資訊清單	250
設定伺服器資訊清單的更新頻率	251
靜態與動態選取伺服器的設定範例	251
內容參照	252
使用 Snapshot 來建立選取元的新實例	256
選取元 Snapshot 概觀	256
選取元 Snapshot 規劃	258
從現有的選取元或選取元 Snapshot 來建立選取元 Snapshot	258
變更預設選取元 Snapshot	259
變更選取元 Snapshot 的 Snapshot 名稱	259
存取和檢視選取元 Snapshot 集當中的 Snapshot	260
刪除選取元 Snapshot	261
第 18 章 使用環境	263
關於環境	263
環境繼承	263
特殊的繼承案例	264
啟動工作時變更專案變數	264
關於變數	265
解譯步驟中的變數	265

解譯未宣告的變數	266
關於「環境」畫面	266
「詳細資料」標籤	267
Snapshot 標籤	269
建立環境	269
使用變數	269
建立變數的下拉清單	270
併入其他環境	271
在步驟執行期間變更變數值	271
對映 Windows 磁碟機	271
在變數中使用點指令	273
系統變數參照	273
觸發變數參照	276
環境 Snapshot	280
「環境 Snapshot」概觀	281
環境 Snapshot 規劃	282
建立環境 Snapshot	282
變更預設環境 Snapshot	283
變更環境的 Snapshot 名稱	283
存取和檢視 Snapshot	284
刪除環境 Snapshot	284
第 19 章 使用專案	287
關於專案	287
關於「專案」畫面	287
變更專案內容	289
複製專案	291
刪除專案	292
將步驟固定在伺服器中	292
鏈：另一個專案或程式庫的條件式執行	293
鏈結的專案中的環境變數繼承	294
啟用等待時取消鏈結的專案	294
定義標籤	294
編輯專案的標籤格式	294
同步化標籤	295
系統定義的標籤變數	296
建立或編輯標籤變數	297
程式庫	297
關於程式庫	298
複製程式庫	298
日誌過濾器	299
關於日誌過濾器	299
建立日誌過濾器	299
將日誌過濾器指派給步驟	300
過濾器型樣	301
過濾器動作	302
過濾器通知	303
錯誤臨界值	304
錯誤與警告計數	304
類別	304
關於類別	304
設定通知	307
關於通知範本	307
配置您的 SMTP 伺服器	310
設定專案和步驟的通知內容	310
通知練習	310

自訂通知範本	310
列入專案的通知	314
使用 Snapshot 為專案建立新的實例	314
專案 Snapshot 概觀	314
專案 Snapshot 規劃	315
專案 Snapshot 選項	316
驗證與編輯 Snapshot 許可權的存取群組	317
從現有的專案或專案 Snapshot 來建立專案 Snapshot	317
變更預設專案 Snapshot	318
變更專案 Snapshot 的 Snapshot 名稱	319
存取和檢視專案 Snapshot 集當中的 Snapshot	319
啟動預設專案 Snapshot 的工作	320
啟動非預設專案 Snapshot 的工作	320
刪除專案 Snapshot	321

第 20 章 使用步驟 323

關於步驟	323
關於「步驟」畫面	323
新增步驟	326
編輯步驟	326
停用步驟	327
其他步驟作業	327
控制執行流程	327
步驟如何執行	328
列入：併入專案或程式庫的步驟	329
步驟的通過鏈和失敗鏈	330
執行緒作業：平行執行步驟	330
將步驟廣播至多部伺服器	331
執行條件式步驟	332
執行 While 迴圈	333
條件函數	333
從步驟啟動專案	335
自訂日誌輸出	335
為步驟的日誌輸出建立標籤	335
以顏色或作用中鏈結來強調顯示步驟輸出	336
使用工作資料	337
在專案檔中內嵌建置號碼	337
在工作期間變更建置標籤	338
在工作期間變更環境變數值	338
使用暫存器	339
專案暫存器	340
在步驟中將檔案複製到伺服器資源，以及從伺服器資源複製檔案	341
在伺服器資源上啟用檔案複製功能	341
從伺服器取得檔案	341
將檔案放置在伺服器上	342
配置直接檔案傳輸	342
疑難排解步驟的處理	342
工作沒有處理 ANT build 指令後面的任何步驟指令	342
沒有執行 Windows 批次指令後面的步驟指令	343
點指令參照	343
點指令語法	343
.bom	344
.bomexport	345

.break	345
.bset	345
.buildstatus	347
.date	347
.defect	348
.drill	349
.edit	350
.email	351
.expand	351
.export	351
.get	351
.load	352
.lock	355
.mkdir	355
.monitor	355
.pack	355
.pop	356
.poptag	357
.purge	357
.push	357
.put	358
.rem	358
.report	358
.retag	359
.retry	359
.rget	359
.rmdir	360
.rput	360
.run 及 .runwait	361
.scan	362
.semget	363
.semput	363
.set	363
.sleep	364
.snapshot	364
.source	365
.stop	366
.strsub	366
.test	367
.tset	367
.unlock	368

第 21 章 使用工作 369

關於工作	369
關於「起始」畫面	369
關於「工作」畫面	369
執行工作並檢視結果	371
啟動工作	371
檢視工作結果	373
工作狀態的 RSS 資料資訊來源	374
重新啟動失敗的工作	374
使用資料清單	376
排定工作	378
排程參數	380
排定清除工作的類別	382
管理工作	382

鎖定工作	382
刪除工作	383
工作的工作目錄	384
號誌	386
第 22 章 使用報告	389
關於報告	389
顯示報告輸出中資料的必備條件	389
顯示報告中資料所需的存取權	390
將內建報告的結果匯出至 CSV 檔	390
效能的標準報告	390
檢視專案的工作效能統計資料	390
檢視專案的工作持續時間	391
依專案來檢視步驟和伺服器的效能	391
查詢的預先定義報告	392
檢視選取元使用率歷程	392
依伺服器來檢視現行伺服器資訊清單	393
檢視工作的通過/失敗/警告結果	393
檢視某個日期範圍中工作的伺服器使用率	394
利用工作檔的 MD5 值來搜尋工作檔	395
使用「快速報告」來建立報告	395
「快速報告」的報告群組許可權	396
「快速報告」的報告類型參照	396
「快速報告」的報告格式與呈現參照	401
範例報告參照	405
使用提供的報告類型來建立報告	407
新增報告輸出至工作 BOM	408
在「快速報告」中修改與管理報告	408
第 23 章 使用公用程式	411
存取和執行指令行公用程式	411
指派隱藏移轉	411
bfassignhiddenmigration 參照	411
匯出專案	413
bfexport 參照	413
使用 .export	415
匯入專案	415
使用「匯入」公用程式來匯入專案與其他物件	416
bfimport 參照	417
如何將存取群組指派給匯入的物件	420
存取群組指派和安全	420
匯入時重新命名與取代物件	420
移轉舊式 Perl 工作控制器專案	422
Bfperlprojectmigration 公用程式參照	422
第 24 章 在「使用者介面配置」標籤中 鏈結至 Web 資源	425
第 25 章 Build Catalyst	427
支援的作業系統	427
安裝概觀	427
在 Linux 和 Solaris 作業系統上安裝 Build Catalyst	428
將 Build Catalyst 安裝在 Windows 上	429
Build Catalyst 範例	429
rafmake 公用程式參照	432

Build Catalyst 環境變數	434
-------------------------------	-----

第 26 章 Rational Build Forge 整合方法 437

使用指令行介面整合	437
使用配接器整合	438
配接器需求	438
配接器	438
配接器作業概觀	446
核心配接器作業	449
更新 ClearQuest 建置記錄	454
進階配接器作業	456
配接器參照	460
配接器 XML 參照	463

第 27 章 Rational Build Forge 與 Rational ClearCase 471

第 28 章 Rational Build Forge 與 Rational ClearQuest 477

第 29 章 Rational Build Forge 與 CVS 483

第 30 章 IDE 整合 485

關於 IDE 整合	485
測試專案的特殊變數	485
適用於 Eclipse 及 Rational Application Developer 的 外掛程式	486
安裝 Eclipse 或 Rational Application Developer 的外掛程式	487
啓用 SSL 時的替代安裝	488
使用 Eclipse 及 Rational Application Developer 的外掛程式	489
Rational Team Concert 的外掛程式	490
使用 Rational Team Concert 外掛程式	490
Rational Team Concert 外掛程式的疑難排解	491

第 31 章 Rational Build Forge 與 Performce 493

第 32 章 Rational Build Forge 與 Rational Quality Manager 495

第 33 章 Rational Build Forge 和 Rational Team Concert 497

Rational Team Concert 3.x 版	497
Rational Team Concert 1.x 及 2.x 版	497
安裝 Rational Team Concert 伺服器延伸	498
安裝 Rational Team Concert 的用戶端外掛程式	499
配置 Rational Team Concert 配接器	501

第 34 章 Rational Build Forge 與 StarTeam	505
---	------------

第 35 章 Rational Build Forge 與 Subversion	507
---	------------

第 36 章 Rational Build Forge 與 WebSphere 產品	509
---	------------

使用 WebSphere Application Server 來取代 Apache Tomcat.	509
配置 Websphere Application Server 來執行 Build Forge 應用程式.	509
在 Websphere Application Server 中為 Build Forge 配置 Java 2 安全.	512
在 Websphere Application Server 中為 Build Forge 配置 SSL 與密碼加密支援.	512
使用 Websphere Application Server 來執行 Build Forge 應用程式的啟動順序.	513
使用 IBM HTTP Server 來取代 Apache HTTP Server.	513
設定經過前置編譯的 PHP 檔.	514
編輯 IBM HTTP Server 的配置檔.	515
在 PHP 中識別 Proxy 伺服器.	516
配置 IHS 的 SSL.	516
Windows：取代 DLL.	518
AIX 系統：其他設定.	518

第 37 章 使用 API	519
----------------------	------------

建立 API 程式的 Build Forge 使用者	519
Java 用戶端 API	519
取得 Java 用戶端 API 套件.	520
設定 Java 用戶端 API	520
Perl 用戶端 API	520
取得 Perl 用戶端 API 套件.	521
設定 Perl 用戶端 API.	521

第 38 章 判斷「管理主控台」的版本號碼	523
------------------------------	------------

第 39 章 診斷	525
------------------	------------

Build Forge 服務層除錯診斷	525
Build Forge 預設日誌.	525
配置 Java 引擎日誌	526
啓用 Build Forge 代理程式活動記載.	526
診斷效能問題	526

第 40 章 效能和可調整性	527
-----------------------	------------

第 41 章 隨著產品一起安裝的可執行指令	529
------------------------------	------------

第 42 章 名詞解釋	531
--------------------	------------

工作 (job)	531
介面 (interface)	531
引擎 (engine)	531
存取群組 (access group)	531
外掛程式 (plug-in)	531
代理程式 (agent)	531
收集器 (collector)	531
伺服器 (server)	532
步驟 (step)	532
步驟日誌 (step log)	532
使用者 (user)	532
服務 (services)	532
保存 (archive)	532
除去 (clobber)	532
信號交換 (handshake)	533
通知範本 (notification templates)	533
配接器 (adaptor)	533
執行緒作業 (threading)	533
專案 (project)	533
動態 (dynamic)	533
程式庫 (library)	533
資料庫 (database)	533
資訊清單 (manifest)	533
號誌 (semaphore)	534
管理主控台 (Management Console)	534
輕量型目錄存取通訊協定 (Lightweight Directory Access Protocol, LDAP)	534
選取元 (selector)	534
靜態 (static)	534
環境 (environments)	534
類別 (class)	534
攔截程式 (interceptor)	535
BOM	535
Snapshot	535

附錄. 注意事項	537
-----------------	------------

商標	538
----	-----

第 1 章 關於 IBM® Rational® Build Forge®

在安裝及使用 Rational® Build Forge® 之前，請閱讀並理解基本概念。您也可以查看各種產品版本。

Rational Build Forge 簡介

IBM® Rational® Build Forge® 提供建置自動化及管理功能，以增速及簡化軟體和系統遞送。

Rational® Build Forge® 產品提供可以建立軟體組合線的調適性執行架構，以實現軟體開發與遞送的自動化並進行加速。它還可以編排、管理及追蹤組合線內的所有程序。現在，您可以實現重複作業的標準化及自動化，發現開發瓶頸，確定特定專案之趨勢，以及管理相符性指令。

Build Forge® 產品會與您的現行環境進行整合，支援主要開發語言、Script、工具和平台。它可讓您在針對程序自動化、加速、通知和排程增加有價值的功能時，繼續利用現有投資，例如資產、Script 軟體及硬體。

Rational® Build Forge® 的調適性執行架構會自動執行軟體開發及建置程序。

- 統一開發及測試中的自動化，刪除部門孤島間的延遲及切換
- 透過標準自動化系統，控制各部門中的現有工具
- 系統性減少或移除對於複雜或舊 Script 的依賴
- 在平台與產品之間，實現一致、可測量及可改良的程序
- 啟用開發者自助式服務，取消對建置團隊的依賴
- 只需按一下滑鼠，即可建立平行的自動化活動
- 提供程序及結果的一致性視圖，以便您輕鬆查看狀態及對問題進行疑難排解
- 透過自動產生的一致性報告，簡化顯示合規性程序
- 透過在相同時間內啟用更多程序，改良質量
- 提供適用於普及平台的代理程式，對其他平台提供原始檔套件支援

Build Forge 產品版本

以下是可用的產品版本：

- IBM® Rational® Build Forge® Standard Edition
- IBM Rational Build Forge Enterprise Edition
- IBM Rational Build Forge Enterprise Plus Edition

下該列出各版不同的元件或特性。

元件或功能	Standard Edition	Enterprise Edition	Enterprise Plus Edition
管理主控台 (Management Console)	Microsoft Windows、UNIX 或 Linux	Windows、UNIX 或 Linux	Windows、UNIX 或 Linux

元件或功能	Standard Edition	Enterprise Edition	Enterprise Plus Edition
資料庫	所有支援的資料庫	所有支援的資料庫	所有支援的資料庫
代理程式	所有支援的作業系統	所有支援的作業系統	所有支援的作業系統
授權伺服器	需要（25 個並行使用者）	需要（150 個並行使用者）	不需要（250 個並行登入）
配接器工具集	支援	支援	支援
快速報告	支援	支援	支援
APIs（Perl、Java™）	支援	支援	支援
動態伺服器管理	不支援	支援	支援

程序自動化

程序自動化可讓企業在不同的應用程式、人員和系統之間編排程序，以提升效率、最佳化成本、確保符合標準，並提高生產力。

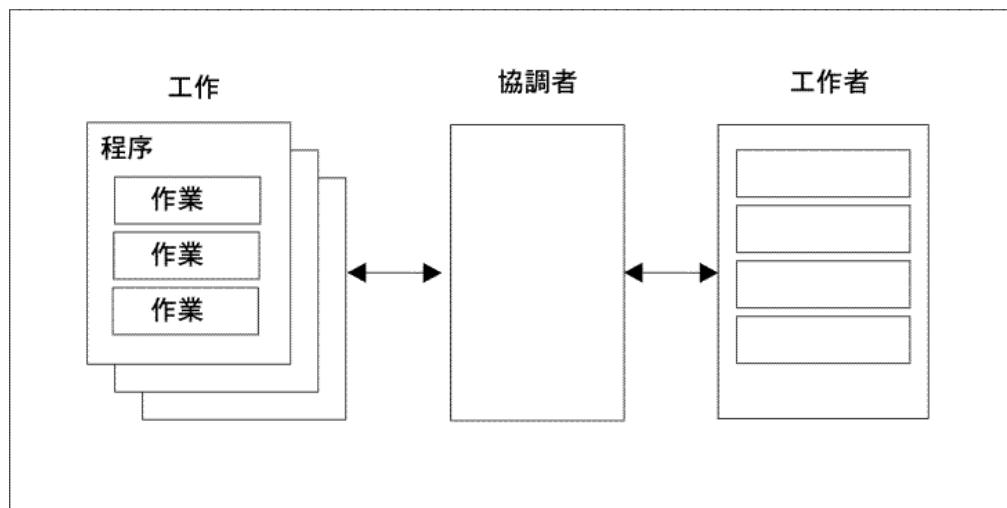
Build Forge 可自動化、編排、管理及追蹤產品開發組合線內的程序。它常被用來將軟體建置和套裝程序自動化。

本節說明程序的人工模型，並對映至 Build Forge 特性。

人工模型

這些角色組成程序自動化的人工型模型。

- **工作：**要執行的可用工作是一組記載下來的程序。每一項程序都是由作業 組成。
- **協調者：**協調者 會執行工作。協調者會選取要執行的程序，然後依序檢查作業。針對每一個作業，程序會定義應該由哪個工作者來執行它。協調者會將作業交給工作者，然後等待結果。
- **工作者：**工作者 會執程序中所定義的作業。



Build Forge 模型

人工型的模型會對映至 Build Forge 模型，如下所示：

- **工作：**工作程序是在 Build Forge 專案物件中定義的。每一個專案都包含一個以上的步驟物件。專案中的步驟對應於人工模型中程序內的作業。
- **協調者：**協調者是執行中的 Build Forge 軟體。它包含一個稱為程序引擎的元件。當您在 Build Forge 中啟動專案時，程序引擎會將它當作工作物件來執行。工作只是執行中的專案。
- **工作者：**工作者是主機。它們在 Build Forge 中是以伺服器物件來表示。主機中必須安裝代理程式。

這是非常簡化的模型。

- Build Forge 包含許多其他物件類型來支援程序自動化。使用者、授權和支援伺服器 and 專案的其他物件，全都儲存在資料庫中。部分 Build Forge 也會儲存在資料庫中，例如：使用者介面小組件。資料庫必須正在執行中，Build Forge 才能執行。
- 您可以配置專案和步驟以許多不同的方式執行，來支援含有複雜相依項目和工作流程的複雜程序。專案可以執行其他專案，而步驟本身則可以執行專案。
- 步驟和專案可以根據準則資料（屬於伺服器定義），動態地選取伺服器。例如，步驟可以根據它執行的是 Windows 還是 UNIX/Linux，來選擇要在哪一部伺服器上執行。伺服器也可以配置成聯合排存的資源，因此步驟可以根據伺服器的現行可用性 or 負載來選擇伺服器。

要安裝的 Build Forge 元件

若要使用 Build Forge，必須安裝以下元件：

- 資料庫：Build Forge 使用資料庫來儲存它所使用的所有物件。
- Build Forge：這是一組技術，可控管資料庫的存取權、執行使用者介面程式碼，以及執行正在執行工作的工作。
- 代理程式：您要在即將執行工作的每一部主機上安裝代理程式軟體。
- Web 瀏覽器：使用者通常會透過 Web 瀏覽器中呈現的主控制台來存取 Build Forge。

Build Forge® 概念和物件

Build Forge 使用一組簡易的物件來建構程序自動化的說明。

本節提供您主要概念和物件的概觀，以及它們之間的相關性如何。其中含有詳細說明的鏈結，以方便您使用。

使用者和角色

若要存取 Build Forge，使用者需要設定其使用者物件。在正式作業系統中，這是透過配置 Build Forge 來存取身分管理系統 LDAP 來完成。

Build Forge 會使用授權系統來控制所有物件的使用者存取權。存取群組物件包含許可權清單。使用者屬於一個以上存取群組。

您可以建立自己的存取群組，或修改所提供的存取群組。系統中定義的存取群組如下所示：

- Build Engineer - 定義程序（建立專案和步驟）
- System Manager - 管理伺服器和其他系統方面的特性
- Security - 管理使用者、存取權和安全
- Developer - 協助開發程序、執行工作、檢視結果
- Operator - 複製專案、執行工作、檢視結果
- Guest - 複製專案、執行工作、檢視結果

伺服器

伺服器 物件會定義可用來執行專案和步驟的位置。專案和步驟可以使用相同的伺服器，或各自選取一部伺服器。

伺服器物件代表用來執行工作的主機。主機必須正在執行 **Build Forge** 代理程式，才能從系統接收工作並傳回結果。

其他物件與伺服器相關，而且需要事先設定好，才能定義專案和步驟：

- **伺服器鑑別：** *伺服器鑑別* 會儲存登入資訊，供伺服器用來存取伺服器所指定的主機。您必須先建立伺服器鑑別，才能建立使用它的伺服器。在伺服器定義中，可以從系統中所定義的所有伺服器鑑別清單中選擇伺服器鑑別。
- **收集器：** *收集器* 物件會收集指定的伺服器內容。資料會儲存在 *資訊清單* 中。伺服器會指派一組預設內容。這些內建內容包含主機架構、網路連線和資源（CPU、記憶體、負載）的相關資訊。您可以定義收集器物件來新增其他內容。您必須先建立收集器，才能將它加入伺服器。在伺服器定義中，可以從系統中所定義的所有收集器清單中選擇收集器。
- **選取元：** *選取元* 物件會定義如何選取伺服器供專案或步驟使用。您必須先建立選取元，才能將它加入使用它的專案。在專案定義中，可以從系統中所定義的所有選取元清單中選擇選取元。
- **環境：** *環境* 物件是一組可供步驟使用的變數。在工作期間，要先在伺服器主機上設定變數，才能執行步驟。環境可以與伺服器物件、專案物件和步驟物件相關聯。當相同的變數在不同的環境中設為不同的值時，繼承架構能判斷要使用哪一個值。您必須先建立環境，才能將它加入伺服器、專案或步驟。在這些物件定義中，可以從系統中所定義的所有環境清單中選擇環境。

環境

環境 是一組變數。您可以針對伺服器、專案和步驟物件來指定環境。執行某個步驟時，將會合併在每個物件中設定的環境，以提供變數給該步驟使用。請參閱第 263 頁的『關於環境』。

步驟執行時，可以變更變數。（請參閱第 271 頁的『在步驟執行期間變更變數值』。）變更的範圍可以是步驟的區域變數、專案的區域變數，或是永久變數（此變數是在儲存的環境中變更）。

您可以使用預先定義的系統變數以及您定義的變數。

專案

專案 會定義要在程序中執行的工作。當專案啟動時，會將它當成工作來執行。

要執行的工作包含在步驟清單中。

與專案相關的其他物件：

- **選取元：**選取元 能決定啟動專案的位置。如果沒有為專案指定選取元，該專案便無法獨立執行，並且會被稱為程式庫。您必須已定義選取元，才能將它指派給工作。
- **環境：**環境 物件是一組可供步驟使用的變數。在工作期間，要先在伺服器主機上設定變數，才能執行步驟。環境可以與伺服器物件、專案物件和步驟物件相關聯。如果將相同的變數設為不同的值，那麼繼承架構能判斷要使用哪一個值。您必須先建立環境，才能建立使用它的伺服器、專案或步驟。在這些物件定義中，可以從系統中所定義的所有環境清單中選擇環境。
- **通知範本：**通知範本 會定義如何送出工作活動（工作啟動、工作通過、工作失敗、其他）的相關通知。通知物件會定義誰要通知整個存取群組。您可以在系統設定中指定 SMTP 伺服器的位置。
- **類別：**基於維護的用途，可使用類別 物件將專案組成群組。一般而言，會使用類別來定期清除或保存已完成的工作。類別會影響使用專案來執行的所有工作。
- **配接器：**配接器 會定義與外部系統（通常是原始碼管理系統）的整合。系統提供了數個範例範本作為起點。您需要進一步配置它們來管理連線，並在系統中執行指定的動作。
- **配接器鏈：**配接器鏈結 會在配接器與使用它的專案之間定義關係。

步驟

步驟 會定義要執行的最小工作單位。（請參閱第 323 頁的『關於步驟』。）其主要元件是其「指令」內容，此內容包含要在選取的伺服器上執行的指令。

「指令」內容也可以用來執行點指令。點指令是指在程序引擎上執行，並提供附加功能的指令。

與專案相關的其他物件：

- **選取元：**步驟可以有它自己的選取元。如果未指定，則會使用其專案的選取元。
- **環境：**步驟可以有它自己的環境。由伺服器、專案和步驟所提供的環境會合併在一起。依預設，會依此順序來套用它們，因此由步驟的環境所定義的任何變數，其優先順序會高於相同變數的定義。您可以在系統設定中控制優先順序。
- **日誌過濾器：**您可以在步驟的「結果」內容中指派日誌過濾器給此步驟。日誌過濾器 物件可以用來指定條件，指出步驟是通過還是失敗。執行 Perl 引擎時，您可以使用正規表示式來掃描日誌中特定的型樣。執行 Java 引擎時，您還可以使用正規表示式來掃描日誌中特定的型樣。通常會使用指令的結束狀態，但日誌過濾器提供了另一個方法。您必須先建立日誌過濾器，才能在步驟中指定它。在步驟的「結果」內容中，您可以從系統中所定義的所有日誌過濾器清單中選擇日誌過濾器。
- **通知範本：**通知範本 會定義如何送出步驟活動（步驟啟動、步驟通過、步驟失敗、其他）的相關通知。通知物件會定義誰要通知整個存取群組。您可以在系統設定中指定 SMTP 伺服器的位置。

- **Build Catalyst**：步驟可以執行 `rafmake`，這是 **Build Catalyst** 中的主要公用程式。**Build Catalyst** 提供了加速 `make` 型建置的方法。除了在執行 `make` 建置的主機上安裝 **Build Forge** 代理程式之外，還必須安裝 **Build Catalyst**。

工作

工作是執行中的專案。當工作啟動時，程序引擎會將它放到佇列中，然後再執行它。您可以在「工作」畫面中檢查其狀態。當它完成時，會顯示下列資訊：

- **結果**：您可以開啓工作來檢閱所有步驟的結果。您也可以開啓執行中的工作來監視其進度。
- **步驟日誌**：步驟日誌會記錄如何執行步驟的完整資訊，包含資訊清單和環境設定以及執行結果的相關資訊。請按一下「結果」頁面中的步驟結果鏈結來檢視日誌。
- **資料清單**：資料清單 (BOM) 包含工作步驟和步驟資訊清單的相關資訊。您可以使用 `.bom` 點指令來格式化 BOM 的其他資訊，並將資料寫入此資料清單。您可以使用 `.scan` 點指令來新增基準線和檢查點到 BOM。

您可以取消及重新啟動工作。您可以新增專案至排程，使它們在排定的時間執行。

第 2 章 Build Forge® 入門

這些主題提供系統的快速簡介。

安裝

一開始，請先安裝系統（如果尚未安裝的話）。您可以將系統（管理主控台和代理程式）安裝在一部主機上，來簡化一開始的作業。

1. 檢查需求
 - 第 21 頁的『資料庫需求』
2. 完成下列步驟：
 - a. 第 29 頁的第 5 章，『安裝規劃』（獨立式、一般正式作業、分散式等等）
 - b. 第 33 頁的第 6 章，『前置安裝設定』（國際化資料、資料庫和安全）
 - c. 第 55 頁的第 7 章，『安裝管理主控台』
 - d. 第 131 頁的第 10 章，『安裝代理程式』

存取和使用主控台

這個主題提供使用系統功能表及標籤的基本資訊。

存取主控台

如果要存取主控台，請執行下列動作：

1. 開啓一個瀏覽器視窗。
2. 輸入主控台的 URL。

`http://host:port/`

Host 是執行主控台之主機的完整網路名稱或 IP 位址。

Port 只有在未使用 HTTP 預設埠時才需提供。除非啓用 SSL，否則預設埠為 80。預設 HTTPS 埠為 443。只有在主控台配置成使用 SSL 時，才會使用 HTTPS。

範例：

- `http://localhost/` 或 `http://127.0.0.1/`：如果您是在安裝主控台的主機上執行瀏覽器，則可以使用此項。
 - `http://my.company.com/`
 - `http://my.company.com:81/`：必須指定埠號，因為主控台是安裝成使用 81 埠。
3. 登入。提供下列資訊，然後按一下**登入**：
 - 使用者名稱
 - 密碼
 - 網域

依預設，會將您重新導向另一個 URL 來輸入您的認證，順利登入之後，再重新導回。URL 是針對服務元件。它提供一個安全 Servlet 來進行鑑別。它的使用與否是

可配置的，如果將它關閉，或利用 LDAP 來進行使用者鑑別，就不使用它。如需相關資訊，請參閱第 101 頁的『關於預設登入安全』。

如果您是以 root 身分或僅定義在 Build Forge 中的另一個使用者身分登入，就不會提示您提供「網域」。最理想的做法是將 Build Forge 配置成使用 LDAP 來鑑別使用者。若採取那樣的設定，就會呈現「網域」欄位。它會參照用來鑑別使用者的 LDAP 網域。

另請參閱 第 206 頁的『LDAP 整合』。

使用者階段作業

使用者階段作業是利用最初在 Cookie 中，所產生的階段作業 ID 來進行追蹤。產生之後，階段作業 ID 便儲存在資料庫中。每項使用者互動都會對照這個階段作業 ID 來檢查。

一般而言，在同一次登入中，您可以開啓多個瀏覽器視窗，但它們必須在相同主機上，且使用同類型的瀏覽器（Firefox、Internet Explorer）。

在任何下列狀況之下，階段作業都會關閉：

- 利用相同的使用者帳戶從另一部主機登入。
- 利用相同的使用者帳戶從相同主機登入，但使用不同類型的瀏覽器。
- 使用者關閉含有「管理主控台」階段作業的瀏覽器視窗或標籤。
- 已達到階段作業時間限制。時間限制設在「自動登出」系統設定中。依預設，它會設為 0，表示沒有時間限制。

限制和提示

請注意以下的使用限制與提示：

- 請勿將顯示器設成 1024 x 768 以下。為求最佳結果，請使用 1280 x 1024 或以上。
- 請勿將瀏覽器視窗縮小至 1024 x 768 以下。
- 使用瀏覽器設定，來控制字型大小、顏色和其他協助工具特性。您可能需要重新整理頁面，以適當顯示新的瀏覽器設定。
- 如果字型太大或您將瀏覽器縮小，則「管理主控台 (MC)」將不會在 Microsoft Internet Explorer 的**工作 > 開始**頁面顯示。

使用主控台視窗




按一下右上方的標籤，以檢視應用程式。您可以按一下**儀表板**（預設值）、**使用者介面配置**、**主控台**或**報告**。系統會記住您登出時所處的標籤，並在您下次登入時呈現該標籤。

按如下所示，在**主控台**應用程式中導覽：

- 左側顯示功能表，右側顯示主要檢視畫面。
- 按一下其中一個功能表項目，以查看畫面或開啓畫面的子功能表。
- 如果畫面大於檢視區域，請使用右上方的換頁控制項：



- 在檢視清單時使用「過濾」欄位。當您輸入字串並按一下**過濾**時，會更新清單。其中只會顯示名稱中含有該字串的項目。
- 拖曳功能表右緣以調整大小。
- 可讓您建立或編輯資料的畫面通常含有下列的控制項：
 - 畫面上半部可讓您檢視和選取項目。請按一下其中一個項目名稱，以查看其內容。按一下**編輯**圖示 ，以編輯項目。
 - 畫面下半部顯示所選項目的內容。
- 在某些情況下，當您選取清單中的項目時，左側功能表下方會顯示其他資訊。範例：
 1. 開啓**管理 > 使用者**
 2. 按一下某個使用者。

該使用者的許可權會出現在左側功能表下方。

過濾和排序清單

您可以將清單快速過濾或排序。

如果要過濾清單，請執行下列動作：

1. 在「過濾」文字框中輸入文字。項目會區分大小寫。
2. 按一下**過濾**。
3. 檢視結果。您可以選擇檢視所有結果，或將它們標頁數。

提示：

- 過濾器會處理所有項目是鏈結（藍色及畫底線）的直欄。您無法過濾以黑色純文字顯示值的直欄。
- 如果只要過濾單一直欄，請在過濾字串之前輸入直欄名稱。比方說，如果您有一個「選取元」直欄，您可以輸入選取元：全選。
- 系統會保留您輸入的過濾字串。請按一下**過濾器**方框旁的箭頭，顯示您或其他人所輸入的過濾器清單。若要刪除過濾器串，請強調顯示某個字串，然後按一下字串右邊的**垃圾桶**圖示。
- 在過濾字串清單中，一律會出現**顯示全部**選項。
- 如果您輸入過濾文字，但沒有符合過濾器的項目，就不會顯示任何項目。

如果要排序清單，請執行下列動作：

1. 按一下直欄標籤旁的雙箭頭。
2. 檢視結果。您可以選擇檢視所有結果，或將它們標頁數。

提示：

- 再按一下雙箭頭，會反轉排序順序。

建立 Hello World 專案

這個主題說明如何建立與執行簡式專案，以驗證建置系統的設定是否適當。

下列項目提供建立 "Hello World" 專案所需的作業概觀。

1. 設定伺服器：
 - a. 建立伺服器鑑別（登入名稱和密碼），使 Rational Build Forge 可以在伺服器上存取及執行指令
 - b. 建立伺服器定義，以指出已安裝代理程式的伺服器
 - c. 建立 HelloWorld 專案的選取元，用來決定執行的伺服器
 - d. 測試伺服器連線，以確定伺服器上的「主要主控台」和代理程式可以彼此通訊
2. 定義專案：
 - a. 命名 HelloWorld 專案，並使用選取元來指定用來執行專案的伺服器
 - b. 建立步驟來執行顯示 Hello World 訊息的指令
3. 執行工作：
 - a. 啟動專案的實例，又稱為工作
 - b. 按一下左畫面中的**工作**項目，以檢查工作狀態
4. 檢視工作結果：

- a. 在完成的工作清單中，按一下工作標籤來顯示工作步驟的清單（如果是 HelloWorld 範例，則是單一步驟）
- b. 按一下工作的步驟以顯示步驟日誌
- c. 在步驟日誌中尋找指令的 Hello World 輸出

設定伺服器

這個主題說明如何設定在“Hello World”習作中使用的伺服器，以驗證是否已適當地設定建置系統。

建立伺服器鑑別

Rational Build Forge 會使用伺服器鑑別（登入名稱和密碼）來連接已安裝代理程式的伺服器。

關於這項作業

您必須先建立伺服器的伺服器鑑別，才能將伺服器定義為供 Rational Build Forge 使用的資源。與登入名稱和密碼相關聯的專用權，能決定 Rational Build Forge 專案在伺服器上執行時所使用的專用權。

程序

1. 在 Rational Build Forge 的左畫面中按一下**伺服器 > 伺服器鑑別**。

「管理主控台」即會在底端顯示空白的「伺服器鑑別詳細資料」畫面。

2. 按一下**新增伺服器鑑別**。
3. 在**名稱**中，輸入鑑別的名稱。請使用登入名稱作為鑑別名稱。
4. 在**登入**中，指定伺服器中的帳戶的登入名稱。

註：如果登入名稱來自網域使用者，請在這個欄位中併入網域。例如輸入：MYDOMAIN/joeuser。

5. 在**密碼**中，輸入密碼。
6. 在**驗證**中，重新輸入密碼。
7. 按一下**儲存伺服器鑑別**。

結果

新的伺服器鑑別即會顯示在上方畫面，而且可以在您定義伺服器時使用。

建立伺服器

「管理主控台」中的伺服器代表可在其中執行專案或步驟的主機。

程序

1. 在 Rational Build Forge 的左畫面中，按一下**伺服器**。
2. 按一下**新增伺服器**。
3. 提供伺服器詳細資料。

- a. **名稱**：提供名稱 HelloWorldServer 給伺服器。您之後在定義選取元時，將會使用這個相同的名稱。
- b. **路徑**：指定伺服器在建立專案和工作目錄時要使用的目錄，例如 C:\buildforgeprojects。系統在建立建置目錄時，即使用這個路徑值來作為起始點。

提示：系統並未建立伺服器路徑。在建置試圖存取伺服器之前，路徑就必須存在。如果路徑不存在，建置就會失敗。

- c. **主機**：提供正在執行代理程式之實體電腦的主機名稱。如果您將「管理主控台」電腦定義成伺服器，請使用 localhost 值。（代理程式也必須安裝在「管理主控台」上。）

註：主機名稱之前請勿加上通訊協定。例如，請勿使用 http://。

- d. **鑑別**：請選取要搭配這部伺服器使用的伺服器鑑別。
- e. 讓其他欄位維持使用其預設值。

4. 按一下**儲存**。新的伺服器即會顯示在內容畫面頂端的伺服器清單中。
5. 若要驗證伺服器的配置是否正確，請在清單中選取伺服器，然後按一下**測試連線**。

系統如果無法與伺服器通訊，則會報告錯誤。如果您收到錯誤，請確定代理程式正在伺服器上執行。

建立選取元

選取元會決定用來執行專案或步驟的伺服器。

關於這項作業

請建立選取元，使 HelloWorld 專案可以決定要在何處完成其步驟。

程序

1. 在 Rational Build Forge 的左畫面中按一下**伺服器 > 選取元**。
2. 按一下**新增選取元**。
3. 在**名稱**中，輸入 HelloWorldSelector。
4. 按一下**儲存**。畫面的底部會變更，使您可以定義選取元內容來決定要選取哪些伺服器。
5. 使用在第 11 頁的『建立伺服器』中指定的伺服器名稱來設定用來執行的伺服器。將**名稱**設為 BF_NAME，並將「運算子」設為 **EQ**（預設值），然後在**值**中輸入 HelloWorldServer。
6. 按一下**儲存**。

定義專案

這個主題說明在 "Hello World" 習作中，如何建立簡式專案來驗證是否已適當地設定建置系統。

建立專案

開始之前

您必須設定要在您的 HelloWorld 專案定義中使用的伺服器和選取元。

程序

1. 選取**專案**。 **專案詳細資料**畫面會顯示在主內容畫面的底端。
2. 在**名稱**欄位中，輸入 HelloWorld。
3. 將**選取元**設為 HelloWorldSelector。
4. 按一下**儲存**。系統會顯示空的專案步驟清單，以及空白的**步驟詳細資料**頁面。依『建立步驟』中的說明來定義步驟。

建立步驟

開始之前

若要定義 HelloWorld 專案中的第一個步驟，您必須已開始定義專案。

程序

1. 在**步驟詳細資料**頁面的**名稱**欄位中，輸入 EchoHelloWorld。
2. 在**指令**欄位中輸入指令，以便將 Hello World 寫入您選擇的伺服器上的標準輸出。例如，假設這個指令是在 Windows®、Solaris、Linux®、UNIX® 和 Apple Macintosh OS X 作業系統上運作：

```
echo Hello World
```
3. 按一下**儲存步驟**。該步驟即會顯示在步驟清單中。

執行工作


這個主題說明在 "Hello World" 習作中，如何啟動工作（執行中專案的實例）並檢查其狀態，來驗證是否已適當地設定建置系統。

啟動工作

開始之前

您必須定義 HelloWorld 專案。

程序

1. 選取**專案**，以重新顯示專案清單。
2. 按一下 HelloWorld 專案旁的**快速啟動**圖示。

检查工作狀態

程序

1. 按一下**工作**，然後按一下**執行中**標籤，來查看 HelloWorld 工作已列為執行中。工作完成之後，會將它移至**已完成**標籤的頁面。

註：如果未列出 HelloWorld 工作，請繼續執行下一步。

2. 按一下**已完成**標籤。 HelloWorld 工作即會顯示在已完成的工作清單中。

檢視工作結果

這個主題說明在 "Hello World" 習作中，如何檢視工作結果來驗證是否已適當地設定建置系統。

開始之前

您必須定義並執行 HelloWorld 專案。

檢視已完成的工作清單

程序

1. 按一下**工作**。
2. 按一下**已完成**標籤。 已完成的工作清單會顯示 HelloWorld 工作已順利完成。

檢視步驟日誌

程序

1. 按一下**工作**。
2. 按一下**已完成**標籤。
3. 按一下 HelloWorld 工作的工作標籤。 起始工作的預設工作標籤為 BUILD_1。 系統會顯示工作步驟的名稱。在這個範例中，工作只有一個步驟：EchoHelloWorld。
4. 按一下 EchoHelloWorld 步驟來檢查其日誌。 在大部分 Hello World 範例中，您會在主控台視窗或蹦現視窗中看到 "Hello World" 文字。「管理主控台」會將指令傳給目標伺服器上的代理程式處理程序來執行其工作；接著代理程式會將這些指令的輸出傳回「管理主控台」，再由它將輸出儲存在步驟日誌中。日誌有很多區段；相關的區段為最後的 EXEC 區段。如果您只要顯示 EXEC 區段，您可以清除所有勾選框，再選取 EXEC 勾選框，然後按一下「重新整理」鏈結即可。指令的結果顯示如下所示。

```
80 04/19/10 11:06AM EXEC Locale set to 'English_United States.1252'
253 04/19/10 11:06AM EXEC Locale set to 'English_United States.1252'
354 04/19/10 11:06AM EXEC Performing variable expansion on command line
356 04/19/10 11:06AM EXEC start [C:\buildforgeprojects\HelloWorld\BUILD_1@mcsystem]
357 04/19/10 11:06AM EXEC Hello World
358 04/19/10 11:06AM EXEC end [C:\buildforgeprojects\HelloWorld\BUILD_1@mcsystem]
```

結果

這個專案示範您已正確地配置系統，而且專案可以順利存取伺服器、執行，以及在伺服器上產生輸出。您可以將 echo 指令換成任何可在目標伺服器上執行的指令。

專案範例

所包含的專案範例可協助您熟悉系統。

專案範例包含在下列目錄中：

<bfinstall>/samples/projects/

依預設，Build Forge 安裝目錄或 *<bfinstall>* 是 C:\Program Files\IBM\Build Forge（若是在 Windows 上）和 /opt/buildforge（若是在 UNIX 和 Linux）。

如果要使用專案範例，請執行下列動作：

- 利用 `bfimport` 指令將它匯入到「管理主控台」中。
- 執行專案。

第 3 章 IBM Rational Build Forge 文件注意事項

© Copyright IBM Corporation 2003, 2013.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

本資訊是針對 IBM 在美國所提供的產品與服務所開發，

在其他國家或地區中，IBM 不見得有提供本文所提及的各項產品、服務或特性。請洽詢當地的 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。本文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以用書面方式查詢有關授權的問題，來信請寄：

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

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國家/地區的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

下列段落若與該國之法律條款抵觸，即視為不適用：IBM 僅以「現狀」提供本書，而不提供任何明示或默示之保證（包含但不限於可售性或符合特定效用的保證）。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供保證。這些網站所提供的資料不是本 IBM 產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

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

如果本程式的已授權人爲了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之「IBM 客戶合約」、「IBM 國際程式授權合約」（或任何同等合約）條款，提供本文件中所述的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料是在控制環境中得出。因此，在其他作業環境中獲得的結果可能有明顯的差異。在開發層次的系統上可能有做過一些測量，但不保證這些測量在目前一般系統上有相同的結果。再者，有些測定可能是透過推測方式來評估。實際結果可能不同。本文件的使用者應驗證適用於其特定環境的資料。

本書所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

所有關於 IBM 未來方向或計劃的聲明僅代表目標與方針，可能隨時更改或撤消，恕不另行通知。

這份資訊含有日常商業運作所使用的資料和報告範例。爲了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這裡的所有名稱全屬虛構，如與任何實際企業所用的名稱及地址雷同，純屬巧合。

著作權授權

這份資訊含有原始語言的範例應用程式，用以說明各作業平台中的程式設計技術。貴客戶可以爲了研發、使用、銷售或散布符合範例應用程式所適用的作業平台之應用程式介面的應用程式，以任何形式複製、修改及散布這些範例程式，不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、可用性或功能。範例程式只「依現狀」提供，不提供任何形式之保證。IBM 對因使用這些程式範例而產生的任何損害概不負責。

這些範例程式的任何部分或任何衍生著作的每一份拷貝，都必須具有下列著作權聲明：

© (貴公司名稱) (年)。部份程式碼來源於 IBM Corp. 範例程式。© Copyright IBM Corp. 2003, 2013.

若貴客戶正在閱讀本項資訊之電子檔，該檔案可能不會有照片及彩色說明。

隱私權條款考量

IBM 軟體產品，包括軟體即服務解決方案（「軟體供應項目」），可能使用 Cookie 或其他技術來收集產品使用資訊，以協助改善一般使用者體驗、調整與一般使用者的互動，或作爲其他用途。在許多情況下，「軟體供應項目」不會收集個人識別資訊。我

們的部分「軟體供應項目」可以協助您收集個人識別資訊。如果這個「軟體供應項目」使用 Cookie 來收集個人識別資訊，下面列舉這個供應項目使用 Cookie 時的特定資訊。

根據部署的配置而定，這個「軟體供應項目」可能使用階段作業 Cookie，以收集每一位使用者的姓名、使用者名稱、密碼、設定檔名稱或職務，作為階段作業管理或鑑別用途。這些 Cookie 無法停用。

如果對這個「軟體供應項目」部署的配置可讓身為客戶的您，透過 Cookie 及其他技術來收集一般使用者的個人識別資訊，則您應該尋找有關這類資料收集的任何適用法律的法律意見，包括公告和同意的任何需求。

如需在這些用途上使用各種技術（包括 Cookie）的相關資訊，請參閱 IBM 的隱私權條款 (<http://www.ibm.com/privacy>) 及 IBM 的線上隱私權聲明 (<http://www.ibm.com/privacy/details/us/en>) 的「Cookie、Web Beacon 與其他技術」和「軟體產品與軟體即服務」章節。

商標確認通知

IBM、IBM 標誌及 ibm.com 是 International Business Machines Corp. 的商標或註冊商標，已在全球許多國家或司法管轄區註冊。其他產品及服務名稱可能是 IBM 或其他公司的商標。最新的 IBM 商標清單可從網站上取得，網址為 www.ibm.com/legal/copytrade.shtml 。

第 4 章 需求

Rational® Build Forge® 產品元件具有軟硬體需求。

下列區域中列有需求：

- 硬體、軟體與資料庫
- Installation Manager
- Licensing
- IPv6 支援的網路需求
- 整合
- 國家語言支援

硬體、軟體與資料庫需求

硬體、軟體與資料庫需求可在特定產品的詳細系統需求網站上，從查詢的報告中取得。您必須提供您所使用的產品、產品版次、版本以及作業系統。其中會針對硬體、作業系統、資料庫、代理主機和 Web 用戶端來說明需求。

資料庫需求

您必須安裝資料庫才能使用 Rational® Build Forge®。在「資料交換中心」檢查所支援資料庫的軟硬體需求。

產品內含資料庫授權

DB2® Workgroup 10.1 版資料庫可透過 Passport Advantage® 另行下載取得。您的產品授權包含此產品。如果有意搭配使用 DB2 Workgroup 10.1 與本產品，您必須在安裝主控台之前，從 Passport Advantage 下載並安裝它。使用 Rational Build Forge 沒有平台或主機位置限制。

Installation Manager 需求

如果已安裝 Installation Manager，它必須符合安裝 Build Forge 的版本需求。

安裝程式會嘗試尋找已安裝的 Installation Manager。它必須符合版本需求。

如果找不到，安裝程式便會安裝 Installation Manager。

最低支援版本是 1.4.4。

產品版本的授權需求

您使用的授權機制會視您擁有的產品版本而定。請參閱 第 1 頁的『Build Forge 產品版本』。

版本	授權機制
Rational Build Forge Standard Edition	授權伺服器

版本	授權機制
Rational Build Forge Enterprise Edition	授權伺服器
Rational Build Forge Enterprise Plus Edition	授權檔

這個主題包含設定 Build Forge 之 Rational License Server 的詳細資料。您可以透過 Build Forge 使用者介面來指定新的授權伺服器。不支援透過 IBM Installation Manager 來管理授權。

在安裝期間指定授權檔

這個主題僅適用於使用 Build Forge Enterprise Plus Edition（需要檔案型授權）的客戶。

若要在安裝期間指定授權檔，請完成下列步驟：

1. 從 Passport Advantage 取得授權金鑰檔 **irbf_license.properties**。將其置於安裝根目錄中。

Windows	C:\Program Files\IBM\Build Forge
UNIX/Linux	/opt/buildforge

2. 在 Installation Manager 安裝程式中，於「授權伺服器配置」頁面上按一下**授權檔**。
3. 按一下**瀏覽**，於安裝根目錄中找出授權碼檔案。
4. 按兩下 **irbf_license.properties** 來選取，然後按**下一步**來繼續安裝。

配置 Build Forge 的 Rational License Server

您的授權管理者會設定 Rational License Server，並為您提供在安裝期間指定的授權伺服器主機名稱。

本節僅適用於需要伺服器型授權的產品版本。

- Build Forge Standard Edition
- Build Forge Enterprise Edition

授權伺服器的 Build Forge 配置需求

在配置 Build Forge 的授權伺服器之前，請先檢閱下列需求。

- 您環境中的所有「管理主控台」主機都必須能夠連接至 Rational License Server 主機。
- UNIX/Linux 或 Windows Rational License Server 必須設為自動啟動並當成服務來執行。
- 若要安裝及配置產品以使用授權伺服器，授權管理者必須為您提供 Rational License Server 的主機名稱和埠號。

註：在 Rational® Build Forge® 8.0 版中，支援的 Rational License Key Server 版本已從 7.0 變更為 8.1.3。如果您要自 8.0 以前的版本升級，則還必須升級授權伺服器。如需相關資訊，請查閱詳細的系統需求報告。

在安裝期間指定授權伺服器

在安裝期間，您可提供 Rational License Server 的主機名稱和 TCP/IP 埠。請從授權管理者取得這項資訊。

如果要配置伺服器型授權，請執行下列動作：

1. 在 Installation Manager 安裝程式中，於「授權伺服器配置」頁面上按一下**授權伺服器**。
2. 接受預設使用者 *root*，或指定不同的 UNIX 或 Linux 使用者來執行 Build Forge。如果您指定不同使用者，則必須以 *root* 身分啟動 Build Forge。Build Forge 會以在安裝步驟期間指定的使用者身分啟動引擎及 Apache Tomcat。對於在 Installation Manager 的「資料庫配置」頁面中所指定的資料庫程式庫及 JDBC Jar 檔，您必須具有讀取及執行許可權。
3. 在**授權伺服器**欄位中，輸入 Rational License Server 的有效主機名稱。

註：如果您提供主機名稱，則這項資訊會自動加入 Build Forge 系統設定中。如果您不知道主機名稱，請在這個欄位中輸入任何字元或值，然後在安裝完成之後，於使用者介面中更新 Build Forge 系統設定。

4. 在**埠**欄位中，輸入授權伺服器的 TCP/IP 埠。預設埠是 27000。
5. 按**下一步**繼續安裝。

變更「管理主控台」的授權伺服器

若要變更「管理主控台」的 Rational License Server，請對產品授權伺服器配置進行下列修改。如果有下列情形，則必須執行此程序：

- 您在安裝期間輸入了不正確的主機名稱或其他值。
 - 授權伺服器主機名稱在安裝期間會變成灰色，表示 FLEXlm 用戶端已登錄主機의授權伺服器。
1. 此步驟僅適用 Build Forge 7.1 至 7.1.3.3 版。若為較新的版本，請繼續進行步驟 2。針對您的作業系統，請變更 *RATIONAL_LICENSE_FILE* 變數的值。

Windows	RATIONAL_LICENSE_FILE 位於登錄中的 HKEY_LOCAL_MACHINE\SOFTWARE\FLEXlm License Manager
UNIX/Linux	RATIONAL_LICENSE_FILE 是設定在 .flexlmrc 檔中，該檔案位於執行 Build Forge 之使用者的起始目錄中

將變數設為授權伺服器的正確主機名稱：

`port@hostname | @hostname`

如果授權伺服器埠是 27000（預設值），則埠不是必要的。

2. 在「管理主控台」使用者介面中，選取**管理** → **系統**。
3. 找出「授權伺服器」設定，並將其值設為新 Rational License Server 的主機名稱。

使用下列其中一種格式：

`<host_name>:<port> | <host_name> | <port>:<host_name>`

如果授權伺服器埠是 27000（預設值），則埠不是必要的。

4. 按一下 Web 瀏覽器上的**重新整理**，以驗證「管理主控台」已連接至新的授權伺服器。

取得授權碼並設定 Rational License Server

如果現有的授權伺服器無法使用，下表指出授權管理者會完成以取得授權碼並設定 Rational License Server 的一般作業。

若要安裝及配置 Rational License Server，請檢閱適用於您 Rational License Server 軟體版本的說明文件。請造訪 <http://www-306.ibm.com/>，選取**支援與下載**，然後搜尋 Rational License Management Guide。

授權管理者作業	資源
從 Rational License Key Center 取得授權碼。	Rational License Management Guide 位於 http://www-306.ibm.com/software/rational/support/licensing 的 Rational License Key Center 快速入門
驗證「管理主控台」主機和 Rational License Server 的網路連線功能。	Rational License Management Guide
透過以下一種方式，取得 Rational License Server 軟體的必要版本： <ul style="list-style-type: none">存取產品發行套件隨附的軟體。從「IBM Rational 下載與授權中心」下載軟體，網址如下所示： http://www-01.ibm.com/software/rational/support/updates/full-product.html	Rational License Management Guide
安裝 Rational License Server。	Rational License Management Guide
依下列方式，將授權碼安裝或匯入至 Rational License Server： <ul style="list-style-type: none">在 Windows 上，使用隨許多 IBM Rational 產品和 Rational License Server 軟體安裝的 IBM Rational License Key Administrator (LKAD)。在 UNIX/Linux 上，使用來自「IBM Rational 下載與授權中心」的 license_setup Script 和授權執行檔，網址如下： http://www-01.ibm.com/software/rational/support/updates/full-product.html	Rational License Management Guide

授權管理者作業	資源
<p>依下列方式，啟動 Rational License Server，並設定 Rational License Server 以自動啟動並當成服務來執行：</p> <ul style="list-style-type: none"> 在 Windows 上，Rational License Server 軟體會自動設為在啟動電腦時當成服務來啟動。如果未自動啟動，請參閱在 Windows 上自動啟動授權伺服器的指示。 在 UNIX/Linux 上，利用所提供的範本啟動 Script 來建立啟動 Script，並根據您的安裝加以修改。請從「IBM Rational 下載與授權中心」取得該範本，網址如下： <p>http://www-01.ibm.com/software/rational/support/updates/full-product.html</p>	Rational License Management Guide

IPv6 支援的網路需求

Rational® Build Forge® 系統可以在「網際網路通訊協定第 6 版 (IPv6)」和 IPv6-IPv4 混合的網路中執行，但附帶一些限制條件。

IPv6 支援需要正確地配置電腦和網路以支援 IPv6。網路配置問題會使得 Build Forge® 系統內指定的主機名稱和位址無法正確解析。

您必須手動配置 Rational® Build Forge® 以使用 IPv6。要執行這項作業，請參閱第 87 頁的『修改 httpd.conf』。

當您在 Build Forge® 中輸入 IP 位址時，請使用正確的位址格式。在 Build Forge® 中，管理者或使用者只會在少數位置輸入主機名稱或 IP 位址：

- 在安裝期間，管理者將指定「管理主控台」用來和授權伺服器（Standard Edition 和 Enterprise Edition）與資料庫通訊的主機名稱或 IP 位址。
- 在代理程式配置（bfagent.conf 檔）中，有一個選用的設定會限制連接代理程式的特定位址或位址範圍。
- 使用者在瀏覽器中輸入 URL，以檢視「管理主控台」使用者介面。這個 URL 是由執行「管理主控台」之伺服器的主機名稱或 IP 位址所組成。例如，若要存取安裝在名為 BFmachine 且已配置 IPv4 和 IPv6 位址之伺服器上的「管理主控台」，使用者可以在 Web 瀏覽器中輸入下列任一地址：
 - <http://BFmachine/>
 - <http://localhost/>
 - <http://127.0.0.1/>（IPv4 迴圈位址）
 - <http://::127.0.0.1/>、<http://0:0:0:0:0:0:127.0.0.1/>（IPv6 的 IPv4 迴圈位址縮寫），或簡單的 <http://::1/>（IPv6 的迴圈位址壓縮表示法）

IPv4 和 IPv6 位址在格式和長度上有別。

- IPv4 格式：**長度為 32 位元。位址是指定成以小數點區隔的四個十進位值，例如：255.255.255.255

- **IPv6 格式**：長度為 128 位元。位址指定成以冒號區隔的八個十六進位值，例如，FE80:0000:0000:0000:0202:B3FF:FE1E:8329。在高順位欄位的使用上，訂有一些使用慣例。此外，縮寫亦有規範。Build Forge® 不會對 IP 位址執行任何的解譯；它們會直接遞送至網路。因此，任何合法和有效的縮寫應該都能運作。請參閱其他參考資料，以取得 IPv6 位址使用慣例的相關資訊。

避免在 /etc/hosts 中使用 localhost 別名

在 UNIX 和 Linux 系統上，如果您同時使用 IPv4 與 IPv6，請不要在 /etc/hosts 中使用 localhost 作為別名。請使用伺服器的完整網域名稱或 IP 位址。

如果您必須在 /etc/hosts 中使用 localhost 作為主機名稱，該名稱僅應定義 IPv4 位址 127.0.0.1。

不支援 IPv6 的元件

不支援 IPv6 的元件必須安裝在使用 IPv4 位址的主機上。請將這些元件安裝在具有 IPv4 位址的電腦上：

- Rational License Key Server 8.1.3，這是產品發行套件內含的 Build Forge® Standard Edition 和 Enterprise Edition 的授權伺服器。

註：若要支援 IPv6 位址，您必須安裝 Rational License Key Server 8.1.3。請從「IBM Rational 支援中心」或「IBM Rational 下載與授權中心」取得該產品。

- DB2 資料庫：PHP 資料庫驅動程式尚未支援 IPv6。
- MySQL 資料庫：PHP 資料庫驅動程式尚未支援 IPv6。
- 11g 以外的 Oracle 資料庫：PHP 資料庫驅動程式尚未支援 IPv6。

國家語言支援需求

Build Forge 產品提供了法文、德文、義大利文、巴西葡萄牙文、西班牙文、日文、韓文、簡體中文和繁體中文等的本地化支援。

本節提供 Build Forge 中語言支援的相關資訊。

「管理主控台」和「代理程式」的語言設定

這個主題說明 Build Forge 如何配置「管理主控台」、「代理程式」和「引擎」元件的語言設定。

「管理主控台」語言設定

現行使用者的語言設定決定在「管理主控台」中顯示介面控制項時所要使用的語言。

請依下列方式設定某個使用者帳戶的「管理主控台」語言設定：

root 使用者

安裝程式會建立一個 root 使用者帳戶，供管理者第一次登入主控台時使用。

在初次登入時，root 使用者的語言設定是以 Build Forge 引擎主機的作業系統語言為基礎。

其他使用者的語言預設值

針對管理者建立的使用者帳戶，一開始會使用登入時所用的 Web 瀏覽器的語言喜好設定，來設定預設主控台語言。

註：這是 Web 瀏覽器的配置設定，不是 Build Forge 的設定。

如果管理者未變更使用者的語言喜好設定，其會繼承自 Web 瀏覽器。

如果管理者要變更您的語言，請繼續完成這項程序。在使用者介面中，按一下 **管理 > 使用者 > 語言**，然後選取語言。

LDAP 建立的使用者帳戶一律會使用配置給 Web 瀏覽器的語言喜好設定。

變更預設使用者語言

登入主控台之後，管理者可以從 Build Forge 視窗的左導覽畫面中按一下 **管理 > 使用者 > 語言**，為個別的 Build Forge 使用者選取不同的語言設定。

如果使用者的語言喜好設定是以這個方式來配置，不論 Web 瀏覽器的語言配置為何，「管理主控台」都會以選給使用者的語言來顯示介面控制項。

代理程式和 Build Forge 引擎語言設定

Build Forge 引擎主機的作業系統語言會指定：

- Build Forge 引擎使用的語言。
- Build Forge 代理程式的預設語言。

代理程式語言設定會控制系統訊息和工作輸出的語言。

不論是針對「管理主控台」、Web 瀏覽器設定語言，還是作為 Build Forge 喜好設定，代理程式都會以 Build Forge 引擎主機的作業系統語言來記載系統訊息和工作輸出的資料。

為了避免「管理主控台」介面中出現混合語言，請確定針對「管理主控台」選取的語言符合 Build Forge 引擎主機所使用的語言。

設定語言

為了確保「管理主控台」中顯示的語言，符合代理程式用來記載系統訊息和工作輸出之資料的語言，請針對 Web 瀏覽器、「管理主控台」主機上的作業系統和 Build Forge 使用者使用相同的語言。

資料庫主機的國際資料支援

如果要顯示及操作國際資料，請依下列方式配置「管理主控台」資料庫的主機：

- 使用 Unicode UTF-8 字集。
- 安裝您要用來顯示資料的字型。

在「管理主控台」中變更使用者語言喜好設定

所有「管理主控台」使用者的預設語言，最初都是設為 Web 瀏覽器的語言。

若要變更 Build Forge 使用者的語言設定，請選取 **管理 > 使用者 > 語言**。

管理主控台以針對使用者選取的語言來顯示使用者介面控制項，但代理程式繼續以 Build Forge 引擎主機的作業系統語言來記載系統訊息和工作輸出的資料。

判斷 UNIX/Linux 主機的語言/字集

如果「管理主控台」或代理程式是安裝在 UNIX/Linux 主機上，請依照下列方式來使用 `locale` 指令。

- 如果要判斷作業系統目前使用的語言/字集，請執行下列動作：
`$ locale`
- 如果要判斷作業系統所能使用的語言/字集組合，請執行下列動作：
`$ locale -a`
- 如果要在登入時設定語言環境，請使用 `rc` 或設定檔 `Script`。

確定 Windows 主機的語言字碼頁

如果 Windows 主機上已安裝管理主控台或代理程式，請依照下列方式來使用 `chcp` 指令：

- 如果要判斷作用中的字碼頁號碼，請輸入：
`> chcp`
- 如果要設定字碼頁，請輸入語言的號碼：
`> chcp code_page`

下表列出 Build Forge 支援的 NLV1 語言的 Windows 字元編碼：

語言	字碼頁
英文	1252
法文	
西班牙文	
義大利文	
德文	
葡萄牙文	
日文 Shift-JIS	932
韓文	949
簡體中文 GBK	936
繁體中文 Big5	950

第 5 章 安裝規劃

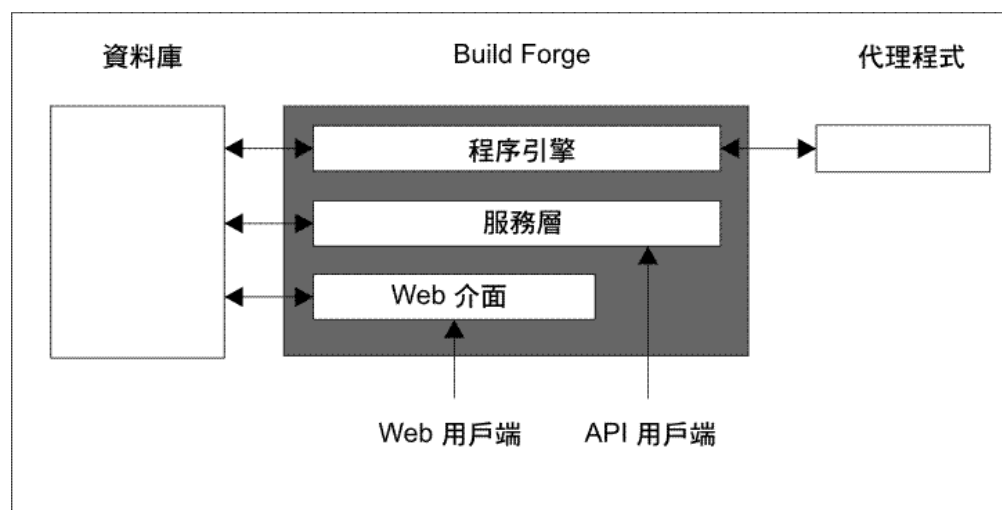
安裝 Rational® Build Forge® 時，需要您考量數個產品元件。

這個主題說明安裝 Rational® Build Forge® 產品元件所需的規劃。它可作為您在選擇安裝及配置的方式及項目時的導覽圖。

另請參閱第 527 頁的第 40 章，『效能和可調整性』中的規劃資源說明。

元件

Rational® Build Forge® 系統具有下列元件：



- **Web 用戶端**：供使用 Web 瀏覽器的使用者和管理者用來存取系統。瀏覽器用戶端會存取 **Web 介面** 元件。
- **API 用戶端**：使用 Java API 或 Perl API 來存取 Rational® Build Forge® 的任何程式。API 用戶端會直接存取**服務層**。
- **Build Forge**：系統的總稱。在安裝期間，系統會顯示下列核心產品特性：
 - **Web 介面**：稱為管理主控台或主控台。這個元件是由一組 PHP 模組所組成。
 - **程序引擎**：稱為引擎。引擎會管理工作排程與執行。
 - **服務層**：API 用戶端、Web 介面及程序引擎用來發出要求的資料庫抽象層。

在簡式安裝中，所有三個元件都安裝在同一部主機上。其可以安裝在個別的主機上。以 Rational® Build Forge® 配置檔為例（例如 buildforge.conf），必須修改檔案以容許元件進行通訊。

- **資料庫**：系統的資訊儲存體。資料庫會儲存專案定義、系統配置及使用者配置。
- **代理程式**：安裝在主機上的程式。代理程式必須安裝在您要管理主控台用作伺服器資源的每一部主機上。代理程式會接收執行工作或步驟的要求，並在安裝該代理程式的主機上執行這些要求。

您可以利用各種方式來部署元件。部署方式的範圍從在單一主機上部署所有元件，到使用叢集式主控台與大量分散式伺服器資源的系統。

部署類型

本節說明下列部署類型：

- 獨立式
- 一般正式作業安裝
- 分層安裝

獨立式

獨立式部署會將資料庫、「管理主控台」和一個代理程式部署在單一主機上。這個部署通常用於評估或開發。

一般

一般安裝會將資料庫、「管理主控台」和代理程式放在不同的主機上。這個部署大多數用於正式作業環境。

分層安裝

在分層部署中，通常會有三個一般部署：

- 開發（又稱為草擬或沙盤推演）：由開發人員用來建立新專案。它也能用來作為升級時的起始區域。
- 測試：由品管及開發人員用來測試系統。
- 正式作業：由所有人在日常業務中用來執行工作。

Build Forge 資料會從某一層匯出，再匯入下一層。

重要：這三個部署應該對稱。每一個部署都使用相同的作業系統、資料庫及元件。

安裝方法

本節說明您可以使用的安裝方法。

- Installation Manager，互動式：在安裝過程中提供輸入
- Installation Manager，無聲自動式：將輸入儲存成檔案，然後在安裝期間使用該檔案來提供輸入
- 安裝來使用您自己的元件：繼續使用 Build Forge 需要的現有元件
- 在虛擬映像檔上安裝：請參閱在 VMware 上安裝的相關指南
- 在 IBM System z® 上安裝：瞭解在 SUSE Linux on System z 上安裝主控台所需的工具及資訊

配置選項

這個主題說明需要您變更產品配置的特性。

- 埠指派：瞭解如何使用預設埠 80 以外的埠

- 第 101 頁的『關於預設登入安全』：您在安裝期間會提供金鑰儲存庫密碼。金鑰儲存庫會供安全登入機制（依預設會提供）和 SSL 使用，但需要額外的設定。如果需要，您可以停用安全登入。
- 密碼加密：將 Build Forge 引擎、代理程式、服務層及資料庫所使用的密碼加密
- SSL 和 HTTPS：瞭解在整個 Build Forge 系統中使用 SSL 所需的步驟
- 單一登入：在簡化登入的同時又能保持安全
- 使用主控台冗餘來支援更高傳輸量：設定多個主控台來使用一個資料庫
- 使用 Build Catalyst 來加速 make 型建置：設定單一系統平行建置及多重系統分散式建置

與其他產品整合

- WebSphere®：使用 WebSphere Application Server，而不使用 Apache Tomcat
- Rational Team Concert™ IDE：將 Build Forge 設為 RTC 建置伺服器，並執行其他作業
- Eclipse 及 Rational Application Developer IDE：瞭解如何從 Eclipse 及 Rational Application Developer IDE 內存取主控台
- 使用配接器來存取原始檔：使用提供的配接器，然後設計您自己的配接器，以與其他產品互動

第 6 章 前置安裝設定

本節說明在執行安裝程式之前所需的前置安裝設定。

註：如果您已安裝 Build Forge 7.1 版或更早的版本，請參閱第 181 頁的第 12 章，『從舊版升級』。

- 安裝角色（必要）。您必須以管理層級使用者的身分安裝軟體。
- 國際資料支援（必要）。您必須先配置 Web 瀏覽器和資料庫以支援 UTF-8 字集，才能安裝 Build Forge。
- 設定資料庫（必要）。設定資料庫，與 Build Forge 搭配使用。您通常需要建立資料庫物件（包含資料庫使用者和密碼）、在執行 Build Forge 的主機上安裝資料庫用戶端，以及組合在 Build Forge 安裝期間提示您輸入的資訊。
- 安全（選用）。如果您打算使用 SSL/HTTPS，您需要提供憑證或計劃，供 Build Forge 安裝自簽憑證。

國際資料設定

您必須設定 Build Forge 元件，才能支援國際資料。

程序

1. 配置 Web 瀏覽器。
 - a. 設定語言。
 - b. 確定已安裝正確的字型。
2. 配置代理主機來使用 UTF-8 字集。

在 Windows 上，請使用 chcp 指令來檢查字碼頁：

```
> chcp
```

在 UNIX 或 Linux 上，請利用下列指令來檢查語言環境和字集：

```
locale
```

您應該會看到指定語言和字集的值。下列範例來自於 Solaris 系統，其語言為 US English、字集為 UTF-8：

```
LANG=en_US.UTF-8  
LC_CTYPE="en_US.UTF-8"
```

3. 配置資料庫來使用 UTF-8 字集以及支援國際資料的字型。

- **DB2：**

- a. 設定字碼集與地區。範例：CREATE DATABASE USING CODESET UTF-8 TERRITORY US（或在「控制中心」中選取適當的字碼集和地區）。
- b. 在 1208 的管理主控台電腦上設定 DB2CODEPAGE 環境變數。

在 Windows 上，使用指令：

```
set DB2CODEPAGE=1208
```

在 UNIX 或 Linux 上，使用指令：

```
export DB2CODEPAGE=1208
```

如果現有資料庫中存在您需要移轉至 UTF-8 的資料，下列文件將非常有用：<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/t0024033.htm>

- **Microsoft SQL Server**（僅限與 Windows 上的「管理主控台」搭配使用）：不支援國際資料。Microsoft SQL Server 利用 UCS-2 來儲存 Unicode 資料，而不支援 UTF-8。
- **MySQL**：設定伺服器字集與對照。如果目前您的 MySQL 安裝並不支援國際資料，您可以從原始檔重新編譯，然後使用 `./configure --with-charset=utf8 --with-collation=utf8_bin`。若未配置此支援，就不會啟動 Build Forge 引擎。
- **Oracle**：在您安裝實例時，在實例上將字集設為 **UTF8 - Unicode 3.0**。在 Database Configuration Assistant 中，您可以在 Character Sets 標籤上的 Initialization Parameters 步驟中進行這項設定。如果您使用指令行，請將字集設為 **AL32UTF8**。

4. （選用）配置 Build Catalyst 特性的訊息。

在 Microsoft® Windows® 平台上，會自動以正確的翻譯文字來顯示 Build Catalyst 訊息。

若要在 UNIX 及 Linux 平台上檢視這些訊息，請將 NLSPATH 環境變數設為下列值：

```
/opt/rational/buildforge/buildcatalyst/lib/nls/%L/utf8/%N.cat:$NLSPATH
```

其中：

- %L 代替 LANG 環境變數的值
- %N 代替傳給 catopen(3C) 的 name 參數值

或者，您也可以明確地指定翻譯。下列值適用於 ja_JP 翻譯：

```
/opt/rational/buildforge/buildcatalyst/lib/nls/ja_JP/utf8/%N.cat:$NLSPATH
```

資料庫備份（必要）

在進行全新安裝、重新安裝或升級作業前，**備份資料庫**是極為重要的。

若是在安裝期間發生問題，您即可利用資料庫備份將資料庫還原至嘗試安裝前的前次狀態。唯一提供的回復方法是進行資料庫備份。

資料庫設定

本節包含每一種受支援資料庫的設定指示。

資料庫可能與「管理主控台」位於同一部主機上，也可能位於不同主機。

註：Rational Build Forge Build Forge 7.1 版本需要您使用空資料庫來安裝。如果資料庫不是空的，安裝會失敗。

設定需要完成下列部分或所有作業，端視您的資料庫而定：

- 建立供 Build Forge 使用的資料庫物件（資料庫、資料庫使用者）。

- 安裝必要的用戶端軟體供 Build Forge 使用。
- 決定在安裝期間需要哪些其他資訊。一般而言，這是 Apache Tomcat 應用程式伺服器會使用的 JDBC 驅動程式位置。
- 配置 UTF-8 字集的支援和適當的對照。一般而言，當您建立資料庫時會指定國際資料的支援；建立資料庫之後就無法配置國際資料支援。您必須安裝打算用來顯示資料的字型。**Build Forge 需要使用國際資料（UTF-8 字集）** >。

DB2 設定

您可以使用這個程序來設定 DB2 支援。

Red Hat 及 SuSe Linux 6 的 DB2 需求

開始之前

Red Hat Linux 6 上的 DB2 需要額外的設定，您才能安裝 Rational Build Forge。

程序

- 安裝相容性共用程式庫。您必須安裝以下套件：
 - libgcc-4.4.4-13.i686
 - glibc.i686，它會拉出 nss-softoken-freebl.i686。
 - gtk2.i686，它會拉出下列其他的檔案：atk.i686、audit-libs.i686、avahi-libs.i686、cairo.i686、cups-libs.i686、dbus-libs.i686、expat.i686、fontconfig.i686、freetype.i686、gamin.i686、glib2.i686、gnutls.i686
 - libXtst.i686
 - libuuid.i686
- 以下會按字母順序列出所有檔案。請注意，副檔名是適用於 Intel 平台。若為 System z 平台，請使用 .s390 副檔名。

```
atk-1.28.0-2.el6.i686
audit-libs-2.0.4-1.el6.i686
avahi-libs-0.6.25-8.el6.i686
cairo-1.8.8-3.1.el6.i686
cups-libs-1.4.2-35.el6_0.1.i686
dbus-libs-1.2.24-4.el6_0.i686
expat-2.0.1-9.1.el6.i686
fontconfig-2.8.0-3.el6.i686
freetype-2.3.11-6.el6_0.2.i686
gamin-0.1.10-9.el6.i686
glib2-2.22.5-5.el6.i686
glibc-2.12-1.7.el6_0.5.i686
gnutls-2.8.5-4.el6.i686
gtk2-2.18.9-4.el6.i686
jasper-libs-1.900.1-15.el6.i686
keyutils-libs-1.4-1.el6.i686
krb5-libs-1.8.2-3.el6_0.7.i686
libcom_err-1.41.12-3.el6.i686
**libgcc-4.4.4-13.el6.i686
libgcrypt-1.4.5-3.el6.i686
libgpg-error-1.7-3.el6.i686
libjpeg-6b-46.el6.i686
libpng-1.2.44-1.el6.i686
libselinux-2.0.94-2.el6.i686
libstdc++-4.4.4-13.el6.i686
libtasn1-2.3-3.el6.i686
libthai-0.1.12-3.el6.i686
libtiff-3.9.4-1.el6_0.3.i686
```

```

libuuid.i686
libX11-1.3-2.el6.i686
libXau-1.0.5-1.el6.i686
libxcb-1.5-1.el6.i686
libXcomposite-0.4.1-2.el6.i686
libXcursor-1.1.10-2.el6.i686
libXdamage-1.1.2-1.el6.i686
libXext-1.1-3.el6.i686
libXfixes-4.0.4-1.el6.i686
libXft-2.1.13-4.1.el6.i686
libXi-1.3-3.el6.i686
libXinerama-1.1-1.el6.i686
libXrandr-1.3.0-4.el6.i686
libXrender-0.9.5-1.el6.i686
libXtst-1.0.99.2-3.el6.i686
nss-softokn-freebl-3.12.8-1.el6_0.i686
pango-1.28.1-3.el6_0.5.i686
pixman-0.18.4-1.el6_0.1.i686
zlib-1.2.3-25.el6.i686

```

- 若您是在 SuSe Linux 64 位元系統上執行，還必須安裝 pam.i686 套件。

DB2 的資料庫物件

程序

1. 在您的作業系統中建立一個使用者，「管理主控台」將會使用此名稱來登入資料庫。範例：使用者名稱爲 **BUILD**，密碼爲 **BUILD**。

在 DB2 中執行剩餘步驟。重要事項：資料庫名稱和綱目名稱皆必須全部使用大寫字母。

2. 建立一個名爲 **BUILD** 的資料庫
3. 建立一個名爲 **BUILD** 的綱目。

結果

註：Build Forge 會利用使用者綱目來存取資料庫。

範例 DB2 SQL 指令 Script

在「DB2 指令行處理器」中使用下列指令，以建立資料庫。

```

// 建立資料庫和綱目
db2 "CREATE DATABASE BUILD USING CODESET UTF-8
    TERRITORY US PAGESIZE 16 K AUTOCONFIGURE
    USING MEM_PERCENT 20 APPLY DB ONLY"
db2 "CONNECT TO BUILD"
db2 "CREATE SCHEMA BUILD"
db2 "CONNECT RESET"

// 授權給使用者
db2 "CONNECT TO BUILD"
db2 "GRANT CREATEIN,DROPIN,ALTERIN ON SCHEMA BUILD
    TO USER BUILD WITH GRANT OPTION"
db2 "GRANT DBADM,CREATETAB,BINDADD,CONNECT,
    CREATE_NOT_FENCED_ROUTINE,IMPLICIT_SCHEMA, LOAD,
    CREATE_EXTERNAL_ROUTINE,QUIESCE_CONNECT
    ON DATABASE TO USER BUILD"
db2 "commit work"
db2 "CONNECT RESET"
db2 "terminate"

```

建議用於 DB2 的調整參數

關於這項作業

設定 DB2 調整參數可以提昇 Build Forge 系統使用現有 DB2 資料庫的效能和可調整性。

註：如果您在安裝並執行 Build Forge 之後 變更這些參數，請先停止 Build Forge 再進行變更。重新啟動 DB2 之後，請重新啟動 Build Forge。

程序

1. 設定調整參數。執行下列 DB2 指令：

```
db2set DB2_EVALUNCOMMITTED=ON
db2set DB2_SKIPDELETED=ON
db2set DB2_SKIPINSERTED=ON
```

請參閱 DB2 文件以取得這些設定的影響的相關資訊。

2. 重新啟動 DB2。要使參數生效，則需要執行此步驟。請先確定資料庫上沒有階段作業在執行中。

```
db2stop force
db2start
```

DB2 用戶端驅動程式

您必須先在主機上安裝 IBM DB2 資料庫用戶端驅動程式，才能安裝產品。在 UNIX 或 Linux 上，請使用 32 位元驅動程式。

程序

1. 請安裝 DB2 Data Server Client 以提供驅動程式。
2. 安裝 DB2 用戶端之後，重新啟動電腦。否則產品安裝便會失敗。

安裝期間所需的 DB2 資訊

關於這項作業

在安裝期間，系統會在**資料庫配置**畫面中要求您輸入下列資訊：

資料庫配置

- **資料庫主機：**用來安裝 DB2 的主機。
- **資料庫埠：**針對 DB2，Build Forge 會在這個欄位中填入預設埠 50000。如果您使用不同的埠，請準備輸入埠號。
- **資料庫名稱：**Build Forge 要使用的資料庫名稱。您已在之前的安裝步驟建立這個資料庫。
- **資料庫綱目名稱：**Build Forge 要使用的綱目名稱。
- **資料庫使用者名稱：**Build Forge 在存取資料庫時所使用的使用者名稱。您已在之前的安裝步驟建立這個使用者。
- **密碼：**資料庫使用者名稱的密碼。

測試資料庫配置

- **DB2 用戶端程式庫的路徑 -** DB2 用戶端程式庫所在的目錄。

重要：在 UNIX 或 Linux 上安裝 Build Forge 時，這個目錄必須是包含 32 位元用戶端驅動程式庫的目錄。

- **JDBC 驅動程式位置** - JDBC 驅動程式所在的目錄。Apache Tomcat 會使用這個驅動程式來存取資料庫。一般位置：
 - Windows : `<db2install>/IBM/SQLLIB/java`
 - UNIX 或 Linux : consult the documentation for your system.

畫面上會顯示下列資訊：

- **所需的驅動程式 JAR 檔** - 顯示所需的驅動程式 JAR 檔。針對 DB2，這些是 `db2jcc.jar` 和 `db2jcc_license_cu.jar`。
- **所需的 JDBC 驅動程式類別** - 顯示所需的 JDBC 驅動程式類別。針對 DB2，這是 `com.ibm.db2.jcc.DB2Driver`。

設定 Microsoft SQL Server

請使用這些程序從 Windows® 型「管理主控台」來安裝及配置對 Microsoft SQL Server 資料庫的存取。

Microsoft SQL Server 的資料庫物件

關於這項作業

在這些步驟中，您將建立一個使用者作為資料庫擁有者，並建立資料庫供 Build Forge 使用。必要的話，您也可以建立供 Build Forge 使用的綱目。

程序

1. **建立使用者以擔任資料庫擁有者。** Build Forge 會使用這個使用者名稱來登入資料庫。除非您使用不同名稱，請使用 **build**。使用者必須具備完整許可權。
 - a. 開啓 SQL Server Management Studio。
 - b. 在「物件瀏覽器」（左側畫面）中開啓資料庫伺服器。
 - c. 用滑鼠右鍵按一下**安全資料夾**，並選擇**新建 > 登入**。
 - d. 在「登入 - 新建」對話框中，指定登入名稱，並按照如下所示來選擇選項。重要事項：請取消勾選「使用者在下次登入時必須變更密碼」。
 - 選擇 **SQL Server 鑑別**，並提供密碼。
 - 取消勾選**施行密碼期限**
 - 取消勾選**使用者在下次登入時必須變更密碼**
2. **建立資料庫。** 您必須使用混合模式的鑑別。下列互動式及 Script 範例假設您將使用 DBO 的預設綱目。
 - a. 開啓 SQL Server Management Studio。
 - b. 在「物件瀏覽器」（左側畫面）中開啓資料庫伺服器。
 - c. 用滑鼠右鍵按一下**資料庫資料夾**，並選擇**新建資料庫**。
 - d. 在「新建資料庫」對話框中，指定資料庫的參數：
 - 指定資料庫名稱。除非您必須使用不同名稱，否則請使用 **build**。名稱會區分大小寫。資料和日誌檔的名稱會在「資料庫檔案」框中自動更新。
 - 指定資料庫擁有者。
 - 按一下欄位右側的 **[..]** 控制項。

- 在「選取資料庫擁有者」對話框中按一下**瀏覽**。
- 勾選您建立的使用者名稱，然後按一下**確定**。
- 在「選取資料庫擁有者」對話框中，按一下**確定**。
- 指定資料庫檔案參數。在「資料庫檔案」表格中，執行下列動作：
 - 針對這兩個檔案：將「起始大小」設為 500 (MB)
 - 針對這兩個檔案：設定「自動成長」。在「自動成長」直欄中，按一下 **[..]** 控制項，以開啓對話框。勾選**啓用自動成長**方框，將成長值設為 500 MB，並選取**不限制成長**，然後按一下**確定**。

或者，您可以使用下列 Script 來建立資料庫。

```
CREATE DATABASE [build] ON PRIMARY
( NAME = N'build',
  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL.2\MSSQL\DATA\build.mdf' ,
  SIZE = 2048KB , FILEGROWTH = 1024KB )
LOG ON
( NAME = N'build_log',
  FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL.2\MSSQL\DATA\build_log.ldf' ,
  SIZE = 1024KB , FILEGROWTH = 10%)
GO
EXEC dbo.sp_dbcmtlevel @dbname=N'build', @new_cmptlevel=90
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [build].[dbo].[sp_fulltext_database] @action = 'disable'
end
GO
ALTER DATABASE [build] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [build] SET ANSI_NULLS OFF
GO
ALTER DATABASE [build] SET ANSI_PADDING OFF
GO
ALTER DATABASE [build] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [build] SET ARITHABORT OFF
GO
ALTER DATABASE [build] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [build] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [build] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [build] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [build] SET CURSOR_CLOSE_ON_COMMIT ON
GO
ALTER DATABASE [build] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [build] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [build] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [build] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [build] SET READ_COMMITTED_SNAPSHOT ON
GO
ALTER DATABASE [build] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [build] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [build] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
```

```

ALTER DATABASE [build] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [build] SET READ_WRITE
GO
ALTER DATABASE [build] SET RECOVERY FULL
GO
ALTER DATABASE [build] SET MULTI_USER
GO
ALTER DATABASE [build] SET PAGE_VERIFY CHECKSUM
GO
USE [build]
GO
IF NOT EXISTS (SELECT name FROM sys.filegroups WHERE is_default=1 AND name = N'PRIMARY') \
    ALTER DATABASE [build] MODIFY FILEGROUP [PRIMARY] DEFAULT
GO

```

註：READ_COMMITTED_SNAPSHOT 屬性必須設為 ON。

若要測試 READ_COMMITTED_SNAPSHOT，請執行下列 SQL 陳述式。

```
SELECT is_read_committed_snapshot_on FROM sys.databases WHERE name='build'
```

如果您不是使用 build，可換成您的資料庫名稱。陳述式必須傳回 "1" 才行。

如果陳述式傳回 "0"，請執行下列 SQL 陳述式。

```
ALTER DATABASE [build] SET READ_COMMITTED_SNAPSHOT ON
```

3. 設定使用者的預設資料庫。

- 開啓 SQL Server Management Studio。
- 在「物件瀏覽器」（左側畫面）中開啓資料庫伺服器。
- 在「物件瀏覽器」中開啓安全 > 登入。
- 用滑鼠右鍵按一下您建立的使用者，然後選擇內容。
- 在「一般」頁面中，選取一個預設資料庫。請選取您建立的資料庫。
- 按一下確定。

- 建立綱目（必要的話）。如果您可以使用預設綱目 **DBO**，則無須建立綱目。建議您使用預設值。如果您必須使用不同的綱目，請在 Build Forge 資料庫中建立，然後將擁有者指派為您所建立的資料庫使用者。下列範例會建立綱目 bf_schema，並將 build 設為擁有者：

```

USE build;
GO
CREATE SCHEMA bf_schema AUTHORIZATION build;
GO

```

在 Microsoft SQL Server 上設定 TCP/IP

關於這項作業

您必須在 Microsoft SQL Server 上啓用 TCP/IP，才能將其與「管理主控台」搭配使用。

若要在 MS SQL Server 上啓用 TCP/IP，請執行下列動作：

程序

- 開啓 MSSQLServer 組態管理員。
- 在 **SQL Server 網路組態** 下方，按一下 **MSSQLSERVER 通訊協定**。
- 用滑鼠右鍵按一下 **TCP/IP**，然後從功能表中選擇啓用。

安裝 Microsoft SQL Server 用戶端和 JDBC 驅動程式 關於這項作業

您需要在「管理主控台」主機上安裝 Microsoft SQL Server 用戶端和 JDBC 驅動程式。
「管理主控台」會使用它們來存取資料庫。

程序

1. 安裝 SQL Native Client。您需要 2005.90.4035.00 版。其包含在 Microsoft SQL Server 的 Service Pack 3 中。
2. 安裝 JDBC 驅動程式。
 - 2005 版：需要「JDBC 程式庫」1.2 版。從 Microsoft 的下列位置下載程式庫：

<http://www.microsoft.com/downloads/details.aspx?FamilyId=C47053EB-3B64-4794-950D-81E1EC91C1BA&displaylang=en>

完成 Microsoft 安裝程序之後，JAR 的位置如下所示：

/sqljdbc_1.2/enu/sqljdbc.jar

- 2008 版：需要「JDBC 程式庫」3.0 版：

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=%20a737000d-68d0-4531-b65d-da0f2a735707&displaylang=en>

- 2012 版：需要「JDBC 程式庫」4.0 版。

Microsoft SQL Server 的 ODBC 資料來源設定 關於這項作業

透過這些步驟，您將會建立一個 Windows 型「管理主控台」所使用的 ODBC 資料來源，以存取您在 Microsoft SQL Server 中建立的資料庫。不論是執行 32 位元或 64 位元的 Windows 系統，您必須建立 32 位元的 ODBC 資料來源。

程序

1. 啟動「資料來源管理員」。
 - 若為 32 位元 Windows 系統：從 Windows 開始功能表，選取**設定 > 控制台 > 系統管理工具 > 資料來源**。
 - 若為 64 位元 Windows 系統：按兩下檔案 C:\Windows\SysWOW64\odbcad32.exe。

這時會顯示**資料來源管理員**對話框，可讓您在 64 位元系統上建立所需的 32 位元 ODBC 資料來源。

2. 在**系統資料來源名稱**標籤中，按一下**新增**。出現一份驅動程式清單。
3. 從驅動程式清單中選取 **MS SQL Server**，然後按一下**完成**。在出現的 **ODBC 設定**對話框中，輸入下列資訊：
 - **資料來源名稱**：這個資料來源的名稱 必須與「資料庫名稱」相同，且不能與資料庫的相關綱目名稱相同。
 - **說明**：此資料來源的說明。
 - **伺服器名稱**：將安裝 MS SQL Server 資料庫所在主機的主機名稱。
 - **資料庫名稱**：您在上述步驟中建立的資料庫名稱。

按一下**確定**以關閉 **ODBC 設定**，然後按一下**確定**來關閉 **ODBC 資料來源管理員**。
資料來源已經建立完成。

結果

請記住下列資訊。在您安裝「管理主控台」時，安裝程式會要求您提供這些資訊。

- **資料來源名稱**：即您在建立 ODBC 資料來源時所指派的名稱
- **資料來源類型**：SQL Server
- **使用者名稱**：您所建立的資料庫使用者名稱（例如，**build**）
- **密碼**：該使用者名稱的密碼（例如，**build**）

重要：請針對「資料庫名稱」和「資料來源名稱」使用相同的值。JDBC 驅動程式中的限制條件要求採用此限制。如果這兩個值不同，「快速報告」報告特性和 Java 與 Perl 的服務層 API 就無法運作。

安裝期間所需的 Microsoft SQL Server 資訊

關於這項作業

在安裝期間，系統會在**資料庫配置**畫面中要求您輸入下列資訊：

資料庫配置

- **資料庫主機**：用來安裝 SQL Server 的主機。
- **資料庫埠**：針對 SQL Server，Build Forge 會在這個欄位中填入預設埠 1521。如果您使用不同的埠，請準備輸入埠號。
- **資料庫名稱**：Build Forge 要使用的資料庫名稱。您已在之前的安裝步驟建立這個資料庫。
- **資料庫綱目名稱**：供 Build Forge 使用的綱目名稱，可以是 DBO 的預設值（建議使用）或您在先前的設定步驟中建立的綱目名稱。
- **ODBC 資料來源名稱**：ODBC 資料來源的名稱。
- **資料庫使用者名稱**：Build Forge 在存取資料庫時所使用的使用者名稱。您已在之前的安裝步驟建立這個使用者。
- **密碼**：資料庫使用者名稱的密碼。

測試資料庫配置

- **SQL Server 用戶端程式庫的路徑** - SQL Server 用戶端程式庫所在的目錄。

重要：Microsoft SQL Server 只在 Windows 平台受支援。

- **JDBC 驅動程式位置** - JDBC 驅動程式所在的目錄。Apache Tomcat 會使用這個驅動程式來存取資料庫。
- 2005 版：使用 2005 版提供的驅動程式。
- 2008 版：使用 3.0 版驅動程式。
- 2012 版：使用 4.0 版驅動程式。

畫面上會顯示下列資訊：

- **所需的驅動程式 JAR 檔** - 顯示所需的驅動程式 JAR 檔。
 - 若為 SQL Server 2005，檔案為 sqljdbc.jar
 - 若為 SQL Server 2008 和 2012，檔案為 sqljdbc4.jar
- **所需的 JDBC 驅動程式類別** - 顯示所需的 JDBC 驅動程式類別。若為 SQL Server，驅動程式類別為 com.microsoft.sqlserver.SQLServerDriver

MySQL 設定

請利用此程序來安裝及配置針對 MySQL 的支援。

開始之前

安裝及配置下列項目。請使用下列各節中的指示。

MySQL 在 Red Hat Linux 4 上的需求

開始之前

Red Hat Linux 4 上的 MySQL 在配置 Rational Build Forge 之前，還需要進行額外的設定。

程序

- 安裝 `ncurses-libs.i686`。它必須存在，才能在下一步中安裝檔案。
- 安裝相容性共用程式庫。您需要安裝套件或 RPM，但不需同時安裝這兩者。
`MySQL-client-community-5.0.51a-0.rhel5.i386`
`MySQL-shared-compat-5.0.51a-0.rhel5.i386`
- 重新命名啟動 Script。您必須重新命名啟動 Script，才能比預設啟動程序中所設定的時間更早地載入 MySQL。下列範例應該已將時間移得夠早了。

```
mv /etc/rc3.d/S99mysql /etc/rc3.d/S50mysql
```

MySQL 的資料庫物件

程序

1. 建立名為 **build** 的空白資料庫。
2. 建立與其關聯的使用者（使用者名稱為 **build**，密碼為 **build**）。

結果

您可以使用下列指令來建立資料庫 **build**，並建立使用者 **build@localhost**，其密碼為「所識別的」**build**：

```
mysql -u root
mysql> create database build;
mysql> grant all on build.* to build@localhost
-> identified by "build";
```

MySQL 程式庫及用戶端驅動程式

開始之前

您必須先在 Build Forge 主機上安裝 MySQL 原生用戶端驅動程式，才能安裝 Build Forge。在 UNIX 或 Linux 上，請使用 32 位元驅動程式。

如果您使用 MySQL 5.1，則必須使用 MySQL 5.0 用戶端。

如果安裝 MySQL 的主機與 Build Forge 主機相同，請安裝下列程式：

MySQL 資料庫伺服器
MySQL-client-community-5.0.51a-0.rhel5.i386 - 用戶端驅動程式
MySQL-shared-compat-5.0.51a-0.rhel5.i386 - 必要的相容性程式庫
Connector/J JDBC 驅動程式

如果您是在不同的主機上安裝 MySQL，請在 Build Forge 主機上安裝下列程式：

MySQL-client-community-5.0.51a-0.rhel5.i386 - 用戶端驅動程式
MySQL-shared-compat-5.0.51a-0.rhel5.i386 - 必要的相容性程式庫
Connector/J JDBC 驅動程式

僅適用於 AIX® 系統：

您必須安裝 Connector/J 驅動程式及 5.0-51a 用戶端驅動程式。它們與 AIX 6.1 版相容。您可以在 MySQL 網站的「保存」區域中找到它們。

安裝程式會嘗試重新包裝 MySQL 用戶端共用程式庫。Build Forge 不能使用 mysql.com 上公告的包裝檔案方式。執行安裝程式的使用者需要對其中安裝用戶端驅動程式檔的目錄（通常是 /opt/mysql）有寫入權。安裝程式會置換 libmysqlclient.a 和 libmysqlclient.so.15 檔。如果安裝程式無法重新包裝檔案，其會在安裝期間通知您並繼續安裝。在此情況下，您必須先手動重新包裝檔案，再執行 Build Forge。

註：如果有其他執行中的應用程式使用這個 MySQL 用戶端，您可能要在另一個目錄中手動重新包裝檔案。

如果要手動包裝檔案，請執行下列動作。

1. 從其中安裝 MySQL 的目錄，建立新的目錄 libbf：

```
root@myaix:/opt/mysql/> mkdir libbf
```

2. 將 lib/libmysqlclient.so.15 檔複製到新的目錄：

```
root@myaix:/opt/mysql/> cp lib/libmysqlclient.so.15 libbf
```

3. 切換至新的目錄，並建置新的保存檔：

```
root@myaix:/opt/mysql/> cd libbf
```

```
root@myaix:/opt/mysql/libbf/> ar -q libmysqlclient.a libmysqlclient.so.15
```

如果尚未執行安裝程序，這個目錄現在可以用來作為安裝程序的用戶端程式庫路徑。如果使用了其中一種無聲自動或指令行方法，請更新 \$BFROOT/rc/buildforge 中的 LIBPATH 來使用這個路徑。

MySQL 配置

程序

1. 將資料庫連線上限數增加至 200。編輯 <mysql-installdir>/my.ini (Windows) 或 /etc/my.cnf (UNIX/Linux) 中的 [mysqld] 區段，如下所示：

```
max_connections=200
```

值至少應該等於**主控台程序數上限與執行佇列大小系統設定值**（位於「管理主控台」的**管理 → 系統**頁面）的總和。

2. 確認您的 MySQL 實例是使用 InnoDB 儲存體引擎。Rational Build Forge 資料庫會使用外部索引鍵限制。MySQL MyISAM 儲存體引擎不會檢查外部索引鍵限制。如果使用 MyISAM，則在建立新的資料庫物件時可能會發生問題，尤其是透過 API 時更是如此。

安裝期間所需的 MySQL 資訊

關於這項作業

在安裝期間，系統會在**資料庫配置**畫面中要求您輸入下列資訊：

資料庫配置

- **資料庫主機**：用來安裝 MySQL 的主機。
- **資料庫埠**：針對 MySQL，Build Forge 會在這個欄位中填入預設埠 3306。如果您使用不同的埠，請準備輸入埠號。
- **資料庫名稱**：Build Forge 要使用的資料庫名稱。您已在之前的安裝步驟建立這個資料庫。
- **資料庫使用者名稱**：Build Forge 在存取資料庫時所使用的使用者名稱。您已在之前的安裝步驟建立這個使用者。
- **密碼**：資料庫使用者名稱的密碼。

測試資料庫配置

- **MySQL 用戶端程式庫的路徑** - MySQL 用戶端程式庫所在的目錄。

重要：在 UNIX 或 Linux 上安裝 Build Forge 時，這個目錄必須是包含 32 位元用戶端驅動程式庫的目錄。

- **JDBC 驅動程式位置** - JDBC 驅動程式所在的目錄。Apache Tomcat 會使用這個驅動程式來存取資料庫。

畫面上會顯示下列資訊：

- **所需的驅動程式 JAR 檔** - 顯示所需的驅動程式 JAR 檔。針對 MySQL，這是 mysql-connector-java-5.*-bin.jar。
- **所需的 JDBC 驅動程式類別** - 顯示所需的 JDBC 驅動程式類別。針對 MySQL，這是 com.mysql.jdbc.Driver。

MySQL 疑難排解

關於這項作業

此作業將說明使用 MySQL 資料庫的主控台失效，且出現「找不到資料庫延伸 'mysql_i」訊息時，所應採取的暫行解決方法。在使用下列檔案時，Red Hat Linux 6.x 版就會發生此問題：

/opt/buildforge/server/apache/lib/php/extensions/no-debug-non-zts-20090626

程序

1. 檢查是否缺少程式庫與許可權。請使用 ldd 指令。範例回應：

```
ldd: warning: you do not have execution permission for `./mysql_i.so'
linux-gate.so.1 => (0x00413000)
libmysqlclient.so.15 => not found
.....
libssl.so.6 => not found
libcrypto.so.6 => not found
```

2. 執行下列其中一項作業，以找出 libmysqlclient.so.15 程式庫。

- 將 libmysqlclient.so.15 的完整路徑附加至 LD_LIBRARY_PATH。
- 將 libmysqlclient.so.15 複製到 /usr/lib。

3. 將鏈結新增至舊版程式庫。您必須以手動方式將鏈結新增至舊版程式庫。範例：

```
ln -s /usr/lib/libcrypto.so.1.0.0 /usr/lib/libcrypto.so.6
ln -s /usr/lib/libssl.so.1.0.0 /usr/lib/libssl.so.6
```

4. 重新啟動主控台。重新啟動是必要作業。

Oracle 設定

您可以使用這個程序來設定 Oracle 資料庫的支援。

開始之前

安裝或配置下列項目。請使用下列各節中的指示。

Oracle 10 的 Red Hat Linux 6 需求

開始之前

Red Hat Linux 6 上的 Oracle 10 需要額外的設定，才能安裝 Rational Build Forge。

程序

安裝相容性共用程式庫。您必須安裝以下套件：

- compat-libs-5.2-1.i386.rpm
- compat-libstdc++-33-3.2.3-68.i686

Oracle 的資料庫物件

關於這項作業

在 Oracle 主機上建立一個本端使用者：使用者名稱爲 **build**，密碼爲 **build**。

- 新增適當的授與權限，包含 CREATE SESSION 及 CREATE TABLE。
- 在 DEFAULT TABLESPACE 中新增適當的 QUOTA 大小，爲系統提供足夠的空間來儲存資料。

```
create user build
  identified by password
  default tablespace users
  quota unlimited on users;
```

```
grant create session, create table
  to build;
```

重要：在安裝期間，資料庫名稱與 Oracle SID 會使用相同的值。JDBC 驅動程式中的限制條件要求採用此限制。如果這兩個值不同，「快速報告」報告特性和 Java 與 Perl 的服務層 API 就無法運作。

調整建議給 Oracle 的參數

關於這項作業

必須變更某些 Oracle 參數的預設值，Build Forge 才能正確執行。

註：如果您在安裝並執行 Build Forge 之後 變更這些參數，請先停止 Build Forge 再進行變更。重新啟動資料庫伺服器之後，請重新啟動 Build Forge。

程序

1. 設定調整參數。執行下列指令：

```
ALTER SYSTEM SET open_cursors=1000 SCOPE=BOTH  
ALTER SYSTEM SET processes=500 SCOPE=BOTH
```

註：在您執行 ALTER SYSTEM SET processes=500 SCOPE=BOTH 時，如果收到「SQL 錯誤：ORA-02095：無法修改指定的起始設定參數」訊息，請改用 ALTER SYSTEM SET processes=500 SCOPE=SPFile。

請參閱 Oracle 文件以取得這些設定的影響的相關資訊。

2. 重新啟動資料庫伺服器。要使參數生效，則需要執行此步驟。請先確定資料庫上沒有階段作業在執行中。

安裝期間所需的 Oracle 資訊

關於這項作業

在安裝期間，系統會在資料庫配置畫面中要求您輸入下列資訊：

資料庫配置

- **資料庫主機**：用來安裝 Oracle 的主機。
- **資料庫埠**：針對 Oracle，Build Forge 會在這個欄位中填入預設埠 1521。如果您使用不同的埠，請準備輸入埠號。
- **資料庫名稱**：Build Forge 要使用的資料庫名稱。您已在之前的安裝步驟建立這個資料庫。
- **資料庫使用者名稱**：Build Forge 在存取資料庫時所使用的使用者名稱。您已在之前的安裝步驟建立這個使用者。
- **密碼**：資料庫使用者名稱的密碼。

測試資料庫配置

- **Oracle 用戶端程式庫的路徑** - Oracle 用戶端程式庫所在的目錄。

重要：在 UNIX 或 Linux 上安裝 Build Forge 時，這個目錄必須是包含 32 位元用戶端驅動程式庫的目錄。

- **ORACLE_HOME 環境變數** - 用來安裝 Oracle 的目錄。
- **tnsnames.ora 檔的路徑 (TNS_ADMIN)** - 包含 tnsnames.ora 檔的目錄。請確定已設定 tnsnames.ora 檔的完整存取權。
- **JDBC 驅動程式位置** - JDBC 驅動程式所在的目錄。Apache Tomcat 會使用這個驅動程式來存取資料庫。

畫面上會顯示下列資訊：

- **所需的驅動程式 JAR 檔** - 顯示所需的驅動程式 JAR 檔。
 - 若為 Oracle 10，檔案為 ojdbc14.jar
 - 若為 Oracle 11，檔案為 ojdbc5.jar
- **所需的 JDBC 驅動程式類別** - 顯示所需的 JDBC 驅動程式類別。針對 Oracle，這是 oracle.jdbc.driver.OracleDriver。

Oracle 用戶端配置

關於這項作業

如果要安裝及配置 Oracle 的用戶端，請執行下列步驟：

程序

1. 在「管理主控台」主機上安裝 Oracle Instant Client。您必須安裝 32 位元版本，不論您是執行於 32 位元或 64 位元平台，都是如此。請僅使用適合您作業系統的 **Instant Client Package - Basic** 10.2 或 11.2 版。
 - 對 Oracle 10.2 使用 10.2 版用戶端。
 - 對 Oracle 11g 使用 11.2 版用戶端。

您可從 Oracle 下載用戶端，網址為 <http://www.oracle.com/technology/software/tech/oci/instantclient/index.html>。

2. 在「管理主控台」主機上設定環境。必須設定數個環境變數。

- LD_LIBRARY_PATH：設為包含用戶端安裝目錄。

註：您可以在 Installation Manager 的「資料庫配置」頁面中指定這個值。

- NLS_LANG：設為適用於國際語言支援的適當值。
 - UNIX 和 Linux 系統：值必須包含 AL32UTF8。範例：AMERICAN_AMERICA.AL32UTF8。
 - Windows 系統：值必須包含對應於 Windows 系統登錄中「作用中字碼頁」設定的字元對映規格。請參閱『Windows 上的 Oracle 用戶端範例配置』。

NLS_LANG 必須依照說明來明確設定。當搭配 *Build Forge* 來使用時，用戶端安裝期間所設的預設字集不正確。

- ORACLE_HOME：設為 Oracle 用戶端安裝目錄的路徑。

註：您可以在 Installation Manager 的「資料庫配置」頁面中指定這個值。

- ORA_NLS10：設為伺服器上字集資料所在的路徑。
- PATH：設定以直接併入用戶端安裝。
- TNS_ADMIN：設為伺服器上 tnsnames.ora 檔所在的路徑。請確定已設定 tnsnames.ora 檔的完整存取權。

註：您可以在 Installation Manager 的「資料庫配置」頁面中指定這個值。

如果要檢查 Oracle 伺服器的現行語言設定，請登入 Oracle，然後執行下列指令：

```
SQL> host echo $NLS_LANG
```

下一步

重要：在安裝期間，tnsnames.ora 會設為對資料庫名稱與 Oracle SID 使用相同的值。JDBC 驅動程式中的限制條件要求採用此限制。如果這兩個值不同，「快速報告」報告特性和 Java 與 Perl 的服務層 API 就無法運作。

Windows 上的 Oracle 用戶端範例配置：

關於這項作業

範例環境：

- Instant Client - Basic，位於 C:\instantclient_11_2 中，使用美式英文
- Oracle 11.2，位於 Windows 系統的 C:\oracle\product\11.2.0\db_1 中，已安裝來支援國際資料

安裝有用戶端與 Build Forge 之系統的變數設定：

- LD_LIBRARY_PATH 包含 C:\instantclient_11_2\
- NLS_LANG=AMERICAN_AMERICA.WE8MSWIN1252
- ORACLE_HOME=C:\instantclient_11_2\
- ORA_NLS10=C:\oracle\ocommon\nls\admin\data
- PATH 包含 C:\instantclient_11_2\
- TNS_ADMIN=C:\oracle\product\11.2\db_1\network\admin

關於 Windows 系統上的 NLS_LANG

Windows 上的 Oracle 用戶端會使用 Windows 中的設定來執行本端字元對映。

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage\ACP

當用戶端傳送資料至伺服器時，字元會在伺服器的字元對映中傳送。

上述範例 NLS_LANG 設定對應於 ACP 設定 1252，這是 US Windows 的預設值。如果是其他語言，請參考 Oracle InstantClient 文件以取得 NLS_LANG 正確的語言、語言環境，以及字元對映部分。請務必檢查 Windows 系統上實際的 ACP 設定，並使設定的字元對映部分符合該設定。

其他語言的字元對映如下所示。請搭配正確的語言和語言環境使用這些對映，來建立 NLS_LANG 設定。

```
1250 EE8MSWIN1250
1251 CL8MSWIN1251
1252 WE8MSWIN1252
1253 EL8MSWIN1253
1254 TR8MSWIN1254
1255 IW8MSWIN1255
1256 AR8MSWIN1256
1257 BLT8MSWIN1257
1258 VN8MSWIN1258
874  TH8TISASCII
932  JA16SJIS
936  ZHS16GBK
949  KO16MSWIN949
950  ZHT16MSWIN950
```

註：如果未正確設定字元對映，您並不會收到警告，且一開始在行為或效能方面並沒有明顯的差異。用戶端會嘗試在記憶體中進行字元轉換。當記憶體不足且需要切換以執行轉換時，用戶端與伺服器之間的通訊效能會大幅降低。

UNIX 或 Linux 上的 Oracle 用戶端範例配置：

關於這項作業

範例環境：

- Instant Client - Basic，位於 /usr/local/instantclient_11_2 中，使用美式英文

- Oracle 11.2，位於 UNIX 系統的 `/usr/local/oracle/product/11.2.0/db_1` 中，已安裝來支援國際資料

安裝有用戶端與 Build Forge 之系統的變數設定：

- `LD_LIBRARY_PATH` (Solaris、Linux) 或 `LIBPATH` (AIX) 包含 `/usr/local/instantclient_11_2`
- `NLS_LANG=AMERICAN_AMERICA.AL32UTF8`
- `ORACLE_HOME=/usr/local/instantclient_11_2`
- `ORA_NLS10=/usr/local/oracle/ocommon/nls/admin/data`
- `PATH` 包含 `/usr/local/instantclient_11_2`
- `TNS_ADMIN=/usr/local/oracle/product/11.2/db_1/network/admin`

使用 Oracle Instant Client 11.2 的 UNIX 和 Linux 系統

Build Forge 應用程式相依於 `libclntsh.so.10.1` 程式庫，其是 10.2 版用戶端的一部分。在安裝期間，安裝程式會檢查 `ORACLE_HOME` 中是否有這個檔案。如果檔案不存在，安裝程式會假設您是使用 11.2 版用戶端，並嘗試建立從 `libclntsh.so.10.1` 指向對應 11.2 版程式庫 `libclntsh.so.11.1` 的符號鏈結。

```
ln -s libclntsh.so.11.1 libclntsh.so.10.1
```

重要：執行 Build Forge 安裝程式的使用者必須對 `ORACLE_HOME` 目錄有寫入權。如果這在您的網站上是個問題，請先要求授權使用者手動建立該鏈結，再安裝 Build Forge。在某些網站上，用戶端可以安裝在裝載為唯讀的檔案系統上（這是 Solaris 系統的常見作法）。

使用 Oracle RAC 實作失效接手

使用本程序可在使用 Oracle RAC 時啟用主控台失效接手

開始之前

Oracle RAC 的基本資料庫設定，與其他 Oracle 資料庫的設定相同。請使用這些設定來安裝主控台。安裝主控台之後，請使用下列程序實作失效接手。

關於這項作業

若要啟用 Oracle RAC 失效接手以與 Build Forge 配合運作，則需要進行其他配置。可用的方法有兩種：

- **SCAN 方法：**修改 `buildforge.conf`。SCAN（單一用戶端存取名稱）是 Oracle 11g 中的一種機能。使用此機能時，叢集中的所有節點均可進行失效接手。至少應該有三個可用的節點，並且設為以循環方式進行失效接手。
- **「明確連線」字串方法：**修改 Oracle Instant Client 的 `tnsnames.ora` 檔，並且修改 `buildforge.conf`。在此方法中，您會指定代表叢集的服務以及每個所需的叢集節點。至少應該有三個可用的節點，並且設為以循環方式進行失效接手。必須設定 `LOAD_BALANCE` 及 `FAILOVER` 參數。

您可能需要 Oracle DBA 以提供服務名稱及要使用的節點清單。您可以配置 Oracle Instant Client 的 `tnsnames.ora` 檔，並在 `buildforge.conf` 中設定 JDBC 連線字串。

這兩種方法皆需要您指定代表整個叢集的 SERVICE_NAME。這與非 RAC 配置不同，在非 RAC 配置中，SERVICE_NAME 通常與 SID 相同。

程序

• SCAN 方法

1. 編輯 buildforge.conf。將 db_url 參數新增至 buildforge.conf，並使用 JDBC 連線字串作為參數。請注意，必須在 Oracle 中設定服務 (mysvc.mycorp.com)，且該服務必須具有多個節點。包含換行是為便於閱讀。請勿在 buildforge.conf 中換行。

```
db_hostname myoracledbhost.mycompany.com
db_url jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=myoracledbhost.mycompany.com)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=mysvc.mycorp.com)))
db_tcp_port 1521
db_database orclrac
db_username
db_password db_type odbc
db_provider oracle
```

2. 更新所有位置中的 buildforge.conf。

– Windows

```
<bfinstall>
<bfinstall>\Apache\tomcat\work\Catalina\localhost\jas\eclipse\plugins\com.ibm.jas-xxx.jar
```

– UNIX 或 Linux

```
<bfinstall>/Platform
bfinstall>/server/tomcat/work/Catalina/localhost/jas/eclipse/plugins/com.ibm.jas-xxx.jar
```

- 「明確連線」字串方法 在本範例中，將會使用下列名稱：

- 服務名稱：myorasvc.mycorp.com
- 節點名稱：clusnode-1vip、clusnode-2vip、clusnode-3vip、clusnode-4vip

1. 編輯 tsnames.ora。

```
myorasvc.mycorp.com=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-1vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-2vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-3vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-4vip)(PORT=1521))
  (LOAD_BALANCE=yes)(FAILOVER=ON)
  (CONNECT_DATA=(SERVER=DEDICATED)
  (SERVICE_NAME=myorasvc.mycorp.com)
  (failover_mode=(type=select)
  (method=basic)(retries=180)(delay=5)))
)
```

2. 編輯 buildforge.conf 將 db_url 參數設為 JDBC 連線字串。包含換行是為便於閱讀。請勿在 buildforge.conf 中換行。

```
db_url jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on)
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-1vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-2vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-3vip)(PORT=1521))
  (ADDRESS=(PROTOCOL=TCP)(HOST=clusnode-4vip)(PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=myorasvc.mycorp.com)))
```

3. 更新所有位置中的 buildforge.conf。

– Windows

```
<bfinstall>
<bfinstall>\Apache\tomcat\work\Catalina\localhost\jas\eclipse\plugins\com.ibm.jas-xxx.jar
```

- UNIX 或 Linux

```
<bfinstall>/Platform  
bfinstall>/server/tomcat/work/Catalina/localhost/jas/eclipse/plugins/com.ibm.jas-xxx.jar
```

安裝期間所需的 Oracle 資訊

關於這項作業

在安裝期間，系統會在**資料庫配置**畫面中要求您輸入下列資訊：

資料庫配置

- **資料庫主機**：用來安裝 Oracle 的主機。
- **資料庫埠**：針對 Oracle，Build Forge 會在這個欄位中填入預設埠 1521。如果您使用不同的埠，請準備輸入埠號。
- **資料庫名稱**：Build Forge 要使用的資料庫名稱。您已在之前的安裝步驟建立這個資料庫。
- **資料庫使用者名稱**：Build Forge 在存取資料庫時所使用的使用者名稱。您已在之前的安裝步驟建立這個使用者。
- **密碼**：資料庫使用者名稱的密碼。

測試資料庫配置

- **Oracle 用戶端程式庫的路徑** - Oracle 用戶端程式庫所在的目錄。

重要：在 UNIX 或 Linux 上安裝 Build Forge 時，這個目錄必須是包含 32 位元用戶端驅動程式庫的目錄。

- **ORACLE_HOME 環境變數** - 用來安裝 Oracle 的目錄。
- **tnsnames.ora 檔的路徑 (TNS_ADMIN)** - 包含 tnsnames.ora 檔的目錄。請確定已設定 tnsnames.ora 檔的完整存取權。
- **JDBC 驅動程式位置** - JDBC 驅動程式所在的目錄。Apache Tomcat 會使用這個驅動程式來存取資料庫。

畫面上會顯示下列資訊：

- **所需的驅動程式 JAR 檔** - 顯示所需的驅動程式 JAR 檔。
 - 若為 Oracle 10，檔案為 ojdbc14.jar
 - 若為 Oracle 11，檔案為 ojdbc5.jar
- **所需的 JDBC 驅動程式類別** - 顯示所需的 JDBC 驅動程式類別。針對 Oracle，這是 oracle.jdbc.driver.OracleDriver。

安全設定

在安裝期間，系統會詢問您關於要如何設定安全的問題。

- **金鑰儲存庫密碼**：您必須提供金鑰儲存庫的密碼。其會用來啟用安全登入（認證加密），並作為啟用 HTTPS/SSL 的起點。
- **憑證**：您可以選擇安裝個人憑證，或是匯入已有的憑證。
- **安全 HTTP**：系統會詢問您是否要安裝啟用 HTTPS/SSL 的 Apache 伺服器。其會使用您選擇的憑證。如果您需要使用預設值 443 以外的埠號，您需要在此時輸入埠號。

使用提供的個人憑證

所提供的憑證有下列屬性集：

- 主體 DN：“CN=*hostname*”，其中 *hostname* 是可用來執行安裝之主機의 完整名稱。
- 有效期限：10 年（表示 3650 天）。您可以變更這個值。一般的有效期限是一到兩年。超過這個有效期限會增加試圖猜測金鑰的安全攻擊漏洞。

您有機會修改提供的憑證。如果您修改憑證，則可以指定下列欄位。

- 一般名稱（必要）
- 地區
- 州/省（縣/市）
- 組織名稱
- 國家/地區名稱（必要）
- 地址

「一般名稱」和「國家/地區名稱」會連結到在建立憑證期間所指定的 X500Principal 類型的「主體 DN」。

系統會提示您輸入由安裝程式所建立，用於金鑰儲存庫的密碼。請記下這個密碼。完成安裝 HTTPS/SSL 需要有這個密碼。

重要：您之後可以變更密碼，但這是相當長的程序。請使用符合您本端複雜度需求的高保護性密碼。

使用您自己的憑證

如果您有憑證，您可以匯入它供系統中使用 SSL 的所有元件和連線來使用：

- 憑證必須位於要用來安裝 Build Forge 的主機上。請將憑證複製到暫存目錄。在安裝期間，系統會提示您輸入完整路徑。
- 您必須在安裝期間指定金鑰儲存庫密碼。
- 憑證必須是 PKCS12 金鑰儲存庫類型。如果憑證是另一種類型，您可以使用 OpenSSL `openssl` 公用程式或 JDK `keytool` 公用程式，將該副本轉換成 PKCS12。
- 系統會提示您輸入由安裝程式所建立，用於金鑰儲存庫的密碼。請記下這個密碼。完成安裝 HTTPS/SSL 需要有這個密碼。

重要：您之後可以變更密碼，但這是相當長的程序。請使用符合您本端複雜度需求的高保護性密碼。

第 7 章 安裝管理主控台

本節說明如何在 Windows、UNIX 和 Linux 平台上安裝「管理主控台」。請使用下列程序來進行 Installation Manager 所支援的任何安裝實務範例。

1. 依照第 33 頁的第 6 章,『前置安裝設定』的說明執行前置安裝設定。其包含下列作業。
 - 國際資料支援設定（必要）
 - 資料庫設定（必要）。這個步驟通常包含建立資料庫物件、安裝資料庫用戶端，以及收集在 Installation Manager 的安裝步驟期間所需的資訊。
 - 安全設定（選用，視需求而定）
2. 執行啟動程式以執行 Installation Manager。

註：啟動程式會在執行的主機上搜尋 Installation Manager。如果不存在，啟動程式會執行套裝的 Installation Manager，在主機上安裝 Installation Manager。接著會使用已安裝的 Installation Manager 來安裝 Build Forge。根據您的偏好，您可以手動安裝 Installation Manager 而非從「啟動程式」安裝。

3. 在 Installation Manager 中執行安裝步驟。
4. 執行必要的後置安裝檢查。

請參閱第 63 頁的第 8 章,『替代安裝方法』以取得下列實務範例的安裝程序：

- 針對下列必要的應用程式使用您自己的安裝架構，而非 Build Forge 所提供且安裝的應用程式：
 - Apache HTTP Server
 - PHP
 - Apache Tomcat
- 利用 IBM Installation Manager 無聲自動安裝產品元件
- 在 VMware 上安裝
- 在 SUSE Linux on System z 的管理主控台上安裝

利用啟動程式來啟動 Installation Manager

請使用啟動程式來啟動 Installation Manager 及安裝 Build Forge。

您可以用下列方式來執行啟動程式：

- 從產品 DVD 執行啟動程式。
- 從下載的檔案套件執行啟動程式。

啟動程式會偵測 Installation Manager 是否已安裝在您的主機上：

- 如果 Installation Manager 存在，啟動程式會加以啟動。在 Installation Manager 中，請選取要安裝 **Build Forge** 套件。

重要：您的 Installation Manager 版本必須符合最低版本需求。請參閱 第 21 頁的『Installation Manager 需求』。

- 如果 Installation Manager 不存在，啟動程式會使用套裝的 Installation Manager。在其中，請選取要安裝 **Installation Manager** 和 **Build Forge** 這兩個套件。套裝的 Installation Manager 會安裝 Installation Manager，接著啟動它來安裝 Build Forge。

如果是 Windows 主機，啟動程式也可以在本端主機上安裝代理程式。您不能利用啟動程式，在非 Windows 作業系統上安裝代理程式。如需代理程式安裝指示，請參閱第 131 頁的第 10 章，『安裝代理程式』。

從產品 DVD 執行啟動程式

請使用這些指令，從產品 DVD 執行啟動程式。

選擇適用於您作業系統的指示。

• UNIX 或 Linux

1. 在要用來安裝 Build Forge 的主機上插入第一片 DVD。
2. 裝載磁碟機。
3. 在磁碟機的根目錄中，執行 `launchpad.sh`。
4. 在 Installation Manager 中選取要安裝的套件。
 - 如果 IBM Installation Manager 已存在於主機中，啟動程式會加以啟動。

在第一個**安裝套件**頁面中，選取 **Build Forge** 套件，然後按下一步。

- 如果找不到 IBM Installation Manager，則會啟動套裝的 Installation Manager 以便安裝 Installation Manager，再利用它來安裝 Build Forge。

在第一個**安裝套件**頁面中，選取 **Installation Manager** 及 **Build Forge** 套件，然後按下一步。

• Windows

1. 在要用來安裝 Build Forge 的主機上插入第一片 DVD。
2. 如果已啟用自動執行，則會自動執行啟動程式。如果未啟用：請在磁碟機的根目錄中，執行 `launchpad.exe`
3. 在 Installation Manager 中選取要安裝的套件。
 - 如果 IBM Installation Manager 已存在於主機中，啟動程式會加以啟動。

在第一個**安裝套件**頁面中，選取 **Build Forge** 套件，然後按下一步。

- 如果找不到 IBM Installation Manager，則會啟動套裝的 Installation Manager 以便安裝 Installation Manager，再利用它來安裝 Build Forge。

在第一個**安裝套件**頁面中，選取 **Installation Manager** 及 **Build Forge** 套件，然後按下一步。

從下載的套件執行啟動程式

請使用這些指示來下載安裝套件並執行啟動程式。

1. 從 IBM Passport Advantage 將適用於您作業系統的安裝套件，下載至要用來安裝 Build Forge 之主機上的暫存目錄中。
2. 將安裝映像檔從下載的檔案解壓縮至本端目錄。檔案的內容會解壓縮至本端目錄中。
3. 從解壓縮檔案的目錄中執行啟動程式，如下所示：

- Windows：run launchpad.exe.
 - UNIX/Linux：執行 launchpad.sh。
4. 在 Installation Manager 中選取要安裝的套件。
- 如果 IBM Installation Manager 已存在於主機中，啟動程式會加以啟動。

在第一個**安裝套件**頁面中，選取 **Build Forge** 套件，然後按**下一步**。

- 如果找不到 IBM Installation Manager，則會啟動套裝的 Installation Manager 以便安裝 Installation Manager，再利用它來安裝 Build Forge。

在第一個**安裝套件**頁面中，選取 **Installation Manager** 及 **Build Forge** 套件，然後按**下一步**。

Installation Manager 中的安裝步驟

使用 IBM Installation Manager 在大部分平台上安裝產品元件。

開始之前

您必須啟動 Installation Manager 並選取要安裝的 **Build Forge** 套件，才能遵循以下指示。

關於這項作業

請遵循提示來安裝想要的套件：

程序

1. 安裝套件 - 選取 **Build Forge** 和**版本**勾選框。如果尚未安裝 Installation Manager，請選取 **Installation Manager** 勾選框。選擇之後，按**下一步**。
2. 授權合約頁面 - 指出您接受授權合約，然後按**下一步**。
3. 安裝套件：位置 - 套件群組 - 選擇安裝套件的安裝目錄，然後按**下一步**。預設位置列示如下。
 - Windows：C:\Program Files\IBM\Build Forge
 - Linux 和 AIX：/opt/buildforge
 - Solaris：/opt/buildforge_0
4. 安裝套件：特性 - 依預設，系統會安裝所有三個核心產品模組：Web 介面、程序引擎和服務層。按**下一步**。
5. 安裝套件：授權伺服器配置 - 選取主控台將會使用的**執行身分**使用者及授權類型。完成下列資訊，然後按**下一步**。
 - UNIX 和 Linux - 在**執行 Build Forge 的使用者身分**提示上，接受預設使用者 (root) 或指定不同的使用者。這個使用者會啟動 Build Forge 引擎及提供的 Apache Tomcat。對於 Installation Manager 的後續頁面上指定的資料庫程式庫及 JDBC Jar 檔，這個使用者必須具有讀取及執行許可權。
 - **Rational License Server 型**：輸入 Rational License Server 的主機名稱。
 - 輸入有效的授權伺服器主機名稱。

重要：如果您打算之後再提供主機名稱，請不要將這個欄位保留空白。請在此欄位中輸入任意字元或值。將這個欄位保留空白可能會造成產品不完整且無法使用。

安裝完成之後，已提供正確的主機名稱。如需相關指示，請參閱第 22 頁的『配置 Build Forge 的 Rational License Server』。

- 如果顯示的授權伺服器呈現灰色，表示 FLEXlm 授權用戶端已對該主機登錄授權伺服器。

安裝完成之後，請提供正確的主機名稱。如需相關指示，請參閱第 23 頁的『變更「管理主控台」的授權伺服器』。

- **檔案型：**瀏覽至下載授權檔的位置。

6. 資料庫配置 - 若要配置資料庫，請完成下列步驟：

- a. 若要測試配置，請配置 JAS 的資料庫連線，然後按一下靠近螢幕底端的**測試連線**。通過測試之後，請按**下一步**。
- b. 配置資料庫用戶端程式庫，然後按一下**測試連線**。通過測試之後，請按**下一步**。
- c. 根據用來安裝 Build Forge 的 OS 平台，以及要安裝的資料庫，您必須指定某些資訊。請參閱第 34 頁的『資料庫設定』中，關於下列項目的資料庫設定指示：
 - 對於 DB2，請參閱第 35 頁的『DB2 設定』。
 - 對於 Microsoft SQL Server，請參閱第 38 頁的『設定 Microsoft SQL Server』。請注意，所指定的 JDBC 驅動程式會視您選擇的版本（SQL Server 2005 或 SQL Server 2008）而定。
 - 若為 MySQL，請參閱第 43 頁的『MySQL 設定』。
 - 若為 Oracle，請參閱第 46 頁的『Oracle 設定』。您選擇的版本必須對應於已安裝的 Oracle Instant Client 版本（Oracle 10 或 Oracle 11）。

註：在 UNIX 和 Linux 平台上，如果您使用 Oracle、DB2 或 MySQL 資料庫，則必須安裝並使用 32 位元資料庫用戶端驅動程式。在 Installation Manager 的「資料庫配置」頁面，針對特定的資料庫類型，確定在「資料庫配置」頁面的 [DB2|Oracle|MySQL] 用戶端程式庫的路徑欄位中指定 32 位元版本的驅動程式庫。另外，若為 DB2，請載入 db2profile。

7. 安裝套件：應用程式與 Web 伺服器配置 - 完成四個頁面上所要求的資訊，並於完成每一個頁面之後按**下一步**。

a. Web 伺服器/PHP 配置

- **提供自己的 Web 伺服器？** 如果您要提供自己的 Web 伺服器，請選取**是**。預設值為**否**。
- **Web 伺服器應使用哪一個埠？** 如果您不想讓 Web 伺服器使用埠 80，請輸入不同的埠號。預設值為 80。
- **您要使用「安全 HTTP」嗎？** 如果您想使用「安全 HTTP」，請選取**是**。預設值為**否**。
- **PHP 的記憶體限制：**如果您不要使用預設值，請輸入 PHP 的記憶體限制。預設值為 256 MB。

b. 應用程式伺服器配置

- **提供自己的應用程式伺服器？** 若要將 Build Forge 配置成使用您已安裝的應用程式伺服器，請選取**是**。預設值為**否**。

註：只有在您將 Build Forge 配置成使用您已安裝的一個以上元件時，才選取**是**。正常情況下，Build Forge 會在安裝期間安裝這些元件。請參閱第 63 頁的『使用您自己的元件安裝』。

- **應用程式伺服器應該將哪一個埠用於 HTTP？** 輸入您要讓應用程式伺服器使用的埠。預設值為 **8080**。
- **應用程式伺服器應該將哪一個埠用於 SSL？** 輸入您要讓應用程式伺服器用於 SSL 的埠。預設值為 **8443**。
- **應用程式伺服器應該將哪一個埠用於 AJP？** 輸入您要讓應用程式伺服器用於 AJP 的埠。預設值為 **8009**。
- **應用程式伺服器應該將哪一個埠用於「關機」？** 輸入應用程式伺服器關機時所使用的埠。預設值為 **8005**。
- **應用程式伺服器的完整網域名稱** 輸入應用程式伺服器的完整網域名稱。例如，**WIN-TB4MVA22E7P.cn.ibm.com**。

c. **服務配置** 在「服務配置」畫面上完成下列欄位。

- **接聽的埠** - 選取這個勾選框。您必須選取一或兩個埠作為服務埠。預設值為 **3966**。
- **接聽的安全埠** - 選取這個勾選框。您必須選取一或兩個埠作為服務埠。預設值為 **49150**。

d. **安全配置** 在下列欄位中，輸入並驗證安裝程式為 Build Forge 建立的金鑰儲存庫密碼。

註：如果未顯示這些欄位，請向下捲動來找出它們。

- **金鑰儲存庫密碼**輸入包含憑證之金鑰儲存庫的密碼。
- **驗證密碼**重新輸入包含憑證之金鑰儲存庫的密碼。啓用預設安全登入（認證加密）必須有密碼。如果您打算將 Build Forge 配置成使用 HTTPS/SSL，也需要有密碼。

重要：您之後可以變更密碼，但這是相當長的程序。請使用符合您本端複雜度需求的高保護性密碼。

在**您要修改預設值或上傳自訂憑證嗎？**提示上，從下列選項中選取：

- 若要依現狀使用安裝程式建立的自簽憑證，請執行下列動作：
 - 1) 在**您要修改預設值或上傳自訂憑證嗎？**提示上，選取**否**。這時會顯示下列提示。
 - 2) **您有現有的安全憑證嗎？** 選取**否**。
- 若要使用安裝程式建立的自簽憑證，但修改其欄位，請執行下列動作：
 - 這時會顯示憑證的其他欄位。完成欄位，並按**下一步**。
 - 1) 在**您要修改預設值或上傳自訂憑證嗎？**提示上，選取**是**。
 - 2) 在**您有現有的安全憑證嗎？**提示上，選取**否**。
- 提供您自己的憑證的位置。該憑證必須位於主機上，其必須是 pkcs12 格式，且您必須提供其所在之金鑰儲存庫的現有密碼。請執行下列動作：

- 1) **金鑰儲存庫密碼**輸入包含憑證之金鑰儲存庫的密碼。
- 2) **驗證密碼**重新輸入包含憑證之金鑰儲存庫的密碼。
- 3) 指定金鑰儲存庫中已簽署的憑證，類型為 **pkcs12**。

註：瀏覽器通常會在存取具有自簽憑證的安全網站時，發出警告。一般而言，您可以選擇繼續進行，但可能需要確認異常狀況。

8. 安裝套件：主控台啟動選項 - 完成所要求的資訊。

- a. 指定資料庫埠。預設值為埠 **3306**。

- 在桌面上建立捷徑？預設為在桌面上建立快速鍵（對於 Windows）。

註：在 Linux 上，不會建立桌面捷徑。

- b. 在**資料庫名稱**欄位中，輸入資料庫名稱。
- c. 在**資料庫使用者名稱**欄位中，輸入登入資料庫時要使用的使用者名稱。
- d. 在**密碼**欄位中，輸入登入資料庫所需的密碼。
- e. 在**確認密碼**欄位中，重新輸入密碼。
- f. 在 **JDBC 驅動程式位置**欄位中，按一下**瀏覽**，以搜尋並選取 JDBC 驅動程式的位置。這時會顯示必要的驅動程式 JAR 檔及必要的 JDBC 驅動程式類別。
- g. 若要測試資料庫配置連線，請按一下**測試連線**。
- h. 通過測試之後，請按**下一步**。這時會顯示主控台啟動選項的下一個畫面。
- i. 在「資料庫用戶端程式庫配置資訊」畫面上，按一下**瀏覽**，以搜尋並選取 SQL 用戶端程式庫的路徑。
- j. 若要再次測試資料庫連線，請按一下**測試連線**。這時會顯示測試連線的結果。
- k. 按**下一步**。

9. 安裝套件：檢閱摘要 – 檢閱這個頁面的摘要資訊，以確認用來安裝 Build Forge 元件的位置，然後按一下**安裝**。

- a. 目標位置

- **套件群組名稱**：預設套件名稱為 `buildforge.console`。
- **安裝目錄**：預設安裝目錄為 `C:\Program Files\IBM\Build Forge`。
- **共用資源目錄**：預設的共用資源目錄。預設值如下：
 - Windows：`C:\Program Files\IBM\SDP70Shared`
 - UNIX 和 Linux：`/opt/IBM/SDP70Shared`

- b. 特性

- **Build Forge 特性**檢閱要安裝的特性或模組。例如，核心產品模組為：Web 介面、程序引擎和服務層。

- c. 環境

- 預設環境為英文。

- d. 儲存庫資訊

- **從下列位置擷取檔案**：使用此區段來檢閱並確認儲存庫位置。

10. 存取「管理主控台」。啟動瀏覽器。前往「管理主控台」的 URL：

- 一般格式：`http://<hostname>[:<portnumber>]`。埠號是您選用的，如果您使用 HTTP 預設值，即為連接埠 80。

- 本端：如果您所執行的瀏覽器與正在執行的「管理主控台」位於同一部主機上，請使用 `http://localhost`。

註：如果安裝之後無法立即登入，請等待一分鐘左右，然後再試一次。第一次啟動時，引擎 (bfengine) 必須載入資料庫綱目。

重要：安裝之後請不要立即停止 bfengine，如此會毀損資料庫綱目。如果資料庫綱目毀損，則您必須從資料庫中捨棄所有 Build Forge[®] 表格，並重新安裝 Build Forge。

11. 登入。以使用者名稱 **root** 和密碼 **root** 來登入。請立即變更 root 密碼。

後置安裝核對清單

本節說明在安裝 Rational[®] Build Forge[®] 系統之後應該執行的動作。

- 檢查 PATH 變數。
- 如果管理主控台必須透過 Proxy 伺服器來存取資料庫，請識別 PHP 要使用的 Proxy 伺服器。
- 設定 JVM 記憶體。
- 日誌訊息移轉。

檢查 Windows 上的 PATH 變數

Windows 上的 PATH 變數必須包含連至資料庫用戶端或驅動程式 DLL 的路徑。手動檢查以下資料庫的 PATH：

- DB2 - 包含 db2cli.dll 及 sqlar.dll 的目錄
- MySQL - 包含 libmysql.dll 的目錄
- Oracle - 包含 oci.dll 的目錄

如果是 Microsoft SQL Server，設定 ODBC 連線需注意這項需求。

識別 Proxy 伺服器

此為選用步驟。只有當管理主控台必須使用 Proxy 伺服器來存取其資料庫時，才需要執行此步驟。您必須配置 PHP 來使用 Proxy 伺服器。

- 編輯 php.ini 檔案，此檔案位於 *bfinstall*/Apache/php，例如 C:\Program Files\IBM\Build Forge\Apache\php。

新增下列項目：

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

設定 JVM 記憶體（「快速報告」的需求）

此為選用步驟；不過，如果您的版本包含「快速報告」（Standard Edition、Enterprise Edition、Enterprise Plus Edition），則需要執行此步驟。

- 將 JVM 的記憶體上限設為 1 MB 以上。執行報告至少需要 1 GB (1024 MB) 的資料堆大小。如果您在執行報告時收到記憶體不足的錯誤訊息（大型報告很可能發生），請提高此設定。如果提高此設定，您可能必須增加主機的記憶體。

日誌訊息移轉

Rational® Build Forge® 儲存工作訊息的方式與舊版不同。因此，如果 Rational® Build Forge® 安裝是從舊版升級而來，則在該工作的移轉完成之前，工作輸出頁面中的「訊息」直欄可能會暫時空白。由於這會發生在小型、低優先順序的批次中，因此移轉可能需要一些時間。如果任何工作有空白訊息，建議您等到工作的訊息移轉完成。

增加 Linux 的檔案控點數目

開始之前

重要事項：為了得到最佳結果，在使用 Rational 產品之前，請先增加 Rational Build Forge 可用的檔案控點數。系統管理者可能需要進行這項變更。

當您遵循下列步驟來增加 Linux 上的檔案描述子時，請特別注意。如果未遵循這些指示，可能會導致電腦無法正確啟動。為了得到最佳結果，請要求系統管理者執行這項程序。

如果要增加檔案描述子，請執行下列動作：

程序

1. 以 root 使用者身分登入。如果您沒有 root 存取權，您必須先取得該存取權才能繼續進行。
2. 切換至 etc 目錄。
3. 找出 initscript Shell Script。請以 Linux 文字編輯器開啓檔案或加以建立。

重要事項：請勿在電腦上留下空的 initscript 檔。如果您這樣做，下次開啓或重新啟動電腦時，電腦會無法啓動。

4. 在第一行將 ulimit 設為遠大於 1024（這是大部分 Linux 電腦上的預設值）的數字。

```
ulimit -n 4096
```

注意事項：將 ulimit 設得太高可能會影響系統面的效能。

5. 在第二行輸入 eval exec "\$4"。
6. 儲存並關閉 Shell Script。

結果

如需 ulimit 指令的相關資訊，請參閱 ulimit 的線上指令說明。

第 8 章 替代安裝方法

本節說明安裝「管理主控台」的替代方法。我們提供了下列實務範例的指示：

- 針對下列應用程式使用您自己的安裝架構，而非 Build Forge 所提供且安裝的應用程式：
 - Apache HTTP Server 和 PHP
 - Apache Tomcat
- 利用 IBM Installation Manager 無聲自動安裝產品元件
- 在 VMware 上安裝
- 在 SUSE Linux on System z 上安裝

使用您自己的元件安裝

如果您已經安裝以下元件，且要使用已有的元件而非 Rational® Build Forge® 所提供的元件，您可以使用本節來設定所需的技術。

Build Forge 會自動執行下列必要元件和技術的安裝與配置：

- Apache HTTP Server 和 PHP
- Apache Tomcat

重要：您使用的每一個元件的版本都必須與產品隨附的元件相同。如需相關資訊，請參閱第 21 頁的『硬體、軟體與資料庫需求』。

您必須將每一個元件的安裝配置為符合 Build Forge 需求，然後執行 Installation Manager 來安裝 Build Forge 元件。在安裝期間，您可以選擇使用您已設定的技術。

以下章節說明如何設定每一項技術以搭配 Build Forge 使用，接著說明如何安裝 Build Forge 來使用這些技術。

必備條件

您需要擁有下列項目，才能在 UNIX 或 Linux 上執行安裝作業：

- 網際網路存取權。如果您無法從正在安裝 Build Forge 的電腦存取網際網路，就必須從具有存取權的電腦下載檔案並傳輸到 Build Forge 電腦，才能完成所有安裝步驟。
- 有效且可在您的平台上運作的 C 編譯器（例如，Linux 上的 gcc 編譯器）。
- 採用編譯器製造商所建議的機能（例如：與 gcc 搭配使用 gnu-make）。
- root 專用權。
- 如果要使用 SSL，您必須編譯 Open SSL。

國際資料支援

您必須設定 Build Forge，才能在「管理主控台」中支援國際資料。

開始之前

- **Web 瀏覽器：**
 - 必須有語言集
 - 必須有用來顯示已安裝之資料的字型
- **代理程式**

Build Forge 建議在代理程式伺服器上使用 UTF-8 字集。

在 UNIX/Linux 上，請利用下列指令來檢查語言環境和字集：

```
locale
```

您應該會看到指定語言和字集的值。下列範例來自於 Solaris 系統，其語言為 US English、字集為 UTF-8：

```
LANG=en_US.UTF-8  
LC_CTYPE="en_US.UTF-8"
```

- **所有資料庫：**

國際資料支援通常是在建立資料庫之時指定；建立資料庫之後，便無法配置國際資料支援。

您想用來顯示資料的字型，必須已安裝在資料庫主機上。

Build Forge 需要使用國際資料（UTF-8 字集）。

- **DB2：**
 1. 設定字碼集與地區。範例：CREATE DATABASE USING CODESET UTF-8 TERRITORY US（或在「控制中心」中選取適當的字碼集和地區）。
 2. 在 1208 的管理主控台電腦上設定 DB2CODEPAGE 環境變數。

在 Windows 上，使用指令：

```
set DB2CODEPAGE=1208
```

在 UNIX 或 Linux 上，使用指令：

```
export DB2CODEPAGE=1208
```

如果現有資料庫中存在您需要移轉至 UTF-8 的資料，下列文件將非常有用：<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/t0024033.htm>

- **MySQL：**設定伺服器字集與對照。如果目前您的 MySQL 安裝並不支援國際資料，您可以從原始檔重新編譯，然後使用 `./configure -- with-charset=utf8 -- with-collation=utf8_bin`。若未配置此支援，就不會啟動 Build Forge 引擎。
- **Oracle：**在您安裝實例時，在實例上將字集設為 **UTF8 - Unicode 3.0**。在 Database Configuration Assistant 中，您可以在 Character Sets 標籤上的 Initialization Parameters 步驟中進行這項設定。

資料庫安裝與配置

您可以使用本節來安裝資料庫，以便將其與使用您自己的元件的 Build Forge「管理主控台」安裝搭配使用。

開始之前

在您安裝和配置其他技術與 Build Forge 之前，必須先以資料庫物件來安裝和配置資料庫。您必須執行下列動作：

一般而言，您必須完成下列作業：

- 識別您想要使用的資料庫系統，驗證資料庫受 Build Forge 支援，且資料庫主機與 Build Forge 主機之間有必要的網路連線。如果必須使用 Proxy 伺服器來存取資料庫，請取得 Proxy 伺服器名稱與路徑。
- 如果您想要使用國際資料，請驗證已將資料庫配置為使用 UTF-8 字集。
- 建立資料庫物件和許可權，一般如下所示：
 - 資料庫：範例中命名為 **build**，但您可以使用不同名稱。
 - 資料庫使用者：「管理主控台」會使用此使用者名稱來存取資料庫。範例中的使用者命名為 **build**，但您可以使用不同的使用者名稱。
 - 資料庫使用者密碼
 - 可供 **build** 使用者在 **build** 資料庫中建立表格空間的許可權。必須具備擁有者許可權才能建立、修改及刪除資料。

將提供每種資料庫類型所適用的特定指示。

Apache HTTP Server 安裝與配置

Installation Manager 會安裝及配置 Apache HTTP Server 作為 Build Forge 的 Web 伺服器。使用所提供的 Apache HTTP Server 是為 Build Forge 配置 Web 伺服器最快的方式。

除了使用標準配置，您也可以配置現有的 Apache HTTP Server 來取代 Build Forge 安裝且配置的 Apache HTTP Server。提供的指示假設您已經有在您的作業系統上設定及配置 Apache HTTP Server 的經驗。

如果要使用現有的 Apache HTTP Server，請依下列方式修改安裝架構：

1. 修改 Apache HTTP Server 配置檔 (httpd-vhosts.conf) 以指向 Build Forge 應用程式。
2. 安裝 PHP 並配置 Apache HTTP Server 所需的 PHP 模組、Build Forge 資料庫以及密碼加密（如果要使用這項安全特性）。
3. 為您的資料庫配置 Apache。

利用 Installation Manager 安裝 Build Forge

在 Installation Manager 的「應用程式與 Web 伺服器配置」頁面，在提供您自己的 Web 伺服器提示欄位中選取是。

必備軟體

- Apache HTTP Server 2.2.4
- PHP 5.2.4

編輯 Apache 伺服器配置檔

1. 在伺服器安裝架構的 extras 目錄中找出 Apache http-vhosts.conf 檔。

```
cd <apache-dir>/conf/extras/  
vi httpd-vhosts.conf
```

2. 編輯 Apache http-vhosts.conf 檔。如果要新增 Build Forge 的相關資訊到 httpd-vhosts.conf 中，請新增以下各行：

```
<VirtualHost *:80>  
    ServerAdmin build@yourdomain.com  
    DocumentRoot /opt/buildforge/webroot/public  
    ServerName ausbuild01.yourdomain.com  
    ServerAlias build.yourdomain.com mc.yourdomain.com  
    ErrorLog logs/ausbuild.error_log  
    CustomLog logs/ausbuild.access_log common  
</VirtualHost>
```

3. 修改 DocumentRoot 設定以指向 Build Forge Web 應用程式。在範例中，Build Forge 安裝目錄是 /opt/buildforge。
4. 將埠保留為 80，或將它變更為在本端執行 Apache HTTP Server 的埠。

```
<VirtualHost *:80>
```

重要：請勿使用埠 8080；它是 Apache Tomcat 的預設埠。

5. 視需要修改 http-vhosts.conf 中 Apache HTTP Server 的任何其他設定：
 - ServerAdmin：Build Forge 管理者的電子郵件位址
 - DocumentRoot：Build Forge 應用程式進入網頁的位置
 - ServerName：安裝 Build Forge 應用程式的伺服器
 - ServerAlias：Build Forge ServerName URL 的選用別名
 - ErrorLog：Build Forge 應用程式的 Apache 錯誤日誌
 - CustomLog：記錄對 Build Forge 應用程式之存取的 Apache 錯誤日誌

針對 Apache HTTP Server 安裝及配置 PHP

PHP 並未隨 Apache HTTP Server 安裝。您必須安裝 PHP 5.2.4 並配置它來指向 Apache HTTP Server 的 httpd-vhosts.conf。

為 Build Forge 資料庫安裝及配置 PHP

在 PHP 安裝期間，針對要用來作為 Build Forge 資料庫的資料庫類型選取並安裝 PHP 延伸。

（選用）配置 PHP OpenSSL 模組以支援密碼加密

如果要支援 SSL，Build Forge 會使用 PHP OpenSSL 模組。PHP 5.2.4 會提供這項支援；不需要任何其他配置。

如果要支援密碼加密，則需要一些其他的配置。需要有 PHP 5.2.4 才能支援這項配置。您必須找出 OpenSSL 延伸的修補程式檔，將其安裝在 OpenSSL 目錄中，再重新編譯 PHP，如下所示：

1. 在 misc 目錄（其位於 Build Forge 安裝目錄下）中找出 php_openssl.h 和 openssl.c 修補程式檔，例如：

Windows	C:\Program Files\Build Forge\misc
UNIX/Linux	/opt/buildforge/Platform/misc

2. 將修補程式檔複製到位於 Build Forge 安裝目錄下的 openssl 目錄。
3. 使用 `--with-openssl=<path_to_openssl>` 配置選項來編譯 PHP，其中 `<path_to_openssl>` 是 Build Forge 的 openssl 目錄。

為您的資料庫配置 Apache

根據您的資料庫，您需要新增特定的資訊到 `httpd.conf` 中。

DB2 的 Apache 配置

1. 將下面這一行加入 Apache 啟動 Script 的開頭（通常是 `/etc/init.d/httpd` 或 `/etc/init.d/apache2`，視發行套件而定）。

```
source /home/db2bf/sql1lib/db2profile
```

2. 將以下各行加入 `httpd.conf`：

```
PassEnv LD_LIBRARY_PATH
PassEnv CLASSPATH
PassEnv LIBPATH
PassEnv VWS PATH
```

MySQL 的 Apache 配置

不需要額外配置。

Oracle 的 Apache 配置

1. 將以下各行加入 `httpd.conf`：

```
PassEnv LD_LIBRARY_PATH
PassEnv NLS_LANG
PassEnv ORACLE_HOME
PassEnv ORA_NLS
PassEnv ORA_NLS32
PassEnv TNS_ADMIN
```

2. 將以下各行加入在開機時啟動 Apache 的 Script 中（通常是 `/etc/init.d/httpd` 或 `/etc/init.d/apache2`），並提供下列設定的值。

```
export LD_LIBRARY_PATH=<value>
export NLS_LANG=<value>
export ORACLE_HOME=<value>
export ORA_NLS=<value>
export ORA_NLS32=<value>
export TNS_ADMIN=<value>
```

啟動 Apache HTTP Server

在啟動 Build Forge 引擎及啟動「管理主控台」之前，請先啟動 Apache HTTP Server。

PHP 安裝與配置

您可以使用這項程序來設定 PHP 以搭配「管理主控台」使用。

開始之前

需求：

- 版本：PHP 必須是 5.2.4 或更新版本
- 資料庫驅動程式：已安裝用於 Build Forge 資料庫的 PHP 模組

關於這項作業

請遵循本節的指示來為 Apache HTTP Server 或其他 Web 伺服器配置 PHP。這些指示假設您已下載所需的 PHP 版本。

- 下載 PHP
- 安裝 PHP
- 配置 PHP
- 編輯 Apache 配置檔
- （選用）識別要用來存取資料庫的 Proxy 伺服器（只有當「管理主控台」主機會透過 Proxy 伺服器來存取資料庫時才需要）

安裝 PHP

關於這項作業

本節說明如何從原始檔編譯及安裝 PHP。如果您有現有的 PHP 安裝架構且不要重新編譯，您只需要決定是否已安裝適當的資料庫驅動程式。如果您需要安裝資料庫驅動程式，請參閱 PHP 文件以得知可使用的安裝方法。請依下列方式安裝要與 Build Forge 搭配使用之資料庫的資料庫驅動程式：

- DB2：ibm_db2 驅動程式
- MySQL：mysqli 驅動程式
- Oracle Instant Client：oci8 驅動程式

註：目前，完整的 Oracle 用戶端與 PHP oci8 不相容。請僅使用 Oracle Instant Client。

程序

1. 配置 PHP 以便在您剛才建立的工作目錄中進行安裝。

```
$ ./configure --prefix=/usr/local/php-5.2.4 --with-<database>=shared \
--with-apxs2 --with-ldap=shared --enable-mbstring --enable-shmop \
--with-xml --with-zlib=shared
```

取代 `--with-<database>`，如下所示：

- DB2：`--with-ibm_db2[=dir]`。如果未指定 `=dir`，則會使用預設值：`/home/db2inst1/sqllib`
- MySQL：`--with-mysqli[=file]`。選用的 `file` 參數是 `mysql_config` 的路徑名稱。
- Microsoft SQL Server：`--with-mssql[=dir]`
- Oracle：您必須安裝另一個 **Oracle Instant Client** 安裝架構，才能使用 **PHP oci8**。當您使用 Oracle Instant Client 來連接資料庫時，請使用 `--with-oci8=instantclient,lib`，其中 `lib` 是 Instant Client lib 目錄的路徑。

請注意程式碼區塊中的行接續字元 `\`。此步驟會指定將安裝 PHP 的位置，以及將要安裝哪些選項。依預設它會安裝在 `/usr/local` 中。範例顯示如何將它放在 `/usr/local/php-5.2.4` 中。此位置會在稍後範例中用到。

2. 編譯 PHP。

```
$ make
```

此步驟可在您的本端目錄中編譯執行檔。

3. 安裝 PHP（以 root 身分執行）。

```
# make install
```

此步驟必須由有權寫入 Apache 安裝目錄（本範例中為 `/usr/local/apache-2.2.4`）的使用者來執行。通常是以 root 身分來執行，您的本端管理設定可能有所不同。

配置 PHP

程序

1. 將延伸檔複製到延伸檔目錄中。資料庫的延伸檔必須從儲存庫中複製到作用中的延伸檔目錄。下例假設 PHP 安裝在 `/usr/local/php-5.2.4` 中。請注意，`<datestamp>` 是一串數字。

```
$ cd /usr/local/php-5.2.4/lib/php/extensions/no-debug-non-zts-<datestamp>/
$ cp <db-extensions> ..
```

`<db-extensions>` 檔必須對應於您的 Build Forge 資料庫，如下所示：

- DB2：ibm_db2.so
- MySQL：mysql.so 和 mysqli.so
- Oracle：oci8.so

2. 編輯 PHP 配置檔 `php.ini`。下例假設 PHP 安裝在 `/usr/local/php-5.2.4` 中：

```
$ cd /usr/local/php-5.2.4/lib/
$ vi php.ini
```

新增下列項目：

```
extension_dir=/usr/local/php-5.2.4/lib/php/extensions
upload_tmp_dir=<directory>
extension=<db-extension-so>
```

請使用 `<db-extensions-so>` 作為資料庫的檔名，如下所示：

- DB2：extension=ibm_db2.so
- MySQL：兩個項目 -
extension=mysql.so
extension=mysqli.so
- Oracle：extension=oci8.so

註：Apache Web 伺服器執行身分的使用者必須能夠寫入 `upload_tmp_dir` 所用的目錄。通常該使用者為 `nobody`，但可能因您的本端管理實務而有所不同。

編輯 Apache 配置檔

程序

編輯 Apache 配置檔。在 `httpd.conf` 中新增 PHP 的相關資訊。

```
cd <apache-dir>
vi httpd.conf
```

新增下列數行：

```
LoadModule php5_module modules/libphp5.so
AddHandler php5-script .php
AddType text/html .php
DirectoryIndex index.php
```

識別 Proxy 伺服器 關於這項作業

選用：只有當「管理主控台」必須使用 Proxy 伺服器來存取其資料庫時才需要執行此步驟。

程序

編輯 PHP 配置檔 `php.ini`。此檔案位於 `<php-install>/lib` 中，例如：`/usr/local/php-5.2.4`。

新增下列項目：

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

Apache Tomcat 安裝及配置

Installation Manager 會安裝及配置 Apache Tomcat 作為 Build Forge 的應用程式伺服器。使用所提供的 Apache Tomcat 應用程式伺服器是為 Build Forge 配置應用程式伺服器最快的方式。

除了使用標準配置，您也可以使用現有的 Apache Tomcat，而非 Build Forge 所提供的 Apache Tomcat。本節說明這項替代方案的必備軟體、前置安裝設定、安裝和後置安裝需求。提供的指示假設您已經有設定及配置 Apache Tomcat 的經驗。

軟體需求

- Apache Tomcat Server：
 - 5.5.28，適用於 Solaris 平台
 - 5.5.9，適用於所有其他平台
- J2SE 5：IBM 或 Sun
- Build Forge 資料庫的 JDBC 資料庫驅動程式：需要「Java 資料庫連線功能 (JDBC)」驅動程式來搭配 Apache Tomcat 使用。Sun 在以下網址提供了 JDBC 供應商清單：
<http://developers.sun.com/product/jdbc/drivers>。

安裝 JDBC 驅動程式的 Jar 檔

下載並解壓縮您資料庫所適用的 JDBC 驅動程式。

重要：JDBC 驅動程式下載檔可能含有許多檔案和子目錄。找到 JDBC 驅動程式的 Jar 檔，僅將 Jar 檔複製到 `$CATALINA_HOME/common/lib`。

`$CATALINA_HOME` 為 Tomcat 安裝根目錄，而必須設定成環境變數。請參閱 JDBC 驅動程式的安裝文件。

- DB2 - <http://www-306.ibm.com/software/data/db2/express/download.html>

按一下 **JDBC 和 SQLJ 的 DB2 驅動程式** 旁的下載鏈結；這需要登錄 IBM 帳戶。您也必須找到並安裝授權 `.jar`：`db2jcc_license_cu.jar`。

- MySQL - <http://www.mysql.com/products/connector/j/>

按一下 **MySQL Connector/J 5.0 或 5.1** 的鏈結。請選取對應於您的 MySQL 版本的 JDBC 驅動程式版本。

- Oracle - http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html

按一下您 Oracle 版本旁的下載鏈結；這需要登錄帳戶。

- Microsoft SQL Server- <http://msdn.microsoft.com/en-us/data/aa937724.aspx>

按一下下載 **SQL Server JDBC 驅動程式** 鏈結。

在 Installation Manager 中配置 Apache Tomcat Server

這些指示會識別透過 Installation Manager 配置 Apache Tomcat 所需的資訊。

1. 關閉 Apache Tomcat。

重要：在啟動 Installation Manager 之前，必須先停止 Apache Tomcat。

2. 啟動 Installation Manager。
3. 在「啟動」頁面中，按一下**安裝**。
4. 請遵循 Installation Manager 精靈中的指示來安裝產品。
5. 在「應用程式伺服器配置」頁面中，按一下**是**來配置您自己的應用程式伺服器。

核對清單：應用程式伺服器配置

✓	欄位	說明
	重新導向 URL	輸入應用程式伺服器的主機名稱和埠號。您必須指定 <code>jas</code> 作為環境定義路徑。例如： <code>http: https://<app_server_host>:<app_server_port>/jas</code> 。
	指定您要在其中安裝 Build Forge 服務外掛程式的目錄	指定應用程式伺服器主機的本端目錄。Installation Manager 會在這個目錄中安裝 Build Forge 服務層應用程式外掛程式延伸。執行應用程式伺服器的使用者必須具有這個目錄的讀取、寫入和執行權。

註：在舊版的 Build Forge 中，必須指定 WAR 部署目錄。現在，WAR 部署目錄會自動設為 `bfinstall/PrepForExternal`。

6. 透過 Installation Manager 完成安裝。

Apache Tomcat 的後置安裝配置

透過 Installation Manager 完成安裝之後，請完成下列後置安裝步驟。

1. 手動將 `buildforge.conf` 檔新增至 `jas.war` 檔。

`jas.war` 的位置視您的作業系統而定。UNIX 和 Linux 會將這個檔案放在 `/opt/buildforge/PrepForExternal` 中；Windows 則會將這個檔案放在 `C:\Program Files\IBM\Build Forge\PrepForExternal` 中。請參閱第 83 頁的『更新 `buildforge.conf` 檔』並完成相關的步驟。

2. 增加 Apache Tomcat Server 的 JVM 資料堆大小。

將 JVM 資料堆大小上限選項 `-Xmx` 設為 1024 M。

請在 catalina.bat 或 catalina.sh 中使用 CATALINA_OPTS 或 JAVA_OPTS 環境變數，來設定這個 JVM 選項。

3. 啓動 Build Forge 之前，請先啓動 Apache Tomcat：

```
$CATALINA_HOME/bin/catalina.sh start.
```

手動安裝 Installation Manager

如果您使用啓動程式來啓動產品安裝作業，即會自動安裝或更新 IBM Installation Manager。請參閱第 55 頁的『利用啓動程式來啓動 Installation Manager』。

具備 Installation Manager 使用經驗或是要設定無聲自動安裝的使用者，可以手動安裝 Installation Manager。請執行下列步驟：

1. 從 Passport Advantage 下載或使用產品 DVD，來取得產品安裝套件。
2. 找出適合您平台的 Installation Manager 檔案：
 - InstallerImage_linux
 - InstallerImage_solaris
 - InstallerImage_win32
3. 輸入下列其中一個指令來啓動安裝程式。
 - 如果要以具有管理權的使用者身分來執行安裝，請執行下列指令：
install
 - 如果要以不具有管理權的使用者身分來執行安裝，請執行下列指令：
userinst
4. 請遵循安裝指示來安裝 Installation Manager。

安裝之後，您可以使用 Installation Manager 或 Installation Manager 安裝程式來無聲自動安裝套件。

啓動 Installation Manager

在 Windows 或 UNIX/Linux 上啓動 Installation Manager。

開始之前

如果您使用啓動程式來啓動產品安裝，則會自動啓動 Installation Manager。如果您已安裝 Installation Manager，您可以用下列其中一種方式來啓動它：

- Windows：按一下開始 > 程式集 > IBM Installation Manager > IBM Installation Manager。
- 切換至 <IM-installdir>，然後執行 .IBMIM。

指定儲存庫 URL

IBM Installation Manager 在每一個產品套件中都使用一個內嵌 URL，以透過網際網路來連接儲存庫伺服器及搜尋最新的產品安裝套件。

開始之前

在 Installation Manager 中，您可以在「喜好設定」視窗的「儲存庫」頁面中設定儲存庫位置。您的組織可能需要重新導向儲存庫以使用企業內部網路網站。

註：在開始安裝程序之前，請務必先從管理者或 IBM 取得安裝套件儲存庫 URL。

如果要指定儲存庫，請完成下列步驟：

1. 啟動 IBM Installation Manager。
2. 在「啟動」頁面中，按一下**檔案 > 喜好設定**。
3. 在「喜好設定」視窗中，按一下**儲存庫**。這時會開啓「儲存庫」頁面，顯示可用的儲存庫、其位置，以及是否已連接。
4. 在「儲存庫」頁面中，按一下**新增儲存庫**。
5. 在「新增儲存庫」對話框中，輸入儲存庫位置的 URL，或利用**瀏覽**來尋找包含儲存庫的 .zip 或 JAR 檔、diskTag.inf 檔，或是已展開之儲存庫的 repository.config 檔；然後按一下**確定**。

這時會列出新的儲存庫位置。如果儲存庫未連接，「連線」直欄會顯示紅色的 x。

註：如果要搜尋更新的套件，請務必選取**在安裝與更新期間搜尋服務儲存庫**。依預設會選取這個選項。

6. 按一下**確定**以關閉「喜好設定」視窗。

執行無聲自動安裝產品元件

您可以在無聲自動安裝模式下執行 Installation Manager，來無聲自動安裝 Rational Build Forge 產品元件。在無聲自動模式下，使用者介面會無法使用；反之，會使用回應檔來輸入安裝產品套件所需的指令。

無聲自動安裝所需的作業如下所示：

1. 安裝 Installation Manager。
2. 建立回應檔。
3. 在無聲自動安裝模式下執行 Installation Manager。

註：在下列情況下，您不能使用無聲自動安裝：

- 您要在未安裝 X11 的 Linux 伺服器上安裝

如需 Installation Manager 和無聲自動安裝的相關資訊，請參閱「Installation Manager 資訊中心」：<http://www.ibm.com/software/awdtools/installmanager/support/index.html>。

以 Installation Manager 建立回應檔

您可以錄製使用 Installation Manager 來安裝產品套件時的動作，來建立回應檔。當您錄製回應檔時，您在 Installation Manager 使用者介面中所做的所有選擇，都會儲存在 XML 檔中。當您以無聲自動模式執行 Installation Manager 時，Installation Manager 會使用 XML 回應檔來完成安裝。

您可以建立回應檔並安裝產品，或是使用 `-skipInstall <agentDataLocation>` 引數來跳過產品安裝，並只建立回應檔。下列指示提供了這兩種選項的範例語法。

如果要建立回應檔以進行安裝，請執行下列動作：

1. 在指令行中，切換至其中安裝 Installation Manager 之目錄中的 eclipse 子目錄。例如：

Windows	C:\Program Files\IBM\Installation Manager\eclipse
UNIX/Linux	/opt/IBM/InstallationManager/eclipse

2. 在指令行中，使用下列其中一個指令來啟動 Installation Manager，將回應檔和（選擇性地）日誌檔換成您自己的檔名和位置。

請確定您輸入的檔案路徑存在；Installation Manager 不會建立回應檔和日誌檔的目錄。如果您使用 -skipInstall 選項，那麼 <agentDataLocation> 必須是可寫入的目錄。

- 錄製回應檔並安裝產品：

```
IBMIM -record <response file path and name> -log <log file path and name>
```

- 錄製回應檔但不安裝產品：

```
IBMIM -record <response file path and name> -log <log file path and name>
-skipInstall <agentDataLocation>
```

3. 遵循「安裝套件」精靈中的指示來進行安裝選擇。
4. 按一下**完成**，然後關閉 Installation Manager。

如此即會建立 XML 回應檔，並位於指令中指定的位置。

以無聲自動模式安裝及執行 Installation Manager

使用 Installation Manager，從指令行無聲自動安裝產品套件。

無聲自動安裝所需的作業如下所示：

如果要以無聲自動模式執行 Installation Manager，請從 eclipse 子目錄執行適用於您平台的指令：

Windows	<pre>IBMIMc.exe --launcher.ini silent-install.ini -input <response file path and name> -log <log file path and name></pre> <p>例如，IBMIMc.exe --launcher.ini silent-install.ini -input C:\mylog\responsefile.xml -log C:\mylog\silent_install_log.xml</p>
UNIX/Linux	<pre>IBMIM --launcher.ini silent-install.ini -input <response file path and name> -log <log file path and name></pre> <p>例如：IBMIM --launcher.ini silent-install.ini -input /root/mylog/responsefile.xml - log /root/mylog/silent_install_log.xml</p>

當 Installation Manager 在無聲自動安裝模式下執行時，其會讀取回應檔，並將日誌檔寫入您指定的目錄中。回應檔是必要的；日誌檔是選用的。這項執行的結果應該是狀態 0（表示成功）和非零的數字（表示失敗）。

下表說明用於無聲自動安裝指令的引數：

引數	說明
-vm	指定 Java 啟動程式。在無聲自動模式下，請一律在 Windows 上使用 java.exe，而在其他平台上使用 java。
-nosplash	隱藏啟動畫面。

引數	說明
--launcher.suppressErrors	隱藏 JVM 錯誤對話框。
-silent	以無聲自動模式來執行 Installation Manager 安裝程式。
-input	指定一個回應檔來作為 Installation Manager 的輸入。回應檔包含安裝程式或 Installation Manager 所執行的指令。
-log	(選用) 建立日誌檔來記錄無聲自動安裝的結果。此日誌檔是 XML 檔。

執行產品元件的無聲自動升級

您可以在無聲自動安裝模式下執行 Installation Manager，無聲自動升級 Rational Build Forge 產品元件。

無聲自動升級的必備項目如下所示：

- 現有的 Build Forge 主控台安裝架構必須是利用 Installation Manager 型無聲自動安裝來安裝的。
- 在含有 Build Forge 主控台主機的相同主機上，必須安裝了 Installation Manager。

無聲自動安裝所需的作業如下所示：

1. 建立升級回應檔。
2. 指定回應檔作為輸入，以無聲自動模式來執行 Installation Manager。

如需 Installation Manager 和無聲自動安裝的相關資訊，請參閱「Installation Manager 資訊中心」：<http://www.ibm.com/software/awdtools/installmanager/support/index.html>。

利用 Installation Manager 來已建立更新回應檔

在您利用 Installation Manager 來安裝產品套件時，將動作記錄下來，以建立一個回應檔。

當您錄製回應檔時，您在 Installation Manager 使用者介面中所做的所有選擇，都會儲存在 XML 檔中。

如果要建立更新安裝的回應檔：

1. 執行 Installation Manager。在「喜好設定」中，將更新安裝的產品儲存庫 URL 新增到 IM 儲存庫清單中，確定已選取它。
2. 結束 Installation Manager。
3. 在指令行中，切換至其中安裝 Installation Manager 之目錄中的 eclipse 子目錄。例如：

Windows	C:\Program Files\IBM\Installation Manager\eclipse
UNIX/Linux	/opt/IBM/InstallationManager/eclipse

4. 開始記錄安裝，但並不實際安裝產品。

輸入完整路徑，其中包含 response_file 和 log_file 的檔名。請確定您輸入的檔案路徑存在。Installation Manager 不會建立回應檔和日誌檔的目錄。agentDataLocation 必須是可寫入的目錄。

```
IBMIM -record response_file -log log_file -skipInstall agentDataLocation
```

5. Installation Manager 隨即啟動。在 Installation Manager 中，按一下**更新**，然後回應提示。
6. Installation Manager 完成時，按一下**完成**。
7. 結束 Installation Manager。

如此即會建立 XML 回應檔，並位於指令中指定的位置。

以無聲自動模式來執行更新安裝

使用 Installation Manager，從指令行無聲自動安裝產品套件。

如果要以無聲自動模式執行 Installation Manager，一般指令形式如下所示：

```
IBMIMc.exe --launcher.ini silent-install.ini -input response_file -log log_file
```

請使用 *response_file* 和 *log_file* 的完整路徑和檔名。

- Windows 範例

```
IBMIMc.exe --launcher.ini silent-install.ini -input C:\mylog\responsefile.xml -log C:\mylog\silent_install_log.xml
```

- UNIX 或 Linux 範例

```
IBMIM --launcher.ini silent-install.ini -input /root/mylog/responsefile.xml -log /root/mylog/silent_install_log.xml
```

當 Installation Manager 在無聲自動安裝模式下執行時，其會讀取回應檔，並將日誌檔寫入您指定的目錄中。回應檔是必要的；日誌檔是選用的。這項執行的結果應該是狀態 0（表示成功）和非零的數字（表示失敗）。

下表說明用於無聲自動安裝指令的引數：

引數	說明
-vm	指定 Java 啟動程式。在無聲自動模式下，請一律在 Windows 上使用 java.exe，而在其他平台上使用 java。
-nosplash	隱藏啟動畫面。
--launcher.suppressErrors	隱藏 JVM 錯誤對話框。
-silent	以無聲自動模式來執行 Installation Manager 安裝程式。
-input	指定一個回應檔來作為 Installation Manager 的輸入。回應檔包含安裝程式或 Installation Manager 所執行的指令。
-log	（選用）建立日誌檔來記錄無聲自動安裝的結果。此日誌檔是 XML 檔。

在 VMware 上安裝 Build Forge 系統

Build Forge 可以在 VMWare 上安裝並執行。

請使用以下準則：

- 將 Build Forge 使用的資料庫安裝在個別的主機上，最好是實體主機而非 VMWare 影像。
- 將 VMware Workstation 用來執行虛擬機器的記憶體至少設為 1 GB。在 VMWare Workstation 中，按一下**編輯 > 喜好設定 > 記憶體**來調整這個值。
- 您可能需要處理其他系統資源參數，以將 VMWare 上的效能最大化。

在 System z 上的 Linux 中安裝管理主控台

使用安裝媒體隨附的 `mc-<version>-<build>.tar.gz` tar 檔，在 z/Linux 上安裝及配置「管理主控台」。IBM Installation Manager 不適用於這項安裝。

z/Linux 主控台隨附 IBM HTTP server，而不是 Apache Web 伺服器。

在您安裝「管理主控台」後，請在 z/Linux 上安裝代理程式 rpm 套件 (`zlinux-bfagent-<version>.rpm`)，以便為 Build Forge® 設定 z/Linux 伺服器。如需安裝指示，請參閱第 133 頁的『將代理程式安裝在 UNIX 和 Linux 系統上』。

安裝期間所需的資訊

在安裝期間，系統會要求您輸入下列資訊。

1. 安裝目錄

- 提供您要 Build Forge® 安裝的位置的絕對路徑。

2. 資料庫資訊

- Build Forge® 使用的資料庫類型 (DB2、Oracle 或 MySQL)
- 資料庫伺服器主機名稱
- 資料庫埠號
- 所使用的資料庫名稱
- Build Forge® 用來連接資料庫的資料庫使用者名稱
- 資料庫使用者名稱的密碼
- 用戶端程式庫用來存取資料庫的位置
- JDBC 驅動程式 Jar 檔的位置

3. 應用程式伺服器資訊

- 要使用的應用程式伺服器 (已提供的 Tomcat 或您已設定的 WebSphere Application Server 安裝)

如果您選擇 WebSphere Application Server，則需提供更多資訊：

- Build Forge® 服務元件安裝在 WebSphere Application Server 上的位置。需要有網域、埠，以及 `jas` 的路徑。範例：
`http://mydomain.com:9080/jas`
- 外掛程式所用的目錄，可供讀寫的目錄。目前它與 Rational Team Concert 整合時方可使用。
- WAR 部署目錄：Build Forge® `jas.war` 檔的暫存位置。請在安裝後將其部署至應用程式伺服器。
- Java 執行檔 (`java.jar`) 的路徑
- 服務層級的暫時儲存體目錄。該目錄必須可供讀寫。它可用來供服務元件儲存暫時資訊。
- Build Forge® 服務的 HTTP 埠 (預設為 3966)
- Build Forge® 服務的 SSL 埠 (預設為 49150)

4. Web 伺服器資訊

- 要使用的 Web 伺服器 (已提供的 IBM HTTP Server 或您已安裝的 Web 伺服器)

如果您選擇已提供的 IHS 伺服器，則需提供更多 SSL 的相關資訊，包括是否使用 SSL、埠、所用的記憶體，以及是要使用現有的安全憑證還是另行建立。

執行安裝程式

1. 跳至您針對主控台從 tar 檔將套件解壓縮的目錄。
2. 執行下列指令：

```
./cmdline-install.sh
```

範例

下列標註清單顯示如何進行安裝程式步驟。這份清單是標註執行，不會反映實際安裝。可用的預設值顯示在方括弧 ([]) 中，例如：[y]。按 **Enter** 鍵來接受預設值。

```
Install directory [/opt/buildforge]
```

```
What database will you be using?
```

```
Enter the # of the database you will use
```

- 1) DB2
- 2) Oracle
- 3) MySQL

註：清單的其餘部分假設已選擇 Oracle。

```
What is your database hostname? 127.0.0.1
```

```
What is your database port number? [1521]
```

```
What is your database name? build
```

```
What is your database user name? build
```

```
What is your database user password?
```

```
Confirm your database user password?
```

```
Would you like this installation to create the Build Forge database schema? (y|n) [y]
```

這時請指定用戶端程式庫和資訊。視您的資料庫選擇而定，會提示您下列資訊。請使用絕對路徑。

- DB2

```
Where are your 32-bit DB2 client libraries (libdb2.so.1)?
```

註：程式庫必須是 32 位元程式庫。請輸入絕對路徑。

```
Where is your DB2 (DB2_HOME) installed?
```

- MySQL

```
Where are your MySQL client libraries (libmysqlclient.so)?
```

註：程式庫必須是 32 位元程式庫。請輸入絕對路徑。

- Oracle

```
Where are your Oracle client libraries (libclntsh.so)?
```

註：程式庫必須是 64 位元程式庫。請輸入絕對路徑。

```
Where is your Oracle instant client (ORACLE_HOME) installed?
```

註：這是即時用戶端程式庫根目錄的絕對路徑。

```
Where is your tns.names file located (TNS_ADMIN)?
```

註：這是包含 tns.names 檔的目錄。請輸入絕對路徑。

```
Please enter the directory with your database JDBC jar file?
```

註：請輸入 `ojdbc14.jar` 的絕對路徑。

Will you be using the supplied Tomcat app server? (y|n) n

註：清單的其餘部分假設您使用 **WebSphere Application Server** 作為應用程式伺服器。

Enter the full URL used to contact the services layer on your application server:

`http://mydomain.com:9080/jas`

Enter the directory to install the Build Forge Services plugins to:

註：這個目錄必須是應用程式伺服器主機上可讀寫的目錄。當 **Rational Team Concert** 與 **Build Forge**® 整合時，服務元件會使用這個目錄。

Specify the war deployment directory:

註：請在本端主機上指定一個目錄。安裝完成時，安裝程式會將 `jas.war` 檔放在此處。然後，您就可以將它部署至應用程式伺服器。

Enter the path to a jar executable (which should be included in any JDK):

註：請輸入資料庫驅動程式的 `.jar` 檔路徑。

Enter the temporary storage path for the Services Layer:

註：請輸入服務元件可以使用的目錄路徑。它必須可供讀寫。

What http port will the Build Forge services layer use? [3966]

What ssl port will the Build Forge services layer use? [49150]

Will you be using the supplied Apache web server? (y|n)[y]

註：清單的其餘部分假設您已輸入 `y`。

Would you like Apache to use SSL? (y|n)[n] Enter 'y' to configure Apache for SSL

註：清單的其餘部分假設您已輸入 `y`。

What ssl port will Apache use? [443]

Please enter a memory limit for PHP (in MB): [256]

Would you like to modify or specify a custom SSL certificate? (y|n) [n]

註：清單的其餘部分假設您已輸入 `y`，並打算建立自訂憑證。

Do you have an existing secure certificate? (y|n)[n]

A validity period is required for this cert please enter in [number][period] format

Examples: 10Y = 10 years, 6M = 6 months, 350D = 350 daysEnter the validity period for this cert:

Enter the common name for the certificate (usually the name of the server) [linux142.rtp.raleigh.ibm.com]:

Please enter your Locality/City:

Please enter your State/Province:

Please enter your Organization Name:

Please enter your Organization Unit:

Please enter your Country from the list below:

- France
- Taiwan
- Italy
- Germany
- Korea
- United States
- China
- Brazil
- Spain

Japan
Please enter your Street Address:
A keystore password is required, and must be at least 6 characters long
Please enter a keystore password

如果您使用 IBM HTTP Server 而非 Apache 作為 Web 伺服器，則請參閱第 513 頁的『使用 IBM HTTP Server 來取代 Apache HTTP Server』，以取得設定及啓用 SSL 的相關資訊。

啓動主控台

1. 啓動「管理主控台」：
`<bfinstall>/rc/buildforge start`
2. 驗證服務元件 (Apache Tomcat Server) 已啓動；開啓 `catalina.out`，並驗證已記載啓動訊息。
`<bfinstall>/server/tomcat/logs/catalina.out`
3. 開啓 Web 瀏覽器，並輸入完整的 z/Linux 主機名稱。例如：`http://myhost.mycompany.com`。

此時會啓動「管理主控台」，並顯示登入提示。

4. 以 **root/root** 身分登入。

安裝授權檔

z/Linux 的授權檔位於 `<bfinstall>` 目錄中。授權檔名稱爲 `IRBF_license`。安裝後它會包含一則文字訊息，指示您從 Passport Advantage 下載實際的授權檔。

下載授權檔並將它放置在 `<bfinstall>` 目錄中之後，請配置 Build Forge® 來使用它：

1. 啓動「管理主控台」。
2. 以 **root/root** 身分登入。
3. 選取**管理>系統**。
4. 找出「授權伺服器」設定，並將其值設爲授權檔的完整路徑。

例如：`<bfinstall>/IRBF_license`。

針對「管理主控台」啓用 SSL

您可以啓用 SSL，以便對在下列 Build Forge® 元件之間傳送的資料加密：

- Web 瀏覽器用戶端和 Apache HTTP Server
- Apache Tomcat Server 和 Apache HTTP Server

如果您在提示時回答**是**，安裝程式會執行啓用 SSL 所需的部分工作。若要啓用 SSL，請完成下列作業：

1. 檢閱「個人憑證」和「金鑰儲存庫」。
2. 配置 IBM HTTP Server 以使用 SSL。
3. 在「管理主控台」使用者介面中啓用 SSL。
4. 啓用 SSL 除錯。

註：在這個版本中，不支援 z/Linux 上的 Build Forge® 執行密碼加密和「單一登入 (SSO)」等其他安全特性。

檢閱個人憑證和金鑰儲存庫

安裝程式所建立的金鑰儲存庫如下所示：

金鑰儲存庫	說明
buildForgeKeyStore.p12	包含受密碼保護的 keyEntry（內含公開/私密金鑰配對的個人憑證）。
buildForgeTrustStore.p12	包含一個受密碼保護的 trustedCertEntry（僅含公開金鑰的憑證）。
buildForgeKey.pem	包含一個受密碼保護的私密金鑰。
buildForgeCert.pem	包含一個無密碼保護的憑證，內含對應至 buildForgeKey.pem 中之私密金鑰的公開金鑰。
buildForgeCA.pem	最初含有和 buildForgeCert.pem 相同的資訊；會新增其他的同層級憑證，以建立信任。
buildForgeKeyForApache.pem	需要這個金鑰儲存庫，才能針對 Apache HTTP Server 啟用 SSL。與 buildForgeKey.pem 不同，它沒有密碼保護，啟動 Apache HTTP Server 時系統不會提示您輸入密碼。

如需轉換現有 PEM 憑證和管理憑證的相關資訊，請參閱第 111 頁的『管理憑證』。

配置 IBM HTTP Server 以使用 SSL

請參閱第 513 頁的『使用 IBM HTTP Server 來取代 Apache HTTP Server』中的配置 IHS 的 SSL。您必須變更金鑰儲存庫格式並新增 httpd.conf 的項目。

在「管理主控台」使用者介面中啟用 SSL

使用「管理主控台」使用者介面設定，在「管理主控台」中啟用 SSL，並更新 Build Forge® 資料庫。您可以透過**管理 > 安全**來存取那些設定。然後，檢查 bfclient.conf 配置檔中已更新必要的內容值。

1. 啟動 Build Forge®。
2. 登入使用者介面。
3. 移至**管理 > 安全**。
4. 將 **SSL 已啟用**變更爲**是**。
5. 按一下**儲存**。
6. 按一下**更新主要 BFClient.conf**。

啟用 SSL 除錯

若要在「管理主控台」中針對 SSL 問題進行除錯，請完成下列步驟來記載 SSL 所需的其他資訊。

1. 在引擎中啟用除錯。在您啟動 Build Forge® 引擎之前，請設定下列環境變數：

```
export BFDEBUG_SECURITY=1
```

 - a. 重新啟動 Build Forge® 引擎。
 - b. 重新啟動 IHS。重新啟動 IHS 後，PHP 就可以使用這個除錯參數。
2. 在 Tomcat 中啟用除錯。在 `<bfinstall>/server/tomcat/common/classes/logging.properties` 中進行下列變更：
 - a. 新增下列字行：

```
com.buildforge.level = ALL
```

- b. 在 handlers 區段中，將其他所有層次從 FINE 變更爲 ALL。
- 重新啓動 Tomcat，讓這些變更生效。

第 9 章 在「管理主控台」中配置其他特性

本節說明配置 Build Forge[®] 以啓用其他特性或提供預設配置的替代方案的方式。

Build Forge 配置檔 (buildforge.conf)

buildforge.conf 檔是 Build Forge 產品配置檔。其包含供不同 Build Forge 元件用來啓動及與 Build Forge 資料庫進行通訊的配置設定。

buildforge.conf 檔儲存在兩個位置；如果您在安裝之後需要修改它，您必須更新這兩個位置中的檔案。請參閱『更新 buildforge.conf 檔』。

- 在應用程式伺服器所使用的 jas.war 檔中，也稱為服務層。
- 在安裝根目錄中。下表列出產品的預設或標準安裝目錄：

Windows	C:\Program Files\IBM\Build Forge
UNIX/Linux	/opt/buildforge/Platform

更新 buildforge.conf 檔

buildforge.conf 檔位於兩個位置。如果您在安裝之後需要更新它，則必須在這兩個位置中進行更新。

如果將 Build Forge[®] 資料庫移至不同的主機，您可能需要編輯 buildforge.conf 檔來更新資料庫主機。編輯這個檔案另一個常見的原因是要更新資料庫密碼；其必須定期變更以符合網路安全原則。

使用下列程序來更新 buildforge.conf 檔，然後以更新後的 buildforge.conf 副本來重建 jas.war 檔。

1. 停止 Build Forge[®] 引擎。
2. 在 Build Forge[®] 安裝目錄中找出 buildforge.conf 檔。

Windows	<bfinstall>
UNIX 和 Linux	<bfinstall>/Platform

3. 使用文字編輯器來開啓檔案並修改配置設定，然後儲存檔案。

註：您需要有 root 使用者或「管理者」專用權，才能編輯這個檔案。

4. 導覽至包含 jas.war 檔的目錄，例如：

Apache Tomcat Server	<bfinstall>/Apache/tomcat/webapps \$CATALINA_HOME/webapps
----------------------	--

5. 使用您偏好的保存管理程式工具，開啓 jas.war 檔，然後在 /WEB-INF/eclipse/plugins/ 下找出 com.ibm.jas-xxx.jar
6. 更新 com.ibm.jas-xxx.jar 中的 buildforge.conf 檔並儲存。

註：確定 jas.war 檔是設為自動更新。

- 在 `$CATALINA_HOME/webapps` 和 `$CATALINA_HOME/work/Catalina/localhost` 下，使用下列範例來刪除 jas 目錄：

Windows	<code>rmdir jas /s /q</code>
UNIX/Linux	<code>rm -rf jas</code>

- 重新啟動 Build Forge® 引擎。

即會重新部署 jas 檔，並以更新的 buildforge.conf 檔來自動重建 jas 目錄。

Buildforge.conf 參照

buildforge.conf 檔會儲存如何執行 Build Forge「管理主控台」的設定。

這個檔案位於安裝目錄中。其會由安裝程式自動建置。如果您需要進行編輯，該檔案可以儲存成 ASCII 文字檔或 XML 檔。其語法如下所示：

- 以一行輸入關鍵字和其值。
- 以空格分隔關鍵字和值（沒有等號）。

關鍵字	值
bf_file_storage	暫存 Build Forge 檔案所在的目錄。範例： C:\Program Files\IBM\BuildForge\temp
bf_plugin_dir	連接至「管理主控台」的 IDE 外掛程式所在的目錄。
birt_home	Eclipse 報告工具 (BIRT) 的檔案位置。
db_database	您建立供主控台使用的資料庫名稱。
db_hostname	執行資料庫之電腦的主機名稱/IP 位址。當您輸入 db_hostname 的值時，請使用實際的名稱或 IP 位址。請勿使用預設值 localhost。
db_password	您建立給資料庫使用者名稱的密碼。
db_provider	您決定用來安裝 Build Forge 的資料庫。請勿編輯這個值。
db_schema	資料庫的綱目名稱（通常與 db_username 相同，但您可以選擇另一個綱目名稱）。
db_tcp_port	您要使用的資料庫連線埠。
db_type	用來安裝 Build Forge 的資料庫類型。預設值是 odbc。請勿編輯這個值。
db_username	資料庫的使用者名稱。這會在執行安裝程式之前設定。 如果是 DB2 和 DB2 Express®，請建立 Windows 中（而非 DB2 中）的使用者。 如果是所有其他資料庫類型，請建立資料庫使用者名稱。 請參閱第 34 頁的『資料庫設定』。 所有主控台類型都需要這個值。
services_hostname	執行 Build Forge 服務層之電腦的主機名稱/IP 位址。這是服務配置中的完整網域名稱 (FQDN)。
services_ssl_port	用來安全連接 Build Forge 服務的 SSL 埠。
services_tcp_port	如果未指定 SSL，用來連接 Build Forge 服務的 TCP 埠。

關鍵字	值
services_url	指定用於服務層之埠的 URL。範例： services_url http://mybfhost.com:8080

配置「管理主控台」以使用替代埠

您可以在非預設埠 80 的埠上執行「管理主控台」。

關於這項作業

您可以用兩種方式，將「管理主控台」配置為在替代埠上執行：

- 在安裝期間，將埠設為所需的值。
- 如果已安裝「管理主控台」，請完成下列步驟：
 1. 啟動主控台並以 root 身分或是具有管理專用權的使用者名稱登入。
 2. 選取**管理 > 系統**，然後將「主控台埠」系統配置設定變更為用來執行「管理主控台」的埠。
 3. 變更 httpd.conf 中的兩項設定（如果是 Windows 安裝，此檔案位於 *<bfinstall>/Apache/conf/* 中；如果是 *nix 安裝，則位於 *<bfinstall>/server/apache/conf/* 中）。比方說，如果 myHost 是本端電腦且您要使用埠 81，請指定以下設定：


```
Listen 81
ServerName myHost:81
```
 4. 停止再重新啟動引擎。
 - Windows：按一下**開始 > 程式集 > IBM Rational Build Forge 管理主控台 > 停止引擎服務**，然後按一下**啟動引擎服務**。

如果 Build Forge 正在前景執行中，跳至它執行中的 Windows 主控台，然後輸入 Ctrl-C。

- UNIX 或 Linux：使用針對 rc 檔而提供的 Script。


```
$ /opt/buildforge/rc/buildforge start
$ /opt/buildforge/rc/buildforge stop
```

您也可以使用手動指令。

- a. 如果要停止，請尋找該處理程序 ID 並結束處理程序。


```
$ ps aux | grep buildforge
$ kill ${<PID>}
```
- b. 若要啟動，請使用下列指令，其中 *<bfinstall>* 是安裝目錄的路徑：


```
<bfinstall>/Platform/buildforge
```

配置備用功能

您可以設定多部電腦來執行 Build Forge，且所有電腦都與相同的 Build Forge 資料庫進行通訊。這項設定稱為「備用功能」。

關於備用功能

備用功能有助於平衡工作處理及增加其中一部機器安裝失敗時的可用性。

重要：備用功能並不提供失效接手功能或其他高可用性功能。其只是會增加工作處理的產能。如果其中一個備用安裝架構失敗，該安裝架構所管理的所有執行中工作都會遺失，但是其餘的安裝架構會繼續處理其執行中的工作並接受新的工作。

當使用者啟動工作時，即會在資料庫中產生該工作的項目。程序引擎會輪詢資料庫，以查看是否有新的工作。如果有多個程序引擎，自然會進行負載平衡，因為每一個引擎都會在其不繁忙的週期期間獨立輪詢。

設定備援功能時，您可以執行管理主控台的一般安裝，然後在其他主機上安裝管理主控台。所有安裝架構都會配置成存取相同的 **Build Forge** 資料庫。

重要：每一個安裝架構都必須安裝在自己的主機上。您不能將多個管理主控台安裝在同一部主機上。

安裝冗餘系統

關於這項作業

下列指示假設您已設定資料庫，並已安裝管理主控台的第一個安裝架構來使用它。若要在其他主機上建立其他安裝，請執行下列步驟。

程序

1. 根據適用於您資料庫的指示，在主機上執行前置安裝設定。根據您的資料庫，設定可能需要在主機上安裝資料庫用戶端，並執行其他配置。請參閱第 33 頁的第 6 章，『前置安裝設定』。
2. 請確定資料庫伺服器已配置為接受外部連線 (TCP)。
3. 遵循安裝指示。從第 55 頁的第 7 章，『安裝管理主控台』開始，執行下列動作：
 - 安裝 **Installation Manager**（如有需要）
 - 啟動 **Installation Manager**
 - 執行安裝

下列步驟只說明需要特定項目的畫面。

4. 在**安裝套件：特性**畫面上，確定已選取所有特性。
5. 在**資料庫配置**畫面上，執行下列動作：
 - 提供資料庫名稱和綱目名稱。這些名稱必須與第一個主控台上指定的名稱相同。
 - 提供資料庫使用者名稱和密碼。請使用第一個主控台上指定的相同名稱。
 - 在**您要在安裝期間在這個資料庫中移入資料嗎？**，選取否。
 - 按一下**執行測試**。當測試通過時，按**下一步**繼續進行。

註：當選取否時，只會執行一種測試，即檢查 **JDBC** 驅動程式的正確路徑。

6. 在**應用程式伺服器配置**畫面上，輸入您在第一個主控台上輸入的相同應用程式伺服器完整網域名稱。

註：所有埠號必須與第一個主控台上指定的埠號相同。

7. 在**主控台啟動選項**畫面上，確定已選取**不啟動 Build Forge** 選項。它應該會呈現灰色，因此不能變更。在安裝之後手動啟動主控台。

8. 執行任何必要的後置安裝配置。請參閱第 173 頁的第 11 章,『後置安裝作業』。
如有需要,請製作資料庫的型錄,使得資料庫用戶端能夠與其連接。

使用備用功能

設定備用功能之後,您可以依下列方式使用它:

- 將使用者指向第一個安裝架構的 URL。
- 如果您不希望使用者存取,您可以停止其他安裝上的 Apache 伺服器。
- 如果您希望增加產能以處理 HTTP 要求,請在所有安裝上執行 Apache。請安裝負載平衡程式,於安裝架構之間分散要求。

註: 備用功能一次僅支援一種類型的工作控制器。您無法同時執行 Perl 與「主要工作控制器 (MJC)」工作。若要使用不同的工作控制器來執行,您必須在「系統配置」頁面上切換工作控制器類型。

啓用 IPv6 網路支援

您可以將「管理主控台」配置為使用 IPv6 網路,以及混合的 IPv6/IPv4 網路。

關於這項作業

配置「管理主控台」以使用 IPv6,需要完成下列步驟:

1. 變更 httpd.conf 中的伺服器項目。
2. 設定所需的 FLEXlm 授權用戶端,以搭配 IPv6 網路使用。

檢閱搭配 IPv6 使用 Build Forge 時的需求和配置步驟。請參閱 第 25 頁的『IPv6 支援的網路需求』。

修改 httpd.conf

關於這項作業

IPv6 支援需要正確地配置電腦和網路以支援 IPv6。網路配置問題會使得 Build Forge 系統內指定的主機名稱和位址無法正確解析。

您必須手動配置 Build Forge 以使用 IPv6。 要執行這項作業,請修改主要 Apache 配置檔 httpd.conf 中的項目。:

1. 導覽至 httpd.conf 檔 (buildforge/server/apache/conf/httpd.conf)。
2. 新增參照完整網域名稱 (FQDN) 的 ServerName。例如: ServerName qlnx500-v6.ipv6.lexma.ibm.com
3. 依下列方式修改接聽指引 0.0.0.0:80。
 - Windows : [::]:80
 - UNIX 或 Linux : 80

安全特性

本節說明在 Build Forge[®] 中啟用安全特性的方式：

- 安全登入，依預設於安裝期間啟用
- 單一登入 (SSO)
- 啟用 HTTPS 和 SSL
- 啟用密碼加密
- 使用 `bfcclient.conf` 檔作為安全配置

這些特性可透過管理主控台其中的選項組合（位於 **管理 > 安全**）以及手動設定配置檔來啟用。本節包含 `bfcclient.conf`（這是用來啟用安全特性的配置檔）的參考區段。

註：本節不適用於在 z/Linux 上執行 Build Forge 的使用者。請參閱第 77 頁的『在 System z 上的 Linux 中安裝管理主控台』，以取得 z/Linux 上的 Build Forge 中可用之安全特性的相關資訊。

實作單一登入

Build Forge 提供了單一登入架構。

單一登入是一種鑑別方法，其可讓使用者存取應用程式時，不需每次都輸入使用者名稱和密碼。Build Forge 提供了一個架構可搭配協力廠商 HTTP 攔截程式一起使用，來實作單一登入。

關於單一登入架構

Build Forge[®] SSO 架構提供了整合市場上許多 SSO 解決方案的功能。SSO 架構是攔截程式型，意指其會截取 HTTP 要求並提供處理方法。您可以撰寫自訂攔截程式來接收及驗證 HTTP 要求中的安全構件。特別是，攔截程式可以在 HTTP 回應中設定記號，然後在後續的要求中尋找這些記號。

Build Forge[®] 提供了兩個 SSO 解決方案：

- SPNEGO（簡式且受保護的協議通訊協定）的攔截程式。請參閱第 92 頁的『在 Active Directory 網域中使用 SPNEGO 來實作單一登入』。
- 與 WebSphere SSO 整合的攔截程式。請參閱第 97 頁的『使用自訂攔截程式來與 WebSphere Application Server 安全整合』。

SSO 架構方法： SSO 攔截程式是一種 Java 類別，其實作 Build Forge[®] SSO 架構所使用的介面：

```
com.buildforge.services.server.sso.ISSOInterceptor
```

其位於服務層元件中：

```
<bfinstall>/Apache/tomcat/webapps/jas/WEB-INF/eclipse/plugins/com.ibm.jas-1.0.jar
```

該介面提供了下列方法。

initInterceptor

於載入攔截程式時呼叫。配置內容的對映會傳給 `initInterceptor()` 方法。配置內容是在 Build Forge 主控台的**管理 > 安全 > SSO** 中建立。

isTargetInterceptor

檢閱入埠要求中的屬性，以判斷這個攔截程式是否需要處理這些屬性。若是如

此，攔截程式會負責以 `authenticateRequest()` 方法鑑別要求。否則會略過這個攔截程式。攔截程式選項會假設多個攔截程式已配置且在執行中。其會依序處理。

authenticateRequest

利用要求中的資料來鑑別要求。其會使用回應屬性將資料傳回用戶端。

logoutRequest

在處理要求之後，清除任何與使用者相關的安全資訊。

攔截程式配置與排序： 攔截程式配置定義在**管理 > 安全 > SSO** 中。Build Forge® 隨附下列配置：

- 表單 SSO 攔截程式 - 依預設為作用中，其會實作簡單的登入表單。
- SPNEGO SSO 攔截程式 - 依預設為非作用中，其會實作 SPNEGO 來執行鑑別。

實作攔截程式類別並將它放在 Build Forge® Apache Tomcat 應用程式伺服器之後，您就可以在這裡配置新的 SSO 配置。該類別是 SSO 配置的其中一個內容。

這份清單的順序會決定參考攔截程式以便處理要求的順序。您可以配置多個攔截程式來處理要求。在登入期間，每一個攔截程式都會依序參考。處理要求的攔截程式會是第一個作用中的攔截程式，其屬性適用於要求中的屬性。只有一個攔截程式會處理要求。其一律是第一個對 `isTargetInterceptor` 回應 `true` 的攔截程式。

註：若要在錯誤時提供撤回，請將「表單 SSO 攔截程式」維持為作用中。在清單中，將自訂攔截程式放在它的前面。

新增自訂 SSO 攔截程式： 若要在 Build Forge® 中建立自訂攔截程式，請完成下列步驟：

1. 建立自訂 Java 類別。

該類別必須實作 `ISSOInterceptor` 介面。

2. 將自訂類別部署至服務層元件 WAR 檔。
 - a. 建立含有經過編譯的自訂 SSO 攔截程式類別的 JAR 檔。
 - b. 將 JAR 檔合併至下列位置中的 Build Forge® 服務層 JAR 檔：`<bfinstall>/server/tomcat/work/Catalina/localhost/jas/eclipse/plugins/com.ibm.jas-xxx.jar`。
 - c. 重新啟動 Build Forge®。
3. 定義環境。這個環境可以當成內容物件傳給 `initInterceptor()` 方法。
 - a. 在「管理主控台」中，移至**環境**。
 - b. 按一下**新增環境**。
 - c. 定義 SSO 攔截程式所需的所有內容以執行起始設定。
4. 將 SSO 攔截程式新增至 Build Forge®：
 - a. 在「管理主控台」中，移至**管理 > 安全 > SSO**。
 - b. 按一下**新增 SSO 配置**，然後輸入內容：
 - **名稱** - 輸入 SSO 配置的名稱。
 - **作用中** - 設為**是**。作用中的配置全都會在鑑別要求期間加以存取。存取時依它們出現在這個畫面上的順序進行。

- **Java 類別** - 輸入類別的完整套件名稱。一個給定的類別只能指派給一個 SSO 攔截程式。
- **環境** - 如果您已定義一個要使用這個 SSO 攔截程式的環境，請選取該環境。

c. 按一下**儲存**。

您的 SSO 攔截程式現在會顯示在清單中。

5. 按一下「SSO 攔截程式」左側的圖示然後選取**移至頂端**，來排序 SSO 配置。

在要求期間，將依出現在這個畫面上的順序來存取作用中的 SSO 配置。因為依預設為作用中，且存取時一律傳回 true，所以您的配置必須放在**表單 SSO** 配置的前面。依預設 **SPNEGO SSO** 配置是在非作用中。

範例 authenticateRequest 實作： 下列範例取自 WebSphere SSO 攔截程式，用來整合 WebSphere Application Server 安全與 Build Forge®。

該攔截程式會使用反射來尋找 WebSphere 的 WSSubject 類別。該類別具有 getCallerPrincipal 方法，來傳回用於登入 AuthServlet 的主體。AuthServlet 需要受保護，才能用來鑑別 WebSphere Application Server。

還有其他可傳回更多資訊的方法可供使用。有類似的方法可用來搭配任何應用程式伺服器使用。

```
public Result authenticateRequest
    (Request requestAttributes, Response responseAttributes)
    throws SSOException {

    Result result = null;

    try {
        Class<?> c1 =
            Class.forName("com.ibm.websphere.security.auth.WSSubject");
        Method theMethod = c1.getMethod("getCallerPrincipal",
            (Class[])null);
        String principal = (String)theMethod.invoke((Object[])null,
            (Object[])null);

        if (principal != null
            && principal.length() > 0
            && !principal.equals("UNAUTHENTICATED")) {
            result = new Result(Result.UseridOnlyOID, domain, principal);
            responseAttributes.setStatus(HttpServletResponse.SC_OK);
        } catch (Exception e) {
            throw new SSOException(e);
        }

        return result;
    }
}
```

在 authenticateRequest 的實作期間，您必須先設定回應狀態再傳回：

- 如果您不需要執行任何重新導向，且找到的資訊完好，請傳回下列資訊：
responseAttributes.setStatus(HttpServletResponse.SC_OK);
- 如果要求中沒有足夠的資訊來繼續進行有效的登入，請傳回下列資訊：
responseAttributes.setStatus(HttpServletResponse.SC_FORBIDDEN);
- 如果您需要執行重新導向來收集其他資訊，請傳回下列資訊：
responseAttributes.setStatus(HttpServletResponse.SC_MOVED_TEMPORARILY);
responseAttributes.sendRedirect(url);

有其他狀態值可供使用。請參閱 JavaDoc 中的 `HttpServletResponse`。

從登入錯誤中回復： 如果您的自訂攔截程式在測試時未正常運作，則會鑑別最可能的問題。這時會顯示一個錯誤頁面，並含下列資訊：

Build Forge Error

Access is denied to the Build Forge console

"Error authenticating:
com.buildforge.services.common.api.APIException - API:
Authentication Error."

Please click [here](#) to try the same type of login again
or click [here](#) to force a form login (user ID/password).

您有兩個選擇來進行回復：

- 重試登入。其會以相同的方式重新進行整份已配置攔截程式的清單。
- 強制執行表單登入。這個選擇會略過自訂攔截程式，並使用表單登入頁面。

方法來源報表： 下列註解和來源報表提供了 `ISSOInterceptor` 介面中關於方法的詳細資訊。

initInterceptor

```
/**
 * This method is called when the interceptor is loaded. A map of the
 * configuration properties is passed into the init method. You can create
 * the configuration properties from a BuildForge Environment and associate
 * it with the SSO configuration.
 *
 * @param initializationProps used to configure the implementation
 * @return true if successful, false if an error should be reported.
 * @throws SSOException if the initialization fails
 */
public boolean initInterceptor (Properties initializationProps) throws SSOException;
```

isTargetInterceptor

```
/**
 * This methods will review the attributes in the requestAttributes Map
 * to determine if there is something that this interceptor should
 * act on. If the interceptor return is "true", then the interceptor will
 * be responsible for authenticating the request and the authenticateRequest
 * method is invoked. If the interceptor return is "false", then this
 * interceptor is skipped and the next isTargetInterceptor in the list will
 * be called. Ordering of the interceptors during the configuration will
 * return which interceptor has the first shot at authenticating a request.
 *
 * @param requestAttributes attributes found in the inbound request
 * @return true if this interceptor will authenticate the request,
 *         false if it will not.
 * @throws SSOException
 */
public boolean isTargetInterceptor(Request requestAttributes) throws SSOException;
```

authenticateRequest

```
/**
 * This method is called on an interceptor that returns true for the
 * isTargetInterceptor method. The Request will contain data used
 * to perform the authentication. The Response is for the interceptor
 * to send information back to the client. The Result returned will contain
 * the following information if the status code is 200:
 *
 *
```

```

* OID: an object identifier of the SecurityContext that can process token
  information stored in this map when going to an Agent.
* Domain: a valid BF domain name or <default> if not known
  (the username must be valid in the configured realm).
* Username: a valid BF username. This will be used to lookup BFUser attributes
  that are used in checking authorization policy.
* @see com.buildforge.services.common.security.context.Result
*
* @param requestAttributes attributes found in the inbound request
* @param responseAttributes sent back in the outbound response
* @return com.buildforge.services.common.security.context.Result - result
  information that tells BF how to handle the authentication request.
* @throws com.buildforge.services.server.sso.SSOException
**/
public Result authenticateRequest(
    Request requestAttributes,
    Response responseAttributes)
    throws SSOException;

```

LogoutRequest

```

/**
* This method is called to logout a request. The first interceptor that
  returns true for the isTargetInterceptor method will perform the logout.
  The main point is to clean up any user-related security information that
  should not be kept. The interceptor can inspect the request and response
  objects to determine what needs to be removed.
*
* @param requestAttributes attributes found in the inbound request
* @param responseAttributes sent back in the outbound response
* @return boolean - true if request redirect to exit page,
  false if redirect to login page.
* @throws com.buildforge.services.server.sso.SSOException
**/
public boolean logoutRequest(
    Request requestAttributes,
    Response responseAttributes)
    throws SSOException;

```

在 Active Directory 網域中使用 SPNEGO 來實作單一登入

系統提供「簡式且受保護的 GSS-API 協議 (SPNEGO)」機制，在 Active Directory 網域中實作單一登入

開始之前

這項作業需要網路中有下列元素：

- Active Directory 網域
- 目錄伺服器主機名稱
- 「Kerberos 金鑰配送中心 (KDC)」主機名稱
- 在 Active Directory 網域中的主機上安裝有 Build Forge
- Active Directory 網域中的用戶端主機
- 在每一個用戶端上有 Kerberos 配置檔
- 在目錄伺服器主機上已安裝 Windows Server 2003 SP2 資源工具集
- 支援的 Web 瀏覽器

註：不支援 Internet Explorer 6 搭配 SPNEGO 使用。請使用支援的瀏覽器。

下列程序包含基於下列設定的範例：

- mycompany.com 是網域中所有主機使用的 TCP/IP 網域名稱。
- ITDEV.COM 是 Active Directory 網域的名稱。
- it_directory.mycompany.com 是其中執行目錄伺服器的主機。其也會執行 Kerberos KDC。
- it_domain.mycompany.com 是其中執行 Active Directory 網域控制站的主機。
- it_buildforge.mycompany.com 是其中安裝 Build Forge 的主機。
- bfuser 是 Build Forge 系統的網域使用者名稱。
- happy_user 是範例使用者的網域使用者名稱，該使用者會在 Web 瀏覽器中使用 SSO 來存取 Build Forge。

關於這項作業

請執行下列作業，於 Active Directory 網域和 KDC 中實作 SPNEGO SSO。每一項都提供了一個含有詳細程序的區段。

註：SPNEGO 攔截程式可以搭配 Active Directory 以外的 KDC 使用。

程序

1. 設定 Active Directory 使用者和服務主體。
2. 設定 Kerberos 檔。
3. 配置 Build Forge 來使用 Active Directory 和 SPNEGO。
4. 配置瀏覽器用戶端以進行安全存取。
5. 透過 SSO 存取 Build Forge。

設定 Active Directory 使用者和服務主體：

Build Forge 伺服器和 Build Forge 用戶端必須設定在 Active Directory 網域中。

開始之前

下列程序需要有 Windows 2003 SP2 的「支援工具」。其包含 setspn 指令，需要有這個指令才能在 Active Directory 中設定服務主體。請從 Windows Server 2003 產品 CD 或「Microsoft 下載中心」安裝「支援工具」。

關於這項作業

當 Build Forge 用戶端和伺服器位於 Active Directory 網域時，使用者會在登入 Windows 主機時產生 Kerberos 認證記號。接著當使用者嘗試存取 Build Forge 伺服器時，SPNEGO 攔截程式會接收使用者記號並加以驗證。驗證過的身分會傳給 Build Forge，透過已配置的 Microsoft Active Directory LDAP 伺服器來執行登入。

程序

1. 登入網域控制站主機。範例中的主機為 it_example.mycompany.com。
2. 將 Build Forge 主機加入 Active Directory 網域（如果其還不是成員）。在本例中，將主機 it_buildforge 加入 ITDEV.COM 網域中。該主機現在在網域中的完整名稱為：it_buildforge.ITDEV.COM
3. 將 Build Forge 使用者新增至 Active Directory 網域。在本例中，建立使用者 bfuser。

重要：

- 選取密碼永不到期。您可以選取其他密碼管理。不過，每次密碼到期時，您都需要為 Build Forge 伺服器輸入新密碼。
 - 在帳戶標籤中，選取帳戶授信可供委派
4. 如果不存在，請在 Microsoft Active Directory 中建立所有用戶端的使用者帳戶。在本例中會建立一個使用者：happy_user。
 5. 建立 Build Forge 的服務主體名稱 (SPN)。在此範例中，Active Directory 使用者 bfuser 與服務名稱 HTTP/it_buildforge.mycompany.com 相關聯，以建立 Build Forge 伺服器的 SPN it_buildforge。
setspn -A HTTP/it_buildforge.mycompany.com bfuser

HTTP 是 Build Forge 服務的服務名稱。

設定 Kerberos 鑑別的檔案：

您需要在 Build Forge 主機上設定啟動檔 (Kerberos 用戶端配置檔) 和 keytab 檔。

程序

1. 在執行 Build Forge 的主機上設定啟動檔。
 - Windows 系統：
 - 命名 krb.ini 檔，並將它放在 C:\winnt 中。如果不存在，請建立 C:\winnt。
 - 將 default_keytab_name 設為 FILE:C:\winnt\krb5.keytab。
 - UNIX 及 Linux 系統：
 - 命名 krb.conf 檔並將它放在 C:\winnt 中。
 - 將 default_keytab_name 設為 FILE:/etc/krb5.keytab。

下列範例檔是利用來自範例系統的網域和領域設定，針對 Windows 所設定的。

```
[libdefaults]
default_realm = ITDEV.COM
default_keytab_name = FILE:C:\winnt\krb5.keytab
default_tkt_enctypes = rc4_hmac
default_tgs_enctypes = rc4_hmac
# kdc_default_options = 0x40800000
forwardable = true
renewable = true
noaddresses = true
clockskew = 300
[realms]
ITDEV.COM = {
    kdc = it_directory.itdev.com:88
    default_domain = mycompany.com
}
[domain_realm]
.mycompany.com = ITDEV.COM
```

註：如果用戶端主機和 Build Forge 伺服器主機之間的時間偏差超過 300 秒，記號即無法運作。請將用戶端和伺服器主機上的時間、日期和時區，設定在偏差限制內。

2. 設定 Kerberos keytab 檔。當用戶端嘗試存取 Build Forge 伺服器 URL 時，Build Forge 伺服器會使用 keytab 檔來驗證 Kerberos 記號。請在「網域控制站」主機上使用 ktpass 指令來建立該檔案。ktpass 指令包含在必備的 Windows 資源工具集中。下例使用 Build Forge 服務的主體名稱，以及在範例實務中設定給 Build Forge

的 Active Directory 使用者名稱。請將 `-pass Rational` 換成您自己的密碼。範例中顯示的換行是便於閱讀。請勿在 `ktpass` 指令中使用換行。

```
ktpass -out C:\it_buildforge.keytab  
-princ HTTP/it_buildforge.mycompany.com@ITDEV.COM  
-mapuser bfuser -mapop set  
-pass Rational /crypto RC4-HMAC-NT /rndpass /ptype KRB5_NT_SRV_HST
```

將 `it_buildforge.keytab` 重新命名為 `krb5.keytab`，並將它放在 Build Forge 主機上含有 Kerberos 啟動檔的目錄中。

- Windows：C:\winnt\
- UNIX 和 Linux：/etc

配置 Build Forge 來使用 Active Directory 和 SPNEGO：

程序

1. 在 Build Forge 中，設定 LDAP 以指向 Active Directory 網域控制站。
 - a. 在 Build Forge 中，按一下**管理 > LDAP**。
 - b. 建立新的 LDAP 配置並設定如下的內容，來設定網域控制站的存取。
 - 名稱：設為 Active Directory 網域的名稱。在範例環境中，這會是 `itdev`。
 - 管理 DN：設為網域中的管理者使用者。
 - 對映存取群組：否
 - 主機：設為網域控制站主機的 IP 位址。
 - 連結使用者帳戶：是
 - 通訊協定：LDAP
 - 顯示名稱：displayname
 - 識別名稱：distinguishedname
 - 群組名稱：memberof
 - 郵件名稱：displayname
 - 搜尋基準：on=users,do=domainname,do=domainextension。在範例環境中，這會是 on=users,do=itdev,do=.com
 - 唯一 ID：sAMAccountNames=%
 - c. 按一下**設為預設值**。這項配置需要作為預設 LDAP 配置。
2. 設定 SPNEGO 的 Build Forge 環境變數。
 - a. 在 Build Forge 中，移至**環境 > SPNEGO SSO 的環境**
 - b. 將 `bf_spnego_service_name` 設為 HTTP 這會符合服務主體名稱。
 - c. 將 `bf_spnego_server_name` 設為 `it_buildforge.mycompany.com`（Build Forge 伺服器主機的完整主機名稱）。如果未設定這個變數，`INetAddress` API 會嘗試找出主機名稱。
 - d. 將 `bf_spnego_realm` 設為 `ITDEV.COM`（Kerberos 領域名稱）。如果未設定這個變數，則會使用 Kerberos 啟動檔中的值。
3. 啟用 SPNEGO 攔截程式。
 - a. 在 Build Forge 中，移至**管理 > 安全 > SSO > SPNEGO SSO 攔截程式**。
 - b. 將「作用中」內容設為「是」，然後按一下**儲存**。

- c. 在**管理 > 安全 > SSO** 中，將 **SPNEGO SSO 攔截程式**移至清單頂端。使用「SPNEGO SSO 攔截程式」之「SSO 選項」功能表中的**移至頂端**選項，然後按一下**儲存**。

配置 SSO 的用戶端瀏覽器：

用戶端瀏覽器必須設定安全設定，才能使用 SPNEGO。

關於這項作業

請使用用來存取 Build Forge 之瀏覽器（可為 Microsoft Internet Explorer 或 Mozilla Firefox）的用戶端設定指示。

程序

- 如果是 Internet Explorer，請執行下列動作。
 1. 登入 Active Directory 網域。在範例配置中，您會登入 itdev.com。
 2. 開啓 Internet Explorer。
 3. 按一下**工具 > 網際網路選項 > 安全性**。
 4. 選取**近端內部網路**，然後按一下**網站**。
 5. 在**近端內部網路**對話框中，勾選**包含所有未列在其他區域的近端內部網路網站**，然後按一下**進階**。
 6. 將執行 where Build Forge 的主機加入**網站**清單，然後按一下**確定**。
 7. 按一下**確定**以關閉**近端內部網路**對話框。
 8. 在**網際網路選項**視窗中，按一下**進階**標籤。
 9. 捲動至「**安全性**」群組，然後選取**啓用整合的 Windows 鑑別**（需要重新啓動）（如果尚未勾選）。
 10. 按一下**確定**。
 11. 重新啓動 Internet Explorer。
- 如果是 Mozilla Firefox，請執行下列動作。
 1. 登入 Active Directory 網域。在範例配置中，您會登入 itdev.com。
 2. 開啓 Firefox。
 3. 在位址欄位輸入 **about:config**。
 4. 在「**過濾器**」方框中，輸入 **network.n**。清單會自行更新。
 5. 按兩下 **network.negotiate-auth.trusted-uris**。輸入授信網域的清單。它應該包括目錄伺服器主機和 where Build Forge 伺服器主機（範例中的 it_directory.mycompany.com 和 it_buildforge.mycompany.com）。按一下**確定**。
 6. 設定委派。按兩下 **network.negotiate-auth.delegation-uris**，然後輸入瀏覽器可以委派使用者鑑別的網站清單。

透過 SSO 存取 Build Forge：

輸入伺服器 URL 來測試透過 SSO 進行登入。

程序

1. 利用 Active Directory 使用者清單中的使用者名稱，登入位於 Active Directory 網域的主機。

2. 開啟瀏覽器。
3. 輸入 Build Forge 伺服器主機的 URL。使用範例配置時，這會是 `http://it_buildforge.mycompany.com`。如果 SSO 已正確配置，您會看到 where Build Forge 管理主控台。
4. 驗證 where Build Forge 主控台右上方顯示的使用者名稱，是否符合用戶端的 Windows 登入名稱。

使用自訂攔截程式來與 WebSphere Application Server 安全整合

本節說明如何建立 SSO 攔截程式，來與 WebSphere Application Server 安全整合。

開始之前

您在 WebSphere Application Server 中必須以 LDAP 使用者認證來配置。

註：您必須使包含 WebSphere Application Server 使用者的 Build Forge LDAP 網域成為「預設」LDAP 伺服器。若要執行這項作業，請移至這個 Build Forge LDAP 網域，然後選取設為預設值。

關於這項作業

所提供的「表單 SSO 攔截程式」會以表單型登入頁面來鑑別使用者。以下是如何建立自訂 SSO 攔截程式的範例。自訂攔截程式會使用自訂攔截程式類別。

攔截程式類別會存取 WebSphere Application Server，以取得經過鑑別的使用者認證。取得這些認證之後，即會加以快取。後續登入會使用快取的認證。

保護授權服務 (AuthServlet)：

Build Forge 通常會在所提供的 Apache Tomcat 應用程式伺服器中，將其服務當成應用程式來執行。

關於這項作業

下列指示會配置 Build Forge 來使用 WebSphere Application Server，而非 Tomcat。請遵循第 509 頁的『使用 WebSphere Application Server 來取代 Apache Tomcat』一節中的指示，但有一個例外：`jas.war` 檔包含名為 `web.xml` 的檔案，您需要擷取並修改它，才能新增安全限制。在 WebSphere Application Server 下安裝這個應用程式之前，修改 `war` 檔之後必須重新產生它，才能使用這個檔案的受保護版本。

若要重新產生 `war` 檔，請完成下列步驟。

程序

1. 導覽至包含 `jas.war` 檔的目錄（Tomcat 伺服器根目錄中的 `webapps` 目錄）。將這個檔案複製到暫存位置，例如：`C:\jas`。
2. 展開 WAR 檔。從指令行執行指令：`%IBM_JAVA_HOME%\jar -xvf jas.war`，以展開 `war` 檔的內容。

註：Java 必須可供使用，且必須已建立 `IBM_JAVA_HOME` 環境變數。

3. 儲存 `jas.war` 檔以供之後擷取：
 - a. Windows：`copy jas.war jas.war.bak`

b. UNIX 或 Linux : cp jas.war jas.war.bak

4. 在 WEB-INF 目錄中找出 web.xml 檔（在從 war 檔展開的檔案中）。請編輯這個檔案來新增安全環境定義。例如，在檔案尾端的 </web-app> 標籤前加入以下各行：

```
<security-constraint id="SecurityConstraint_1">
    <web-resource-collection id="WebResourceCollection_1">
        <web-resource-name>*/</web-resource-name>
        <url-pattern>/AuthServlet/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
        <http-method>PUT</http-method>
        <http-method>DELETE</http-method>
    </web-resource-collection>
    <auth-constraint id="AuthConstraint_1">
        <description>myconstraint:+</description>
        <role-name>User</role-name>
    </auth-constraint>
    <user-data-constraint id="UserDataConstraint_1">
        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>
<login-config id="LoginConfig_1">
    <auth-method>BASIC</auth-method>
    <realm-name>full-qualified-domain</realm-name>
</login-config>
<security-role id="SecurityRole_1">
    <role-name>User</role-name>
</security-role>
```

註：<auth-method> 可以是 WebSphere Application Server 支援的任何 J2EE 鑑別方法。最常用的鑑別方法是 FORM，其需要額外的配置參數。請參閱 WebSphere Application Server 文件，以取得在應用程式中配置 FORM 的指示。

5. 從指令行執行下列指令，以重新產生 war 檔（從解壓縮的同一個目錄中呼叫）：

```
%IBM_JAVA_HOME%\jar -cvf jas.war
```

您現在應該有新版的 jas，其已修改來保護含有 J2EE 限制的 AuthServlet。請完成搭配 WebSphere Application Server 執行的安裝指示，並透過應用程式->安裝新的應用程式來安裝這個版本的 jas.war。在配置 WebSphere Application Server 時，請確定 Build Forge 不在執行中。

安裝之後，移至**應用程式 > 企業應用程式**。按一下要配置的應用程式名稱。在**詳細內容**下，按一下**安全角色至使用者/群組對映鏈結**。請對「使用者」選取**所有已鑑別的使用者**勾選框。進行這項變更之後，將變更儲存至主要配置。

WebSphere Application Server 下也需要啓用應用程式安全。若要執行這項作業，請移至**安全 > 安全管理、應用程式和基礎架構**。請務必勾選**啓用應用程式安全**。

此時，請重新啓動 WebSphere Application Server，然後重新啓動 Build Forge 伺服器。

註：jas 現在會在 WebSphere Application Server 啓動過程中啓動，所以在啓動 Build Forge 之前，請先啓動 WebSphere Application Server。

建立新的 SSO 配置：

建立新的 SSO 配置來使用攔截程式。

程序

1. 在 Build Forge 主控台中，移至**管理 > 安全 > SSO**。
2. 按一下**新增 SSO 配置**。
3. 設定配置的內容。
 - **名稱** - 輸入這個配置的名稱。
 - **Java 類別** - `com.buildforge.services.server.sso.was.WebSphereSSOInterceptor` 輸入
 - **作用中** - 選取是。
4. 按一下**儲存**。
5. 移動這個配置，使其成為清單的第一個項目。在配置名稱左邊的功能表中，選取**移至頂端**。

將 **LDAP 使用者或 LDAP 群組** 對映到本端使用者：

將 LDAP 使用者或 LDAP 群組對映到本端使用者，例如本端 root 使用者。

關於這項作業

建立 WebSphere Application Server SSO 攔截程式後，您可以將攔截程式配置成：

- 將 LDAP 使用者對映到本端使用者，包含 root
- 將 LDAP 群組對映到本端使用者，例如 root

若要設定對映，請定義環境，然後在攔截程式中參照該環境。

定義環境：

程序

1. 在 Build Forge 主控台中，按一下**環境**。
2. 指定環境的名稱，然後按一下**儲存環境**。
3. 將 LDAP 使用者對映到本端使用者，設定及儲存下列變數：
 - a. 將 `ldap_user_list_mapping` 設為 `LDAP_user1|local_user1;LDAP_user2|local_user2;LDAP_user3|local_user3`
 - b. 將 `ldap_realm_name` 設為 `LDAP_domain_name`

例如，假設設定下列環境：

```
ldap_user_list_mapping = user1|root;user2|root;user3|root
ldap_realm_name = bluepages.ibm.com:389
```

在此情況下，LDAP user1、user2 和 user3 全部都會對映到本端 root 使用者。

4. 若要將特定 LDAP 群組的成員對映到本端使用者，請設定並儲存下列變數：
 - a. 將 `ldap_group_name_mapping` 設為 `LDAP_group|local_user`
 - b. 將 `ldap_realm_name` 設為 `LDAP_domain_name`

註：如果您同時使用這兩種對映，則使用者對映優先於群組對映。

群組對映範例，假設設定下列環境：

```
ldap_group_name_mapping = cn=bf_admin,ou=memberlist,ou=ibmggroups,o=ibm.com|root
ldap_realm_name = bluepages.ibm.com:389
```

在此情況下，LDAP 群組 `cn=bf_admin,ou=memberlist,ou=ibmggroups,o=ibm.com` 的成員就會對映到本端 `root` 使用者。

參照環境：

程序

1. 在 Build Forge 主控台中，按一下**管理 > 安全 > SSO**。
2. 按一下 WebSphere Application Server SSO 攔截程式的名稱。
3. 針對您剛剛定義的環境配置設定**環境**內容。
4. 按一下**儲存**。
5. 移動這個配置，使其成為清單的第一個項目。在配置名稱左邊的功能表中，選取**移至頂端**。

執行 SSO 自訂攔截程式：

您可以使用新配置登入。

關於這項作業

這個自訂的 SSO 攔截程式現在容許 WebSphere Application Server 安全技術鑑別使用者（透過將 `AuthServlet` 要求當成使用者傳給 Build Forge 的方式）。

程序

1. 開啟 Web 瀏覽器，並輸入位址 `http://localhost`。您現在會看到鑑別頁面，而非 Build Forge 登入表單。
2. 輸入使用者認證，然後按一下 **Enter** 鍵。

註：鑑別之後，就會自動登入。

登出之後，會顯示預設的 `jsp` 頁面，而非登入表單。如果您仍然通過鑑別，則會自動完成後續的登入。

回復至表單型 SSO：

您可以回復使用 SSO 登入表單。

關於這項作業

若要重新配置系統以使用「表單登入」，您必須在 WebSphere Application Server 下解除安裝 `jas`，然後重新安裝原始的 `jas.war` 檔。請在 Build Forge 下，確定已啟用表單型 SSO 攔截程式，且是列出的最上方項目。停用自訂 WebSphere Application Server 攔截程式。若要讓變更生效，您必須重新啟動 WebSphere Application Server 及 Build Forge。

程序

1. 解除安裝 `jas`，並重新安裝原始 `jas.war` 檔。
2. 在 Build Forge 下，確定已啟用表單 SSO 攔截程式，且是列出的第一個項目（請參閱上述「Build Forge SSO 安全配置」）。
3. 停用自訂 WebSphere Application Server 攔截程式。
4. 重新啟動 WebSphere Application Server。

5. 重新啟動 Build Forge。

啓用 SSL 和 HTTPS

將 Build Forge 系統配置成使用 SSL 與 HTTPS，可提高系統的安全。SSL 包含端點鑑別和資料加密。

依預設，Build Forge 系統對表單登入（其在 Apache Tomcat 上會使用鑑別 Servlet）只會使用 SSL。如果要在整個 Build Forge 系統中啓用其他 SSL 保護，則需要下列設定：

1. 在 Apache 伺服器上啓用 SSL。只有當您未於安裝期間指定啓用 SSL 時，才需要這個步驟。
2. 啓用 SSL 以進行用戶端和內部通訊
3. 對代理程式啓用 SSL

註：如果您要與 WebSphere 元件整合，請確定已符合 SSL 支援的必要條件。

- 如需與 WebSphere Application Server 進行整合的相關資訊：請參閱第 509 頁的『使用 WebSphere Application Server 來取代 Apache Tomcat』。

關於預設登入安全

依預設，Build Forge 包括登入安全。當使用者登入時，該要求會重新導向至鑑別 Servlet。輸入的使用者名稱和密碼會加密供 Servlet 使用。如果登入成功，則會顯示主控台使用者介面首頁。用戶端和主控台之間的後續階段作業通訊，可能透過 http（預設值）或透過 https 來進行。使用 https 需要其他系統配置。請參閱『啓用 SSL 和 HTTPS』。

在安裝期間，您會提供用於加密的金鑰儲存庫密碼。您也可以選擇安裝自簽憑證。

關於自簽憑證的憑證訊息： 如果 Build Forge 已安裝自簽憑證，透過啓用安全的瀏覽器來存取系統的使用者會收到關於憑證的警告訊息。

如果要避免收到這些警告，請將憑證分送給使用者以安裝在其瀏覽器中。安裝憑證的特性會視瀏覽器而有所不同。請參考瀏覽器文件。

該憑證位於 `<bfinstall>/keystore` 中。

停用預設登入安全： 如果已停用登入安全，則在登入期間，使用者認證會以未加密的明碼來傳遞至主控台。如果主控台已配置使用 HTTPS/SSL，停用登入安全並不會影響主控台使用 HTTPS/SSL。

若要停用鑑別 Servlet，請完成下列步驟：

1. 停止 Build Forge（如果其在執行中）。
2. 編輯 `<bfinstall>/buildforge.conf` 來指定 HTTP 和埠 8080，以便與服務層通訊。

將下列這一行：

```
services_url https://hostname:8443/jas
```

變更成下列這一行：

```
services_url http://hostname:8080/jas
```

3. 編輯服務層配置檔來關閉強制使用的 SSL。編輯 `<bfinstall>/Apache/tomcat/webapps/jas/WEB-INF/web.xml`。將 ForceHttps 設定變更為 false。如果此設定不在檔案中，請新增它，如粗體字所顯示。


```

<display-name>A Services Layer Bootstrap Servlet</display-name>
<servlet>
  <servlet-name>ServicesBootstrap</servlet-name>
  <servlet-class>com.buildforge.services.server.web.BootstrapServlet</servlet-class>
  <init-param>
    <param-name>port</param-name>
    <param-value>3966</param-value>
  </init-param>
  <init-param>
    <param-name>sslPort</param-name>
    <param-value>49150</param-value>
  </init-param>
  <init-param>
    <param-name>ForceHttps</param-name>
    <param-value>false</param-value>
  </init-param>
  <load-on-startup>0</load-on-startup>
</servlet>

```

4. 啟動 Build Forge。

註：如果已停用鑑別 Servlet，則使用者認證會以明碼透過網路傳遞，而引起安全風險。

關於 SSL 和 Build Forge 元件

依預設，Build Forge 元件會設為在啟用 SSL 時，使用特定的埠和安全設定。

預設 SSL 設定： 當您使用預設憑證時，啟用 SSL 相對簡單。本節的程序是以該實務範例為基礎。

不過，通常在每一個系統上使用相同的憑證（私密金鑰）並不明智。如果其中一個系統上的私密金鑰受損，整個基礎架構也可能受損。您可以強制實施實體安全來減少受損的機會。

更安全的系統會對每一個處理程序都使用一個憑證。在 Build Forge 中，這意味著您需要執行下列動作：

- 建立每一個代理程式的憑證。
- 建立每一個引擎的憑證。這適用於設定備用功能的狀況。請參閱第 85 頁的『配置備用功能』。

這項設定需要其他憑證管理。您有幾個選擇：

- 您可以使用 CA（憑證管理中心）來產生憑證。這樣做可減少簽章者交換的數目。
- 您可以確定每一個信任儲存庫或 CA 儲存庫都有建立連線所需的簽章者。

以下各節識別在強制實施 SSL 安全的 Build Forge 系統中的介面。

用戶端介面： 使用者會透過用戶端介面來存取 Build Forge 系統。

Build Forge 的 Web 用戶端

Web 用戶端會透過其 Apache Web 伺服器來存取 Build Forge。當您啟用 SSL 並使用啓用安全的 Web 瀏覽器時，會使用下列介面。

- **Apache Web 伺服器埠 443**

Web 用戶端會透過其 URL 來存取 Build Forge。當您啟用 SSL 時，URL 如下所示：

`https://host/`

host 是執行 Build Forge 所在的主機。如果您設定 443 以外的埠來對 Apache 進行安全存取，使用者也必須指定埠：

`https://host:port/`

Web 用戶端會重新導向至在 Apache Tomcat 伺服器上執行的鑑別 Servlet。

- **Apache Tomcat 應用程式伺服器埠 8443**

鑑別 Servlet 會接受登入認證並鑑別使用者。該 Servlet 會加密認證，因此其在連線中絕不會以明碼出現。

Apache Tomcat Servlet 所使用的接聽器埠配置，是透過配置檔來管理。此檔案位於 `<bfinstall>/Apache/tomcat/conf/server.xml` 中。請找出下列連接器配置。

```
<Connector port="8443" maxHttpHeaderSize="8192" algorithm="IbmX509"
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
  enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" scheme="https" secure="true"
  clientAuth="false" sslProtocol="SSL_TLS"
  keystoreFile="C:\BuildForge71.536\keystore\buildForgeKeyStore.p12"
  keystorePass="password"
  keystoreType="PKCS12"
  truststoreFile="C:\BuildForge71.536\keystore\buildForgeTrustStore.p12"
  truststorePass="password"
  truststoreType="PKCS12"/>
```

Build Forge 的 API 程式用戶端

- **Apache Tomcat 應用程式伺服器埠 49150**

API 用戶端會透過其服務層元件（這是一個在 Apache Tomcat 應用程式伺服器上執行的應用程式）來存取 Build Forge。API 用戶端需具備有效的 `bfclient.conf` 檔。

服務層元件會使用 SSL 配置來進行入埠通訊。其定義在 Build Forge 主控台的管理 > 安全 > **SSL** 中。使用的預設值為預設 JSSE 入埠 SSL。

內部介面： Build Forge 是由 Web 介面元件（Apache Web 伺服器和 PHP）、服務層元件和引擎元件所組成。Web 介面和引擎元件是服務層元件的用戶端。API 程式用戶端也是服務層元件的用戶端。

服務層級入埠

- **Apache Tomcat 應用程式伺服器埠 49150**

服務層元件會使用 SSL 配置來進行入埠通訊。其定義在 Build Forge 主控台的管理 > 安全 > **SSL** 中。使用的預設值為預設 JSSE 入埠 SSL。

出埠服務層用戶端

Apache Tomcat 應用程式伺服器埠 49150 Web 介面元件（透過 PHP）和引擎元件兩者都使用一個 SSL 配置，來專用於服務層元件的出埠通訊。其定義在 Build Forge 主控台的管理 > 安全 > **SSL** 中。使用的預設值為預設 JSSE 出埠 SSL。

用戶端出埠配置和服務層入埠配置的 SSL 內容必須相容，SSL 信號交換才能成功。管理 > 安全 > **SSL**，類型和信號交換通訊協定內容必須相符。

每一個 SSL 配置都有一個參照金鑰儲存庫配置：

- 金鑰儲存庫配置：定義含有私密憑證之金鑰儲存庫的內容。
- 信任儲存庫配置：定義含有授信簽章者之金鑰儲存庫的內容。

配置是以名稱來指定。您可以在**管理 > 安全 > SSL**中定義它們。系統提供了一些預設值。

外部介面： 外部介面是 Build Forge 用來與外部系統交談的介面。

- Build Forge 引擎會與代理程式進行通訊。
- Build Forge 服務層元件會與資料庫通訊。

Build Forge 引擎與代理程式的通訊

對這個介面啓用 SSL 需要下列項目：

- 代理程式的配置。它要求變更代理程式配置檔，並將憑證放在代理程式主機上。
- 對使用代理程式的每一個「伺服器」資源啓用 SSL 通訊。您可以在**伺服器**畫面的主控台中執行這項作業。

請參閱第 107 頁的『啓用 SSL 來進行代理程式通訊』。

Build Forge 服務層元件與資料庫的通訊

這個介面的 SSL 配置定義在資料庫的裝置驅動程式中。

在 Apache 伺服器上啓用 SSL

依預設，Build Forge 元件會設為在啓用 SSL 時，使用特定的埠和安全設定。

在安裝期間，系統會在「Web 和應用程式伺服器」畫面上詢問您下列問題：**您要使用安全 HTTP 嗎？**

如果您回答「是」，即會在 Apache 伺服器上啓用 SSL。請略過本節。

如果您回答「否」，但確實要啓用 SSL，請登入 Build Forge 主機，並依下列方式配置 Apache 伺服器：

- httpd.conf 檔

編輯 `<bfinstall>/Apache/conf/httpd.conf` 以使用下列設定：

```
Listen 0.0.0.0:443
ServerName localhost:443
```

- ssl.conf 檔

編輯 `<bfinstall>/Apache/conf/ssl/ssl.conf` 以使用下列設定：

```
<VirtualHost *:443>
ServerName localhost:443
SSLEngine on
```

如果您選擇，您可以指定主機的完整網域名稱。

對用戶端和內部連線啓用 SSL

依預設，Build Forge 元件會設為在啓用 SSL 時，使用特定的埠和安全設定。

如果要啓用從用戶端到 Build Forge 以及 Build Forge 內部元件之間的 SSL 連線，請確實依指定的順序執行下列動作：

- 在主控台中，移至**管理 > 安全**。
- 將 **SSL 已啓用**設為「是」。這時會顯示數個其他內容。請將它們保留設為預設值。如有需要，之後可以自訂這些內容。
- 按一下**儲存**。SSL 配置會儲存在 Build Forge 資料庫中。
- 按一下**更新主要 BFClient.conf**。SSL 配置會用來更新 BFClient.conf 檔。這些設定必須在這個檔案中，Build Forge 才能使用。
- 停止 Build Forge 並重新啓動。這是必要的。

確認系統可以用預設值運作之後，您可以編輯影響 SSL 的內容。

註：如果您要用戶端使用 SSL，但不要在內部用戶端（Web 介面元件和引擎元件）與服務層元件之間使用 SSL，請執行下列動作：

- 手動編輯 BFClient.conf 檔。此檔案位於 `<bfinstall>` 中。
- 變更 `bf_services_preferred_protocol` 設定 `tcp`。
`bf_services_preferred_protocol to tcp`

如果 Build Forge 主機實際上是安全的，這項配置可在最低的安全風險下提昇效能。

在鎖定的系統上重新啓用 TCP 通訊

如果 SSL 中發生配置錯誤，系統會將您鎖定。

如果要存取已鎖定的系統，請執行下列動作：

- 停止 Build Forge。
- 在 `<bfinstall>` 目錄中，以文字編輯器開啓 `bfclient.conf` 檔。
- 變更通訊協定內容，如下所示：
`bf_services_preferred_protocol=tcp`
- 啓動 Build Forge。

您應該能夠登入。

註：變更通訊協定並不會停用預設已啓用的安全登入鑑別。使用者會重新導向至安全連線，允許登入認證與 Build Forge 進行安全通訊。

為 API 用戶端（Perl 或 Java）啓用 SSL

使用 `bfclient.conf`，為使用 Perl 或 Java 來撰寫的 API 用戶端配置 SSL 連線。

關於這項作業

若要為 API 用戶端程式建立 SSL 連線，請設定 `bfclient.conf`。此檔案包含 SSL 配置內容。為了簡化程序，以下步驟假設，您在第一個步驟中提到的用戶端目錄裡執行您的用戶端。

程序

1. 在您要執行 Script 的地方建立用戶端目錄。

Windows : `mkdir c:\client`

UNIX 或 Linux : `mkdir /temp/client`

2. 將 `bfclient.conf` 複製到用戶端目錄。

Windows : `copy c:\BuildForge\bfclient.conf c:\client`

UNIX 或 Linux : `cp /opt/BuildForge/Platform/bfclient.conf /temp/client`

3. 在用戶端目錄中建立金鑰儲存庫。

Windows : `mkdir c:\client\keystore`

UNIX 或 Linux : `mkdir /temp/client/keystore`

4. 將金鑰儲存庫從 BuildForge 安裝複製到用戶端金鑰儲存庫目錄。

Windows : `copy \BuildForge\keystore*.pem \client\keystore`

UNIX 或 Linux : `cp /opt/BuildForge/Platform/keystore/*.pem /temp/client/keystore`

5. 若為 Perl，請設定下列 OS 環境變數，這樣即可找到 `bfclient.conf` 位置。

Windows : `set BFCLIENT_CONF=c:\client\bfclient.conf`

UNIX 或 Linux : `export BFCLIENT_CONF="/temp/client/bfclient.conf"`

6. 若為 Java，請完成下列步驟：

- a. 在您的程式碼中使用 `SecureAPIClientConnection`，而不是 `APIClientConnection`，以便使用 `bfclient.conf` 建立 SSL 連線。
- b. 執行您的 Script 時，在 Java 指令行設定下列系統指令。

Windows : `-Dcom.buildforge.client.config=c:\client\bfclient.conf`

UNIX 或 Linux : `-Dcom.buildforge.client.config=/temp/client/bfclient.conf`

7. 編輯 `bfclient.conf` 並檢查下列內容：

- `bf_services_preferred_protocol=ssl`

確定值為 `ssl`，而不是 `tcp`。

- `bf_services_ssl_port=49150`

確定 49150 是您的服務層 SSL 埠。

- `bf_keystore_location=./keystore/buildForgeKey.pem`

金鑰儲存庫位置有很多個。在用戶端目錄以外的目錄中執行您的 Script，將每一個金鑰儲存庫位置變更為完整路徑。

8. 如果希望確定您的 Script 正確使用 `bfclient.conf`，請在您的環境中設定下列除錯內容。執行您的 Script 時，您應該會看到更多有關 SSL 連線內容的輸出。

Windows : `set BFDEBUG_SECURITY=1`

UNIX 或 Linux : `export BFDEBUG_SECURITY=1`

啓用 SSL 來進行代理程式通訊

啓用 SSL 時，Build Forge® 元件依預設會設為使用特定的埠和安全設定。

若要在 Build Forge® 和代理程式之間啓用 SSL 通訊，您必須執行下列動作：

- 在 UNIX 電腦上，確定 GCC 程式庫已安裝在管理代理程式的電腦上。
- 必要條件：對用戶端和內部通訊啓用 SSL。請參閱第 104 頁的『對用戶端和內部連線啓用 SSL』。
- 配置每一個代理程式。這項作業包含：
 - 新增憑證至代理程式主機
 - 編輯代理程式的 `bfagent.conf` 檔。
- 在主控台上，在連接代理程式的每一個伺服器定義中啓用 SSL。

配置代理程式使用 SSL

1. 如果代理程式在執行中，請將它停止。
2. 將憑證的 .PEM 檔放在代理程式的安裝根目錄中。

若要快速地實作並測試 SSL，請從 Build Forge 安裝中複製 .PEM 檔。這些檔案位於 `<bfinstall>/keystore` 中。

註：如果您的代理程式版本 (`bfagent -v`) 不等於或大於 8.0，則必須使用代理程式安裝資料夾中的舊版代理程式 `openssl` 公用程式，從新的 `buildForgeKeyStore.p12` 來重新產生 `buildForgeKey.pem`，以供舊版代理程式使用。例如，

```
openssl pkcs12 -in buildForgeKeyStore.p12 -passin pass:<bfpasword> -passout  
pass:<bfpasword> -out buildForgeKey.pem
```

SSL 的最佳實務是對每一個代理程式使用個別的憑證：

- a. 對下列每一項建立個別的金鑰儲存庫檔 (.PEM)：
 - 私密金鑰（金鑰）
 - 私密金鑰的公開憑證（憑證）
 - 授信簽章者（CA 或憑證管理中心）
 - b. 如果您是使用代理程式的唯一憑證（而非 Build Forge® 引擎憑證的副本），請將代理程式的憑證加入 Build Forge® 的憑證管理中心金鑰儲存庫 `<bfinstall>/keystore/buildForgeCA.pem`。如果您正在執行多個引擎（冗餘引擎），請將憑證加入每一個引擎的憑證管理中心金鑰儲存庫中。
 - c. 如果您要在 on System z 上使用 `kdb`，請使用 `gskkeyman` 建立 `kdb`。
 - d. 如果您要使用「系統授權機能 (SAF)」金鑰環，請使用 `RACF®` 建立金鑰環，並將憑證連接到此金鑰環。
3. 編輯 `BFAgent.conf`。檔案中的以下各行已註銷。請移除註解字首。

```
ssl_key_location buildForgeKey.pem  
ssl_key_password password  
ssl_cert_location buildForgeCert.pem  
ssl_ca_location buildForgeCA.pem  
ssl_protocol TLSv1  
ssl_cipher_group ALL
```

`password` 會用於 `buildForgeKey.pem` 金鑰儲存庫。如果您要將它加密，請參閱第 123 頁的『在 `buildforge.conf` 和 `bfagent.conf` 中加密密碼』。

如果您要在建立代理程式的連線時需要進行用戶端鑑別，請解除下面這一行的註解：

```
ssl_client_authentication true
```

這項設定需要將引擎憑證加入代理程式的憑證管理中心金鑰儲存庫 `buildForgeCA.pem` 中。

如果您要使用特定的密碼，請解除以下這一行的註解，並加入您的密碼清單：

```
ssl_cipher_override cipher_list
```

4. 對於在 `n System z` 上執行的代理程式，編輯 `BFAgent.conf`，並解除以下幾行註解：

```
gsk_ssl_key_location <root/cert SAF>
gsk_ssl_key_location /etc/key.kdb
gsk_ssl_kdb_password <password>
gsk_keyring_label <certificate label>
gsk_ssl_protocol ALL
gsk_ssl_cipher_v2 6321
gsk_ssl_cipher_v3 0906030201
gsk_ssl_client_authentication true
gsk_password_encrypt false
```

請只使用一個 `gsk_ssl_key_location`。您可以選擇 `SAF` 金鑰環或 `kdb`。SAF 金鑰環由 `<userid>/<keyring>` 所指定。

5. 啟動代理程式。代理程式必須在執行中，才能測試來自主控台的連線。

在伺服器定義中啟用 SSL

主控台會使用伺服器定義來連接代理程式。

針對連接已啟用 SSL 之代理程式的每一個伺服器定義，請執行下列動作：

- 在主控台上，移至**伺服器**畫面。
- 按一下伺服器定義名稱。
- 在伺服器定義的**詳細資料**標籤上，執行下列動作：
 - 將 **SSL 已啟用**設為是。
- 按一下**儲存**。
- 按一下**測試連線**。

對代理程式的 SSL 通訊進行疑難排解

下列核對清單說明在啟用 SSL 時的常見問題。

- 代理程式
 - 雖然安全和伺服器定義中已啟用 SSL，但未配置代理程式 SSL。
 - Build Forge® 引擎不信任代理程式的憑證。代理程式憑證需要加入引擎的 CA 金鑰儲存庫中：

```
<bfinstall>/keystore/buildForgeCA.pem
```
 - BFAgent.conf 中指定的金鑰儲存庫密碼不正確。
 - BFAgent.conf 中已指定用戶端鑑別，但引擎的憑證尚未加入代理程式的憑證管理中心 `buildForgeCA.pem` 中。

- Build Forge® 主控台
 - 主控台已啟用 SSL。其未加以儲存，或是在儲存前按了**更新主要 Bfclient.conf**，或是在儲存後未按**更新主要 Bfclient.conf**。
- 伺服器定義
 - 未順利啟用 SSL。您必須先按一下**儲存**，再按一下**測試連線**。
- 引擎和代理程式設定相符
 - 信號交換通訊協定不符。在引擎 SSL 配置和代理程式 SSL 配置中，信號交換通訊協定必須設為相同的值：TLSv1 或 SSLv3。預設值為 **TLSv1**。
 - 密碼組合不符。引擎 SSL 配置和代理程式 SSL 配置中指定的密碼組合，必須有通用的密碼。預設密碼組合群組為 **ALL**。

啟用除錯訊息

您可以在引擎和代理程式中啟用除錯。如果啟用除錯，即會產生其他詳細輸出，可協助您識別配置中的問題。

- 啟用引擎除錯：
 1. 在啟用 SSL 之伺服器定義所使用的環境中，新增下列變數：


```
BFDEBUG_SECURITY=1
```
 2. 停止引擎再重新啟動。
 - 在 Windows 上，於前景中啟動引擎。輸出會顯示在指令視窗中。
 - 在 UNIX 或 Linux 上，開啓除錯來啟動引擎：


```
cd <bfinstall>/rc
./buildforge start
```

來自引擎的輸出會寫入 `<bfinstall>/log` 中的引擎日誌檔。

- 啟用代理程式除錯：
 1. 停止代理程式。
 2. 將下面這一行加入 BFAgent.conf：


```
activity_log bfaagent.log
```

在本例中，代理程式會將輸出寫入 `bfaagent.log`。您可以指定不同的檔名。

註：如果代理程式是當成服務來執行，請指定絕對路徑。

3. 啟動代理程式。

成功 SSL 連線的範例引擎除錯輸出

當引擎順利連接代理程式時，就會產生下列輸出。

```
SSL_ca_file: ./keystore/buildForgeCA.pem
SSL_cert_file: ./keystore/buildForgeCert.pem
SSL_key_file: ./keystore/buildForgeKey.pem
SSL_verify_mode: 0x01
SSL_version: TLSv1
SSL_cipher_list: ALL
SSL_use_cert: 1
Making as SSL connection using socket IO::Socket::INET=GLOBAL(0x1e8f0f4).
SSL connection to agent.
DEBUG: .../IO/Socket/SSL.pm:1387: new ctx 80662848
DEBUG: .../IO/Socket/SSL.pm:880: dont start handshake: IO::Socket::SSL=GLOBAL(0x1e8f0f4)
DEBUG: .../IO/Socket/SSL.pm:284: ssl handshake not started
```

```

DEBUG: .../IO/Socket/SSL.pm:327: Net::SSLeay::connect -> 1
DEBUG: .../IO/Socket/SSL.pm:382: ssl handshake done
Socket is of type: ref(IO::Socket::SSL=GLOB(0x1e8f0f4))
ReadyLine: 202 HELLO TLS - BuildForge Agent v_VERSION_
.
Storing Agent Version [999.999.999.999-999-9999] for [08974C8E-6C3B-1014-972D-D9B2901D9F42]
cmd ping
username pbirk
encpass c1713f4a31af3f1300f7b2414a24559c4d6097e07310cf9c412e
go
Sending agent request...

```

成功 SSL 連線的範例代理程式除錯輸出

正常執行的代理程式在建立 SSL 連線時，會產生下列輸出。

```

[2256] main.c          : 409: === NEW AGENT ===
[2256] io.c             : 264: In start SSL
[2256] io.c             : 89: Key location: buildForgeKey.pem
[2256] bfpwdloader.c: 134: Looking for password locator: ssl_key_password_locator
[2256] bfpwdloader.c: 244: Looking for password for prop
                        ssl_key_password from bfaagent.conf.
[2256] bfcryptloader.c : 202: Loading password encryption module.
[2256] bfcryptloader.c : 276: Password encryption property
                        password_encrypt_module is not configured.
[2256] bfcryptloader.c : 539: Password decoded.
[2256] io.c             : 98: Cert location: buildForgeCert.pem
[2256] bfpwdloader.c: 134: Looking for password locator:
                        ssl_cert_password_locator
[2256] bfpwdloader.c: 244: Looking for password for prop
                        ssl_cert_password from bfaagent.conf.
[2256] io.c             : 153: Setting key password in default userdata.
[2256] io.c             : 160: Getting private key from PEM.
[2256] io.c             : 166: Checking private key from PEM.
[2256] io.c             : 172: Getting CA store information.
[2256] bfpwdloader.c: 134: Looking for password locator:
                        ssl_ca_password_locator
[2256] bfpwdloader.c: 244: Looking for password for prop
                        ssl_ca_password from bfaagent.conf.
[2256] io.c             : 178: CA location: buildForgeCert.pem
[2256] io.c             : 184: Checking the CA store.
[2256] io.c             : 230: Returning from init_CTX.
[2256] io.c             : 281: Calling SSL_new
[2256] io.c             : 294: Calling SSL_accept.
[2256] io.c             : 346: Cipher chosen: AES256-SHA
[2256] io.c             : 367: ssl_state = SS_CERTIFIED

```

代理程式的金鑰儲存庫密碼不正確時的範例輸出

如果配置在代理程式端的金鑰儲存庫密碼是錯誤的，則會顯示在引擎和代理程式輸出中。

引擎輸出（摘錄）：

```

SSL_use_cert: 1
Making as SSL connection using socket IO::Socket::INET=GLOB(0x1e8f0f4).
SSL connection to agent.
DEBUG: .../IO/Socket/SSL.pm:1387: new ctx 80662848
DEBUG: .../IO/Socket/SSL.pm:880: dont start handshake: IO::Socket::SSL=GLOB(0x1e8f0f4)
DEBUG: .../IO/Socket/SSL.pm:284: ssl handshake not started
DEBUG: .../IO/Socket/SSL.pm:327: Net::SSLeay::connect -> -1
DEBUG: .../IO/Socket/SSL.pm:1135: SSL connect attempt failed with unknown error
error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number

DEBUG: .../IO/Socket/SSL.pm:333: fatal SSL error: SSL connect attempt failed with unknown error

```

```
error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version number
DEBUG: .../IO/Socket/SSL.pm:1422: free ctx 80662848 open=80662848 80566656
DEBUG: .../IO/Socket/SSL.pm:1425: OK free ctx 80662848
```

代理程式輸出（摘錄）：

```
[ 5272] io.c : 98: Cert location: buildForgeCert.pem
[ 5272] bfpwdlocloader.c: 134: Looking for password locator: ssl_cert_password_locator
[ 5272] bfpwdlocloader.c: 244: Looking for password for prop ssl_cert_password from bfaagent.conf.
[ 5272] io.c : 153: Setting key password in default userdata.
[ 5272] io.c : 160: Getting private key from PEM.
[ 5272] io.c : 218: Failure reason: SSL_ERROR_BAD_KEY_FILE
[ 5272] io.c : 221: OpenSSL error string: error:00000000:lib(0):func(0):reason(0)
[ 5272] io.c : 281: Calling SSL_new
[ 5272] platform.c :2693: platform_release_credentials
[ 5272] main.c : 412: --- EXITING ---
```

代理程式輸出中的錯誤碼

這份清單包括您可能會遇到的一些其他錯誤碼和其原因：

- **SSL_ERROR_BAD_CA**: 在 `buildForgeCA.pem` 中載入簽章者時發生問題。其可能是因新增簽章者的方式導致之檔案格式的問題。
- **SSL_ERROR_BAD_CERT**: 在 `buildForgeCert.pem` 中載入憑證時發生問題。可能是憑證與私密金鑰不符，或是在 PEM 中已毀損。
- **SSL_ERROR_BAD_KEY_FILE**: 指定給 `BFAgent.conf` 中之 `ssl_key_password` 內容的 `buildForgeKey.pem` 密碼發生問題。
- **SSL_ERROR_BAD_KEY**: `SSL_CTX_check_private_key` 傳回 1 以外的值。私密金鑰格式無效，或是與憑證不符。
- **SSL_ERROR_FIPS_ENABLEMENT**: 在 FIPS 啟用期間發生錯誤。這個錯誤通常是由於 FIPS 自我檢查期間發現的問題。這個錯誤可能是內部錯誤。
- **SSL_ERROR_INVALID_CIPHER**: 密碼規格與 OpenSSL 允許的不符。請檢查 `BFAgent.conf` 中的 `properties ssl_cipher_group` 或 `ssl_cipher_override` 指定的密碼。
- **SSL_ERROR_NO_CTX**: 建立新的 SSL CTX 物件時發生問題。這個錯誤可能是內部錯誤。

管理憑證

您可以在安裝之後修改 Build Forge 所使用的憑證和金鑰儲存庫。

在安裝期間，您有機會指定要使用的憑證（您自己的憑證或是由 Build Forge 產生的憑證）和金鑰儲存庫密碼。本節說明以下程序：

- 將 PEM 金鑰儲存庫轉換成 Build Forge 所需要的 OpenSSL 和 JSSE 金鑰儲存庫
- 變更金鑰儲存庫密碼，然後修改 Build Forge 來使用新的密碼
- 建立新的自簽憑證
- 使用用戶端憑證登入
- 下載未限定原則檔案，與強式加密憑證搭配使用

將 PEM 金鑰儲存庫轉換成 Build Forge 金鑰儲存庫：

從「憑證管理中心」接收的 PEM 金鑰儲存庫可以轉換成用於 Build Forge 的金鑰儲存庫。

開始之前

下載適用於您的 SDK 的無限制原則檔。只有當金鑰大小對限制的原則檔而言太大時，才適用這項必要條件。請從 https://www14.software.ibm.com/webapp/iwm/web/reg/signup.do?source=jcesdk&lang=en_US&S_PKG=142ww 下載檔案。

註：您必須使用 IBM 提供的 keytool 公用程式。

關於這項作業

如果您有一組來自「憑證管理中心」的 PEM 檔，您必須利用它們來建立一組 OpenSSL 和 JSSE 金鑰儲存庫，以用於 Build Forge。

程序

1. 將 Build Forge 工具目錄併入 PATH 中。

- `<bfinstall>/openssl`
- `<bfinstall>/ibmjdk/bin` (若為 Windows)
- `<bfinstall>server/ibmjdk/bin` (若為 UNIX 或 Linux)

如果是 UNIX 和 Linux，請將下列目錄併入 LD_LIBRARY_PATH 中：

`<bfinstall>/openssl`

2. 將 PEM 檔轉換成 PKCS12 金鑰儲存庫。

請使用下列指令：

```
openssl pkcs12
  -export
  -name "buildforge"
  -out buildForgeKeyStore.p12
  -inkey <key.pem>
  -passin pass:<pempassword>
  -in <crt.pem>
  -password pass:<bfpasword>
```

3. 確認憑證已經新增，且可以讀取。

```
keytool -v
  -list
  -keystore buildForgeKeyStore.p12
  -storepass <bfpasword>
  -storetype pkcs12
```

如果出現金鑰大小無效的錯誤，請下載無限制的原則檔。請使用這一節開頭的指示。

4. 匯出公開憑證。

在指令視窗中，跳至 `<bfinstall>/keystore`，然後執行以下指令：

```
keytool -export
  -alias buildforge
  -file cert.der
  -keystore buildForgeKeyStore.p12
  -storepass <bfpasword>
  -storetype pkcs12
```

- 該憑證會儲存在 `cert.der` 檔中。
- 使用在安裝期間指定給金鑰儲存庫的相同 `<bfpasword>`。否則，您需要變更配置。

5. 建立信任儲存庫並匯入公開憑證。

在指令視窗中，移至 `<bfinstall>/keystore`，然後執行以下指令：

```
keytool -import
        -noprompt -trustcacerts
        -alias buildforge
        -file cert.der
        -keystore buildForgeTrustStore.p12
        -storepass <bfpasword>
        -storetype pkcs12
```

6. 將公開用戶端憑證放在 `buildForgeCert.pem` 中。

在指令視窗中，跳至 `<bfinstall>/keystore`，然後執行以下指令：

```
openssl pkcs12 -clcerts -nokeys
        -in buildForgeKeyStore.p12
        -passin: pass:<bfpasword>
        -out buildForgeCert.pem
```

7. 將憑證和金鑰放在 `buildForgeKey.pem` 中

在指令視窗中，跳至 `<bfinstall>/keystore`，然後執行以下指令：

```
openssl pkcs12
        -in buildForgeKeyStore.p12
        -passin pass:<bfpasword>
        -passout pass:<bfpasword>
        -out buildForgeKey.pem
```

8. 建立「PEM 憑證管理中心」`buildForgeCA.pem`。

a. 將 CA 主要憑證下載至 `<bfinstall>/keystore`。其名稱爲 `CARootCert.crt`。其需要加入 PEM 金鑰儲存庫中，且可以匯入 `buildForgeTrustStore.p12` 中。

b. 在指令視窗中，跳至 `<bfinstall>/keystore`，然後執行以下指令：

```
cat CARootCert.crt > buildForgeCA.pem
keytool -import -noprompt -v -trustcacerts
        -alias "CA Root"
        -file CARootCert.crt
        -keystore buildForgeTrustStore.p12
        -storepass <bfpasword>
        -storetype pkcs12
```

結果

Build Forge 會使用受密碼保護的 PEM 金鑰儲存庫 `buildForgeKey.pem`。在啓動期間，Apache 伺服器會提示您輸入密碼。

如果您不要系統在啓動期間提示您輸入密碼，請產生不受密碼保護的 PEM 金鑰儲存庫，並供 Apache 伺服器使用。以下是範例指令。

```
openssl rsa -in buildForgeKey.pem
        -passin pass:<password>
        -out buildForgeKeyForApache.pem
```

請確定需要存取執行 Build Forge 之程序 ID 的任何使用者，可讀取不受保護的 PEM 金鑰儲存庫。

變更金鑰儲存庫密碼： 在安裝期間，您有機會指定金鑰儲存庫密碼。如果您要變更該密碼，則需要執行下列動作：

1. 修改 Build Forge 金鑰儲存庫以使用新的密碼。

2. 修改 Build Forge 配置以使用新的密碼。

修改金鑰儲存庫密碼：

關於這項作業

Build Forge 有三個受密碼保護的預設金鑰儲存庫，它們全都安裝在安裝了 Build Forge 引擎的主機上的 *<bfinstall>/keystore* 中：

- buildForgeKey.pem - 由 OpenSSL 使用，需要用 openssl 工具來變更密碼。
- buildForgeKeyStore.p12 - 由 JSSE 使用，需要用 ibmjdk 工具來變更密碼。
- buildForgeTrustStore.p12 - 由 JSSE 使用，需要用 ibmjdk 工具來變更密碼。

這些工具都隨附於 Build Forge 軟體中。

註：範例指令使用了換行以便於閱讀。請勿在指令中使用換行。請當成一個字串或使用行接續字元（Windows 為 ^，UNIX 或 Linux 為 \）來輸入。

重要：所有金鑰儲存庫都使用相同的密碼。其在範例中會顯示為 *newpassword*。

程序

1. 登入其中安裝 Build Forge 引擎的主機。
2. 將工具目錄放在 PATH 中。
 - *<bfinstall>/openssl*
 - *<bfinstall>/ibmjdk/bin*
3. 停用 SSL。在主控台中，移至**管理 > 安全**。將 **SSL 已啓用**設為「否」。
4. 按一下**儲存**。
5. 按一下**更新主要 BFClient.conf**。
6. 停止 Build Forge 引擎。
7. 備份現有的金鑰儲存庫。將現有的 Build Forge 金鑰儲存庫複製到暫存目錄。如果修改過的檔案毀損，您可以使用備份的金鑰儲存庫。
8. 修改 buildForgeKey.pem。在 *<bfinstall>/keystore* 目錄中，執行下列指令：

```
openssl rsa
-in buildForgeKey.pem
-passin pass:oldpassword
-out buildForgeKey.pem
-passout pass:newpassword -aes128
```
9. 修改 buildForgeKeyStore.p12。在 *<bfinstall>/keystore* 目錄中，執行下列指令：

```
keytool -storepasswd -all
-new newpassword
-keystore buildForgeKeyStore.p12
-storepass oldpassword
-storetype pkcs12
```
10. 修改 buildForgeTrustStore.p12。在 *<bfinstall>/keystore* 目錄中，執行下列指令：

```
keytool -storepasswd -all
-new newpassword
-keystore buildForgeTrustStore.p12
-storepass oldpassword
-storetype pkcs12
```

結果

變更密碼之後，您需要修改 Build Forge 配置以使用新的密碼。

修改 **Build Forge** 配置以使用新的密碼。

您必須變更 Build Forge 配置，才能使用變更過的金鑰儲存庫密碼。

開始之前

必備項目：

- 自您關閉 SSL 以來尚未啟動 Build Forge 引擎，已停止 Build Forge 引擎，並修改過金鑰儲存庫密碼。

關於這項作業

Apache Tomcat 應用程式伺服器在 `server.xml` 配置檔中包含金鑰儲存庫密碼。其是以明碼儲存。Apache Tomcat 在這項設定中不支援編碼或加密密碼。在這項程序中，您會在 Build Forge 主控台中修改 `server.xml` 和安全內容。

程序

1. 請在 Tomcat 配置中輸入新的密碼。編輯 `<bfinstall>/Apache/tomcat/conf/server.xml`。用於 SSL 的 Connector 陳述式就位在 `<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->` 註解下方。

```
<Connector port="8443" maxHttpHeaderSize="8192" algorithm="IbmX509"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="C:\Program Files\IBM\Build Forge\keystore\buildForgeTrustStore.p12"
keystorePass="newpassword"
keystoreType="PKCS12"
truststoreFile="C:\Program Files\IBM\Build Forge\keystore\buildForgeTrustStore.p12"
truststorePass="newpassword"
truststoreType="PKCS12"/>
```

2. 啟動 Build Forge。
3. 登入主控台。請使用 root 使用者或是具有「安全」存取角色的登入。
4. 輸入金鑰儲存庫的新密碼。在**管理 > 安全 > 金鑰儲存庫**中，編輯以下項目來使用「密碼」內容中的 `newpassword`。
 - 預設 JSSE 金鑰儲存庫
 - 預設 JSSE 信任儲存庫
 - 預設 OpenSSL 金鑰儲存庫
5. 啟用 SSL。
 - a. 在主控台中，移至**管理 > 安全**。
 - b. 將 **SSL 已啟用**設為「是」。
 - c. 按一下**儲存**。
6. 將變更匯出至 `BFClient.conf`。按一下**更新主要 BFClient.conf**。
7. 啟動 Build Forge 引擎。

建立新的自簽憑證：

請使用提供的工具來建立新的自簽憑證。

開始之前

您需要有安裝期間指定給金鑰儲存庫的密碼。如果您不知道密碼，請查看 `bfinstall/Apache/tomcat/conf/server.xml`。SSL/HTTP 連接器的 `keystorePass` 屬性包含此密碼。

關於這項作業

此程序說明如何取代在 Rational® Build Forge® 安裝期間自動建立的憑證。其會以下列內容建立憑證：

- 金鑰儲存庫：`buildForgeKeyStore.p12`
- 期限：15 年（設為 5475 天）
- 主體 DN：`CN=hostname`，其中 `hostname` 是完整主機名稱。

若要建立憑證，請使用 `openssl` 和 `ibmjdk` 工具。這些工具都隨附於 Rational® Build Forge® 軟體中。

需要五個金鑰儲存庫：

- `buildForgeKeyStore.p12` - 金鑰儲存庫，憑證與金鑰的儲存器
- `buildForgeTrustStore.p12` - 信任儲存庫，憑證與金鑰的儲存器
- `buildForgeKey.pem` - PEM 金鑰儲存庫
- `buildForgeCert.pem` - 公開憑證
- `buildForgeCA.pem` - PEM 憑證管理中心 (CA)

註：在範例指令中，使用了換行以便於閱讀。請勿在指令中使用換行。請當成一個字串或使用行接續字元（Windows 為 `^`，UNIX 或 Linux 為 `\`）來輸入。

重要：所有金鑰儲存庫都使用相同的密碼。其在範例中會顯示為 `password`。

程序

1. 登入其中安裝 Rational® Build Forge® 伺服器的主機。
2. 將工具目錄新增至下面所列的檔案路徑：
 - `<bfinstall>/openssl`
 - `<bfinstall>/ibmjdk/bin`
3. 將 `openssl` 目錄新增至 `LD_LIBRARY_PATH` 檔案路徑。例如，`<bfinstall>/openssl`。
4. 建立金鑰儲存庫 `buildForgeKeyStore.p12`、憑證和公開私密金鑰配對。
 - a. 在暫存目錄中，使用下列 Script 來建立金鑰儲存庫：

```
keytool -genkey -alias buildforge
-keyalg RSA -keysize 1024 -validity 5475 -dname "CN=hostname"
-keystore buildForgeKeyStore.p12
-storepass password
-storetype pkcs12
```
 - b. 將金鑰儲存庫檔 `buildForgeKeyStore.p12` 複製到 `<bfinstall>/keystore`。其會改寫現有的檔案。

5. 在 *<bfinstall>/keystore* 目錄中，輸入下列指令來匯出公開憑證：

```
keytool -export -alias buildforge
-file cert.der -keystore buildForgeKeyStore.p12
-storepass password
-storetype pkcs12
```

6. 建立信任儲存庫：

- a. 從暫存目錄中，執行 `keytool` 來建立信任儲存庫：

```
cd /temp
keytool -import -noprompt -trustcacerts -alias buildforge
-file cert.der -keystore buildForgeTrustStore.p12
-storepass password
-storetype pkcs12
```

- b. 將信任儲存庫檔 `buildForgeTrustStore.p12` 複製到 *<bfinstall>/keystore*。其會改寫現有的檔案。

7. 將公開用戶端憑證放在 `buildForgeCert.pem` 中。在 *<bfinstall>/keystore* 目錄中，執行下列指令：

```
openssl pkcs12 -clcerts -nokeys
-in buildForgeKeyStore.p12 -passin pass:password
-out buildForgeCert.pem
```

8. 將憑證和金鑰放在 `buildForgeKey.pem` 中在 *<bfinstall>/keystore* 目錄中，執行下列指令：

```
openssl pkcs12
-in buildForgeKeyStore.p12 -passin pass:password
-passout pass:password -out buildForgeKey.pem
```

9. 在 *<bfinstall>/keystore* 目錄中，執行下列指令來建立「PEM 憑證管理中心」`buildForgeCA.pem`。此指令會建立 `buildForgeKey.pem` 的副本。

- Windows：

```
copy buildForgeKey.pem buildForgeCA.pem
```

- UNIX 和 Linux：

```
cat buildForgeKey.pem > buildForgeCA.pem
```

下一步

`buildForgeKey.pem` 是受密碼保護的 PEM 金鑰儲存庫。在啟動期間，Apache 伺服器會提示您輸入密碼。如果您不要系統提示您輸入此密碼，請產生不受密碼保護的 PEM 金鑰儲存庫供 Apache 伺服器使用。

若要從私密金鑰移除密碼，請輸入下面的指令。確定 `buildForgeKeyForApache.pem` 檔可供需要存取執行 Build Forge 之程序 ID 的人員讀取。

```
openssl rsa -in buildForgeKey.pem -passin pass:password
-out buildForgeKeyForApache.pem
```

使用用戶端憑證登入：

配置使用用戶端憑證登入功能時，您有兩種選擇。

使用 **WebSphere Application Server** 來管理 **Build Forge** 的登入：

關於這項作業

使用 WebSphere Application Server 來管理 Build Forge 時，有一個選項可讓您稍微修改之後使用。此選項需要用到 WebSphereSSOInterceptor，其中包含 WebSphere 執行鑑別後的已鑑別主體。

利用此選項，在下列程序中進行修改。

程序

1. 保護 Build Forge WAR 檔案，這樣 WebSphere Application Server 儲存器管理鑑別就會鑑別 jas 的要求。如需如何設定此保護的相關資訊，請參閱第 97 頁的『使用自訂攔截程式來與 WebSphere Application Server 安全整合』。
2. 配置 WebSphere Application Server 以支援 jas Web 應用程式的用戶端憑證鑑別對映。若要設定此支援，請配置下列項目：
 - 用戶端憑證的 SSL
 - 用戶端憑證的 jas Web 應用程式
 - 用戶端憑證的 Web 伺服器
 - 用戶端憑證的 LDAP 伺服器對映

如需為 WebSphere Portal 配置用戶端憑證鑑別的相關資訊，以引導您配置 WebSphere Application Server，請參閱 <http://publib.boulder.ibm.com/infocenter/wpdoc/v6r0/index.jsp?topic=/com.ibm.wp.ent.doc/wpf/certauth.html>。遵循步驟 1 到 3。該程序參照您在第 97 頁的『使用自訂攔截程式來與 WebSphere Application Server 安全整合』中修改的 web.xml 檔案。您可修改位在 `${WAS_INSTALL_ROOT}/profiles/${PROFILE_NAME}/installedApps/${CELL_NAME}/jas_war.ear/jas_war/WEB-INF/web.xml` 的檔案。

根據自訂 Build Forge SSO 攔截程式登入：

關於這項作業

此選項需要您新增自訂 SSO 攔截程式，方法如第 88 頁的『關於單一登入架構』中所述。

程序

1. 請確定 SSO 攔截程式從要求屬性接收如下所示的 X509 憑證：

```
X509Certificate[]
certs = (X509Certificate[])request.getAttribute("javax.net.ssl.peer_certificates");
```
2. 請確定自訂 SSO 攔截程式將憑證主體 DN 之部分對映到 LDAP 屬性。通常您可以將憑證的通用名稱 (CN) 對映到 LDAP 目錄中的使用者 ID。在 Build Forge SSO 攔截程式的 `authenticateRequest` 方法中回覆結果。

```
responseAttributes.setStatus(HttpServletResponse.SC_OK);
principal = mapCert(certs); /* Custom method to map from certificate to
                             Build Forge LDAP principal. */
result = new Result(Result.UseridOnlyOID, domain, principal); /* Specify
the domain name of the LDAP server the principal exists in. */
return result;
```

下載未限定的原則檔案：

強式加密憑證需要下載 JCE 的未限定原則檔案

因為匯出和匯入規定，隨附於 Installation Manager 和 Build Forge 的預設 JCE 適用範圍原則檔案，只可讓您使用強式但有限的加密法。如果您想使用的憑證加密強度超過匯出允許強度，則必須下載未限定 JCE 原則檔案，並更新 Installation Manager 和 Rational Build Forge 所需的 JVM。

若當您嘗試提供要使用的憑證時，在安裝期間遇到下列錯誤，則可能需要更新 JCE 原則檔案。

無法驗證此憑證，安裝時若沒有它，請在安裝之後嘗試新增。

警告： Build Forge 升級包含「軟體開發套件 (SDK)」的更新，它可能會改寫原則檔案。在此情況下，請在升級後將未限定原則檔案重新套用至 SDK。

重要： 您所在的國家/地區可能會對加密軟體的進口、佔有、使用或轉口至其他國家/地區施加限制。在下載或使用未限定的原則檔案之前，您必須先查核您所在國家/地區的法令、規章，以及其對於加密軟體的輸入、佔有、使用或轉口等方面的政策，來判定是否允許這麼做。

1. 下載 1.4.2 以上版本 SDK 適用的未限定 JCE 原則檔案。
2. 解壓縮檔案。
3. 將解壓縮的檔案 local_policy.jar 複製到適當的 Build Forge 安裝目錄。改寫現有的檔案。使用您作業系統的目錄。

- Windows 系統

```
<bfinstall>\ibmjdk\jre\lib\security
```

- UNIX 及 Linux 系統

```
<bfinstall>/Platform/ibmjdk/jre/lib/security
```

4. 若要在將來安裝 Build Forge 時能夠驗證憑證，請更新 Installation Manager。請將解壓縮的檔案 local_policy.jar 複製到使用 Installation Manager 安裝的 IBM JRE。改寫現有的檔案。

- Windows 系統

```
<IInstall>\eclipse\jre_version\jre\lib\security
```

- UNIX 及 Linux 系統

```
<IInstall>/eclipse/jre_version/jre/lib/security
```

啓用密碼加密

將 Build Forge 系統配置成使用密碼加密，可以提高系統的安全性。

請務必強制使用實體安全，以避免未獲授權存取系統。

註：如果您要與 WebSphere 元件整合，請確定已符合密碼加密的必要條件，再配置密碼加密。

- 如需與 WebSphere Application Server 進行整合的相關資訊：請參閱第 509 頁的『使用 WebSphere Application Server 來取代 Apache Tomcat』。
- 關於整合 IBM HTTP Server (IHS)，請參閱第 513 頁的『使用 IBM HTTP Server 來取代 Apache HTTP Server』。

關於 Build Forge 中的密碼安全

依預設，Build Forge 系統會使用已編碼的密碼，但可以使用加密密碼以提供更高的安全性。

當您啓用密碼加密時，其會啓用爲對稱金鑰密碼架構。使用密碼的用戶端和被存取的服務，兩者必須使用相同的金鑰。

- Build Forge 引擎與 Build Forge 代理程式
- Build Forge 服務層和 Build Forge 所使用的資料庫

在 Build Forge 系統中，金鑰會保存在 `bfpwcrypt.conf` 檔中。該檔案位於 Build Forge（針對引擎）及代理程式的安裝目錄中。

密碼加密會使用對稱金鑰。所有需要解密一般資料庫密碼的系統，都需要相同的金鑰。此外，所有從引擎接收加密密碼的代理程式，都需要引擎的金鑰。如果有多個引擎在執行中（冗餘配置），則代理程式需要每一個引擎的金鑰。

針對一個 Build Forge「管理主控台」位於一部主機上，而一個代理程式位於另一部主機上的簡單安裝架構，啓用密碼加密需要下列程序：

- 在 Build Forge 主控台上啓用密碼加密（**管理 > 安全**）。
- 將現行金鑰匯出至檔案。（此金鑰可供代理程式使用，也可以供其他引擎使用。）
- 對代理程式產生新的金鑰。將它匯出至檔案。（這個金鑰是供代理程式用來加密其金鑰儲存庫密碼。）
- 以這兩個金鑰來更新代理程式的 `bfpwcrypt.conf`。請將新的金鑰放在最後。
- 在 Build Forge 主控台中，對使用該代理程式的所有伺服器定義啓用密碼加密。（**伺服器畫面**）
- 在 Build Forge 主機中，使用 `bfpwencrypt` 公用程式來加密 Build Forge 用來存取資料庫的密碼。請在 `buildforge.conf` 中將現行密碼（已編碼）取代爲加密密碼。
- 更新服務層的 `buildforge.conf` 副本。請參閱 第 83 頁的『Build Forge 配置檔 (`buildforge.conf`)』。

針對「管理主控台」啓用密碼加密

如果要在所有其他元件中啓用密碼加密，則必須在「管理主控台」中啓用密碼加密：

- 「冗餘管理主控台」需要從第一個「管理主控台」匯出要併入其 `bfpwcrypt.conf` 檔的密碼金鑰。所有「管理主控台」都必須使用相同的金鑰。一般而言，它們還必須使用該金鑰來加密資料庫密碼。
- 如果引擎會傳送加密密碼給代理程式，則代理程式需要匯出密碼金鑰。引擎和代理程式兩者應該使用相同的金鑰。從「管理主控台」匯出的金鑰可用來解密已加密的「伺服器鑑別」密碼。

如果代理程式未接收加密密碼，但需要產生加密密碼以用於其 `BFAgent.conf` 檔，則每一個代理程式都應該使用不同的密碼金鑰。請從「管理主控台」個別產生每一個金鑰。

如果要在「管理主控台」中啓用密碼加密，請執行下列動作：

1. 在主控台中，移至**管理 > 安全**。
2. 將**密碼加密已啓用**設爲「是」。

3. 按一下**儲存**。這個步驟會將配置儲存在 Build Forge 資料庫中。
4. 按一下**更新主要 BFCClient.conf**。這個步驟會將配置儲存在 Build Forge `bfcclient.conf` 檔中。
5. 重新啟動 Build Forge。您需要執行這個步驟，執行中的 Build Forge 程序才能使用 `bfcclient.conf` 檔中的新設定。

如果**密碼加密已啓用**為「是」且在配置中，您就能夠執行下列動作：

- 將密碼金鑰匯出至檔案
- 產生新的密碼金鑰
- 執行 `bfencrypt` 和 `bfagent` 指令，以建立要併入主控台和代理程式配置檔中的加密密碼

管理密碼金鑰

依預設，Build Forge 系統會使用已編碼的密碼，但可以使用加密密碼以提供更高的安全性。

密碼加密金鑰檔：

`bfencrypt.conf` 檔含有密碼加密配置內容。

第一次啟動 Build Forge 時，會自動在與 `bfcclient.conf` 檔相同的位置中產生 `bfencrypt.conf` 檔。

- Windows：<*bfinstall*>
- UNIX 或 Linux：<*bfinstall*>/Platform

重要：請勿重新命名這個檔案。其名稱必須一律為 `bfencrypt.conf`。

該檔案包含以下內容：

bfencrypt_key_alias

加密金鑰的別名。該別名是以這個金鑰加密之任何密碼的一部分。系統會用它來決定要使用的金鑰。這個內容可以有多个定義，每一個金鑰各一個。最後一個定義會用來加密密碼。所有其他定義都會在讀取時，用來解密已加密的密碼。您可以在配置檔、資料庫，或是代理程式與引擎之間的通訊中，找到加密密碼。

bfencrypt_key

已加密的主要金鑰，使用 128 位元的 AES 加密法加密。這個金鑰會用來加密密碼。

bfencrypt_key_password

解密 `bfencrypt_key` 所需的密碼。這個密碼已編碼。

範例 `bfencrypt.conf` 檔：

```
***** Password Encryption Configuration Properties *****
bf_encrypt_key=MKuoiwD+MsWBFgl/2xeG0TEtpY+hAzXQu21fBcofM0M=
bf_encrypt_key_alias=8a679d430c401000b55e00007d1a7d1a
bf_encrypt_key_password=Tq0eDXc4G/bdaWeatKTYUx6Sw4S3i6wX
# Creation_date=Thu Nov 20 03:44:48 CST 2008
# Origination host=myhost.mycompany.com
```


匯出密碼金鑰：

從 Build Forge 主控台匯出密碼金鑰，以便將它放在其他位置。

匯出用於 Build Forge 引擎的密碼金鑰，以便將它放在下列位置：

- 存取同一個資料庫的其他 Build Forge 主控台
- 管理在「伺服器」定義中所識別且已啓用密碼加密的代理程式之電腦

如果要匯出密碼金鑰，請執行下列動作：

1. 在主控台中，移至**管理 > 安全**。

註：密碼加密已啓用必須已設為「是」，且主要 BFCClient.conf 檔中的設定已儲存並更新。

2. 按一下**匯出金鑰檔**。系統會詢問您儲存檔案的位置。
3. 指定位置，然後按一下**儲存**。

將檔案放到需要金鑰來解密密碼（其他冗餘管理主控台、代理程式）或加密密碼（代理程式）的主機。請將檔案的內容（所有三個內容設定）新增至該主機上的 bfpwcrypt.conf 檔中。下表說明這些內容的放置位置如何決定金鑰的使用時間。

bfpwcrypt.conf 檔中的放置位置	在下列案例中使用的密碼金鑰
結尾	<ul style="list-style-type: none">• 解密包含其別名的密碼• 加密新密碼
其他任何位置	<ul style="list-style-type: none">• 解密包含其別名的密碼

註：當您產生新的金鑰時，另一個金鑰會放在 bfpwcrypt.conf 檔的尾端。當您匯出金鑰時，只會匯出最新的金鑰。

產生新的密碼金鑰： 您可以從「管理主控台」產生新的金鑰，以變更系統方面的金鑰。這項作業應該定期執行，以維持良好的系統安全性。新的金鑰必須在所有引擎和代理程式上加以更新。

產生新的密碼金鑰會新增一個金鑰到 bfpwcrypt.conf 檔中：

- 如果使用者或主控台密碼是利用舊的金鑰所產生或儲存，其可以繼續運作。
- **重要事項：**如果之前的金鑰已匯出並納入代理程式的 bfpwcrypt.conf 檔中，這時與代理程式的通訊會失敗，直到加入新產生的金鑰為止。
- 任何儲存在 Build Forge 中或是以 bfpwcrypt 產生的新密碼，都會使用新的金鑰。
- 匯出金鑰檔只會匯出最新的金鑰。

如果要產生新的密碼金鑰，請執行下列動作：

1. 在主控台中，移至**管理 > 安全**。

註：密碼加密已啓用必須已設為「是」，且主要 BFCClient.conf 檔中的設定已儲存並更新。

2. 按一下**產生新的金鑰**。系統會要求您確認。
3. 按一下**是**。

對代理程式啓用密碼加密

代理程式的密碼加密是在其配置檔中啓用。

如果要對代理程式啓用密碼加密，請執行下列動作：

1. 停止代理程式（如果其在執行中）。
2. 移至其中安裝代理程式的目錄。
3. 編輯 `bfagent.conf` 並開啓加密設定：

```
password_encrypt_module ./bfcrypt.dll;./bfwpcrypt.conf
```

如果代理程式是從安裝目錄以外的目錄啓動，請將這個路徑變更為直接參照檔案。

4. 取得「管理主控台」加密金鑰。將金鑰匯出至檔案。代理程式需要這個金鑰來解密已加密的「伺服器鑑別」密碼。其也會用來加密本端金鑰儲存庫密碼。
5. 將金鑰放在代理程式的 `bfwpcrypt.conf` 檔中。請將針對代理程式產生的金鑰放在檔案最後。當您手動加密密碼時，會使用檔案中的最後一個項目。
6. 啓動代理程式。
7. 在「管理主控台」中，移至**伺服器**。針對使用這個代理程式的每一個伺服器定義，將**密碼加密配置**內容設為「啓用」。
8. 按一下**測試連線**來確定連線可透過加密密碼來運作。

在 `buildforge.conf` 和 `bfagent.conf` 中加密密碼

請使用匯出的密碼金鑰來建置加密密碼，以用於 `buildforge.conf` 和 `bfagent.conf` 中。

`buildforge.conf` 檔包含 `Build Forge` 用來存取資料庫的使用者名稱和密碼 (`db_password`)。該密碼通常已編碼，但可以加密。如果要加密「管理主控台」的密碼，請執行下列動作：

1. 移至「管理主控台」根目錄。
 - Windows：`<bfinstall>`
 - UNIX 或 Linux：`<bfinstall>/Platform`
2. 執行下列指令：

```
bfwncrypt -e password
```

請對 `password` 使用您要加密的純文字密碼。

加密密碼會傳送至標準輸出。

`bfagent.conf` 檔含有代理程式用來存取金鑰儲存庫的密碼金鑰 (`ssl_key_password`)。該密碼通常是明碼，但可以加密。如果要加密代理程式的密碼，請執行下列動作：

1. 移至代理程式根目錄。
2. 執行下列指令：

```
bfagent -e password
```

請對 `password` 使用您要加密的純文字密碼。加密密碼會傳送至標準輸出。

註：如果您使用的是 AIX 且 GCC 程式庫不在 `/lib` 或 `/usr/lib` 中，您可能會收到錯誤指出「無法載入模組 `/usr/local/bin/bfcrypt.dll`」。您可以更新 `inetd.conf` 來解決這個問題。

若要更正錯誤，請執行下列動作：

- a. 請找出下面這一行：

```
bfagent stream tcp nowait root /usr/local/bin/bfagent bfagent
```

- b. 將這一行變更為下面這一行：

```
bfagent stream tcp nowait root /usr/bin/env env LIBPATH=path /usr/local/bin/bfagent
```

path 是 GCC 程式庫的位置。

- c. 請以 `refresh -s inetd` 指令重新載入 `inetd.conf`。

加密密碼會以字串 `bfcrypt:` 開頭，而密碼金鑰別名會括在大括弧中，後面接著已加密 (AES 128 位元) 並已編碼 (Base63) 的密碼。已編碼並加密的密碼範例：

Encoded:

```
dd8b42eed5cc051500f5bffe2b82b1aa6a67baee028a85d0cefa
```

Encrypted:

```
{bfcrypt:7427ab360c4010008f9d000049664966}drAIT1zLDGX/xRcvw65+B8aFpTqvmAdbmnh6FpwkHjU=
```

除錯密碼加密的問題

如果加密密碼有問題，以下各節說明除錯的方法。

在主控台中除錯密碼加密問題：

服務層、Web 介面和引擎會共用相同的金鑰檔。

當所有三個元件都安裝在同一部主機上時，其會使用相同的金鑰檔：

- Windows：

```
<bfinstall>/bfpwcrypt.conf
```

- UNIX 或 Linux：

```
<bfinstall>/Platform/bfpwcrypt.conf
```

如果啓用密碼加密之後發生問題，請檢查下列問題：

- 請確定在啓用密碼加密之後，已重新啓動 Build Forge[®]。請確定所有程序都已適當地停止並重新啓動 (Apache、Apache Tomcat、引擎)。
- 冗餘主控台：如果您有多個使用相同資料庫的管理主控台安裝架構，它們必須全都使用相同的 `bfpwcrypt.conf` 檔。最安全的方法是手動分送該檔案，而非透過網路分送。
- 伺服器定義：如果主控台內的**測試連線**失敗，請確定金鑰已正確地匯出，並放在 `bfagent.conf` 檔中。爲了確定這是密碼問題，請停用密碼加密，然後嘗試**測試連線**。
- 登入：如果在啓用密碼加密之後無法登入，請確定 Build Forge[®] 在 `bfclient.conf` 和 `buildforge.conf` 中都使用正確的 `bfpwcrypt.conf` 金鑰。您必須在 `<bfinstall>` 目錄以及它的服務層副本中更新 `buildforge.conf`。請參閱 第 83 頁的『Build Forge 配置檔 (`buildforge.conf`)』。

如果做完所有這些檢查但問題仍然存在，請嘗試啓用追蹤並檢查輸出日誌。

- Web 介面 (使用者介面)：設定環境變數 `BFDEBUG_SECURITY=1`。

Web 介面：輸出會顯示在檔案中。

- Windows：

```
<bfinstall>/Apache/logs/php_error.log
```

– UNIX 或 Linux :

```
<bfinstall>/server/apache/logs/php_error.log
```

- 引擎：在前景中啟動引擎。在安裝目錄中，執行 `bfengine -d`。在 UNIX 或 Linux 中，您可以使用 `bfengine -d 2>&1 | tee out.txt`，將此項目輸出到檔案中。在 Windows 上，如果您取得 `tee` 公用程式，則可以執行相同的作業。
- 服務：執行下列動作：
 1. 停止 Build Forge。
 2. 在編輯器中開啟日誌檔。
 - Windows :

```
<bfinstall>/Apache/tomcat/common/classes/logging.properties
```
 - UNIX 或 Linux :

```
<bfinstall>/server/apache/tomcat/common/classes/logging.properties
```
 3. 將下面這一行加到檔案結尾。

```
com.buildforge.services.common.security.level=ALL
```
 4. 啟動 Build Forge®。
 5. 視察輸出。
 - Windows :

```
<bfinstall>/Apache/tomcat/logs/catalina.out
```
 - UNIX 或 Linux :

```
<bfinstall>/server/apache/tomcat/logs/catalina.out
```

在代理程式中除錯密碼加密問題：

除錯代理程式通訊包含代理程式和引擎元件。

如果啟用密碼加密之後發生問題，請檢查下列問題：

- 檢查 `bfagent.conf`。您應該解除下列行的註解：

```
password_encrypt_module ./bfcrypt.dll;./bfpcrypt.conf
```
- 檢查 `bfpcrypt.conf` 是否存在於用來啟動代理程式的目錄中。其必須至少包含一個金鑰項目。
- 檢查引擎 `bfpcrypt.conf` 中的最後一個項目，存在於代理程式 `bfpcrypt.conf` 中的某處。

如果做完所有這些檢查但問題仍然存在，請嘗試啟用追蹤並檢查輸出日誌。如果要啟用追蹤，請執行下列動作：

1. 開啟 `bfagent.conf`。
2. 解除下面這一行的註解：

```
activity_log bfagent.log
```

您可以指定另一個路徑而非 `bfagent.log`。

bfcrypt.dll 的路徑問題

在 bfaagent.conf 中，password_encrypt_module 內容必須指向 bfcrypt.dll 的正確路徑。範例：

```
password_encrypt_module /opt/buildforge/bfcrypt.dll
```

如果已開啓追蹤，指出這個路徑問題的輸出類似如下所示：

```
[ 8928] bfcryptloader.c : 208: Loading password encryption module.
[ 8928] bfcryptloader.c : 223: module: bfcrypt
[ 8928] bfcryptloader.c : 232: Loading module: C:/BuildForge71.181.Agent/bfcrypt.dll
[ 8928] bfcryptloader.c : 262: Failed loading DLL, error code = 0
```

成功載入會產生如下所示的輸出：

```
[ 12248] bfpwdlocloader.c: 134: Looking for password locator: ssl_key_password_locator
[ 12248] bfpwdlocloader.c: 244: Looking for password for prop ssl_key_password from bfaagent.conf.
[ 12248] bfcryptloader.c : 208: Loading password encryption module.
[ 12248] bfcryptloader.c : 223: module: bfcrypt
[ 12248] bfcryptloader.c : 232: Loading module: ./bfcrypt.dll
[ 12248] bfcryptloader.c : 269: Loading procedure bfcrypt_init.
```

密碼解密失敗

當密碼因金鑰錯誤或一些其他原因而無法解密時，日誌會含有如下所示的字行：

```
[ 4912] agent.c : 237: AUTH failed
```

如果您確定密碼正確，您可以進一步診斷問題。請啓用 bfcrypt.dll 模組的除錯功能。如果要啓用除錯，請設定以下環境變數：

```
BFDEBUG_SECURITY=1
```

如果代理程式是當成服務執行，則需要廣域設定該變數。

除錯輸出會放在用來啓動代理程式之目錄下的 bfcrypt.txt 中。

下列輸出指出正確的金鑰不在代理程式的 bfpwcrypt.conf 中：

```
load_keys_from_file: Parsed 1 key configurations.decrypt:
Looking for key matching info: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Comparing against: 4d553f110c401000ac08000051f651f6, length=32
decrypt: Warning! No matching key found.
```

下列輸出指出正確的相符金鑰：

```
load_keys_from_file: Parsed 2 key configurations.decrypt:
Looking for key matching info: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Comparing against: 7427ab360c4010008f9d000049664966, length=32
decrypt: Comparing against: 922492fe0c4010008304c3670e1e0e1e, length=32
decrypt: Found match: 922492fe0c4010008304c3670e1e0e1e, length=32
```

bfaclient.conf 參照

bfaclient.conf 檔會儲存 Build Forge 的相關安全設定。該檔案位於 Build Forge 安裝根目錄中。

bfaclient.conf 檔含有用來啓用安全通訊 (SSL) 和密碼加密的設定。其含有以下區段：

- 連線內容
- 登入內容

- OpenSSL 和 JSSE 兩者使用的 SSL 內容
- 僅用於 OpenSSL 的 SSL 內容
- 僅用於 JSSE 的 SSL 內容
- 金鑰儲存庫內容 (bf_keystore_*)
- 加密內容

表 1. 連線內容

屬性名稱	預設值	可能值	必要	說明
bf_services_hostname	在安裝期間指定	不適用	是	Build Forge 服務層所在的主機名稱
bf_services_tcp_port	在安裝期間指定	不適用	是	用來連接 Build Forge 服務的 TCP 埠。如果沒有指定 SSL，就會使用這個項目。
bf_services_ssl_port	在安裝期間指定	不適用	是	用來安全連接 BuildForge 服務的 SSL 埠
bf_services_preferred_protocol	tcp	ssl、tcp	是	如果是 Perl 或 PHP 用戶端，指定 SSL 或 TCP 來建立連線。如果是 Java 用戶端，SecureAPIClientConnection 物件會指定 SSL，而 APIClientConnection 則指定 TCP。

表 2. 登入內容

屬性名稱	預設值	可能值	必要	說明
bf_login_user	無	Build Forge 使用者清單中的使用者 ID	否	用來作為登入 ID。登入 ID 也可以指定在用戶端程式內。
bf_login_password	無	bf_login_user 的密碼	是（如果使用 bf_login_user）	bf_login_user 的密碼
bf_login_realm	無	LDAP 網域名稱	否	要查詢的 LDAP 網域（如果使用者尚未在「使用者」表格中）。

表 3. openssl 和 JSSE 兩者使用的 SSL 內容

屬性名稱	預設值	可能值	必要	說明
bf_ssl_usage	無	jsse、openssl	是	選取 SSL 實作。根據該選擇，有不同的內容可用。
bf_ssl_cipher_group	ALL	ALL、HIGH、MEDIUM、LOW	否	指定要在 SSL 信號交換期間提供的該組密碼。HIGH 是最安全，LOW 的效能最佳，ALL 則是最能夠進行交互作業的。
bf_ssl_cipher_override	無	您提供的密碼套組	否	置換 bf_ssl_cipher_group。可以用來選擇在 SSL 信號交換期間要使用的一組較少的密碼。
bf_ssl_protocol	TLSv1	TLSv1、SSLv3。會因實作而有所不同。	否	SSL 所使用的信號交換通訊協定。TLSv1 是偏好的通訊協定。
bf_ssl_cert_alias	無	在配置的金鑰儲存庫中有效的憑證別名	否	指定要使用的憑證。如果同一個金鑰儲存庫中有多個憑證，則可能使用這個內容。

表 4. 僅用於 *openssl* 的 *SSL* 內容

屬性名稱	預設值	可能值	必要	說明
bf_ssl_key_ref	openssl_key	含有私密金鑰的任何有效的 PEM 金鑰儲存庫參照	否	含有在連接伺服器時，供用戶端使用之私密金鑰的金鑰儲存庫配置的參照。如果使用，您也必須在 bf_ssl_cert_ref 中指定這個私密金鑰的有效憑證。請只在設定伺服器來要求個人憑證時，才使用這個內容。
bf_ssl_cert_ref	openssl_cert	含有指定私密金鑰之憑證的任何有效 PEM 金鑰儲存庫參照	否	含有上述私密金鑰之憑證的金鑰儲存庫配置的參照。請只在設定伺服器來要求個人憑證時，才使用這個內容。
bf_ssl_ca_ref	openssl_ca	含有一個以上的憑證（用來驗證這個用戶端連接的伺服器憑證）的任何有效 PEM 金鑰儲存庫參照	是	含有一個以上的簽章者憑證（用來在 SSL 信號交換期間驗證伺服器憑證）之金鑰儲存庫配置的參照。該憑證可以是 CA 主要憑證、中繼憑證或自簽憑證。

表 5. 僅用於 *JSSE* 的 *SSL* 內容

屬性名稱	預設值	可能值	必要	說明
bf_ssl_keystore_ref	jsse_keystore	含有 keyEntry（私密金鑰和憑證）的任何有效 PKCS12、JKS 或 JCEKS 金鑰儲存庫參照	否	含有在連接伺服器時，供用戶端使用之個人憑證（私密金鑰和關聯的憑證）的金鑰儲存庫配置的參照。只有當伺服器要求個人憑證來進行用戶端鑑別時，才需要這個內容。
bf_ssl_truststore_ref	jsse_truststore	含有 keyEntry（私密金鑰和憑證）的任何有效 PKCS12、JKS 或 JCEKS 金鑰儲存庫參照	是	含有簽章者憑證（用來在 SSL 信號交換期間驗證伺服器憑證）之金鑰儲存庫配置的參照。金鑰儲存庫含有一個以上的 trustedCertEntries，這是用來驗證其他憑證簽章的憑證。

表 6. 金鑰儲存庫內容

屬性名稱	預設值	可能值	必要	說明
bf_keystore_alias	各種不同的值	字串	是	這是 SSL 配置用來參照金鑰儲存庫配置的名稱。
bf_keystore_location	各種不同的值	指定類型之金鑰儲存庫的相對或完整路徑	是	這是指定類型之金鑰儲存庫的路徑和位置。此路徑可以是相對路徑，但是對相關的起始目錄而言必須是正確的。
bf_keystore_type	PEM 用於 openssl，PKCS12 用於 jsse	PEM 用於 openssl。PKCS12、JCEKS，或是 JKS 用於“jsse”	是	金鑰儲存庫的類型。必須符合 bf_keystore_location 內容所參照的實際金鑰儲存庫類型。
bf_keystore_password	在安裝期間指定	金鑰儲存庫類型所支援的字串。某些金鑰儲存庫不支援非 ASCII 字串。	否	用來存取金鑰儲存庫的密碼。如果是 OpenSSL，只含公開金鑰的 Cert 或 CA 金鑰儲存庫通常不需要密碼。

表 7. 加密內容

屬性名稱	預設值	可能值	必要	說明
bf_pw_crypt_enabled	false	true、false	否	指定密碼是已編碼 (false) 或已加密 (true)。如果啓用，密碼加密實作會使用名稱爲 bfpwcrypt.conf 的檔案（其與 bfclient.conf 位於相同的目錄中）。

第 10 章 安裝代理程式

本節說明如何安裝、執行、配置及疑難排解代理程式。

請在要用來作為「管理主控台」之伺服器資源的每一部主機上，安裝代理程式。代理程式是一種服務，其可接收來自「管理主控台」的要求以執行專案與步驟。

在 Windows 平台上安裝代理程式

如果要在 Windows 平台上安裝代理程式：

1. 在安裝媒體中找出並啟動代理程式安裝程式。安裝程式的檔名是 `win-bfagent-version.exe`。

提示：當您選擇 **Rational Build Forge 代理程式安裝**時，「啟動程式」會開始這個安裝程序。

2. 如果安裝程式偵測到現有的代理程式版本，其會提示您確認是否要加以改寫。按一下**確定**。預設值：確定。
3. 開啓「歡迎使用」訊息之後，按**下一步**。
4. 如果您同意授權合約的條款，請按一下**我同意**。
5. 在「選擇安裝位置」視窗中，設定**目的地資料夾**，然後按**下一步**。請使用預設位置 `C:\Program Files\IBM\Build Forge\Agent`，以便能夠輕易找到檔案。
6. 在「配置」視窗中，選擇所需的**代理程式選項**，然後按一下**安裝**。
7. 選取下列其中一個安裝方法：
 - **安裝為服務**
 - **安裝使用者模式代理程式**

只有在代理程式必須能夠執行 GUI 應用程式時，才選取使用者模式代理程式。

註：當 Build Forge 代理程式是安裝在 Windows 7 上，且選取**使用者模式**，則需要重新開機。

8. 選用：按一下啓用 **Cygwin 支援**

提示：如果您使用 Cygwin Linux[®] 模擬環境，則當您安裝代理程式時，可以選擇安裝 Cygwin 支援。如果您安裝 Cygwin 支援，請執行下列步驟。

- a. 在 Cygwin 安裝期間，選擇 **DOS/text** 行結尾。
- b. 在專案中，針對指令使用 UNIX[®] 樣式的語法。

重要：

Cygwin 只能使用 US ASCII。它不支援 UTF-8，因此無法在其他系統中使用。

9. 指定代理程式用來與「管理主控台」通訊的**埠**。預設埠為 5555。
10. 在「安裝完成中」畫面，按一下**完成**。

註：請勿在安裝期間關閉任何蹦現視窗。在執行安裝時，讓它們自行出現及消失。

使用者模式代理程式

安裝為「使用者模式」的代理程式可讓使用者與專案所啟動的應用程式進行互動。

使用者模式是只供 Windows 代理程式使用的選項。此選項可在安裝期間與以設定，安裝後就無法進行配置。使用者模式擁有下列應用程式：

- 在執行工作時，透過 GUI 應用程式來手動收集輸入。這會讓工作受到人為輸入的影響。
- 對專案和步驟進行疑難排解。使用者模式可以看到在服務模式作業期間所隱藏的輸出內容，每個步驟都會在執行時產生一個主控台視窗。

重要：Windows 7 系統。如果您在 Windows 7 系統上安裝或重新安裝使用者模式代理程式，則必須將作業系統重新開機。除非將作業系統重新開機，否則代理程式將無法運作。

當您要設定專案以便在使用者模式中使用電腦時，請務必記住下列差異：

- 使用者模式代理程式會以系統上的目前已登入使用者身分來運作。只有在此使用者登入電腦時，代理程式才會作用中。如果使用者已登出，就無法使用執行使用者模式代理程式的伺服器。
- 電腦上的所有使用者，都能看到電腦在使用者模式下執行的步驟。
 - 每個步驟都會在執行代理程式的 Windows 電腦上開啓一個主控台視窗。其中顯示來自「管理主控台」的指令活動。
 - 如果您在某個步驟中啟動了 GUI 應用程式，應用程式視窗便會出現在執行代理程式的 Windows 電腦上。「管理主控台」則會等到該應用程式結束後才繼續下一步驟。

替代方案：如果您要繼續執行工作而不要等待，請使用 `start` 指令。

- 不要使用 `_USE_BFCREDS` 變數，使用此變數的任何步驟都會失敗。
- 使用者必須具備下列專用權。依預設，通常不會提供它們。必須明確新增。

```
SeInteractiveLogonRight
SeAssignPrimaryTokenPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeTcbPrivilege
```

在 Windows 作業系統上執行代理程式的無聲自動安裝

如果要在 Windows 上執行代理程式的自動無聲安裝，請使用 `/S`（大寫 S）選項。比方說，在命令提示字元中輸入下列指令。該選項會區分大小寫：

```
win-bfagent-7.0.1.2305.exe /S
```

無聲自動安裝會使用下列設定，且無法與以修改。

- 改寫現有的安裝：`yes`
- 安裝位置：`C:\Program Files\IBM\Build Forge\Agent`
- 安裝為服務：`yes`
- Cygwin 支援：`no`
- 埠：`5555`

將代理程式安裝在 UNIX 和 Linux 系統上

請遵循適用於您平台的安裝指示：

• AIX

1. 使用 `aix5-bfagent-<version>.tar.gz` 檔或 `aix5np-bfagent-<version>.tar.gz` 檔。

`aix5np` 檔不包含 PAM 鑑別支援。

重要：

如果您將 `aixnp` 代理程式安裝成以 `root` 身分執行，在執行時期，代理程式會使用 AIX 鑑別呼叫，利用您指定的「伺服器鑑別」認證來進行鑑別。如果您未將代理程式安裝成以 `root` 身分執行，您也必須在 `bfagent.conf` 檔中使用 `magic_login` 設定以限制其存取權。

2. 輸入以下指令來解壓縮檔案：

```
gzip -d gzipfilename.gz
```

3. 輸入以下指令來解壓縮檔案：

```
tar xvf tarfilename.tar
```

4. 輸入以下指令來安裝代理程式：

```
cd extracted-agent-directory
./install.sh
```

重要：如果 AIX 的代理程式是使用 `- without-pam` 配置選項來編譯，則會關閉該代理程式的鑑別。

如果以 `root` 專用權來安裝它，則可讓人員以任何有效的使用者身分來連接，而不會考慮他們指定的密碼。

如果您必須編譯代理程式，以便在不使用 PAM 的 AIX 系統上執行，請確保使用專用的代理程式帳戶，將代理程式安裝成以該使用者身分執行，並在 `bfagent.conf` 檔中使用 `magic_login` 設定以限制其存取權。

• HP-UX

1. 確定 Rational® Build Forge® 代理程式可以找到您在專案中指定的所有作業系統指令。您可以使用下列其中一個選項來完成這項作業：

- 編輯系統的 `/etc/PATH` 檔來併入所有需要的目錄。

- 在 HP-UX 系統中，於 `bfagent.conf` 檔中指定 `nologonshell` 設定。這項設定會使 Build Forge® 不處理 `/etc/profile`。接著，您必須在 Build Forge® 環境下的 `/etc/profile` 中，指定 `PATH` 和任何其他必要設定的值，這些值會套用到每一個專案。

2. 從安裝媒體中取得 `hpux11-bfagent-<version>.tar.gz` 檔。將其放在您想要安裝代理程式的位置。

3. 解壓縮檔案。

```
gzip -d gzipfilename.gz
```

4. 從 `tar` 檔解壓縮。

```
tar xvf tarfilename.tar
```

5. 安裝代理程式。

```
cd extracted-agent-directory
./install.sh
```

6. 修改 `/etc/profile` 中的下面這一行，使代理程式能夠在非互動式登入 Shell 中執行指令。

```
if [ ! $VUE]
```

將這一行變更如下所示：

```
if [ -z "$VUE" -a -n "$PS1" ]
```

- **Mac OS**

1. 從安裝媒體中取得 `mac-bfagent-<version>.dmg` 檔，並將它放在所需的位置。
2. 按兩下檔案以解壓縮其所含元件。

- **Red Hat Linux 和 SuSE Linux**

1. 從安裝媒體中取得 `rh9-bfagent-<version>.rpm` (Red Hat Enterprise Linux 4 或 SuSE Linux 10) 或 `rhel5-bfagent-<version>.rpm` 檔 (Red Hat Enterprise Linux 5)
2. 請使用 RPM 公用程式來安裝檔案。

```
rpm -iUvh rh9-bfagent-version.rpm
```

- **Solaris**

1. 對於 SPARC 上的 Solaris 9 或 Solaris 10，使用 `sol9-bfagent-<version>-sparc-opt.gz` 檔。
2. 將套件解壓縮：

```
gzip -d solN-bfagent-<version>-platform-opt.gz
```

3. 在命令提示字元中，輸入以下指令：

```
pkgadd -d ./unzipped-package
```

- **其他平台 - 從原始檔進行編譯**

如果您需要適用於其他平台的代理程式，請使用 `src-bfagent-<version>.tar.gz` 檔，從原始檔編譯代理程式：

1. 解壓縮已下載的 `tar` 檔。
2. 執行位於 `src` 目錄的配置 Script。

若要針對 SSL 進行配置，請使用下列選項：

```
./configure --with-ssl=/usr/include/openssl
```

若要針對 SSL 和密碼加密進行配置，選項如下所示：

```
./configure --with-ssl=/usr/include/openssl --enable-bfcrypt-dll
```

3. 在 `src` 目錄中執行 **make** 指令。

原始檔套件需要使用適用於您系統的 GNU C 編譯器或 C 編譯器。原始檔套件和預先建置的代理程式（其不含用於本端電腦的安裝程式）包含一個安裝程式，可用來安裝電腦之系統 `inetd/xinetd` 配置檔中的代理程式。

在 System i 平台上安裝代理程式

請使用下列指示，在 System i® 上手動安裝代理程式。

步驟 7 的指令 Script 會建立將在啟動時執行的工作說明，並以具有 *ALLOBJ 特殊權限的 BFAGENT 使用者身分來起始代理程式。

- 任何具有 *ALLOBJ 特殊權限的使用者或是 QSECOFR 使用者，都可以利用您在「管理主控台」中指定的伺服器鑑別認證來鑑別。
- 如果要鑑別不具這些專用權的使用者，您必須在 bfaagent.conf 檔中配置 magic_login 設定。請參閱 第 159 頁的『bfaagent.conf 參照』以取得詳細資料。

如果要在 System i 平台上安裝代理程式：

1. 使用產品安裝媒體或下載映像檔，找出 iseries-bfaagent-<version>.tar.gz 檔。
2. 輸入以下指令，從保存檔中解壓縮 tar 檔：


```
gzip -d iseries-bfaagent-<version>.tar.gz
```
3. 從 tar 檔解壓縮檔案。


```
tar xvf iseries-bfaagent-<version>.tar
```
4. 在 iSeries® 伺服器中，將 bfaagent 執行檔放在代理程式安裝目錄中，例如：/bin。
5. 在 iSeries 伺服器中，將 bfaagent.conf 檔放在 /etc 中。
6. 在 bfaagent.conf 檔中，解除 Shell 選項的註解，並指定用於 PASE 的預設 Shell（如下例所示），或指定您偏好的 Shell。


```
shell /bin/sh
```
7. 配置 System i 於啟動時以 BFAAGENT 使用者的身分執行。

請輸入下列指令來建立具有 *ALLOBJ 特殊權限的 BFAAGENT 使用者，並建立在啟動時以 BFAAGENT 使用者身分執行的工作說明。在下例中，bfaagent 執行檔安裝在 /bin 中。

CRTLIB BFAAGENT

CRTSBSD SBSD(BFAAGENT/BFAAGENT) POOLS((1 *BASE)) TEXT('Build Forge Agent subsystem')

CRTJOBQ JOBQ(BFAAGENT/BFAJOBQ) TEXT('Build Forge Agent job queue')

CRTUSRPRF USRPRF(BFAAGENT) PASSWORD(*NONE) INLMNU(*SIGNOFF) LMTCPB(*YES)
SPCAUT(*ALLOBJ) TEXT('Build Forge Agent user profile')

CRTJOBQ JOBQ(BFAAGENT/BFAJOBQ) JOBQ(BFAAGENT/BFAJOBQ)
TEXT('Build Forge Agent autostart')USER(BFAAGENT) RQSDTA('CALL PGM(QP2SHELL)
PARM('/bin/bfaagent' ' ' -s'))

CRTCLS CLS(BFAAGENT/BFACLS) TEXT('Build Forge Agent job class')

ADDRTGE SBSD(BFAAGENT/BFAAGENT) SEQNBR(1) CMPVAL(*ANY) PGM(QCMD) CLS(BFAAGENT/BFACLS)

ADDJOBQE SBSD(BFAAGENT/BFAAGENT) JOBQ(BFAAGENT/BFAJOBQ) MAXACT(*NOMAX) SEQNBR(10)

ADDAJE SBSD(BFAAGENT/BFAAGENT) JOB(BFAAGENT) JOBQ(BFAAGENT/BFAJOBQ)

在 System z 平台上安裝及執行代理程式

請遵循以下指示，在 System z 上手動解壓縮並編譯 Build Forge 代理程式原始碼。所提供的 z/OS® 代理程式原始碼只是未編譯的原始碼。無法使用二進位碼發行套件。

所需的軟體與程式如下所示：

- c89 編譯器和 UNIX 標頭檔。在 z/OS 系統上，代理程式會在「UNIX 系統服務 (USS)」環境中執行。

- z/OS UNIX shell 介面。在安裝期間，您是在 z/OS 上的 z/OS UNIX shell 中執行所有的指令。
- gzip 公用程式。

註：如果 z/OS 系統上有 gzip 可用，您可以在將原始檔套件傳送至 z/OS 之後，再於 z/OS 系統上解壓縮 tar 檔。若非如此，則必須先在非電腦上將檔案解壓縮，然後再將它們傳送至 z/OS 系統。

- Build Forge 代理程式原始檔套件（z/OS版）：src-bfagent-<version>.tar.gz.

必備條件

- 對用戶端和內部通訊啟用 SSL。請參閱第 104 頁的『對用戶端和內部連線啟用 SSL』。
- 必須安裝系統 SSL 程式庫。
- 當您在編譯代理程式的程式碼時，請使用最新版的 GSKit。

如果要在 System z 平台上安裝代理程式，請執行下列動作：

1. 使用產品安裝媒體或下載產品映像檔，找出代理程式原始檔套件的檔案：src-bfagent-<version>.tar.gz。

將原始檔套件複製或下載到非 z/OS 電腦上的目錄中。

2. 在非 z/OS 電腦上的 Shell 提示中，輸入下列指令以從代理程式原始檔套件中解壓縮 tar 檔：

```
gzip -d src-bfagent-<version>.tar.gz
```

3. 使用 ftp 或另一種傳送方法，將 tar 檔當成二進位映像檔傳送到 z/OS 系統，並將其放在專用的 HFS 子目錄中（通常是使用者帳戶的 USS 起始目錄）。
4. 在 z/OS 系統上，執行下列指令來建置代理程式原始碼：

- a. 首要步驟：./configure-zos。請注意 -with-system-ssl 的用法。您必須提供系統 SSL 的路徑。使用最新版的 GSKit：

- 從 BuildForge 8.0.0.1 開始，bfagent 支援 TLS v1.2 及 TLS v1.1
- 從 z/OS 1.13 版開始，支援 TLS v1.2。若您無法滿足版本下限需求，則必須在不包含系統 SSL 的 z/OS 上編譯 bfagent。

```
pax -rf src-bfagent-<version>.tar -ofrom=ISO8859-1,to=IBM-1047
cd bfagent-<version>/src
./configure-zos --with-system-ssl=<path>
```

- b. 完成 ./configure-zos Script 之後，請執行下列指令：

```
./build-zos
```

註：如果您在執行 ./build-zos Script 之後收到錯誤，請參閱第 137 頁的『在 z/OS 上對代理程式安裝作業進行疑難排解』。

5. 在 z/OS 系統上，將 bfagent.conf 檔放在 /etc 中。

如果 bfagent.conf 不在 /etc 中，代理程式必須以 -f 選項啟動。請參閱 第 156 頁的『bfagent 參照』。

6. 在 z/OS 系統上，將 bfagent 執行檔放在適當的位置，例如：/usr/bin or /usr/local/bin。
7. 在 z/OS 系統上，以 root 身分執行下列指令：


```
# extattr +p -s bfagent
```

8. 如果您要使用系統 SSL，請將系統 SSL 程式庫新增至 PROGRAM CONTROL。請使用下列指令。

```
ralter program * addmem('SYS1.SIEALNKE'//nopadchk) uacc(read)
```

9. 在 z/OS 系統上，以 root 身分登入，然後利用 -s 選項來手動啟動代理程式：
bfagent -s

如果安全原則不允許以 root 身分登入，請參閱 第 159 頁的『bfagent.conf 參照』，查看 bfagent.conf 中之 magic_login 設定的指示。

代理程式會當成獨立式常駐程式來執行，並使用預設代理程式埠 5555。如果要變更預設埠，請使用 bfagent.conf 中的埠設定。請參閱 第 156 頁的『bfagent 參照』。

註：如果已安裝 UNIX TCP/IP 常駐程式 (inetd 或 xinetd) 且在 z/OS 系統上作用中，您可以將 Build Forge 代理程式設定當成服務來執行並自動啟動。請參閱第 154 頁的『在 UNIX、Linux 及 MacOS 上執行代理程式』。

10. 在 z/OS 系統上，使用 telnet 指令來測試連線。請參閱第 169 頁的『測試連線』。

在 z/OS 上對代理程式安裝作業進行疑難排解

可能會在 z/OS 上建置代理程式原始碼之後，收到錯誤訊息。這個主題說明一些常見錯誤的修正程式。

configure-zos Script 會設定一些通用值，並執行一些基本檢查來識別可用於系統的標頭和函數。

由於 z/OS 系統配置的差異，雖然 ./configure-zos Script 可能執行無誤，但在您執行 ./build-zos Script 時，可能會看到下列錯誤。

CEE3501S The module CCNDVR was not found.

FSUM3066 「編譯」步驟結束時產生下列回覆碼：

```
-1: EDC5083I An error occurred attempting to load a module into storage.
```

這項錯誤指出編譯器無法載入所需的動態程式庫。

執行指令：`% export STEPLIB="SYS1.SCCNCMP"`

重新執行 ./build-zos 指令。如果指令再度失敗，請洽詢系統管理者以取得協助來找出所需的程式庫。

IKJ56228I DATA SET CEE.SCEE0BJ NOT IN CATALOG OR CATALOG CAN NOT BE ACCESSED

FSUM3066 「編譯」步驟結束時產生下列回覆碼：

FSUM3052 無法解析資料定義名稱 C8961。找不到資料集。請確定已正確指定資料集名稱 CEE.SCEE0BJ。

這項錯誤指出鏈結器找不到完成編譯所需的系統檔案庫。請執行指令：

```
% export _C89_LSYSLIB=SYS1.SCEELKED:SYS1.SCEELKEX
```

```
% export _C89_PSYSLIB=SYS1.SCEE0BJ
```

重新執行 ./build-zos 指令。如果指令再度失敗，請洽詢系統管理者以取得協助來找出所需的程式庫。

IEW2456E 9207 SYMBOL xxx UNRESOLVED

無法解析的符號錯誤指出，建置所預期的符號（由系統 C 程式庫定義）實際上並不存在。在大部分情況下，這是其他系統也常會遺漏的符號，而在 `config.h` 中有一個設定會解決這個問題。

比方說，您的系統可能未定義 `unsetenv` 函數。`configure-zos` Script 通常應該會偵測到這個狀況；如果未偵測，請編輯代理程式原始檔套件所提供的 `config.h` 檔，如下所示：

請將 `#define HAVE_UNSETENV 1` 變更爲 `#undef HAVE_UNSETENV`。

重新執行 `./build-zos` 指令來更正問題。

註：其他函數有類似的 `#define` 陳述式。

代理程式更新及部署公用程式

代理程式更新及部署公用程式 **agentupdate**，可以簡化在多部電腦上安裝及更新 IBM Rational Build Forge 代理程式的程序。

在 Rational Build Forge 自動執行作業的所有電腦上，均需要 Rational Build Forge 代理程式。在大量電腦上安裝或更新代理程式，可能會很耗時。此公用程式可以同時在所有目標電腦上安裝或更新代理程式。您可以在指令行上，或是透過 Rational Build Forge 提供的專案來啟動此公用程式。此公用程式會使用您提供的 XML 檔案，來判斷目標電腦、目標電腦的登入認證、連線方法及相關資訊。

重要：本工具支援單向安裝或升級作業。

- 本工具不支援從安裝或升級進行回復。
- 本工具不會針對要安裝的版本檢查現有代理程式的版本。如果您指定要安裝的版本低於已經安裝的版本，則會安裝較低的版本。

當您啟動此公用程式時，它會針對每部目標電腦完成下列動作：

1. 登入電腦
2. 判斷作業系統
3. 停止所有執行中的代理程式

您可以在與此公用程式搭配使用的 XML 檔中利用 **StopCommand** 參數，來使用自己的自訂指令以停止 Rational Build Forge 代理程式。

4. 判斷電腦是否有足夠的可用暫存空間
5. 將作業系統的代理程式安裝程式複製到目標電腦
6. 執行安裝程式以在電腦上更新或安裝代理程式

如果代理程式安裝在預設位置，此處理程序會保留您的 `BFAgent.conf` 檔。如果代理程式安裝在其他位置（您在與此公用程式搭配使用的 XML 檔中，使用 **ExistingAgentPath** 參數指定的位置），此處理程序也會保留該檔案。

7. 從暫存空間移除安裝程式

註：此公用程式還會建立日誌檔以擷取其作業的歷程。此日誌檔儲存在執行此公用程式所處的電腦上。

支援的作業系統

代理程式更新及部署公用程式可以與兩種作業系統配合運作。

此公用程式可在下列作業系統上執行：

- Windows
- Linux

附註：

- 執行此公用程式所處的電腦，必須裝有 IBM Rational Build Forge 隨附的 IBM Java。
- 此公用程式需要 Rational Build Forge 隨附的 IBM Tivoli® Remote Execution and Access。

此公用程式可以在下列作業系統上安裝及更新 Rational Build Forge 代理程式：

- Windows
- Linux
- IBM AIX
- HP-UX
- Solaris

安裝

IBM Rational Build Forge 安裝程式會安裝下列檔案中的代理程式更新及部署公用程式：

- Windows 系統
`<bfinstall>\java\agentupdate.jar`
`<bfinstall>\agentupdate.bat`
- UNIX 及 Linux 系統
`<bfinstall>/Platform/java/agentupdate.jar`

執行 UNIX 或 Linux 的目標電腦連線功能設定

您必須配置執行 UNIX 或 Linux 的目標電腦，以便能使用 IBM Tivoli Remote Execution and Access 取這些電腦。

若要確保透過 Tivoli Remote Execution and Access 進行存取，請確保目標電腦符合下列需求。

SSH：在代理程式更新及部署公用程式將要存取的任何 UNIX 或 Linux 目標電腦上，確保已啓用 SSH。

執行 Windows 的目標電腦連線功能設定

您必須配置執行 Windows 的目標電腦，以便能使用 IBM Tivoli Remote Execution and Access 探索這些電腦。

若要確保透過 Tivoli Remote Execution and Access 進行存取，您有兩個選項。

- **SSH/RSH：**在代理程式更新及部署公用程式將要存取的目標電腦上，確保已啓用 SSH/RSH。

如果執行 Windows 作業系統的目標電腦具有 SSH 或 RSH，您可以跳過其餘配置步驟。

- 配置執行 Windows 作業系統的目標電腦，以符合下列小節中所述的 Tivoli Remote Execution and Access 需求。第一節適用於所有 Windows 作業系統。後面幾節則適用於所示的 Windows 作業系統。

註：本主題中的 Windows 資訊來源於透過下列鏈結取得的資訊。如需最新資訊，請參閱 Windows 作業系統上有關 IBM Tivoli Remote Execution and Access 配置的下列鏈結：

- http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/index.jsp?topic=/com.ibm.tivoli.tpm.scenario.doc/sdi/rept_prereq_win.html
- http://publib.boulder.ibm.com/infocenter/rafwhelp/v7r1/index.jsp?topic=/com.ibm.rafw.install.doc/topics/r_network_connection_reqs_Windows.html

所有支援的 Windows 作業系統

- 您必須在目標電腦上啟用遠端登錄管理（預設配置）。如下所示，開啓「服務」：
 - Windows XP、Windows Vista：選取**開始 > 控制台 > 系統管理工具 > 服務**
 - Windows Server 2003、Windows Server 2008：選取：**開始 > 設定 > 控制台 > 系統管理工具 > 服**

在 **Remote Registry** 服務中，確定該服務的狀態為已啓動。

- 您必須啓用管理共用，才能用 Tivoli Remote Execution and Access 連接至執行 Windows 的目標電腦。預設管理磁碟共用的範例為 C\$ 及 D\$。如果您停用共用，Tivoli Remote Execution and Access 會將這些磁碟機中的目錄視為隱藏。在此情況下，系統會顯示下列訊息：

```
XCIM0009E: Error connecting to remote target <host_name>. Exception: java.io.FileNotFoundException:
CTGRI0003E The remote path name specified cannot be found: <file_or_directory_path>.
Cause: com.starla.smb.SMBException: The network name is incorrect.
```

請遵循下列步驟來啓用管理共用：

1. 按一下**我的電腦**。
 2. 用滑鼠右鍵按一下要啓用管理共用的磁碟機。
 3. 按一下**共用及安全**。
 4. 選取**共用此資料夾**。
 5. 指定共用名稱（如 C\$ 或 D\$），然後按一下**確定**。
- 如果已在目標上停用 Windows Scripting Host (WSH) 或 WMI 服務，或已停用 VBScript，部分 Windows 通訊協定方法將無法運作。

Windows XP Professional

- 執行 Windows XP 作業系統的目標電腦必須停用「簡易檔案共用」，Tivoli Remote Execution and Access 才能正常運作。簡式網路會將所有登入強制當作訪客來鑑別。訪客登入不需要 Tivoli Remote Execution and Access 正常運作所需的授權。若要停用「簡易檔案共用」，請執行下列動作：
 1. 在「Windows 檔案總管」視窗中，按一下**工具 > 資料夾選項**，然後按一下**檢視標籤**。
 2. 在「進階設定」清單中，清除**使用簡易檔案共用**勾選框。
 3. 按一下**套用和確定**。

- 必須在目標電腦上同時啟用埠 135 (RPC) 及 445 (TCP)，才能確保用 Tivoli Remote Execution and Access 成功進行通訊。如果 Tivoli Remote Execution and Access 發現已停用埠 445，則會使用埠 139 (NetBIOS over TCP/IP)。
- 防火牆：Windows XP 作業系統包含稱為「網際網路連線防火牆」的內建防火牆，該防火牆預設為停用。此外，Windows XP Service Pack 2 (SP2) 隨附「Windows 防火牆」，該防火牆預設為啟用。如果在執行 Windows XP 的目標電腦上啟用任何防火牆，則防火牆會阻止 Tivoli Remote Execution and Access 進行存取。

在 Windows XP SP2 上，您可以在「Windows 防火牆」配置的「例外」頁面上選取檔案及印表機共用勾選框，以容許存取。

Windows Vista

- 您必須為 Guest 或 Everyone 帳戶啟用檔案共用，並停用受密碼保護的共用。

若要停用受密碼保護的共用，請執行下列動作：

1. 按一下**開始** > **控制台**。
 2. 按一下**網路和網際網路**，然後按一下**共用及探索**。
 3. 按一下**以密碼保護的共用**旁邊的向下箭頭。
 4. 按一下**關閉以密碼保護的共用**。
 5. 按一下**套用**，然後結束「控制台」。
- 您可能需要修改登錄項目：
 1. 在**開始** > **開始搜尋**欄位（位於左下角）中，鍵入 regedit，然後按 Enter 鍵。
 2. 在左窗格中，瀏覽至 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system 資料夾。
 3. 用滑鼠右鍵按一下右窗格中的空白區域。
 4. 按一下**新增**。
 5. 按一下 **DWORD 值**。
 6. 鍵入 LocalAccountTokenFilterPolicy。
 7. 按兩下您所建立的項目。
 8. 在方框中鍵入 1。
 9. 按一下**確定**。
 10. 重新啟動電腦
 - 防火牆：Windows Vista 包含稱為「網際網路連線防火牆」的內建防火牆，該防火牆預設為停用。如果在執行 Windows Vista 作業系統的目標電腦上啟用任何防火牆，則防火牆會阻止 Tivoli Remote Execution and Access 進行存取。

Windows Server 2003

- 必須在目標電腦上同時啟用埠 135 (RPC) 及 445 (TCP)，才能確保用 Tivoli Remote Execution and Access 成功進行通訊。如果 Tivoli Remote Execution and Access 發現已停用埠 445，則會使用埠 139 (NetBIOS over TCP/IP)。

Windows Server 2008

- 您必須為 Guest 或 Everyone 帳戶啟用檔案共用，並停用受密碼保護的共用。

若要停用受密碼保護的共用，請執行下列動作：

1. 按一下**開始** > **控制台**。
 2. 按一下**網路和網際網路**，然後按一下**共用及探索**。
 3. 按一下**以密碼保護的共用**旁邊的向下箭頭。
 4. 按一下**關閉以密碼保護的共用**。
 5. 按一下**套用**，然後結束「控制台」。
- 您可能需要修改登錄項目：
 1. 按一下**開始** > **執行**。
 2. 鍵入 `regedit`，然後按 **Enter** 鍵。
 3. 在左窗格中，瀏覽至 `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system` 資料夾。
 4. 用滑鼠右鍵按一下右窗格中的空白區域。
 5. 按一下**新增**。
 6. 按一下 **DWORD 32 位元值**。
 7. 鍵入 `LocalAccountTokenFilterPolicy`。
 8. 按兩下您所建立的項目。
 9. 在方框中鍵入 `1`。
 10. 按一下**確定**。
 11. 重新啟動電腦
 - 防火牆：Windows Server 2008 包含稱為「網際網路連線防火牆」的內建防火牆，該防火牆預設為停用。如果在執行 Windows Server 2008 作業系統的目標電腦上啟用防火牆，則防火牆會阻止 Tivoli Remote Execution and Access 進行存取。

用法概觀

使用代理程式更新及部署 (**agentupdate**) 公用程式，需要執行三個步驟。

若要在多部電腦上安裝或更新 IBM Rational Build Forge 代理程式，請執行下列動作：

1. 將代理程式安裝程式置於將要執行 **agentupdate** 公用程式之電腦上的某個目錄中。

您可以從安裝媒體，以及 Jazz™.net 上 Jazz Community 網站的 Rational Build Forge 專案區域中取得安裝程式。

2. 建立 XML 檔，以指定要安裝或更新 Rational Build Forge 代理程式的電腦。

註：如果您將登入認證包含在 XML 檔中，請以明碼指定這些認證。如需加密密碼的相關資訊，請參閱第 146 頁的『執行 **agentupdate** 公用程式』。

3. 從指令行或透過提供的 Rational Build Forge 專案，執行 **agentupdate** 指令。

如需 XML 檔的相關資訊，請參閱『在 XML 檔中指定目標系統』。

如需如何執行此公用程式的相關資訊，請參閱第 146 頁的『執行 **agentupdate** 公用程式』。

在 XML 檔中指定目標系統

使用 XML 檔可指定要更新或安裝代理程式的目標系統。

XML 檔概觀

代理程式更新及部署公用程式需要 XML 檔，來決定要更新或安裝代理程式的電腦。此檔案還必須指定登入認證，以及公用程式所需的其他項目。

此檔案具有下列需求：

- 用於包含其他元素的一個 <AgentDeploymentAndUpdate> 元素
- 用於指定每部目標電腦的一個以上 <Target> 元素
- 如果使用 <Globals> 元素，則只能使用一個。請針對適用於所有目標電腦的項目使用此元素

下列範例顯示單一 <Target> 元素及 <Globals> 元素。只有與單一 <Target> 元素搭配使用時，<Globals> 元素沒有用。當您有多個 <Target> 元素時，<Globals> 元素會很有用。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Target id="9.184.112.152">
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="passWord"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="GZIPPath" Value="/usr/contrib/bin/" />
  </Target>

  <Globals>
    <Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
  </Globals>

</AgentDeploymentAndUpdate>
```

XML 檔語法

每個 <Parameter> 元素皆適用於 <Target> 及 <Globals> 元素。由於具有此彈性，當參數適用於所有目標電腦時，您可以使用 <Target> 元素以便僅指定 ID 屬性，然後將每個 <Parameter> 元素置於 <Globals> 元素中。

使用 Value="" 指定參數，與不指定該參數的結果相同。

下列表格說明該檔案的語法。

目標屬性	值
ID	目標電腦的名稱或 IP 位址。

參數	值
AgentInstallLocation	僅適用於 IBM AIX、HP-UX 及 Windows 系統 - 對其他平台不起作用 放置新代理程式的位置。如果未指定，則會使用系統預設暫存目錄。
BFAgentPort	僅適用於 IBM AIX 及 HP-UX 系統 代理程式使用的埠。產生回應檔時使用。

參數	值
BFASolarisAdminFile	僅適用於 Solaris 系統 目標系統上儲存用於無聲自動安裝的回應檔之位置。如果未指定，則會使用系統預設暫存目錄。
BFASolarisResponseFile	僅適用於 Solaris 系統 目標系統上儲存回應檔的位置。如果未指定，則會使用系統預設暫存目錄。
ConnectionType	UNIX 或 Linux：SSH 或 RSH Windows：SSH、RSH、SMB 或 CIFS (RSH 是遠端 shell。SMB 是「伺服器訊息區塊」。CIFS 是「共用網際網路檔案系統」。)
ConnectionPort	(選用) 如果目標不使用預設埠，則為用於指定連線類型 (使用 ConnectionType 參數) 的埠號。如果目標使用預設埠，則無需指定此參數。
ExistingAgentPath	IBM AIX、HP-UX 及 Windows 系統的必要參數。 目標電腦上現有代理程式所處的位置。公用程式會在您指定的位置建立 BFAgent.conf 檔的備份，並在安裝或升級完成後還原該檔案。若為其他系統，則會使用預設系統安裝位置。
GZIPPath	IBM AIX、HP-UX 及 Solaris 系統。 代理程式安裝程式可在其中找到 gzip 指令 (解壓縮安裝程式) 的目錄。
IsSudoEnabled	如果已在目標系統上啟用 sudo ，且您要使用它，請將其設為 yes 。請將 SudoPassword 設為正確的 sudo 密碼。如果未設定，公用程式會嘗試使用 Password 來執行 sudo 登入。
KeyFile	執行公用程式所處電腦上的檔案，用於儲存加密及解碼密碼所用的金鑰。
LocalTempLocation	僅適用於 IBM AIX、HP-UX 及 Windows。 目標系統上儲存回應檔的位置。如果未指定，則會使用系統預設暫存目錄。若為 Solaris 系統，則會使用 BFASolarisResponseFile 或 BFASolarisAdminFile 。
ManagedScriptsDir	設為執行公用程式所處的目錄。此目錄位於執行 agentupdate 公用程式所處的電腦上。
SourceDirectory	用於儲存目標電腦的安裝程式 (.exe、.rpm 及 .gz 格式) 的目錄。此目錄位於執行 agentupdate 公用程式所處的電腦上。
StartCommand	(選用) 用於在目標電腦上啟動 Rational Build Forge 代理程式的自訂指令。 請確保將此指令包含在 PATH/path 變數中，或在 XML 檔中提供完整路徑。
StopCommand	(選用) 用於在目標電腦上停止 Rational Build Forge 代理程式的自訂指令。 請確保將此指令包含在 PATH/path 變數中，或在 XML 檔中提供完整路徑。
SudoPassword	供 sudo 使用的密碼。請參閱 IsSudoEnabled 。
Password	供 UserName 在登入目標電腦時使用的密碼。 重要： 您最初可在 XML 檔中以明碼來定義 Password 參數。然後如第 146 頁的『執行 agentupdate 公用程式』中所示，使用 -encrypt 指令行選項來加密該密碼。
TemporaryLocation	在執行公用程式之前，目標電腦上儲存安裝程式的位置。
UseManagedScripts	設為 yes 或 no 。如果設為 yes ，公用程式會嘗試在 ManagedScriptsDir 目錄中，尋找 StartCommand 及 StopCommand 。
UserName	登入目標電腦時使用的使用者名稱。

範例 XML 檔

下列範例將示範如何使用 XML 檔。

- 範例 1

本範例只有一個目標。目標 IP 位址為 9.164.259.143。<Target> 元素不包含選用參數 **SourceDirectory** 或 **TemporaryLocation**。但是，<Globals> 元素具有 **SourceDirectory** 參數。公用程式會針對此目標使用該參數的值。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Globals>
    <Parameter Name="SourceDirectory" Value="..\test"/>
    <Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
  </Globals>

  <Target id="9.164.259.143">
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="toor"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
  </Target>

</AgentDeploymentAndUpdate>
```

- 範例 2

本範例具有四個目標。但這些目標皆不需要專屬參數。因此，可以在 <Globals> 區段中指定所有參數，包含 **UserName**、**Password** 及 **ConnectionType**。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

  <Globals>
    <Parameter Name="SourceDirectory" Value="..\test"/>
    <Parameter Name="ConnectionType" Value="ssh"/>
    <Parameter Name="UserName" Value="root"/>
    <Parameter Name="Password" Value="toor"/>
    <Parameter Name="TemporaryLocation" Value="/tmp"/>
    <Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
    <Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
  </Globals>

  <Target id="9.164.259.143">
  </Target>

  <Target id="9.164.102.169">
  </Target>

  <Target id="9.164.102.196">
  </Target>

  <Target id="9.106.189.157">
  </Target>

</AgentDeploymentAndUpdate>
```

- 範例 3

您可以為目標系統指定某個範圍內的 IP 位址。下列範例適用於 9.184.112.152 至 9.184.112.160 範圍內的 IP 位址，這些位址在每個系統上均使用相同登入認證：登入為 root，密碼為 password。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>
<Target id="9.184.112.152-160">
<Parameter Name="ConnectionType" Value="ssh">
<Parameter Name="ConnectionPort" Value="" />
<Parameter Name="UserName" Value="root">
<Parameter Name="Password" Value="passWord">
</Target>
</AgentDeploymentAndUpdate>
```

• 範例 4

如果所有目標均具有相同內容（連線類型、連線埠、使用者名稱、密碼），則可在檔案的 Globals 區段中指定這些內容。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>
<Globals>
<Parameter Name="ConnectionType" Value="ssh">
<Parameter Name="ConnectionPort" Value="">
<Parameter Name="UserName" Value="root">
<Parameter Name="Password" Value="passWord">
</Globals>

<Target id="9.184.112.152-160">
</Target>

<Target id="9.184.80.12-16">
</Target>
</AgentDeploymentAndUpdate>
```

執行 agentupdate 公用程式

建立 XML 檔之後，您可以在指令行上，或是透過提供的 BuildForgeAgentUpdateUtilityProject 來執行代理程式更新及部署公用程式。

安裝考量

本工具支援單向安裝或升級作業。

- 本工具不支援從安裝或升級進行回復。
- 本工具不會針對要安裝的版本檢查現有代理程式的版本。如果您指定要安裝的版本低於已經安裝的版本，則會安裝較低的版本。

在指令行上執行公用程式

下列範例將示範如何在指令行上執行公用程式。如需公用程式語法，請參閱 第 147 頁的『agentupdate 公用程式參照』。

下列指令使用 **-preview** 選項。利用此選項，公用程式可連接至目標電腦，並列印將在這些目標上執行的指令，而實際上並不執行這些指令。

```
java -jar agentupdate.jar -preview -targets agent_deploy_config.xml -keyfile mykeyfile.txt
```

下一個指令會加密輸入 XML 檔中的所有明碼密碼。此指令不會連接至任何遠端電腦。

```
java -jar agentupdate.jar -encrypt -targets agent_deploy_config.xml -keyfile mykeyfile.txt -logFile LogFile.txt
```

此指令會在目標電腦上更新或安裝 IBM Rational Build Forge 代理程式。 **-sourceDir** 選項可指出哪個目錄包含所有目標的代理程式安裝程式。如果您在執行公用程式時未使用此選項，請在 XML 檔中包含對應的參數。如果安裝程式在 XML 檔中所有目標的公用目錄中，請在 XML 檔中指定 <Globals> 元素內的 **SourceDirectory** 參數。但是，如果任何一個目標的來源目錄不相同，請定義該電腦 <Target> 元素內的 **SourceDirectory** 參數。如需相關範例，請參閱第 145 頁的『範例 XML 檔』中的「範例 2」。

```
java -jar agentupdate.jar -sourceDir .\resources -targets agent_deploy_config.xml -logFile LogFile.txt
```

透過 BuildForgeAgentUpdateUtilityProject 執行公用程式

IBM Rational Build Forge 提供 bfaagentupdate.xml 專案範例，示範如何使用此公用程式。

若要使用此專案範例，請使用「管理主控台」中的匯入機能，或使用 **bfimport** 指令匯入該專案。從 C:\Program Files\IBM\Build Forge\samples\agentupdate\BFproject\ (Windows) 或 /opt/buildforge/samples/agentupdate/BFproject/ (Linux) 匯入該專案。

專案需要三個步驟：預覽、加密及執行。專案範例不包含預覽和加密步驟。

- 預覽步驟：建置系統連接至目標電腦，並顯示爲了更新或安裝代理程式，將在每個目標上執行的指令，而實際上並不執行這些指令。
- 加密步驟：建置系統加密所有密碼，而不在目標電腦上更新或安裝代理程式。此步驟會使用已加密密碼來取代輸入 XML 檔中的明碼密碼。
- 執行步驟：建置系統連接至目標電腦、識別其作業系統、停止所有執行中的代理程式、檢查用於安裝程式的暫存空間、將安裝程式複製到目標電腦、完成更新或安裝，然後從暫存空間移除安裝程式。

請根據您的需求執行上述步驟，然後執行該專案。

agentupdate 公用程式參照

此公用程式提供數個指令行選項及環境變數來控制其行爲。

下表按字母順序呈現這些指令行選項。

選項	必要	說明
-debug	否	啓用延伸除錯輸出。 您也可以透過將 DEBUG 環境變數設爲任何值來啓用除錯。如需優先順序的相關資訊，請參閱 第 148 頁的『XML 檔案參數、指令行選項及環境變數的優先順序』。
-encrypt	否	使用加密形式的密碼取代 XML 檔中的明碼密碼。
-genkey <key_file>	否	產生用於加密及解密密碼的秘密金鑰。 指定 -genkey 選項時，必須同時指定 -keyfile <key_file> 選項，以將金鑰儲存在指定檔案中。
-keyfile <key_file>	否	保存用於加密及解密密碼的金鑰。 您也可以透過設定 KeyFile XML 檔案參數來指定此檔案。如需優先順序的相關資訊，請參閱 第 148 頁的『XML 檔案參數、指令行選項及環境變數的優先順序』。

選項	必要	說明
-logFile <log_file>	否	指定公用程式在其中記載訊息的檔案。 依預設，此名稱爲 RemoteAgentDeployerUpdater_Log.txt。
-maxThreads	否	設定公用程式在執行時使用的執行緒數目上限。依預設，此值會設為主機硬體中處理器核心數目的兩倍。
-preview	否	列出（但不執行）公用程式將針對每個目標執行的指令。
-sourceDir <path>	否	指定用於儲存不同平台的代理程式安裝程式（.exe、.rpm 及 .gz 格式）的目錄。 agentupdate 公用程式會判斷每個目標的作業系統，然後使用此目錄中對應的代理程式安裝程式。 註： 此公用程式會使用代理程式安裝程式。它不會使用代理程式原始碼。 您也可以透過設定 SourceDirectory XML 檔案參數或 SOURCEDIR 環境變數來指定此目錄。如需優先順序的相關資訊，請參閱『XML 檔案參數、指令行選項及環境變數的優先順序』。
-targets <XML_file>	是	指定您建立的 XML 檔的路徑及檔名，以指出要更新或安裝 IBM Rational Build Forge 代理程式的目標電腦。

下表說明環境變數。

環境變數	必要	說明
DEBUG	否	啓用延伸除錯輸出。 您也可以透過指定 -debug 指令行選項來啓用除錯。如需優先順序的相關資訊，請參閱『XML 檔案參數、指令行選項及環境變數的優先順序』。
SOURCEDIR	否	指定用於儲存不同平台的代理程式安裝程式（.exe、.rpm 及 .gz 格式）的目錄。 agentupdate 公用程式會判斷每個目標的作業系統，然後使用此目錄中對應的代理程式安裝程式。 註： 此公用程式會使用代理程式安裝程式。它不會使用代理程式原始碼。 您也可以透過設定 SourceDirectory XML 檔案參數或 -sourceDir 指令行選項來指定此目錄。如需優先順序的相關資訊，請參閱『XML 檔案參數、指令行選項及環境變數的優先順序』。

XML 檔案參數、指令行選項及環境變數的優先順序

您可以採用多種方式來控制公用程式。優先順序將決定實際上會影響公用程式的方法。

代理程式更新及部署公用程式，按優先順序提供下列方法來控制其行爲：

1. <Target> 元素中的參數
2. 指令行選項
3. <Globals> 元素中的參數
4. 環境變數

公用程式會按順序考量每種方法。如果使用有效的值定義了某種方法，公用程式即會使用該值，並忽略其後的所有方法。

範例

本範例說明公用程式如何尋找安裝程式。<Target> 及 <Globals> 元素皆已定義 **SourceDirectory** 參數。這些參數指出安裝程式可能所處的位置。但是，公用程式有數個位置需要檢查。一旦公用程式找到安裝程式，即會使用該安裝程式。

1. 公用程式首先會檢查 <Target> 元素中的參數。如果存在 `..\test\wind` 目錄，公用程式會檢查該目錄中是否有安裝程式。如果安裝程式在該目錄中，公用程式即會使用該安裝程式。
2. 如果該目錄不存在或不含安裝程式，公用程式會檢查是否已指定 **-sourceDir** 指令行選項。如果指定了該選項並包含有效的目錄，公用程式即會檢查該目錄並使用其中的安裝程式。
3. 如果未指定 **-sourceDir**、該目錄無效或不含安裝程式，公用程式即會檢查 <Globals> 元素中的參數。此時，公用程式會檢查 `..\test` 目錄。
4. 如果 `..\test` 目錄中未出現安裝程式，公用程式即會檢查 **SOURCEDIR** 環境變數。如果該目錄不存在或不含安裝程式，公用程式即會產生錯誤。

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentDeploymentAndUpdate>

<Globals>
<Parameter Name="SourceDirectory" Value="..\test"/>
<Parameter Name="KeyFile" Value="..\test\keyfile.txt"/>
</Globals>

<Target id="9.164.102.169">
<Parameter Name="SourceDirectory" Value="..\test\wind"/>
<Parameter Name="ConnectionType" Value="ssh"/>
<Parameter Name="UserName" Value="root"/>
<Parameter Name="Password" Value="toor123"/>
<Parameter Name="ExistingAgentPath" Value="/usr/local/bin/bfagent"/>
<Parameter Name="AgentInstallLocation" Value="/usr/local/bin/bfagent"/>
</Target>

</AgentDeploymentAndUpdate>
```

透過日誌檔檢視公用程式動作

日誌檔提供公用程式動作的歷程。

您可以使用 **-logFile** 選項，來指定記載訊息的檔案。如果您不指定 **-logFile** 選項，公用程式會將訊息記載至 `RemoteAgentDeployerUpdater_Log.txt`。此檔案建立在儲存 **agentupdate** 公用程式的目錄中。

日誌檔的格式為 [MSGTYPE] [IP address/MachineName] [Msg]。

所有目標的訊息均儲存在相同日誌檔中。

疑難排解

agentupdate 公用程式為您提供多種方式來顯示資訊，以協助對問題進行疑難排解。

依預設，公用程式會提供足夠的資訊以讓您瞭解一般問題，如連線功能問題、磁碟空間不足等。但是，如果您需要更多除錯資訊，請使用 **-debug** 指令行選項。或者，您可以使用任何值來定義 **DEBUG** 環境變數，以便從工具中取得額外的日誌訊息（或更多除錯資訊）。

agentupdate 的訊息參照

agentupdate 公用程式在作業期間會提供錯誤及參考訊息。

下列主題提供所有 **agentupdate** 訊息的參照。

CRRBU0001I 已順利啟動代理程式更新及部署公用程式。	CRRBU0020I 不明節點
CRRBU0002I 除錯模式開啓。	CRRBU0021I 目標數總計：{0}
CRRBU0003I 已完成密碼加密。	CRRBU0022I 元素計數 = {0}。
CRRBU0004I 已剖析 XML 配置檔。正在處理目標...	CRRBU0023I 正在針對 {0} (userid={1}) 嘗試 Windows 通訊協定。
CRRBU0005I 正在處理目標 [{0}]	CRRBU0024I 正在使用 RXA 介面檢查目標機器上的可用空間。
CRRBU0006I [{0}] 通訊協定 = {1} 使用者 = {2}	CRRBU0025I 目標系統上的可用空間 = {0} KB。
CRRBU0007I [{0}] 已加密密碼 = {1}	CRRBU0026I 正在 OS {0} 中的安裝位置 {1} 上檢查代理程式。
CRRBU0008I 已順利建立受管理 Script 目錄：{0}。	CRRBU0027I 在此位置上找到安裝的代理程式：{0}。
CRRBU0009I I[{0}] 已解密密碼 = #{1}#	CRRBU0028I [正在停止處理程序]
CRRBU0010I 參數 keyfile = {0}。	CRRBU0029I 程序 ID = {0}，程式名稱 = {1}
CRRBU0011I 子項數：{0}。	CRRBU0030I 正在剖析暫置：元素：{0} = {1}
CRRBU0012I 參數 genkey = {0}。	CRRBU0031I 無法停止目標上的代理程式處理程序 (bfagent)。
CRRBU0013I [{0}] 已使用 sudo 方式加密密碼 = {1}。	CRRBU0032I 正在停止目標 {0} 上執行的代理程式。
CRRBU0014I 文字節點	CRRBU0033I 正在嘗試停止 Windows 上的服務 {0}。
CRRBU0015I 註解節點	CRRBU0034I 正在使用 RXA 介面停止目標上的代理程式服務。
CRRBU0016I 屬性節點	CRRBU0035I 正在停止目標 {0} 上的代理程式服務
CRRBU0017I 文件節點	CRRBU0036I 正在使用目標上的暫存目錄 {0}。
CRRBU0018I 文件類型節點	CRRBU0037I 正在檢查目錄中的字首相符項。{0}
CRRBU0019I 元素節點	

CRRBU0038I 正在 <{0}> 下尋找安裝程式。

CRRBU0039I 正在使用 **RXA** 介面將安裝程式 <{0}> 複製到目標。

CRRBU0040I 已建立金鑰檔：{0}

CRRBU0041I 使用者配置檔備份：{0}

CRRBU0042I 正在清除目標上的 {0}。

CRRBU0043I 正在移除目錄 [{0}]

CRRBU0044I 代理程式安裝位置 {0}

CRRBU0045I 安裝作業正在繼續執行 @ {0}

CRRBU0046I 安裝/升級目標：{0}

CRRBU0047I 已啟用預覽模式。

CRRBU0048I 已在目錄 {0} 中建立 **response.txt**。

CRRBU0049I 正在還原使用者配置檔：{0}。

CRRBU0050I 正在執行指令：{0}

CRRBU0051I 現行工作目錄是 [{0}]。

CRRBU0052I 正在複製到 {0}。

CRRBU0053I 正在結束目標 {0} 上執行的 **bfdispatch** 處理程序

CRRBU0054I 已為目標啟動新的執行緒。

CRRBU0055I 現已在目標上安裝代理程式。

CRRBU0056I 遠端存取問題：無法繼續執行預覽模式。

CRRBU0057I 正在使用 {0} 停止目標上執行的代理程式

CRRBU0058I 已順利停止目標上的代理程式。

CRRBU0059I 尚未在目標上安裝代理程式。

CRRBU0060I 目標上的暫時儲存位置為 {0}。

CRRBU0061I 加密所有密碼。

CRRBU0062I 將會針對來自 {2} 的 {1} 使用安裝程式 {0}。

CRRBU0063I 已順利將 {0} 複製到暫存位置。

CRRBU0064I 備份配置檔成功。

CRRBU0065I 現有代理程式安裝位置：{0}。

CRRBU0066I 使用 {0} 啟動目標上執行的代理程式

CRRBU0067I 正在結束執行緒。

CRRBU0068I 已建立名為 {0} 的管理檔

CRRBU0069I 節點：{0}

CRRBU0070I 節點屬性：{0}

CRRBU0071I 參數 **logFile** = {0}

CRRBU0072I 參數 **sourceDir** = {0}

CRRBU0073I 參數 **sourceFile** = {0}

CRRBU0074I 參數 **targets** = {0}

CRRBU0075I **getOS**：[{0}][{1}]

CRRBU0084I 包含目標資訊的檔案路徑（必要）。

CRRBU0085I 建立金鑰的來源位置：{0}。

CRRBU0088I 執行緒上限設為：*{0}*

CRRBU1000W [*{0}*] 尚未指定通訊協定。

CRRBU1001W [*{0}*] 不支援目標通訊協定：*{1}*

CRRBU1002W XML 檔中沒有 **Globals** 區段。

CRRBU1003W XML 檔中沒有 **Targets** 區段。

CRRBU1004W 目標項目無效：找不到 XML 檔中目標區段的 ID。

CRRBU1005W [stderr] *{0}*

CRRBU1006W [stdout] *{0}*

CRRBU1007W 目標上的暫存位置設為空值。

CRRBU1008W 不受支援：Sun Solaris *{0}* 版

CRRBU1009W 在 XML 檔中目標區段的指定來源目錄中找不到安裝程式。

CRRBU1010W 尚未在指令行中指定安裝程式位置。

CRRBU1011W 尚未在 XML 檔的 **Globals** 區段中指定安裝程式位置。

CRRBU1012W 尚未在環境中指定安裝程式位置。

CRRBU1013W 目標錯誤：無法產生暫存管理檔。

CRRBU1014W 目標錯誤：無法建立暫存回應檔。

CRRBU1015W [回覆碼] *{0}*

CRRBU2000E 捕捉到異常狀況。

CRRBU2001E 本端或遠端位置上不存在啟動 **Script**。

CRRBU2002E 無法建立受管理 **Script** 目錄：*{0}*

CRRBU2003E 遺漏必要的引數：*{0}*

CRRBU2004E **TargetInfoTable** 資料結構的用法不正確。

CRRBU2005E 在 XML 檔中找到多個 **Globals** 標籤。請僅指定一個。

CRRBU2006E 尚未定義參數 **ManagedScriptsDir**。

CRRBU2007E 執行啟動指令 *<{0}>* 時發生問題

CRRBU2008E [*{0}*] 尚未指定明碼密碼，尚未指定 **KeyFile**

CRRBU2009E 為啟動/停止 **Script** 建立檔案物件時發生錯誤。

CRRBU2010E 目標項目無效：XML 檔中目標的主機名稱或位址為空值。

CRRBU2011E 本端或遠端位置上不存在停止 **Script**。

CRRBU2012E 寫入 XML 檔時發生錯誤。

CRRBU2013E 正在忽略 XML 檔中的重複目標項目：目標 *{0}*。

CRRBU2014E 可用空間不足。*{2}* 上的可用空間 = *{0}*，所需的可用空間 = *{1}*

CRRBU2015E 捕捉到連線異常狀況：*{0}*

CRRBU2016E 捕捉到「找不到檔案」異常狀況：*{0}*

CRRBU2017E 捕捉到 I/O 異常狀況：*{0}*

CRRBU2018E 停止目標上的代理程式處理程序 (**bfagent**) 時發生錯誤。

CRRBU2019E 停止目標上的代理程式服務時發生錯誤。

CRRBU2020E 目錄無效或安裝程式不存在：{0}

CRRBU2021E 符合目錄中字首的檔案過多：{0}

CRRBU2022E 沒有任何檔案符合目錄中的字首：{0}

CRRBU2023E 尚未指定安裝程式來源目錄，或在指定目錄中找不到安裝程式。

CRRBU2024E 錯誤：無法結束目標上的 **bfdispatch** 處理程序。

CRRBU2025E 執行停止指令 <{0}> 時發生問題

CRRBU2026E 無法完成停止指令。在 {0} 上將不會繼續進行

CRRBU2027E 停止目標上的代理程式時發生問題。

CRRBU2028E 無法完成停止指令。將不會繼續進行。

CRRBU2029E 目標 {0} 上的磁碟空間不足

CRRBU2030E 找不到安裝程式。

CRRBU2031E 將 {0} 複製到暫存位置失敗。

CRRBU2032E 備份配置檔失敗。

CRRBU2033E 還原配置檔失敗。

CRRBU2034E 由於 {0} 發生鑑別錯誤，因此遠端存取失敗

CRRBU2035E 不受支援的作業系統。

CRRBU2036E 無法完成啟動指令。在 {0} 上將不會繼續進行

CRRBU2037E 尚未定義參數 **ManagedScriptsDir**。

CRRBU2038E 不受支援的平台。

CRRBU2039E **StopScript** 沒有本端檔案，或是多個項目具有字首 **StopScript_**。

CRRBU2040E **StartScript** 沒有本端檔案，或是多個項目具有字首 **StartScript_**。

CRRBU2041E 無法向伺服器進行鑑別。

CRRBU2042E 無法建立金鑰檔 {0}。

CRRBU2043E 加密演算法錯誤。請確定在您的環境中可以使用 **DES**。無法建立金鑰檔。

CRRBU2044E 金鑰檔 {0} 不是有效的金鑰檔。無法設定金鑰檔。

CRRBU2045E 加密演算法錯誤。請確定在您的環境中可以使用 **DES**。無法設定金鑰檔。

CRRBU2046E 無法存取金鑰檔 {0}。無法設定金鑰檔。

CRRBU2047E 加密演算法錯誤。請確定在您的環境中可以使用 **DES**。加密失敗。

CRRBU2048E 在此環境中無法使用利用 **DES** 進行加密的填補功能。加密失敗。

CRRBU2049E 從金鑰檔產生的金鑰無效。請確定您使用的是正確的金鑰檔。解密失敗。

CRRBU2050E 不支援用於密碼的字元編碼。加密失敗。

CRRBU2051E 為密碼提供的資料長度不正確。加密失敗。

CRRBU2052E 不存在預期的資料填補。加密失敗。

CRRBU2053E 加密演算法錯誤。請確定在您的環境中可以使用 **DES**。解密失敗。

CRRBU2054E 在您的環境中無法使用利用 **DES** 進行加密的填補功能。解密失敗。

CRRBU2055E 從金鑰檔產生的金鑰無效。請確定您使用的是正確的金鑰檔。解密失敗。

CRRBU2056E 為密碼提供的資料長度不正確。解密失敗。

CRRBU2057E 不存在預期的資料填補。解密失敗。

CRRBU2058E 不支援用於密碼的字元編碼。解密失敗。

CRRBU2059E 尚未指定金鑰檔。

CRRBU2060E {0} 是無效的回答。

執行代理程式

本節說明如何設定代理程式以便執行，通常代理程式會執行為自動啟動服務或常駐程式。

在 Windows 上執行代理程式

代理程式通常會安裝為服務並設為「自動」，使其在開啓電腦時即可啟動。您必須登入已安裝代理程式的電腦，才能啟動或停止代理程式。

如果要啟動及停止代理程式，您可以使用**開始**功能表：

- 如果要啟動代理程式，請按一下**開始 > 程式集 > IBM Rational Build Forge > 啟動代理程式服務**。
- 如果要停止代理程式，請按一下**開始 > 程式集 > IBM Rational Build Forge > 停止代理程式服務**。

您也可以指令提示字元中使用下列指令：

- net start bfagent
- net stop bfagent

在 UNIX、Linux 及 MacOS 上執行代理程式

代理程式主要是執行以作為服務，必須在重新啟動系統時也能自動重新啟動。

視狀況新增 bfagent 項目至 inetd 或 xinetd 配置中。下例是 Linux 系統上 xinetd.d 中的 bfagent 項目，其中代理程式安裝在 /usr/local/bin 中：

```
# description: The IBM Rational Build Forge Agent serves build requests
#   from the IBM Rational Build Forge Management Console.
service bfagent
{
    disable           = no
    flags             = REUSE
    socket_type       = stream
    wait              = no
    user              = root
    server             = /usr/local/bin/bfagent
    log_on_failure    += USERID
}
```

必要的話，代理程式可以在 inetd/xinetd 環境以外執行。如果要將它當成獨立式常駐程式來執行，請使用 -s 選項。

bfagent -s

當您使用此選項時，代理程式會移至背景，再開始接聽連線。請將這個指令放在啟動 Script 中，以便能夠在啟動電腦時自動啟動代理程式。

在 System i 上執行代理程式

如果您打算在 System i 平台上執行代理程式，請檢閱這個主題中的資訊。

驗證代理程式埠號是唯一的

埠 5555（這是標準 Build Forge 代理程式埠）可能會重新指派給 System i 伺服器上的其他代理程式。在這個情況下，請先將 Build Forge 代理程式埠改成未指派的埠，再啟動這個代理程式。要執行這項作業，請直接編輯 `bfagent.conf` 檔。如需詳細資料，請參閱第 157 頁的『變更代理程式埠』。

手動啟動代理程式

如果您已完成安裝指示第 134 頁的『在 System i 平台上安裝代理程式』中的步驟 7，代理程式會在 System i 啟動時，以 BFAgent 使用者身分來啟動。

另一種方法是，您可以利用下列指令，在 System i 上手動啟動代理程式。

```
bfagent -s
```

註：如果 `bfagent.conf` 檔未安裝在 `/etc`（預設位置）中，請使用 `-f` 選項來指定 `bfagent.conf` 位置。

當您發出 `bfagent` 指令並手動啟動代理程式時，代理程式會以啟動代理程式的使用者身分來啟動。

- 如果 QSECOFR 使用者或是具有 *ALLOBJ 特殊權限的使用者啟動代理程式，該使用者會利用您在「管理主控台」中指定的伺服器鑑別來加以鑑別。
- 如果有另一個使用者啟動代理程式，請在 `bfagent.conf` 檔中配置 `magic_login` 設定來鑑別該使用者。請參閱 第 159 頁的『`bfagent.conf` 參照』。

驗證已安裝 i5/OS™ PASE 程式

代理程式會當成「i5/OS 可攜式應用程式解決方案環境 (PASE)」程式來執行。PASE 包含在 i5/OS 中，可供執行 AIX 二進位檔和指令。依預設通常會安裝 PASE。

如果要判斷是否已安裝 PASE 程式，請在指令行執行 `DSPSFWRSC`。

如果未安裝 PASE 程式，請從安裝 CD 載入它。

在 PASE 中使用代理程式

在 i5/OS 上建置應用程式所需的大部分作業，都可從 PASE 環境存取。在針對 iSeries 平台規劃及定義程序自動化時，請務必牢記這一點。

PASE Shell 會解譯步驟中的指令，您也可以使用下列語法來執行原生指令：

```
system -bi0E "<native commands>"
```

重要：步驟中的每一個系統指令都會執行自己的處理程序。引申來說，即指令只能在其本身處理程序中運作。

比方說，如果您想要為一組步驟設定其程式庫清單：

- 您不能使用 `CHGSYSLIBL` 或 `ADDLIBL` 作為步驟指令，因為它們是原生指令（PASE 無法辨識）。

- 您不能在步驟中使用支援的原生指令語法（例如：`system -bi0E "ADDLIBLE FLGHT400"`），因為它只會變更適用於指令本身程序的程式庫清單，變更並不會影響後續的指令和步驟。

雖然您不能只針對一個步驟、一組步驟或一個專案設定程式庫清單，但是您可以在 **BFAGENT** 使用者的啟動指令 **Script** 中加以設定。請參閱第 134 頁的『在 **System i** 平台上安裝代理程式』中的範例啟動 **Script**。在啟動指令 **Script** 中設定程式庫清單，會設定範例中以 **BFAGENT** 使用者身分執行之所有專案和步驟的程式庫清單。執行專案和步驟的使用者必須有權存取所需的程式庫。

如果要設定程式庫清單，請新增列出所需程式庫之代理程式的工作說明。以下範例工作說明包含程式庫 **FLGHT400** 和 **FLGHT400M**。

```
10    UTLIB
20    QGPL
30    QTEMP
40    FLGHT400
50    FLGHT400M
```

代理程式在其啟動例行程序中指定此工作說明。比方說，如果工作說明為 **BFAJOB**D，則在系統啟動常式中的字行如下所示：

```
ADDAJE SBSDB(BFAGENT/BFAGENT) JOB(BFAGENT) JOBD(BFAGENT/BFAJOB
```

這個解決方案會影響在與這個代理程式關聯的 **System i** 伺服器上，所執行的所有指令（來自任何步驟和專案）。

bfagent 參照

bfagent 執行檔會啟動 **Build Forge** 代理程式。其會從相同目錄中的 **BFAgent.conf** 檔讀取其配置。

指令語法為：

```
bfagent [-f configfile | -s]
```

選項

-f configfile

使用 **configfile** 中的配置檔而非 **BFAgent.conf** 來執行。這是 **UNIX** 或 **Linux** 上的執行時期選項。在 **Windows** 上手動執行代理程式時它會作為除錯選項，而無法用來啟動 **Windows** 上的服務。

-s 啟動為獨立式服務。您只能在 **UNIX** 或 **Linux** 上使用這個選項。以這種方式執行可以替代以 **inetd** 或 **xinetd** 來啟動 **bfagent**。

配置代理程式

本節說明如何在安裝之後配置代理程式。

找出代理程式配置檔

代理程式配置檔 **BFAgent.conf** 會提供代理程式作業的執行時期配置。其中包含解釋所有可能選項的註解。該檔案位於代理程式的安裝目錄中：

- **Windows** 預設值：**C:\Program Files\IBM\Build Forge\Agent\BFAgent.conf**
- **UNIX** 和 **Linux** 預設值：**/etc/bfagent.conf**

重要：如果您變更安裝目錄中的 BFAgent.conf 檔，在後續重新安裝或升級代理程式時，都必須重複該項變更。每次安裝期間都會改寫此配置檔。

您也可以指定替代的配置檔：

- 在 UNIX 或 Linux 系統上，您可以使用不在代理程式安裝目錄中的配置檔來保留原有的代理程式配置。當您這麼做時，請在用來啟動代理程式的指令中使用 `-f` 指令行選項。範例：

```
bfagent -f /opt/bfagent.conf
```
- 在 Windows 系統上，您不能以這個選項啟動服務。只有在手動執行代理程式時，才能使用服務。它主要是作為除錯工具。

變更代理程式埠

如果代理程式要安裝在已佔用埠 5555 的伺服器上，您可以在安裝之後變更代理程式埠。

若要變更 Windows 作業系統上的埠：

1. 開啟登錄編輯器：按一下**開始 > 執行**，然後鍵入 `regedit`。
2. 移至 `HKEY_LOCAL_MACHINE\SOFTWARE\BuildForge Agent`。
3. 將 `AgentPort` 的值變更為所需的埠號。

在 UNIX、Linux 和 Macintosh 作業系統上：

1. 開啟 `/etc/services` 檔。
2. 將 `BuildForge Agent Port` 的值變更為所需的埠號。

配置不同的 Shell

您可以編輯 BFAgent.conf 檔案中的參數，來配置代理程式以使用預設 Shell 以外的 Shell。

例如，如果要將 Windows 系統變更為使用 MKSTools 所提供的 Korn Shell，您可以使用以下指令來變更 Shell 參數：

```
shell C:\MKSTools\mksnt\ksh.exe -L -c \"%s\"
```

當系統傳送指令給伺服器時，這個指令中的 % 會換成步驟指令。在此情況下，請使用反斜線跳出字元，以包含引號作為指令中的文字。

在「網路共用檔案系統」上執行代理程式指令 (Windows)

Build Forge 代理程式一開始是以 Windows 系統帳戶認證來啟動。為了執行指令，之後代理程式會使用 Build Forge 伺服器鑑別認證向 Windows 進行鑑別。

本端指令可接受伺服器鑑別認證，但對於代理程式必須在外部網路共用磁碟機上執行的某些指令來說，就可能失敗。例如，當在 ClearCase® 動態視圖中修改檔案時，代理程式必須存取網路共用磁碟機中的 ClearCase 檔案。

指令失敗的原因在於，外部檔案系統會忽略代理程式伺服器鑑別認證；它只會辨識代理程式最初的系統帳戶認證。

如果您在網路共用磁碟機上執行指令時遇到問題，請嘗試下列動作：

利用伺服器鑑別認證來執行指令

如果要使用具備網路共用區存取權的 Build Forge 伺服器鑑別認證來執行指令，請將 `win_reexec_after_auth` 設定加入 `BFagent.conf` 檔中。

如果您要使用 Build Forge 伺服器鑑別認證來建立網路共用區的存取，您必須新增這項設定。

`win_reexec_after_auth` 設定會使得代理程式在向 Windows 鑑別後，啟動新的程序。新的程序會強制共用檔案系統能夠辨識代理程式已變更使用者認證。

當設定 `win_reexec_after_auth` 時，代理程式會以服務方式執行，而不會區別指令是否存取網路共用區，因此您可能需注意效能衝擊問題。

在單一使用者模式下執行代理程式

在代理程式安裝期間，將代理程式設定成在單一使用者模式下執行指令，而不需要 Build Forge 伺服器鑑別認證。請選取**安裝使用者模式代理程式**選項。

如果指定的使用者是「管理者」群組的成員，則必須使用伺服器鑑別認證來指定使用者的認證。

如果使用者不是管理者，請在 `BFagent.conf` 中使用 `magic_login` 設定，以防未獲授權存取代理程式。

在您登入「管理主控台」時，代理程式會以您提供的使用者名稱啟動和執行，這會立即授權使用該使用者的認證來存取網路共用區。

使用專用的使用者帳戶，當代理程式當成服務執行

將代理程式設為以 Windows 服務方式執行，並使用專用的使用者帳戶。這個選項會限制您以單一使用者帳戶來執行代理程式，但不要求代理程式啟動新的處理程序來重新鑑別，因此不會影響效能。

如果要以專用的使用者帳戶，將代理程式當成服務來執行，請執行下列動作：

1. 在 Build Forge 伺服器上，按一下**管理工具 > 服務**來開啟 Windows 控制台。這時會開啟服務的清單。
2. 開啟「IBM Rational Build Forge 代理程式」服務。
3. 提供要執行代理程式指令之使用者的使用者帳戶資訊。例如，提供 ClearCase 管理使用者的資訊，或是對 ClearCase 動態視圖與 VOB 具有存取權的其他使用者資訊。

配置代理程式之間的直接檔案傳輸

當 `_XSTREAM_PROTOCOL` 觸發程式變數位於工作環境中時，可啟用代理程式之間的直接檔案傳輸。傳送代理程式、接收代理程式及引擎都必須具有直接檔案傳輸的功能。有數個參數可控制編碼、網路及檔案傳輸性質。

在某些作業系統上，代理程式對於直接檔案傳輸的支援有限：

- System z：不支援直接檔案傳輸。
- System i：僅針對 PRNG 或 PLAIN 的 `_XSTREAM_PROTOCOL` 值支援直接檔案傳輸。不支援對檔案傳輸使用 SSL（AES-CBC 的 `_XSTREAM_PROTOCOL` 值）。

請參閱 第 159 頁的『`bfagent.conf` 參照』參照中的第 276 頁的『觸發變數參照』及 `xstream_*` 參數。

觸發變數和代理程式效能

通過降低針對步驟日誌所產生的訊息數，兩個觸發變數可以影響代理程式的整體效能：

- `_SUPPRESS_ENV_OUTPUT`：如果設定，會在步驟日誌中抑制列印 ENV 訊息。
- `_SUPPRESS_LOG_OUTPUT`：如果設定，會在步驟日誌中抑制列印幾乎所有訊息。

請參閱「在使用環境中觸發變數參照」。當觸發變數包含在專案環境或步驟環境中時，便會影響工作。

bfagent.conf 參照

`bfagent.conf` 檔會儲存如何執行 Build Forge 代理程式的設定。該檔案與 `bfagent` 執行檔位於相同目錄中。

該檔案會列出所有設定和內部預設值。非作用中設定值則會加以註銷。

設定

`activity_log path`

開啓活動記載。這項資訊會附加到 `path` 指定的檔案中。該路徑必須存在，且代理程式使用者必須有其寫入權。

註：如果路徑不存在，或是無法寫入檔案，代理程式並不會報告錯誤。

重要：檔案大小沒有限制。該檔案必須手動刪除。這項設定的目的是暫時用來除錯代理程式。其目的並非用來作為運作中之代理程式的永久日誌。

`allow IP-address-or-range [...]`

請僅在下列情況下使用此設定：

- 在 Windows 上執行的代理程式
- 在 UNIX 或 Linux 上以獨立式模式執行的代理程式（啓動時使用 `-s` 選項）

這項設定會限制代理程式的連線。只接受來自符合 `IP-address-or-range` 之 IP 位址的連線。預設值是允許來自所有位址的連線。

請指定一或兩個項目：

- **IP 位址：**完整的 IPv4 或 IPv6 位址。例如，IPv4 可以是 255.192.192.003。只接受來自指定 IP 位址的連線。
- **IP 位址範圍：**部分 IPv4 或 IPv6 位址。以下是 IPv4 正確位址的範例：192.168 或 192.168.63。所有符合此資格的 IP 位址都可被接受。

註：如果您是在 `inetd` 或 `xinetd` 之類的超級伺服器上執行代理程式，請使用其他方法來控制存取權。您可能要使用防火牆、TCP Wrapper（`hosts.allow` 和 `hosts.deny`），或是 `xinetd` 的內建過濾功能。

`bind`

這項設定可讓使用者指定代理程式明確的連結位址。這項設定加上 `"port"` 設定會決定在以 `-s` 指令行選項啓動時，代理程式接聽連線的方式。`bfagent.conf` 檔中給定的值，會強制代理程式連結至 IPv4 本端主機位址；因此，代理程式只會從同一部電腦上的主控台接收連線。範例：`bind 255.192.192.003`

註：它不會影響由系統的服務架構所啓動的 Windows 或 UNIX 代理程式，如：`inetd`、`xinetd` 或 `launchd`。

ccviewroot root-path

這項設定會指定這部主機的起始視圖根目錄。請參閱 ClearCase 文件，以取得 init 的詳細資訊。內部預設值如下所示：

- Windows 系統：ccviewroot M:
- UNIX 或 Linux 系統：ccviewroot /view

cc_suppress_server_root

若已設定，請檢視路徑是 ccviewroot 所設定的路徑。若未設定，則在伺服器定義中所設定的路徑就會附加至 ccviewroot 所設定的路徑。此設定不需要值。如果已在 bfaagent.conf 中出現，則表示已設定。

command_output_cache size

這項設定會使代理程式快取輸出，直到達到指定的大小（位元組）為止。內部預設值不會快取。使用快取可以大幅改善代理程式的效能，並可降低網路負載。快取大小會視指令產生多少輸出而定。

最小值：2048。如果設為小於 2048 的值，則會在內部使用該值。

cygwin

對於 Windows 上的代理程式，僅使用這項設定。

這項設定可讓代理程式利用 Cygwin（一種類似 Linux 的環境）在 Windows 主機上運作。使用 Cygwin 時，代理程式就可以使用多種 Linux 工具。

當您使用這項設定時，可能也需要設定 cygwin_script_magic 和 shell 設定。下列顯示配置這些設定的一種方法：

```
cygwin
shell C:\cygwin\bin\bash.exe --login -c "%s"
cygwin_script_magic #!/bin/bash
```

此 shell 設定必須符合您的 Cygwin 安裝。

cygwin_script_magic

這項設定只能在設定 **cygwin** 時，用於 Windows 上的代理程式。

這項設定會指定 #! 字行，於執行步驟時使用。預設值是 #!/bin/bash。

default_logon_domain

指定當鑑別要求不含網域時，所要使用的網域。如果沒有指定，則會使用代理程式電腦的網域。

digest_algorithm SHA2

啟用密碼加密時，bfaagent 依預設會使用 SHA1。若要使用 SHA2，請啟用這項設定。這是 8.0 版中新增的設定。

當 bfaagent 升級到 8.0 時，這項設定不會自動新增至 bfaagent.conf。當您要使用 SHA2 時，必須將這項設定新增至 bfaagent.conf。如果您直接安裝 bfaagent 8.0，則 bfaagent.conf 會有這項設定。

如果您已將密碼加密，則啟用這項設定之後，必須再次加密密碼。

若要使用 SHA2，請確定 Build Forge® 主控台上的「摘要演算法」系統配置設定是設為 SHA2。如需這項設定的相關資訊，請參閱主題第 212 頁的『系統配置設定』。

註：若要使用整個安全雜湊演算法 2 (SHA2)，並在 Build Forge 管理主控台及代理程式中啓用密碼加密，那麼在**管理 > 安全**中將 SHA1 變更爲 SHA2 之後，可能需要更新密碼加密配置內容檔 `bfpwcrypt.conf`。

若已啓用密碼加密，請參閱第 167 頁的『如果已啓用密碼加密』，以了解下一步做什麼的指示。

若未啓用密碼加密，請參閱第 168 頁的『如果未啓用密碼加密』，以了解下一步做什麼的指示。

disable_telnet_support

爲了得到最佳結果，請使用 `telnet` 來測試代理程式連線。

對代理程式來說，有一些內建的處理超載和處理程序有關，以便能正確處理 Telnet 控制順序。

您可以使用這項設定，讓代理程式停止處理特殊 `telnet` 字元碼，因而能稍微提昇效能。在正式作業環境中，請使用這項設定來提昇效能。

disable_transcode

當作業系統未使用 UTF-8 編碼時，關閉代理程式轉換國際資料的處理程序。爲了避免混合編碼及資料毀損，請對代理程式的作業系統採用 UTF-8。

如果作業系統未使用 UTF-8 編碼，代理程式必須將資料轉換爲作業系統語言環境設定的正確編碼。

如果操作未使用 UTF-8，請使用這項設定以得到最佳結果並提昇代理程式的效能。

enable_agent_dll

這項設定會啓用 DLL 程序追蹤，這是一項除錯工具。

env_recursion_limit number-of-recursions

設定用於事先剖析的變數取代型遞迴限制。若未設定，則限制爲 32。

extensions

這項設定會指定指向外部功能程式庫的路徑。這些功能可以作爲步驟中的點指令。如果未指定這項設定，則不會載入外部程式庫。

在剖析期間，步驟指令中的第一個符記會被視爲功能名稱，第二個符記爲字串，第三個符記則爲整數逾時值（秒）。

需求：作業系統含有動態載入器支援。例如，在 UNIX 或 Linux 中，您需要有 `/usr/include/dlfcn.h`。這些是內部使用的預設值。

- UNIX 或 Linux：`/usr/local/bin/bfextensions.so`
- Windows：`C:\program files\ibm\build forge\agent\bfextensions.dll`

getaddrinfo_using_addrconfig

這項設定只適合在 UNIX 或 Linux 作業系統上，用來將代理程式當作一項獨立式服務來執行 (`bfagent -s`)。這項設定會讓代理程式在呼叫 `getaddrinfo()` 時使用 `AI_ADDRCONFIG`，以便選取接聽的介面。依預設，不會使用 `AI_ADDRCONFIG`。

如果您使用這項設定，代理程式會忽略未適當配置位址的介面。它只會使用有適當配置位址的介面來接聽。

gsk_ssl_key_location [<kdb_path> | <SAF_specification>]

指定 kdb 檔的完整路徑或 SAF 金鑰環規格。

gsk_ssl_kdb_password <password>

kdb 檔的密碼。這可以是純文字或加密文字。如果使用 SAF 金鑰環，請使用 NULL。請使用 `bfagent -e <plaintext>`，從純文字建立加密密碼。

gsk_ssl_protocol <protocol>

要使用的通訊協定，為 ALL（預設值）、SSLV2、SSLV3、TLSV1 或 TLSV1_1 其中之一

gsk_ssl_cipher_v2 <seed>

要用於系統 SSL 第 2 版 (SSLV2) 的密碼組合。預設值是 6321，應可適用於大部分應用程式。如需相關資訊，請參閱 System z 文件。

gsk_ssl_cipher_v3 <seed>

要用於系統 SSL 第 3 版 (SSLV3) 的密碼組合。預設值是 0906030201，應可適用於大部分應用程式。如需相關資訊，請參閱 System z 文件。

gsk_keyring_label <label>

kdb 檔中的金鑰標籤。

gsk_password_encrypt [true | false]

用來參照加密密碼。如果設為 True，請使用 `bfagent -e <plaintext>` 建立加密值，並設定 `gsk_ssl_kdb_password`。依預設會設為 False。

gsk_ssl_client_authentication [true | false]

指定是否要驗證用戶端憑證。預設值是 false

lang lang-code

請只有在「管理主控台」未提供有效的語言時，才使用這項設定。

這項設定會指定代理程式用來撰寫訊息和指令輸出的語言。一般而言，您不會明確設定這項設定，因為代理程式會使用「管理主控台」指定的語言。不過，如果電腦上無法使用所需的語言環境，設定語言就非常有用。如果「管理主控台」無法以某種語言通訊或是以無效的語言通訊，這項設定也可用來作為備份。

內部預設值是 en，如同明確設定如下所示：

```
lang en
```

leave_tmp_file

請只有在進行疑難排解時，才使用這項設定。

這項設定會將用來保留步驟指令的暫存檔予以保存，而非在執行指令之後加以刪除。在疑難排解時，該檔案可以與「管理主控台」中顯示的步驟相比較。

註：請勿對一般作業使用這項設定。

locale locale-code.charset-code

這項設定只能用於 UNIX 和 Linux 作業系統。Windows 以不同的方式來處理語言環境。

這項設定會指定本地化應用程式要使用的語言和多位元組字集。這個設定是藉由設定代理程式環境定義的 LANG 環境變數來運作。

如果要將代理程式設為將指令輸出視為美式英文 UTF-8，請在您的作業系統中使用 UTF-8 語言環境。例如，在 Linux 中使用下列的表示法。


```
locale en_US.UTF-8
```

如果要判斷作業系統 UTF-8 語言環境的正確表示法，請執行 **locale -a** 指令。

如果未指定這項設定，代理程式會使用作業系統的語言環境。這項設定相當的方便。如果作業系統的預設語言環境不是代理程式要使用的語言環境，這項設定特別有用。如果變更系統語言環境來符合代理程式需求的做法並不實際，這項設定特別有用。

magic_login user:encoded-password

代理程式通常會使用管理專用權（如：root 使用者或管理者）來登入作業系統。您可以改用 **magic_login** 設定，來進行標準的系統鑑別。利用這項設定，系統可以透過單一使用者名稱與密碼來鑑別您的登入。

如果是以 **root** 或具有管理權的使用者來執行代理程式，會忽略這項設定，並嘗試正常的鑑別。

代理程式在執行任何指令時，所用的是啟動代理程式的使用者（而不是登入時所用的使用者名稱）的許可權。

這項設定只能用於以下情況：

- 無法以管理專用權來執行代理程式。例如，針對未使用 PAM 的 UNIX 系統使用這項設定。
- 基於安全原則，不允許以管理專用權來執行代理程式。

如果要配置代理程式的登入資訊：

1. 建立使用使用者名稱和密碼的伺服器鑑別。請在「管理主控台」中，按一下**伺服器 > 伺服器鑑別**。
2. 以本例來說，使用者名稱為 **build**，密碼為 **MySecretPassword**。
3. 建立一個使用代理程式的伺服器。請將伺服器鑑別與**鑑別**欄位中的這部伺服器相關聯。
4. 為代理程式產生一個編碼密碼。請在代理程式的安裝目錄中，以您選擇的密碼來執行 **bfagent -P**。

會傳回一個以 SMD5 雜湊編碼的密碼，如下所示：

```
bfagent -P "MySecretPassword"
eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a
```

5. 在 **BFAgent.conf** 中，設定 **magic_login** 以使用您想要的使用者名稱與編碼密碼。

```
magic_login build:eca0b7f2f4fbf110f7df570c70df844e1658744a4871934a
```

6. 啟動代理程式。
7. 測試伺服器連線。在**伺服器**中選取伺服器，然後按一下**測試伺服器**。

map drive-and-user-spec[; ...]

這項設定會指定對映磁碟機。某些系統可能需要磁碟機對映。例如，由於 shell 是從共用磁碟機執行，所以可能需要磁碟機對映。將會先執行在代理程式上指定的對映，然後才執行「管理主控台」中以 **_MAP** 環境變數所指定的對映。本例說明兩個磁碟機對映：

```
map X:="//host1/share;Z:="//host2/share(username,password)
```

map_drive_is_failure

如果指定，這項設定會在執行步驟之前遇到無法對映的磁碟機規格時，導致步

驟失敗。如果未指定這項設定，步驟會忽略磁碟機失敗，而嘗試執行步驟。在這種情況下，請確定失敗會產生有意義的錯誤訊息。

no_preparse_command

這項設定會停用變數擴充剖析；代理程式通常會在將指令傳給 Shell 之前，對指令執行這個動作。另請參閱 `_NO_PREPARSE_COMMAND` 環境變數，它適用於單一專案或步驟。

no_pty

這項設定只能用於在 UNIX 或 Linux 系統上執行的代理程式。

使用這項設定有助於在 Shell 與代理程式的虛擬終端機互動時，避免系統 Shell 被鎖定。對於 HP-UX 及 z/OS，通常使用這項設定。您也可以使用兩種其他的方法來協助避免這類鎖定：

- 使用替代 Shell
- 使用 `nologonshell` 設定

no_pty 設定會停用虛擬終端機配置。

註：使用 **no_pty** 會影響某些指令。例如，**ls** 指令會以一欄傳回輸出，而非三欄。如果您使用這項設定，請先完整測試，再將工作部署到正式作業環境。

nologonshell

這項設定僅適用於在 UNIX 或 Linux 上執行的代理程式。

這項設定會使得代理程式執行的 Shell 成為一般 Shell，而非登入 Shell。這項設定在以下情況下通常很有用：

- 登入 Shell 提供了詳細輸出
- 登入 Shell 以不需要的方式變更環境設定
- 登入 Shell 嘗試以互動方式與使用者進行通訊

如果設定，則會使用要求 Shell 為一般 Shell 而非登入 Shell 的標準方法。這不一定適用於所有平台；如果不適用，可使用 `shellflag` 設定，將旗標傳給 Shell，以修改其行為。

這些都不是代理程式想要的行為，因為它的執行身分為一般使用者，而非互動式使用者。

註：Mac OS X 10.5 系統會使用 `/bin/bash`，其不會回應 `nologonshell`。請使用 `shellflag -l`。

註：z/OS 作業系統對登入 Shell 和非登入 Shell 兩者都一律使用 `/etc/profile` Script。如果其行為不適用於代理程式，您可能需要變更 Script 的內容或使用另一個 Shell。

另請參閱 **shellflag** 設定。您可以使用旗標來變更登入 Script 的行為。

password_encrypt_module *dll_path;conf_path*

在代理程式上啟用 SSL 需要這項設定。其會指定 DLL 和配置檔的路徑。

- *dll_path* 是 `bfcrypt.dll` 的路徑（其通常是 `./bfcrypt.dll`）。
- *conf_path* 是 `bfpwcrypt.conf` 的路徑（其通常是 `./bfcrypt.conf`）。

port *port-number-or-range* [...]

這項設定只能用於在 UNIX 或 Linux 上以獨立式模式執行的代理程式（啟動時發出 **-s** 選項）。

這項設定會指定代理程式用來接聽「管理主控台」連線的埠。

指定代理程式用來接聽「管理主控台」連線的連接埠。

註：依預設，連接埠會設為 5555，如果是 UNIX 或 Linux，此設定位於 `/etc/services` 中。

read_timeout

代理程式等待要求直到它中斷連線的時間（以秒為單位）。預設值是 1800 秒（30 分鐘）。將值設為 0，表示停用逾時。

此指引可以協助用戶端連線聯絡人，如果未收到合法的引擎要求，就不讓埠保持開啓狀態。部分網路埠掃描軟體是以這種方式來運作。

請不要對此指引設定非常小的值。一般的引擎行為會在要求之間包含數分鐘的差距。

shell *shell_name* [*options*]

這項設定會指定預設 Shell。內部預設值如下所示：

- Windows：shell `cmd.exe /q /c "%s"`，除非使用下列設定：
 - 如果使用 cygwin 設定，則預設值是 shell `C:\cygwin\bin\bash.exe --login -c "%s"`
 - 如果未使用 cygwin 設定，預設值便是 shell `cmd.exe /u /q /c "%s"`
- UNIX 或 Linux：設定給使用者帳戶的 Shell；如果無法判斷使用者的 Shell，則使用 `/bin/sh`。請注意，您不能在這項設定中指定參數，但您可以使用 `shellflag` 設定來傳遞這些參數。代理程式會自動在預設值中強制插入連字號以作為登入 Shell。例如，`/bin/ksh` 會傳送為 `-ksh`。如果已明確設定了 Shell，則同時會隱含地設定 `nologonshell`。請參閱 `nologonshell`。
- *System i*：將 Shell 值設為 `/bin/sh`

您可以在步驟內置換這項設定。從內含 `#!` 之字行開始的步驟，會置換 Shell 設定，並使用 `nologonshell` 設定來執行步驟指令。

shell_compatible_undef_vars

這項設定會強制以空字串表示未定義的變數。若未設定，`$VAR`、`${VAR}` 或 `%VAR%` 格式的變數會表示為變數名稱，`[$VAR]` 則表示為空字串。

shellarg

這項設定只能用於在 UNIX 或 Linux 上執行的代理程式。

如果指令似乎會鎖碼，請使用這項設定。Red Hat Linux Enterprise 上的某些 Shell 需要這項設定。

該設定會變更指令 Script 傳給 Shell 的方式。通常 Script 會透過標準輸入來傳送：

```
/bin/sh < /tmp/bfshellscript.sh
```

這項設定是在參數中傳遞 Script，從而執行 Script：

```
/bin/sh /tmp/bfshellscript.sh
```

shellflag *flag*

這項設定只能用於在 UNIX 或 Linux 上執行的代理程式。

這項設定會在 Shell 執行時加入旗標。但只能指定一個旗標。它通常用來停用 rc Script 處理，以便降低輸出或不需要的處理。範例：

- csh 與衍生項：使用 `shellflag -f` 來停用 rc Script 處理。
- bash：使用 `shellflag --noprofile` 來停用設定檔 Script 處理。

ssl_ca_location *path*

指定含有憑證管理中心的金鑰儲存庫檔。如果代理程式是當成服務來執行，請使用絕對路徑。

ssl_cert_location *path*

指定含有私密憑證的金鑰儲存庫。如果代理程式是當成服務來執行，請使用絕對路徑。

ssl_client_authentication [true | false]

設為 true 表示在建立代理程式的連線時，需要有用戶端鑑別。如果為 true，Build Forge 引擎的憑證必須加入代理程式的憑證管理中心金鑰儲存庫中。

ssl_cipher_group [*grouplist* | ALL]

指定要使用的個別密碼群組。其可以設為 ALL。

ssl_cipher_override *cyphers*

置換密碼群組。請指定要使用的密碼。

ssl_key_location *path*

指定包含金鑰的金鑰儲存庫檔。如果代理程式是當成服務來執行，請使用絕對路徑。

ssl_key_password *password*

金鑰的密碼。依預設，這個內容會以明碼儲存。您可以配置代理程式，利用它自己的金鑰或 Build Forge 伺服器的金鑰來加密此密碼。

ssl_protocol *protocol*

要使用的 SSL 信號交換通訊協定，TLSv1、TLSv1.1、TLSv1.2 中的一個。該通訊協定必須符合 Build Forge[®] 伺服器所使用的通訊協定。啟用之後，TLSv1、TLSv1.1 或 TLSv1.2 僅接受對應的連線。例如，啟用 TLSv1.2 之後，bfagent 僅接受 TLSv1.2 連線。

update_path *path*

這項設定會識別指向 Build Forge 代理程式執行檔的完整路徑。在安裝期間會自動建立這項設定。其目錄是作業系統的預設目錄，或是您所指定的安裝目錄。

註：在 Windows 代理程式中，會忽略這項設定。更新路徑會從登錄機碼取得。這些機碼會在代理程式安裝期間設定。

win_reexec_after_auth

如果您需要利用 Build Forge 伺服器鑑別認證，對網路共用檔案系統執行代理程式指令，請新增這項設定。例如，當在 ClearCase 動態視圖中修改檔案時，代理程式必須存取網路共用檔案系統中的 ClearCase 檔案。

Build Forge 代理程式一開始是以 Windows 系統帳戶認證來啟動。為了執行指令，之後代理程式會使用 Build Forge 伺服器鑑別認證向 Windows 進行鑑別。

如果不使用這項設定，網路共用區只能辨識最初的 Windows 系統帳戶認證，且會忽略在存取及寫入網路共用檔案系統上的檔案時，所需的後續伺服器鑑別認證。

在使用伺服器鑑別認證重新向 Windows 進行鑑別之後，`win_rexec_after_auth` 會啟動新的程序，並強迫共用檔案系統承認已變更的認證。

當您使用 `win_rexec_after_auth` 設定時，代理程式會當成服務來執行，而不會區別指令是否存取網路共用檔案，因此您可能需注意對效能的影響。

xstream_allow_ssl_mismatch

如果使用 OpenSSL 來編譯的代理程式與未使用 OpenSSL 來編譯的代理程式之間需要檔案傳送，則需要此設定。依預設，使用 OpenSSL 來編譯的代理程式需要 AES_CBC 加密的檔案傳送。除非使用此設定，否則它們會拒絕任何使用 PLAIN 或 PRNG 編碼所要求的檔案傳送。

xstream_bind *ip_address*

指定只用於直接檔案傳送的 IP 位址。接收檔案的代理程式必須可以存取此位址。依預設，代理程式會在所有網路介面進行接聽。另請參閱[連結](#)。

xstream_conn_timeout *seconds*

代理程式等待連線的時間（以秒為單位）。引擎必須將連線要求轉遞至接收代理程式，而且接收代理程式必須在此時間內與傳送代理程式建立連線。依預設，它會設為 20 秒。

xstream_listen_range *port-range*

為埠範圍，代理程式會在其上接聽是否有連線。當連接主機之間有防火牆時，此設定很有用。防火牆管理者可以配置防火牆，讓允許的埠可以進行連線，例如 22880-22889。預設埠範圍為 16384-32767。不過，如果使用 `xstream_bind`，但未使用 `xstream_listen_randomize`，則代理程式不會指定範圍，而且作業系統會決定要使用的埠。

xstream_listen_randomize

會導致在 `xstream_port_range` 內隨機選取一個埠。如果沒有指定，代理程式會從最低的埠號開始檢查。高度建議使用此設定作為安全措施。

xstream_recv_timeout *seconds*

等待檔案傳送的時間。在檔案傳送期間，如果過了這段期間，接收代理程式仍未從傳送代理程式取得資料，則傳送會失敗，並且會關閉連線。預設值是 20 秒。

xstream_send_timeout *seconds*

等待檔案傳送的時間。在檔案傳送期間，如果過了這段期間，傳送代理程式仍無法將資料寫入至接收代理程式，則傳送會失敗，並且會關閉連線。預設值是 20 秒。

如果已啓用密碼加密

如果配置內容檔 `bfpwcrypt.conf` 已啓用密碼加密，將 SHA1 變更為 SHA2 之後，請使用此主題中的步驟。

程序

1. 導覽至**管理 > 安全**，停用密碼加密。
2. 如果在啓用密碼加密之後，新增或更新了「伺服器授權」、「LDAP」及「使用者」密碼，請移至**管理 > 安全 > 金鑰儲存庫**，然後再次更新每一個密碼。

3. 重新命名密碼加密配置內容檔。例如，將 `bfpwcrypt.conf` 檔案的名稱變更爲 `bfpwcrypt.conf.sha1`
4. 返回**管理 > 安全**，將「摘要演算法」變更爲 SHA2。
5. 重新啓動 Build Forge® 管理主控台。這會建立新的 `bfpwcrypt.conf` 檔案。
6. 返回**管理 > 安全**，啓用密碼加密。
7. 選擇性的：返回**管理 > 安全 > 金鑰儲存庫**，然後更新「伺服器授權」、「LDAP」及「使用者」密碼。
8. 如果代理程式也啓用了密碼加密，請將新產生的 `bfpwcrypt.conf` 檔案複製到代理程式所在的伺服器。
9. 如果 `ssl_key_password` 已加密，請重新加密密碼。例如，`re-encrypt bfaagent -e xxxxxx`。在本例中，`xxxxxx` 代表密碼。
10. 在 `bfaagent.conf` 檔案中，透過移除行前的 `#` 或新增行來啓用 `digest_algorithm SHA2`。
11. 重新啓動 `bfaagent`。

如果未啓用密碼加密

如果配置內容檔 `bfpwcrypt.conf` 從未啓用密碼加密，將 SHA1 變更爲 SHA2 之後，請使用此主題中的步驟。

程序

1. 重新命名密碼加密配置內容檔。例如，將 `bfpwcrypt.conf` 檔案的名稱變更爲 `bfpwcrypt.conf.sha1`
2. 移至**管理 > 安全**，將「摘要演算法」變更爲 SHA2。
3. 重新啓動 Build Forge® 管理主控台。這會建立新的 `bfpwcrypt.conf` 檔案。
4. 選擇性的：移至**管理 > 安全**，啓用密碼加密。
5. 選擇性的：移至**管理 > 安全 > 金鑰儲存庫**，然後更新「伺服器授權」、「LDAP」及「使用者」密碼。
6. 如果代理程式也啓用了密碼加密，請將新產生的 `bfpwcrypt.conf` 檔案複製到代理程式所在的伺服器。
7. 如果 `ssl_key_password` 已加密，請重新加密密碼。例如，`re-encrypt bfaagent -e xxxxxx`。在本例中，`xxxxxx` 代表密碼。
8. 在 `bfaagent.conf` 檔案中，透過移除行前的 `#` 或新增行來啓用 `digest_algorithm SHA2`。
9. 重新啓動 `bfaagent`。

對代理程式進行疑難排解

本節說明您可以用來對無法正確運作的代理程式進行疑難排解的程序。請依順序執行這些程序。如果利用這些程序仍無法使代理程式運作，請洽詢「支援中心」。

測試主機名稱解析

驗證「管理主控台」主機是否能夠呼叫到代理程式主機。請從「管理主控台」主機使用 `ping` 公用程式來測試代理程式主機：

```
ping hostname
```

這個範例階段作業會在同時安裝「管理主控台」和代理程式的 Windows 上執行。

```
C:\> ping localhost
```

```
Pinging somehost.city.company.com [127.0.0.1] with 32 bytes of data:
```

```
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

```
Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

類似如下所示的訊息則表示發生問題：

```
Unknown host
```

問題出在「管理主控台」主機的網路配置。請洽詢您的網路管理者。

測試連線

您可以使用 telnet 或「管理主控台」中的測試，來測試與代理程式之間的連線。

如果要從指令行測試連線，請執行下列動作：

1. 利用 telnet 指令連接代理程式。如果您已登入正在執行代理程式的主機，您可以使用 localhost 作為主機名稱。

```
telnet hostname 5555
```

這個回應表示連線成功：

```
200 HELLO - BuildForge Agent v7.0.1.buildnumber
```

2. 利用您的登入認證，發出下列指令來檢查鑑別：

```
telnet localhost 5555
username user name
password password
cmd ping
go
```

下列訊息表示成功：

```
AUTH: set user account to <user name>
```

如果可以進行上述測試但工作失敗，且伺服器的測試顯示使用者鑑別錯誤，則請檢查可外掛鑑別模組 (PAM) 的配置。如果您看到如下訊息，則請繼續下一步驟。

```
AUTH: unable to set user account to <user name>: unknown account (1)
```

3. 輸入下列指令：

```
cmd ping go
```

以下是 telnet 階段作業的一般輸出。請特別尋找輸出結尾的 RESULT 0 代表成功。這項測試是針對 Windows 上執行的代理程式來執行。

```
300 DATA s 67
AUTH: Running as: [SYSTEM] in domain [NT AUTHORITY] SID Type [User]
300 DATA s 52
AUTH: Running with Privilege: [Lock pages in memory]
300 DATA s 66
AUTH: Running with Privilege: [Adjust memory quotas for a process]
```

```
300 DATA s 63
AUTH: Running with Privilege: [Create permanent shared objects]
300 DATA s 46
AUTH: Running with Privilege: [Debug programs]
300 DATA s 56
AUTH: Running with Privilege: [Bypass traverse checking]
300 DATA s 61
AUTH: Running with Privilege: [Back up files and directories]
300 DATA s 54
AUTH: Running with Privilege: [Change the system time]
300 DATA s 68
AUTH: Running with Privilege: [Remove computer from docking station]
300 DATA s 73
AUTH: Running with Privilege: [Impersonate a client after authentication]
300 HEARTBEAT 1
300 DATA s 16
PLAT: Windows XP
250 RESULT 0
PING: internal loopback test complete
260 EOR
```

如果要從「管理主控台」測試連線，請執行下列動作：

註：只有當伺服器已經設為使用代理程式時，才能使用此方法。

1. 跳至**伺服器**。
2. 在伺服器的清單中，按一下要測試的伺服器。
3. 按一下**測試連線**。

測試之後，結果會顯示在**測試結果**標籤中。

與下列輸出類似的訊息表示有問題。

```
Could not open a connection to host on port 5555
```

可能是「管理主控台」無法連接至主機，或是代理程式無法運作。

在 Windows 上進行代理程式疑難排解

若要在 Windows 上進行代理程式疑難排解，請執行下列動作：

1. 檢查所安裝的執行檔。確認以下檔案存在於代理程式安裝目錄中：
 - bfaagent.exe
 - bfdispatch.exe

註：繼續之前，請先決定是否有自訂配置。如果有自訂配置，請先將 BFAgent.conf 儲存在安裝目錄外，然後在安裝之後再加以還原。

2. 重新安裝代理程式。重新安裝代理程式即可解決大部分 Windows 代理程式問題。重新安裝會更新執行檔並還原登錄機碼。

在 UNIX、Linux 或 MacOS 上對代理程式進行疑難排解

如果要在 UNIX、Linux 或 MacOS 上對代理程式進行疑難排解，請嘗試以下程序：

- 從 Shell 執行 **bfaagent**。正確的回應類似以下訊息：

```
200 HELLO - Build Forge Agent v7.0.1.122
```

如果您收到類似範例的訊息且發生共用程式庫問題，您會收到這些問題的相關訊息。您可以透過正確地設定路徑，來解決大部分共用程式庫問題。

- 檢查代理程式是否正在接聽。請使用下列指令（假設埠為預設值 5555）：

```
telnet localhost 5555
```

200 HELLO 的回應表示代理程式正在接聽中。如果您未取得這項回應，請檢查系統網路配置。請驗證 **inetd** 配置是否正確，或是洽詢 Linux 或 UNIX 系統管理者。

- 檢查鑑別內容。請使用您的登入認證發出下列指令：

```
telnet localhost 5555
username <user name>
password <password>
cmd ping
go
```

與下列訊息類似的訊息表示鑑別已正常運作：

```
AUTH: set user account to <user name>
```

如果可以進行之前的測試但建置失敗，且伺服器的測試顯示使用者鑑別錯誤，則請檢查可外掛鑑別模組 (PAM) 的配置。如果您看到與下列訊息類似的訊息，請繼續進行下一個程序。

```
AUTH: unable to set user account to user name: unknown account (1)
```

- 檢查 PAM 配置。PAM 配置的問題是 AIX 平台上的常見問題。視您的作業系統而定，PAM 會以兩種方式之一進行配置：以 `pam.conf` 檔案中的一行，或以 `pam.d` 目錄中的一個檔案。

提示： Solaris 10 是下列程序的例外：請刪除用來指定 `pam_dial_auth` 模組的任何字行，例如：`pam_dial_auth.so.1`。如果含有該模組，代理程式鑑別將無法運作。

1. 驗證 `/etc/pam.conf` 已存在。如果不存在，請參考本主題之後關於 `pam.d` 的指示。如果該檔案存在，請繼續進行下一步。
2. 在檔案中建立 `bfagent` 的項目。
3. 複製另一個應用程式的字行（例如：`sshd` 或 `login`），然後將 `[application]` 欄位替代成 `bfagent`。
[application] [when] [mode] [module]

欄位如下所示：

- `[application]` 是需要鑑別使用者的應用程式名稱
- `[when]` 是鑑別要求的類型
- `[mode]` 是鑑別要求的模式
- `[module]` 是要呼叫的鑑別模組。下列範例顯示從 `login` 複製到 `bfagent` 的項目。請注意，不同系統的模組名稱可能會有所不同。

```
bfagent auth requisite      pam_authtok_get.so.1
bfagent auth required      pam_dhkeys.so.1
bfagent auth required      pam_unix_cred.so.1
bfagent auth required      pam_unix_auth.so.1
```

4. 設定好 PAM 項目之後，請依步驟 3 的說明嘗試重新登入。
 5. 如需相關資訊，請參閱 PAM 文件：<http://www.sun.com/software/solaris/pam>。
- 如果要對 `pam.d` 中配置的 PAM 進行疑難排解，請執行下列動作：

1. 找出 `/etc/pam.d` 目錄；請注意它含有幾個檔案，每一個都指名一個應用程式。在每一個檔案內，每一行的格式皆為：
`[when] [mode] [module]`
2. 從另一個應用程式（如：`sshd` 或 `login`）複製檔案，並將它重新命名為 `bfagent`。
3. 設定好 PAM 項目之後，請依步驟 3 的說明嘗試重新登入。
4. 如需相關資訊，請參閱 PAM 文件：<http://www.sun.com/software/solaris/pam/>。

第 11 章 後置安裝作業

本節說明在順利安裝之後所執行的作業。

其中將討論下列主題：

- 啟動與停止引擎
- 設定使用者
- 驗證安裝架構
- 一般問題的疑難排解

啟動與停止引擎

下列各節說明如何啟動及停止引擎。

在 Windows 上啟動和停止引擎

在 Windows 上：

- 在**開始 > 程式集 > IBM Rational Build Forge 管理主控台**中，您必須選取下列其中一項：
 - 啟動引擎服務
 - 停止引擎服務

控制台：**系統管理工具 > 服務控制台**選項也可用來啟動或停止 **IBM Rational Build Forge 管理主控台服務**。

在前景中執行：如果您在執行引擎時發生問題，則可以在前景中執行它。它會顯示以下產生的狀態及錯誤訊息：選取**開始 > 程式集 > IBM Rational Build Forge 管理主控台 > 啟動引擎（前景）**。當「管理主控台」正在執行時，主控台視窗中會顯示日誌輸出。若要在這個模式下停止引擎，請在主控台視窗中按下 **Ctrl-C**。

在 UNIX or Linux 系統上啟動和停止引擎

如果已安裝 rc 檔，請使用下列指令來啟動與停止本產品：

```
$ /opt/buildforge/rc/buildforge start
$ /opt/buildforge/rc/buildforge stop
```

如果尚未安裝 rc 檔，請使用下列指令來啟動本產品：

```
$ /<bfinstall>/Platform/buildforge &
```

如果要停止，請判斷其處理程序 ID，然後發出 kill 指令：

```
$ ps aux | grep buildforge
$ kill ${<PID>}
```

設定使用者

本節說明如何在 Build Forge[®] 系統中設定使用者帳戶。

將討論下列主題：

- root 使用者
- 新增使用者帳戶
- 適用於報告的唯讀使用者

root 使用者

如果您以 *root* 使用者身分登入，您將具有特殊專用權及特質。下列顯示這些專用權及特質：

- **於安裝時建立：**root 使用者是安裝程式所建立的唯一預設使用者。預設密碼為 *root*。

註：您必須在安裝之後立即變更密碼。

- **不需要授權：**root 使用者不會耗用使用者授權。無論已登入的使用者數目為何，您一律能夠以 root 使用者身分登入。

註：當您嘗試登入時，如果目前已有其他人以 *root* 使用者身分登入，則他們會自動登出。

- **系統時區：**root 使用者時區為「管理主控台」的預設時區。依預設，系統內及 LDAP 使用者的時區與 root 使用者時區相同。登入之後，使用者可以設定他們自己的時區。系統中所報告的所有時間和日誌都會以使用者時區來表示。
- **所有許可權：**root 使用者擁有所有可用的許可權，且可以編輯其他使用者內容。

註：無法移除 root 使用者存取權。Root 使用者不是任何存取群組的成員，但能夠檢視、編輯或使用系統中的任何資料物件。

- **優先順序：**root 使用者為優先使用者。
- **登出現行使用者：**當 root 使用者按一下**登出使用者**時，即可登出使用者。按一下**管理 > 使用者**及使用者名稱，以完成此程序。

建立及編輯使用者

您可以利用**管理 > 使用者**來建立使用者並指派內容。您也可以讓系統連接 LDAP/Active Directory 資料庫，以取得使用者資訊。您可以將使用者指派給群組，以管理使用者的安全許可權。建議您建立少數使用者來測試安全特性。

按一下**管理 > 使用者**，來顯示現行使用者的清單。清單後面會顯示使用者畫面。系統會顯示每位使用者的名稱、登入、電子郵件、限制、活動（自前次使用者活動以來的經歷時間）及時區。

- 若要編輯使用者：按一下使用者名稱、編輯內容，然後按一下**儲存**。
- 若要建立新的使用者：請在未選取使用者的情況下，於使用者畫面中指定內容。如果已選取使用者，則按一下**新增使用者**來清除畫面，並於編輯使用者資訊之後按一下**儲存**。
- 若要登出使用者：按一下使用者名稱及**登出使用者**。
- 若要釋出固定授權位置：以 **root** 身分登入。按一下使用者名稱及**清除名額**。主控台即會從計入這組固定授權的 ID 清單中，移除該使用者。如果該使用者已經登入，也會一併被登出。主控台在計算固定授權時，是以曾經登入的使用者數量來計算。當

到達限制時，新的使用者就無法取得授權，必須刪除或清除現有的使用者，才能釋出固定授權供另一個使用者使用。清除名額，並不會從主控台刪除這個使用者。如果該使用者再次登入，固定授權計數便會增加。如果您對浮動授權使用者使用**清除名額**，該動作的效果與按一下**登出使用者**相同。

- 若要刪除使用者帳戶：按一下使用者名稱及**刪除**。

如果**刪除**已停用，則表示該使用者帳戶擁有排定的工作。若要刪除具有已排定工作的使用者帳戶，您必須先刪除排定的工作。

使用者記錄會設定預設內容，制定使用者與系統互動的方式。這也可以控制使用者登入名稱、密碼及密碼有效期限。您可以透過「管理主控台」提供使用者記錄的資料給系統，也可以從 LDAP/Active Directory 資料庫中擷取資料。

註：當您編輯使用者，而其記錄是從 LDAP 資料庫中擷取時，「使用者」頁面上的許多欄位都會停用。您可以在來源資料庫中變更這些內容。

若要新增使用者：按一下**新增使用者**、編輯畫面，然後按一下**更新**。

使用者記錄會顯示下列三個標籤：

- **詳細資料：**顯示可編輯的使用者內容。這個主題後面會說明可用的內容。
- **現行群組：**顯示您透過另一個群組的一個成員群組而成為其成員的存取群組。
- **變更群組：**顯示您為其直接成員的群組，可讓您新增或移除群組。

針對每一個使用者，您可以在**詳細資料**標籤中設定下列內容：

名稱 指定使用者的顯示名稱和標籤。

電子郵件

指定系統可以用來傳送電子郵件通知給此使用者的電子郵件位址。

註：電子郵件只會傳送給明確選出要通知的使用者。

使用者名稱

指定使用者輸入以登入「管理主控台」的名稱。

密碼 輸入使用者提供以登入「管理主控台」的密碼。已登入的使用者不會顯示這個欄位。請使用這個欄位來輸入新的密碼或變更密碼。請在**驗證**欄位中輸入同一密碼。

限制 指定使用者一天內所能啟動的工作數上限。到達限制時，系統會顯示訊息指出已超出使用者執行配額。

時區 指定使用者時區。系統會使用 **root** 使用者的時區作為所有顯示時間的預設時區。

依預設，系統內使用者與 LDAP 使用者都會被指派與 **root** 使用者相同的時區。使用者可以編輯為其指派的時區。

註：從舊版升級至 Build Forge 7.1 時，您需要手動重設 **root** 使用者的時區。

驗證 重新輸入密碼，以確定您第一次輸入的密碼正確。

優先登入

指定使用者是否為優先使用者。優先使用者可隨時登入系統。如果已沒有可用的使用者授權，系統會將擁有最舊階段作業的使用者登出，讓優先使用者登入。**root** 使用者一律為優先使用者。

日期格式

設定使用者偏好的日期格式。

語言 設定使用者偏好的語言。

密碼到期

指定使用者密碼會到期。如果選取這個選項，則經過幾天之後，使用者密碼就會到期。此資訊是在**密碼到期天數**系統設定中指定。

使用螢幕閱讀器

讓介面支援螢幕閱讀器特性，例如，動態強調顯示與定焦。

行事曆的一週開始日

選取「排程」行事曆要最先顯示星期幾，預設值為**星期日**。

驗證安裝架構

本節說明如何測試已安裝和配置的 Build Forge® 系統。

其中將討論下列主題：

- 配置伺服器
- 建立測試專案
- 執行專案

伺服器鑑別

使用伺服器鑑別來建立登入認證與伺服器的關聯。藉由管理一組伺服器鑑別，您可以對許多伺服器使用相同的認證，也可以全面更新認證。

伺服器鑑別會將登入名稱與密碼儲存成已命名的單一物件，可讓您將該物件與一或數部伺服器產生關聯。請使用「伺服器鑑別」頁面來建立及編輯伺服器鑑別。

配置伺服器

為了使系統中可以動態選取的伺服器數目達到最多，您必須依特定順序來設定一些資料物件。

本主題簡述使用伺服器的最低需求。

1. 建立伺服器鑑別。

伺服器鑑別會提供伺服器的登入名稱與密碼。您可以對多部伺服器套用伺服器鑑別。

註：不需要單獨登入每一部伺服器。如果您決定變更登入，您可以對一組系統進行變更。

2. 建立伺服器群組的收集器。

收集器會從伺服器收集已指派的內容。您可以使用不具收集器的伺服器。您必須根據 *BF_NAME* 之類的預設內容來選取伺服器。

3. 建立選取元。

請考慮建立下列類型的選取元：

- 名稱型：為每部伺服器建立一個選取元，以根據其主機名稱來選取伺服器。您可以依名稱來選取伺服器。
 - 作業系統型：為您環境中每個類型的作業系統建立一個選取元。您的專案可以依作業系統來選取伺服器。
 - 容量型：您可以更具體地根據可用的 RAM 或硬碟空間來選取伺服器。
4. 在您計劃用作 Build Forge 伺服器的每部電腦上安裝代理程式。
 5. 在「管理主控台」中，為您計劃與 Build Forge 搭配使用的每部電腦建立伺服器。
 6. 測試伺服器。
 7. 按一下**測試連線**，以測試您的伺服器連線。

請檢閱它們的資訊清單，確定其中的內容符合您的預期。

建立測試專案

如果要驗證 Build Forge® 系統是否能正常運作，請如第 10 頁的『建立 Hello World 專案』中所述，建立並執行簡式專案。


執行專案

有數種不同方式可以啟動專案。

開始之前

這項作業假設您已經建立選取元、伺服器和專案。

程序

- 在檢視專案的清單時，按一下任何專案前面的**快速啟動**  圖示，以立即啟動專案。如果專案中沒有任何步驟，或是包含任何在「於專案」動作為**必須變更**的環境變數，您就無法使用此方法。以這種方式來執行專案，會使用其選取元、類別、標籤及環境變數的預設值。
- 當檢視專案步驟時，按一下**啟動專案**。此方法會顯示專案的「啟動專案」頁面，您可以在其中變更專案參數、環境變數值，並選取要從作業中排除的步驟：
 - 選取新的專案參數值。
 - 編輯專案標籤變數值。
 - 編輯專案環境變數值。如果您想將變更內容儲存為這些變數的新預設值，請按一下**儲存環境**勾選框。
 - 選取**工作步驟**標籤以顯示專案步驟清單。您可以選取個別步驟，只從這個作業中排除該步驟。

當您做好選擇時，請按一下**執行**，來啟動專案。

- 選取**工作 > 啟動**，然後按一下專案名稱。如同使用**啟動專案**一樣，此方法會顯示「啟動專案」頁面。

工作詳細資料

工作步驟

專案參數

專案環境

儲存環境 ☐

Snapshot: 基本 Snapshot
選取元:
類別:
標籤格式:
標籤範例: BUILD_4

專案標籤 ☐ 可編輯的標籤
B

結果

正在執行專案時，請檢視工作 > 執行中頁面，檢查專案狀態。

如果要檢視工作結果，請選取工作 > 已完成來顯示已完成的工作。按一下「標籤名稱」即可存取用來檢視工作結果的選項。

日誌的位置與設定

Rational Build Forge 由多個元件組成。每個元件都有各自的日誌。

Build Forge

<bfinstall>/db.log

Apache

<bfinstall>/Apache/logs

Apache Tomcat

<bfinstall>/Apache/tomcat/logs。Tomcat 是應用程式伺服器，用來執行服務層 (jas.war) 及說明 (BuildForgeHelp.war)。

PHP <bfinstall>/db.log。PHP 是用來執行使用者介面。

日誌輪轉

在某些狀況下，元件會撰寫大量資料。對於 Tomcat 所產生的 catalina 日誌檔來說尤其是如此。根據設計，Tomcat out 日誌沒有輪轉。在少數情況下，out 日誌的成長幅度可能超出預期。

有兩個方法可解決此問題：

- 在伺服器的環境定義中使用 swallowOutput 屬性。輸出會重新導向至轉輪的 Web 應用程式日誌。如需詳細資料，請參閱 Apache Tomcat 配置參照。
- 修改 Catalina 啟動 Script。特性會根據您的喜好設定而不同，但是下列管道 Catalina 輸出會使用 cronolog 功能來輪轉這些喜好設定。修改 <bfinstall>/Apache/tomcat/bin/catalina.sh 以包含下列或類似指令：

```
start >> catalina.out 2>&1 & --replaced with
```

```
start |cronolog /tomcat/logs/%Y-%M-%D.catalina.out >> /dev/null 2>&1 &
```


相關概念:

第 525 頁的第 39 章,『診斷』

本節協助您判斷 Build Forge 無法正常運作的原因。

一般問題的疑難排解

在 Solaris 升級代理程式需要執行 **pkgrm** 指令

請先使用 **pkgrm BFAgent** 指令來移除現有的 Solaris Build Forge 代理程式,再執行 **pkgadd** 指令。

7.0 通知範本的 URL 可能無法在較新版本中運作

當您按一下通知電子郵件中的 URL 鏈結時,通知範本 URL 會開啓 Build Forge 工作報告。

在 7.0.1 和 7.1 版中,通知範本中的 URL 已變更;因此,當您從 7.0 升級至最新版時,URL 可能無法運作。

如果您遇到鏈結錯誤,請完成以下步驟來手動編輯通知範本:

1. 選取專案 > 範本。
2. 按一下通知範本名稱,以便在「詳細資料」標籤中顯示其內容。
3. 在「內文」欄位中,尋找範本的 URL。此 URL 應該類似下列範例中的 URL:

```
http://${CONSOLEHOST}:${CONSOLEPORT}/fullcontrol/index.php?mod=projectruns&action=edit&bfid=${PID}&bfid=${BID}&bfid=${UID}
```

4. 以適當的 7.0.1 以及更新版本的 URL 元素,取代下列 URL 元素:

URL 元素	7.0.1 (以及更新版本) URL 元素
projectruns	jobs
&	&
action=edit	action=build.view
&bfid=\${PID} &bfid=\${BID} &bfid=\${UID}	&bf_id=\${BID}

產品沒有回應

如果 Rational Build Forge 沒有回應,請查看 `<bfinstall>/Apache/tomcat/logs/` (UNIX 或 Linux) 或 `<bfinstall>\Apache\tomcat\logs\` (Windows) 中的日誌,找出類似下列內容的訊息:

```
Services: 20075: CRRBF1381I: Established connection to Build Forge Services.
DBD::DB2::st execute failed: [IBM][CLI Driver] SQL30081N A communication error has been detected.
Communication protocol being used: "TCP/IP". Communication API being used: "SOCKETS".
Location where the error was detected: "XXX.XXX.XXX.XXX".
Communication function detecting the error: "recv".
Protocol specific error code(s): "131", "*", "0". SQLSTATE=08001
August 17, 2010 7:18:14 AM EDT
Database: 20075: CRRBFEEEE: DBD::DB2::st execute failed:
[IBM][CLI Driver] SQL30081N A communication error has been detected.
Communication protocol being used: "TCP/IP".
Communication API being used: "SOCKETS".
Location where the error was detected: "XXX.XXX.XXX.XXX".
Communication function detecting the error: "recv".
Protocol specific error code(s): "131", "*", "0". SQLSTATE=08001
```

```

Database: 20075: CRRBF0551I: StackTrace from Process id [20075] called from [
  BuildForge::DB::db2(/PerlApp/BuildForge/DB/db2.pm:65)
  BuildForge::Utilities::SysParams(/PerlApp/BuildForge/Utilities/SysParams.pm:57)
  main(.buildforge.pl:305)
  main(.buildforge.pl:213)
]
Database: 20075: CRRBF0556I: Trying to call [execute] on a non-existent database handle
DBD::DB2::st fetchrow_hashref failed: no statement executing at /PerlApp/BuildForge/DB/db2.pm
  line 78, <$sock> line 1054.
20075: CRRBF0555E: Problem performing Database Operation [fetchrow_hashref] : DBD::DB2::st
  fetchrow_hashref failed: no statement executing at /PerlApp/BuildForge/DB/db2.pm
  line 78, <$sock> line 1054.

Database: 20075: CRRBFEEEE: DBD::DB2::st fetchrow_hashref failed: no statement executing
  at /PerlApp/BuildForge/DB/db2.pm line 78, <$sock> line 1054.

Database: 20075: CRRBF0551I: StackTrace from Process id [20075] called from [
  BuildForge::Utilities::SysParams(/PerlApp/BuildForge/Utilities/SysParams.pm:59)
  main(.buildforge.pl:305)
  main(.buildforge.pl:213)
]
[IBM][CLI Driver] CLI0106E Connection is closed.
SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm
  line 385, <$sock> line 1054.
Database: 20075: CRRBF0555E: Problem performing Database Operation [prepare] :
  query [SELECT * FROM bf_jobcount WHERE bf_engine_id=?]
DBD::DB2::db prepare failed: [IBM][CLI Driver] CLI0106E Connection is closed.
  SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm line 385, <$sock> line 1054.

Database: 20075: CRRBFEEEE: DBD::DB2::db prepare failed: [IBM][CLI Driver] CLI0106E
  連線已關閉。SQLSTATE=08003 at /PerlApp/BuildForge/DB/Handle.pm
  line 385, <$sock> line 1054.

```

其中 XXX.XXX.XXX.XXX 是管理您資料庫的伺服器。

這則訊息指出 Rational Build Forge 與資料庫之間沒有連線。

若要重新建立連線，請執行下列動作：

1. 關閉 Rational Build Forge，如第 234 頁的『啟動和停止 Perl 引擎』所討論的方式。
2. 確定資料庫在執行中。
3. 啟動 Rational Build Forge，如第 234 頁的『啟動和停止 Perl 引擎』所討論的方式。

第 12 章 從舊版升級

下列主題說明如何將元件從前一個版本升級至現行版本：

- 升級必要條件。

重要：如何未能符合這些必要條件，則可能損壞與停用您的系統。

- 升級管理主控台
 - 升級 7.1.x 版主控台
 - 升級 7.0.2.x 版主控台
- 升級代理程式

升級必要條件

在執行升級安裝前，您應該先完成下列事項。

- **備份資料庫。**另請參閱第 34 頁的『資料庫備份（必要）』
- 如果您想要升級資料庫和資料庫用戶端的版本，請先完成這些升級作業。您無法變更安裝後的系統以使用新版本。系統會使用您在安裝程式中所指定的資料庫和資料庫用戶端檔案。
- 備份 Apache 和 tomcat 的任何自訂配置檔。

```
<bfinsastall>/Apache/conf  
<bfinsastall>/Apache/tomcat/conf
```

通常會自訂 Apache 的 httpd.conf 和 php.ini 檔案。

- 如果將系統配置為使用 HTTPS 和 SSL，則請備份您的憑證目錄。升級作業不會影響上述作業。

升級 7.1 版主控台

如果您要從任何 7.1.x 版升級至最新版本，請使用本節的資訊。

在開始升級前，請先檢查『升級必要條件』。

如果要升級，請執行更新安裝。

註：在升級 7.1 版主控台時，還需要執行其他作業，才能啓用效能加強功能。如需詳細資料，請參閱 jazz.net 上的版本頁面。

執行更新安裝

這項作業說明如何執行更新安裝。

開始之前

- 如果 Build Forge 在執行中，請立即停止它。
- 如果您是在 Windows 平台上執行，請檢查安裝目錄中的 bfengine.pid 檔。如果它存在，請將它移除。如果它存在，Installation Manager 無法執行更新。

- 在更新之前，必須先將儲存庫 URL 設定成已更新的儲存庫。如需相關指示，請參閱第 72 頁的『指定儲存庫 URL』。

關於這項作業

Build Forge® 8.0 僅支援從 7.1.1.3 版以及更新版本來升級。若是 7.1.1.3 以前的版本，您必須先升級到 7.1.1.3，再升級到 8.0

程序

1. 啟動 IBM Installation Manager。
2. 在 Installation Manager 中按一下**更新**。
3. 在「更新套件」畫面上，選取套件群組來尋找更新項目，然後按**下一步**。

註：若只要看見建議的更新套件，請選取**只顯示建議**勾選框。

4. 選取要安裝的更新項目或修正程式：

- a. 從清單中選取安裝。
- b. 按**下一步**。

這時會顯示您可以更新的版本。

5. 從版本清單中，選取安裝並按**下一步**。
6. 選取**我接受授權條款** 勾選框，然後按**下一步**。這時會顯示「更新套件」頁面，並選取您的特性。
7. 按**下一步**。
8. 若要測試配置，請配置 JAS 的資料庫連線，然後按一下靠近螢幕底端的**測試連線**。通過測試之後，請按**下一步**。
9. 配置資料庫用戶端程式庫，然後按一下**測試連線**。通過測試之後，請按**下一步**。這時會顯示您在 *buildforge.conf* 檔中編輯的資訊。
10. 在「更新套件」及「主控台啟動選項」畫面上，於下列提示中輸入資訊：

- 在「JAS 資料庫配置」畫面上：
 - a. 在**資料庫類型**欄位中，輸入您要更新的資料庫類型。
 - b. 在**您要在安裝期間在這個資料庫中移入資料嗎？**提示上，選取**是或否**勾選框。
 - c. 在**資料庫主機**欄位中，輸入主機的 IP 位址。
 - d. 在**資料庫埠**欄位中，輸入要用於這個資料庫連線的埠。預設值為 **3306**。
 - e. 在**資料庫使用者名稱**欄位中，輸入登入資料庫時要使用的使用者名稱。
 - f. 在**密碼**欄位中，輸入登入資料庫所需的密碼。
 - g. 在**確認密碼**欄位中，重新輸入密碼。
 - h. 在 **JDBC 驅動程式位置**欄位中，按一下**瀏覽**，以搜尋並選取 JDBC 驅動程式的位置。這時會顯示必要的驅動程式 JAR 檔及必要的 JDBC 驅動程式類別。
 - i. 按**下一步**。這時會顯示第二個「更新套件」及「主控台啟動選項」畫面。
- 「主控台啟動選項」畫面
 - a. 在**資料庫名稱**欄位中，輸入資料庫名稱。**bfdotcmd** 是預設值。
 - b. 在**密碼**欄位中，輸入登入資料庫所需的密碼。

- c. 在**確認密碼**欄位中，重新輸入密碼。
 - d. 在 **JDBC 驅動程式位置**欄位中，按一下**瀏覽**，以搜尋並選取 JDBC 驅動程式的位置。這時會顯示必要的驅動程式 JAR 檔及必要的 JDBC 驅動程式類別。
 - e. 若要測試資料庫配置連線，請按一下**測試連線**。
 - f. 通過測試之後，請按**下一步**。
 - 在「資料庫用戶端程式庫配置資訊」畫面上，完成這些欄位：
 - a. 在「我的 SQL 用戶端程式庫的路徑」欄位中，按一下**瀏覽**，以搜尋並選取 SQL 用戶端程式庫的路徑。
 - b. 若要再次測試資料庫連線，請按一下**測試連線**。這時會顯示測試連線的結果。
 - c. 按**下一步**。
11. 在下列提示上輸入資訊來繼續更新：
- a. UNIX 和 Linux - 在執行 **Build Forge** 的使用者身分提示上，輸入使用者名稱或接受預設使用者 root。這個使用者會啟動 Build Forge 引擎及提供的 Apache Tomcat。對於 Installation Manager 的後續頁面上指定的資料庫程式庫及 JDBC Jar 檔，這個使用者必須具有讀取及執行許可權。
 - b. 在**是否要讓安裝程式對資料庫進行必要的修改**？提示上，選取這個勾選框，讓安裝程式變更您的綱目。如果沒有選取這個勾選框，在安裝之後，您必須手動執行 `bfschema -u`，以套用這些綱目變更。
 - c. 在**您要在升級之後啟動主控台嗎**？提示上，選取這個勾選框，讓主控台在安裝完成之後自動啟動。
 - d. 按**下一步**。這時會列出您選取要安裝的特性。
12. 按一下**更新**來開始安裝。
13. 當更新順利完成時，按一下**完成**。
14. 關閉 Installation Manager。
15. 啟動 Build Forge。如果您是從 7.1.1.x 版升級，則要等到「字元大型物件 (CLOB)」移轉完成，才會看到來自您建置的步驟日誌。當您啟動 Build Forge 時，即會開始移轉。移轉完成時，Build Forge 會通知您。

升級到 7.1.3.1 版之後複製配接器

從 7.1.1.x 升級到 7.1.3.1 以及更新版本之後複製配接器時，發生一個問題。部分配接器已升級為 7.1.3.1 版。如果您升級到 7.1.3.1 版，並嘗試複製在升級之前所建立的配接器，則會報告錯誤：

CRRBF0789I：必須指定有效的「配接器範本 ID」。

暫行解決方法：嘗試複製配接器之前，請執行下列動作：

1. 開啓配接器。
2. 重新命名配接器。
3. 儲存配接器。

此外，您也可以開啓配接器、複製其內容，然後建立新的配接器。

升級 7.0.2.x 版主控台

如果您要將 7.0.2.x 版的系統升級至最新版本，請使用本小節。

如果要升級，請執行下列動作：

1. 檢查第 181 頁的『升級必要條件』。
2. 從 7.0.2.x 版升級至 7.1.1.3 版。

從 IBM 支援中心網站下載 7.1.1.3 版套件，然後遵循《*IBM Rational Build Forge 7.1.1.3 版安裝手冊*》中的指示。由於資料庫綱目變更，從 7.0.2.x 版升級至 7.1.1.3 版需要您移轉配置資料（專案）。

註：您也可以選擇移轉歷程資料（工作的日誌）。

重要：並非所有項目都會進行移轉。尤其是**專案 > 範本**中的自訂通知範本項目。

3. 在 7.1.1.3 版上執行更新安裝，將它更新至最新版本。

註：在升級 7.1 版主控台時，還需要執行其他作業，才能啓用效能加強功能。如需詳細資料，請參閱 jazz.net 上的版本頁面。

升級代理程式

您可以在較舊的代理程式上安裝較新的代理程式，來升級代理程式。

您可以在較舊的代理程式上安裝較新的代理程式，來升級代理程式。請遵循安裝指示來安裝代理程式。請參閱第 131 頁的第 10 章，『安裝代理程式』。

第 13 章 解除安裝產品元件

使用 IBM Installation Manager 來解除安裝已使用 Installation Manager 來安裝的產品元件。如果要解除安裝「Build Forge 代理程式」軟體，請使用本節說明的作業系統工具和指令。

使用 Installation Manager 來解除安裝產品

您可以使用此實務範例，來解除安裝利用 Installation Manager 所安裝的 Build Forge 產品元件。

註：此程序不會解除安裝 Build Forge 已建立的資料庫或資料庫物件。您必須使用工具，為資料庫分別移除這些項目。若為安裝在 Windows 上的資料庫，您可以使用**控制台 > 新增或移除程式**。如需相關指示，請參閱資料庫說明文件。請在移除 Build Forge 產品元件之後，才移除資料庫或資料庫物件。

如果要解除安裝產品元件，請完成下列步驟：

1. 使用用來安裝產品套件的相同使用者帳戶登入作業系統。

關閉任何以 Installation Manager 安裝之執行中的程式。

2. 如果 Build Forge 引擎和服務在執行中，請依下列方式將其停止：

Windows	<ul style="list-style-type: none">• 如果引擎是在前景中執行：按 Ctrl + c 鍵來停止引擎和 Build Forge 服務。 <p>重要：停止引擎和服務最可靠的方法是使用 Ctrl + c 鍵。當您解除安裝時，如果引擎或任何 Build Forge 服務仍在執行中，解除安裝會失敗。</p> <ul style="list-style-type: none">• 如果引擎是在背景中執行：開啓指令視窗，並輸入下列指令來停止引擎和服務： <pre>net stop bfengine<major number><Fix Pack Number>-<Build Number></pre> <p>例如，net stop bfengine 8.0.0.0-0-0308。</p> <p>這些指令會停止 Build Forge 引擎、Apache HTTP Server 和 Apache Tomcat 應用程式伺服器。</p>
UNIX/Linux	<ol style="list-style-type: none">1. 在指令視窗中，移至 Build Forge rc 目錄。 <pre>cd /opt/buildforge/rc</pre> <ol style="list-style-type: none">2. 停止引擎。 <pre>./buildforge stop</pre> <p>這個指令會停止 Build Forge 引擎、Apache HTTP Server 和 Apache Tomcat 應用程式伺服器。</p>

3. 啓動 IBM Installation Manager。
4. 在「啓動」頁面中，按一下**解除安裝**。

5. 在「解除安裝套件」頁面中，從「安裝套件」清單選取要解除安裝的產品套件，然後按**下一步**。
6. 在「摘要」頁面中，查看要解除安裝的套件清單，然後按一下**解除安裝**。移除套件之後，會顯示「完成」頁面。
7. 按一下**完成**。
8. 關閉 Installation Manager 視窗並結束 Installation Manager。

您必須先結束 Installation Manager 才能清除 Build Forge 檔案、選擇性地移除 DB2 Express 資料庫，或是重新安裝產品。

在 Installation Manager 安裝失敗時手動解除安裝產品

如果您未先停止 Build Forge 引擎和服務就啟動 IBM Installation Manager，解除安裝會失敗。

如果要在 Installation Manager 失敗之後手動解除安裝產品元件，請執行下列動作：

1. 在指令視窗中，切換至 Build Forge 安裝目錄中的 Manager 目錄，例如：

Windows	C:\Program Files\IBM\Build Forge\Manager
UNIX/Linux	/opt/buildforge/Manager

2. 在命令提示字元中，輸入 `main.exe uninstall main.res`。
3. 當指令執行完成時，刪除 Build Forge 安裝目錄。
4. 利用 Installation Manager 重新安裝產品。

解除安裝後清除 Build Forge 檔案

在解除安裝之後，需要執行某些手動清除。

在透過 Installation Manager 解除安裝產品元件之後，請手動刪除下列目錄。如果這些目錄存在，則不能利用 Installation Manager 來重新安裝 Build Forge。

目錄	預設位置
Build Forge 安裝目錄	Windows : C:\Program Files\IBM\Build Forge UNIX/Linux : /opt/buildforge
共用檔案目錄	Windows : C:\Program Files\IBM\SDPShared UNIX/Linux : /opt/IBM/SDPShared

使用 Installation Manager 來重新安裝產品並使用現有的 DB2 Express

如果您在解除安裝其他產品元件時，並未移除現有的 DB2 Express 資料庫，您可以重新安裝產品並使用該資料庫。這個實務範例可讓您存取現有的 Build Forge 專案和工作日誌。

在啟動 Installation Manager 之前，請使用下列步驟中的核對清單來取得所需的 DB2 Express 資訊。

重要：如果您已配置系統使用 HTTPS 和 SSL，即需要備份位於 `<bfinstall>/keystore` 和 `<bfinstall>/bfclient.conf`（憑證密碼的儲存位置）的憑證。在 UNIX 和 Linux 系統上，此目錄會位於 `<bfinstall>/Platform`。

如果要重新安裝並使用現有的 DB2 Express 資料庫，請執行下列動作：

1. 啟動 Installation Manager。
2. 在「啟動」頁面中，按一下**安裝**。
3. 請遵循 Installation Manager 精靈中的指示來重新安裝產品元件。
4. 在「資料庫配置」頁面中選取下列項目：
 - a. 對於安裝 *DB2 Express*，選取**否**。
 - b. 針對您要在安裝期間在這個資料庫中移入資料嗎？，選取**否**。
 - c. 如果要完成其餘欄位，請使用下表中的資訊：

核對清單：DB2 Express 配置資訊

✓	欄位	說明
	資料庫主機	所提供的 DB2 Express 資料庫會安裝在本端主機名稱 (127.0.0.1) 上。
	資料庫名稱	資料庫名稱為 BUILD（全部大寫）。
	資料庫綱目名稱	資料庫綱目的名稱為 BUILD（全部大寫）。
	資料庫使用者名稱	您提供來建立 DB2 Express 資料庫的資料庫使用者名稱。
	資料庫密碼	您提供給資料庫使用者名稱的密碼。
	DB2 用戶端程式庫的路徑	用於 DB2 Express 的 DB2 用戶端程式庫 (db2cli.dll) 路徑為 C:\Program Files\IBM\SQLLIB\bin。
	JDBC 驅動程式位置	用於 DB2 Express 的 JDBC 驅動程式 (db2jcc.jar) 路徑為 C:\Program Files\IBM\SQLLIB\java。

5. 按一下**測試連線**。
6. 按**下一步**繼續進行並完成安裝。

解除安裝 Windows Build Forge 代理程式

請使用 Windows 的「新增或移除程式」工具，來移除 Build Forge 代理程式軟體。

註：如果您要重新安裝，您可以改寫相同版本的代理程式軟體。

1. 在 Windows 中，找到「新增或移除程式」。比方說，選取**開始** → **程式集** → **控制台** → **新增或移除程式**。
2. 在目前安裝的程式清單中，找到 IBM Rational Build Forge 代理程式。
3. 按一下**變更/移除**。
4. 遵循指示來解除安裝代理程式。

解除安裝 UNIX 或 Linux Build Forge 代理程式

使用下列指令，從 UNIX 或 Linux 平台解除安裝代理程式軟體。

Linux 代理程式

如果要移除利用 rpm 套件安裝的代理程式軟體，請執行下列動作：

1. 找出代理程式軟體並列出套件名稱和版本：

```
rpm -qa | grep bfaagent
```

2. 刪除代理程式軟體：

```
rpm -e bfaagent-<version_number>
```

Solaris 代理程式

如果要移除利用 pkgadd 程式安裝的代理程式軟體，請執行下列指令：

```
pkgrm BFAgent
```

其他代理程式

針對其他平台，解除安裝程序是手動的，且視平台而有所不同。請遵循適用於您平台和超級伺服器實作的指示。

註：為了執行大部分指令，您需要有 root 存取權，且 /sbin and /usr/sbin 目錄必須設定在現行 PATH 環境變數中。

1. 移除代理程式服務常駐程式 bfaagent。請使用適用於您平台之超級伺服器實作（inetd、xinetd、launchd 或 SMF）的指示。

超級伺服器	程序
inetd，通用於較舊的 UNIX 系統	<ol style="list-style-type: none">1. 編輯 /etc/inetd.conf 檔，並移除 bfaagent 這一行。2. 找出 inetd 的處理程序 ID。 <pre>ps -ef grep [i]netd</pre> 針對 BSD 衍生的系統（如：FreeBSD 和 Mac OS/X 10.4 和先前的版本），請將 ps -ef 換成 ps auwwwx。3. 讀取更新後的 inetd.conf 並啟動 inetd。 <pre>kill -HUP <PID></pre>
xinetd，通用於較新的 UNIX 系統	<ol style="list-style-type: none">1. 如果要移除代理程式服務，請執行下列指令： <pre>rm /etc/xinetd.d/bfaagent</pre>2. 找出 inetd 的處理程序 ID。 <pre>ps -ef grep [i]netd</pre> 針對 BSD 衍生的系統（如：FreeBSD 和 Mac OS/X 10.4 和先前的版本），請將 ps -ef 換成 ps auwwwx。3. 讀取更新後的 inetd.conf 並啟動 inetd。 <pre>kill -HUP <PID></pre>

超級伺服器	程序
Mac OS/X 和 OpenBSD 系統的 launchd	<ol style="list-style-type: none"> 1. 執行 launchctl。 2. 輸入 stop com.ibm.rational.bfagent。 3. 輸入下列指令：unload /Library/LaunchDaemons/com.ibm.rational.bfagent.plist 4. 輸入 quit。 5. 執行下列指令：rm Library/LaunchDaemons/com.ibm.rational.bfagent.plist
Solaris 10 的「Solaris 系統管理機能 (SMF)」	<ol style="list-style-type: none"> 1. 執行 inetadm -d network /bfagent/tcp 2. 執行 svccfg delete -f network/bfagent/tcp

2. 從 PAM 介面移除代理程式服務。
 - a. 編輯 /etc/pam.conf，並移除所有以 bfagent 開頭的字行。
 - b. 執行 rm /etc/pam.d/bfagent
3. 從 etc/services 檔移除通訊協定項目。

編輯 /etc/services，並移除 bfagent 這一行。

4. 移除代理程式安裝的下列檔案：

```

/etc/bfagent.conf
/etc/bfagent.conf-example
/usr/local/bin/bfagent/usr/local/bin/bfcrypt.dll

```

第 14 章 版本注意事項

本節包含每一版產品的版本注意事項。其中也可能包含項目來記錄「資訊中心」在各個版本之間有所更新的位置。版本注意事項包含下列項目：

- 版本中新增和變更的特性清單。
- 需求、安裝指示、支援及其他產品資訊的參照。

版本注意事項 - IBM Rational Build Forge 8.0 版

Rational Build Forge 8.0 版可用。其已解決相容性、安裝和其他入門問題。

© Copyright International Business Machines Corporation 2003, 2013. All rights reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

內容

- 『說明』
- 『新增和變更的特性』
- 『這個版本修正的 APAR』
- 第 192 頁的『公告』
- 第 192 頁的『舊版的相容性問題』
- 第 192 頁的『系統需求』
- 第 192 頁的『安裝』
- 第 192 頁的『已知問題』

說明

IBM® Rational® Build Forge® 是一種可調適的程序執行架構，其會自動化、編排、管理及追蹤軟體開發組合線內每一個轉換之間的所有程序，用以建立自動化的軟體工廠。IBM Rational Build Forge 會整合到您的現行環境中，並支援主要開發語言、Script、工具和平台。它可讓您在針對程序自動化、加速、通知和排程增加有價值的功能時，繼續使用現有的投資。

新增和變更的特性

下列清單指出此版本中新增和變更的特性。如需最新的特性清單，請參閱 [Jazz .net](#) 上 GA 版本的 [New & Noteworthy](#) 頁面（需要帳戶）。

這個版本修正的 APAR

如需這個版本修正的 APAR 最新清單，請參閱 [Jazz .net](#) 上 GA 版本的「版本注意事項」頁面（需要帳戶）。

公告

這個版本的發行公告位於 www.ibm.com/common/ssi/index.wss。請參閱公告以取得下列資訊：

- 詳細產品說明，包含新特性的說明
- 軟硬體需求
- 軟體服務及支援可用性
- 包裝與排序詳細資料
- 國際相容性資訊

舊版的相容性問題

8.0 版中變更的資料庫表格架構。需要從舊版資料庫表格架構移轉 Build Forge 的配置資料和歷程資料。

升級至 8.0 版的需求視現行版本而定。

- 7.1.0.x、7.1.1.x 和 7.1.2.x：如果您執行的是其中任一版本，您可以使用 Installation Manager 的「更新」特性來進行升級。
- 7.0.2.x：如果您執行的是其中任一版本，那麼必須先升級到 7.1.1.3 版，然後再升級到 8.0 版。

系統需求

軟硬體需求可在特定產品的詳細系統需求網站上，從查詢的報告中取得。您必須提供您所使用的產品、產品版次、版本以及作業系統。其中會針對硬體、作業系統、資料庫、代理主機和 Web 用戶端來說明需求。

安裝

如需逐步安裝指示，請參閱緊接在第 21 頁的第 4 章，『需求』後面的主題。

已知問題

已知問題會以個別 TechNotes 的形式，記載於「支援中心」知識庫中 (<http://www.ibm.com/software/rational/support/>)。在發現及解決問題時，「IBM 支援中心」團隊會更新知識庫。透過搜尋知識庫，您可以快速尋找暫行或徹底的解決方法。

- 已報告的問題，Rational Build Forge 8.0 版
<http://www.ibm.com/support/search.wss?rs=3099&tc=SS2MGB&q=RN8.0&rankprofile=8>
- 已解決的問題，Rational Build Forge 8.0 版
<http://www.ibm.com/support/search.wss?rs=3099&tc=SS2MGB&q=RN8.0C&rankprofile=8>

在 IBM Rational Build Forge 支援網站，您也可以執行您自己的查詢來要求其他類型的資訊：

- 下載
- Flash（警示）
- 新聞
- 產品資訊與出版品
- 疑難排解（範圍比 URL 提供的還廣泛）

相關資訊:

 [聯絡 IBM 支援中心](#)

第 15 章 「儀表板」標籤

「儀表板」標籤可為您提供系統狀態及活動的概觀。

工作成功

依週顯示在過去四週內通過或失敗的工作數目。您可以選擇查看所有專案或單一專案的趨勢。

排程 排定在將來執行的工作（依其排定的執行時間列出）。您可以選取如何呈現清單（所有或限定數目）及翻看清單。

伺服器狀態

伺服器及其狀態：「可用」或「關閉」。您可以選擇檢視所有伺服器的狀態，或僅檢視與所選收集器相關聯的伺服器狀態。您可以選取如何呈現清單（所有或限定數目）及翻看清單。

工作佇列

執行中及等待中的工作數目。此圖表會隨著時間自動更新。「同時執行的工作數上限」系統設定在圖表上顯示為臨界值。

清除佇列

執行中及等待中的清除數目。此圖表會隨著時間自動更新。「同時執行的清除數上限」系統設定在圖表上顯示為臨界值。

授權位置

使用中的授權位置數。它會顯示在指示器刻度盤及標題中：「已使用 (X) 個，共 (總計) 個授權位置」。

註：在 Internet Explorer 7 及 8 中，瀏覽器中的限制會阻止號碼顯示在此儀表板上。

第 16 章 管理

這個主題說明 Build Forge® 系統的管理作業。

關於管理

使用「管理」畫面來管理配置和喜好設定。

在「管理」畫面內，您可以處理使用者、安全專用權、通知設定和系統設定來配置「管理主控台」。

若要存取「管理」畫面，請在左側功能表中按一下**管理**。

The screenshot shows the '系統' (System) configuration page. At the top, there are tabs for '使用者介面配置' (User Interface Configuration), '主控台' (Dashboard), '報告' (Reports), and a '登出: Root User' (Logout) link. Below the tabs, there's a navigation bar with '系統' (System) selected. A table lists system settings: '終止執行緒' (Terminate Thread) set to '是' (Yes), '繼承標籤' (Inherit Tag) set to '否' (No), '自動登出分鐘數' (Auto Logout Minutes) set to '0', '警示電子郵件限制' (Alert Email Limit) set to '10/30', and '資料庫大小臨界值' (Database Size Threshold) set to '2G'. Below the table, there's a section for '警示電子郵件限制' (Alert Email Limit) with a '儲存' (Save) button and a '回復到預設值' (Reset to Default) button. Under the '詳細資料' (Details) tab, the '警示電子郵件限制' (Alert Email Limit) is set to '10/30'. A note below states: '輸入下列項目值：容許的警示電子郵件訊息數目上限（以分鐘為時間單位）（訊息/分鐘）。0/0 表示無限制。' (Enter the following item value: Maximum number of alert email messages allowed (in minutes as time unit) (messages/minute). 0/0 indicates unlimited.) The default value is shown as '預設值: 10/30'.

存取群組

存取群組是系統用來控制許可權的使用者集合。

使用**管理 > 存取群組**畫面可以建立新的存取群組、新增或移除使用者，以及修改群組內容。此畫面會顯示現有存取群組的清單。按一下群組的名稱來選取它，並將其內容顯示在畫面的下方。

[使用者介面配置](#) | [主控台](#) | [報告](#) | [登出](#): Root User

存取群組 | **新增群組** 說明 ?

過濾條件
顯示 1 - 4 (總共 6)
[顯示全部](#)

 << < 第 1 頁 (共 2 頁) > >>

存取群組名稱	新增使用者預設值	LDAP 群組 DN
建置工程師	否	
開發人員	否	
訪客	是	*
操作員	否	

新建群組
儲存
複製
刪除

詳細資料 | 使用者 | 子群組 | 許可權

☐ 預設值

名稱:

擁有者: 建置工程師

LDAP 群組 DN:

- 若要建立新群組，請按一下**新增群組**，即可清除畫面下方的欄位（必要的話）。然後，為群組提供一個**名稱**，並選取**擁有者**群組。（**擁有者**群組可控制存取新群組。如果要編輯群組，使用者必須是**擁有者**群組的成員）。如果您是使用 LDAP 鑑別，請填寫「LDAP 群組 DN」欄位，告訴系統哪個 LDAP 群組要對映您的群組。例如，您可以將「開發人員」群組對映至名為 SoftwareEngineers 的 LDAP 群組，然後指派許可權給「開發人員」群組，以提供您希望軟體工程師擁有的存取權類型。
 - 在「LDAP 群組 DN」欄位中，列出要讓成員接收與此存取群組相關聯之「管理主控台」安全專用權的所有 LDAP 群組的「識別名稱」。
 - 您可以將多個 LDAP 群組對映至任何存取群組。您可以在這個內容中使用星號 (*) 字元，將成員資格提供給這個存取群組中的所有 LDAP 使用者。您可以列出多個 LDAP 群組，並以分號區隔它們。
 - 如果要刪除存取群組，請選取群組，然後按一下**刪除**。
- 註：**您無法刪除在他處指派成**存取**內容的存取群組。例如，如果您建立一個存取群組，並將專案的**存取**內容設為使用它，則無法刪除此群組。您必須先將專案編輯成使用另一個存取群組。
- 如果要新增或移除使用者，請選取群組，然後按一下**使用者**標籤。系統會在左邊顯示非成員清單，在右邊顯示成員清單。選取使用者，並使用**新增**及**移除**，將其從一個清單移至另一個清單。
- 註：**將「對映存取群組對映」設為「是」的 LDAP 使用者，將不會顯示在**使用者**標籤中。此行為可確保「對映存取群組對映」設定會真正執行群組對映，而不會與 Build Forge 手動設定相衝突。
- 若要將群組巢狀化，請將一個群組新增為另一個群組的子群組。當您這麼做時，所有套用至母群組的許可權，也都會套用至成員群組中的所有使用者。若要使一個群組成為另一個群組的子群組，請選取所需的**上層群組**，然後在畫面下方按一下**子群**

組標籤。選取要作為子群組的群組，然後按一下**新增**。您可以用遞迴方式將群組巢狀化，例如，將母項新增至子項，讓群組 A 包含群組 B，而群組 B 又包含群組 A。如果您這麼做的話，系統會將群組 A 的所有成員視為群組 B 的成員，反之亦然。

- 如果要管理群組的許可權，請選擇群組，然後按一下**許可權**標籤。您可以檢視群組的現行許可權，以及新增或移除許可權。

「存取」概觀

系統會管理其資料庫中的使用者。您可以透過為使用者指派的群組來控制使用者的專用權。您可以指派專用權給群組，然後使每一個使用者成為適當群組的成員。

這是一種角色型系統 — 每一個群組代表使用者可在您組織中扮演的角色。角色具備專用權。使用者的專用權是該使用者所屬群組的總和。您不能直接指派專用權給個別使用者，只能指派給群組。

系統也使用存取群組來進行通知。當您將系統配置為傳送通知訊息時，訊息的目標必須是存取群組。請參閱第 307 頁的『設定通知』。

安全專用權（或許可權）定義群組可以執行和查看的內容。它們可以用來當作過濾器，用來過濾群組在系統中的體驗。例如，身為「訪客」群組成員的使用者（不屬於其他任何群組），只能查看已指派「訪客」群組作為其「存取」內容的專案。該使用者只以「訪客」存取權來啟動專案。如果該使用者也是「開發人員」群組的成員，則他可以查看其「存取」內容為「訪客」或「開發人員」的所有專案。

註：您可以使用現有的 LDAP 資料庫，而不使用進行使用者鑑別時所用的資料庫。當您使用 LDAP，而不在系統中的使用者時，將會容許 LDAP 資料庫的部分或全部使用者存取系統。您也可以將存取群組對映到 LDAP 群組。如需設定 LDAP 的詳細資料，請參閱第 206 頁的『關於 LDAP 整合』。

您可以使用存取群組控制的活動和資源包含：許可權、伺服器、專案、步驟和存取群組。

- 若要將資源（伺服器、專案或步驟）的存取權延伸至特定群組，請選取該資源，然後將其「存取權」欄位變更為該群組的名稱。例如，若要讓 Developer 群組存取名稱為 Win234 的特定伺服器，請將該伺服器的「存取權」內容設為 Developer。

註：使用者若不是伺服器或專案的「存取群組」成員，就不會在「伺服器」或「專案」清單頁面中看到該物件。使用者若不是步驟的「存取群組」成員，儘管可以在專案的清單中看到該步驟，但無法編輯或執行它；如果使用者執行的專案包含該步驟，系統會跳過使用者沒有其存取權的步驟。

- 如果要容許一個存取群組的成員編輯另一個存取群組，請將一個群組設為另一個群組的「控制群組」。例如，如果要容許「群組 A」成員新增成員至「群組 B」，必須使「群組 A」成為「群組 B」的「控制群組」。
- 如果要將廣域專用權向某群組延伸，請使用**管理 > 許可權**頁面來啟用該群組的特定許可權。

這個彈性模型可讓您安全地提供一個專用權（例如執行建置的能力）給某些類型的使用者，同時限制其他使用者的專用權（例如編輯專案或使用特定伺服器的權限）。

「存取」範例：讓群組有能力執行工作

您可以使用安全特性，將執行某些工作的能力延伸至您的其中一個存取群組。例如，您可能有一個裝置驅動程式的程式設計師群組，您希望他們能執行與其工作相關的工作，而不要因為其他工作而使其系統視圖雜亂無章。但是，您不想讓程式設計師編輯這些工作。如果要建立此實務範例：

1. 在組織中建立這個角色的存取群組 (例如：DeviceDriverDevs)。
2. 指派新的存取群組，作為您希望使用者能夠執行的所有專案的**存取**內容。
3. 請確定專案的步驟也含有適當的**存取**內容。在執行工作時，會跳過使用者沒有存取權的任何步驟。
4. 將「執行建置」許可權指派給該群組。
5. 將所有需要啟動這些建置的使用者新增至新的 DeviceDriverDevs 群組。您可能也需要讓系統的管理者成為群組的成員。在您變更專案的「存取」內容時，非 DeviceDriverDevs 群組成員的使用者會喪失檢視、執行或編輯該專案的能力。

請注意，使用者可以是許多群組的成員，且許可權是累加的。您可以有另一個專案小組的群組 (例如：PlatformDevs)，而同時身為這兩個群組成員的使用者，可以檢視及啟動已將任一群組設為「存取」內容的專案。

團隊和專案安全計劃

如果您有許多處理不同專案的使用者，下列一般計劃提供您管理他們的能力，讓個別的使用者可以取得本身所需的許可權，但只能看到他們需要與其互動的專案和其他資源：

- 針對使用者執行的各種活動建立角色型存取群組。例如，建立 Manager 以及 Developer 群組。指派適合其工作的許可權給這些群組。建置的 Manager 可能擁有大部分可用的許可權，Developer 可能只擁有執行工作的相關許可權。
- 為組織中的每一個跨功能團隊建立其他群組。您可能需要 IDE 團隊、PrinterDriver 團隊和其他團隊。
- 對團隊群組設定專案、伺服器和其他資源的存取內容。與 PrinterDriver 團隊有關的所有專案應該有 PrinterDriver 存取群組作為其存取內容。
- 當您在系統中新增使用者時，請指派使用者給所有適當的存取群組。所有使用者都必須指派給至少一個角色群組和至少一個團隊群組。

如果您遵循這些準則，使用者只會看到與他們有關的專案，並在那些專案中擁有適合他們角色的許可權。而且，當組織內的工作變更時，您可以輕易變更使用者許可權。

管理存取內容

使用者可以根據使用者的存取群組來設定存取內容。使用者必須為存取群組的成員，才能將資料物件（例如專案、服務或步驟）指派給該存取群組。

例如，如果您不是管理者群組的成員，就不能指派專案給該群組。

存取群組清單只限於您所屬的存取群組。

步驟會繼承其上層專案的存取群組內容。步驟建立者可變更步驟的存取群組內容，使其具有不同於該專案的存取群組內容。如果使用者不是針對步驟所指定的存取群組成員，則無法執行該步驟。這可讓您防止某些使用者執行專案中特定的步驟。

使用者

這些主題說明如何管理系統中的使用者。

使用者內容

您可以利用**管理 > 使用者**，來建立使用者，並指派內容給他們。您也可以讓系統連接 LDAP/Active Directory 資料庫，以取得使用者資訊。您可以將使用者指派給群組，以管理使用者的安全許可權，因此您必須建立一些使用者，以測試安全特性。

當您顯示使用者記錄時，共有三個可用標籤：

- **詳細資料**：使用此標籤可編輯大部分的使用者內容，可用的內容如下所述。
- **現行群組**：顯示使用者為其成員的存取群組，使用者可能是直接存取、或透過其他群組成員的直接群組來存取。
- **變更群組**：顯示使用者為其直接成員的群組，並可讓您新增使用者至群組，或是從群組中移除使用者。

The screenshot shows a web-based user management interface. At the top, there are several buttons: (新增使用者), 儲存, 複製, 切換至使用者, 使密碼到期, 登出使用者, 清除位址, 解除鎖定, and 刪除. Below these is a tabbed interface with three tabs: 詳細資料 (selected), 現行群組, and 變更群組. The 'Detailed Information' tab contains a form with the following fields and options:

- 使用者名稱: [Text input]
- 電子郵件: [Text input]
- 名稱: [Text input]
- 密碼: [Text input]
- 限制: [Dropdown menu, set to 無限制]
- 時區: [Dropdown menu, set to US/Central]
- 驗證: [Text input]
- 優先登入: [Dropdown menu, set to 否]
- 日期格式: [Dropdown menu, set to 03/1/1 下午 2:30]
- 語言: [Dropdown menu, set to 繁體中文]
- 密碼到期: [Dropdown menu, set to 是]
- 使用螢幕閱讀器: [Dropdown menu, set to 否]
- 行事曆的一週開始日: [Dropdown menu, set to 星期日]
- 截斷: [Dropdown menu, set to 預設值]
- 使用者類型: [Dropdown menu, set to 一般]
- 搜尋: [Dropdown menu, set to 區分大小寫]
- 步驟日誌起始視圖: [Dropdown menu, set to 第一頁]

針對每位使用者，您可以在**詳細資料**標籤中設定這些內容：

名稱 使用者的顯示名稱與標籤。

電子郵件

系統可以用來為此使用者傳送電子郵件通知的電子郵件位址。

註：電子郵件只會傳送給明確選出要通知的使用者。系統支援下列電子郵件位址格式。**使用者名稱** 是電子郵件使用者名稱。**名稱** 是任意字串，通常是某形式的給定名稱。

使用者名稱@host.com
名稱 使用者名稱@host.com
名稱 "<使用者名稱@host.com>"
名稱 <使用者名稱@host.com>
名稱 "使用者名稱@host.com"

使用者名稱

用來登入「管理主控台」的名稱。

密碼 用來登入「管理主控台」的密碼。目前已登入的使用者並不會顯示此欄位。使用此欄位即可輸入新密碼，或變更現有密碼。請在**驗證**欄位中輸入同一密碼。

限制 設定使用者一天內所能啟動的最大工作數目。當達到此限制時，系統會顯示訊息指出使用者已超出其執行配額。

時區 使用者的時區。系統會使用 root 使用者的時區作為所有顯示時間的預設時區。公佈的時間會使用伺服器主機時鐘。

依預設，系統內使用者與 LDAP 使用者都會被指派與 root 使用者相同的時區。使用者可編輯指派給他們的時區。

驗證 在此重複密碼以便驗證。

優先登入

指定此選項，可讓使用者成為優先使用者。優先使用者可隨時登入系統。如果已沒有可用的使用者授權，系統會將擁有最舊的階段作業的使用者登出，讓給優先使用者登入。root 使用者可隨時登入，且無需使用授權。

日期格式

選取使用者偏好的顯示格式。

語言 選取使用者的語言。

密碼到期

如果勾選這個選項，則在經過**密碼到期天數**系統設定中所指定的天數之後，使用者密碼就會到期。

使用螢幕閱讀器

如果設定為「是」，就會啟用適用於視障使用者的介面，來支援螢幕閱讀器功能，如動態強調顯示及集中焦點。

使用者類型

使用者可以屬於下列三種類型之一：

- 一般：可以使用 Web 用戶端存取主控台。根據其存取群組成員資格及許可權，他們可以變更系統。每位使用者只容許一個階段作業。
- 唯讀：可以使用 Web 用戶端存取主控台。除了變更自己的個人資訊（如密碼及喜好設定）以外，他們無法進行任何其他變更。每位使用者只容許一個階段作業。
- API：無法使用 Web 用戶端存取主控台。此類型的使用者會使用以 Java API 或 Perl API 所開發的用戶端來存取系統。相同 API 使用者可能會擁有多個登入階段作業。系統會自動共用階段作業資訊。

行事曆的一週開始日

選取「排程」行事曆要最先顯示星期幾，預設值為星期日。

截斷 控制在清單和下拉功能表中所顯示的字元數。例如，如果設為 20，則專案名稱只會顯示名稱的前 20 個字元。

搜尋 決定在整個產品中，「過濾器」欄位如何執行搜尋。其值如下所示：

- 區分大小寫（預設值） - 搜尋會考量每個字母的大小寫。
- 不區分大小寫 - 搜尋不考量大小寫。

步驟日誌起始視圖

指定在工作結果中開啓步驟日誌時，如何放置其位置供檢視。如果設為「第一頁」（預設值），則視圖會放在日誌的第一頁。如果設為「最後一頁」，則視圖會放在日誌的最後一頁。

管理使用者

這些主題說明針對使用者的基本管理作業。

選取**管理 > 使用者**來顯示現行使用者的清單，其下方為使用者畫面。系統會顯示每位使用者的姓名、使用者名稱、電子郵件、限制、活動（自前次使用者登入以來的經歷時間，以伺服器主機時鐘為準）及時區。

- 若要建立新的使用者，請在使用者畫面中，不選取任何使用者的情況下，開始輸入內容。如果已選取使用者，請按一下**新增使用者**來清除畫面。當您編輯好使用者資訊之後，請按一下**儲存**。
- 如果要編輯使用者，請在使用者畫面中，按一下使用者的名稱並編輯內容，然後按一下**儲存**。
- 如果要登出使用者，請按一下使用者的名稱，再按一下**登出使用者**。
- 如果要以使用者身分登入且不使用密碼，請先以 **root** 身分登入。按一下使用者的名稱，然後按一下**切換至使用者**。**root** 使用者可以使用任何使用者的身分登入，而不需要使用密碼。如果您目前是以非 **root** 使用者的身分登入，但想要以另一個非 **root** 使用者的身分登入，則必須先以 **root** 身分登入。
- 如果要釋放固定的授權位置，請先以 **root** 身分登入。按一下使用者的名稱，然後按一下**清除名額**。這時候主控台就會從計入這組固定授權的 ID 清單中，移除該使用者。如果該使用者已經登入，也會一併被登出。主控台在計算固定授權時，是以曾經登入的使用者數量來計算。當到達限制時，新的使用者就無法取得授權，必須刪除或清除現有的使用者，才能清出空間容納另一個使用者。清除名額，並不會從主控台刪除這個使用者。如果該使用者再次登入，固定授權計數便會增加。如果是用在浮動授權使用者，則**清除名額**與**登出使用者**的作用相同。
- 如果要複製使用者，請按一下該使用者的名稱，然後按一下**複製**。出現在清單中的新使用者名稱會附加 **"Copy"** 單字。

註：所複製的使用者的密碼會重設為 **password**。若要變更密碼，請以 **root** 或以複製的使用者身分登入。

- 如果要刪除使用者帳戶，請按一下該使用者的名稱，然後按一下**刪除**。

如果「刪除」停用，表示有一份排定的工作為該使用者帳戶所擁有，因而無法刪除該使用者帳戶。如果您想刪除已排定工作的使用者帳戶，必須先刪除該排定的工作。

使用者記錄會設定預設內容供使用者體驗系統，並且會控制使用者的登入名稱、密碼和密碼有效期限。使用者記錄的資料可透過「管理主控台」而輸入到系統，或是來源於 LDAP/Active Directory 資料庫。

註：當您編輯其記錄是來源於 LDAP 資料庫的使用者時，使用者頁面上的許多欄位都會被停用。您必須在來源資料庫中才能變更這些內容。

利用使用者類型控制存取權

「使用者類型」內容設定將決定使用者擁有的存取權類型。

- **一般：**可以使用 Web 用戶端存取主控台。根據其存取群組成員資格及許可權，他們可以變更系統。
- **唯讀：**可以使用 Web 用戶端存取主控台。除了變更自己的個人資訊（如密碼及喜好設定）以外，他們無法進行任何其他變更。

- **API**：無法使用 Web 用戶端存取主控台。此類型的使用者可透過使用 Java API 或 Perl API 所開發的用戶端來工作。相同 API 使用者可能會擁有多個登入階段作業。系統會自動共用階段作業資訊。

視寫入權群組 **DN** 內容的設定而定，透過 LDAP 登入的使用者可能會視為「一般」或「唯讀」使用者類型。請參閱 第 206 頁的『LDAP 整合』。

帳戶鎖定

您可以使用系統設定來施行鎖定原則。

鎖定原則會決定多次失敗的登入嘗試，將導致禁止使用者登入的條件。此條件設為連續失敗的登入嘗試次數上限。計時器會決定是否將某組登入視為連續的登入嘗試。系統可以根據計時器來「忘記」失敗的登入。

如果使用者帳戶在上次成功登入後，有失敗的登入嘗試，則在下次成功登入時，便會顯示蹦現畫面告知不成功的登入嘗試次數。只有在尚未符合鎖定條件時，才會顯示蹦現畫面。

當符合鎖定條件時，使用者即會進入鎖定狀態。此時可能會要求使用者完成下列兩件事之一：

- 在系統接受其他登入嘗試之前，等待預先決定的一段時間。
- 聯絡系統管理者以解除鎖定帳戶。

系統設定會決定鎖定原則：

- **帳戶鎖定嘗試次數上限**：如果為 0，則會停用帳戶鎖定。此值可以設為觸發鎖定所需連續失敗的嘗試次數。預設值為 3。
- **帳戶鎖定重設計時器**：如果設為負值，使用者必須請求管理者解除鎖定其帳戶，然後才能再次登入。此值可設為使用者可以再次登入之前所需的分鐘數。預設值：120 分鐘。
- **帳戶鎖定消除時間**：如果設為負值，則會永不忘記失敗的嘗試。此值可設為在忘記前次失敗的登入嘗試，並且不將其計入失敗的嘗試次數上限之前的分鐘數。預設值：60 分鐘。

請參閱**管理 > 系統**。

root 使用者

root 使用者（即登入名稱為 **root** 的使用者）在系統中擁有特殊性質：

- **於安裝時建立**：root 使用者是安裝程式所建立的唯一預設使用者，其預設密碼為 **root**（在安裝之後請立即變更密碼）。
- **不需要授權**：root 使用者不會耗用使用者授權。無論已有多少使用者登入，您一律能夠以 root 使用者身分登入。（如果有人正以 root 身分登入，則已經登入為 root 的使用者將會被登出。）
- **系統時區**：root 使用者的時區為「管理主控台」的預設時區。依預設，其他使用者（系統內及 LDAP 使用者皆是）將採用 root 使用者的時區。登入之後，使用者可以設定他們自己的時區。系統中所報告的所有時間和登入都會以使用者的時區來表示。

- **所有許可權**：root 使用者擁有所有可用的許可權，並且可以編輯其他使用者的內容。您無法移除 root 使用者的任何存取權。儘管 root 使用者並不是任何存取群組的成員之一，root 使用者仍可在系統中檢視、編輯、或使用任何資料物件。
- **優先順序**：root 使用者一律為優先使用者。
- **以任何使用者身分登入**：root 使用者能以其他使用者身分在不使用密碼的情況下登入，其做法是在**管理 > 使用者 > <使用者名稱>**頁面上按一下**切換至使用者**。
- **登出現行使用者**：root 使用者可以在**管理 > 使用者 > <使用者名稱>**頁面上，按一下**登出使用者**來登出使用者。
- 依預設，LDAP 使用者會被指派與 root 使用者相同的時區。然而，使用者可以在登入後自行編輯所指派的時區，系統會記住這個新的喜好設定。

API 使用者

若使用者透過以 Java API 或 Perl API 開發的用戶端來存取主控台，則應該將他們的「使用者類型」設為 API。此類型的使用者無法使用 Web 用戶端來存取主控台。但是，此類型的使用者可以擁有多個登入階段作業。階段作業間會自動彼此共用資訊。

相同 API 使用者所開啓的多個階段作業，只會使用一個授權。必須登出所有階段作業，才能釋出授權。您也可以使用 root 帳戶來釋出授權。請參閱第 203 頁的『管理使用者』。

如果您的使用者需要透過 Web 用戶端和 API 用戶端來存取主控台，請為他們設定不同的使用者 ID。存取 Web 主控台的使用者會是「一般」或「唯讀」的使用者類型。這些使用者類型無法擁有多個階段作業。如果使用者嘗試第二次登入，就會終止第一個階段作業。

許可權

許可權可定義哪些使用者可以在系統中執行。您是指派許可權給存取群組；而不是直接將許可權指派給使用者。

若要處理許可權，請選取**管理 > 許可**。

[使用者介面配置](#)
[主控台](#)
[報告](#)
[登出: Root User](#)

許可權

說明

過濾條件

顯示 1 - 4 (總共 105)

[顯示全部](#)

第 1 頁 (共 27 頁)

名稱	群組
以使用者存取層次匯入	執行
以完整存取層次匯入	執行
儲存公用報告	報告
切換來源鏈結	來源

新增建置附註

詳細資料

名稱: 新增建置附註

群組: 專案

選取下方清單中的名稱，按一下「新增」，將它們新增至右邊的成員清單中。

操作員

訪客

新增 >>

<< 移除

安全人員

建置工程師

系統管理員

開發人員

第 16 章 管理 205

如果要指派許可權給群組，請選擇許可權，並確定被列為擁有該許可權的所需群組。選取**管理 > 許可權**，從清單中選取許可權，然後再處理畫面下方的部分。左邊的群組沒有許可權；右邊的群組有許可權。如果要提供許可權給群組，請在左側的群組中選取它，然後按一下**新增**。

許可權練習

在這個範例中，您要將新增和編輯伺服器的專用權授與使用者 Jane Doe，透過將那些許可權提供給新的存取群組，再將她指派給該群組即可。

1. 以 root 使用者身分登入。
2. 建立稱為「伺服器管理」的新存取群組。
3. 將 Jane Doe 新增至這個新的存取群組。
4. 跳至**管理 > 許可權**。
5. 捲到**新增伺服器許可權**，然後按一下它。
6. 在**詳細資料**標籤中，使用**新增**和**移除**，讓「伺服器管理」成為唯一具備這項許可權的存取群組。
7. 按一下**更新**。現在只有 Jane Doe 具有將伺服器新增至系統的許可權。

註：Root 使用者一定具備所有的許可權。您無法移除 root 使用者的任何存取權。

請注意，雖然只有 Jane 可以新增伺服器，她卻無法輸入該伺服器的登入資訊。其做法是將「伺服器管理」群組，新增至**編輯伺服器鑑別許可權**中。

LDAP 整合

您可以將 Build Forge 設為搭配 LDAP 伺服器運作來登入使用者。透過使用 LDAP，使用者可以利用他們在您組織中其他位置所用的相同登入名稱和密碼，來登入 Build Forge。當您使用 LDAP 時，並不需要手動在 Build Forge 中建立使用者。當每一個使用者第一次登入 Build Forge 時，即會在 Build Forge 中建立該使用者。

您可以選擇將 LDAP 群組對映至 Build Forge 存取群組。此整合可讓您管理 LDAP 中的群組，並在使用者登入時自動更新使用者許可權。

您仍然可以手動建立和維護 Build Forge 系統中的使用者。其存取權必須手動管理。

如果要啓用此整合，您可以在**管理 > LDAP** 中建立項目。

註：只有 root 使用者可以在 Build Forge 使用者介面中建立及編輯「LDAP 網域」項目。

關於 LDAP 整合

當使用者初次使用 LDAP 認證登入 Build Forge 時，使用者會經過鑑別並設定在 Build Forge 中，如下所示：

重要：如果您想要使用群組對映，請在使用者登入之前，啓用 LDAP 群組對映。

如果在停用群組對映之後，有若干使用者登入，而您稍後又再啓用群組對映，則在現有的使用者上並不會執行對映。如果您在使用者登入後才啓用 LDAP 群組

對映，請從「Build Forge 使用者」清單中刪除使用者，將他們重新登入。這樣一來，Build Forge 存取群組的使用者成員資格便會以 LDAP 群組對映為基礎，而不是以您所做的任何手動變更為基礎。

1. 使用者會在登入畫面中看到**網域**欄位。如果配置多個網域，欄位會是一個下拉清單。使用者選取網域，然後登入。

註：如果您配置多個網域，則個別的唯一使用者 ID 在這些網域之間都必須是唯一的。系統只容許每個唯一使用者登入一次。如果一位使用者登入之後，有其他使用者利用相同的唯一使用者 ID 進行登入，便會關閉第一個使用者的階段作業。請參閱第 7 頁的『存取和使用主控台』，以取得使用者階段作業的相關資訊。

2. Build Forge 檢查 LDAP 伺服器上是否有該帳戶。您可以將 Build Forge 配置成利用一般使用者或管理使用者來執行檢查。
3. 若有找到使用者名稱，Build Forge 會試圖使用使用者在 Build Forge 登入畫面中提供的（或利用 API 用戶端從程式登入時使用的）認證，來登入 LDAP。
 - 如果認證不符，或者找不到使用者名稱，則登入失敗。
 - 如果認證相符，則繼續進行登入。
4. 如果使用者先前不曾登入，Build Forge 會自動在其使用者清單中建立一個使用者。如果使用者是透過 LDAP 登入，則會停用**使用者名稱、密碼、登入、確認和電子郵件**欄位，這是因為該資訊是由 LDAP 提供。

註：系統會在 LDAP 使用者第一次登入時，指派 *root* 使用者 時區給他們，這是因為系統沒有從 LDAP 取得時區資訊。之後，您可以手動設定時區。

5. Build Forge 會將存取群組套用至使用者。
 - 若有啟用 LDAP 群組對映，就會套用指定的存取群組。也會套用預設 Build Forge 存取群組。如果要啟用群組對映，必須配置 Build Forge LDAP 網域內容。

註：每當使用者登入時，就會執行群組對映。這可讓 Build Forge 與 LDAP 中的群組成員資格變更維持同步。

- 若未啟用 LDAP 群組對映，則套用 Build Forge 預設存取群組。而之後就可以手動管理存取群組成員資格。

LDAP 網域內容

如果要編輯所建立 LDAP 網域的內容，請執行下列動作：

1. 選取**管理 > LDAP<網域名稱>**
2. 選取要編輯的網域。內容會顯示在「LDAP 網域內容」畫面中。

詳細資料

名稱:
 管理 DN:
 對映存取群組: ☐

主機: <input type="text" value="ldap.example.com:389"/>	顯示名稱: <input type="text" value="displayname"/>	授權的群組 DN: <input type="text"/>
密碼: <input type="text"/>	識別名稱: <input type="text" value="distinguishedname"/>	寫入權群組 DN: <input type="text"/>
驗證: <input type="text"/>	群組名稱: <input type="text" value="memberof"/>	搜尋庫: <input type="text" value="cn=users,dc=example,dc=com"/>
連結使用者帳戶: <input type="checkbox"/> 是 <input type="checkbox"/> 否	郵件名稱: <input type="text" value="mail"/>	群組搜尋庫: <input type="text"/>
通訊協定: <input type="text" value="LDAP"/>	唯一 ID: <input type="text" value="sAMAccountName=%"/>	群組唯一 ID: <input type="text"/>

3. 編輯任何欄位的值，然後按一下**儲存**。以下是必要欄位：

- 名稱
- 主機
- 連結使用者帳戶
- 通訊協定
- 顯示名稱
- 識別名稱
- 郵件名稱
- 唯一 ID

4. 如果您想將網域設為預設的使用網域，請按一下**設為預設值**。

名稱 （必要）。LDAP 網域在 Build Forge 中的名稱。如果至少配置了一個 LDAP 網域，Build Forge 登入表單會依這個名稱來列出它們。

管理 DN

用來提供 LDAP 伺服器資料庫搜尋存取權的帳戶。如果您的伺服器容許使用匿名連結來搜尋資料庫，請將這個欄位留白。

部分 LDAP 伺服器需要管理連結來搜尋資料庫。這項設定可讓您指定管理者帳戶的 DN，如下例所示。

cn=Administrator,cn=users,dc=example,dc=com

在**密碼**和**驗證密碼**欄位中，指定「管理 DN」帳戶的密碼。

對映存取群組

決定是否要將 LDAP 伺服器中的對映群組資訊，對映至「管理主控台」中的存取群組。預設值為「否」。Build Forge 中每一個存取群組的 **LDAP 群組 DN** 內容，必須設定成 LDAP 中的正確群組名稱。

- 若為**否**，就不會將 LDAP 群組對映至 Build Forge 存取群組。在使用者至少登入一次後，您可以將使用者指派給 Build Forge 中的存取群組。使用這個選項時，意味著您管理的是 Build Forge 中之使用者的存取群組。當使用者第一次登入，並在 Build Forge 中建立使用者名稱時，會套用預設存取群組。
- 若為**是**，每當使用者登入 Build Forge 時，Build Forge 會重新整理 LDAP 伺服器中該使用者的群組成員資格資訊。自使用者前次登入以來，在 Build Forge 中對該使用者之存取群組成員資格所做的任何變更都會被改寫。使用這個選項時，意味著您管理的是 LDAP 中的所有群組成員資格。LDAP 群組成

員資格會自動對映（新增或移除）至 Build Forge 中的存取群組。如下所示，會使用群組內容來判斷使用者的群組成員資格：

1. 如果**群組名稱**不是空白，請查詢指定關鍵字值。請使用傳回的值，作為使用者的群組。
2. 如果**群組名稱**空白，或者其查詢沒有傳回一值，請使用**群組搜尋基準與群組唯一 ID**來查詢使用者隸屬的 LDAP 群組。
3. 如果 (1) 和 (2) 中沒有傳回任何群組資訊，則容許使用者登入，並將指定作為新使用者之預設存取群組的存取群組中的成員資格指派給該使用者。

主機 (必要)。LDAP 伺服器的主機名稱和埠。範例：

```
ldapserver.mycompanyname.com  
ldap.mycompany.com:9000
```

密碼 「管理 DN」帳戶的密碼。若有指定**管理 DN**，則此項為必要的。

驗證 重複輸入「管理 DN」的密碼。

連結使用者帳戶

(必要)。決定 Build Forge 在登入時，是否要試圖向 LDAP 驗證使用者的認證。預設值為「是」。

- 若為**是**，則 Build Forge 會向 LDAP 伺服器確認登入時所提供的使用者名稱與密碼。
- 若為**否**，則 Build Forge 會接受使用者名稱，不會進行驗證。當針對 Build Forge 實作「單一登入 (SSO)」之類的外部密碼驗證時，會使用這項設定。

通訊協定

(必要)。識別 Build Forge 從目錄服務讀寫資料，以鑑別 Build Forge 使用者時所用的通訊協定。預設值為 LDAP。如果您使用的是 LDAP over SSL (LDAPS)，請輸入 LDAPS。這個選項需要其他的設定。請參閱啟用安全 LDAP (LDAPS)。

顯示名稱

(必要)。輸入指定使用者完整名稱的索引鍵名稱。

識別名稱

(必要)。輸入指定使用者帳戶之「識別名稱」的索引鍵名稱。

郵件名稱

(必要)。輸入指定使用者電子郵件位址的索引鍵名稱。

群組名稱

輸入 LDAP 綱目中保留使用者所隸屬之群組清單的索引鍵名稱。只有在**對映存取群組**為「是」，或在使用**授權的群組 DN**時，才會使用此選項。

授權的群組 DN

LDAP 群組的識別名稱。如果設定此選項，則只容許指定群組的成員登入。如果空白，則任何有效的 LDAP 使用者都可以登入主控台。

寫入權群組 DN

判斷使用者具有一般還是唯讀存取權。其值可以是下列其中一項：

- 空白 - 若為新登入，使用者類型會設為「一般」。現有使用者將保持已指派的使用者類型（「一般」、「唯讀」或 API）。在**管理 > 使用者**中，可以設定類型。

- * (星號) - 所有登入的使用者類型皆為「一般」。
- LDAP 群組名稱 - 如果使用者屬於此群組，則其類型會設為「一般」。如果使用者不屬於此群組，則其類型會設定成「唯讀」。
- 其他 - 使用任何其他值，會將所有使用者強制設定成「唯讀」。例如：RO。

搜尋庫 (必要)。用來在 LDAP 記錄中查詢使用者的搜尋字串。範例：

`cn=users,dc=buildforge,dc=com`

唯一 ID

(必要)。識別 LDAP 資料庫中的欄位，以便用來和使用者在登入時所輸入的使用者名稱相比較。使用 % 字元作為使用者輸入的登入名稱。範例：

`(sAMAccountName=%)`

群組搜尋庫

需要**群組唯一 ID**。只有在對映存取群組為「是」，或在使用授權的群組 DN 時，才會使用此選項。用來在 LDAP 記錄中查詢群組資料的搜尋字串。如果您的 LDAP 資料庫用來儲存群組成員資格的資料庫，有別於用來儲存使用者記錄的資料庫，則此項為必要的。範例：

`cn=groups,dc=buildforge,dc=com`

群組唯一 ID

需要**群組搜尋基準**。只有在對映存取群組為「是」，或在使用授權的群組 DN 時，才會使用此選項。識別 LDAP 使用者資料庫中，用來取得群組成員資格資訊的欄位。過濾器可以利用使用者帳戶的任何資料欄位，作為群組表格的索引鍵。請使用 %*fieldname*% 語法來識別欄位。如果您的群組表格使用 sAMAccountname 欄位作為使用者的索引鍵，就可適用下列範例。

`sAMAccountName=%sAMAccountname%`

作業

這些主題會識別使用 LDAP 網域的作業。

建立 LDAP 網域項目

您可以建立任意數量的 LDAP 網域項目。當使用者試圖登入時，必須指定「網域」。如果採行 API 用戶端登入，網域必須指定在登入呼叫中，或者要使用的網域必須配置在 `bfcclient.conf` 中。

如果要新增網域項目，請執行下列動作：

1. 選取**管理 > LDAP**。
2. 按一下**新增 LDAP 網域**。
3. 填寫或變更網域的內容。**名稱**內容供 Build Forge 內部使用。依預設所提供的值，是設計成搭配 LDAP 或標準 Microsoft® Active Directory 伺服器運作。
4. 按一下**儲存**。

測試 LDAP 網域項目

如果要驗證您的 LDAP 網域設定無誤，請執行下列動作：

1. 選取**管理 > LDAP**。
2. 從清單中選取一個網域。
3. 按一下**測試 LDAP 網域**。

系統會利用 LDAP 網域項目中的內容來查詢 LDAP 伺服器。

啓用安全 LDAP (LDAPS)

如果您的 LDAP 伺服器支援 LDAP over SSL (LDAPS)，您也可以將 Build Forge 網域項目配置成使用 LDAPS。依預設，會配置嚴格的 SSL。嚴格的 SSL 需要伺服器憑證。

1. 在 Build Forge 中建立 LDAP 網域項目。
2. 將**通訊協定**內容設為 LDAPS。這會啓用 LDAPS 的僅加密方法。
3. 將**主機**設定成您 LDAP 伺服器的完整網域名稱與 SSL 埠。埠 636 是定義給嚴格安全 LDAP 的預設值。例如：myldap.mycompany.com:636。
4. 從 LDAP 伺服器取得簽章者憑證，並將它新增至 Build Forge 信任儲存庫。依預設，在**管理 > 安全**中，會將出埠 LDAP 配置成使用下列設定：
 - SSL 畫面：預設 JSSE 出埠 SSL
 - 「金鑰儲存庫」畫面：預設 JSSE 信任儲存庫。依預設，會將此信任儲存庫設定成使用 `<bfinstall>/keystore/buildForgeTrustStore.p12`。請將簽章者憑證放在這裡。
5. 重新啓動 Build Forge。
6. 跳至**管理 > 安全**，並選取您的安全 LDAP 配置。
7. 按一下**測試連線**。

註：依預設，會在 Build Forge 中設定嚴格的 LDAPS SSL。嚴格的配置需要進行伺服器憑證驗證。如果您不想使用嚴格的 LDAP，請執行下列動作：

1. 在 `JAVA_OPTS` 環境變數中，設定 `Tomcat` 系統內容 `-Dcom.buildforge.services.server.ldap.strict=false`。Tomcat Script 會讀取這個變數，並將任何指定的系統內容套用至 Tomcat 程序。
2. 重新啓動 Build Forge。

在這項配置中，您不需將 LDAP 伺服器憑證新增至 Build Forge 信任儲存庫。不過，這項配置只是些微實作 SSL 通訊協定設計。Build Forge 在和 LDAP 伺服器通訊期間，並不會驗證其身分。

變更 LDAPS SSL 配置

依預設，會設定出埠 LDAP 要求使用的 SSL 配置。其中的兩個層面可以變更：

- SSL 配置。如果 LDAP 伺服器無法利用預設通訊協定或信號交換，來和 Build Forge 通訊，就必須執行此動作。
- 金鑰儲存庫配置。嚴格的 SSL 會要求您將簽章者憑證放在用戶端使用的信任儲存庫 (Build Forge) 中，以供用戶端和 LDAP 伺服器進行安全的通訊。如果您想使用不同的信任儲存庫，或將它放在不同的位置，您必須在 Build Forge 中為信任儲存庫建立一項新的金鑰儲存庫配置。

這些指示是假設您已針對 Build Forge 啓用安全 LDAPS，且尚未針對 Build Forge 元件啓用 SSL。

如果要變更 LDAPS SSL 配置，請執行下列動作：

1. 如果您要變更信任儲存庫的位置或名稱，請將它放在 Build Forge 主機上的所要位置。在該位置中新增 LDAP 伺服器的簽章者憑證。

2. 必要的話，在**管理 > 安全 > 金鑰儲存庫**中建立信任儲存庫配置。信任儲存庫配置包含信任儲存庫的位置和名稱等內容。
3. 必要的話，在**管理 > 安全 > SSL**中建立 SSL 配置。將它配置成使用新的信任儲存庫配置（如果您已建立一個的話）。視需要對配置進行其他的調整。
4. 在**管理 > 安全**中，將 **SSL 已啓用**設為「是」（如果尚未設定的話）。這時會出現其他的欄位。
5. 在**出埠 LDAP** 清單中，選取您已建立的 SSL 配置。請勿變更其他設定。
6. 按一下**儲存**。
7. 按一下**更新主要 BFClient.conf**。
8. 如果之前沒有啓用 SSL，請執行下列動作：
 - a. 按一下 **SSL 已啓用**將其設為「否」。
 - b. 按一下**儲存**。
 - c. 按一下**更新主要 BFClient.conf**。
9. 重新啓動 Build Forge。
10. 在**管理 > LDAP**中，選取您的 LDAP 配置。
11. 按一下**測試連線**。

關閉 LDAP/Active Directory 支援

如果要中斷向 LDAP 或 Active Directory 進行鑑別，請執行下列動作：

1. 選取**管理 > LDAP**。
2. 刪除所有的網域項目。按一下網域項目旁的垃圾筒圖示，以便將它刪除。

當沒有網域存在時，只有手動新增至 Build Forge 的使用者可以登入。

系統配置設定

您可以利用各種設定來配置管理主控台。在**管理 > 系統**頁面上尋找這些設定。

當您按一下**管理 > 系統**時，系統會顯示設定清單。按一下系統設定的名稱，即可顯示設定的編輯畫面。

註：針對會獲得數值的系統設定，管理主控台會接受由一個以上整數 (0 - 9) 所組成的任何值。但不支援數值分組字元，如逗點 (,)、小數點 (.) 以及其他非整數分隔字元。

畫面包含下列按鈕：

- **儲存** - 儲存對設定值的變更。
- **還原為預設值** - 將設定重設為其預設值。

下表說明可用的設定。

設定	說明
帳戶鎖定消除時間	預設值：60 分鐘。在此段時間之後，系統會捨棄前次失敗的登入嘗試。如果設為負值，則不會捨棄失敗的登入。

設定	說明
帳戶鎖定嘗試次數上限	預設值：3。鎖定帳戶之前容許的連續嘗試登入失敗次數。如果設為 0，則會停用帳戶鎖定。
帳戶鎖定重設計時器	<p>預設值：120 分鐘。下限：5 分鐘。</p> <p>允許使用者在鎖定之後嘗試再次登入所需的時間。如果設為負值，鎖定將永不到期。</p> <p>註：管理者必須重設使用者，使用者才能再次登入。</p>
警示電子郵件限制	設定系統在指定的分鐘數內傳送的警示電子郵件數目上限。例如，10/60 這個值將上限設為每小時 10 個訊息。預設值 0/0 表示警示電子郵件訊息數目沒有限制。
套用列入的步驟儲存器環境	<p>預設值：否。如果為「是」，則會套用包含列入步驟之專案或程式庫的環境。</p> <p>註：在 Java 引擎上不受支援。</p>
最後套用伺服器環境	<p>預設值：否。如果為「是」，則會在最後時才套用步驟的伺服器環境。伺服器環境是在步驟環境或專案環境之後套用。這只有在指定環境時才適用。</p> <p>註：在 Java 引擎上不受支援。</p>
審核字串長度上限	<p>預設值：128 個字元。</p> <p>控制可以為詳細變更記錄的資訊量。如果字串值超出此字元數，則會儲存加密雜湊，而非原始值。加密雜湊可節省儲存空間。若為包含機密性資訊（如密碼）的欄位，則無論字串長度為何，都會使用加密雜湊。</p>
審核原則	<p>決定審核記載量。</p> <p>NONE 不記錄審核資訊。</p> <p>SECURITY 僅記錄安全相關資訊。顯示下列範例：</p> <ul style="list-style-type: none"> 存取群組建立 使用者階段作業活動 <p>BASIC 此設定是預設值。如果審核原則值無效，則也會使用此值。僅會記錄最少的資訊；例如，會記錄使用者、物件類型、使用者 ID 及動作。</p> <p>DETAILED 記錄已變更的內容的相關詳細資訊，包括舊值及新值。並非所有物件皆可提供此資訊。對於無法提供此資訊的物件，簡略與詳細原則之間沒有任何差異。使用「審核字串長度上限」系統配置設定可限制儲存的資料量。</p>
自動登出分鐘數	系統會自動登出閒置使用者。此設定指定在系統登出使用者之前必須經過的閒置時間分鐘數。當設為 0 時，系統不會自動登出使用者。

設定	說明
AutoClean 錯誤日誌天數 AutoClean 參考日誌天數 AutoClean 警告日誌天數	這些值表示每一個項目種類留在審核日誌中的最大天數。會自動刪除較舊的項目。如果值為 0，則系統永不會刪除該種類的項目。字串值在求值後為 0（代表整數），可讓您使用 Never 之類的值來代替 0。
建置取消檢查頻率	指定系統檢查建置取消要求的頻率（以秒為單位）。
主控台埠	Web 伺服器用來接聽 Build Forge 要求的埠號。
主控台 URL	Web 伺服器用來接聽 Build Forge 要求的 URL。 如果主控台在非 80 的埠上執行，則必須設定此值。如果設定此值，則會將預設的主控台 URL 置換為此值。它會採用 <code>protocol://hostname[:port]</code> 格式。範例： <code>http://myHost:81</code> 。
失敗時繼續進行步驟部分	預設值：否。此設定決定在步驟部分失敗時，引擎是否繼續處理。若設為是，失敗之後，程序會繼續。若設為否，現行步驟部分程序會停止，同時程序會繼續建置中的下一個步驟。
建立遺漏的路徑	預設值：否。如果為「是」，則系統會建立專案的路徑（如果該路徑尚未存在的話）。
資料庫大小臨界值	資料庫大小達到此臨界值時，主控台就會傳送通知。 記住： 當逼近臨界值時，使用者介面效能會降到最低。執行資料庫清理或增加臨界值大小可以還原效能。 預設值：2G。 註： 在 Java 引擎上不受支援。
資料庫大小臨界值通知	資料庫大小達到臨界值時，要傳送電子郵件的目標使用者名稱或通知群組。如果利用使用者名稱，則會使用該使用者的 電子郵件欄位 中的位址。請參閱 管理 > 使用者 。 註： 在 Java 引擎上不受支援。
設定一次日期	決定 .date 指令值是每個工作設定一次，還是每個步驟設定一次。此行為取決於所指派的值，如下所示： <ul style="list-style-type: none"> 預設值：否。每個步驟設定 .date 指令值一次。每個步驟均會使用代理程式機器的時區，對 .date 值重新求值。代理程式會提供值。重新啟動後仍會保持 .date 值。對這些值重新求值的方式，與起始執行時完全相同。在使用相同變數的每個步驟中，相同變數可能會有不同的值，因為會針對每個步驟重新解譯 .date 值。 是：每個工作設定現行 .date 指令值一次。此日期值是使用引擎主機的時區（而非代理程式的時區）產生的。重新啟動後仍會保持 .date 值。 當您在電子郵件通知（例如 .email 指令或通知範本）中使用包含 .date 的環境變數時，請輸入是。電子郵件中所使用的環境變數即會使用引擎的時區。
預設代理程式連接埠	設定用來建立代理程式連線的預設埠號。

設定	說明
預設匯入類別	<p>如果匯入的專案沒有定義類別，或者有不存在的類別，就會使用這個類別。</p> <p>預設值：正式作業</p>
摘要演算法	<p>訊息摘要的演算法。代理程式配置中也有對應的 <code>digest_algorithm</code> 設定。有效值為 <i>SHA1</i> 和 <i>SHA2</i>。</p> <p>預設演算法：<i>SHA1</i></p> <p>註：若要使用整個安全雜湊演算法 2 (SHA2)，並在 Build Forge 管理主控台及代理程式中啟用密碼加密，那麼在管理 > 安全中將 <i>SHA1</i> 變更為 <i>SHA2</i> 之後，可能需要更新密碼加密配置內容檔 <code>bfpassword.conf</code>。</p> <p>若已啟用密碼加密，請參閱第 167 頁的『如果已啟用密碼加密』，以了解下一步做什麼的指示。</p> <p>若未啟用密碼加密，請參閱第 168 頁的『如果未啟用密碼加密』，以了解下一步做什麼的指示。</p>
啟用快速啟動	<p>預設值：否。所有專案都會顯示下列圖示：。當您按一下專案時，會檢查專案的所有併入的變數，找出必須變更變數類型。如果專案不含必須變更變數，就會啟動專案。如果專案包含必須變更變數，就不會啟動專案。會有一個對話框</p> <p>說明不啟動的原因。下列圖示會變更顯示：。</p> <p>註：只會檢查為專案定義的變數。不會檢查為專案中步驟所定義的變數。另請參閱第 264 頁的『啟動工作時變更專案變數』。</p> <p>如果設為是，「專案」頁面會檢查專案的所有環境，以判斷是否有任何變數設為必須變更類型。下列圖示表示適合立即啟動的專案：。</p>
強迫執行限制	<p>預設值：否。如果為是，則系統會啟動專案的排程建置，但前提是這樣啟動不違反專案的執行限制設定。如果為否，系統會忽略專案對排程建置的執行限制設定。</p>
匯入預設安全存取群組	<p>當使用安全存取來匯入設定是設為 <i>Y</i> 時，此設定會指出所匯入專案的預設存取群組。</p>
匯入不安全預設存取群組	<p>當使用安全存取來匯入設定是設為 <i>N</i> 時，此設定會指定所匯入專案的預設存取群組。唯有當匯入檔案缺少存取群組時，才會使用預設群組。</p>
使用安全存取來匯入	<p>當設為 <i>Y</i> 時，系統會指派之前設定中所列出的預設存取群組給匯入的資料物件。此值會置換您所匯入的 XML 檔中指定的任何存取群組，以協助避免因匯入資料而置換安全性。若設為 <i>N</i>，系統會允許使用所匯入檔案中的任何存取群組設定。</p>

設定	說明
繼承標籤	<p>設為是時，由鏈結所啟動的工作必須使用與呼叫端相同的工作標籤。如果專案 MasterProject 的 BUILD_15 呼叫專案 ComponentProject，則 ComponentProject 執行的工作標籤及工作目錄名稱會變成 BUILD_15。</p> <p>註：如果呼叫端標籤在執行期間變更，則被呼叫的專案一律會繼承呼叫端的原始標籤。舉例來說，由於 .retag 指令，被呼叫的專案會維護呼叫端開始時的標籤。</p>
無效相對目錄字元	設定的字元若使用於專案名稱中，將由系統變更為底線。
LASTRUN 格式	使用對 .date 指令定義的日期格式字元，輸入 BF_LASTRUN 格式環境變數的值。請參閱第 347 頁的『.date』。
授權伺服器	授權伺服器主機名稱。它是在安裝期間設定的。例如：myhost.mycompany.com。此值可能包括埠號。例如：myhost.mycompany.com:80。若要變更授權伺服器，請參閱第 23 頁的『變更「管理主控台」的授權伺服器』。
鏈結除錯模式	<p>當設為是時，已定義配接器鏈結的工作會執行鏈結測試，而不是執行相關聯的專案。工作輸出包含單一步驟，其中有配接器的輸出。當您進行配接器介面的疑難排解時，這個資料很有用。</p> <p>註：您可以將配接器鏈結的狀態設為除錯，以設定個別配接器鏈結的除錯功能。該狀態的優先順序高於「鏈結除錯模式」設定。</p>
鏈結手動工作	<p>決定當手動快速啟動專案，而不是從排程器執行時，系統是否透過配接器鏈結來執行配接器。</p> <p>鏈結檢查可能會在工作的 BOM 中產生其他輸出。</p> <p>如果設為 <i>N</i>，就不會在工作啟動時，檢查或執行鏈結。</p>
主控台處理程序數目上限	<p>設定主控台一次可執行的處理程序數目上限。使用它作為對主控台活動的一般節流控制。系統會在資料庫中儲存每一個處理程序的 ID，並在啟動新的外部處理程序之前檢查總計，以管理處理程序。請確定此值至少比執行佇列大小設定大 5。否則，系統無法執行足夠的處理程序來支援執行佇列。</p> <p>註：在 Java 引擎上不受支援。</p>
最大列入深度	<p>控制系統容許的專案列入層數，以避免專案的巢狀無限地增加。預設值是 32。如果此值設為 0，則系統會使用 32。當系統達到列入限制時，就不會執行超出此限制的列入專案。步驟不會插入到內含專案中。訊息會寫入系統訊息清單中：已放棄列入。</p> <p>註：在 Java 引擎上不受支援。</p>
同時進行的伺服器測試數上限	指定一次可以執行的伺服器測試數。視系統資源而定，一次執行太多伺服器測試會嚴重減緩或鎖定主控台。

設定	說明
同時執行的清除數上限	控制可同時執行多少清除。您可以清除任意數量的建置。不過，如果建置數量超出 同時執行的清除數上限 欄位中的值，則會同時刪除建置。預設值：20。
重新整理次數上限	自動重新整理網頁的次數上限。預設值：50。
鏈結時置換類別	決定系統是否將鏈結的專案類別取代為其呼叫端的類別。預設值 <i>Y</i> 會使系統置換鏈結的專案類別，並改用呼叫端類別。
密碼到期天數	指出在此天數之前，有設定密碼到期的使用者必須變更其密碼。當時間到期，使用者需要在下次登入時變更其密碼。
密碼格式	<p>使用以句點區隔的六欄位格式字串，來指定使用者密碼的需求：</p> <p><i>length.char_types.upper.lower.numeric.special</i></p> <p>例如：5.2.u1.l1.n1.s1。</p> <p>前兩個欄位指定下列項目：</p> <ul style="list-style-type: none"> 密碼長度下限（字元）。 要在其餘四個欄位中使用的字元類型數下限（1 到 4 的整數）。 <p>其餘欄位指定字元類型和頻率需求。每一個欄位都包含一種類型和一個號碼。</p> <ul style="list-style-type: none"> 類型：u（大寫）、l（小寫）、n（數字）或 s（特殊）其中之一。大寫（U、L、N、S）表示該字元為必要字元。小寫（u、l、n、s）表示該字元為選用字元。 數字：必要。表示必要的字元數（如果有使用其中任何一個的話）。 <p>其類型如下所示：</p> <ul style="list-style-type: none"> U 或 u 代表大寫字元。這些字元包括在其各自的語言環境中被視為字母，但並非小寫的所有字元。具體來說，就是包括下列的字元：大寫、首字母大寫，或無特定大小寫的語言（如中文）的任何字母。 L 或 l 代表小寫字元。這包括在其各自的語言環境中被視為小寫的所有字元。 N 或 n 代表數字字元。這些字元包括在其各自的語言環境中被視為數字的任何字元。 S 或 s 代表特殊字元。不符合上述三個種類的任何字元。其中包含所有既非字母，也非數字的字元。 <p>範例：字串 5.2.u1.l1.n1.s1 表示下列密碼需求：</p> <ul style="list-style-type: none"> 長度至少有 5 個字元。 至少必須含有四個種類（大寫、小寫、數字、特殊字元）中的兩種的字元。 對於每一種類型而言，該類型的一個字元代表符合需求的一個項目。 <p>例如，<i>abCl1x</i> 和 <i>Abc2%</i> 就是合格的密碼。</p>

設定	說明
暫停 Build Forge 引擎	當設為 <i>Y</i> 時，系統會完成任何目前的工作，然後暫停引擎。將它設為 <i>N</i> ，則會回到正常作業。
公用主機名稱	設定之後，系統會以這項設定值，取代通知範本中 <i>CONSOLEHOST</i> 變數的伺服器主機名稱。
清除檢查時間	設定系統檢查要清除之工作的頻率，以檢查間隔分鐘數表示。
QuickReport 公用目錄	<p>公用報告設計的檔案系統位置。</p> <p>在 7.1 中，請使用此系統設定來指定公用報告的完整位置。您的報告設計必須位於這個目錄中，才會自動移轉它們。</p> <p>在舊版中，預設檔案位置 (<i>../reports/public</i>) 是相對於應用程式伺服器的安裝目錄，例如：<i><bfinstall>/Apache/tomcat/webapps/quickReport</i>。</p>
QuickReport 暫時目錄	<p>在 7.1 中，請使用此目錄在與服務層元件相同的主機上，指定一個完整的目錄。服務層會使用這個工作目錄，來列出已順利移轉至資料庫的報告設計。</p> <p>在舊版中，在將「快速報告」的報告設計儲存至檔案系統上的公用或專用目錄之前，會先使用這個目錄來暫時儲存它們。</p>
QuickReport 使用者目錄	<p>專用報告設計的檔案系統位置。</p> <p>在 7.1 中，請使用此系統設定來指定專用報告的完整位置。您的報告設計必須位於這個目錄中，才會自動移轉它們。</p> <p>在舊版中，專用報告的預設檔案位置 (<i>../reports/users</i>) 是相對於應用程式伺服器的安裝目錄，例如：<i><bfinstall>/Apache/tomcat/webapps/quickReport</i>。</p>
重新載入語言套件	預設值：否。如果設為「是」，主控台會在重新啟動時重新載入其語言套件，並且將此值重設為否。從 7.0.1 版開始，不再需要此設定。
重設配接器範本	使用此設定來重設配接器範本（將更新版的變更複製到配置中）。若要使用它，請將值設為是，然後等待一分鐘。系統會重設範本，然後將此值設回否。
重設伺服器工作計數	<p>使用是這個設定，將所有伺服器的工作計數 (<i>BF_JOBS</i>) 同時重設為零。當執行資訊清單檢查間隔時，就會進行重設。預設值為每隔 10 秒。</p> <p>當所有伺服器的 <i>BF_JOBS</i> 全部重設之後，「重設伺服器工作計數」值就會回復為預設值否。</p>
重新啟動報告移轉	預設值：否。在 7.1 中，如果您不重新啟動服務層元件就要開始移轉，請將此值設為是。
執行鏈的鏈結	控制已啟動的鏈專案是否也會啟動任何連接的配接器鏈結。

設定	說明
執行佇列大小	此值會限制系統同時執行的工作數。當佇列中的執行數目大於或等於這個數字時，系統便會停止將「等待」佇列的執行移到「執行」佇列，直到工作數降到此值以下為止。如果您變更「執行佇列大小」，請檢查「主控台處理程序數目上限」設定，它應該比「執行佇列大小」至少大 5。
儲存開始環境	控制專案的手動起始頁上的儲存環境勾選框的預設值。當設為 Y，會選取勾選框。這是預設值。否則，不會選取勾選框。如果您選取儲存環境勾選框，則您所進行的任何變更將儲存到「啟動」頁面上的環境變數，以及資料庫中的環境記錄。未來執行的建置會預設為那些值。
鏈前的伺服器環境	決定系統是要在 Y 之前還是 N 之後設定步驟伺服器環境。這個值會在步驟內設定鏈結專案環境。在第二個環境中處理的變數會置換第一個環境中的變數。預設值為 Y，表示會以第二順位處理鏈結專案環境，並且會置換步驟伺服器環境。 註： 在 Java 引擎上不受支援。
伺服器讀取逾時	在系統停止嘗試讀取代理程式之前等待的時間（秒）。 註： 在 8.0 中新增的功能。僅用於 Java 主要工作控制器。
伺服器重試次數	設定當找到符合所有選取元條件的伺服器時，系統在停止並放棄步驟之前，嘗試將步驟配置至伺服器的次數。
伺服器寫入逾時	在系統停止嘗試寫入代理程式之前等待的時間（秒）。 註： 在 8.0 中新增的功能。僅用於 Java 主要工作控制器。
伺服器測試頻率	<p>與已啓用的伺服器數目一起用於決定測試及重新整理伺服器資訊清單資料的頻率。預設值是 120 分鐘或 2 小時。值 0 表示您不需要檢查伺服器。</p> <p>在這些檢查期間，系統會聯絡所有已啓用的伺服器來驗證：</p> <ul style="list-style-type: none"> 仍可抵達伺服器。 伺服器的登入資訊是正確的。 伺服器的資訊清單資料是最新資料。 <p>執行伺服器測試的速率下限是每分鐘一台伺服器。如果伺服器數目大於此設定，該速率會增加。</p> <p>系統會在該段間隔內平均分散這些測試活動。例如：</p> <ul style="list-style-type: none"> 您擁有 120 台伺服器且間隔設為 120 分鐘。系統會嘗試每分鐘測試 1 台伺服器（120 台伺服器/120 分鐘）。 您擁有 12,000 台伺服器且間隔設為 1200 分鐘。系統會嘗試每分鐘測試 10 台伺服器。 您擁有 10 台伺服器且間隔設為 120 分鐘。每分鐘會檢查 1 台伺服器。在此實務範例中的更新間隔期間，會檢查每台伺服器 12 次。 <p>手動測試伺服器從主控台開始，優先順序高於自動化測試。</p> <p>伺服器收集器的複雜度會影響傳輸量。執行許多手動指令以收集資料的收集器，可能需要超過一分鐘來完成資料收集。</p>

設定	說明
伺服器使用連接逾時值	<p>指出在建立代理程式連線之後，在連線失敗之前，管理主控台等待連線開啓的秒數上限。如果在達到逾時期間之前發生 Socket 錯誤或程式碼失敗，連線也可能會失敗。</p> <p>指出連至代理程式之現有連線的逾時值。代理程式的設計是每 15 秒聯絡管理主控台一次。如果在逾時期間內未聯絡到，代理程式可能已停止，或者可能是發生網路通訊問題。如果超過此值，步驟會失敗。</p>
伺服器等待時間	<p>設定檢查以判定伺服器是否變成可用的間隔秒數。</p> <p>註：在 8.0 版中新增的功能。僅用於 Java 主要工作控制器。</p>
服務層級鑑別 Servlet URL	<p>當設定此項時，會將程式化方式建構的 URL 置換為服務層級鑑別 Servlet。</p> <p>如果您使用別名或非預設埠，則必須使用下列格式來更新這項設定：<code>http://server:port/jas/AuthServlet</code></p>
SMTP 伺服器	<p>設定在傳送電子郵件通知時要作為 SMTP 伺服器的電腦。</p> <p>預設值為 <code>localhost</code>。</p>
堆疊 BuildForge 環境變數	<p>系統通常會把下傳至鏈結專案的 BF_ 變數名稱變更為 BF_CALLER_。當鏈結深度不止一層時，此設定將決定系統是否會堆疊名稱。預設值為 <code>N</code>。當設定變更為 <code>Y</code> 時，從深入呼叫端專案兩層所衍生的 BF_TAG 變數會接收 BF_CALLER_CALLER_TAG 名稱。</p>
步驟重試數上限	<p>控制如果第一次嘗試失敗，步驟會嘗試連接代理程式的次數。如果步驟在指定的重試次數內未連接，步驟就會失敗。</p>
在本端儲存使用者鑑別	<p>決定系統是否快取加密格式的 LDAP 或 Active Directory 使用者鑑別資訊。預設值為是。僅當您使用 LDAP 或 Active Directory 鑑別時，才會與系統有關。當設定為是時，系統會快取加密格式的使用者鑑別資訊，並將其與 <code>_USE_BFCREDS</code> 和 <code>_USE_BFCREDS_DOMAIN</code> 特殊變數搭配使用。</p> <p>註：此值會將使用者鑑別套用至伺服器。您可以將值變更為否來關閉快取。然而，如果您關閉快取，則當登入伺服器時，系統無法使用 <code>_USE_BFCREDS</code> 和 <code>_USE_BFCREDS_DOMAIN</code> 特殊變數來利用使用者認證。</p>
系統警示電子郵件	<p>系統會傳送警示電子郵件訊息到此設定所定義的位址。</p> <p>預設值為 <code>root@localhost</code>。支援下列電子郵件位址格式：</p> <p>使用者名稱@host.com 名稱使用者名稱@host.com 名稱 "<使用者名稱@host.com>" 名稱 <使用者名稱@host.com> 名稱 "使用者名稱@host.com"</p> <p>其中</p> <p>使用者名稱是電子郵件使用者名稱。</p> <p>名稱 是任意字串，通常是某形式的給定名稱。</p> <p>若要傳送至多個位址，請以逗點 (,) 或分號 (;) 區隔它們。</p>

設定	說明
系統警示來源	<p>當系統傳送警示電子郵件訊息時，它會使用此設定中定義的位址作為傳送者。</p> <p>預設值為 <code>root@localhost</code>。支援下列電子郵件位址格式：</p> <p>使用者名稱@host.com 名稱 使用者名稱@host.com 名稱 "<使用者名稱@host.com>" 名稱 <使用者名稱@host.com> 名稱 "使用者名稱@host.com"</p> <p>其中</p> <p>使用者名稱 是電子郵件使用者名稱。</p> <p>名稱 是任意字串，通常是某形式的給定名稱。</p>
全系統登入訊息	可讓您定義要顯示在登入表單上方的訊息。
全系統使用者訊息	可讓您定義要顯示在每一頁頂端的訊息，它就位於導覽按鈕的下面。
標籤：日期格式	<p>定義用來顯示 BF_D 標籤變數之日期的格式。使用 <i>y</i>、<i>m</i> 和 <i>d</i> 字元作為年、月、日的變數。此值會顯示您想要的格式和任何特殊字元作為分隔符號。例如，若日期為 2005 年 9 月 21 日，則會顯示下列：</p> <p>格式字串....輸出</p> <p>ymd...050921</p> <p>m/d/y...09/21/05</p>
標籤：時間格式	<p>定義用來顯示 BF_T 標籤變數中的時間的格式。「標籤：時間格式」設定使用 <i>h</i>、<i>m</i> 和 <i>s</i> 字元來代表小時、分鐘和秒。<i>h:m:s</i> 設定會呈現下列輸出：12:53:42。</p>
郵件範本的結尾日誌數量	設定日誌結尾的行數，在通知範本中使用 <code>TAILNORMALLOG</code> 變數時，這些行數會顯示在通知中。
終止執行緒	決定含執行緒作業的步驟何時失敗。相同專案中的所有其他作用中執行緒區塊都會停止。
追蹤代理程式資料鏈結	<p>指出是否對主控台與代理程式之間的通訊啟用詳細追蹤。</p> <p>預設值：否</p>
使用 Java 主要工作控制器作為工作執行引擎	<p>設為是時，系統會使用 Java 主要工作控制器來執行主要工作控制器專案。無法啟動 Perl 引擎專案。新建的專案預設為使用主要工作控制器，而不是 Perl 引擎。</p> <p>設為否時，系統只會使用 Perl 引擎來執行 Perl 引擎專案。</p>

其他系統設定

有些配置設定是在系統外部進行設定。

I/O 逾時

依預設，I/O 逾時為 300 秒。該值可透過建立 `BF_SOCKET_SELECT_TIMEOUT` 作為作業系統變數，並為其指定一個值（以秒為單位）來進行調整。該值不得

設為 60 秒以下。只有在啟動時才會讀取該值。若要變更該值，必須停止系統，再重新啟動。如果該值過低，則在大量平行負載（如同時啟動許多工作）下，系統可能會像當機。預設值適用於大部分狀況。

審核記載

審核記載機能提供指令行介面，可用於管理審核記載及檢視審核日誌。審核日誌可提供系統中的物件變更歷程。它們按全文索引進行儲存。

如果使用審核記載，則對於它所追蹤的物件，其記載機能會取代傳訊機能（可在**管理 > 訊息**中看到）。

重要：審核記載僅適用於在執行中系統上直接執行的作業。它不適用於匯入至系統的資料。bfimport 機能會直接更新資料庫，而不與審核記載機能互動。

管理審核原則

審核原則可控制執行審核的數量。它決定於「審核原則」系統配置設定的設定。

NONE 不記錄審核資訊。

SECURITY

僅記錄安全相關資訊。範例：建立存取群組及使用者階段作業活動。

BASIC

此為預設值。如果存在無效的「審核原則」值，則也會使用此值。僅會記錄最少的資訊，如使用者、物件類型、UUID 及動作。

DETAILED

記錄哪些內容已變更的相關詳細資訊，包含舊值及新值。並非所有物件皆可提供此資訊。對於無法提供此資訊的物件，「簡略」與「詳細」原則之間沒有任何差異。「審核字串長度上限」系統配置設定，可用於限制儲存的資料數量。

「審核字串長度上限」系統配置設定，可透過設定字串資料欄位的長度上限，來控制可以為 DETAILED 變更記錄的資訊量。如果審核項目超出此字元數，則會儲存 SHA 訊息摘要（而非字串值）。若為包含機密性資訊（如密碼）的欄位，則無論字串長度為何，皆會使用訊息摘要。

顯示審核資訊

指令行介面可讓您指定要顯示的資訊，以及如何格式化資訊。

需要「檢視所有審核日誌」許可權。依預設，此許可權會指派給「安全及建置工程師」存取群組。

用法：bfauditlog [connection-options][command-options][query]

文字執行檔如下所示：

- Windows 系統：bfauditlog.bat
- UNIX 及 Linux 系統：bfauditlog.sh

連線選項

-H hostname | --hostname=hostname

連線的遠端主機名稱。預設值為 localhost。

- P *port* | --port=*port***
連線的通訊埠。預設值為 3966。
- d *domain* | --domain=*domain***
鑑別的網域或領域。預設值：無。
- u *login* | --user=*login***
用於鑑別的使用者名稱。預設值為 root。
- p *password* | --password=*password***
鑑別使用者的密碼。預設值為 root。
- E *encoding* | --encoding=*encoding***
要使用的輸出字元集編碼。如果指定的編碼無效，便會列出所有可用的編碼且程式會結束。請注意，視用戶端上使用終端機軟體的功能而定，來決定是否能夠以所要求的編碼來顯示文字。預設值會視您的作業系統和 JVM 支援而定。

指令選項

- h** 顯示說明。
- b | --bare**
在輸出中顯示較少資訊（簡略）。確切的效果取決於輸出格式（請參閱 -t）。例如，在 XML 輸出中，將會省略開始標籤、結束標籤及列計數。在 CSV 中，將會省略列計數及直欄名稱。
- c *auditClass[,auditClass ...]* | --audit.class=*class-list***
指定要在查詢中使用的審核類別清單。請以逗點分隔多個類別名稱。如果指定此選項多次，便會使用所有指定審核類別的聯集。如果沒有指定該選項，便會使用使用者可以指定的所有審核類別。使用 -L 選項來列出可用的審核類別。預設值為授權使用者可檢視的所有審核類別。
- f *field[,field ...]* | --field=*field-list***
指定輸出中要包含的欄位。請以逗點分隔多個欄位名稱。使用 -L 選項來列出可用的欄位。
- s *field[+|-][,field[+|-] ...]* | --field=*field-list***
列出用於對輸出進行排序的欄位。依預設，輸出會以 audit.stamp 的降冪排序。+ 標記指定升冪。- 標記指定降冪。請以逗點分隔多個欄位名稱。預設值為 -audit.stam。
- l [*offset*,]*count* | --limit=[*offset*,]*count***
用於為結果標頁數。偏移與計數以項目數表示。例如，“-l 100,50” 表示顯示第 100 個項目後的 50 個項目。非簡略輸出格式將在輸出中包含符合的項目總數。
- a *user[,user ...]* | --audit.user=*user-list***
將輸出限制為與指定使用者相關聯的項目。請以逗點分隔多個使用者名稱。如果您使用多個 -a 選項，指令會使用在選項中指定的所有使用者聯集。可以使用登入 ID 或 UUID 來指定使用者。如果 -- audit-user 指定時沒有搭配使用者引數，則會使用鑑別時所使用的使用者。
- t *format* | --format=*format***
要使用的輸出格式為 CSV、XML、JSON 或 TEXT 其中之一。預設值為 TEXT。
- L | --list**
列出使用者可用的審核類別。以星號 (*) 標示的審核類別，即為該使用者可存取的審核類別。如果與 -c（審核類別清單）搭配使用，便會改為顯示各類別可用的欄位

清單。當指定此選項時，其他指令選項皆不會發揮作用。具有 ViewAllAudits 許可權的使用者可以檢視任何類別的審核日誌。其他使用者只可以使用 -a 選項來檢視自己的審核日誌。

查詢

查詢是使用 Web 搜尋引擎常用格式的字串：以空格分隔的指定項目清單。指定項目的格式為 'field:value'，表示在指定 field 中搜尋 value。如果沒有指定查詢，但指定有至少一個選項，便會顯示所有符合的審核日誌。無法在查詢中指定審核使用者和審核類別。請使用指令選項來指定它們。

註：請在 Windows 系統上使用雙引號，在 UNIX 及 Linux 系統上使用單引號。

範例

1. 此範例會列出 Project 以及 Step 審核類別的所有已知查詢欄位。
 - Windows

```
$ bfauditlog.bat -c Project,Step -L
```
 - UNIX 及 Linux

```
$ bfauditlog.sh -c Project,Step -L
```
2. 此範例會顯示 10 個最近專案建立作業的審核日誌，它使用以逗點區隔的變數輸出格式，且沒有直欄標題或列的總計數。每一個記錄只有給定的專案唯一 ID 和名稱。
 - Windows

```
$ bfauditlog.bat -c Project -t CSV -f audit.target,
  sxProjectName --bare -l 10 "audit.type:Create"
```
 - UNIX 及 Linux

```
$ bfauditlog.sh -c Project -t CSV -f audit.target,
  sxProjectName --bare -l 10 'audit.type:Create'
```
3. 此範例會在使用提供的網域、使用者名稱和密碼登入後，從機器 'buildforge.example.com' 上安裝作業擷取 100 個最近的審核日誌。
 - Windows

```
$ bfauditlog.bat -H buildforge.example.com
  -d ExampleLDAP -u apiuser -p "My$ecret" -l 100
```
 - UNIX 及 Linux

```
$ bfauditlog.sh -H buildforge.example.com
  -d ExampleLDAP -u apiuser -p 'My$ecret' -l 100
```
4. 此範例會依據鑑別連線之使用者的時區設定，擷取自 2011 年 5 月 23 日當天開始到 2011 年 5 月 27 日當天結束之間所記錄的審核日誌。
 - Windows

```
$ bfauditlog.bat "audit.stamp:[2011/05/23 TO 2011/05/27]"
```
 - UNIX 及 Linux

```
$ bfauditlog.sh 'audit.stamp:[2011/05/23 TO 2011/05/27]'
```
5. 此範例會針對存取群組層次從舊值的 4、5 或 6 變更為新值 3 的專案，擷取審核日誌。請注意，方括弧是用來表示包含的範圍，所以 '[4 TO 6]' 表示值 4 和 6 皆符合。此外，也請注意，只有在審核原則設為 'DETAILED' 時，才可使用舊層次和新層次值來進行搜尋。
 - Windows

```
$ bfauditlog.bat -c Project "ioLevel:[4 TO 6] AND inLevel:3"
```

- UNIX 及 Linux

```
$ bfauditlog.sh -c Project 'ioLevel:[4 TO 6] AND inLevel:3'
```

6. 此範例會針對存取群組層次從舊值的 4 或 5 變更爲新值 2 或 3 的專案，擷取審核日誌。請注意，大括弧是用來表示不包含的範圍，所以 '{3 TO 6}' 表示值 3 和 6 不符合。此外，也請注意，只有在審核原則設爲 'DETAILED' 時，才可使用舊層次和新層次值來進行搜尋。

- Windows

```
$ bfauditlog.bat -c Project "ioLevel:{3 TO 6} AND inLevel:(2 OR 3)"
```

- UNIX 及 Linux

```
$ bfauditlog.sh -c Project 'ioLevel:{3 TO 6} AND inLevel:(2 OR 3)'
```

7. 下列範例限制查詢結果，僅查詢審核使用者 root 及 sv，並以 CSV 檔的形式產生輸出。它會傳回「專案」及「步驟」類別的資料，並且包含類別、類型、目標、使用者及時間戳記的資訊：

- Windows

```
$ bfauditlog.bat -c Project,Step -f audit.class,
audit.type,audit.target,audit.user,audit.stamp
-a root,sv -l 10,5 -t CSV
```

- UNIX 及 Linux

```
$ bfauditlog.sh -c Project,Step -f audit.class,
audit.type,audit.target,audit.user,audit.stamp
-a root,sv -l 10,5 -t CSV
```

類別

類別是審核日誌項目的標籤。使用 -L 選項來列出您可用的類別。

欄位

欄位代表指派給每筆審核日誌記錄的資訊。

- audit.class：專案（字串）
- audit.stamp：時間戳記（整數）
- audit.stamp.localized：代表 audit.stamp 整數值的本地化字串
- audit.type：更新（字串）
- audit.target：專案的 UUID（字串）
- audit.user：執行更新的使用者 UUID

審核日誌記錄中的部分欄位將反映其狀態（o 代表舊，n 代表新，x 代表延伸），以及其資料類型（s 代表字串，i 代表整數）。範例：

```
soDescription：舊專案說明（字串）
snDescription：新專案說明（字串）
soSelectorUuid：舊選取元的 UUID（字串）
snSelectorUuid：新選取元的 UUID（字串）
ioRunLimit：舊執行限制（整數）
inRunLimit：新執行限制（整數）
```

訊息

系統訊息頁面會顯示系統及使用者動作的日誌，以供您檢閱。

請選取**管理 > 訊息**來顯示訊息的清單。這些訊息組成動作的資訊軌跡。



您可以使用一或多個所提供的控制項，然後按一下**過濾**來過濾清單：

- 文字框：輸入要搜尋的字串。
- 嚴重性：選取下列其中一項：
 - 全部（預設值）
 - 錯誤
 - 參考
 - 警告
 - 審核
- 持續時間：選取下列其中一項：
 - 全部（預設值）
 - 12 小時
 - 24 小時
 - 2 天
 - 3 天
 - 4 天
 - 5 天

清單更新時間會隨著訊息數目和過濾準則而不同。過濾器範圍越窄，清單的更新速度越快。請注意，搜尋所有嚴重性和所有時段的字串，都需要費時處理。

每則訊息都會顯示三個欄位：

- 時間戳記
- 使用者
- 訊息

在**管理 > 系統**中，提供了一些用來控制日誌大小的設定：

- AutoClean 錯誤日誌天數
- AutoClean 參考日誌天數
- AutoClean 警告日誌天數

如需相關資訊，請參閱第 212 頁的『系統配置設定』。

翻譯的訊息

動態翻譯非英文的訊息。

當您以非 en_US 的語言來使用 Build Forge 時，會動態翻譯訊息。請選取**管理 > 訊息**來顯示訊息的清單。訊息以批次方式，每 15 秒翻譯 250 則。

如果尚未翻譯某訊息，「訊息」欄位會顯示「訊息尚未翻譯」。

重要：如果新增的使用者其語言不同於現行設定，則訊息會使用該語言。現有的訊息不會以之前的語言或是新使用者的語言來顯示。

訂閱工作狀態的 RSS 資料資訊來源

現在您可以使用 RSS 資料資訊來源，來追蹤和過濾個別工作的狀態。工作的 Build Forge RSS 資料資訊來源所顯示的資訊，與 Build Forge「管理主控台」中的伺服器狀態相同。

如果要訂閱工作狀態的 RSS 資料資訊來源，請執行下列動作：

1. 在 Build Forge「管理主控台」中，選取**工作**。

Web 瀏覽器會偵測 RSS 資訊來源，並且在瀏覽器位址列顯示 RSS 圖示。

2. 在 RSS 聚集器工具中，載入 Build Forge RSS 資料資訊來源。

例如，複製 URL，將它新增至 RSS 資料資訊來源清單中，或者拖放 RSS 圖示，將 URL 新增至 RSS 資料資訊來源清單中。

3. 訂閱 RSS 資料資訊來源來儲存 URL，並且只要一有更新，就接獲通知。

註：

- 如需載入 URL 和訂閱 RSS 資料資訊來源的相關資訊，請參閱您 RSS 聚集器工具的相關文件。
- 如果要透過非英文的 RSS 資料資訊來源，來檢視 Build Forge 工作狀態、系統訊息或伺服器狀態，您的 RSS 聚集器工具必須支援 UTF-8 多位元組字元編碼。
- RSS 資訊來源會使用時間戳記來報告活動。時間戳記是以針對您的 Build Forge 使用者帳戶所設定的時區為基礎。如果您的瀏覽器使用另一個時區，您應該將它設為與您的 Build Forge 帳戶相同的時區，以取得正確的時間戳記。

過濾系統訊息的 RSS 資料資訊來源

您可以過濾 RSS 資料資訊來源所顯示的訊息。方法是使用管理主控台的**管理 > 訊息**頁中的過濾器內容。

例如，將「嚴重性」內容變更為「審核」，或者將「持續時間」內容變更為「12 小時」。只要一變更過濾器內容，「訊息」頁上 RSS 圖示的 URL 就會自動更新。

如果要將已更新的 URL 載入至 RSS 聚集器工具，請用滑鼠右鍵按一下「訊息」頁面上的 RSS 圖示來複製鏈結，或者將已更新的 RSS 圖示，拖放到工具中的 RSS 資料資訊來源清單。

註：

- 有關載入 URL 和訂閱 RSS 資料資訊來源的詳細資料，請參閱 RSS 聚集器工具的相關文件。
- 如果要透過非英文的 RSS 資料資訊來源，來檢視 Build Forge 系統訊息或伺服器狀態，您的 RSS 聚集器工具必須支援 UTF-8 多位元組字元編碼。

- RSS 資訊來源會使用時間戳記來報告活動。時間戳記是以針對您的 Build Forge 使用者帳戶所設定的時區為基礎。如果您的瀏覽器使用另一個時區，您應該將它設為與您的 Build Forge 帳戶相同的時區，以取得正確的時間戳記。

「安全」畫面

「安全」畫面可讓您啓用安全服務：

- **SSL**：在這個畫面中啓用 SSL，只是啓用整個系統之 SSL 安全特性中的一個環節。在開啓前後，還需要執行其他工作。
- **密碼加密**：在這個畫面中啓用密碼加密，只是啓用整個系統中密碼加密的一個環節。

若要檢視「安全」畫面，請選取**管理 > 安全**。

重要：在「安全」畫面中只是完成其中的部分設定。如需相關資訊，請參閱第 88 頁的『安全特性』。

啓用 SSL

安裝中的必備條件：在安裝期間，您會指定兩件供 SSL 配置使用的事項：

- **SSL 埠**：在「Web 與應用程式伺服器」畫面中指定。該埠必須和您在下面所選之配置中指定的埠相符。安裝期間以及配置中的預設埠為 8443。在登入期間，Apache Tomcat 上的鑑別 Servlet 會使用此埠來編碼或加密使用者登入認證。
- **憑證**：在「Web 與應用程式伺服器」畫面中指定。您已提供自己的憑證，或容許安裝程式為您建立自簽憑證。憑證儲存在預設金鑰儲存庫中。金鑰儲存庫位置定義在指名的 SSL 配置中。
- 將 **SSL 已啓用**設為「是」。會顯示其他內容：
 - **LDAP 出埠**：指定透過 LDAP 進行出埠通訊時所用的配置。預設值為「預設 JSSE 出埠 SSL」。
 - **引擎至代理程式的預設出埠**：指定從引擎元件和代理程式進行通訊時所用的配置。預設值為「預設 OpenSSL 出埠 SSL」。
 - **服務層入埠**：指定「服務層」元件在接受來自 Web 介面元件與引擎元件的通訊時，所要使用的配置。預設值為「預設 JSSE 入埠 SSL」。
 - **服務層出埠 (JSSE)**：指定引擎元件和 Web 介面元件的「服務層」元件在和資料庫通訊時，要使用的 JSSE 配置。預設值為「預設 JSSE 出埠 SSL」。
 - **服務層出埠 (OpenSSL)**：指定引擎元件和 Web 介面元件的「服務層」元件在和資料庫通訊時，要使用的 OpenSSL 配置。預設值為「預設 OpenSSL 出埠 SSL」。
- 按一下**儲存**。

- 按一下**更新主要 BFClient.conf**。這個步驟是使用這些內容設定來編輯 BFClient.conf 檔。
- 重新啟動 Build Forge。安全通訊必須等到使用這些設定來啟動系統後才會生效。

您選取的配置是在 **SSL** 畫面中定義的。

在這個畫面中啟用 **SSL** 之後的需求如下所示：

- 憑證公佈：憑證必須安裝在代理主機、資料庫主機以及任何正在執行的其他 Build Forge 安裝作業（冗餘配置）上的金鑰儲存庫中。
- 啟用代理程式 **SSL**：如果您有意在引擎元件和代理程式之間採行 **SSL** 通訊，必須將每一個代理程式配置成使用 **SSL**。
- 啟用 **API** 用戶端：所有的 **API** 用戶端皆必須配置 **SSL**，才能和服務層元件通訊。

啟用密碼加密

必備條件：

- **SSL 埠**：在「Web 與應用程式伺服器」畫面中指定。該埠必須和您在下面所選之配置中指定的埠相符。安裝期間以及配置中的預設埠為 8443。在登入期間，Apache Tomcat 上的鑑別 Servlet 會使用此埠來編碼或加密使用者登入認證。
- **憑證**：在「Web 與應用程式伺服器」畫面中指定。您已提供自己的憑證，或容許安裝程式為您建立自簽憑證。憑證儲存在預設金鑰儲存庫中。金鑰儲存庫位置定義在指定的 **SSL** 配置中。
- 將**密碼加密已啟用**設為「是」。
- 按一下**儲存**。
- 按一下**更新主要 BFClient.conf**。這個步驟是使用這些內容設定來編輯 BFClient.conf 檔。
- 重新啟動 Build Forge。密碼加密必須等到使用這些設定來啟動系統後才會生效。
- 一旦啟用，在主控台中輸入的任何新密碼都會經過加密，包含「伺服器授權」密碼和使用者在主控台中建立的使用者密碼。

其他需求：

啟用加密後，必須執行下列動作：

- 在所有代理程式上啟用加密。匯出金鑰，並將它用來更新每一個代理程式配置中的伺服器授權密碼。必須在 BFAgent.conf 中手動更新密碼。
- 針對資料庫存取啟用加密密碼。匯出金鑰，並將它用來更新 Build Forge 用於登入資料庫的資料庫密碼。必須在 buildforge.conf 中手動更新密碼。

「金鑰儲存庫」畫面

「金鑰儲存庫」畫面含有個別金鑰儲存庫的配置。當您在**管理 > 安全 > SSL** 畫面中編輯 **SSL** 配置時，您可以選取這些個別的配置，使其成為 **SSL** 配置的一部分。

您可以建立配置，或使用所提供的配置：

- 預設 **JSSE 金鑰儲存庫**
- 預設 **JSSE 信任儲存庫**
- 預設 **OpenSSL CA 儲存庫（憑證管理中心）**

- 預設 OpenSSL 憑證儲存庫（用於憑證）
- 預設 OpenSSL 金鑰儲存庫

每一個金鑰儲存庫各有下列內容：

名稱

存取 定義哪些使用者可以編輯或刪除此金鑰儲存庫的存取群組。

位置 金鑰儲存庫檔的位置。預設金鑰儲存庫全都會使用 Build Forge 金鑰儲存庫的預設位置：`<bfinstall>/keystore`。如果您使用 WebSphere Application Server 作為應用程式伺服器，而不使用所提供的 Tomcat 應用程式伺服器，請指定絕對路徑。

金鑰儲存庫類型

金鑰儲存庫必須是下列其中一種類型：

- JKS
- PKCS12
- JCEKS
- PEM

密碼 指定在存取金鑰儲存庫時必須使用的密碼。

驗證 在這裡重新指定密碼，以便驗證。

金鑰儲存庫和 WebSphere Application Server

如果您使用的是 WebSphere Application Server 而不是提供的 Tomcat 應用程式伺服器，則配置金鑰儲存庫時還會有其他需求：

- 「位置」欄位：您必須提供絕對路徑，而不是相對路徑。
- 多重服務元件：如果您安裝多重 Build Forge 服務元件，它們是安裝在不同的主機上。您分別配置每個服務元件的安全。對於每個服務元件，「位置」欄位所指定的金鑰儲存庫路徑必須相同。

SSL 畫面

SSL 畫面含有 SSL 的個別配置。當您在**管理 > 安全**畫面中，將 **SSL 已啓用**設為是時，可以選取這些個別的配置，來啓用 SSL。

您可以建立自己的配置，或使用所提供的配置：

- 預設 JSSE 入埠 SSL
- 預設 JSSE 出埠 SSL
- 預設 OpenSSL 入埠 SSL
- 預設 OpenSSL 出埠 SSL

每一項配置各有下列內容：

名稱 這項配置的名稱。

存取 定義哪些使用者可以編輯或刪除此金鑰儲存庫的存取群組。

類型 選取 JSSE 或 OpenSSL。

用戶端鑑別

請選取下列其中一項：

- 絕不
- 支援
- 必要

伺服器憑證別名

輸入伺服器憑證的別名。

用戶端憑證別名

輸入用戶端憑證的別名。

金鑰儲存庫配置

選取一項金鑰儲存庫配置。這些是在金鑰儲存庫畫面中配置。

信任儲存庫配置

選取一項金鑰儲存庫配置。這些是在金鑰儲存庫畫面中配置。

信號交換通訊協定

請選取下列其中一項：

- SSLv2
- SSLv3
- SSL
- TLSv1
- TLS
- SSL_TLS - 對於 Default OpenSSL Inbound SSL 與 Default OpenSSL Outbound SSL，不適用
- TLSv1_1
- TLSv1_2

註：

下列資訊適用於信號交換通訊協定的 TLSv1_1 及 TLSv1_2 內容：

- 若要啓用 TLSv1.1 或 TLSv1.2，請選取所有配置（Default JSSE Inbound SSL、Default JSSE Outbound SSL、Default OpenSSL Inbound SSL、Default OpenSSL Outbound SSL）。
- 若啓用了 LDAP，並且 LDAP 出埠使用 Default JSSE Outbound SSL，請首先複製 Default JSSE Outbound SSL，然後再將協定變更爲 TLSv1_1 或 TLSv1_2。LDAP 隨後會使用複製的配置。
- TLSv1_1 與 TLSv1_2 僅在 BFagent 與 Java 引擎、Java API 與服務層、Perl API 與服務層之間受支援。因此，在已啓用 TLSv1_1 或 TLSv1_2 時，若伺服器上已啓用 TLSv1.1 或 TLSv1.2，Perl 引擎將無法連接到伺服器。
- 僅透過 JDK 1.7 支援 TLSv1_1 與 TLSv1_2。因此，在 WebSphere Application Server 上部署 Build Forge WAR 檔時，由於部署使用低於 1.7 的 JDK 版本，因此不支援 TLSv1.1 及 TLSv1.2。

- BFagent 在 Rational® Build Forge® 8.0.0.1 中新增了 TLSv1.1 及 TLSv1.2 支援。因此，在 SSL 頁面上選取 TLSv1_1 或 TLSv1_2 並且伺服器的 **SSL 已啓用** 內容設為是時，請確保代理程式的版本等於或高於 8.0.0.1 版。

密碼組合群組

下列其中一項：「高」、「中」、「低」或「全部」。愈高階的密碼愈安全，但必須承擔效能變慢的後果。

密碼置換清單

SSO 畫面

SSO 畫面含有「單一登入 (SSO)」的配置。您可以建立新配置，或使用所提供的配置：

- 表單 SSO 攔截程式
- SPNEGO SSO 攔截程式

啓用單一登入服務，還需要進行其他設定。單從這個畫面無法全面完成它。請參閱第 88 頁的『實作單一登入』。

每一項配置各有下列內容：

名稱 SSO 配置的名稱。

作用中 選取「是」，則會讓這個攔截程式變成作用中。

存取 定義哪些使用者可以編輯或刪除此配置的存取群組。

Java 類別

實作此 SSO 配置的 Java 類別。系統提供下列兩種：

- 表單攔截程式：

`com.buildforge.services.server.sso.form.FORMSSOInterceptor`

- SPNEGO 攔截程式：

`com.buildforge.services.server.sso.spnego.SPNEGOSSOInterceptor`

管理授權

當您建立使用者時，系統會將使用者的相關資料儲存在資料庫中。當使用者登入系統之後，系統會指派授權給該使用者。

授權使用者

會將授權指派給使用者。使用者可隨時登入。登入會使用一個授權。

浮動使用者

浮動授權使用者只能在有可用的授權時登入。如果所有授權都在使用中，這些使用者就無法登入。

記號 (Token) 授權

客戶若擁有多項 Rational 產品的授權，可選擇使用記號授權。這類使用者會購買多個可用於這些產品的記號授權。可使用特定特性與登入來使用記號授權。

記號授權支援可供「浮動伺服器」、「浮動使用者」、「配接器」和 QuickReport 授權使用。但它不支援固定授權（授權的伺服器或授權的使用者）。如果「伺服器」、「配接器」和 QuickReport 授權已被移出，則在 Build Forge 關閉之前，將不會加以釋出。

記號授權可支援 Rational Build Forge 7.1.3.4 版以及更新版本。若要支援記號授權，還必須有 Rational Common Licensing 8.1.2 版或更新版本。

如需記號授權的相關資訊，請與 IBM 業務代表聯絡。

如果您建立的使用者多於您有的授權，新的使用者就必須等到您再額外購買授權，或者使用 root 使用者帳戶，移除部分使用者的授權（或者從系統刪除部分使用者）之後才能登入。

使用者只能登入一次。如果您登入一部電腦之後，再以同樣的名稱從另一部電腦登入，原始階段作業就會無效。

註：如果您購買的系統早於 3.8 版，則授權綱目可能會與此處所說明的不同。如果您有任何授權方面的問題，請聯絡客戶支援中心。

輸入新授權碼

如果要變更授權碼，請選取**管理 > 系統**，然後在設定清單中找出「授權伺服器」設定。（在「過濾器」方框中鍵入 "License"，即可快速顯示這項設定，而不需要翻看整個清單）。

按一下清單中的「授權伺服器」項目，系統便會在內容畫面的底端顯示一個標籤，內含可供編輯的**授權伺服器**欄位。編輯欄位之後，請按一下**儲存**按鈕。

記號授權日誌

Rational Common Licensing 配置檔 RCL_Log_Config.txt 位於 Build Forge 安裝目錄。依預設，安裝目錄是在下列位置：

- Windows 系統
C:\Program Files\IBM\Build Forge
- UNIX 及 Linux 系統
opt/buildforge/platform

依預設，系統會在安裝目錄的 temp/rc1 子目錄中建立 Rational Common Licensing 日誌檔。

若要將配置檔和日誌檔存放在相同的位置，請遵循下列步驟：

1. 停止 Build Forge。
2. 在 RCL_Log_Config.txt 檔案中，將 Location = ./temp/rc1 這一行變更為 Location = ./token_libs/rc1。
3. 重新啟動 Build Forge。現在，所有記號日誌將於 <bfinstall>/token_libs/rc1 中產生。

管理 Perl 引擎

本節說明如何暫停、啟動及停止 Perl 引擎。

暫停 Perl 引擎

若要暫時延遲處理新的工作，請暫停 Perl 引擎。

註：如果您要直接在資料庫上操作（例如進行備份或還原），則必須先停止所有執行中的 Build Forge 服務和程序。在此情況下，暫停引擎還不夠，還可能導致不一致及（或）毀損。

1. 若要暫停引擎，請按一下**管理 > 系統**，然後使用**過濾器**方框找出**暫停 Build Forge 引擎**內容。系統會顯示一個畫面，供您編輯**暫停 Build Forge 引擎**設定。
2. 將內容變更為 **Y**，然後按一下**儲存配置**。引擎會暫停，但目前正在執行的所有專案都會繼續進行。您必須等待所有作用中的工作完成，以確保沒有任何資料正在寫入資料庫。

若要重新啟動引擎，請將內容值變更回 **N**，然後再按一次**儲存配置**。

啟動和停止 Perl 引擎

下列章節說明如何在 Windows[®]、Linux[®] 與 Solaris 上啟動和停止 Perl 引擎。

在 Windows 上啟動和停止 Perl 引擎

使用 Rational[®] Build Forge[®] 使用者介面來啟動和停止 Perl 引擎。啟動和停止 Perl 引擎一律是手動執行。

啟動 Perl 引擎

1. 在 Build Forge[®] 使用者介面上，啟用 Perl 引擎：
 - a. 按一下**管理 > 系統**。
 - b. 在**預設為 Java 主要工作控制器**欄位中，選取**否**。

註：如果 Java 引擎工作在執行中，則您無法將**預設為 Java 主要工作控制器**欄位的值變更為**否**。相反地，Perl 引擎工作也是如此。

2. 從指令行，移至 Build Forge[®] 安裝路徑並輸入這個指令：`bfengine -p`，以啟動 Perl 引擎。

停止 Perl 引擎

若要停止 Perl 引擎，請手動關閉正在執行引擎的視窗。

在 Linux 及 Solaris 上啟動和停止 Perl 引擎

使用 Rational[®] Build Forge[®] 使用者介面來啟動和停止 Perl 引擎。您一律必須手動啟動和停止 Perl 引擎。

啟動 Perl 引擎

1. 在 Build Forge[®] 使用者介面上，啟用 Perl 引擎：
 - a. 按一下**管理 > 系統**。
 - b. 在**預設為 Java 主要工作控制器**欄位中，選取**否**。

註：如果 Java 引擎工作在執行中，則您無法將**預設為 Java 主要工作控制器**欄位的值變更為**否**。相反地，Perl 引擎工作也是如此。

2. 從 Shell 指令行，移至 Build Forge[®] 安裝路徑並輸入這個指令：`rc/perl-engine start`，以啟動 Perl 引擎。

停止 Perl 引擎

1. 從 Shell 指令行，移至 Build Forge[®] 安裝路徑並輸入這個指令：`rc/perl-engine stop`，以停止 Perl 引擎。
2. 在 Build Forge[®] 使用者介面上：
 - a. 按一下**管理 > 系統**。
 - b. 在預設為 **Java 主要工作控制器** 欄位中，選取是。

管理資料庫

這個主題說明設定「管理主控台」資料庫時的重要問題，尤其是當您想要變更預設配置時。

刪除資料庫日誌檔

定期刪除資料庫日誌檔。

系統會將資料庫除錯資訊儲存在「管理主控台」安裝目錄中的 `db.log` 檔中。您應該每月檢查一次此檔案的大小，並在需要釋放主控台電腦上的空間時刪除此檔案。

錯誤訊息

這個主題說明您在使用「管理主控台」時可能會遇到的錯誤訊息。

沒有作用中的步驟

當這個訊息出現在排程項目的「下一個執行」直欄中時，相關聯專案中的所有步驟都會失效。如果要顯示專案的步驟清單，並啟用其中的一些步驟，請按一下每一個停用步驟旁的紅色圓圈。

授權碼無效或者Build Forge 授權碼毀損或遺漏。

對您已安裝的產品版本而言，授權碼過期或無效。請輸入新的授權碼。請參閱第 233 頁的『輸入新授權碼』。

需要資料庫授權

LicMgr: 5140: A database license is required.

如果主控台輸出中出現上述訊息（當您在前景執行主控台時，或者如果您是在 Linux[®] 或 UNIX[®] 系統上檢視主控台日誌，就可以檢視該輸出），表示您在沒有 Enterprise 授權的權限下，嘗試使用進階資料庫特性。例如，您可能嘗試使用含有非 DB2[®] 資料庫的系統。如果您要升級授權，請聯絡支援中心。

第 17 章 伺服器

這個主題說明如何在「管理主控台」中設定和管理伺服器資源。

關於伺服器資源

在主控台中，伺服器資源代表一個可讓您執行專案或步驟的主機。

- 伺服器含有資訊清單 (*manifest*)。資訊清單是一份伺服器內容清單。當收集器執行時會填寫資訊清單。如果沒有指派收集器給伺服器，則會在伺服器資訊清單中自動移入一些內容。
- 資訊清單是透過收集器來移入資料。收集器會指派給伺服器。收集器除了設定內容值外，也會從代理程式收集伺服器內容的值。
- 選取元會從資訊清單讀取伺服器內容。專案可以使用選取元來決定要執行步驟的伺服器。

身為設定系統的管理者，必須先建立伺服器。建立要指派給伺服器的收集器。執行收集器來移入資料至伺服器資訊清單。完成這些作業之後，建置工程師會建立專案，並含有可決定執行專案步驟位置的選取元。

每一項伺服器資源各指向一個已安裝代理程式的主機。新增伺服器資源時，您可說明「管理主控台」存取和使用特定主機的方式。

建立伺服器之前，請確定相依於伺服器的資料物件已存在。請將這些項目指派給伺服器：

項目	必要或選用	說明
伺服器鑑別	必要	指定要用於伺服器的登入名稱與密碼。
收集器	選用	除了預設內容之外，還定義系統從伺服器收集的內容。
環境	選用	指定當專案在伺服器上執行時，所指派的環境變數。

註：您可以在「管理主控台」內，針對單一實體伺服器建立多個伺服器物件。這些伺服器物件稱為邏輯伺服器。通常會使用邏輯伺服器，讓專案可以存取具有不同內容的相同硬體。例如，兩個邏輯伺服器可能使用不同的路徑或不同的環境：

- 兩個採用不同路徑的邏輯伺服器，會在相同的伺服器上建立個別的工作目錄。您可以區別出由不同伺服器執行的工作。所有輸出會儲存在不同目錄中。
- 兩個採用不同環境的邏輯伺服器，會使用不同的起始環境值來執行步驟。

關於「伺服器」畫面

您可以使用「伺服器」畫面來指定要執行專案的伺服器。

若要在「管理主控台」上管理伺服器，請按一下左側功能表中的**伺服器**。

「詳細資料」標籤

「詳細資料」標籤提供下列欄位：

主機 正在執行 Rational Build Forge 代理程式的主機名稱。

收集器 要與伺服器搭配使用的收集器。

環境 要與伺服器搭配使用的環境。

伺服器提供者

協助與端點伺服器進行通訊的提供者。預設伺服器提供者為「代理程式伺服器提供者」。它提供與 Build Forge 代理程式端點伺服器的通訊。

鑑別 伺服器鑑別登入 及密碼。

密碼加密

決定代理程式密碼是否加密。

檔案 要與 **.get** 及 **.put** 指令搭配使用的檔案傳送類型

存取 伺服器上的使用者存取群組。

已啓用 SSL

指出您是否已配置 Rational Build Forge 系統來使用 SSL。

工作數上限

應該同步執行的工作數上限。

「資訊清單」標籤

「資訊清單」標籤顯示各種伺服器內容。它提供預設內容集。您可以使用收集器來包含其他內容。

「測試結果」標籤

「測試結果」標籤指出前次測試「管理主控台」與伺服器之間的連線時的狀態。

「附註」標籤

「附註」標籤提供方便的位置來儲存及共用伺服器的相關註解。

建立伺服器

伺服器資源代表可讓您用來執行專案和步驟的主機。

如果要建立伺服器，請執行下列動作：

1. 選取左功能表中的**伺服器**。系統會在主要內容畫面底端顯示「新建伺服器」畫面。如果您選取了現存的伺服器，請按一下**新增伺服器**來消除畫面，使您可以新增伺服器。
2. 提供伺服器詳細資料：
 - **名稱**：輸入伺服器的名稱。這個名稱是伺服器的 BF_NAME 內容。您可以在選取元中參照這個名稱，以依照名稱來指定伺服器。
 - **路徑**：指定伺服器在建立專案和工作目錄時要使用的目錄，例如 c:/buildforgeprojects。系統在建立建置目錄時，即使用這個路徑值來作為起始點。

提示：系統並未建立伺服器路徑。在建置試圖存取伺服器之前，路徑就必須存在。如果路徑不存在，建置就會失敗。

- **主機：**輸入正在執行代理程式之實體電腦的主機名稱。如果您將「管理主控台」電腦定義成伺服器，請使用 *localhost* 值。代理程式也必須安裝在「管理主控台」上。

註：您可以在主機名稱中包含埠號；例如 *host_name:port_number*。如果您在主機名稱中指定了埠號，該埠號會置換「預設代理程式埠」系統設定所定義的埠號。按一下**管理 > 系統 > 預設代理程式埠**。

註：主機名稱之前請勿加上通訊協定。例如，請勿使用 *http://*。

- **鑑別：**請選取要搭配這部伺服器使用的伺服器鑑別。
- **存取：**選取可以使用這部伺服器的使用者存取群組。
- **收集器：**選取要搭配此伺服器使用的收集器。
- **環境：**選取當使用這部伺服器來執行專案時，所要套用的一組環境變數。這些變數會比其他所有的變數先行套用。它們會設定該伺服器特有的參數。
- **伺服器提供者：**選取提供者來協助與端點伺服器進行通訊。預設伺服器提供者為「代理程式伺服器提供者」。它提供與 Build Forge 代理程式端點伺服器的通訊。
- **檔案：**定義這部伺服器的檔案傳送類型（可搭配 **.get** 和 **.put** 指令使用）。您可以選擇不傳送（None）、檔案讀取（.get）、檔案寫入（.put），或檔案讀寫（.get 和 .put）。
- **工作數上限：**輸入可同步執行的工作數上限。預設值為 3。
- **密碼加密配置：**如果您要加密代理程式密碼，請選取**已啓用**。預設值為**已停用**。
- **SSL 已啓用：**如果您已將 Build Forge[®] 系統配置成使用 SSL，並且希望此伺服器資源透過 SSL 與代理程式進行通訊，請選取**是**。預設值為**否**。

註：如果您選取**是**，但 Build Forge[®] 未正確配置以使用 SSL，則這項伺服器資源無法與代理程式進行通訊。

3. 按一下**儲存**。新的伺服器即會顯示在內容畫面頂端的伺服器清單中。若要驗證伺服器的配置是否正確，請在清單中選取伺服器，然後按一下**測試連線**。系統如果無法與伺服器通訊，則會報告錯誤。

測試伺服器

您可以在伺服器上執行一組診斷測試。

若要執行診斷測試，請執行下列動作：

1. 按一下**伺服器- > server_name**。
2. 按一下**測試連線**。

即會執行伺服器連線測試，並且會更新伺服器資訊清單。**測試結果**標籤上會出現進度列，直到重新整理完成為止。

您可以在**測試結果**標籤上檢視結果，方法是按一下**檢視測試詳細資料**以取得其他資訊。

如果伺服器測試失敗，請嘗試下列其中一個動作：

- 驗證在伺服器鑑別中的使用者名稱和密碼正確。
- 驗證您使用的主機名稱無誤。
- 在伺服器上重新安裝代理程式，或者驗證已安裝它。

變更伺服器

您可以視需要編輯伺服器內容。

重要：如果存在下列狀況，您必須重新啟動主控台：

- 主機參數使用代理主機的主機名稱
- 代理主機的 IP 位址發生變更

來自 DNS 的 IP 位址將會快取在管理 Build Forge 服務元件的 JVM 中。您必須重新啟動才能清除快取，並讓服務元件在新 IP 位址上尋找代理程式。

如果您在主機參數中使用 IP 位址，則無需重新啟動。當代理主機的 IP 位址發生變更時，您可以將主機參數變更為新位址，然後儲存變更。

啓用及停用伺服器

您可以暫時停用伺服器。一旦停用，工作就無法在其上執行。

若要停用或啓用伺服器，請執行下列動作：

1. 按一下**伺服器**，顯示伺服器清單。
2. 按一下要修改的伺服器資源名稱。系統會顯示該伺服器的詳細資料。
3. 在主畫面底端的「詳細資料」標籤中，選取**停用**。
4. 按一下**儲存伺服器**。

限制伺服器上的並行工作

使用**最大工作數目**伺服器內容來指定可讓特定伺服器同步執行的工作數目上限。

關於這項作業

系統會限制它在任一伺服器上試圖執行的程序數量。**最大工作數目**內容會指定處理程序的數量限制。這項內容的預設值為 3。

註：其他程式可在伺服器上執行。系統限制僅套用到 Build Forge 工作上。

註：內建變數 BF_RESERVE 及 BF_EXCLUSIVE 能控制預約保留伺服器的一個或所有工作屬性的作業。如需相關資訊，請參閱第 255 頁的『預設內容』。

如果要為伺服器提供一個非預設的值，請執行下列動作：

程序

1. 按一下**伺服器**。
2. 選取現有的伺服器或新建一個。
3. 將**工作數上限**內容設為所要的值。
4. 按一下**儲存**。

重設工作計數

將伺服器上的工作計數重設為零

在「伺服器」頁面上，使用**重設工作計數**，將所選伺服器的工作計數 (BF_JOBS) 重設為零。BF_JOBS 是目前在伺服器上執行的步驟或工作數；如果此數目的設定太低，工作可能會在工作競用時失敗。

這個選項可讓您在工作完成、失敗或取消時，重設 BF_JOBS（如果沒有正確地重設它的話）。

例如，取消多個工作時，有時候會導致無法重設 BF_JOBS。如果 BF_JOBS 沒有重設，則可能達到「工作數上限」設定的限制，而導致步驟或工作不會執行。

1. 選取**伺服器**，顯示伺服器清單。
2. 選取一部伺服器。
3. 按一下**重設工作計數**。

將所有伺服器上的工作計數重設為零

使用**重設伺服器的工作計數**頁面，可同步將所有伺服器的工作計數 (BF_JOBS) 一併設為零。BF_JOBS 內容是正在伺服器上執行的步驟或工作數；如果此數目的設定太低，工作可能會在工作競用時失敗。

在您重設所有伺服器的 BF_JOBS 值之後，「重設伺服器的工作計數」值就會回復為「否」（這是預設值）。

例如，在您取消數個工作後，建置系統有時候會無法重設 BF_JOBS 值。如果 BF_JOBS 沒有重設，其值可能達到預設「工作數上限」系統設定的限制，而導致步驟或工作不會執行。

1. 按一下**管理 > 系統**，以顯示系統配置設定的清單。
2. 尋找重設伺服器工作計數。
3. 按一下**重設伺服器工作計數**。
4. 按一下「詳細資料」標籤，然後按一下**是**，作為其值。
5. 按一下**儲存**。

伺服器狀態的 RSS 資料資訊來源

管理主控台會執行伺服器狀態檢查，來驗證伺服器可以通過功能測試，並且驗證代理程式可以登入。「測試結果」標籤會顯示狀態檢查的結果。只要一建立或編輯伺服器，「管理主控台」就會自動檢查伺服器狀態。您也可以隨時起始伺服器狀態檢查，例如，在執行專案之前。

伺服器狀態的 Build Forge RSS 資訊來源所顯示的資訊，與 Build Forge 管理主控台的「測試結果」標籤相同。

如果要訂閱伺服器狀態的 RSS 資料資訊來源，請執行下列動作：

1. 在 Build Forge「管理主控台」中，選取**伺服器**。

Web 瀏覽器會偵測 RSS 資訊來源，並且在瀏覽器位址列顯示 RSS 圖示。

2. 在 RSS 聚集器工具中，載入 Build Forge RSS 資料資訊來源。

例如，複製 URL，將它新增至 RSS 資料資訊來源清單中，或者拖放 RSS 圖示，將 URL 新增至 RSS 資料資訊來源清單中。

3. 訂閱 RSS 資料資訊來源來儲存 URL，並且只要一有更新，就接獲通知。

註：

- 如需載入 URL 和訂閱 RSS 資料資訊來源的相關資訊，請參閱您 RSS 聚集器工具的相關文件。
- 如果要透過非英文的 RSS 資料資訊來源，來檢視 Build Forge 系統訊息或伺服器狀態，您的 RSS 聚集器工具必須支援 UTF-8 多位元組字元編碼。
- RSS 資訊來源會使用時間戳記來報告活動。時間戳記是以針對您的 Build Forge 使用者帳戶所設定的時區為基礎。如果您的瀏覽器使用另一個時區，您應該將它設為與您的 Build Forge 帳戶相同的時區，以取得正確的時間戳記。

伺服器鑑別

這個主題說明如何針對伺服器中執行的專案授與存取權。

關於伺服器鑑別

使用伺服器鑑別來建立登入認證與伺服器的關聯。

伺服器鑑別會將登入名稱與密碼儲存成已命名的單一物件，您可以將該物件與一或多部伺服器產生關聯。請使用「伺服器鑑別」頁面來建立及編輯伺服器鑑別。

您可以藉由管理一組伺服器鑑別，讓多部伺服器使用相同的認證，以及全面更新認證。

關於伺服器鑑別畫面

您可以使用伺服器鑑別畫面來建立及管理伺服器鑑別。

若要檢視此畫面，請選取**伺服器 > 伺服器鑑別**。

使用者介面配置 主控台 報告 登出: Root User

伺服器鑑別 新增伺服器鑑別 說明 ?

過濾條件 顯示 1 - 1 (總共 1) 顯示全部

名稱	使用者名稱
Admin	admin

(新建伺服器鑑別) 儲存伺服器鑑別 複製伺服器鑑別 刪除伺服器鑑別

詳細資料

名稱: 存取: 安全人員

登入: 密碼: 驗證:

這個畫面提供下列欄位：

名稱	用於鑑別本身的名稱
存取	可使用此鑑別的使用者存取群組
登入	登入時要使用的帳戶名稱
密碼	與登入相關聯的密碼
驗證	同樣的密碼

建立伺服器鑑別

使用伺服器鑑別來儲存伺服器集的登入資訊。

關於這項作業

各指派一項伺服器鑑別給每一部伺服器，讓「管理主控台」能以適當的專用權登入伺服器。伺服器鑑別會將登入資訊從伺服器記錄中分離出來，以讓您可以將相同的登入資訊套用至多部伺服器。

如果要建立伺服器鑑別，請執行下列動作：

程序

1. 在 Build Forge 的左畫面中按一下**伺服器 > 伺服器鑑別**。

建置系統會在主要內容畫面的頂端顯示現有伺服器鑑別的清單，並在底端顯示空白的「伺服器鑑別詳細資料」畫面。

提示：當您選取伺服器鑑別時，系統會將所選伺服器的鑑別資訊，移入「伺服器鑑別詳細資料」畫面中。若要清除畫面使您可以建立新的鑑別，請按一下**新增伺服器鑑別**。

2. 在**名稱**中，輸入鑑別名稱，這是一個用來在系統中識別該伺服器鑑別的邏輯名稱。
3. 在**登入**中，指定伺服器登入名稱。

註：如果登入名稱來自網域使用者，請在這個欄位中併入網域。例如輸入：
MYDOMAIN/joeuser。

4. 在**密碼**中，輸入密碼。
5. 在**驗證**中，重新輸入密碼。
6. 按一下**儲存伺服器鑑別**。系統會以您選取的名稱來儲存新的伺服器鑑別。

結果

建置系統會以您選取的名稱來儲存新的伺服器鑑別。

置換伺服器鑑別

您可以利用特殊的環境變數，強制伺服器改用您的「管理主控台」登入認證，而不使用指派給伺服器的伺服器鑑別。如果要置換指派的鑑別，請在您專案或步驟使用的環境中，新增一個名為 `_USE_BFCREDS` 的變數，並將值設為 1。如果您在專案環境中新增該變數，建置系統便會在專案的每一個步驟上使用置換後的值。

當建置系統試圖在含有 `_USE_BFCREDS=1` 的環境中執行步驟時，系統會使用專案啟動者的主控台登入認證，來執行步驟的指令。

註：如果您使用 LDAP/Active Directory 鑑別，則必須將**在本端儲存使用者鑑別**系統設定設為「是」（預設值），`_USE_BFCREDS` 變數才能運作。若設為「是」，則系統會快取加密格式的使用者鑑別資訊，然後存取要使用於 `_USE_BFCREDS` 的使用者鑑別資訊。否則，系統就不會儲存 LDAP 資訊，也無法使用它。

提示：在 Windows 上，請考慮同時設定變數 `_USE_BFCREDS_DOMAIN`。將此變數設為值 1 會包含使用者網域。

容許使用受限的伺服器鑑別

利用「執行無法存取的伺服器鑑別」許可權，可讓使用者使用他們無權存取的伺服器鑑別，在伺服器上執行步驟。

但先決條件是使用者必須已具備或獲授與對伺服器的存取權。（若要授與對伺服器的存取權，請按一下**伺服器 > 存取**。）

您將使用伺服器鑑別來登入並存取伺服器。伺服器鑑別會和存取群組相關聯（**伺服器 > 伺服器鑑別 > 存取**）。

您可以為伺服器建立下列伺服器鑑別：

- `dev/dev` 伺服器鑑別。請讓這項伺服器鑑別和伺服器的開發人員存取群組產生關聯。
- `qa/qa` 伺服器鑑別。請讓這項伺服器鑑別與 QA 存取群組產生關聯。
- `prod/prod` 伺服器鑑別。請讓這項伺服器鑑別與「建置」存取群組產生關聯。

如果要允許只對 `qa/qa` 伺服器鑑別有存取權的使用者，以 `prod/prod` 伺服器鑑別的身分來執行步驟，請將「執行無法存取的伺服器鑑別」許可權新增至 QA 存取群組。

註：如果使用者對伺服器有存取權，但卻無法透過「執行無法存取的伺服器鑑別」許可權取得伺服器鑑別的存取權，則必須設定 `_USE_BFCREDS` 環境變數來置換伺服器鑑別，步驟才能繼續執行。

選取元

這個主題說明選取元。

關於選取元

選取元會選擇要執行專案或步驟的伺服器資源。

選取元含有一份變數清單。您可以為每一個變數指定一個值和一個比較項。例如，您可以指定 `CompilerVersion = 1.1` 內容，只選取具有該內容的伺服器資源。您也可以指定 `CompilerVersion >= 1.1`，以選取具有 1.1、1.3、2 及 2.0 版的伺服器資源。

在專案或步驟執行時，為其指派的選取元將決定它在哪個伺服器資源上執行。

- 靜態選取元使用 `BF_NAME` 變數，依名稱來識別伺服器資源。
- 動態選取元使用選取元中變數所指定的準則，從系統的所有伺服器資源中選擇一項伺服器資源。

關於「選取元」畫面

您可以使用「選取元」畫面來建立及管理選取元，而這些選取元會選擇要執行專案或步驟的伺服器資源。

若要檢視此畫面，請選取**伺服器 > 選取元**。



「詳細資料」標籤： 「詳細資料」標籤提供下列欄位：

名稱 選取元的名稱。

存取 可使用此選取元的使用者存取群組。

Snapshot 標籤： 選取元 Snapshot 是選取元的實例。Snapshot 標籤提供了 Snapshot 的名稱。請使用這個標籤來檢視或變更 Snapshot 名稱以及 Snapshot 的相關註解。

選取元設定實務

選取元說明適用於專案或步驟的伺服器種類。它可以依照收集器所收集並儲存在資訊清單中的名稱或內容，來指定伺服器。它可以指定多個內容，必要和選用的都包含在內。

可用來設定選取元的方法有下列多種：

- 依名稱來選取伺服器資源。在您的伺服器資源之後建立選取元並為其命名。選取元依照其 **BF_NAME** 值（系統中使用的唯一名稱）來指定伺服器資源。當您想要指定用來執行專案或步驟的伺服器資源時，請使用其中一個選取元。
- 依伺服器儲存區來選取伺服器。您可以將伺服器組織成數個具名的儲存區，並為每一個儲存區建立一個收集器。在收集器中將儲存區名稱定義成內容（「設定-值」內容）。然後為每一個儲存區名稱各建立一個選取元。專案或步驟的伺服器資源會根據其現行負載來選取。
- 依伺服器屬性來選取伺服器。您可以根據功能性內容來選擇伺服器，例如，可用的硬碟空間、作業系統或 CPU 數量。如果要實作動態選取，請執行下列動作：
 1. 建立收集器，以便收集和指派適當的內容。
 2. 然後指派這些收集器給適當的伺服器。
 3. 為各代表一個伺服器選項的每一個內容或一組內容，分別建立一個選取元

例如，您可以建立選取元，以根據下列準則來選取：

- 其作業系統包含 "Windows®" 的伺服器資源。

- 具有多個 CPU 的伺服器資源。
- 執行時的負載低於指定值的伺服器資源。
- 依巢狀收集器來選取伺服器資源。在「類型」內容方面使用「併入」，來建立指向另一個收集器的收集器。收集器本身可由一組收集器指標構成。例如，您可能會想要為每一部伺服器建立個別的收集器，讓每一部伺服器都有一些您指定的唯一內容。而您可以使用「併入」類型，來指向通用的收集器。例如，您可以建立一個名為「版本」的收集器，以指定您環境中一些重要資源的版本號碼（如：Perl 和 Java）。

選取元的變數類型

選取元定義在執行時期如何為專案或步驟選擇伺服器資源。

選取元中可供定義的變數類型如下：

- **標準內容**：這個變數類型的指定項目如下：
 - **名稱**：要使用的內容名稱。您可從自動產生的清單中選擇，這份清單是由所有內建的內容以及收集器所定義的任何「設定-值」內容構成。
 - 註**：如果要建置取自「設定-值」內容的選取元，您必須先在一個以上收集器中定義「設定-值」內容。
 - **運算子**：下列其中一個比較運算子：
 - EQ - 等於。值必須完全相符。值可以是數字或字串。
 - NE - 不等於。值不得等於指定的值。值可以是數字或字串。
 - GT - 大於。
 - GE - 大於或等於。
 - LT - 小於。
 - LE - 小於或等於。
 - Contains。
 - **值**：運算子用來比較的值。
 - **必要**：如果選取元必須符合此變數，則為「是」；如果是選用的，則為「否」。
- **併入**：「併入」類型可讓您透過巢狀作業，定義從簡式選取元建置而成的複式選取元。您可以指定下列項目：
 - **選取元**：選擇要併入的選取元。這個選取元指定的所有內容都會併入。您可以藉由併入多個簡式選取元，來建置複式選取元。

選取元變數的比較規則

在系統針對專案或步驟選擇要使用的伺服器資源時，它會比較選取元變數的值與同名的資訊清單內容的值。

除非這兩個值都符合下列的數字準則，否則系統會執行字串比較：

- 如果該值以數字開頭，而且只包含數字和後面至少接一位數的小數點，系統就會執行數值比較。
 - 5、5.5、0.5、5.0 及 5.5.5 皆視為數字。
 - 5.、.5、5..、5..5、5.4.6_05 及 5.6i5 皆視為字串

- 如果該數值包含多個小數點，就會進行子版本數值比較，其中，系統會比較每一個以小數點區隔的欄位。就一般的數值比較來說，5.21 小於 5.3，但就子版本數值比較而言，5.21.0 卻是大於 5.3。

註：在「包含」運算子方面，系統一律執行不區分大小寫的字串比較。

下表顯示的範例說明如何套用比較規則。

內容名稱	資訊清單內容值	運算子	選取元變數值	比較類型	相符？
PerlVersion	v5.8.4	>=	5.2.1	字串	是
PerlVersion	v5.8.4	>=	v.5.2.1	字串	是
PerlVersion	v5.8.4	>=	v5.22.1	字串	是
OS_VERSION	1.15	>=	1.1	數值	是
OS_VERSION	1.10	>=	1.1.0	子版本數值	是
BF_NAME	WinServer1	contains	win	字串	是
BF_NAME	Server123	=	123	字串	否

合格伺服器資源的選取元評量方式

爲了替專案或步驟選擇一項伺服器資源，系統使用選取元來評量所有合格的伺服器資源：

1. 系統會就含有選取元中所有必要變數的伺服器清單，進行編譯。如果沒有伺服器資源符合必要的選取元準則，則專案或步驟會失敗，系統會建立一份附註。
2. 如果有多個伺服器資源符合必要準則，系統會評比每一個合格的伺服器資源，並依下列方式指派分數：
 - 每符合一個選用的 變數，就給予一分。如果選取元包含相同變數的多個副本，系統對於每一個副本各給一分。
 - BF_LOADRATIO 值最低的伺服器資源再給一分。
3. 系統會選擇得分最高的伺服器。如果有多個伺服器資源擁有最高分，系統會從中挑選。

您可以在選取元中重複使用選用的變數。如此做可以提高某個符合這些變數的伺服器分數。比方說，如果需要的記憶體是 1 GB，但您強烈希望能有 2 GB 或以上的記憶體，則可以依下列方式設定選取元變數：

- 指定必要的 MEM_TOTAL GE 1 GB。
- 指定三次選用的 MEM_TOTAL GE 2 GB。

收集器

關於收集器

收集器定義所要收集並指派給伺服器資源的內容。

「伺服器」畫面的「收集器」區段會列出可用的收集器，並且可讓您建立新的收集器。

若要開啓「伺服器」畫面的「收集器」區段，請按一下**伺服器 > 收集器**。

收集器由一個以上內容組成。每一個內容各指定資訊清單中所要包含的資訊。

註：收集器可讓您使用動態選取元。如果要使用收集器，則需要 Enterprise Edition。

關於「收集器」畫面

您可以使用「收集器」畫面來建立、編輯及刪除收集器。若要檢視此畫面，請選取**伺服器 > 收集器**。



此畫面具有下列欄位。

名稱 收集器的名稱。

存取 可使用此收集器的使用者存取群組。

建立收集器

建立收集器來動態選取專案的伺服器資源。

程序

1. 選取**伺服器 > 收集器**。
2. 指定收集器的名稱。
3. 指定收集器的存取權。
4. 按一下**儲存**。畫面會變更，使您可以輸入變數來決定要收集哪些資料。
5. 建立並儲存一個以上的變數。如需變數內容的相關資訊，請參閱『收集器變數內容類型』。

收集器變數內容類型

收集器變數內容能指定如何收集伺服器資訊清單的相關資訊。

您可以在收集器中定義下列類型的內容：

設定值 這些內容會指派一個具名的靜態值給伺服器。您將指定內容名稱和值。

值當中可以使用一些特殊值，以便取得預先定義的回應。特殊值以底線字元 (_) 爲開頭。請參閱第 256 頁的『收集器中的特殊內容』。

內建 這些內容會傳回指派給伺服器資源之主機的相關資訊。如需內建內容清單，請參閱第 252 頁的『內建內容』。

執行指令

這種類型的內容指定系統所要執行的指令。內容值會設為執行指令所產生的輸出。依預設，會使用輸出的前 255 個字元。您可以使用正規表示式來擷取指定的輸出段落。

- **內容：**內容的名稱
- **指令：**要在指派給伺服器資源之主機上執行的指令
- **正規表示式：**（選用）用來過濾輸出的正規表示式。如果指定的話，建置系統就會試圖比對正規表示式與指令輸出的每一行。當系統第一次找到相符的字行時，就會擷取 \$1 的值（這是 Perl 的慣例），並以 \$1 作為該內容的值。正規表示式必須至少包含一組括弧，才能傳回值。如需建構 Perl 正規表示式的相關資訊，請參閱 Perl 文件。

併入 這種類型的內容會指定收集器清單。您可以將收集器巢狀化。當您建立「併入」類型的收集器時，您將指定另一個收集器的名稱以作為其值。在建置系統建立或更新資訊清單時，系統就會從所參照的收集器來插入內容。

提示：系統會依照在收集器中列示收集器變數內容的順序來套用這些內容；後面的同名內容會置換前面的內容。當您將一個收集器併入另一個收集器時，請使用這個特性。如果您只要使用收集器的部分變數內容，而不要收集全部的內容，請置換您不使用的內容。

系統也會自動套用少數內容，例如：含有伺服器邏輯名稱的 BF_NAME 內容。這些會視為特殊資訊清單內容的一部分。請參閱第 255 頁的『預設內容』。

資訊清單和動態選取伺服器

您可以合併使用收集器、資訊清單與選取元，以便在執行時期為專案或步驟選擇一個伺服器資源。

有三個不同的資料物件可讓系統動態選擇伺服器，分別如下：

- **收集器：**這個物件定義一組內容，以供系統從伺服器資源中收集，或指派給伺服器資源。系統會在檢查伺服器資源的內容時，執行收集器。所收集的內容值會儲存在資訊清單中。
- **資訊清單：**是特定伺服器的內容清單。其中含有因執行收集器而產生的結果。
- **選取元：**這是一份內容與比較項的清單，例如，MEM_TOTAL = 512。系統可以比較選取元的內容與資訊清單，看看是否有伺服器符合特定選取元的需求。專案和步驟會指定一個選取元，以作為其中一個內容。在專案或步驟執行時，會將選取元與所有已定義伺服器資源的資訊清單進行比較，以便選擇要在其中執行的伺服器資源。

下列範例顯示如何建立和使用簡式選取元：

1. 建立一個名為 Mercury 的伺服器資源，並讓它與代理程式產生關聯。
2. 建立一個名為 Mercury 的選取元。將它設為使用 BF_NAME = Mercury 來選取伺服器。
3. 建立一個名為 Lincoln 的專案。將 Mercury 選取元指派給它。

在您執行 Lincoln 專案時，系統就會選取名為 Mercury 的伺服器資源。如果該伺服器資源無法使用，專案就會失敗。

下列範例顯示如何設定從一組伺服器中動態選取伺服器：

1. 建立一個名為 RAMSIZE 的收集器。將它設為收集內建內容 MEM_TOTAL。
2. 建立伺服器資源，並與主機產生關聯。將每一個設為使用收集器 RAMSIZE。
 - Mercury 主機，具有 512 MB RAM
 - Mars 主機，具有 1 GB RAM
 - Jupiter 主機，具有 3 GB RAM
3. 建立一個名為 BigRam 的選取元。將它設為選取一個「標準內容」：內容=MEM_TOTAL，運算子=GE（大於或等於），以及值=2048。MEM_TOTAL 以 MB 表示。這個選取元只會選取 RAM 為 2 GB 或以上的主機。
4. 建立一個名為 SmallRam 的選取元。將它設為選取「標準內容」：內容=MEM_TOTAL、運算子=GE（大於或等於），以及值=256。此選取元只會選取 RAM 為 256 MB 或以上的主機。
5. 建立兩個專案：
 - HighMaint：將這個專案設為使用選取元 BigRam。
 - LowMaint：將這個專案設為使用選取元 SmallRam。

當您執行 HighMaint 時，系統會選擇 Jupiter 伺服器，因為只有它符合至少 2 GB RAM 的選取元需求。

當您執行 LowMaint 時，系統會從這三個可用的伺服器資源中挑選任一個。

如果您之後為主機新增一個名為 Neptune 且擁有 2 GB RAM 的伺服器資源，則在 HighMaint 專案下次執行時，便會為該專案選取 Neptune 或 Jupiter。如果 Jupiter 因某些原因而關閉，就會選取 Neptune。因為只剩下它符合選取元。

檢視資訊清單

如果要檢視伺服器的資訊清單，請執行下列動作：

1. 按一下**伺服器**。
2. 按一下伺服器資源的名稱。
3. 按一下**資訊清單**標籤。

您不可以直接變更伺服器的資訊清單。資訊清單內容是以收集器來定義。

資訊清單的自動重新整理間隔是由伺服器內容或系統設定來決定。若要在間隔期間重新整理，請按一下**佇列資訊清單重新整理**。

手動重新整理資訊清單

如果要手動重新整理資訊清單，請執行下列動作：

1. 按一下**伺服器**，顯示伺服器清單。
2. 選取一部伺服器。
3. 按一下**佇列資訊清單重新整理**。

即會執行伺服器連線測試，並且會更新伺服器資訊清單。**資訊清單**標籤中會出現**重新整理資訊清單**進度列，直到重新整理完成為止。

伺服器資訊清單自動更新間隔是由**伺服器測試頻率**系統設定來設定。在執行佇列中的任何資訊清單自動重新整理之前，會先執行佇列中的資訊清單手動重新整理。

設定伺服器資訊清單的更新頻率

您可以使用系統設定來控制系統向伺服器查詢其資訊清單內容的頻率。

關於這項作業

您可以使用名稱 `OS_SYSNAME` 來定義變數，讓收集器收集由收集器指派的伺服器的 Build Forge[®] 代理程式版本。請完成下列步驟來建立這個變數。

程序

1. 按一下**伺服器 > 收集器**。
2. 選取現有的收集器，或建立新的收集器。
3. 將變數新增至收集器：
 - a. 選取**內建類型**。
 - b. 選取變數名稱 `OS_SYSNAME`。
 - c. 按一下**儲存**。
4. 必要的話，可變更多部伺服器的收集器內容，以符合您剛才編輯的收集器。

靜態與動態選取伺服器的設定範例

下列範例顯示如何建立與使用簡式靜態選取元：

1. 建立一個名為 `Mercury` 的伺服器資源，並讓它與代理程式產生關聯。
2. 建立一個名為 `Mercury` 的選取元。將它設為選取「標準內容」：內容=`BF_NAME`、運算子=`EQ`（等於）、值=`Mercury`，以及「必要」。
3. 建立一個名為 `Lincoln` 的專案。將 `Mercury` 選取元指派給它。

在您執行 `Lincoln` 專案時，系統就會選取名為 `Mercury` 的伺服器資源。如果該伺服器資源無法使用，專案就會失敗。

下列範例顯示如何設定從一組伺服器中動態選取伺服器：

1. 建立一個名為 `RAMSIZE` 的收集器。將它設為收集內建內容 `MEM_TOTAL`。
2. 建立伺服器資源，並與主機產生關聯。將每一個設為使用收集器 `RAMSIZE`。
 - `Mercury` 主機，具有 512 MB RAM
 - `Mars` 主機，具有 1 GB RAM
 - `Jupiter` 主機，具有 3 GB RAM
3. 建立一個名為 `BigRam` 的選取元。將它設為選取「標準內容」：內容=`MEM_TOTAL`、運算子=`GE`（大於或等於）、值=`2048`，以及「必要」。 `MEM_TOTAL` 以 MB 表示。這個選取元只會選取 RAM 為 2 GB 或以上的主機。
4. 建立一個名為 `SmallRam` 的選取元。將它設為選取「標準內容」：內容=`MEM_TOTAL`、運算子=`GE`（大於或等於），以及值=`256`。此選取元只會選取 RAM 為 256 MB 或以上的主機。
5. 建立兩個專案：
 - `HighMaint`：將這個專案設為使用選取元 `BigRam`。
 - `LowMaint`：將這個專案設為使用選取元 `SmallRam`。

當您執行 `HighMaint` 時，系統會選擇 `Jupiter` 伺服器，因為只有它符合至少 2 GB RAM 的選取元需求。

當您執行 LowMaint 時，系統會從這三個可用的伺服器資源中挑選任一個。

如果您之後為主機新增一個名為 Neptune 且擁有 2 GB RAM 的伺服器資源，則在 HighMaint 專案下次執行時，便會為該專案選取 Neptune 或 Jupiter。如果 Jupiter 因某些原因而關閉，就會選取 Neptune。因為只剩下它符合選取元。

內容參照

下列主題說明會影響資訊清單的內容：

- 『內建內容』
- 第 255 頁的『預設內容』
- 第 256 頁的『收集器中的特殊內容』

內建內容

管理主控台會從伺服器收集內建內容，然後將這些值指派給伺服器資訊清單。

內建內容

內建內容供建置系統中幾個不同的資料物件使用：

- **選取元**可使用內建內容作為選取元變數，以比對伺服器與那些內容中的特定值。
- **收集器**會使用內建內容來收集伺服器的資料。
- **資訊清單**儲存內建內容的值（如果已收集的話）。

內建內容不會自動新增。您必須將內建內容新增至收集器，才會在資訊清單中顯示該內容。這份表格列出和說明內建內容。

註：內容的可用性會隨著平台而有所不同。

表 8. 收集器及資訊清單的內建內容

內容	說明
CPU_ARCH	回覆值是架構名稱的 label，如下所示： <ul style="list-style-type: none">• HP-PA：HP Precision Architecture• IA-64：Intel Itanium• MVS：IBM S/390®• PPC：PowerPC®• PPC-64：PowerPC 64• SPARC：Sun SPARC• X86：Intel、AMD、Cyrix 及其他公司使用的 x86 相容架構
CPU_LOAD（僅適用 Windows）	CPU 資料流量或 CPU 使用率，以容量的百分比表示（在 0 和 100 之間）。
CPU_LOAD1	回覆值會報告在過去 1 分鐘以來正在執行或者等待執行的平均程序數（平均資料流量）。 CPU_LOAD1 是 CPU 活動的度量單位。一部閒置的電腦，其資料流量為 0。每有一項處理程序正在使用 CPU 或者等待使用 CPU，就會讓資料流量增加 1。 註： 在 Windows 中，每一個處理程序不論是否為作用中，都會增加資料流量。此外，這項資訊是由 bfdispatch 處理程序所收集，並且使用共用記憶體區段發佈給代理程式。如果連接代理程式時所用的使用者認證未獲得專用權，將無法使用這些統計資料。

表 8. 收集器及資訊清單的內建內容 (繼續)

內容	說明
CPU_LOAD5	<p>回覆值會報告在過去 5 分鐘以來正在等待執行的平均程序數（平均資料流量）。</p> <p>CPU_LOAD5 是 CPU 活動的度量單位。一部閒置的電腦，其資料流量為 0。正在使用 CPU 或者等待使用 CPU 的每一個處理程序，是以 1 為增量新增至資料流量中。</p> <p>註：在 Windows 中，每一個處理程序不論是否為作用中，都會增加資料流量。此外，這項資訊是由 bfdispatch 處理程序所收集，並且使用共用記憶體區段發佈給代理程式。如果連接代理程式時所用的使用者認證未獲得專用權，將無法使用這些統計資料。</p>
CPU_LOAD15	<p>在過去 15 分鐘內等待執行的平均程序數（平均資料流量）。</p> <p>CPU_LOAD15 是 CPU 活動的度量單位。一部閒置的電腦，其資料流量為 0。正在使用 CPU 或者等待使用 CPU 的每一個處理程序，是以 1 為增量新增至資料流量中。</p> <p>註：在 Windows 中，每一個處理程序不論是否為作用中，都會增加資料流量。此外，這項資訊是由 bfdispatch 處理程序所收集，並且使用共用記憶體區段發佈給代理程式。如果連接代理程式時所用的使用者認證未獲得專用權，將無法使用這些統計資料。</p>
CPU_MHZ	<p>這個內容報告處理器的速率（以 MHz 為單位）。需要某些條件，才能順利填寫這個內容：</p> <ul style="list-style-type: none"> • 在 Linux 中：必須啟用頻率調整。 • Windows：~MHz 登錄項目必須存在且已填寫。 • 若為 x86 和 x86-64 處理器，則列入組合必須運作。
CPU_MANUFACTURER	<p>這個內容會傳回處理器製造商的公司名稱。如果無法直接取得資訊，則會根據架構來假設名稱。如果沒有足夠的處理器資訊可用，則不會傳回任何值。支援的值如下：</p> <ul style="list-style-type: none"> • AMD：代表其 x86 和 AMD64 處理器 • Cyrix：代表其 x86 相容處理器 • DEC：代表 Alpha 和 VAX • HP：Hewlett-Packard Precision Architecture • IBM：IBM S/390 和 PowerPC G5 • Intel：Intel x86（包括 Intel64）、IA-64 Itanium • Motorola：PowerPC G4 • NexGen：x86 相容處理器 • National：National Semiconductor x86 相容處理器 • Rise：Rise x86 相容處理器 • Sis：Sis x86 相容處理器 • Sun：Sun Microsystems SPARC • TransMeta：TransMeta x86 相容處理器 • UMC：UMC x86 相容處理器 • VIA：VIA® Technologies x86 相容處理器

表 8. 收集器及資訊清單的內建內容 (繼續)

內容	說明
CPU_MODEL	<p>這個內容會傳回製造商特有的 CPU 型號。所報告的值如下：</p> <ul style="list-style-type: none"> • x86 架構 <ul style="list-style-type: none"> – 386 – 486 – 586 – 686 – X86_64 • PowerPC 架構 <ul style="list-style-type: none"> – 6xx – POWER® – RS64 – G3 – G4 – G5 – Cell
CPU_SERIAL	<p>這個內容會傳回 CPU 或電腦的序號。目前這項功能只限於下列架構：</p> <ul style="list-style-type: none"> • x86：僅限於 Intel 或 Transmeta 序號。附註：大部分 x86 處理器不會報告序號。在這類情況下，不會傳回任何值。 • MacOS/X：指派的序號擷取自 I/O 登錄。如果要報告內容，必須能夠找到 CoreFoundation 和 IOKit 架構。
DISK_FREE	<p>在 UNIX 和 Linux 中，這個內容會傳回伺服器「路徑」所指定的檔案系統上，可用的空間量 (MB)。</p> <p>在 Windows, 中，這是指伺服器「路徑」所指定之磁碟機上的可用磁碟空間 (MB)。</p> <p>例如，有 4 GB 可用空間的磁碟會報告成 4096 MB。</p>
DISK_TOTAL	<p>這個內容會傳回可用的磁碟空間總計。這個值是針對代理程式的基本路徑而報告的，此路徑的個別配置可能小於整個剩餘磁碟或分割區。磁碟空間管理會因作業系統的不同而有很大的差別。</p>
MEM_LOAD (僅適用於 UNIX/Linux)	<p>在 UNIX 和 Linux 中，這是指目前使用中的 RAM 量或系統記憶體量，以總實際記憶體量的百分比表示 (0 到 100 之間)。</p>
MEM_FREE	<p>這個內容會傳回 RAM 量或系統記憶體量 (MB)。例如，2 GB 的可用 RAM 會報告成 2048 MB。</p>
MEM_PAGESIZE	<p>這個內容會傳回 RAM 或系統記憶體的分頁大小 (MB)。此圖例代表主機系統的標準分頁大小。例如，分頁大小為 4 KB 的主機系統會報告成 4096 MB。</p>
MEM_TOTAL	<p>這個內容會傳回 RAM 量或系統記憶體量總計 (MB)。</p> <p>例如，具有 2 GB RAM 的電腦會報告成 2048 MB。</p>
NET_FQDN	<p>這個內容會傳回正在執行代理程式之電腦的完整網域名稱 (FQDN)。這會根據代理程式通訊時所用的位址，來報告 FQDN。傳回的位址可能是 IPv4 型或 IPv6 型，這取決於實際使用的位址而定。另請參閱 NET_IPV、NET_IPV4 和 NET_IPV6。</p>
NET_HWADDR	<p>這個內容會傳回 NET_IFACE 中所報告介面的硬體位址。</p>

表 8. 收集器及資訊清單的內建內容 (繼續)

內容	說明
NET_IFACE	這個內容會傳回代理程式通訊時所用的介面名稱。 <ul style="list-style-type: none"> 在 Windows 中，所報告的名稱會是 ipconfig 指令傳回的名稱，例如 Intel(R) PRO/100 VE Network Connection - Packet Scheduler Miniport 在其他作業系統中，所報告的名稱為 ifconfig 傳回的名稱，例如 en0、eth0 或 OSA1。
NET_IPV	這個內容會傳回用來與代理程式通訊的 IP 連線類型，4 代表 IPv4，6 代表 IPv6。
NET_IPV4	這個內容會傳回代理程式通訊時所用的 IPv4 位址。在透過 IPv6 的連線上，如果代理程式能夠識別相同介面的 IPv4 位址，則會報告該位址。
NET_IPV6	這個內容會傳回代理程式通訊時所用的 IPv6 位址。
NET_SPEED (僅適用於 Windows)	這個內容會傳回介面速度 (以 Mb/sec 為單位)，例如 1000 代表超高速乙太網路。
NUM_CPU	這個內容會傳回電腦上的 CPU 數目。
OS_HOSTID	這個內容會傳回 gethostid() 系統呼叫的結果。一般而言，除非系統管理者將 /etc/hosts 設成一個具參考性的值，否則此結果不具參考性。
OS_SYSNAME	這個內容會傳回伺服器的作業系統名稱。例如：Microsoft Windows XP、AIX、Macintosh OS。
OS_RELEASE	這個內容會傳回伺服器的作業系統版本層次。 例如，如果伺服器作業系統是 Microsoft XP 5.1.2600 版，此回覆值會是 5。
OS_VERSION	這個內容會傳回伺服器的作業系統版本。 例如，如果伺服器作業系統是 Microsoft XP 5.1.2600 版，此回覆值會是 1。
WIN_SERVICEPACK (僅適用於 Windows)	這個內容會傳回安裝在伺服器之 Windows 服務套件的版本號碼。比方說，如果是 Service Pack 2，這個值便是 2。

預設內容

某些資訊清單內容會自動設定。

以下是會自動設定的資訊清單內容。這些內容與內建內容的不同之處在於，不需要將它們新增至收集器，即可直接移入。

在表格說明中標示為**選取元**的內容，可用在選取元中。其他內容則只能作為在資訊清單中提供資訊用。

表 9. 自動設定資訊清單內容

內容	說明
BF_AGENT_VERSION	選取元 。安裝在伺服器上的代理程式版本號碼。
BF_EXCLUSIVE	選取元 。它是沒有運算子或值的旗標。如果選取元包含這個內容，在工作持續期間，會保留所選伺服器上的所有插槽。 如果工作中的某個步驟指定要在不同的伺服器上執行，在另一部伺服器執行這個步驟期間，會繼續保留現行伺服器上的所有現行插槽。
BF_JOBS	選取元 。可讓伺服器資源同時執行的工作（步驟）數目。每當主控台指派步驟給伺服器，這個值就會更新，而這與其他資訊清單內容的更新無關。

表 9. 自動設定資訊清單內容 (繼續)

內容	說明
BF_LAST_REFRESH	前次更新資訊清單中內建內容的時間。此值採用 UNIX [®] 時間戳記格式來報告：從 1970 年 1 月 1 日至今的秒數。
BF_LASTJOBS	前次重新整理資訊清單時，伺服器正在執行的工作數。
BF_LAST_UPDATE	前次將執行指令內容更新至資訊清單的時間。此值採用 UNIX [®] 時間戳記之類的格式來報告，以 1970 年 1 月 1 日至今的秒數來表示。
BF_LOADRATIO	選取元。 這是一個報告成比例的計算值：工作數 (BF_JOBS) 除以容許伺服器執行的工作數上限（「工作數上限」設定）。假設伺服器正在執行一項工作，而「工作數上限」為 4，則負載比例為 .25。
BF_NAME	選取元。 指定要在其上執行的伺服器資源。此值為伺服器資源的名稱。BF_NAME 內容不會顯示在資訊清單的清單中。
BF_RESERVE	選取元。 它是沒有運算子或值的旗標。如果選取元包含這個內容，則在工作持續時間，就會在所選的伺服器上保留一個插槽。 <ul style="list-style-type: none"> 如果工作中的某個步驟指定要在不同的伺服器上執行，則在另一部伺服器執行該步驟期間，會保留所選伺服器上的插槽。 如果步驟明確指定所選的伺服器，則伺服器會將保留的插槽用於該步驟上。 <p>當專案中有一個以上步驟在其他伺服器上執行時，一旦專案在某部伺服器上遺失其插槽，這個旗標可避免該專案發生延遲。</p>

收集器中的特殊內容

當在收集器中指名一些設定值內容時，內容會在系統中造成某些行為。這些內容以底線字元為開頭。建置系統會使用這些內容的值，而在伺服器從收集器獲得這些內容時，將行為套用在那些伺服器上。

註：您不能建立以 "BF_" 字串開頭的內容，因為這些名稱已保留供系統使用。

表 10. 收集器和資訊清單的特殊設定值內容

內容	說明
_AGE	此內容定義應該重新整理資訊清單的頻率（以秒為單位）。預設值 86400 提供每天一次的更新。3600 的值會使系統每小時更新一次資訊清單。

使用 Snapshot 來建立選取元的新實例

建立選取元的 Snapshot，可針對您想變更或修改的選取元，快速建立新實例。

選取元 Snapshot 概觀

請檢閱這些主題，以瞭解選取元 Snapshot 及其用法。

選取元 Snapshot 使用案例

下列範例說明選取元 Snapshot 的一些一般使用案例：

- 建立選取元的 Snapshot，以變更選取元的配置或執行新工具或 Script 的測試，而同時還能繼續使用現有的選取元來執行工作。
- 將選取元的 Snapshot 儲存成暫時備份或正式保存檔的一部分。
- 建立選取元的 Snapshot，以擷取和某個里程碑（例如：外部或內部發行）相對應的某個時間點選取元配置。

選取元 Snapshot 的概念和術語

Snapshot 引進某些配合選取元使用的新概念和術語。

選取元 Snapshot：Snapshot 是現有選取元的新實例。關於 Snapshot，需要切記的一些關鍵點如下所示：

- Snapshot 是個別的選取元物件。變更 Snapshot 集當中的某個 Snapshot，並不會影響集合中的其他 Snapshot。
- Snapshot 不是副本。如果您針對選取元的相關聯物件建立 Snapshot，Snapshot 會建立物件的另一個實例。這會複製物件之間的對映關係，而不是建立新物件。
- Snapshot 不是選取元的修訂：
 - Snapshot 不支援比較兩個選取元 Snapshot 間的變更。
 - 不同於來源控制系統，選取元 Snapshot 的變更並不會以版本號碼來追蹤或識別。不過，您可以使用包含版本號碼（例如，7.5.0、3.4.01）的 Snapshot 命名方法，讓選取元 Snapshot 和里程碑產生關聯。

Snapshot 集：Snapshot 集是一組作為一個基礎 Snapshot 之後代的所有選取元 Snapshot。此集合中至少含有一個基礎或母項 Snapshot，以及一個子項 Snapshot。在使用者

介面中，選取元名稱旁的 Snapshot 圖示  表示選取元的 Snapshot 集已存在。

基礎 Snapshot：一開始，所有選取元的 Snapshot 名稱皆為「基礎 Snapshot」。您可以將「基礎 Snapshot」變更為另一名稱。基礎 Snapshot 是 Snapshot 集的母項。

預設選取元 Snapshot：預設選取元 Snapshot 是目前正在運作的選取元。Snapshot 集中只能有一個 Snapshot 作為預設值。如果您沒有指定預設 Snapshot，就會以基礎 Snapshot 作為預設值。

- 在主控台中，預設 Snapshot 會顯示在選取元清單的最上層。選取**伺服器 > 選取元**，即會顯示選取元清單。
- 在您選取具有 Snapshot 的選取元時，除非您在清單框中選取不同的選取元 Snapshot，否則會使用預設選取元 Snapshot。
- 如果要存取和處理選取元 Snapshot 集當中的其他 Snapshot，必須按一下 Snapshot

 圖示。

選取元 Snapshot 視圖

選取 Snapshot 圖示 ，以顯示 Snapshot 視圖。Snapshot 視圖會顯示 Snapshot 在集合中的階層：

- 基礎 Snapshot 位於最上層，如果您沒有另外指派一個唯一名稱，其名稱會是「基礎 Snapshot」。
- 所有的選取元 Snapshot 皆為基礎 Snapshot 的子項。相同基礎 Snapshot 的子項在 Snapshot 直欄中全縮排在同一層。
- 從子項 Snapshot 建立而來的選取元 Snapshot，會成為該子項 Snapshot 的子項，並在 Snapshot 直欄中縮排在下一層。

選取元 Snapshot 規劃

檢閱某些最佳實務，以選取預設選取元 Snapshot，並為選取元 Snapshot 命名。

• 用來在集合中選取預設 Snapshot 的策略

使用者介面只能辨識 Snapshot 集的一個預設或現行選取元 Snapshot。在選取預設 Snapshot 時，請採取一致的策略：

- 使用基礎 Snapshot 作為預設 Snapshot

使用 Snapshot 作為備份。僅對基礎 Snapshot 進行變更。不對備份的選取元 Snapshot 進行變更。

- 使用最新 Snapshot 作為預設 Snapshot

每當您建立新的選取元 Snapshot 時，就會讓它成為新的預設選取元 Snapshot。不會對基礎選取元 Snapshot 或更早的選取元 Snapshot 進行變更。

• 識別 Snapshot 集的 Snapshot 命名方法

此選取元 Snapshot 名稱在選取元 Snapshot 集當中必須是唯一的。

請使用下列一種準則，來協助您建立選取元 Snapshot 名稱：

- 名稱應具備敘述性：應指出 Snapshot 的用法或用途。
- 命名方法應遵循定義的標準。您可以使用 Snapshot 標籤上的「註解」框，來說明命名方法。

• 讓集合使用單一選取元名稱

在您建立選取元 Snapshot 後，可讓您選擇變更選取元的名稱。如果您變更選取元名稱，會更新每一個選取元 Snapshot 的這個名稱。

從現有的選取元或選取元 Snapshot 來建立選取元 Snapshot



建立選取元 Snapshot 時，將會建立新的選取元實例。Snapshot 並非副本；它是一個新選取元。

提示：如果要建立 Snapshot 和變更預設 Snapshot，使用者必須具備必要許可權。請參閱第 317 頁的『驗證與編輯 Snapshot 許可權的存取群組』。

1. 按一下您要建立 Snapshot 之選取元或選取元 Snapshot 旁的**編輯**圖示：

- 若要建立預設選取元 Snapshot 的 Snapshot，請在選取元清單（**伺服器 > 選取元**）

中，按一下最上層 Snapshot 旁的「編輯」圖示 。

- 若要建立非預設選取元 Snapshot 的 Snapshot，請按一下 Snapshot 圖示 。Snapshot 視圖會顯示集合中的選取元 Snapshot。按一下非預設選取元 Snapshot 旁的「編輯」圖示 。

2. 按一下**建立新 Snapshot**。
3. 在 Snapshot 標籤的**名稱**中，輸入 Snapshot 名稱。名稱會指派給隨選取元一起建立 Snapshot 的所有物件。

此名稱在選取元 Snapshot 集當中必須是唯一的。

4. 選取當您建立選取元 Snapshot 時，要建立 Snapshot 的 Build Forge 物件。下表說明您所能選取的物件。

物件	說明
預設值	在使用者介面中，預設 Snapshot 會顯示在選取元清單的最上層。 選取 伺服器 > 選取元 ，即會顯示選取元清單。
遵循選取元併入項目	如果選取元使用 併入 內容類型來併入其他的選取元，則會為這些選取元建立 Snapshot。 註：併入變數類型取代了舊版提供的 .include 功能。



5. 按一下**儲存**，以儲存選取元 Snapshot。

變更預設選取元 Snapshot

預設選取元 Snapshot 是選取元 Snapshot 集當中的最上層 Snapshot，它會顯示在選取元清單中（**伺服器 > 選取元**）。

提示：如果要建立 Snapshot 和變更預設 Snapshot，使用者必須具備必要許可權。請參閱第 317 頁的『驗證與編輯 Snapshot 許可權的存取群組』。

如果要變更預設選取元 Snapshot，請編輯您要作為新預設值之 Snapshot 的 Snapshot 定義：

1. 選取**伺服器 > 選取元**。
2. 在選取元清單中，按一下預設選取元 Snapshot 的 **Snapshot** 圖示 。
3. 在 Snapshot 清單中，按一下要作為新預設值的選取元 Snapshot 的**編輯**圖示 。
4. 按一下**設為預設值**。
5. **重要事項：**在蹦現式畫面中，選擇「確定」或「取消」。



確定	更新參照： 若有物件參照了先前的預設選取元，請更新這些物件以使用新的預設選取元。
取消	不更新參照： 針對參照先前預設值的任何物件，不將參照更新為新的預設選取元 Snapshot。

變更選取元 Snapshot 的 Snapshot 名稱

您可以變更選取元 Snapshot 的 Snapshot 名稱，也可以針對當初您建立選取元 Snapshot 時選取要建立 Snapshot 的物件，變更其 Snapshot 名稱。

在基礎 Snapshot 方面，您可以使用這個選項，僅針對單一選取元 Snapshot 或針對所有現行和未來的選取元，將其「基礎 Snapshot」預設名稱變更為另一個 Snapshot 名稱。

如果要變更 Snapshot 名稱，請執行下列動作：

1. 選取**伺服器 > 選取元**。
2. 在選取元清單中，按一下預設選取元 Snapshot 的 **Snapshot** 圖示 。
3. 在 Snapshot 清單中，按一下選取元 Snapshot 的**編輯**圖示 。
4. 選取 **Snapshot** 標籤。
5. 在**名稱**中，輸入新名稱。
6. **選用**：在**註解**中輸入一項註解。
7. **重要事項**：在蹦現式畫面中，選擇「確定」或「取消」。

確定	<p>變更選取元 Snapshot 名稱和其他 Snapshot 物件名稱：針對當初您建立選取元 Snapshot 時選取要建立 Snapshot 的物件，變更這些物件和選取元 Snapshot 的名稱。</p> <p>在「基礎 Snapshot」方面：變更所有現行選取元 Snapshot 和所有未來選取元 Snapshot 的「基礎 Snapshot」名稱。</p>
取消	<p>變更選取元 Snapshot 名稱，但不變更其他 Snapshot 物件名稱：針對當初您建立選取元 Snapshot 時選取要建立 Snapshot 的物件，不變更這些物件的名稱。僅變更選取元 Snapshot 的名稱。</p> <p>在「基礎 Snapshot」方面：維持所有現行選取元 Snapshot 和所有未來選取元 Snapshot 的「基礎 Snapshot」名稱。</p>

存取和檢視選取元 Snapshot 集當中的 Snapshot

建立選取元的 Snapshot，可針對您想變更或修改的選取元，快速建立新實例。

建立選取元 Snapshot 時，會建立一個 Snapshot 集，其中至少含有下列兩個選取元：基礎選取元和新選取元 Snapshot。



如果要檢視 Snapshot 集當中的所有選取元 Snapshot，請執行下列動作：

1. 選取**伺服器 > 選取元**。

選取元清單顯示選取元和選取元 Snapshot 清單。最上層 Snapshot 是預設選取元 Snapshot。

2. 按一下 **Snapshot** 圖示 ，即會顯示 Snapshot 集中的選取元 Snapshot。

在 Snapshot 視圖中，您可以：



- 建立新的選取元 Snapshot。若要開始，請按一下**編輯**圖示 。
- 變更選取元的預設 Snapshot。按一下**編輯**圖示 ，然後按一下**設為預設值**。
- 以編輯標準選取元的方式，來編輯選取元 Snapshot 定義。

刪除選取元 Snapshot

您可以使用「刪除」選項，來刪除選取元 Snapshot。

若有另一個物件正在使用選取元，則無法刪除該選取元。例如，如果有另一個物件併入它，或者某個專案、步驟或排程使用它，則無法將它刪除。

如果要刪除選取元 Snapshot，請執行下列動作：

1. 選取**伺服器 > 選取元**。
2. 在選取元清單中，按一下基礎 Snapshot 的 **Snapshot** 圖示 。
- Snapshot 視圖會顯示集合中的選取元 Snapshot。
3. 按一下要刪除的選取元 Snapshot 旁的**編輯**圖示 。
4. 按一下**刪除**。

第 18 章 使用環境

本節說明如何設定與管理環境。

關於環境

環境是一組具名的變數。

一旦定義環境，即可用來執行下列事項：

- 對工作要使用的步驟設定變數。環境會指派給伺服器、專案和步驟。在工作執行期間，執行中的步驟會繼承下列三個環境中的變數值：
 - 執行工作的伺服器環境。
 - 定義步驟的專案環境。
 - 與步驟相關聯的步驟環境。
- 設定排定的工作所要使用的變數。設定給所排定工作的環境會取代指定給專案的環境。
- 設定配接器所要使用的變數。環境可以指派給配接器鏈結。專案的起始配接器步驟會使用它。

環境繼承

在系統執行步驟之前，會先建立步驟環境。步驟環境由適用於該步驟的所有變數組成。變數依序繼承自伺服器環境、專案環境，以及步驟環境。以下是基本案例：

1. 伺服器環境：將伺服器環境變數複製到步驟環境。
2. 專案環境：將專案環境變數套用到步驟環境。如果專案環境中有一個變數和伺服器環境中的變數同名，則會根據這兩個變數定義中的變數動作 來更新值。
3. 步驟環境：將步驟環境變數套用到步驟環境。如果步驟環境中有一個變數和繼承自伺服器與專案環境的變數同名，則會根據變數動作 來更新值。

變數的變數動作會在繼承值時，直接影響這些值的套用方式。例如：

- 案例 1：當變數動作為「設定」時，則經由繼承來改寫值。
 - 伺服器環境：X = 1，動作：設定
 - 專案環境：X = 100，動作：設定
 - 步驟環境：X = 3，動作：設定
 - 步驟執行期間的最終值：X = 3

變數 X 先是設為 1，接著是 100，然後是 3。每次套用新值時，「設定」變數動作就會取代變數值。

- 案例 2：由於「若未設定則設定」變數動作而繼承值。
 - 伺服器環境：Y = 1，動作：設定
 - 專案環境：Y = 100，動作：設定
 - 步驟環境：Y = 3，動作：若未設定則設定
 - 步驟執行期間的最終值：x = 100

變數 Y 先是設為 1，之後由於對伺服器與專案環境中的 Y 執行「設定」動作，改設為 100。由於 Y 在步驟環境中使用**若未設定則設定**變數動作，而會繼承設定在專案環境中的值。

特殊的繼承案例

以下案例會影響繼承。

列入專案

當在步驟的**列入**內容中指定專案時，步驟中會列入一個專案。當步驟中列入專案時，就不會使用被呼叫的專案伺服器環境和專案環境。繼承的順序如下所示：

1. 呼叫端步驟的伺服器環境。
2. 呼叫端步驟的專案環境。
3. 呼叫端步驟的步驟環境。
4. 被呼叫專案中的每一個步驟都會繼承步驟環境（若有指定的話）。

鏈結的專案

專案或步驟可以將專案指定成**通過**或**失敗**鏈。當採取這種方式來呼叫專案時，它會在本身的環境中執行。此外，它有權存取呼叫端專案或步驟中的所有變數。這些變數會複製成新名稱，並使用字首 **BF_CALLER_**。例如：呼叫端專案或步驟中的 **BF_NAME** 變數，可在被呼叫的專案和步驟中以 **BF_CALLER_BF_NAME** 形式提供。

排定的工作

當專案已列入**排程**中的某項排程時，您可以選擇在專案上套用非預設的不同環境。一旦指定環境，就會使用**環境**標籤，設定該環境中的變數值。在**環境**標籤中，會根據於**專案**內容設定來呈現變數。未排定的工作啟動也遵循相同的規則（如果有的話）。

置換繼承順序

使用**最後套用伺服器環境**系統設定來置換繼承順序。如果其值為是，則繼承順序的設定如下：

1. 專案環境
2. 步驟環境
3. 伺服器環境

啟動工作時變更專案變數

當您啟動工作時，可以變更專案，置換專案環境中設定的變數值。

當您啟動工作時，**工作啟動**頁面上會呈現專案環境中的變數。視每一個變數的於專案內容而定，必要時您可以變更所呈現的值。

在工作啟動時所進行的變更，必須遵守專案環境中所定義的變數所採用的相同繼承規則。

例如：

1. 您將 **JavaEnv** 環境定義成具有一個變數，其起始值為 **JavaVersion = 1.4**。
2. 您將 **MyBuild** 專案定義成使用 **JavaEnv** 環境。

3. 請您啟動一項工作來執行 MyBuild 專案。在**工作啟動**畫面中，您將 JavaVersion 的值變更為 1.5。

如此一來：

- 未置換專案環境 *JavaEnv* 的步驟，會繼承修改後的 JavaVersion 值 1.5。
- 明確使用 JavaEnv 環境作為步驟環境的步驟，會使用定義在專案環境中的 JavaVersion 值：1.4。

關於變數

變數是定義在環境中。

變數中除了值，另有其他內容，用來在解譯期間控管其行為。

名稱	類型	值	動作	於專案
MyStandardLevel	標準變數	High	設定	一般
Flavor	下拉清單		設定	必要
包含	包含	suppress ENV		

(新建變數) 儲存變數 刪除變數

詳細資料 下拉選項

類型: 標準變數 名稱: 值:

動作: 設定 於專案: 一般

如果要建立新變數，請執行下列動作：

1. 按一下左功能表中的**環境**。
2. 按一下**新增環境變數**。
3. 指定變數的名稱。
4. 視需要指定變數的其他內容。如需相關資訊，請參閱第 267 頁的『環境內容』。
5. 按一下**儲存變數**。

在建立完成後，就可以按一下變數來編輯。

解譯步驟中的變數

您可以在步驟指令或環境變數定義中，使用 UNIX[®] 樣式或 Windows[®] 樣式的變數語法。

系統使用前置處理器，將 UNIX 樣式 (\$VAR) 或 Windows 樣式 (%VAR%) 的語法，解譯成適用於執行步驟之伺服器的格式。預先剖析可以讓步驟在 Windows 型伺服器或 UNIX 型伺服器上執行。

範例：

- 在步驟中，下列兩項指派陳述式是同義的：

```
echo %fooVar%      # Windows syntax
echo $fooVar       # UNIX or Linux syntax
```

- 變數的指派不會預先處理。因此，請避免在指令行中使用直接指派，尤其是在沒有作業系統限制就直接選取伺服器的情況下，更不可如此做。在環境中使用變數。

```
set fooVar=100     # Windows
fooVar=200         # UNIX or Linux syntax
```

如何剖析變數：

- 預先剖析器會求值變數指派。除非已用反斜線字元跳出，否則會使用特殊字元（\$、%、{、}、"、'）。如果關閉了預先剖析，則會傳遞所有字元。
- 變數指派的每一端由目標環境求值。
- 執行求值的變數指派。

預先剖析器、Windows 環境及各種 UNIX 和 Linux Shell，對於特殊字元的解譯各不相同。請謹慎使用特殊字元和反斜線跳出字元。

解譯未宣告的變數

如果在步驟中所呼叫的變數尚未定義，所傳回的值，取決於預先剖析為開啓（預設）或關閉。預先剖析行為的設定方式，是編輯 `bfagent.conf` 檔案中的 `no_preparse` 指令，或是 `_NO_PREPARSE_COMMAND` 環境變數。請參閱第 276 頁的『觸發變數參照』。

變數格式	傳回的值 - 開啓預先剖析（預設值）	傳回的值 - 關閉預先剖析
echo %foo%	foo	Windows： %foo% UNIX 或 Linux：空白
echo \$foo	foo	Windows： \$foo UNIX 或 Linux：空白
echo \${foo}	foo	Windows： \${foo} UNIX 或 Linux：空白
echo \$[foo]	空字串	Windows： \$[foo] UNIX 或 Linux：系統錯誤

關於「環境」畫面

您可以使用「環境」畫面來建立及管理專案的環境。若要檢視畫面，請選取環境。



在「環境」畫面中，您也可以按一下某個環境，以顯示和編輯其變數。

「詳細資料」標籤

環境內容

請指定下列環境內容：

名稱 環境的名稱。

存取 可以使用這個環境之使用者的存取群組。

變數內容

在「環境」畫面中，選取變數來檢視「變數」畫面。您可以使用「變數」畫面來指定下列內容：

名稱 變數名稱只能在名稱中使用英數字元（a-z、A-Z、0-9）和底線字元（_）。長度上限為 255 個位元組：

- 單位元組字集：255 個字元
- 雙位元組字集：127 個字元

值 變數值具有下列性質：

- **長度**：值的長度不拘（若有作業系統限制，則以其為上限）。
- **特殊字元**：對於預先剖析器（對作業系統求值之前）以及作業系統上的求值，%、\$、[、]、{、}、\、" 和 ' 等字元都有其特殊的意義。請避免使用它們。在傳遞時，請使用反斜線讓它們跳出。請參閱第 265 頁的『解譯步驟中的變數』。

如果用單引號括住值，就會依照字面來傳遞變數名稱，不進行解譯。例如，`MyEnv='$env'` 這項變數指派會將 `$env` 字串指派給 `MyEnv`，而不是指派 `$env` 變數的值。

- **值中的變數**：如果變數值中有變數，則會在執行使用變數的步驟時，解譯該變數。
- **下拉值**：若要指定下拉變數的項目，請將變數類型設為下拉清單、儲存變數，然後進行編輯。按一下**下拉選項**標籤，將項目新增至下拉清單中。請參閱第 270 頁的『建立變數的下拉清單』。
- **以點指令作為值**：某些點指令可以作為環境變數的值；在這些情況下，系統會以其他的值取代點指令。請參閱第 273 頁的『在變數中使用點指令』。
- **回車字元**：變數不會儲存回車字元。您可以指派一個多行值給變數，如下例所示。將 `text.txt` 檔的內容指派給 `test` 變數：

```
.bset env "test = `type text.txt`"
```

檔案中的字行是連續的。例如，假設檔案內容如下所示：

```
A first line  
And a second line
```

則變數值將變成下面這一行：

```
A first lineAnd a second line
```

類型 指派給變數的類型可為下列其中一項：

- **標準** - 預設值。可指派一個值與動作給變數。
- **併入** - 變數值指向另一個要併入的環境。該環境中的所有變數都會併入。

註：併入變數類型取代了舊版提供的 `.include` 功能。

- **下拉清單** - 變數含有一組可供使用者選擇的值。建立下拉清單變數之後，您可以選取它，並按一下**下拉選項**標籤以新增值。

動作

下列其中一項：

- **設定**：預設選項。將指定的值指派給變數。如果變數不存在，則會加以建立。
- **設定（如果沒有設定的話）**：只有在變數已無任何值時，這個動作才會指派值給該變數。請參閱第 265 頁的『關於變數』。
- **附加**：將值附加到變數的現行值後面。值之間會加上作業系統特有的「路徑」定界字元：
 - Windows：分號 (;)
 - UNIX 或 Linux：冒號 (:))
- **置前**：將值插入現行值前面。值之間會加上作業系統特有的「路徑」定界字元：
 - Windows：分號 (;)
 - UNIX 或 Linux：冒號 (:))
- **清除**：將值設為空字串。如果「值」內容含有一值，則不會採用。
- **刪除/取消設定**：將變數從現行套用的環境中刪除。如果「值」內容含有一值，則不會採用。
- **指派隱藏**：值在資料庫中會加密，且在日誌和使用者介面中會顯示為『*****』。請使用此選項來儲存機密性資訊。

依預設，「指派隱藏」變數是受到保護而無法編輯的。如果您沒有設定「編輯隱藏標籤」許可權，則無法編輯這些值。但是，如果已設定許可權，則您可以刪除再重建它們。如果設定「編輯隱藏標籤」許可權，則可以編輯值，然後輸入以供使用。此行為可防止您將「指派隱藏」變數的類型變更為設定以顯示其值。

如果環境變數的「動作」值是指派隱藏，則無法變更變數的「類型」及「動作」值。如果環境變數的「動作」值不是指派隱藏，則無法將「動作」變更為指派隱藏。

註：加長您的值可讓值變得更安全。

註：如果您先前已具有指派隱藏變數，則可以在 Build Forge 使用者介面「環境」頁面中編輯並重新儲存來加密它們。您也可以使用 `bfassignhiddenmigration` 公用程式，選擇以批次模式來加密。Build Forge 8.0 安裝不會變更指派隱藏值。如果您未重新儲存或升級 Build Forge 7.x 版中舊的指派隱藏變數值，則將無法使用新的指派隱藏特性。此行為與在 7.x 版中使用 Perl 引擎相同，但在新的 Java 引擎步驟日誌中，指派隱藏值會顯示為明碼。建議您重新儲存舊的指派隱藏值，或使用 `bfassignhiddenmigration` 公用程式來加密所有指派隱藏值。如需 `bfassignhiddenmigration` 公用程式的相關資訊，請參閱主題第 411 頁的『`bfassignhiddenmigration` 參照』。

註：如果步驟中的某個變數設為指派隱藏，則該步驟中的其他變數都預設為「指派隱藏」。

註：如果您在 Build Forge 7.x 中以「多位元組字元系統 (MBCS)」字元來設定指派隱藏變數（例如中文、日文或韓文），則不會自動升級隱藏移轉公用程式。建議您將變數取代為 ASCII 字元，以提高安全性。若沒有這麼做，由於這些行為與 Build Forge 7.x 相同，您將無法使用新的指派隱藏變數。

於專案 定義當您手動啟動工作時，要如何使用變數。這個內容只會影響指派給該專案的環境中所用的變數。當工作在執行中時，此內容不會影響變數。值可為下列其中一項：

- **一般**：當將變數指派給專案時，變數的行為正常。
- **必要**：變數必須有值存在。使用這種內容的變數，會在「啟動」畫面中強調顯示。定義在變數定義中的值便已夠用。如果沒有定義值，工作就無法快速啟動或啟動。

如果是排程器而不是使用者啟動了含有「必要」變數的工作，若變數目前有值，變數會維持不變，若沒有值，便保留空白。

- **唯讀**：值無法變更。
- **抑制顯示**：變數不會顯示在「啟動工作」畫面上。不過，變數仍在，並且可在步驟中使用。
- **必須變更**：變數值必須變更。在「啟動」畫面中會強調顯示具有這個內容的最上層變數；如果沒有輸入新的值，就無法快速啟動或啟動工作。

註：如果含有「必須變更」變數的工作是由排程器啟動，而非由使用者啟動，則變數值會維持不變。如果含有「必須變更」變數的工作是由類別內容啟動，變數值也會維持不變。類別內容包含「清除時啟動」、「進入時啟動」以及「結束時啟動」。

Snapshot 標籤

環境 Snapshot 是環境的實例。Snapshot 標籤提供了 Snapshot 的名稱。請使用這個標籤來檢視或變更 Snapshot 名稱以及 Snapshot 的相關註解。

建立環境

程序

1. 在左功能表中，按一下**環境**。
2. 按一下**新增環境**。
3. 指定環境的名稱。
4. 指定其成員應該能夠使用環境的存取群組。
5. 按一下**儲存**。此時會呈現變數畫面，供您新增變數至環境。

使用變數

以下各節說明使用變數來完成一般作業的程序。

建立變數的下拉清單

關於這項作業

您可以為變數定義多個可能的值。您提供的值，會在下拉清單中顯示成可供選取的選項。


如果要建立環境變數的下拉清單，請執行下列動作：

1. 選取**專案 > 環境**。
2. 在清單中選取環境。
3. 按一下**新增環境變數**。依下列方式填寫其內容。
 - 名稱：輸入環境變數的名稱。
 - 類型：選取**下拉清單**。
 - 動作：選取一個動作。
 - 於專案：選取一個內容。
4. 按一下**儲存**。
5. 按一下**變數**。
6. 按一下**下拉選項**標籤。
7. 依下列方式新增下拉清單的值：
 - a. 指定下拉選項的名稱。在下拉清單中，使用者會看到此項顯示成選項名稱。下拉清單名稱只能使用英數字元（a-z、A-Z、0-9）和底線字元（_）。長度上限為 255 個位元組：
 - 單位元組字集：255 個字元
 - 雙位元組字集：127 個字元
 - b. 指定下拉選項的值。如果您想讓使用者看到正在使用的值，可設定一個和變數名稱相同的值。如果值中使用了變數，則會在執行步驟的作業系統上解譯它。不過，在步驟執行之前，並不會預先處理它或對它求值。
 - c. 按一下**建立**。此時會將選項新增至清單中。
 - d. 針對每一個所要的值，重複這項程序。
8. 按一下**儲存變數**。
9. 設定要呈現的預設選項。您必須先在下拉選項中移入資料，儲存變數，然後再執行這個動作。按一下**環境**。在**詳細資料**標籤中，將**預設選項**內容從 **--NONE--** 變更為您想要預設呈現之值的名稱。

請注意，當檢視變數的**詳細資料**標籤時，如果未設定**預設選項**（它的值是 **--NONE--**），依預設，**值**欄會呈現**下拉選項**清單中的第一個項目。

10. 按一下**儲存變數**。

您可以依下列方式進一步處理選項：

- 使用每一個選項名稱左邊的**編輯**圖示，來設定選項在清單中的位置，或將它們刪除。
- 按一下選項以編輯它。您可以同時編輯**名稱**和**值**欄位。完成時請按一下**儲存**。
- 按一下**清除**，以清除**名稱**和**值**欄位。一般而言，您是在檢視現有選項後才執行此動作，藉此建立新選項。

併入其他環境

您可以使用「併入」變數類型，以併入另一個現有環境中的所有變數。

1. 選取**專案 > 環境**。
2. 建立新環境，然後按一下**儲存**。
3. 按一下環境名稱。
4. 按一下**新增環境變數**。依下列方式填寫其內容。
 - 類型：選取**併入**。使用者介面會變更為顯示**併入環境**下拉清單。其中移除了「動作」和「於專案」內容。
 - 併入環境：從清單中選取要併入的環境。
5. 按一下**儲存變數**。

在步驟執行期間變更變數值

在配合步驟、專案或永久範圍執行期間，可以變更變數。

- 步驟範圍：當在步驟中使用指令時，可利用明確的指派來置換變數值。這些值僅在現行步驟期間才會生效。
- 專案範圍：當在步驟中使用 `.bset` 指令時，可變更執行中工作範圍的變數值。您可以使用 `.bset` 來建立新變數。其範圍以工作的其餘部分為限。使用 `.bset` 所做的變更，會在使用 `.bset` 之步驟後面的步驟中生效。
- 永久範圍：當在步驟中使用 `.set` 指令時，可變更環境變數定義。變數定義在伺服器環境、專案環境和步驟環境中。當使用 `.set` 來變更伺服器變數或專案變數時，並不會變更現行工作的變數副本。在現行工作之後執行的工作會取得已變更的變數。不過，如果 `.set` 指令變更了步驟環境變數，後續使用該步驟環境的步驟會取得已變更的變數。`.set` 指令無法建立新變數。

比方說，如果您利用名稱為 `Java` 的專案環境（包含變數 `JavaVersion = 1.4`）來啟動專案，且您使用 `.bset` 將值變更為 `1.5`，任何繼承這個專案環境的步驟都會取得 `1.5` 這個值，而任何參照 `Java` 環境的步驟則特別取得原始值 `1.4`。

請注意，當系統啟動工作時，會將專案環境變數複製到該工作的資料庫記錄，並從取得專案預設值時開始參照這個工作環境。如果使用者在啟動工作時，修改了任何專案變數的起始值，則會將這些值記錄在工作記錄中。

對映 Windows 磁碟機

Microsoft Windows[®] 作業系統會以不同的方式來管理對映磁碟機。代理程式會針對使用者帳戶，重新對映它所記得的連線，不過不見得能夠在執行時期順利完成對映。您可以使用特殊的環境變數，在 Windows 上協助進行磁碟機對映：`_MAP` 變數。當您設定這個變數時，Windows 代理程式會先對映磁碟機，再執行您的步驟。

在使用 `_MAP` 變數時，一般做法是在專案環境中指派它，以便透過環境變數的繼承，將相同的磁碟機對映傳遞給所有的步驟環境。請注意，如果您也在步驟環境中定義 `_MAP` 變數，則步驟環境的值會置換專案環境，這是因為您只能為特定的步驟定義一個 `_MAP` 值。

雖然這原本就是針對 Windows 環境而設，不過還是請使用正斜線來區隔 `_MAP` 變數中的目錄路徑名稱。當您使用路徑時，代理程式會自動配合需求更正它們。

例如，將 `_MAP` 設為

```
X:>//server/share
```

，會定義執行時期對映，將 X: 磁碟連接到 Windows UNC 路徑名稱 \\server\\share。

您可以在 _MAP 變數中另外提供其他對映規格，以分號加以區隔，來對映多個磁碟機：

```
X:>//server/share;Y:>//server/share2
```

依預設，Windows 使用和邏輯伺服器相同的使用者名稱與密碼，來執行磁碟機對映。您可以依照下列範例所示，在對映之後，新增用括弧括住的使用者名稱和密碼，以對映不同使用者名稱的磁碟機：

```
X:>//server/share(alternateusername,password)
```

註：如果您的密碼包含 \$ 字元，請使用另一個 \$ 字元讓它跳出。例如：將 pas\$word 密碼輸入為 pas\$\$word。請避免在密碼中使用下列的特殊字元：%、[、]、{、}、" 或 '。

指令一旦完成，透過 _MAP 變數對映的磁碟機就會被取消對映。

即使磁碟機順利對映，但如果登入系統主控台的使用者使用的是有問題的磁碟機或共用區，則可能仍然無法存取 Windows 上的磁碟機對映。

使用下一個可用的磁碟機代號，來對映 Windows 磁碟機

您可以讓系統挑選下一個可用的磁碟機代號。請使用下列語法：

```
<driveletter>?=//<directory path>
```

例如，您可以依下列方式設定 _MAP：

```
X?="//server/share
```

在此情況下，系統不會將磁碟機對映至 X。相反地，它會對映至下一個可用的磁碟機，並將它選取的磁碟機代號儲存在一個名叫 _MAP_X 的變數中。如果選取的磁碟機是 F，_MAP_X 變數的值為 『F:』。您可以利用變數來存取此對映關係。

您可以依下列方式使用多項對映：

```
X?="//server/share; Y?="//server/public
```

您可以使用所要的任何代號，甚至可以使用多項對映，如下列範例所示：

- F: 對映至 //server/share
- G: 對映至 //server/public

此範例亦建立下列變數：

- _MAP_X，含有值 『F:』
- _MAP_Y，含有值 『G:』

註：當您設為目標的 Windows[®] 系統使用 Cygwin 時，如果您使用下一個可用的磁碟機語法，必須以反斜線跳出問號，如下所示：

```
Y\?="//server/share
```

代理程式型磁碟機對映

您可以在代理程式中使用配置參數來對映磁碟機。將 *map* 參數新增至 *BFAgent.conf* 檔時，請使用和 *_MAP* 變數相同的語法。您可以使用這個參數來建立特定伺服器的磁碟機對映。如果您也使用 *_MAP* 變數，其對映會置換代理程式的對映。

在變數中使用點指令

某些點指令可用來作為環境變數的名稱或值。

使用 **.source** 讓 **Script** 在指令之前執行

系統可讓您定義名為 *.source* 的特殊環境變數，以在系統執行指令之前，先在伺服器上執行 *Script*。這可讓您從伺服器上的原始檔載入一組環境變數，或執行自訂準備指令。

如果要試用這項特性，請執行下列動作：

1. 在系統上建立一個稱為 *mybatch.bat* 的批次檔，以回應某個句子。將批次檔儲存至 *C:\temp*。
2. 建立一個名為 *Step Variables* 的新環境。
3. 新增一個名為 *.source* 且值為 *C:\temp\mybatch.bat* 的變數。
4. 將步驟的環境設為新建的 *Step Variables* 環境。
5. 執行專案並檢查該步驟的日誌輸出。

請注意其他的日誌資料，其會顯示在進行此步驟指令之前已執行過 *mybatch.bat* 檔。以下是關於 *.source* 的一些重要注意事項：

- 指定的路徑不能含有引數。
- 在 Windows[®] 平台上，*Script* 透過呼叫來啟動。
- 在 UNIX[®] 平台上，*Script* 必須放在原生的 *Shell* 語法之內，因為它源自於執行中的 *Shell*。

使用 **.date** 指令來儲存變數中的日期或時間

您可以使用 *.date* 環境點指令來提供變數的現行日期或時間。使用指令作為變數的值。當您執行會使用變數的專案時，系統就會以 *.date* 指令的結果來更新該變數。

例如，一個含有 *.date %B* 值的變數 *MONTH* 若併入專案中，在五月時就會以『May』值提供給工作。

如需使用此指令的相關資訊（包含日期格式程式碼清單），請參閱 第 347 頁的『*.date*』。

系統變數參照

系統定義的變數可以用在變數中。

系統會在工作的每一個步驟中，自動設定下列變數的值。這些變數是唯讀的。其工作值會列在步驟日誌的 *ENV* 字行中。前四個是專案層次的通知。其他所有 *BF_* 變數是在步驟層次使用。

專案層次的變數	值
BF_D	日期。可用於標籤中。格式由標籤決定：日期格式系統設定。
BF_J	一年中的第幾天。可用於標籤中。
BF_T	時間。可用於標籤中。格式是由標籤決定：時間格式系統設定。
BF_W	星期幾，以 0（星期日）到 6（星期六）的一個數字表示。

步驟層次的變數	值
BF_D	日期。可用於標籤中。格式由標籤決定：日期格式系統設定。
BF_J	一年中的第幾天。可用於標籤中。
BF_T	時間。可用於標籤中。格式是由標籤決定：時間格式系統設定。
BF_W	星期幾，以 0（星期日）到 6（星期六）的一個數字表示。
BF_AGENT_PLATFORM	這個字串識別正在執行代理程式的作業系統平台。
BF_AGENT_VERSION	現行伺服器的代理程式版本號碼。
B	預設標籤變數，從 1 開始，依每個工作依序增量。可用在以 BF_TAG 表示的標籤中。
BF_BID	工作 ID 號碼，對於相同專案的工作來說是唯一的。
BF_CALLER_	從呼叫端專案傳遞至鏈結的專案的變數所要套用的字首。
BF_CLASS	專案的 Build Forge 類別
BF_ENGINE	用來唯一識別引擎的字串。此值也會儲存在安裝目錄中的下列檔案內：engine.id。例如：D8531015-6C07-1014-8CA0-BD58317220B3。
BF_HOST	邏輯伺服器的主機名稱（TCP/IP 主機名稱）。（這個變數是伺服器環境的一部分。）
BF_ITERATION	步驟在「While 迴圈」中已順利啟動的次數。當步驟的「條件」求值為 true 時，此值會增量。工作重新啟動作業會使用這個變數的值，作為要重新啟動的反覆運算。
BF_ITERATION_MAX	「While 迴圈」所能執行的次數上限。這是在步驟內容中設定的。
BF_ITERATION_MAX_REACHED	依預設，不會建立或設定。當 while 迴圈的反覆運算次數達到 BF_ITERATION_MAX 時，會將「While 迴圈」步驟的步驟唯一 ID (BF_SSID) 附加至這個變數。如果專案中有多個「While 迴圈」步驟達到其 BF_ITERATION_MAX，這個變數會包含多個值，每一個各代表一個達到反覆運算數上限的步驟。

步驟層次的變數	值
BF_LASTGOODRUN	同一專案最近一項通過工作的日期；如果沒有通過的工作，則為前一個工作的日期。
BF_LASTGOODTAG	最近一項通過工作的標籤（如果同一個專案沒有儲存通過的工作，則為前一個工作）。
BF_LASTGOODUNIX	與 BF_LASTGOODRUN 相同，但它是 UNIX [®] 格式來表示日期。
BF_LASTRUN	現行工作前次執行的日期。
BF_LASTTAG	相同專案之前一項工作的標籤字串。
BF_LASTUNIX	與 BF_LASTRUN 相同，但它是 UNIX 格式來表示日期。
BF_ONFAIL	步驟的「中止/繼續」旗標。
BF_PID	專案 ID 號碼。
BF_PROJECTNAME	工作的專案名稱。
BF_PROJECTNAME_PHYS	用來建立專案目錄的專案名稱。系統建立專案目錄時，會將 無效的相對目錄字元 系統設定中所指定的字元變更為底線字元。比方說，如果設定包含一個空格，則名為 My Project 的專案，其專案目錄名稱就是 My_Project。
BF_ROOT	工作的基本工作目錄，取自工作內容。另請參閱 BF_STEP_ROOT。
BF_SERVER	正在執行現行工作的伺服器名稱（這個變數是伺服器環境的一部分）。
BF_SERVER_ROOT	指派給伺服器內容中之邏輯伺服器的路徑（這個變數是伺服器環境的一部分）。
BF_SID	建置內的步驟結果的序號。
BF_SPID	若有另一項工作已呼叫現行工作，則會包含呼叫端專案 ID。如果沒有值，則和 BF_PID 相同。
BF_STEPNAME	步驟名稱。設定在步驟內容中。
BF_STEP_ROOT	步驟的基本工作目錄，取自步驟內容。另請參閱 BF_ROOT。
BF_SSID	步驟 ID，為現行步驟在專案中的唯一 ID。
BF_TAG	工作的標籤。標籤定義可以包含變數。這個變數含有在工作啟動時因解譯這些變數而產生的值。
BF_TAG_PHYS	工作的標籤，會以底線取代出現在 BF_TAG 值中的任何空格。如果步驟選取了「絕對」選項，則 BF_TAG_PHYS 等同於 BF_TAG。
BF_USER	工作擁有者的使用者名稱。

觸發變數參照

系統會監視下列的變數名稱。當步驟的環境包含其中一個變數時（不論是具體的或是繼承自專案或伺服器），就會執行動作。

變數	內容
_CI_BUILD_DELETE	將此變數設為任何值，以便在工作執行之後，刪除該建置及相關聯的建置資料。（如果沒有執行其他專案建置，標籤變數便會重設為刪除建置之前的起始值）。
_CI_BUILD_KEEP	將此變數設為任何值，在工作執行之後保留該建置及相關建置資料。例如，如果您的工作包含配接器鏈結，當配接器步驟失敗時，並不會執行其他專案步驟。舉個例說，這時候您可能希望保留該工作的一份建置記錄副本，作為除錯用。
CLEARCASE_VIEW	啟動指定的 ClearCase 視圖。這個變數中指定的視圖必須存在，而且使用這個變數的步驟必須設為 "absolute"。在執行 Microsoft Windows 的系統中，這個變數必須搭配 bfagent.conf 中代理程式的 cc_suppress_server_root 配置選項來使用。
_CLEARCASE_VIEWS	指定指令執行之前要啟動的 ClearCase 視圖清單。將此值設為以逗點區隔的視圖清單；例如，『View1,View2,View3』。
_CLEARCASE_VOBS	指定指令執行之前要裝載的 ClearCase VOB 清單。將此值設為以逗點區隔的 VOB 清單；例如，『\Vob1,\Vob2,\Vob3』。
_CONTEXT_LOG_RANGE	使用此變數將日誌輸出限制為接近過濾相符項的行數。它接受正整數值，而且會使系統省略除了下列項目以外的日誌輸出：命中的每一個過濾字串附近的行號，其大小相當於此變數的值。例如，如果您將變數設為 5，則日誌只會顯示含有過濾相符項的字行，以及這些相符項的前面 5 行和後面 5 行。 註： 這個變數僅用於 Perl 引擎專案。

變數	內容
<code>_ERROR_THRESHOLD</code>	<p>建立接受的錯誤數上限（您定義的「設定失敗」過濾器所捕捉的錯誤）。您可以利用這個變數，來建立個別步驟或一個專案的失敗和訊息臨界值。</p> <p>請使用下列其中一種格式：</p> <ul style="list-style-type: none"> • 5 或 F5 值表示發生超過 5 個錯誤時，工作便應失敗。 • N7 值表示在發生超過 7 個錯誤時，系統應該在工作附註中新增一則訊息。訊息表示已達到這個臨界值。 <p>當您在步驟中使用變數時，系統會在個別步驟中計算錯誤。其他可用的格式有：</p> <ul style="list-style-type: none"> • W9 之類的值指出在 9 次錯誤之後，該步驟都會處於警告狀態，不論過濾器未來是否捕捉到錯誤。 • C8 之類的值指出在 8 次錯誤之後，會將步驟設為失敗狀態，但是 Clear Fail 過濾器可清除失敗。 <p>附註：此變數計算的錯誤會定義為利用「設定失敗」動作符合過濾器的字串，且它們已指派給專案中的步驟。過濾器已識別為失敗的每一個字串，將在步驟總計中算一個錯誤，也在專案總計中算一個錯誤。</p>
<code>_EXITCODE_MAP</code>	<p>指定系統應接受為步驟成功指標的數字清單（以逗點、空格、分號或冒號區隔）。依預設，結束碼 0 表示成功；指定此變數之後，它所列出的任何值也表示成功。</p>

變數	內容
_InterfaceLoggingLevel	<p>控制當 Build Forge 執行配接器步驟時，要記載多少日誌資料。請在您的配接器環境中建立一個環境變數，將它命名為 _InterfaceLoggingLevel。然後指派 0 到 8 的一個整數值給它。記載層次也包含在內，例如，層次 2 包含層次 1 和層次 0 的資訊。</p> <ul style="list-style-type: none"> • 0：執行程式行號及伺服器連線錯誤或取消通知；僅此而已 • 1：剖析的指令（會傳送至伺服器的指令） • 2：未剖析的指令（在設定其區域變數之前的指令） • 3：建置和環境變數 SET 行號 • 4：臨時及內部變數 SET 行號 • 5：環境求值、電子郵件群組新增、BOM 文字記載行 • 6：區塊及子區塊開始/結束行號 • 7：（預設日誌記載層次）要根據比對型樣來檢查的代理程式輸出，以及符合型樣的字行。 • 8：所有代理程式輸出
_LOG	<p>指定路徑名稱以建立包含 Build Forge Agent 原始輸出的日誌檔。</p> <p>註：這個日誌不包含時間戳記（除非您還指定了 _LOG_TIMESTAMP）。這個檔案中的日誌資料格式通常如下所示：代理程式碼、日誌儲存區和訊息。</p> <p>請使用此變數，在伺服器上儲存工作日誌的副本。如果該檔案存在，系統會附加在它後面。</p>
_LOG_TIMESTAMP	<p>以時間戳記作為從 _LOG 輸出的每一行的字首。這個變數的值應該是採用 strftime 語法的格式字串，此語法與 .date 和 .gmdate 環境指令所使用的語法相同。</p> <p>註：需要有 _LOG。</p>
_MAP	<p>如需如何使用此變數的一些討論，請參閱第 271 頁的『對映 Windows 磁碟機』。</p>
_NO_PREPARSE_COMMAND	<p>系統通常會在傳送指令至代理程式之前試圖解析環境變數的值。當您定義 _NO_PREPARSE_COMMAND 變數（及任何值）之後，系統會將變數傳送至代理程式，而不解析它們。使用此變數可確保作業系統 Shell 會處理變數。</p>

變數	內容
_PRISM_DIR_POSTCMD	搭配使用 IDE 的外掛程式。它可以指定一個指令，在執行專案步驟之後，對目錄執行此指令。請參閱第 485 頁的『測試專案的特殊變數』。
_PRISM_DIR_PRECMD	搭配使用 IDE 的外掛程式。它可以指定一個指令，在目錄複製到專案步驟的伺服器之前，對這些目錄執行這個指令。請參閱第 485 頁的『測試專案的特殊變數』。
_PRISM_FILE_POSTCMD	搭配使用 IDE 的外掛程式。它可以指定一個指令，在執行專案步驟之後，對檔案執行此指令。請參閱第 485 頁的『測試專案的特殊變數』。
_PRISM_FILE_PRECMD	搭配使用 IDE 的外掛程式。它可以指定一個指令，在檔案複製到專案步驟的伺服器之前，對這些檔案執行這個指令。請參閱第 485 頁的『測試專案的特殊變數』。
_SUPPRESS_ENV_OUTPUT	指定系統會省略來自日誌的環境訊息。依預設不會設定此變數，而且會先列印環境中的所有變數值，之後再執行步驟指令。這些值在步驟日誌中會顯示為 ENV 項目。這個變數可以設為下列值： <ul style="list-style-type: none"> • ALWAYS：一律省略 ENV 訊息 • 任何其他值：省略 ENV 訊息。不過，如果指令失敗，則會在指令訊息之後列印 ENV 訊息。在對失敗的指令執行進行除錯時，這項資訊可能會很有用。
_SUPPRESS_AGENT_LOG_OUTPUT	設為 1 時，可以防止代理程式將日誌資料傳送至引擎。與 _SUPPRESS_LOG_OUTPUT 相比，其中的日誌資料是從代理程式傳送，但被引擎捨棄。 註： 使用這個變數會時，不會進行過濾器比對。
_SUPPRESS_LOG_OUTPUT	將此變數設為任何值會導致引擎捨棄幾乎所有從代理程式收到的日誌輸出。仍會保留部分主控台記錄。會顯示過濾相符項。
_TIMEOUT	這個值會置換專案中的一個步驟或所有步驟的 Timeout 內容。
_TRAP	現行步驟失敗時執行的字串；此字串可設為執行檔或指令的名稱。 附註： 指令輸出不會傳回至主控台，因為當步驟失敗時，主控台和代理程式之間的連線會關閉；如果您想要保留透過 _TRAP 發出的指令輸出，請讓該指令將其輸出寫入檔案以供稍後擷取。

變數	內容
_USE_BFCREDS	<p>若設為 1，系統會使用使用者的 登入認證來登入伺服器，而不是儲存在附加至伺服器的伺服器權限中的認證。系統會使用啟動專案的使用者的「管理主控台」登入認證，來執行專案中的指令。您可以對單一步驟設定此變數，或對整個專案設定。</p> <p>註：如果您使用 LDAP/Active Directory 鑑別，則必須將在本端儲存使用者鑑別系統設定設為「是」（預設值），_USE_BFCREDS 函數才能運作。若設為「是」，則系統會快取加密格式的使用者鑑別資訊，然後存取要使用於 _USE_BFCREDS 的使用者鑑別資訊。</p> <p>提示：在 Windows 上，請考慮同時設定變數 _USE_BFCREDS_DOMAIN。</p>
_USE_BFCREDS_DOMAIN（僅適用於 Windows）	<p>若設為 1，系統除了使用 _USE_BFCREDS 用來登入伺服器的登入認證外，還會使用使用者的網域。</p>
_XSTREAM_PROTOCOL <i>type</i>	<p>在代理程式之間啟用直接檔案傳送。</p> <p>重要：在某些作業系統上，代理程式對於直接檔案傳輸的支援有限，甚至不支援。請參閱第 158 頁的『配置代理程式之間的直接檔案傳輸』。</p> <p>引擎、傳送代理程式及接收代理程式必須全都支援直接檔案傳送。如果有任一個不支援，則會忽略 _XSTREAM_PROTOCOL 且沒有任何警告，然後使用正常的檔案傳送方法。</p> <p>接收代理程式必須能夠在傳送代理程式主機上建立 TCP 連線。如果它們存在，則必須將防火牆配置為容許連線。</p> <p>通訊協定類型能決定編碼資料的方法，並且為下列其中一項：</p> <p>AES-CBC</p> <p>強型加密演算法是用來編碼資料。這兩種代理程式都必須利用 OpenSSL 來編譯，而且與引擎通訊時必須使用 SSL。加密金鑰是從引擎取得。</p> <p>PRNG 虛擬亂數產生器是用來遮蔽檔案內容。</p> <p>PLAIN 檔案是依現狀傳送而不加以編碼。</p>

環境 Snapshot

建立環境的 Snapshot，可針對您想變更或修改的環境，快速建立新實例。

「環境 Snapshot」概觀

請檢閱這些主題，以瞭解環境 Snapshot 及其用法。

環境 Snapshot 使用案例

下列範例說明環境 Snapshot 的一些一般使用案例：

- 建立環境的 Snapshot，以變更環境的配置，或執行新工具或 Script 的測試。
- 將環境的 Snapshot 儲存成暫時備份或正式保存的一部分。
- 建立環境的 Snapshot，以擷取和某個里程碑（例如：外部或內部發行）相對應的某個時間點環境配置。

環境 Snapshot 的概念和詞彙

在使用者介面中，Snapshot 引進某些配合環境使用的新概念和詞彙。

環境 Snapshot：Snapshot 是現有環境的新實例。關於 Snapshot，需要切記的一些關鍵點如下所示：

- Snapshot 是個別的環境物件。變更 Snapshot 集當中的某個 Snapshot，並不會影響集中的其他 Snapshot。
- Snapshot 不是副本。

如果您針對環境的相關聯物件建立 Snapshot，Snapshot 會建立物件的另一個實例。副本會對映物件之間的關係；它不會建立新物件。

- Snapshot 不是環境的修訂：
 - Snapshot 不支援比較兩個環境 Snapshot 間的變更。
 - 不同於來源控制系統，環境 Snapshot 的變更並不會以版本號碼來追蹤或識別。不過，您可以使用包含版本號碼（例如，7.5.0、3.4.01）的 Snapshot 命名方法，讓環境 Snapshot 和里程碑產生關聯。

Snapshot 集：Snapshot 集是一組作為一個基礎 Snapshot 之後代的所有環境 Snapshot。此集中至少含有一個基礎或母項 Snapshot，以及一個子項 Snapshot。在使用者

介面中，環境名稱旁的 Snapshot 圖示  代表環境現存的 Snapshot 集。

基礎 Snapshot：一開始，所有環境的 Snapshot 名稱皆為「基礎 Snapshot」。您可以將「基礎 Snapshot」變更為另一名稱。基礎 Snapshot 是 Snapshot 集的母項。

預設環境 Snapshot：預設環境 Snapshot 是現行的工作環境。Snapshot 集中只能有一個 Snapshot 作為預設值。如果您沒有指定預設 Snapshot，就會以基礎 Snapshot 作為預設值。

- 在使用者介面中，預設 Snapshot 會顯示在環境清單的最上層。選取**環境**，以顯示環境清單。
- 在您選取具有 Snapshot 的環境時，除非您在清單框中選取不同的環境 Snapshot，否則會使用預設環境 Snapshot。
- 若要存取及處理環境 Snapshot 集中的其他 Snapshot，您必須按一下 Snapshot 圖示



環境 Snapshot 視圖

選取 Snapshot 圖示 ，以顯示 Snapshot 視圖。在使用者介面中，Snapshot 視圖會顯示 Snapshot 在集合中的階層：

- 基礎 Snapshot 位於最上層，如果您沒有另外指派一個唯一名稱，其名稱會是「基礎 Snapshot」。
- 所有的環境 Snapshot 皆為基礎 Snapshot 的子項。相同基礎 Snapshot 的子項在 Snapshot 直欄中全縮排在同一層。
- 從子項 Snapshot 建立而來的環境 Snapshot，會成為該子項 Snapshot 的子項，並在 Snapshot 直欄中縮排在下一層。

環境 Snapshot 規劃

檢閱某些最佳實務，以選取預設環境 Snapshot，並為環境 Snapshot 命名。

• 用來在集合中選取預設 Snapshot 的策略

使用者介面只能辨識 Snapshot 集的一個預設或現行環境 Snapshot。在選取預設 Snapshot 時，請採取一致的策略：

- 使用基礎 Snapshot 作為預設 Snapshot

採取這種策略時，是將 Snapshot 建立成時間點備份，且不會對所備份的環境 Snapshot 進行任何變更。您要變更的是基礎 Snapshot。

- 使用最新 Snapshot 作為預設 Snapshot

如果採取這種策略，在您建立新環境時，會認定您有意讓它成為新的預設環境 Snapshot。您不會變更基礎 Snapshot 或更早的環境 Snapshot。

• 識別 Snapshot 集的 Snapshot 命名方法

環境 Snapshot 名稱在環境 Snapshot 集當中必須是唯一的。

請使用下列一種準則，來協助您建立環境 Snapshot 名稱：

- 名稱應具備敘述性：應指出 Snapshot 的用法或用途。
- 命名方法應遵循定義的標準。您可以使用 Snapshot 標籤上的「註解」框，來說明命名方法。

• 讓集合使用單一環境名稱

在您建立環境 Snapshot 後，可讓您選擇變更環境的名稱。如果您變更環境名稱，會更新每一個環境 Snapshot 的這個名稱。

建立環境 Snapshot



建立環境 Snapshot 時，將會建立新的環境實例。Snapshot 並非副本；它是一個新環境。

您可以從環境或環境 Snapshot，來建立環境 Snapshot。

提示：如果要建立 Snapshot 和變更預設 Snapshot，使用者必須具備必要許可權。請參閱第 317 頁的『驗證與編輯 Snapshot 許可權的存取群組』。

1. 按一下您要建立 Snapshot 之環境或環境 Snapshot 旁的編輯圖示：

- 若要建立預設環境 Snapshot 的 Snapshot，請在環境清單（環境）中，按一下最上層 Snapshot 旁的「編輯」圖示。

- 若要建立非預設環境 Snapshot 的 Snapshot，請按一下 Snapshot 圖示。Snapshot 視圖會顯示集合中的環境 Snapshot。按一下非預設環境 Snapshot 旁的「編輯」圖示。

2. 按一下**建立新 Snapshot**。

3. 在 Snapshot 標籤中輸入**名稱**。該名稱會指派給隨環境一起建立 Snapshot 的所有物件。

此名稱在環境 Snapshot 集當中必須是唯一的。

4. 選取當您建立環境 Snapshot 時，要建立 Snapshot 的 Build Forge 物件。下表說明您所能選取的物件。

物件	說明
預設值	在使用者介面中，預設 Snapshot 會顯示在環境清單的最上層。 選取環境，以顯示環境清單。
遵循環境併入項目	使用「併入環境」變數類型，以建立環境所併入環境的 Snapshot。

5. 按一下**儲存**，以儲存環境 Snapshot。

變更預設環境 Snapshot

預設環境 Snapshot 是 Snapshot 集中的最上層 Snapshot，它會顯示在環境清單中（環境）。

提示：如果要建立 Snapshot 和變更預設 Snapshot，使用者必須具備必要許可權。請參閱第 317 頁的『驗證與編輯 Snapshot 許可權的存取群組』。

如果要變更預設環境 Snapshot，請編輯您要作為新預設值之 Snapshot 的 Snapshot 定義：

1. 選取環境。
2. 在環境清單中，按一下預設環境 Snapshot 的 **Snapshot** 圖示。
3. 在 Snapshot 清單中，按一下要作為新預設值之環境 Snapshot 的**編輯**圖示。
4. 按一下**設為預設值**。
5. **重要事項：**在蹦現式畫面中，選擇「確定」或「取消」。



確定	更新參照： 針對參照先前預設的任何物件，將先前的預設參照環境 Snapshot 參照，更新為新預設值。
取消	不更新參照： 針對參照先前預設的任何物件，不將參照更新為新預設環境 Snapshot。

變更環境的 Snapshot 名稱

您可以變更環境 Snapshot 的 Snapshot 名稱，也可以針對當初您建立環境 Snapshot 時選取要建立 Snapshot 的物件，變更其 Snapshot 名稱。

在基礎 Snapshot 方面，您可以使用這個選項，僅針對單一環境 Snapshot 或針對所有現行和未來的環境，將其「基礎 Snapshot」預設名稱變更為另一個 Snapshot 名稱。

如果要變更 Snapshot 名稱，請執行下列動作：


1. 選取環境。
2. 在環境清單中，按一下預設環境 Snapshot 的 **Snapshot** 圖示。
3. 在 Snapshot 清單中，按一下環境 Snapshot 的**編輯**圖示。
4. 選取 **Snapshot** 標籤。
5. 在**名稱**中，輸入新名稱。
6. **選用**：在**註解**中輸入一項註解。
7. **重要事項**：在蹦現式畫面中，選擇「確定」或「取消」。

確定	<p>變更環境 Snapshot 名稱和其他 Snapshot 物件名稱：針對當初您建立環境 Snapshot 時選取要建立 Snapshot 的物件，變更這些物件和環境 Snapshot 的名稱。</p> <p>在「基礎 Snapshot」方面：變更所有現行環境 Snapshot 和所有未來環境 Snapshot 的「基礎 Snapshot」名稱。</p>
取消	<p>變更環境 Snapshot 名稱，但不變更其他 Snapshot 物件名稱：在當初您建立環境 Snapshot 時選取要建立 Snapshot 的物件方面，不變更這些物件的名稱。僅變更環境 Snapshot 的名稱。</p> <p>在「基礎 Snapshot」方面：維持所有現行環境 Snapshot 和所有未來環境 Snapshot 的「基礎 Snapshot」名稱。</p>



存取和檢視 Snapshot

建立環境 Snapshot 時，會建立一個 Snapshot 集，其中至少含有下列兩個環境：基礎 Snapshot 環境和新環境 Snapshot。

如果要檢視 Snapshot 集當中的所有環境 Snapshot，請執行下列動作：

1. 選取環境。
2. 按一下 **Snapshot** 圖示，即會顯示 Snapshot 集中的環境 Snapshot。

在 Snapshot 視圖中，您可以：



- 建立新的環境 Snapshot。若要開始，請按一下**編輯**圖示。
- 變更環境的預設 Snapshot。按一下新預設 Snapshot 的**編輯**圖示，然後按一下「設成預設值」。
- 以編輯標準環境的方式，來編輯環境 Snapshot 定義。

刪除環境 Snapshot

您可以使用「刪除環境」選項，來刪除環境 Snapshot。

若有另一個物件正在使用環境，則無法刪除該環境；比方說，如果它是包含在另一個環境中，或者由某個專案、步驟、排程或伺服器使用。

如果要刪除環境 Snapshot，請執行下列動作：

1. 選取環境。
 2. 在環境清單中，按一下基礎 Snapshot 的 **Snapshot** 圖示。
- Snapshot 視圖會顯示集中的環境 Snapshot。
3. 按一下要刪除的環境 Snapshot 旁的編輯圖示。
 4. 按一下刪除環境。

第 19 章 使用專案

這個主題說明如何在「管理主控台」中建立和管理專案。

關於專案

專案是可執行的步驟集，內含步驟的環境群組及伺服器內容。

除了專案，您還可以建立程式庫。與專案不同，程式庫沒有選取元可決定輸入步驟的位置。

關於「專案」畫面

您可以使用「專案」畫面來建立新專案，並檢視或編輯現有的專案。若要存取左側功能表上的「專案」畫面，請按一下**專案**。

「專案」畫面提供下列標籤：

- 『「詳細資料」標籤』
- 第 288 頁的『「標籤」標籤』
- 第 289 頁的『「暫存器」標籤』
- 第 289 頁的『「附註」標籤』
- 第 289 頁的『Snapshot 標籤』

「詳細資料」標籤

您可以在「詳細資料」標籤上指定各種專案內容。

工作控制器

選擇適合專案引擎類型的工作控制器。選項為 **Java 主要工作控制器**或「Perl 引擎」。系統會使用 **Java 主要工作控制器**來執行主要工作控制器專案。系統只會使用 **Perl 引擎**來執行 **Perl 引擎**專案。如需引擎類型的相關資訊，請參閱第 212 頁的『系統配置設定』主題中的**使用 Java 主要工作控制器作為工作執行引擎**設定。您也可以參閱第 233 頁的『管理 Perl 引擎』小節中的主題。

名稱 專案名稱。系統使用此名稱來參照清單與資料庫中的專案。

如果您執行專案，則專案名稱是用來建構專案目錄。由於專案包含的步驟可能會在不同作業系統上執行，因此請避免在專案名稱中使用特殊字元和符號。如果專案名稱必須使用的字元會產生無效的目錄名稱，請將這些字元新增至**無效相對目錄字元系統配置設定**。在建立專案目錄時，此設定中列出的字元會轉換成底線 (_)。使用 & 符號可能會不小心在專案名稱中產生 **HTML 實體**。

存取 群組已容許檢視並使用專案。您可以使用此內容及許可權，以決定您能夠存取的內容。比方說，如果您要啟動工作，則您必須是指定給專案的存取群組成員，也必須是具有執行工作許可權的群組成員。如需存取群組的相關資訊，請參閱第 199 頁的『「存取」概觀』。

停用 請選取這個勾選框來停用專案。當您嘗試執行已停用的專案時，**Rational Build Forge** 會顯示一則訊息，指出專案不在作用中，無法執行專案。

執行緒上限

專案可以啟動之平行處理的最大數量。請使用這個欄位，以避免專案使用太多系統資源。每一個具有執行緒能力的步驟和任何列入專案（它們本身可啟動具有執行緒能力的步驟）都可能導致平行處理程序，但那些處理程序將都不會計算在母專案的上限中。當系統達到「執行緒數目上限」值時，就不會再啟動新的平行處理程序，並會等到該專案的平行處理程序數目降到「執行緒數目上限」值以下，才會繼續。如需執行緒作業的相關資訊，請參閱第 330 頁的『執行緒作業：平行執行步驟』。

執行限制

「執行限制」內容會設定允許一次執行的專案工作數目上限。

- 如果您啟動專案，而且作用中的工作數目等於「執行限制」，則新工作會停留在等待佇列中，直到至少一項工作完成為止。
- 如果排程試圖啟動專案時，作用中的工作數目等於「執行限制」，且「強迫執行限制」系統配置設定的值為「是」，則系統不會啟動新工作。如果「強迫執行限制」設為「否」，系統會忽略排程建置的「執行限制」設定。
- 透過列入啟動的專案不被視為原始專案的實例，因此不會算在「執行限制」內。

類別 每個專案都必須指派給一個類別，而此類別會將廣域內容指派給工作群組。如需相關資訊，請參閱第 304 頁的『類別』。

選取元 選擇專案的伺服器時的選取元名稱。對於專案內未指定本身之選取元的所有步驟，系統都會指定這個選取元來作為預設值。請參閱第 244 頁的『選取元』。如果未指定選取元，則會將專案新增至「程式庫」畫面，而非「專案」畫面。程式庫會使用呼叫端步驟的選取元。如果步驟沒有選取元，程式庫會使用呼叫端專案的選取元。

通過鏈結、失敗鏈結

選取在專案建置通過或失敗時執行的專案。在專案層次設定通過或失敗鏈結，可讓您根據專案的通過或失敗狀態，呼叫個別的通過或失敗動作。這項功能類似於在專案內的步驟層次中，設定通過或失敗動作。在專案層次中，通過或失敗動作是由專案執行狀態（而非步驟狀態）來觸發。

環境 在伺服器環境之後和步驟環境之前套用的環境。如需這些環境共同運作之方式的相關資訊，請參閱第 265 頁的『關於變數』。

固定 啟用「固定」勾選框，可將使用預設專案選取元的所有專案步驟，都強制停留在相同的伺服器上。如果伺服器忙碌中，則您必須等到伺服器可供使用為止。如需這個選項的相關資訊，請參閱第 292 頁的『將步驟固定在伺服器中』。

預設步驟提供者

實作步驟指令的執行。在端點代理程式伺服器上執行指令文字時，此步驟提供者會提供符合預期的舊式行為。唯一的預設步驟提供者是 Java 主要工作控制器。Perl 引擎專案沒有可用的步驟提供者。

啟動通知、通過通知、失敗通知

在其中一個或全部欄位中選取存取群組，以便使用這些欄位，指示系統在專案啟動、通過或失敗時，傳送通知電子郵件。

「標籤」標籤

使用「標籤」標籤來管理專案的建置標籤。

標籤格式

爲一字串，使用純文字和標籤變數參照來定義專案的標籤。如需標籤格式的相關資訊，請參閱第 338 頁的『在工作期間變更建置標籤』。

標籤同步化

同步化兩個專案的標籤變數。選取您要將哪個專案的標籤變數，與現行專案進行同步化。兩個專案同步化時，會從相同的儲存區中選取變數。依序執行兩個專案時，起始專案的值爲 1，下一個專案的值則爲 2。如需相關資訊，請參閱第 295 頁的『同步化標籤』。

標籤名稱

變數名稱。當您以標籤格式來使用標籤變數時，請使用 `$<標籤名稱>` 格式來參照其名稱。例如，若要建立使用 `MainVer` 和 `B` 變數的標籤，請使用標籤格式 `Build_$(MainVer).$B`。例如，會擷取標籤 `Build_005.1`。

起始值 設定標籤變數的值。如果您不使用**自動增量** 選項，變數會保留這個值，直到您加以變更爲止。

填補 如果您選取**填補**值而非**無**，則當變數值用於標籤時，系統會在值之前補上零。這是爲了確保位數等於**填補**值。比方說，如果現行變數是 2，而其「填補」爲 3，則系統會以 002 來呈現這個值。填補範圍是從 1 到 8。

自動增量

如果設爲是，則系統會針對每個專案工作，將變數值以 1 爲單位遞增。

「暫存器」標籤

使用專案暫存器可儲存持續跨建置的資訊。

暫存器 專案暫存器的名稱。

內容 暫存器的值。

如需相關資訊，請參閱第 340 頁的『專案暫存器』。

「附註」標籤

使用「附註」標籤可儲存專案的相關項目。

對於每一個附註，此標籤會顯示下列資訊：


- 使用者建立附註的日期和時間
- 輸入附註的使用者
- 附註

Snapshot 標籤

專案 Snapshot 是專案的實例。Snapshot 標籤提供了 Snapshot 的名稱。

使用這個標籤可檢視、變更及新增註解至 Snapshot 名稱。

變更專案內容

若要變更專案層次內容，請選取**專案**，然後按一下所需專案名稱旁邊的**編輯**圖示 。

專案名稱

專案名稱。系統使用此名稱來參照清單與資料庫中的專案。

如果您執行專案，則專案名稱是用來建構專案目錄。由於專案包含的步驟可能會在不同作業系統上執行，因此請避免在專案名稱中使用特殊字元和符號。如果專案名稱必須使用的字元會產生無效的目錄名稱，請將這些字元新增至無效相對目錄字元系統配置設定。在建立專案目錄時，此設定中列出的字元會轉換成底線 (_)。使用 & 符號可能會不小心在專案名稱中產生 HTML 實體。

存取 群組已容許檢視並使用專案。您可以使用此內容及許可權，以決定您能夠存取的內容。比方說，如果您要啟動工作，則您必須是指定給專案的存取群組成員，也必須是具有執行工作許可權的群組成員。如需存取群組的相關資訊，請參閱第 199 頁的『「存取」概觀』。

標籤格式

為一字串，使用純文字和標籤變數參照來定義專案的標籤。如需標籤格式的相關資訊，請參閱第 338 頁的『在工作期間變更建置標籤』。

標籤同步化

同步化兩個專案的標籤變數。選取您要將哪個專案的標籤變數，與現行專案進行同步化。兩個專案同步化之後，就會從相同的儲存區提取其變數，因此當它們依序執行時，其中一個專案會取得值 1，下一個專案則會取得值 2，依此類推。如需相關資訊，請參閱第 295 頁的『同步化標籤』。

執行緒上限

專案可以啟動之平行處理的最大數量。請使用這個欄位，以避免專案使用太多系統資源。每一個具有執行緒能力的步驟和任何列入專案（它們本身可啟動具有執行緒能力的步驟）都可能導致平行處理程序，但那些處理程序將都不會計算在母專案的上限中。當系統達到「執行緒數目上限」值時，就不會再啟動新的平行處理程序，並會等到該專案的平行處理程序數目降到「執行緒數目上限」值以下，才會繼續。如需執行緒作業的相關資訊，請參閱第 330 頁的『執行緒作業：平行執行步驟』。

執行限制

「執行限制」內容會設定允許一次執行的專案工作數目上限。

- 如果您啟動專案，而且作用中的工作數目等於「執行限制」，則新工作會停留在等待佇列中，直到至少一項工作完成為止。
- 如果排程試圖啟動專案時，作用中的工作數目等於「執行限制」，且「強迫執行限制」系統配置設定的值為「是」，則系統不會啟動新工作。如果「強迫執行限制」設為「否」，系統會忽略排程建置的「執行限制」設定。
- 透過列入啟動的專案不被視為原始專案的實例，因此不會算在「執行限制」內。

類別 每個專案都必須指派給一個類別，而此類別則會將廣域內容指派給專案群組。如需相關資訊，請參閱第 304 頁的『類別』。

選取元 選擇專案的伺服器時的選取元名稱。對於專案內未指定本身之選取元的所有步驟，系統都會指定這個選取元來作為預設值。請參閱第 244 頁的『選取元』。如果未指定選取元，則會將專案新增至「程式庫」畫面，而非「專案」畫面。程式庫會使用呼叫端步驟的選取元。如果步驟沒有選取元，程式庫會使用呼叫端專案的選取元。

通過鏈結、失敗鏈結

選取在專案建置通過或失敗時執行的專案。在專案層次設定通過或失敗鏈結，可讓您根據專案的通過或失敗狀態，呼叫個別的通過或失敗動作。這項功能類

似於在專案內的步驟層次中，設定通過或失敗動作。在專案層次中，通過或失敗動作是由專案執行狀態（而非步驟狀態）來觸發。

環境 在伺服器環境之後，以及在步驟環境之前套用環境。如需這些環境共同運作之方式的相關資訊，請參閱第 265 頁的『關於變數』。

固定 啟用「固定」勾選框，可將使用預設專案選取元的所有專案步驟，都強制停留在相同的伺服器上。如果伺服器忙碌中，則您必須等到伺服器可供使用為止。如需這個選項的相關資訊，請參閱第 292 頁的『將步驟固定在伺服器中』。


啓動通知、通過通知、失敗通知

在其中一個或全部欄位中選取存取群組，以便使用這些欄位，指示系統在專案啓動、通過或失敗時，傳送通知電子郵件。

複製專案

這個主題說明如何複製現有的專案。

註：您無法將 Java 主要工作控制器專案複製到 Perl 專案。

1. 選取**專案**。
2. 選取您要複製的專案旁邊的**編輯**圖示 。
3. 按一下**複製專案**。這時會顯示「選取引擎」對話框。
4. 從下拉清單中，針對您要複製的專案選取引擎類型，然後按一下**確定**。

當您複製專案時，系統會將下列參照從現有專案複製到新專案。如果來源專案有鏈結的專案，或來源專案包含與其他專案鏈結或列入的步驟，則會顯示警告訊息。這個訊息提醒您要複製鏈結或列入的專案，才能順利建置專案。

- 步驟以及在步驟**詳細資料**標籤上列出的所有步驟內容。
- 在專案**詳細資料**標籤上列出的所有專案內容；例如，專案類別、選取元及其他內容。
- **標籤**標籤上的專案標籤格式。

註：這個標籤在專案內容畫面上。

- 如果來源專案使用 Perl 引擎，而目標專案使用 Java 引擎，則 Perl 專案的通過/失敗鏈結專案也會複製為 Java 引擎。Perl 專案的步驟列入專案會複製為 Java 引擎。

註：不支援將 Java 引擎專案複製到 Perl 引擎專案。

系統不會複製下列內容：

- **標籤**標籤上的標籤變數。

註：這個標籤在專案內容畫面上。會複製標籤格式中使用的標籤變數。

- **暫存器**標籤上的專案暫存器。

註：這個標籤在專案內容畫面上。

- **附註**標籤上的步驟附註。

註：這個標籤在專案內容畫面上。

- **附註**標籤上的專案附註。

註：這個標籤在專案內容畫面上。

刪除專案


視專案是否有任何相關聯的工作而定，有兩種方法可用來刪除專案。

選擇下列其中一個選項來刪除專案：

刪除專案按鈕

註：專案一旦刪除便無法復原。


刪除專案按鈕會刪除沒有工作的專案。若要使用這個按鈕來刪除專案，您必須先刪除專案的所有工作。此按鈕位於專案內容編輯頁面以及專案步驟清單中。若要檢視專案內容編輯頁面，請完成下列步驟：

1. 在左側功能表中，選取**專案**。
2. 按一下所需專案名稱旁的**編輯**圖示 。

除去按鈕


註：即使專案已鎖定，除去作業還是會刪除這些專案。此外，除去專案之後就無法復原。

除去按鈕將專案和其所有相關聯的工作從 Build Forge 資料庫中刪除。除去專案之前，系統會先要求您確認。此按鈕位於專案內容編輯頁面中。若要檢視專案內容編輯頁面，請完成下列步驟：

1. 在左側功能表中，選取**專案**。
2. 按一下所需專案名稱旁的**編輯**圖示 。

將步驟固定在伺服器中

只要選取元容許，專案中的步驟都可以在不同的伺服器上執行。不過您可能會希望所有或大部分的專案步驟，都在同一部伺服器上執行，不管您是否有事先指定該伺服器都一樣。您可以利用專案層次的**固定**內容做到這一點。

若要檢視專案層次內容，請選取**專案**，然後按一下所需專案名稱旁的**編輯**圖示 。

「固定」內容只適用於沒有指定專屬選取元的專案步驟。如果步驟具有非「預設值」的選取元選項，系統便會使用該選取元來替步驟選擇伺服器，即使該選取元與專案的選取元相同亦然。

當您設定「固定」內容時，對於其選取元欄位設為「預設值」的所有步驟，專案會使用相同的伺服器。即使重新啟動專案，這個內容也持續有效。

當系統啟動列入專案時，即會將列入專案的選取元，作為列入步驟的預設選取元。執行呼叫端專案的「固定」內容，不會影響列入專案，而列入專案也會遵從它自己的「固定」內容（如果有設定的話）。

當系統啟動列入程式庫時，它會遵從下列規則：

- 沒有勾選「固定」內容的列入程式庫：會使用呼叫端步驟的選取元，作為列入步驟的預設選取元。

- 有勾選「固定」內容的列入程式庫：會使用呼叫端步驟的伺服器，作為列入步驟的預設伺服器。

註：您可以使用 `.bset` 伺服器指令，在進行工作期間變更專案的預設伺服器。在 `.bset` 指令之後執行的步驟，會使用該指令所設定的新預設值，並且固定在該新伺服器中。

鏈：另一個專案或程式庫的條件式執行

您可以在專案層次上呼叫兩種鏈：

- 通過鏈：指定當專案通過時，要執行的專案或程式庫。
- 失敗鏈：指定當專案失敗時，要執行的專案或程式庫。

此特性有數種用途：

- 在專案層次上採用條件式執行。在步驟層次提供其他流程控制功能。此外，步驟可以有它自己的「通過鏈」及「失敗鏈」。請參閱第 323 頁的第 20 章，『使用步驟』中的第 327 頁的『控制執行流程』。
- 單獨維護常用的步驟群組，而不需顧及依賴這些群組的專案。為此，也可以使用程式庫。
- 在專案通過或失敗之後清除檔案。
- 當軟體建置專案通過時，呼叫自動化測試及部署專案。

來自呼叫端專案或程式庫的鏈繼承

鏈結的專案或程式庫會繼承呼叫端專案的部分性質：

- 依預設，鏈結的專案會繼承呼叫端專案的類別。您可以在**管理 > 系統**中將「鏈結時置換類別」設為「否」，即可變更此行為。
- 鏈結的程式庫會繼承呼叫端步驟的選取元，因為程式庫本身並無選取元。如果該步驟沒有選取元，程式庫便會繼承建置的選取元。鏈結的程式庫的步驟會使用繼承的選取元，除非這些步驟有明確的選取元。

否則，鏈結的專案或程式庫會使用其本身的性質來執行：

- 由本身的選取元指定其伺服器。
- 使用其本身的內容，包含本身的通知設定及鏈設定。
- 在呼叫端專案的環境之後套用其環境。呼叫端專案的變數會重新命名，而且可在被呼叫的專案中使用。另請參閱第 294 頁的『鏈結的專案中的環境變數繼承』。

巢狀鏈結

當您鏈結專案時，會對呼叫端專案中呼叫的專案建立巢狀。巢狀層次的上限為 32 層。巢狀層次也可能會受限於執行「管理主控台」的主機上可用的記憶體。

執行及岔斷鏈

如果您在鏈結的專案中使用 `.break` 指令，系統會停止鏈的專案，但會將控制權交回給呼叫端專案，由呼叫端專案繼續。請參閱 第 345 頁的『`.break`』。

鏈結的專案中的環境變數繼承

當透過通過/失敗鏈啟動專案時，系統會套用呼叫端專案的環境變數。被呼叫專案會依下列順序，從呼叫端專案的環境及其本身環境來設定變數：

1. 被呼叫專案的伺服器環境。
2. 呼叫端專案的變數（以集為單位），"BF_" 變數名稱變成 "BF_CALLER_"。
3. 被呼叫專案的伺服器環境（若已被呼叫端的變數修改，則會再次套用）。
4. 被呼叫專案的環境。
5. 執行時所用的步驟環境（若有指定的話）。

啓用等待時取消鏈結的專案

一般而言，系統不會取消鏈結的專案。您可以將「通過等待」或「失敗等待」屬性設為「是」，讓系統自動取消「通過鏈」或「失敗鏈」中被呼叫的專案。當取消呼叫端專案或呼叫端步驟時，系統就會取消被呼叫的專案。

定義標籤

系統使用標籤 來識別專案的特定工作，並建構執行程序活動所處的預設工作目錄名稱。系統會從專案的標籤格式內容中建立工作的標籤，它可包含靜態文字和數值標籤變數。

專案的預設標籤格式是 `BUILD_$B`，它使用預設標籤變數 `B`，這是系統定義給每一個專案的自動增量值。此預設標籤格式會產生建置標籤串流，如下所示：

`BUILD_1`

`BUILD_2`

`BUILD_3`

不過，不是只有這些標籤可供使用。您可以定義自己的標籤變數及設定自己的標籤格式，來產生各種不同的標籤類型。您也可以在工作期間使用 `.retag` 指令，將標籤變更為任意字串。（如需相關資訊，請參閱 第 359 頁的『`.retag`』。）

目前工作的標籤可作為系統在工作期間定義的環境變數 (`BF_TAG`)，這樣您就可以存取並將它用來標示來源儲存庫，或作為其他追蹤或標示用途。（如需這些變數的相關資訊，請參閱第 273 頁的『系統變數參照』。）

您可以將兩個專案的標籤變數同步化；這會建立鏈結，而當其中一個專案執行時，就採用相同的標籤變數值。（如需相關資訊，請參閱第 295 頁的『同步化標籤』。）

本節的主題說明如何設定標籤格式和標籤變數，來產生能反映您所要之值的動態標籤。

編輯專案的標籤格式

標籤格式定義系統建構標籤的方法。標籤格式包含純文字以及以 `$` 符號指示的變數參照。您在標籤格式中使用的任何變數，都必須出自前一區段中由系統定義的標籤變數清單，或者您必須在執行專案之前，為該專案定義這些變數。未定義的變數會視為靜態文字。

標籤格式是一個專案內容。如果要編輯它，請按一下「專案」按鈕以顯示專案清單，然後按一下想要編輯之專案的專案名稱。系統會顯示專案中的步驟清單；按一下頁面頂端的專案名稱，以顯示專案內容。

在您的標籤格式中，使用 \$ 符號來表示標籤變數的開頭。若您願意，可併入數個標籤變數。例如，您可以為專案的主要修訂 (\$MAJ) 定義一個不會增量的變數，為次要修訂 (\$MIN) 定義一個會增量的變數，然後使用標籤格式來反映專案的版本號碼，例如 \$MAJ.\$MIN 版。這可讓您手動控制主要版本號碼，但隨著版次的發行自動增加次要版本號碼，產生的標籤如下所示：

Version1.1

Version1.2

同步化標籤

您可以利用專案層次 Tag Sync 內容，在不同專案之間，將標籤同步化，使兩個或更多專案使用相同的變數值。當您設定 Project B 的 Tag Sync 內容等於 Project A 時，便在 Project A（母項）和 Project B（子項）之間建立了上下代關係。

當您執行有 Tag Sync 內容的專案時，系統會檢查子專案中是否有任何標籤變數符合母專案中的標籤變數。如果找到，子專案的變數會設為母專案前次使用的值。

如果子專案的標籤格式中沒有變數符合母專案標籤格式中的變數，則 Tag Sync 內容沒有作用。

同步化只適用於標籤變數的值。「自動增量」和「填補」內容不會同步化。

只有標籤中的變數同步化，因此您仍然可以區分不同專案。

例如，假設定義了兩個專案，如下表所示：

專案	標籤格式	自動增量	標籤同步化
Project A	Project_A_\$B	是	-- None --
Project B	Project_B_\$B	是	Project A

如果您接著輪流執行這些專案（從 Project A 開始），完成的工作清單會顯示標籤如下所示。最後執行的項目最先顯示，在已完成的工作清單中，也以相同方式來顯示工作。

專案	標籤
Project B	Project_B_4
Project A	Project_A_3
Project B	Project_B_2
Project A	Project_A_1

如果您只將母專案的「自動增量」內容設為「是」，結果便不同。專案的設定方式如下所示：

專案	標籤格式	自動增量	標籤同步化
Project A	Project_A_\$B	是	-- None --
Project B	Project_B_\$B	否	Project A

如果您接著輪流執行這些專案（從 Project A 開始），完成的工作清單會顯示標籤如下所示。最後執行的項目最先顯示，在已完成的工作清單中，也以相同方式來顯示工作。

專案	標籤
Project B	Project_B_2
Project A	Project_A_2
Project B	Project_B_1
Project A	Project_A_1

系統定義的標籤變數

您可以在工作標籤中使用下列預先定義的變數：

變數	值
B	工作號碼：從 1 開始的整數值，並隨每一個新工作而遞增。
BF_D	日期，採「標籤：日期格式」系統設定所設定的格式。此變數是在執行時進行計算，因此無法在專案起始頁上的「標籤範例」中計算。使用此變數時，請以 { } 將其括住。
BF_J	一年中的第幾天。此變數是在執行時進行計算，因此無法在專案起始頁上的「標籤範例」中計算。使用此變數時，請以 { } 將其括住。
BF_T	時間，採「標籤：時間格式」系統設定所設定的格式。此變數是在執行時進行計算，因此無法在專案起始頁上的「標籤範例」中計算。使用此變數時，請以 { } 將其括住。
BF_W	星期幾（0 到 6 的數值）。此變數是在執行時進行計算，因此無法在專案起始頁上的「標籤範例」中計算。使用此變數時，請以 { } 將其括住。

系統定義變數與起始頁上的標籤範例


部分變數是在工作啟動時進行計算，因此無法在起始頁上的「標籤範例」中計算。使用這些變數時，請以 { } 將其括住。

範例：

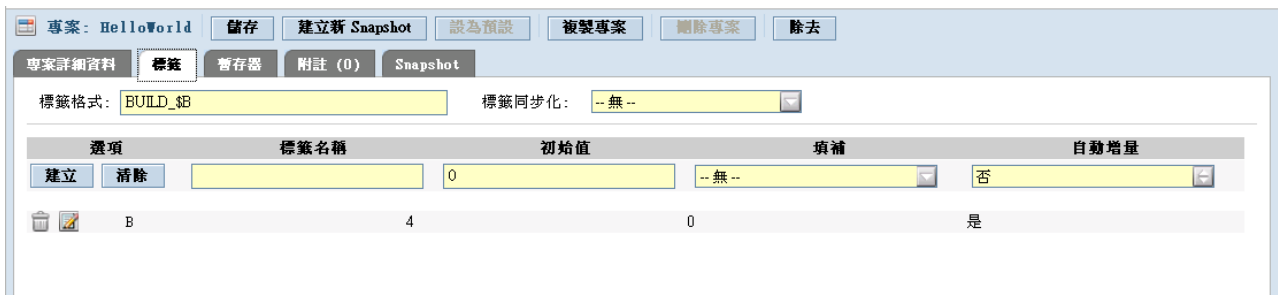
```
BUILD_${B}.${BF_T}  
BUILD_${B}.${BF_D}.${BF_T}
```

建立或編輯標籤變數


您可以定義自己的標籤變數，以包含在標籤格式中。標籤變數採用數值，如有需要，可由系統對每個工作進行自動增量。

若要為專案新增變數或編輯現有的變數，請選取**專案**，然後按一下所需專案名稱旁邊的**編輯**圖示 。此時專案內容會出現在畫面底端；請按一下**標籤**標籤來顯示專案的標籤變數。

系統會顯示適用於專案的標籤變數清單。



選項	標籤名稱	初始值	填補	自動增量
<input type="button" value="建立"/>	<input type="text" value="B"/>	<input type="text" value="0"/>	<input type="text" value="-- 無 --"/>	<input type="text" value="否"/>
<input type="button" value="清除"/>				

- 若要編輯標籤變數，請按一下標籤變數名稱旁的**編輯**圖示 。系統會在畫面中填入標籤變數的值，並將**建立**變更為**儲存**按鈕。變更這些值，然後按一下**儲存**，即可儲存您所做的變更。
- 如果要刪除標籤變數，請按一下標籤變數名稱旁的垃圾筒圖示。
- 若要新增變數，請輸入變數的內容，然後按一下**建立**按鈕。

如需「標籤名稱」、「起始值」、「填補」及「自動增量」內容的相關資訊，請參閱第 288 頁的『「標籤」標籤』。

程式庫

程式庫是指其「選取元」內容設為「無」的任何專案。這些專案預期在其他專案內執行。這些是在呼叫它們之步驟的伺服器資源上執行。

如果您要儲存的專案沒有選取元，系統會警告您，表示此專案將儲存成程式庫。程式庫會列在**程式庫**畫面中。

如果要從步驟呼叫程式庫，請在步驟的「列入」、「通過鏈」或「失敗鏈」內容中選擇它。

關於程式庫

「程式庫」畫面中所含的程式庫，是沒有指定選取元的專案。

程式庫會使用呼叫它們的任何步驟的選取元。如果呼叫端步驟沒有選取元，程式庫便會使用步驟的專案選取元。其他的專案通常會將程式庫當成步驟的「列入」、「通過鏈」或「失敗鏈」來呼叫。

從「程式庫」畫面，您可以檢視、編輯、建立或啟動程式庫。您可以讓程式庫自動執行，但在這樣做時，您必須指定選取元。

註：依預設，當啟動程式庫時，會指派清單中的第一個選取元。如果您想指定不同的預設選取元，請讓它成為專案。

若要將程式庫變更為專案，您可以編輯專案，然後替它選擇選取元。當您儲存具有選取元的程式庫時，它就會變成專案，而從「程式庫」清單中消失。


除了沒有選取元之外，程式庫與其他專案並無不同。

若要存取「程式庫」畫面，請在左側功能表中按一下**程式庫**。



複製程式庫

若要建立現有程式庫的副本，請完成下列步驟：

1. 選取**程式庫**。
2. 選取您要複製的程式庫旁邊的**編輯**圖示 .
3. 按一下**複製程式庫**。

當您複製程式庫時，系統會將現有程式庫的下列內容複製到新程式庫：

- 步驟以及在這些步驟的**詳細資料**標籤上列出的所有內容
- 程式庫的**詳細資料**標籤上列出的所有程式庫內容，例如類別及其他內容
- 程式庫的標籤格式（位於程式庫內容畫面上的**標籤**標籤）

系統不會複製下列內容：

- 標籤變數（位於程式庫內容畫面上的**標籤**標籤）；不過，也會複製標籤格式中所使用的標籤變數
- 程式庫暫存器（位於程式庫內容畫面上的**暫存器**標籤）
- 步驟附註（位於步驟內容畫面上的**附註**標籤）
- 程式庫附註（位於程式庫內容畫面上的**附註**標籤）

日誌過濾器

這個主題說明如何建立及使用日誌過濾器。

關於日誌過濾器

日誌過濾器可用來指定步驟的成功準則。過濾器可儲存一個以上的正規表示式。

如果沒有設定過濾，Build Forge 會依照指令的結束狀態，來判斷步驟成功或失敗，其中 0 為成功、1 為失敗。如果「指令」內容中使用了多個指令，則只有發出的最後一個指令的結束狀態，才會影響步驟結果狀態。請注意，有些指令一律傳回 0 結束狀態。像 `net use` 這樣的報告指令會列印對映網路磁碟機的清單。指令一律會成功，即使您所要的磁碟機未列在清單中。

日誌過濾器可讓您對步驟中指令的輸出求值，而非對結束狀態求值。

以 `net use` 指令為例，您可以使用日誌過濾器來尋找特定的磁碟機，如果找到，就將步驟標示為成功。

關於「日誌過濾器」畫面

您可以使用「日誌過濾器」畫面來建立、編輯及刪除日誌過濾器。若要檢視此畫面，請選取專案 > 日誌過濾器。



此畫面具有下列欄位。

名稱 日誌過濾器的名稱。

存取 可使用此過濾器的使用者存取群組。

建立日誌過濾器

日誌過濾器可以包含一個以上過濾器型樣。每個過濾器型樣都與一個動作相關聯，並且選擇性地與存取群組相關聯以進行通知。您會先定義日誌過濾器，然後將日誌過濾器與專案中的步驟相關聯。

如果要建立日誌過濾器，請執行下列動作：

1. 選取專案 > 日誌過濾。 「管理主控台」會顯示「日誌過濾器」清單和「新日誌過濾器」畫面。
2. 在**名稱**中，輸入日誌過濾器名稱，然後按一下**儲存**。「管理主控台」會儲存日誌過濾器並顯示「新型樣」畫面。

3. 請針對您為日誌過濾器所定義的每個過濾器型樣，執行下列動作：
 - a. 在**型樣**欄位中輸入正規表示式。正規表示式在 Java 引擎專案中必須與 Java 相容，在 Perl 引擎專案中必須與 Perl 相容。Build Forge 會在專案執行時，搜尋型樣的步驟輸出。如需詳細資料，請參閱第 301 頁的『過濾器型樣』。
 - b. 在**動作**處，選擇在找到過濾器型樣時，要採取的過濾器動作。預設內容為「設定失敗」，會將步驟狀態設為「失敗」。如需詳細資料，請參閱第 302 頁的『過濾器動作』。
 - c. 在「通知」中，選擇性地選取存取群組，以便在啟動過濾器時，傳送通知電子郵件給成員。
 - d. 按一下**儲存**。

如果要使用日誌過濾器，請選擇專案步驟，並將步驟的**結果**內容設為新的日誌過濾器。請參閱『將日誌過濾器指派給步驟』。

將日誌過濾器指派給步驟

如果要使用日誌過濾器，則對於使用**結果**內容的專案步驟，必須指定日誌過濾器。如果您將日誌過濾器指派給步驟，每當專案執行時，就會在步驟輸出中，執行日誌過濾器中的過濾器型樣。不過，日誌過濾器不會套用至配接器輸出。

如果您將日誌過濾器指派給步驟，日誌過濾器設定的步驟結果會置換所有其他準則（這些準則可以決定步驟成功還是失敗），包括步驟指令或任何步驟內容的結束狀態。比方說，如果步驟的執行時間超過步驟「逾時」內容指定的時間，步驟就會停止。但是並不會將步驟的狀態視為失敗，除非其關聯的日誌過濾器動作使其設為失敗。

如果要將日誌過濾器指派給步驟，請執行下列動作：

1. 選取**專案或程式庫**以存取步驟。
2. 選取內含步驟的專案或程式庫。
3. 選取步驟以開啓步驟「詳細資料」畫面。
4. 在**結果**中，選取每當步驟執行時，您要執行的日誌過濾器。

啓用日誌過濾器以包含步驟日誌中的項目

日誌過濾器可以套用到下列步驟日誌明細行類型：

- EXEC
- RESULT
- ERROR
- READ
- WRITE
- MANIFEST
- TMO

套用之後，過濾結果中會包含對應的步驟日誌輸出行。日誌過濾器無法套用到 STEP 類型。日誌過濾器一律跳過 STEP 類型。

過濾器型樣

過濾器型樣會定義您要在步驟輸出中比對的字串或表示式。您所建立的每個過濾器型樣都與單一過濾器動作相關聯。過濾器型樣和動作都是在過濾器日誌集中定義。由於您可以直接將多個過濾器型樣併入日誌過濾器然後套用至單一步驟的輸出，因此無須建立複雜的表示式，就能使用多重搜尋準則。

如果要建立日誌過濾器，請選取**專案 > 日誌過濾器**。如需詳細資料，請參閱第 299 頁的『日誌過濾器』。

過濾器型樣的語法

檢閱過濾器型樣的建立準則：

- 過濾器型樣定義為正規表示式，且在 Java 引擎專案中必須使用與 Java 相容的語法，而在 Perl 引擎專案中必須使用與 Perl 相容的語法。如需建構相容表示式的相關詳細資料，請參閱 Java 正規表示式文件或 Perl 文件。
- 系統會新增正斜線定界字元（`/<表示式>/`），因此請指定**沒有**用正斜線定界字元在前後包住的（表示式）。
- 如果表示式包含 meta 字元（例如 `a /b`），該 meta 字元之前必須加上反斜線跳出字元（`a\b`）。

下表顯示部分標準正規表示式的語法。

表示式	相符項
Production	在字串中的任何位置符合 <i>Production</i> 。
^Production	在字串開頭符合 <i>Production</i> 字串。
Error:.*[0-9]\$	符合包含 <i>Error</i> 字串的一行，後面有任意字元集，且字串尾端有一個數字。
[Ww]arning	符合 <i>Warning</i> 或 <i>warning</i> 字串。
.*	符合任意字元 0 次以上。點 (.) 代表符合任一字元，而星號 (*) 則代表符合 0 次以上。

同一行中有多個型樣相符

如果要建構型樣過濾器，很重要的一點是要瞭解系統如何搜尋型樣相符項目。

系統會根據所有過濾器型樣，依序檢查輸出的每一行是否有相符項；找到相符項時便會停止，然後再繼續下一個型樣。因此，如果一行出現型樣兩次，系統可能不會發現。例如，請考量下列輸出行：

```
exception retrying exception
```

使用下表中的過濾器型樣，系統會比對第一個 *exception*，並將步驟結果設為「失敗」，接著比對 *retrying* 並將步驟結果設為「通過」，然後移至下一行，而不會比對第二個 *exception*。

過濾器型樣	過濾器動作	範例說明
[Ee]xception [Rr]etrying	設定失敗 - 失敗清除失敗 - 通過	此型樣對 Java 專案非常有用；它的步驟在發生異常狀況時失敗，但在重試時會清除失敗。如果重試失敗，便會產生新的異常狀況，因此，指令的最終狀態是有效的。

解決這個問題的方式之一，是以下列過濾器型樣取代表格中的過濾器型樣：

retrying.*exception

過濾器動作

過濾器動作是指每當在步驟輸出中發現過濾器型樣時，所要採取的動作。您所建立的每個過濾器型樣都與單一過濾器動作相關聯。過濾器動作和型樣都定義於日誌過濾器中。

如果要建立日誌過濾器，請選取**專案 > 日誌過濾器**。如需詳細資料，請參閱第 299 頁的『日誌過濾器』。

過濾器動作	定義	步驟結果
設定失敗（預設值）	系統找到過濾器型樣時，會將步驟結果狀態設為失敗，並繼續在目前行搜尋集合中的過濾器型樣。	失敗
設定失敗/中止	系統找到過濾器型樣時，會將步驟結果狀態設為失敗，並停止在目前行搜尋集合中的過濾器型樣，然後跳到下一行，重新開始搜尋型樣。	失敗
清除失敗	系統找到過濾器型樣時，會將步驟結果狀態設為通過，並繼續在目前行搜尋集合中的過濾器型樣。	通過
清除失敗/中止	系統找到過濾器型樣時，會將步驟結果狀態設為通過，並停止在目前行搜尋集合中的過濾器型樣，然後跳到下一行，重新開始搜尋型樣。	通過
中止	系統找到過濾器型樣時，會停止在型樣集中尋找過濾器型樣的現行行，然後跳到下一行，重新開始搜尋型樣。它不會變更步驟結果狀態。	不適用
併入	「併入」可讓您參照其他日誌過濾器中的一個以上日誌過濾器。請在「型樣」欄位中指定您要併入的日誌過濾器，並在「動作」欄位中選取併入。	不適用
警告	系統找到過濾器型樣時，會將步驟結果狀態設為「警告」，並繼續在現行行中搜尋集合中的過濾器型樣。（附註：「警告」狀態是通過狀態；指派給此步驟的任何通過鏈都將會執行。）	警告
清除警告	系統找到過濾器型樣時，會將步驟結果狀態設為通過，並繼續在目前行搜尋集合中的過濾器型樣。	通過
清除警告/中止	系統找到過濾器型樣時，會將步驟結果狀態設為通過，並停止在目前行搜尋集合中的過濾器型樣，然後跳到下一行，重新開始搜尋型樣。	通過

過濾器動作	定義	步驟結果
通知變更器	<p>如果要使用通知變更器，則建立關係清單的配接器必須包含在專案中，而且該配接器步驟必須在包含通知變更器日誌過濾器的步驟之前執行。</p> <p>配接器關係清單是以使用者和物件（例如，變更的檔案）作為配對組合。如需詳細資料，請參閱「配接器 XML 參照」。</p> <p>當配接器執行並建立關係清單之後，如果含有通知變更器動作的日誌過濾器，符合它在步驟輸出線內的過濾器型樣時，就會再次掃描該行，以尋找關係清單中相符的物件。如果找到相符的物件，則與該物件配對的使用者，就會收到電子郵件通知。</p> <p>例如，在下列步驟輸出行中，符合過濾器型樣 <code>Error</code> 的物件是 <code>MyFile.c</code>。因此在關係清單中，與 <code>MyFile.c</code> 物件配對的使用者，就會收到有關該錯誤的電子郵件通知。</p> <p><code>Error: Invalid token on line 55 of MyFile.c</code></p>	不適用
停止「失敗」結果的建置	<p>系統找到過濾器型樣時，會將工作結果設為失敗，並結束工作。</p> <p>若為 Java 引擎專案，則步驟結果為失敗。若為 Perl 引擎專案，則會根據結果來設定步驟結果。不會執行進一步的步驟。</p>	若為 Java 引擎專案，則步驟結果為失敗。若為 Perl 引擎專案，則會根據結果來設定步驟結果。
停止「通過」結果的建置	<p>系統找到過濾器型樣時，會將工作結果設為通過，並結束工作。</p> <p>若為 Java 引擎專案，則步驟結果為通過。若為 Perl 引擎專案，則會根據結果來設定步驟結果。不會執行進一步的步驟。</p>	若為 Java 引擎專案，則步驟結果為通過。若為 Perl 引擎專案，則會根據結果來設定步驟結果。
停止「警告」結果的建置	<p>系統找到過濾器型樣時，會將工作結果設為警告，並結束工作。</p> <p>若為 Java 引擎專案，則步驟結果為警告。若為 Perl 引擎專案，則會根據結果來設定步驟結果。不會執行進一步的步驟。</p>	若為 Java 引擎專案，則步驟結果為警告。若為 Perl 引擎專案，則會根據結果來設定步驟結果。

過濾器通知

對於日誌過濾器中的每個過濾器型樣，您可以選擇性地設定通知，以傳送電子郵件給存取群組，通知其成員已經啟動步驟的型樣過濾器。

錯誤臨界值

您可以使用特殊的環境變數（_ERROR_THRESHOLD），來建立個別步驟及/或專案的臨界值。然後，系統會計數符合的過濾器數目，並在符合臨界值時讓步驟或專案失敗，或者在工作附註中註明已屆臨界值一事。

如需相關資訊，請參閱第 276 頁的『觸發變數參照』。

錯誤與警告計數

建立過濾器與步驟之間的關聯性以判斷步驟成功與否時，系統會顯示過濾器捕捉到的錯誤和警告數目。在**工作 > 已完成**標籤中，此數目以括弧括住，顯示在**結果**直欄中的工作結果後面。格式為 (<失敗相符項> / <警告相符項>)。

範例：**失敗 (1 / 0)** 工作結果顯示工作失敗，符合 1 個「失敗」過濾器，但沒有符合「警告」過濾器。

類別

這個主題說明如何建立及使用類別。

關於類別

類別是工作群組。每一項工作都只能是唯一一個類別的成員。您可以使用類別，將不同的廣域管理行為套用至類別中的每一項工作。工作會從其專案的內容取得其預設類別。從**工作 > 啟動**頁面啟動工作時，您也可以手動為工作選擇不同類別。

類別具有管理下列活動的內容：

- 自動刪除工作
- 當系統清除此類別的工作，或將現有的工作變更為此類別（或從此類別進行變更）時，即會啟動工作

註：您可以在工作完成之後，變更工作的類別。若要變更工作的類別，請選取**工作 > 已完成**來檢視這個工作，然後按一下工作標籤（例如，BUILD_5）。在**類別**欄位選取不同類別。

關於「類別」畫面

您可以使用「類別」畫面來新增、編輯及刪除類別。若要檢視此畫面，請選取**專案 > 類別**。系統會顯示類別清單。選取要編輯其內容的類別。

[使用者介面配置](#) | [主控台](#) | [報告](#) | [登出](#): Root User

類別 新增類別 說明 ?

過濾條件 顯示 1 - 2 (總共 2) [顯示全部](#)
第 1 頁 (共 1 頁)

建置類別	存取	刪除	日	次數	其
暫存	訪客	所有項目	5	無限制	任何建置
正式作業	訪客	主控台資料	永遠	無限制	任何建置

(新建類別) 儲存 複製 刪除

詳細資料

名稱: 存取: 安全人員

刪除檔案:

清除時啟動:

所有項目

-- 無 --

日:

進入時啟動:

永遠

-- 無 --

次數:

結束時啟動:

無限制

-- 無 --

其:

任何建置

類別的**存取**內容會根據您所指派的存取群組，控制哪些使用者可以檢視或變更類別。

用於自動刪除工作的內容：

類別的大部分內容都可控制要刪除哪種類型的專案資料，以及要在哪種情況下進行刪除。

系統會定時檢查要刪除的工作，間隔由「清除檢查時間」系統設定所定義，預設值為 15 分鐘。

註：您也可以使用排程來表示要執行清除的時間，以免系統在忙碌時，仍然嘗試執行清除作業。您可以使用這個特性，讓清除作業只在晚上進行，或者例如一週一次。請參閱第 304 頁的『類別』。

當執行清除工作時，系統會根據類別中的設定，來保存工作和刪除資料。

刪除檔案

判斷要刪除的資料類型。其選項如下所示：

所有項目

從資料庫刪除工作的所有相關資訊，並從執行該工作之伺服器上刪除工作目錄。

主控台資料

從資料庫刪除工作的所有相關資訊，但將工作目錄完整地保留在伺服器上。

日誌與檔案

刪除工作目錄及日誌，但在**工作 > 已保存**頁面上保留步驟通過/失敗資訊。

僅日誌 僅刪除工作日誌。

僅檔案 刪除執行工作之伺服器上的工作目錄。日誌和一些其他資訊（例如，步驟成功/失敗狀態）會保留在資料庫內；工作記錄會移動到**工作 > 已保存**頁面。

天數* 在刪除工作之前，此工作必須歷經的天數。

計數* 允許的工作數上限。當工作數超過「計數」值時，系統會排定消除工作的排程，以刪除額外的建置。預設值（無限制）會預防系統因為存在的工作數而刪除工作。

條件 「條件」內容會另外設定其他條件，心須符合這些條件才能刪除工作。其選項如下所示：

任何建置

選取這個選項時，「條件」內容便不會影響刪除工作的作業。

僅失敗 系統僅會刪除失敗的工作。

僅通過 系統僅會刪除成功的工作。

保持讓 1 通過

系統一律會保留最近的成功工作，即使該工作符合其他刪除準則。

* 當超過「天數」**或者**「計數」值時，系統便會刪除工作。比方說，如果「計數」設為 10，而「天數」設為 2，共有 8 個工作，但有 3 個工作存在 2 天以上，則系統會刪除那 3 個工作。同樣地，如果有 12 個工作，全部都低於 2 天，則會刪除存在最久的那 2 個工作。

當執行清除工作時，系統會根據類別中的設定，來保存工作和刪除資料。

用於在發生事件時啟動專案的內容：

您可以在發生與類別相關的某些事件時，啟動（鏈）專案。使用這些內容時，您可以建立處理程序狀態進度的模型。

下列類別內容可讓您在發生某些事件時啟動工作：

清除時啟動

只要類別中有任何工作被清除（也就是說，每當系統針對這個類別的工作，啟動清除工作時），這個內容就會啟動指定的專案。您可以使用這個內容，確定刪除某些特定的檔案，這些檔案沒有隨著清除工作而自動刪除。

進入時啟動

這個內容會在工作的類別內容變更為這個類別時，啟動指定的專案。您可以使用這個內容，將程序連結到工作重新分類；例如，建立 Test 類別，然後當系統對工作提示 Test 類別時，啟動一些標準測試。

結束時啟動

這個內容會在工作的類別內容從這個類別變更為另一個類別時，啟動指定的專案。

這些內容會以鏈的形式啟動專案。

註：如果這些內容會啟動含有「必須變更」變數的工作，變數值會維持不變。

設定通知

當專案或步驟通過或失敗，或發生某些其他事件時，系統可送出電子郵件通知。本節說明如何配置電子郵件通知，以及如何修改控制電子郵件通知外觀的通知範本。

通知會傳送至存取群組，因此，在設計存取群組時，要同時注意通知和安全性。通知一律傳送至群組，而不直接傳送至個別使用者，但必要的話，您可以設定只包含一個使用者的群組。

如果您已將存取群組設為階層式（存取群組包含子群組），通知的運作方式如下所示：

- 如果通知的存取群組是母群組且含有一個以上使用者，通知只會傳給母群組中的使用者。
- 如果通知的存取群組是母群組且只含子群組，不含任何使用者，通知會傳給所有子群組使用者。

如果要建立通知事件，請選取通知內容的存取群組。

- 對於專案，您可以選擇「啟動通知」、「通過通知」和「失敗通知」群組。
- 對於步驟，您可以選擇「通過通知」和「失敗通知」群組。

只要為這些內容選取了存取群組，系統就會在發生適當事件時將電子郵件傳至群組成員。這些電子郵件訊息中的任何時間參照，都是以傳送通知的「管理主控台」的時區為基礎。

當專案併入另一個專案作為列入專案時，會忽略該列入專案的啟動、通過和失敗通知設定，但允許使用其步驟的任何通知設定。如需詳細資料，請參閱第 314 頁的『列入專案的通知』。

在使用通知之前，您必須：

- 配置「SMTP 伺服器」系統設定，讓系統知道該使用哪一個 SMTP 伺服器來傳送電子郵件。預設值為 localhost。此外，您或許需要設定系統警示來源的系統設定。這個位址是作為來源端位址使用，而且大部分的 SMTP 伺服器都要求使用有效的來源端位址。預設值為 root@localhost。
- 建立一個或多個通知群組並指派使用者給它們。
- 選取針對個別專案及/或步驟而通知的群組。

此外，您可以透過編輯通知範本，來配置系統送出的通知電子郵件。請參閱第 310 頁的『自訂通知範本』。

關於通知範本

通知範本提供一種方法來傳送自訂訊息給您，通知您有關系統中發生的事件。

提供的範本類型

本系統包含一組如下所示類型的通知範本：

- 變更構件符合訊息
- 過濾符合訊息
- 工作岔斷訊息
- 工作失敗訊息

- 工作通過訊息
- 工作啟動訊息
- 工作警告訊息
- 清除失敗訊息
- 步驟電子郵件訊息
- 步驟失敗訊息
- 步驟通過訊息
- 步驟警告訊息
- 系統警示訊息

每種類型均包含一組已翻譯的訊息。這些訊息使用變數來提供環境定義。

廣域用法

全面使用提供的範本，如下所示：

- 專案：每個專案中的內容均容許您指定專案啟動、通過及失敗的通知類型。「工作警告」通知是為完成但出現警告的工作而提供。
- 步驟：每個步驟中的內容均容許您指定通過及失敗的通知類型。「步驟警告」通知是為步驟失敗後繼續執行的工作而提供。
- 其他通知類型如下所示：
 - 變更構件符合訊息 - 構件已變更，且與使用者相關
 - 工作 `.break` 訊息 - 工作遇到包含 `.break` 指令的步驟
 - 步驟 `.email` 訊息 - 步驟執行 `.email` 指令
 - 系統警示訊息

翻譯中儲存的訊息

所用的通知訊息儲存在與範本類型相關聯的翻譯中。每份翻譯均會指定其通知訊息的下列相關資訊：

- 語言
- 說明
- 主旨（用作通知電子郵件中的主旨）
- 內文（用作通知電子郵件中的訊息內文）

您可以編輯提供的通知範本，以及自己建立的通知範本中的翻譯。

訊息格式

使用下列兩種訊息類型之一，均可傳送訊息：

- 純文字訊息（預設值）
- MIME - 內文欄位中包含特定 HTML 標籤的通知，會自動以 MIME 訊息傳送。請參閱第 312 頁的『在「內文」欄位中使用 HTML』。

範本中的變數

系統會針對許多變數來剖析範本（請參閱第 312 頁的『使用環境變數及暫存器變數』。）

指定專案及步驟的自訂範本

您可以根據提供的通知類型，建立自訂通知範本。自訂範本必須與專案或專案中的個別步驟相關聯。請參閱第 310 頁的『為特定專案及步驟建立新範本』。

關於「範本」畫面

您可以使用「範本」畫面來設定事件的通知。若要檢視此畫面，請選取**專案 > 範本**。



此畫面會列出 Rational® Build Forge® 隨附的所有範本。建立新範本時，請在下列欄位中輸入資訊：

類型 訊息的類型。選取現有範本類型。

專案 指定傳送訊息的專案。

步驟 指定專案內傳送訊息的步驟。

關於「翻譯」畫面

使用「翻譯」畫面可設定範本所用的訊息。若要檢視畫面，請選取**專案 > 範本**，然後從清單中選取範本。即會顯示範本的翻譯清單。清單中的每個項目均會顯示下列資訊：

語言 必要。訊息的語言。

說明 用作清單中的翻譯標題。

從 選用項目：要在訊息的「寄件者」欄位中使用的電子郵件位址。系統支援下列電子郵件位址格式。

使用者名稱@host.com
名稱 使用者名稱@host.com
名稱 "<使用者名稱@host.com>"
名稱 <使用者名稱@host.com>
名稱 "使用者名稱@host.com"

其中

使用者名稱是電子郵件使用者名稱。

名稱 是任意字串，通常是某形式的給定名稱。

主題 必要：要在訊息的「主題」欄位中使用的字串。它可以包含變數。

主體 必要：訊息的文字。它可以包含變數。

配置您的 SMTP 伺服器

如果要配置 SMTP 伺服器，請選取**管理 > 系統 > SMTP 伺服器**。系統會顯示「SMTP 伺服器」值的編輯畫面。請輸入您站台的 SMTP 伺服器名稱。預設值為 `localhost`。

您可能需要設定**系統警示來源**參數，因為它是作為來源端位址使用，而且大部分的 SMTP 伺服器都要求使用有效的來源端位址。預設值為 `root@localhost`。

如需相關資訊，請參閱第 212 頁的『系統配置設定』。

設定專案和步驟的通知內容

當通知的 SMTP 和群組配置妥當之後，您可以配置專案和步驟在發生特定事件時傳送通知。

- 對於專案，您可以設定「啟動」、「通過」和「失敗」通知內容。它們是專案內容。
- 對於步驟，您可以設定「通過」或「失敗」通知內容。它們是步驟內容。

如果您在單一步驟中設定多項通知，只有第一個符合的結果會出現在電子郵件通知中。

通知練習

下列程序說明如何設定及嘗試電子郵件通知。這個練習需要有 SMTP 伺服器和電子郵件帳戶。

1. 根據『配置您的 SMTP 伺服器』的說明，設定您的 SMTP 伺服器

請確定您的使用者帳戶，可以將電子郵件傳送到您可以存取的電子郵件帳戶。

2. 選取**管理 > 存取群組**
3. 建立一個名為「電子郵件測試」的新存取群組，並以您所選擇的使用者作為起始成員。
4. 選取一個專案（例如，Hello World 專案），然後編輯其專案內容。請在「啟動通知」、「通過通知」和「失敗通知」欄位中，選取「電子郵件測試」群組。
5. 執行專案。
6. 驗證您已經收到兩封電子郵件：一封告訴您專案已經啟動，另一封則告訴您專案成功還是失敗。如果您沒有收到電子郵件，請驗證所用的 SMTP 伺服器值是否正確。

自訂通知範本

您可以依下列方式自訂通知範本：

- 為專案及步驟建立自訂範本：您可以為個別專案以及專案中的個別步驟建立自訂範本。您可以使用提供的範本類型來建立自訂範本。您還可以在自己建立的自訂範本中建立翻譯。
- 編輯提供的範本翻譯：您可以在提供的每個通知範本中，新增或編輯翻譯。您無法編輯提供範本的基本性質。

為特定專案及步驟建立新範本

系統隨附多種可能發生在系統上之事件的範本。您可以針對特定專案或專案中的步驟，建立新的範本。建立自訂通知之前，必須在系統中建立專案或步驟。

若要建立新範本，請執行下列動作：

1. 選取**專案 > 範本**。系統會顯示現行的範本清單。
2. 按一下**新增範本**。
3. 指定範本的資訊：
 - **類型**：選擇現有通知類型，以用作新範本的基礎。
 - **專案**：選擇要套用新範本的專案。範本只會用於執行選定專案時所產生的通知訊息。

註：如果針對專案定義了多個 Snapshot，便會出現一份 Snapshot 清單。請選擇要用的 Snapshot。通知只適用於利用這個 Snapshot 來執行的專案。如果您指定「預設 Snapshot」，便會使用所指派的預設 Snapshot。

 - **步驟**：您可以選擇特定步驟（這樣範本只會套用至該步驟的通知），或是選取「專案事件」，使範本套用至選定專案的所有通知。
4. 按一下**儲存**。
5. 為範本建立翻譯。
 - a. 從**專案 > 範本**中選擇範本
 - b. 按一下**新增翻譯**。
 - c. 指定範本的資訊：
 - **語言**：必要 - 訊息的語言。
 - **說明**：必要 - 此欄位將用作翻譯的標題。
 - **寄件者**：選用項目 - 訊息「寄件者」欄位中使用的電子郵件位址。
 - **主旨**：必要 - 訊息「主旨」欄位中使用的字串。它可以包含變數。
 - **內文**：必要：訊息的文字。它可以包含變數。如果它包含特定 HTML 標籤，則會以 MIME 訊息（而非純文字）來傳送訊息。
 - d. 按一下**儲存**。
 - e. 視需要為其他翻譯重複此處理程序。

編輯通知範本中的翻譯

每個通知範本均會使用「翻譯」清單來儲存通知訊息。

您可以編輯提供的通知範本，以及自己為個別專案與步驟建立的通知範本中的翻譯。

若要編輯翻譯，請執行下列動作：

1. 從**專案 > 範本**中選擇要編輯的範本。即會顯示翻譯清單。
2. 按一下要編輯的翻譯。
3. 指定範本的資訊：
 - **語言**：必要 - 訊息的語言。
 - **說明**：必要 - 此欄位將用作翻譯的標題。
 - **寄件者**：選用項目 - 訊息「寄件者」欄位中使用的電子郵件位址。
 - **主旨**：必要 - 訊息「主旨」欄位中使用的字串。它可以包含變數。
 - **內文**：必要：訊息的文字。它可以包含變數。如果它包含特定 HTML 標籤，則會以 MIME 訊息（而非純文字）來傳送訊息。
4. 按一下**儲存**。

在「內文」欄位中使用 HTML

依預設會以純文字傳送通知。如果您使用下列任何 HTML 標籤或部分，則會以 MIME 訊息傳送通知，並且會解譯所有 HTML 標籤：

```
<html
<body
<a
<b>
<p>
<table
```

當您建立翻譯時，系統會剖析「內文」欄位。它還會決定屆時要使用的格式。系統僅會掃描所列的標籤及標籤部分。

如果訊息不含任何要掃描的元素，則會以純文字傳送該訊息。並且不會解譯訊息中的 HTML 標籤。這些標籤將包含在訊息中。

使用環境變數及暫存器變數

只要您使用 \${VAR} 語法，就可以在通知範本中參照環境變數（您所定義的環境變數以及標準系統變數）。

您也可以參照通知範本中併入專案的暫存器變數。如果您參照的暫存器是空的，系統會傳回空字串。

特殊通知範本變數

下表列出通知範本可用的特殊變數。部分變數與環境定義相關，因此只有在相關時才可以使用（例如，STEPNAME 變數不能針對專案層次通知設定，只能針對步驟層次通知設定）。

變數	包含
ACTION	用於建置清除，說明執行的刪除類型。
BID	指定工作 ID 號碼。用來建構回到管理主控台的鏈結以存取報告。
CONSOLEHOST	「管理主控台」電腦的主機名稱。
CONSOLEPORT	管理主控台使用的埠號。容許您在通知範本內建構有效的 URL。
CONTEXTLOGLINKS	列出日誌中以『FILT:』開始的行號，每一個項目有三行內容。系統會在訊息中提供管理主控台日誌項目的鏈結。
DURATION	用於步驟，指定步驟及其列入的任何步驟的執行時間（秒）。
EID	指定環境 ID 號碼。用來建構回到管理主控台的鏈結以存取報告。

變數	包含
FULLNORMALLOG	會顯示工作中每一個步驟的日誌資訊，並排除出現在詳細日誌中的環境設定動作。
LINK	用於建置，指定鏈結名稱。
MESSAGE	包含錯誤或失敗的訊息文字或警示訊息。
ONFAIL	用於步驟，會保留步驟的繼續內容。
PATH	用於步驟，指定諸如伺服器或步驟之類的資料項目的路徑（適當的話）。
PID	指定專案 ID 號碼。用來建構回到管理主控台的鏈結以存取報告。
PROJECTNAME	包含專案的名稱。
RUNACTION	指定電子郵件範本要運用的變數。
SELECTOR	含有步驟或專案的選取元名稱。
SERVER	含有步驟或專案的選取元名稱。
SID	用於步驟，指定步驟 ID 號碼。用來建構回到管理主控台的鏈結以存取報告。
SRVRHOST	用於步驟，包含步驟的伺服器 TCP/IP 主機名稱。
START	包含啟動工作的日期/時間。
STEPNAME	用於步驟，包含步驟的名稱。
STEPNORMALLOG	用於步驟，顯示工作中現行步驟的日誌資訊，並排除出現在詳細資料日誌中的環境設定動作。
TAG	包含工作的標籤字串。與 \$BF_TAG 值相同。
TAILNORMALLOG	用於步驟時，作用如同 STEPNORMALLOG，但只會顯示日誌結尾。 用於建置時，作用如同 FULLNORMALLOG，但只會顯示每個步驟的日誌結尾。 所顯示的行數由郵件範本的結尾日誌數量系統設定控制。
USEREMAIL	包含工作/事件擁有者的電子郵件位址。
USERNAME	包含工作/事件擁有者的完整名稱。

列入專案的通知

系統處理列入專案通知的方式，就好像列入專案的步驟是列入在執行呼叫端專案中：

- 當專案包含列入專案時，列入專案的專案層次通知設定便不予處理。當專案 A 列入專案 B 時，就不會傳送任何有關專案 B 的啟動、通過或失敗的訊息。
- 步驟層次通知設定不會受到影響。不管步驟是在最上層專案或是列入專案中，只要該步驟具有通過或失敗通知存取群組集，就會傳送適當的訊息。
- 列入步驟可以判斷執行呼叫端專案成功還是失敗。舉例來說，列入步驟失敗可能會導致呼叫端專案失敗，如果該步驟設為「失敗時繼續」，則將呼叫端專案的狀態變更為「失敗但繼續」。

使用 Snapshot 為專案建立新的實例

進行專案的 Snapshot，可以快速建立您想變更或修改之專案的新實例。專案 Snapshot 是一個獨立且可執行的專案。您也可以使用 Snapshot 為程式庫建立新實例。

專案 Snapshot 概觀

請檢閱這些主題，以瞭解專案 Snapshot 及其用法。

專案 Snapshot 使用案例

下列範例說明專案 Snapshot 的一些一般使用案例：

- 建立專案的 Snapshot，以變更專案的配置或執行新工具或 Script 的測試，而同時間還能繼續使用現有的專案來執行工作。
- 將專案的 Snapshot 儲存成暫時備份或正式保存的一部分。
- 建立專案的 Snapshot，以擷取和某個里程碑（例如：外部或內部發行）相對應的某個時間點專案配置。

專案 Snapshot 的概念和詞彙

在使用者介面中，Snapshot 引進某些配合專案使用的新概念和詞彙。

專案 Snapshot：Snapshot 是現有專案的新實例。關於 Snapshot，需要切記的一些關鍵點如下所示：

- Snapshot 是一個個別的專案。變更 Snapshot 集當中的某個 Snapshot，並不會影響集合中的其他 Snapshot。
- Snapshot 是一個可執行的專案。它會搭配您在建立專案 Snapshot 時選取也要建立 Snapshot 的物件一起執行，或搭配程式碼專案（亦稱為基礎 Snapshot）相關聯的物件一起執行。
- Snapshot 不是副本。

如果您針對專案的相關聯物件建立 Snapshot，則會為該物件建立個別的實例。當複製專案時，是複製物件之間的關係，而不是建立選取元、環境，或是列入或鏈結的專案等的新實例。


- Snapshot 不是專案的修訂：
 - Snapshot 不支援比較兩個專案 Snapshot 間的變更。

- 不同於來源控制系統，專案 Snapshot 的變更並不會以版本號碼來追蹤或識別。不過，您可以使用包含版本號碼（例如，7.5.0、3.4.01）的 Snapshot 命名方法，讓專案 Snapshot 和里程碑產生關聯。


Snapshot 集：Snapshot 集是一組作為一個基礎 Snapshot 之後代的所有專案 Snapshot。此集合中至少含有一個基礎或母項 Snapshot，以及一個子項 Snapshot。在使用者介面中，專案名稱旁邊的 **Snapshot** 圖示  表示已為專案建立 Snapshot 集。

基礎 Snapshot：一開始，所有專案的 Snapshot 名稱皆為「基礎 Snapshot」。您可以將「基礎 Snapshot」變更為另一名稱。基礎 Snapshot 是 Snapshot 集的母項。

預設專案 Snapshot：預設專案 Snapshot 是目前正在運作的專案。Snapshot 集當中只能有一個 Snapshot 作為預設值。如果您沒有指定預設 Snapshot，就會以基礎 Snapshot 作為預設值。

- 在使用者介面中，預設 Snapshot 會顯示在專案清單的最上層。選取**專案或工作 > 啟動**，即會顯示專案清單。
- 在您選取以 Snapshot 作為列入專案或鏈結專案的專案時，除非您在清單框中選取不同的專案 Snapshot，否則會使用預設專案 Snapshot。
- 如果要存取和處理專案 Snapshot 集當中的其他 Snapshot，您必須按一下 **Snapshot** 圖示 .

專案 Snapshot 視圖

選取 **Snapshot** 圖示 ，以顯示 Snapshot 視圖。在使用者介面中，Snapshot 視圖會顯示 Snapshot 在集合中的階層：

- 基礎 Snapshot 位於最上層，如果您沒有另外指派一個唯一名稱，其名稱會是「基礎 Snapshot」。
- 所有的專案 Snapshot 皆為基礎 Snapshot 的子項。相同基礎 Snapshot 的子項在 Snapshot 直欄中全縮排在同一層。
- 從子項 Snapshot 建立而來的專案 Snapshot，會成為該子項 Snapshot 的子項，並在 Snapshot 直欄中縮排在下一層。

專案 Snapshot 規劃

檢閱某些最佳實務，以選取預設專案 Snapshot，並為專案 Snapshot 命名。

- 用來在集合中選取預設 **Snapshot** 的策略

使用者介面只能辨識 Snapshot 集的一個預設或現行專案 Snapshot。在選取預設 Snapshot 時，請採取一致的策略：

- 使用基礎 Snapshot 作為預設 Snapshot

採取這種策略時，是將 Snapshot 建立成時間點備份，且不會對所備份的專案 Snapshot 進行任何變更。您要變更的是基礎 Snapshot，並且只使用基礎 Snapshot 專案繼續執行工作。

- 使用最新 Snapshot 作為預設 Snapshot

如果採取這種策略，在您建立新專案時，會認定您有意讓它成為新的預設專案 Snapshot。您不會變更基礎 Snapshot 或更早的專案 Snapshot。最新的 Snapshot 就是用來執行工作的 Snapshot。

- **識別 Snapshot 集的 Snapshot 命名方法**

此專案 Snapshot 名稱在專案 Snapshot 集當中必須是唯一的。

請使用下列一種準則，來協助您建立專案 Snapshot 名稱：

- 名稱應具備敘述性：應指出 Snapshot 的用法或用途。
- 命名方法應遵循定義的標準。您可以使用 Snapshot 標籤上的「註解」框，來說明命名方法。

- **讓集合使用單一專案名稱**

在您建立專案 Snapshot 後，可讓您選擇變更專案的名稱。如果您變更專案名稱，會更新每一個專案 Snapshot 的這個名稱。

專案 Snapshot 選項

在您建立 Snapshot 時，您必須選取要併入至 Snapshot 的物件。下表說明可供 Build Forge 物件使用的選項。此表格有針對下列物件的直欄：

- 自動併入至專案 Snapshot 的物件。
- 選擇性建立和併入的物件（如果您在建立專案 Snapshot 時有選取它們的話）。

對於這些物件，會在使用者介面中建立一個新物件，且其 Snapshot 名稱和專案 Snapshot 相同。例如，如果您專案 Snapshot 的名稱為 release_7.1，則環境、選取元、列入專案或程式庫，以及鏈結的專案或程式庫等的 Snapshot 名稱，也會是 release_7.1

- 物件不會併入至專案 Snapshot；您必須手動建立這些物件，然後新增至專案。

	選擇性地隨專案 Snapshot 併入	
自動併入至專案 Snapshot	* 僅複製，而不會建立另一個實例	不隨專案 Snapshot 併入
專案步驟及其設定（日誌過濾器、通知群組等）	專案和其步驟的環境	專案附註
專案標籤	「併入」類型的環境變數為 Snapshot 環境所新增的環境	
	列入專案（或程式庫）與其步驟	
	鏈結的專案（或程式庫）與其步驟	
	專案與其步驟的選取元	
	「併入」類型的選取元內容為 Snapshot 選取元所新增的選取元	
	* 專案暫存器（複製的）	
	* 專案標籤變數值（複製的）	
	* 通知範本（複製的）	

自動併入至專案 Snapshot	選擇性地隨專案 Snapshot 併入	不隨專案 Snapshot 併入
	* 僅複製，而不會建立另一個實例	
	* 配接器鏈結（複製的）	

驗證與編輯 Snapshot 許可權的存取群組

驗證使用者具備建立 Snapshot 以及設定預設 Snapshot 所需的許可權。若無，請使用存取群組，指派許可權給使用者。

許可權是透過存取群組指派給使用者，而存取群組可由 Build Forge 提供或由 Build Forge 管理者來建立。

如果要驗證並編輯指派至 Snapshot 許可權的存取群組，請執行下列步驟：

1. 選取**管理 > 許可權**。
2. 在許可權清單中，選取**顯示全部**，以列出所有許可權。
3. 驗證有權存取下列 Snapshot 許可權的存取群組與使用者無誤：

建立 Snapshot	建立專案、環境和選取元的 Snapshot 時，所需的使用者許可權。
設定預設 Snapshot	設定或變更專案、環境和選取元的預設 Snapshot 時，所需的使用者許可權。

從現有的專案或專案 Snapshot 來建立專案 Snapshot

建立專案 Snapshot 時，將會建立此專案的新實例，以及您選擇要 Snapshot 的物件。Snapshot 並非副本；它是一個可執行的新專案實例。

註：管理主控台不會顯示前 2999 個專案之後的專案 Snapshot。

1. 按一下您要建立 Snapshot 之專案或專案 Snapshot 旁的**編輯**圖示：
 - 如果要建立預設專案 Snapshot 的 Snapshot，請在專案清單（**專案**）中，按一下最上層 Snapshot 旁邊的**編輯**圖示 。
 - 如果要建立非預設專案 Snapshot 的 Snapshot，請按一下 **Snapshot** 圖示 。Snapshot 視圖會顯示集合中的專案 Snapshot。按一下非預設專案 Snapshot 旁邊的**編輯**圖示 。
2. 按一下**建立新 Snapshot**。
3. 在 Snapshot 標籤的**名稱**中，輸入 Snapshot 名稱。

此名稱在專案 Snapshot 集當中必須是唯一的。名稱會指派給隨專案一起建立 Snapshot 的所有物件。

4. 選取當您建立專案 Snapshot 時，要建立 Snapshot 的 Build Forge 物件。下表說明您所能選取的物件。



物件	說明
預設值	在使用者介面中，預設專案 Snapshot 會顯示在專案清單的最上層。 選取 專案或工作 > 啟動 ，即會顯示專案清單。
併入專案環境	為專案和專案中的步驟環境，建立 Snapshot。
遵循環境併入項目	如果選取「併入專案環境」，也可以為「併入」類型之環境變數所併入的其他環境，建立 Snapshot。
併入專案選取元	為專案以及併入在專案中的步驟選取元，建立 Snapshot。
遵循選取元併入項目	如果選取「併入專案選取元」，也可以為「併入」類型之選取元內容所併入的其他任何選取元，建立 Snapshot。
複製專案配接器鏈結	複製配接器鏈結，以作為 Snapshot 的一部分。 配接器鏈結會將配接器新增至專案。配接器會當成專案中的第一個步驟（步驟 0）執行。
複製專案暫存器	複製專案暫存器，以作為 Snapshot 的一部分。
複製專案標籤變數值	複製專案標籤變數的標籤值。標籤變數會自動複製，但其值不會。如果您沒有複製標籤值，會將它們重設為 1。
複製專案範本	針對在專案層次和步驟層次設定的通過與失敗通知事件，複製其通知範本。
併入鏈的專案	為鏈結專案（或程式庫）以及他們在專案層次或步驟層次被參照的步驟，建立 Snapshot。 鏈由專案的通過/失敗條件，或由步驟的通過/失敗條件來觸發。
併入專案列入項目	為列入專案（或程式庫）以及它們在專案層次或步驟層次被參照的步驟，建立 Snapshot。 列入項目是由步驟觸發，並在步驟完成之後執行。

- 按一下**儲存**，以儲存專案 Snapshot。

變更預設專案 Snapshot

預設專案 Snapshot 是專案 Snapshot 集當中的最上層 Snapshot，它會顯示在專案清單中（**專案**）。

如果要變更預設專案 Snapshot，請編輯您要作為新預設值之 Snapshot 的 Snapshot 定義：

- 選取**專案**。
- 在專案清單中，按一下預設專案 Snapshot 的 **Snapshot** 圖示 。
- 在 Snapshot 清單中，按一下要作為新預設專案 Snapshot 的**編輯**圖示 。
- 按一下**設為預設值**。
- 重要事項：**在蹦現式畫面中，選擇「確定」或「取消」。



確定	更新參照： 針對參照先前預設值的任何物件，將參照從先前的預設專案 Snapshot 更新為新預設值。
取消	不更新參照： 針對參照先前預設值的任何物件，不將參照更新為新的預設專案 Snapshot。

變更專案 Snapshot 的 Snapshot 名稱

您可以變更專案 Snapshot 的 Snapshot 名稱，也可以針對當初您建立專案 Snapshot 時選取要建立 Snapshot 的物件，變更其 Snapshot 名稱。

在基礎 Snapshot 方面，您可以使用這個選項，僅針對單一專案 Snapshot 或針對所有現行和未來的專案，將其「基礎 Snapshot」預設名稱變更為另一個 Snapshot 名稱。

如果要變更 Snapshot 名稱，請執行下列動作：

1. 選取專案。
2. 在專案清單中，按一下預設專案 Snapshot 的 **Snapshot** 圖示 。
3. 在 Snapshot 清單中，按一下專案 Snapshot 的編輯圖示 。
4. 選取 **Snapshot** 標籤。
5. 在名稱中，輸入新名稱。
6. 選用：在註解中輸入一項註解。
7. 重要事項：在蹦現式畫面中，選擇「確定」或「取消」。

確定	<p>變更專案 Snapshot 名稱和其他 Snapshot 物件名稱：針對當初您建立專案 Snapshot 時選取要建立 Snapshot 的物件，變更這些物件和專案 Snapshot 的名稱。</p> <p>在「基礎 Snapshot」方面：變更所有現行專案 Snapshot 和所有未來專案 Snapshot 的「基礎 Snapshot」名稱。</p>
取消	<p>變更專案 Snapshot 名稱，但不變更其他 Snapshot 物件名稱：在當初您建立專案 Snapshot 時選取要建立 Snapshot 的物件方面，不變更這些物件的名稱。僅變更專案 Snapshot 的名稱。</p> <p>在「基礎 Snapshot」方面：維持所有現行專案 Snapshot 和所有未來專案 Snapshot 的「基礎 Snapshot」名稱。</p>

存取和檢視專案 Snapshot 集當中的 Snapshot

建立專案 Snapshot 會建立一個 Snapshot 集，其中至少含有基本專案和新專案 Snapshot。

若要檢視 Snapshot 集中的所有專案 Snapshot，請執行下列動作：

1. 選取專案。

專案清單顯示專案和專案 Snapshot 清單。最上層 Snapshot 是預設專案 Snapshot。

2. 按一下 **Snapshot** 圖示 ，以顯示 Snapshot 集中的專案 Snapshot。

在 Snapshot 視圖中，您可以執行下列動作：

- 建立新的專案 Snapshot：按一下編輯圖示 。
- 變更專案的預設 Snapshot：按一下編輯圖示 ，然後按一下設為預設值。
- 以編輯標準專案的方式，來編輯專案 Snapshot 定義。

啓動預設專案 Snapshot 的工作

您可以使用「快速啓動」圖示或「啓動專案」頁面來啓動預設專案 Snapshot。

若要使用「快速啓動」圖示來啓動預設專案 Snapshot，請執行下列動作：

1. 選取專案。
2. 在專案清單中，按一下**快速啓動**圖示 。

在執行工作時，會使用選取元、類別、標籤格式和環境的預設值。

如果要使用「啓動專案」頁面來啓動預設專案 Snapshot，請執行下列動作：

1. 選取**工作 → 啓動**。
2. 在專案清單中，按一下預設 Snapshot 的專案名稱，以顯示其「啓動專案」頁面。

在「啓動專案」頁面中，您可以變更環境變數、選取元、類別和標籤格式。

如果選取元具有選取元 Snapshot，Snapshot 欄位下方的欄位會列出可讓您選取的選取元 Snapshot。

註：如果要快速選取預設選取元 Snapshot，請選取「預設 Snapshot」。「預設 Snapshot」對映至預設選取元 Snapshot 的名稱。

3. 按一下**執行**。

啓動非預設專案 Snapshot 的工作

您可以使用**快速啓動**圖示或啓動專案頁面來啓動非預設專案 Snapshot。

若要使用「快速啓動」圖示來啓動非預設專案 Snapshot，請執行下列動作：

1. 選取專案。
2. 在專案清單中，按一下基礎 Snapshot 的 **Snapshot** 圖示 。

Snapshot 視圖會顯示集合中的專案 Snapshot。

3. 按一下非預設專案 Snapshot 旁邊的**快速啓動**圖示 。

在執行工作時，會使用選取元、類別、標籤格式和環境的預設值。

若要使用「啓動專案」頁面來啓動非預設專案 Snapshot，請執行下列動作：

1. 選取**工作 → 啓動**。
2. 在專案清單中，按一下基礎 Snapshot 的 **Snapshot** 圖示 。

Snapshot 視圖會顯示集合中的專案 Snapshot。

3. 按一下非預設專案 Snapshot 的專案名稱，以顯示其「啓動專案」頁面。

在「啓動專案」頁面中，您可以變更環境變數、選取元、類別和標籤格式。

如果選取元具有選取元 Snapshot，Snapshot 欄位下面的欄位會列出可供您選取的選取元 Snapshot。

註：選取**預設 Snapshot** 可快速選取預設選取元 Snapshot。「預設 Snapshot」對映至預設選取元 Snapshot 的名稱。

4. 按一下**執行**。

刪除專案 Snapshot

使用**刪除專案** 或**除去專案** 選項，可以刪除專案 Snapshot。

- 如果專案 Snapshot 沒有任何工作，且其他物件不會參照專案 Snapshot，則可以使用「刪除專案」選項。類別或其他專案可以參照專案 Snapshot 作為列入專案或鏈結專案。
- 「除去」選項會將專案和其工作從 Build Forge 資料庫中刪除，並移除其他物件對該專案的任何參照。

若要刪除專案 Snapshot，請執行下列動作：

1. 選取**專案**。

2. 在專案清單中，按一下基礎 Snapshot 的 **Snapshot** 圖示 。

Snapshot 視圖會顯示集合中的專案 Snapshot。

3. 按一下要刪除的專案 Snapshot 旁邊的**編輯**圖示 。

4. 按一下**除去或刪除專案**。

第 20 章 使用步驟

這個主題說明如何在「管理主控台」中建立和管理步驟。

關於步驟

步驟是專案的一個元件。將專案當成工作執行時，會依序執行每個步驟。步驟包含一個以上指令且具有會影響其行為的步驟內容。

關於「步驟」畫面

「詳細資料」標籤

步驟內容指定如何執行步驟、處理其輸出，以及步驟完成時要執行的動作。步驟也可執行其他專案或程式庫。

如果要檢視步驟內容，請在專案中選取一個步驟。依預設，會顯示「詳細資料」標籤。其中顯示步驟的內容。

如果步驟內容未明確設定，其值會繼承自專案。設定給步驟的步驟內容會置換繼承值。

步驟內容包含：

名稱 步驟的名稱。這是作為步驟在系統和日誌中的標籤。

作用中 指定是否執行步驟。依預設，步驟為「已啟用」。如果要阻止執行步驟，請選取「已停用」。停用的步驟就無法用來在工作中執行。

目錄 設定步驟指令的執行位置。系統會自動為每一項工作建立唯一目錄。目錄欄位便於您在專案於工作期間建構的目錄中執行指令。（Build Forge 不會建構目錄欄位所提及的目錄）

路徑 指定目錄是絕對或相對路徑。

- **相對**：在將伺服器、專案、工作和步驟目錄加在一起之後所找到的路徑中，執行步驟指令。
- **絕對**：在將伺服器和步驟目錄加在一起之後所找到的路徑中，執行步驟指令。這個選項可讓您存取不在專案目錄結構中的目錄。範例：可用來啟動永久安裝在伺服器上的應用程式。

步驟類型

決定如何執行步驟。此內容影響「指令」的內容，以及「列入」中指定的專案（若有的話）。

- **一般**：執行步驟一次。
- **條件式**：如果「條件」內容中的表示式求值為 `true`，則執行步驟一次。選取條件式時，會顯示「條件」、「Else 列入」和「Else 指令」內容。如果「條件」內容求值為 `false`，就不執行「指令」和「列入」。但如果指定了「Else 指令」和「Else 列入」，就會執行它們。

- **While 迴圈**：可以多次執行步驟。它會持續執行，直到「條件」內容中的表示式為 `false` 或達到反覆運算數上限為止。選取「**While 迴圈**」時，會顯示「條件」和「反覆運算數上限」內容。

While 迴圈每一次執行反覆運算時都會對選取元求值，以決定要用來進行反覆運算的伺服器。

列入 指定要列入現行專案中執行的專案或程式庫。專案或程式庫中的步驟，可以使用現行專案的環境和大部分內容來執行。但是，系統會將列入專案的選取元，作為列入步驟的預設選取元。其行為就像是將指定專案中的步驟複製到現行步驟後面。

存取 選擇一個存取群組，以定義容許哪些使用者使用該步驟。您可以使用這個內容來限制對專案內特定步驟的存取權。如果使用者不是某步驟的「存取群組」成員，當他啟動內含該步驟的專案時，就會跳過該步驟。

選擇「專案預設值」會使步驟繼承專案的存取內容。

步驟提供者

步驟指令執行的實作。預設步驟提供者為「**MJC 步驟提供者**」。在端點代理程式機器上執行指令文字時，它會提供符合預期的舊式行為

反覆運算數上限

只有在「步驟類型」為「**While 迴圈**」時才會顯示。指定步驟在迴圈中所能執行的反覆運算數上限。系統強制的預設值為 100。步驟在步驟日誌中會顯示成順利完成（通過）。使用**一旦抵達上限，就讓步驟失敗**，可在抵達「反覆運算數上限」時讓步驟失敗。

在執行工作時，唯讀變數 `BF_ITERATION` 含有已順利進入的反覆運算數。如果工作停止再重新啟動，就會從 `BF_ITERATION` 中的反覆運算重新啟動。

達到上限時則步驟失敗

若為「是」，則一旦抵達「反覆運算數上限」，**While 迴圈**步驟就會失敗。若為「否」，就讓步驟順利通過。

Else 列入

只有在「步驟類型」為條件式時才會顯示。指定如果指定的條件為 `false`，則專案將列入執行。預設值為「否」。

指令

一個以上的指令。指令可以是作業系統指令、點指令或這兩者的組合。請參閱第 328 頁的『步驟如何執行』。

條件 只有在您選取的步驟類型為「條件式」或「**While 迴圈**」時才會顯示。

- 條件式：如果條件求值為 `true`，則執行指令。
- **While 迴圈**：只要條件求值為 `true`，就可以多次執行指令。您可以使用反覆運算次數上限來設定限制。

條件可以是要在所選伺服器資源上執行的函數或指令。

- 如果使用函數，就必須用在「條件」欄位的開頭。它採用 **Build Forge** 引擎來求值。不會傳送給伺服器資源。如需函數清單及其用法指示，請參閱第 333 頁的『條件函數』。
- 指令 是在選取的伺服器上執行。在這裡使用的任何指令，必須適用於代理程式的 **Shell** 環境。執行後所產生的回覆碼會決定條件通過或失敗。

專案的 Build Forge 變數可用在條件表示式中。如需如何表示與求值變數的相關資訊，請參閱第 265 頁的『解譯步驟中的變數』。

Else 指令

只有在您選取的步驟類型為「條件式」時才會顯示。指定條件求值為 false 時要執行的指令。

環境

指定在執行指令之前，所要套用的環境。這個環境中的值，會置換繼承自伺服器環境、專案環境和步驟變數的任何值。

選取元 指定一個選取元，以用來選擇這個步驟的伺服器。如果保持使用**預設值**，則步驟會在專案選取元所決定的伺服器上執行。

廣播 當勾選此項時，會在符合現行選取元的**每一個**伺服器上執行步驟；所謂的現行選取元是指步驟選取元（若有指定的話），如果沒有指定，則為專案選取元。在執行時期，系統會以一系列的步驟（每部伺服器一個步驟）來取代廣播步驟，然後視廣播步驟的**執行緒**內容而定，以序列方式或平行地執行這些步驟。

*重新啟動時的廣播步驟行為：*當廣播步驟重新啟動時，它不會播送。該設定只適用於步驟的新啟動。重新啟動時，引擎會為步驟隨機挑選單一伺服器。

逾時（分鐘）

指定系統應該等待現行指令產生輸出的分鐘數（預設值為 5 分鐘）。值 0 表示如果步驟適當地連接至代理程式，步驟就不會逾時。只要一達到逾時值，系統就會讓步驟失敗。除非將步驟設為「失敗時繼續」，否則專案也會失敗。

結果 「結果」內容決定系統如何判斷步驟成功或失敗。使用「結束碼」的預設值，根據指令 Shell 傳回的結束碼來判斷成功與否。您也可以選擇一項「日誌過濾器」，來檢查指令的輸出。如果要選取「日誌過濾器」，您必須先建立它。

失敗時 指定一旦步驟失敗，則要中止或繼續工作。依預設，系統會中止工作。

執行緒 若為「是」，則讓這個步驟與其他步驟平行執行。將這個內容設為「是」，以容許此步驟進行執行緒作業（與其他步驟平行地執行此步驟）。如果您要避免執行緒作業，請將內容設為「否」。如果您要區隔步驟的執行緒區塊，請將內容設為「合併」。您必須先完成第一組步驟之後，才能啟動「合併」步驟之後的下一組含執行緒作業的步驟。

通過通知

指定一旦步驟通過，則要通知的存取群組。

通過鏈 指定一旦現行步驟通過，則要啟動的專案。（具有「警告」狀態的步驟視為通過，因此將會啟動通過鏈）。

通過等待

若有勾選，則系統會暫停現行專案，直到通過鏈結的專案完成為止。如果取消這個步驟（或其專案），也會一併取消鏈結的專案。如果未勾選，則會非同步地啟動鏈結的專案，現行專案則會繼續進行下一個步驟。

失敗通知

指定一旦步驟失敗，則要通知的存取群組。

失敗鏈 指定在現行步驟失敗時要啟動的專案。（設為失敗時繼續的步驟視為失敗，因此將會啟動指派給步驟的任何失敗鏈。）

失敗等待

若有勾選，則系統會暫停現行專案，直到失敗的鏈專案完成為止。如果取消這個步驟（或其專案），也會一併取消鏈結的專案。


「附註」標籤

附註標籤含有步驟的相關附註清單（附有時間戳記）。您將手動建立附註。它不會自動記錄對步驟本身所做的編輯。標籤顯示目前的附註數目，例如**附註 (2)**。


如果要新增附註，請執行下列動作：

1. 按一下**附註**標籤。
2. 在文字欄位中撰寫新附註。
3. 按一下**提交**。

如果要編輯附註，請執行下列動作：

1. 按一下**附註**標籤。
2. 按一下您要編輯的附註旁的**編輯**圖示 。進行編輯。
3. 按一下**提交**。

如果要刪除附註，請執行下列動作：

1. 按一下**附註**標籤。
2. 按一下**垃圾**圖示 。畫面上會出現提示，詢問您是否確定要刪除附註。
3. 按一下**確定**。

新增步驟

關於這項作業

這個程序會將步驟新增至專案結尾。如需如何插入步驟使它成為第一個步驟或在現行步驟之間插入步驟的相關資訊，請參閱第 327 頁的『其他步驟作業』。

程序

1. 選取**專案**，然後按一下專案名稱。系統會顯示已選取的專案的步驟清單。
2. 按一下主畫面頂端的**新增步驟**。系統會顯示空的步驟詳細資料畫面。
3. 輸入內容的值。**名稱及指令**為必要的。
4. 按一下**儲存步驟**。


編輯步驟

程序

1. 選取**專案**，然後按一下專案名稱。系統會顯示已選取的專案的步驟清單。
2. 按一下步驟名稱。系統會在畫面的下半部顯示步驟的內容。
3. 編輯內容。
4. 按一下**儲存步驟**。

停用步驟

程序

1. 選取**專案**，然後按一下專案名稱。系統會顯示已選取的專案的步驟清單。
2. 按一下步驟名稱前面的勾選框。當勾選某個步驟時 ，對專案的影響如下：
 - 當您將專案當成工作執行時，並不會執行該步驟。
 - 當您正常啟動工作時，該步驟在步驟清單中會呈現灰色：請選取**工作 > 啟動**，再按一下工作名稱，然後按一下**工作步驟**。您可以看見該步驟，但無法在啟動的工作中使其成為作用中步驟。

註：

您也可以停用步驟，方法是在該步驟的**詳細資料**標籤中，將**作用中**內容設為「已停用」。當您儲存該步驟時，其勾選框會呈現勾選狀態。

其他步驟作業

若要使用步驟，請選取**專案**，然後按一下專案名稱。系統會顯示已選取的專案的步驟清單。

按一下步驟名稱前的**動作**圖示 ，以顯示其他選項：

- **插入新步驟**：在選取的步驟上方插入步驟。
- **複製步驟**：複製步驟及其所有內容。名稱會變更為在步驟尾端加上一個數字。複製名為 **Build** 的步驟會產生名為 **Build COPY 0** 的新步驟。數字會自動設定。您可以複製到以下位置：
 - 頂端：清單開頭
 - 上方：現行步驟之前
 - 下方：現行步驟之後
 - 底端：清單尾端
- **移動步驟**：將步驟移至清單中的其他位置。您可以移至以下位置：
 - 頂端：清單開頭
 - 上移：向上移一個位置
 - 移至...：會有一個對話框要求您提供步驟號碼。步驟會移至該位置，而其他步驟號碼會視情況調整。
 - 下移：向下移一個位置
 - 底端：清單尾端
- **刪除步驟**

控制執行流程

在步驟中，有數項特性可用來控制專案中的執行流程：

- **列入**：使用步驟的「列入」內容，來指定專案或程式庫。在執行完這個步驟的指令之後，就會立即列入執行該專案或程式庫中的步驟。在步驟日誌中，列入專案或程式庫的步驟會縮排。

- **通過鏈與失敗鏈**：步驟可以有它自己完全不同於指定給專案的「通過鏈」與「失敗鏈」。
- **執行緒作業**：您可以平行執行標示為進行執行緒作業的步驟。請使用步驟的「執行緒」內容，將它標示為進行執行緒作業。
- **播送**：您可以在多部伺服器上執行標示為進行播送的步驟。請使用步驟的「廣播」內容。
- **條件式**：您可以將步驟設為只在條件為 `true` 時執行。您可以設定一組替代的指令與一個列入的專案或程式庫，以便在條件為 `false` 時執行。請將「步驟類型」內容設為條件式，並使用相關的「條件」與 `Else` 內容，來使用這項特性。
- **While 迴圈**：您可以在每當條件求值為 `true` 時，讓步驟在迴圈中執行。請將「步驟類型」內容設為「While 迴圈」，並使用相關的內容，來使用這項特性。
- **點指令**：`.run` 與 `.runwait` 指令會從步驟的指令啟動一個程式庫或專案。

複式執行流程的常見用法是工作最佳化，也就是說，只有在需要時才執行步驟。

在軟體建置工程環境中，工作最佳化可能意味著只依所需來建置部分的應用程式，而非建置整個應用程式。工作可就前次編譯的二進位檔來檢查原始檔狀態，只有在原始檔有所變更時，才會執行編譯。就複式應用程式來說，執行流程可以對模組相依關係以及原始檔狀態做出回應。

步驟如何執行

在步驟定義中，指令內容包含作業系統指令、點指令或這兩者的組合。

您可以在一個個別的步驟中執行多個指令。將個別指令放在不同行上加以分隔。

註：當您在步驟的**結果**內容中採用預設的「結束碼」設定時，整個步驟的成功或失敗，是依據步驟中最後一個指令所傳回的結束碼。如果要偵測任何指令中的失敗，請建立「日誌過濾器」，並在**結果**內容中指定其用法。

在系統執行步驟之前，會先建構步驟環境。變數是使用伺服器環境、專案環境及步驟環境中所指定的值來設定。請參閱第 263 頁的『環境繼承』。依預設，變數會經過剖析，然後可在指令中使用。請參閱第 265 頁的『解譯步驟中的變數』。

指定 Shell

您可以使用 `#!` 指引來指定要用來執行指令的 Shell。這在 Windows[®] 和 Linux[®] 及 UNIX[®] 系統上都有效（Windows 代理程式負責將指令傳至指定的直譯器）。在 Windows 上，如果要將步驟中的指令傳送至 `C:\perl\bin` 中的 Perl 副本，請使用 `#!C:\perl\bin\perl.exe`。如果您的 Windows 代理程式是搭配 Cygwin 使用，而您需要將指令引導至 Windows Shell `cmd.exe`，則可以使用下列字行，以利用 Windows 的隱含路徑：

```
#!cmd.exe /C
```

請注意，`/C` 是 `cmd.exe` 的必要選項，否則，它會在步驟指令傳給它之後等待其他指令。您可以在 UNIX 或 Linux 電腦上使用 `#!/bin/perl` 指令。

註：當您在 Linux 或 UNIX 系統上使用 `#!` 指令時，系統不會變更到標準預設目錄（利用伺服器路徑、專案、名稱和步驟路徑欄位的組合而建構的路徑），因為它無法預測所需的語法；您必須包括自己的目錄變更指令。使用系統建立的特殊環境變數，例如 `BF_SERVER_ROOT` 和 `BF_PROJECTNAME_PHYS` 來執行此動作。

系統如何將步驟分割成幾個部分

Rational® Build Forge® 會將步驟分割成幾個部分，每一個部分都是由一組作業系統指令或單一點指令所組成。例如，下列步驟有六個部分。

```
cmd1          # Part 1
cmd2
.dot_cmd1     # Part 2
cmd3          # Part 3
cmd4
.dot_cmd2     # Part 4
.sleep 30     # Part 5
.dot_cmd3     # Part 6
```

步驟的各部分會依序執行。每一個部分的環境會傳給下一個部分。當任何部分發生錯誤時，即會立即停止處理。

在舊版中，上方步驟的每一個部分都必須是單一步驟。

註：請勿在各個部分之間建立參照。請勿在各個部分之間建立跳躍，例如在一個部分中使用 GOTO，然後在另一個部分中使用其目的地標籤。

列入：併入專案或程式庫的步驟

使用步驟的「列入」內容來包含指定的專案或程式庫的所有步驟。

若要在步驟中使用列入，請將「列入」內容設為專案或程式庫的名稱。步驟執行時會發生下列事件：

1. 步驟執行「指令」內容中的一個以上指令。
2. 步驟執行「列入」內容中指定的專案或程式庫的步驟。

如果您要執行列入的步驟，但沒有使用「指令」內容，請在「指令」內容中使用 `.sleep 0`。

繼承呼叫端步驟的列入繼承

被呼叫的專案或程式庫的所有步驟都是在呼叫端步驟的環境定義中執行。列入的步驟繼承呼叫端步驟的環境。

但是，系統會將列入專案的選取元，作為列入步驟的預設選取元。

列入步驟的狀態對呼叫端步驟的狀態造成的影響

如果呼叫端步驟的指令通過，就會依照下列方式來考量列入之步驟的執行狀態：

- 如果步驟的指令失敗，工作會正常停止。不過，如果步驟的「失敗時」內容設為「繼續」，並且指定「列入」，則列入會執行。
- 如果所有步驟傳回的執行狀態都是「通過」，便將呼叫端步驟標示為「通過」。如果指定了呼叫端步驟的「通過鏈」，就執行這個鏈。
- 如果有任何列入的步驟傳回「失敗」執行狀態，便將呼叫端步驟標示為「失敗」。如果指定了呼叫端步驟的「失敗鏈」，就執行這個鏈。

這個行為讓您可以很容易地進入深層巢狀的列入和鏈，來追蹤執行狀態。

列入巢狀

當您列入專案或程式庫的步驟時，會以巢狀結構將被呼叫的步驟包含在呼叫端步驟中。

巢狀層次的上限是由系統設定**最大列入深度**來設定。預設值為 32。工作開始時間不會測試巢狀。如果執行中的工作超出限制，它會在超出限制的點上發生失敗。

巢狀層次也可能會受限於執行「管理主控台」的主機上可用的記憶體。

步驟的通過鏈和失敗鏈

個別步驟可以有「通過鏈」和「失敗鏈」。

步驟的「通過鏈」和「失敗鏈」與專案的「通過鏈」和「失敗鏈」，是各自獨立執行的。

「通過鏈」和「失敗鏈」內容會設為專案或程式庫的名稱。運作方式與專案的鏈集一樣。請參閱第 294 頁的『啟用等待時取消鏈結的專案』。

執行緒作業：平行執行步驟

執行緒作業可讓步驟在相同伺服器或不同伺服器上平行執行。執行緒作業是以步驟的「執行緒」內容設定來控制。依預設，「執行緒」內容會設為「否」。當專案中的各部分之間可獨立執行時，執行緒作業有助於縮短專案的執行時間。

如果有多個相鄰的步驟將「執行緒」內容設為「是」，則系統會試圖平行執行這些步驟。這類步驟可視為啟用執行緒，而每一個步驟可個別執行，同時間工作的其餘部分也會繼續進行。執行緒作業遵循以下的規則。

- 至少連續兩個步驟的「執行緒」內容必須設為 **Yes**，才會進行執行緒作業。一組循序的含執行緒作業的步驟稱為**執行緒區塊**。執行緒區塊可以繼續進入「列入項目」中的步驟。比方說，如果專案中的某個步驟含有「列入項目」，且這個「列入項目」的第一個步驟也是執行緒步驟，則這兩個步驟屬於相同的執行緒區塊。它們會同時執行。執行緒區塊跟著含執行緒作業的步驟，其中包含巢狀「列入」步驟，直到發現「合併」步驟或非執行緒步驟。當使用巢狀「列入」步驟時，請小心避免競爭狀況。相依於執行緒上層步驟的結果或資料的列入含執行緒作業的步驟，可能會造成競爭狀況。
- 遇到其「執行緒」內容設為**結合**的步驟，或遇到不含執行緒作業的步驟時，就會終止執行緒區塊。屆時，步驟的執行會再次變成循序的。
- 當系統遇到啟用執行緒的步驟時，它會試著啟動該步驟。如果接下來的步驟也採取執行緒作業，系統會試圖啟動該步驟，並繼續進行至下一個步驟，就這樣重複到已無啟用執行緒的步驟為止，或已達工作限制為止。如果專案的選取元指定伺服器儲存區，依照概念，工作限制會是儲存區中伺服器的工作限制總和。

註：含執行緒作業的步驟的開始時間，取決於支援執行該步驟之伺服器的可用性。如果無法啟動步驟，系統會等待然後重試。您無法明確控制先從哪些步驟開始。

- 步驟最後可能同時在一部伺服器（視該伺服器的容量而定）或數部伺服器上執行，此視符合選取元的伺服器數量而定。
- 如果要強制所有步驟同在單一伺服器上執行，請針對專案使用**固定**內容。

- 若有多個執行緒區塊，必須等第一個執行緒區塊完成後，下一個執行緒區塊才能啟動。

在下一個範例當中，您必須先完成步驟 2、3 和 4 之後，才能開始步驟 5 和 6。

專案	步驟的執行緒內容
步驟 1	否
步驟 2	是
步驟 3	是
步驟 4	結合
步驟 5	是
步驟 6	是
步驟 7	否

- 使用專案的**執行緒數上限**內容，來限制可同時執行的執行緒數目。每一個啟用執行緒的步驟和其列入專案（若有的話）都可能產生平行的程序。所有程序都計算在內，直到抵達母專案的上限為止。當達到**執行緒數上限**值時，系統就會停止啟動新的平行程序。它會等到專案的平行程序數目降到**執行緒數上限**值以下，才會再繼續。

將步驟廣播至多部伺服器

當您有一個活動可在許多伺服器上有效執行，則可使用廣播特性，在許多伺服器上重複相同的步驟。

通常一個步驟只會在一部伺服器上執行。然而，每個步驟都有一個**廣播**勾選框。勾選步驟的**廣播**方框時，系統會在執行時期，以一組非廣播步驟來取代該步驟，符合步驟選取元的每部伺服器各一組。

註：如果步驟的選取元僅符合一部伺服器，則該步驟只會執行一次。

廣播的可能用法包含：

- 重新啟動伺服器群組。
- 在伺服器群組上執行測試。
- 將同一組檔案複製到整個伺服器群組。
- 將同一組原始碼移出至多部伺服器，以單一、容易維護的步驟來為之後的個別作業做準備。

重新啟動廣播步驟

請參閱第 375 頁的『重新啟動廣播步驟』。

廣播步驟中的執行緒作業

在執行時期為廣播步驟建立取代步驟時，系統會將步驟執行緒化，如下所示：

- 如果廣播步驟的**執行緒**內容設為**否**，則取代步驟會取得相同的**執行緒**值，因而這些步驟都會依序執行。一個步驟必須完成之後，下一個步驟才能開始。

- 如果廣播步驟的**執行緒**內容設為**是**，則取代步驟也會取得相同的**執行緒**值。這會產生一組步驟，彼此之間平行執行，並且也會與廣播步驟之前或之後的任何含執行緒作業的步驟平行執行。
- 如果廣播步驟的**執行緒**內容設為**結合**，則系統會建立取代步驟，並將**執行緒**設為**是**，但最後一個步驟例外，它會標示為**結合**。這會產生一組步驟，彼此之間平行執行，並且也會與它們之前的任何含執行緒作業的步驟平行執行，不過整組步驟必須先完成，才能啟動廣播步驟之後的步驟。

從廣播步驟啟動其他的專案

您可以廣播含有列入專案的步驟，或是廣播當步驟通過或失敗時會鏈專案（**通過鏈/失敗鏈**）的步驟。當您廣播會啟動（鏈）另一個專案的步驟時，請注意，廣播步驟並不會置換所啟動之專案的選取元。一般來說，如果您的目的是要在符合廣播步驟選取元的每一部伺服器上啟動專案，當您從廣播步驟啟動專案時，請使用程式庫（沒有它自己的選取元的專案）。

如果您不使用程式庫，廣播步驟的每個副本就會在不同的伺服器上執行，但是列入或鏈結的專案會遵循它自己的選取元，而不會選擇與廣播步驟副本相同的伺服器。最後就是每個廣播步驟在不同的伺服器上執行，而列入專案中的所有步驟在相同的伺服器執行好幾次。

註：如果您的目的是要使用**廣播**，在符合選取元的每一部伺服器上各啟動一次程式庫，請務必也在程式庫上設定**固定**選項，使其所有步驟（使用預設專案伺服器）都在同一部伺服器上執行。

執行條件式步驟

條件式執行作業是實作步驟的 `if-then-else` 分支。

執行簡式 `If-Then`：

1. 將步驟類型設為「條件式」。
2. 將條件設定成一個可供求值的表示式。
3. 填寫要執行的指令。
4. 若有需要，可指定要執行的列入。（如果設定「列入」，則可讓「指令」維持空白。）

執行工作時，如果「條件」求值為 `true`，就會執行步驟。若有指定「列入」專案或程式庫，則也會執行。如果表示式求值為 `false`，則會跳過，且工作執行會繼續下一個步驟。

執行 `If-Then-Else`：

當「條件」傳回 `false` 時，如果您想執行不同的指令及/或列入，請填寫其他的內容：

- 填寫要執行的 `Else` 指令。
- 若有需要，可指定要執行的「`Else` 列入」專案或程式庫。（如果設定「`Else` 列入」，則可讓 `Else` 指令維持空白。）

在工作執行期間，如果順利求值條件，並且順利執行「指令」中的指令或 `Else` 指令，就會將步驟結果標示為「通過」。若想瞭解所採取的是哪條路徑，必須查看日誌才行。

另請參閱『條件函數』。

執行 While 迴圈

執行 While 迴圈時，可讓您根據條件來重複執行某個步驟。

如果要將步驟當成 While 迴圈來實作，請執行下列動作：

1. 將步驟類型設為「While 迴圈」。
2. 將「條件」設定成一個指令或可供求值的表示式。
3. 填寫要執行的指令。
4. 若有需要，可指定要執行的列入。（如果設定「列入」，則可讓「指令」維持空白。）
5. 若有需要，可將「反覆運算上限」設為您希望步驟執行的次數上限。請在開發期間使用這項限制，直到您認為指定的條件已能正確運作為止。預設值為 100。
6. 如果您希望一旦達到「反覆運算數上限」，步驟就該失敗，請將達到上限時則步驟失敗設為「是」。否則，達到最大反覆運算之後，將會通過該步驟。

步驟的每一個反覆運算會記錄在日誌中。會根據「結果」準則，將每一個反覆運算結果設為「通過」或「失敗」。

另請參閱『條件函數』。

條件函數

條件函數用於「條件」步驟內容和配接器 XML 元素的 condition 屬性中。

- 若為使用「條件」內容的步驟：如果在「條件」欄位開頭使用下列函數，則會交由引擎來求值，而除非條件求值為 true，否則不會傳送資訊給選取的伺服器。如果條件求值為 true，就會在選取的伺服器上執行步驟。

重要：

- 請勿試圖在伺服器資源之 Shell 環境中所設定的變數上，使用這些函數。由於求值是在 Build Forge 引擎上進行，因此它們只會對定義在步驟之 Build Forge 環境中的變數起作用。
- 請勿在要評估的字串中使用運算子字元。運算子字元如下：=, <, >, !
- 若為配接器範本：下列函數可用在具有 condition 屬性的配接器 XML 元素中。它們用來指定如何執行配接器。**重要事項：**在 condition 屬性中，必須用雙引號括住 condition 函數：condition="condition_function"。

可用的函數如下：

true(expression)

如果 expression 為 true，則傳回 true。

false(expression)

如果表示式為 false，則傳回 true。

contains(a,b)

如果字串 a 包含字串 b，則傳回 true。a 和 b 參數可以是文字字串或變數。不應使用引號括住文字字串。如果文字字串括上引號，則引號會變成所要求值的字串的一部分。

註：在 Rational Build Forge 7.1.2 版之前，如果字串 *a* 在字串 *b* 中，這個函數會傳回 `true`。

hastext(*var*)

如果變數不是空的，則傳回 `true`。*Var* 為設定在 Build Forge 中的一個變數。

isempty(*var*)

如果變數為空的，則傳回 `true`。*Var* 為設定在 Build Forge 中的一個變數。

a eq b

如果 *a* 等於 *b*，則返回 `true`。*a* 與 *b* 參數可以是 Build Forge 中的變數集或文字值。可以使用字元和數值類型。在參數和運算子之間，請使用一個空格。

a ne b

如果 *a* 不等於 *b*，則返回 `true`。*a* 與 *b* 參數可以是 Build Forge 中的變數集或文字值。可以使用字元和數值類型。可以使用字元和數值類型。在參數和運算子之間，請使用一個空格。

a contains b

如果在 *a* 字串中找到 *b* 字串，則傳回 `true`。文字字串不應括上引號。如果文字字串括上引號，則引號會變成所要求值的字串的一部分。可以使用字元和數值類型。在參數和運算子之間，請使用一個空格。

函數中的表示式

`true()` 與 `false()` 函數的 *expression* 參數可以使用下列的運算子：

a==b 測試是否相等。參數可以是字串或數字。參數可以是 Build Forge 中定義的文字或變數。

a eq b 測試是否相等。參數可以是字串或數字。參數可以是 Build Forge 中定義的文字或變數。在參數和運算子之間，請使用一個空格。

a!=b 測試是否不相等。參數可以是字串或數字。參數可以是 Build Forge 中定義的文字或變數。

a ne b 測試是否不相等。參數可以是字串或數字。參數可以是 Build Forge 中定義的文字或變數。在參數和運算子之間，請使用一個空格。

a>b 測試 *a* 大於 *b*。參數必須是數值。參數可以是 Build Forge 中定義的文字或變數。文字可以使用算術運算子，例如 `2+2`。

a<b 測試 *a* 不大於 *b*。參數必須是數值。參數可以是 Build Forge 中定義的文字或變數。文字可以使用算術運算子，例如 `2+2`。

a>=b 測試 *a* 大於或等於 *b*。參數必須是數值。參數可以是 Build Forge 中定義的文字或變數。文字可以使用算術運算子，例如 `2+2`。

a<=b 測試 *a* 不大於或等於 *b*。參數必須是數值。參數可以是 Build Forge 中定義的文字或變數。文字可以使用算術運算子，例如 `2+2`。

a contains b

測試是否在字串 *a* 中發現字串 *b*。參數可以是 Build Forge 中定義的文字或變數。不應用引號括住文字字串。

條件函數範例

在下表的範例中，將變數設定成如下：

- \$AVAL contains the value String.
- \$BVAL contains the value 3.

條件	求值為	Notes®
A String contains \$AVAL	TRUE	字串比較
A String contains "String"	FALSE	String 兩側的引號成為比較的一部分。
true(A String contains \$AVAL)	TRUE	字串比較
\$AVAL contains String	TRUE	字串比較
\$AVAL contains "String"	FALSE	String 兩側的引號成為比較的一部分。
contains(A String,\$AVAL)	TRUE	字串比較
true(A String contains "\$AVAL")	FALSE	\$AVAL 兩側的引號成為比較的一部分；"A String" 中的 "String" 部分沒有括上引號。
A String != \$AVAL	TRUE	字串比較
A String ne \$AVAL	TRUE	字串比較
false("Not Here" contains \$AVAL)	TRUE	測試字串比較
true(2+1 == \$BVAL)	TRUE	相等的數值表示式
false(2+2 < \$BVAL)	TRUE	不相等的數值表示式
\$AVAL eq \$AVAL	TRUE	測試字串比較
true(\$AVAL ne Linus)	TRUE	測試字串比較
true(\$BVAL > 2+2)	FALSE	(3 > 2+2) 不是 true
contains(Not Here, \$AVAL)	FALSE	測試字串比較

從步驟啟動專案

請利用 `.run` 或 `.runwait` 指令，從步驟啟動專案。

用這個方法啟動的專案會像鏈一樣運作。專案會使用本身的選取元和環境來執行。請參閱 第 361 頁的『`.run` 及 `.runwait`』。

自訂日誌輸出

這些主題提供一些方法，協助在步驟中使用指令的特性來自訂日誌輸出。

為步驟的日誌輸出建立標籤

建立標籤，以便在步驟日誌中，將步驟的輸入列在其自己的種類中。

開始之前

這項作業假設您已經建立選取元、伺服器和專案。它也假設您使用的伺服器是與接受 **echo** 指令的作業系統（例如 Windows、Linux 或 UNIX）搭配使用。

關於這項作業

您可以在輸出中任何一行的起始處，加入一個大寫的標籤。標籤會一直用到步驟結束或遇到新的標籤為止。

標籤具有下列語法：

- 只包含大寫字母，後面接著冒號（標籤不能包含空格、標點符號、數字或小寫字母。"SPACESHIPS:" 為有效的。"Space Ships:" 是無效的。）
- 至少有三個字元

您可以在步驟中使用 **echo** 指令來建立標籤。系統會將 **echo** 指令的第一個引數辨識為標籤（如果它遵循標籤語法）。

註：系統會將任何建置輸出中的行首文字辨識為標籤（如果它遵循語法），即使該文字不是 **echo** 指令的引數亦然。

如範例所示，您可以將標籤設為現有的輸出標籤名稱，也可以設為新名稱。

程序

1. 在專案中建立新的步驟。以本例來說，專案的名稱為 Say_hi。
2. 將步驟命名為 LabeledLogOutput。
3. 在「指令」欄位中輸入下列文字：

```
echo SPACESHIPS: Voyager I
echo Voyager II
echo EXEC: You can add text to existing categories as well
```
4. 執行專案。
5. 當工作完成時，請檢視日誌。

結果

請注意種類標頭中的 SPACESHIPS 勾選框，以及輸出中標籤名稱為 SPACESHIPS 的第 354 行與第 355 行。

```
79      7/9/10 10:24 AM EXEC      Locale set to 'English_United States.1252'
250     7/9/10 10:24 AM EXEC      Locale set to 'English_United States.1252'
349     7/9/10 10:24 AM EXEC      Performing variable expansion on command line
353     7/9/10 10:24 AM EXEC      start [C:\data\BuildForge\buildforge_projects\Say_hi\BUILD_1@RBF-14]
354     7/9/10 10:24 AM SPACESHIPS      Voyager I
355     7/9/10 10:24 AM SPACESHIPS      Voyager II
356     7/9/10 10:24 AM EXEC      You can add text to existing categories as well
357     7/9/10 10:24 AM EXEC      end [C:\data\BuildForge\buildforge_projects\Say_hi\BUILD_1@RBF-14]
```

註：您可以選取或清除 SPACESHIPS 勾選框，以顯示或隱藏種類。

註：將文字新增至 EXEC 種類時，務必要避免使用特殊字元（例如括弧）。在部分系統上，如果用引號括住，則這些用法都可行，但一般而言，最好只使用純文字。

以顏色或作用中鏈結來強調顯示步驟輸出

日誌檢視器可辨識輸出文字中的 [STATUS] 與 [URL] 指令。指令沒有區分大小寫。您可以在 **echo** 指令或任何會產生輸出的指令中使用它們。起始和結束標籤必須出現在輸出的同一行。

- `[STATUS=condition]` 與 `[/STATUS]` 標籤會標示要強調顯示的文字。*condition* 會依下列方式設定強調顯示的顏色：
 - 通過 - 綠色
 - 警告 - 黃色
 - 失敗 - 紅色
 - 執行 - 藍色
- `[URL]` 與 `[/URL]` 標籤會將文字標示成作用中的超鏈結。

`[status]` 的使用範例如下：

```
echo [STATUS=WARN]Access to source control timed out[/STATUS]
```

`[url]` 的使用範例如下：

```
echo See the support forums at [url]http://www.ibm.com[/url]
```

日誌會將文字顯示成指向所指 URL 的作用中超鏈結。當按一下它時，會開啓新的瀏覽器視窗或標籤，並顯示頁面。但請注意，符合部分 `[URL]` 位址的日誌過濾器會導致鏈結失效。

使用工作資料

這些主題提供一些方法，協助使用指令的特性來修改專案和工作。

在專案檔中內嵌建置號碼

您可以使用 `.strsub` 指令來交換檔案中的一個字串與另一個字串；一般用途是將標準記號取代成系統變數，例如提供現行工作號碼的 `$B` 變數。

您可以使用 `.strsub` 點指令，在程式檔中內嵌建置或版本號碼。在專案初期放置 `.strsub` 指令，後來的步驟就可以編譯包含已更新資訊的檔案。

例如，下列步驟設定內嵌建置號碼的專案：

1. 在專案的檔案中新增唯一的字串，如 `_BUILD_`。例如，修改 `README.TXT` 檔案並變更版本宣告如下：

```
Application version 5.0.123
Application version 5.0._BUILD_
```

2. 專案的初期步驟應該移出要處理的檔案。在移出 `README.TXT` 之後，新增一個步驟，以便將 `$B` 系統變數取代 `_BUILD_`。請使用下列指令：

```
.strsub _BUILD_ $B README.TXT
```

3. 執行專案並確認 `README.TXT` 檔案包含現行工作號碼。以專案的第三次執行而言，`README.TXT` 檔案應該包含這一行：

```
Application version 5.0.3
```

加強功能

您可以利用下列方式來改良此作法：

- 使用其他環境變數。例如，建立名叫 `$MAJORVERSION` 和 `$MINORVERSION` 的變數，並以下列方式使用它們：

```
.strsub _MAJORVERSION_ $MAJORVERSION README.TXT
.strsub _MINORVERSION_ $MINORVERSION README.TXT
```

- 啟動專案時更新環境變數。透過選取工作 > 啟動來啟動專案，您可以查看現行環境變數，並在啟動專案之前編輯它們的值。例如，您可以在工作中併入註解作為變數。對註解變數使用專案動作**必須變更**，強迫使用者在執行專案時輸入新值。

在工作期間變更建置標籤

您可以在工作期間使用 `.retag` 點指令，將標籤值設為全新的值；點指令的語法如下：

```
.retag <new tag value>
```

以下為簡單的使用範例：

```
.retag MyProject
```

也可能用於較複雜的情況中：

```
.retag Job_${B}_${BF_D}
```

此範例將標籤設為要使用執行增量和現行的日期系統變數。您可以將指令用於伺服器的指令直譯器以設定結果。如果要在點指令中使用指令，請以反引號或左引號 (```) 字元來括住指令：

```
.retag `hostname`
```

此範例將標籤設為在執行步驟的伺服器上，執行主機名稱指令的結果。

註：請不要混合使用指令的反引號格式和標準指派格式。

在工作期間變更環境變數值

您可以使用 `.set`、`.bset` 和 `.tset` 指令，從步驟中變更環境變數。這些指令會變更現有環境變數的值，如下所示：

- 使用 `.set` 指令，可變更環境的主要記錄。當系統執行專案時，會從主要記錄複製一份專案環境，並用它來作為專案預設值。其效果如下：
 - 如果 `.set` 指令修改了專案環境，後面使用該預設環境的步驟並不會看到變更，因為系統不會回頭參照主要記錄。
 - 如果您使用 `.set` 指令來修改環境，而後面的步驟直接要使用相同的環境，該步驟就會看到您所做的變更。當該步驟選取特定的環境時，系統會回到環境的主要記錄。只要步驟的環境設定不是「預設值」，即使指名的群組與專案預設群組相同，這樣仍然有效。
 - 工作結束之後，`.set` 指令所做的變更仍會持續。未來的工作會使用先前執行之 `.set` 指令所建立的值。

請使用下列基本語法：

```
.set env <EnvGroupName>[(<SnapshotName>)] "<VariableName>=<DesiredValue>"
```

- `.bset` 指令可用來在工作執行期間，新增或變更變數值。這些變更會在出現 `.bset` 步驟後的步驟中生效。它們也會影響工作中的其餘部分。

```
.bset env "<VariableName>=<DesiredValue>"
```

註：您在為 `.bset` 指令指定變數時，該變數不必存在，所以您可以在工作期間，使用 `.bset` 指令來建立新的變數，這一點與 `.set` 指令是不一樣的。在經過現行的工作之後，該變數的值不會留存。

- `.tset` 指令可用來在工作執行期間，新增或變更變數值。所做的變更會在現行步驟中生效。它們會影響步驟中的其他任何指令，以及指定給步驟的任何「列入項目」。在經過現行步驟之後，不會持續保存該變數的值。

```
.tset env "<VariableName>=<DesiredValue>"
```

註：和 `.set` 指令不同的是，當您設定 `.tset` 指令時，您針對 `.tset` 指令所指定的變數不一定要存在，因此您可以在工作期間，使用 `.tset` 指令來建立新的變數。

設定多個變數

您可以使用這些指令同時設定多個變數，方法是包含其他變數和值配對，並以空格區隔，如下例所示：

```
.set env MyGroup "X=5" "X2=45"  
.bset env "Y=7" "CompilerVersion=4.511"  
.tset env "Z=9" "Z2=54"
```

使用指令輸出來設定值

您可以將指令傳送至伺服器的指令直譯器，為 `.set` 或 `.bset` 指令產生變數值。如果要在點指令中使用指令，請以反引號字元 (```) 括住指令。例如，指令：

```
.set env SetupGroup "PerlVer=`perl --version`"
```

這個指令可以將變數 `PerlVer` 設為 `perl` 的輸出 `-- version` 指令。

變數只能儲存 256 個字元；如果指派了更多字元給變數，該值就會被截斷。

依預設，系統會指派以反引號括住的整個指令輸出給變數，但是您可以使用以方括弧括住的範圍指令，從指令輸出選擇您要將哪幾行指派給變數。範圍號碼是採用 0 索引的方式（第一行編號為零，第二行編號為 1，依此類推），從輸出指派輸出行。在下面的這個範例中，

```
.set env SetupGroup "WindowsIPinfo[0,5-8]=`ipconfig`"
```

變數 `WindowsIPinfo` 會收到 `ipconfig` 指令輸出的第 1 行以及第 6 行到第 9 行。

下面是所有選取單行、行組或兩者之組合的有效範圍修飾元：

[5]

[4-6]

[1,2,5,8-11]

系統以不區隔的方式組合字行；中間不加任何空格或分行字元。

註：請不要混合使用指令的反引號格式和標準指派格式。

使用暫存器

暫存器是一般用途的緩衝區，可供步驟用來儲存持續資料。一般暫存器的名稱可以是單一字元，或是以字母開頭的多字元。

基礎資料庫的區分大小寫性質可決定您建立暫存器名稱時可用的大小寫。

您可以將暫存器變數併入通知範本中；參照通知範本中的暫存器時，請使用 `${X}` 大括弧形式。參照空的暫存器會傳回空字串。

使用 `.push` 及 `.pop` 點指令來將資訊儲存在暫存器中，或是從暫存器擷取資訊。另請參閱 `.poptag` 指令（第 357 頁的『`.poptag`』），這個指令可以讓現行工作標籤等於暫存器的內容。

註：暫存器不能像變數一樣用在指令中。您必須先將暫存器的值取出至檔案之後，才能使用它。

表 11. 特殊暫存器

暫存器	包含
!	包含符合「失敗」過濾器型樣的指令輸出行。 註： 這個暫存器只在套用過濾器的步驟範圍內可以看見。在 Rational Build Forge 使用過濾器處理步驟之後，該步驟過濾器可能設定的暫存器內容就再也看不見。
@	包含符合「通過」過濾器型樣的指令輸出行。 註： 這個暫存器只在套用過濾器的步驟範圍內可以看見。在 Rational Build Forge 使用過濾器處理步驟之後，該步驟過濾器可能設定的暫存器內容就再也看不見。
=	指定工作的附註資料庫。可讓步驟將檔案中的資料當作附註來新增至工作。這個暫存器與其他暫存器不同： <ul style="list-style-type: none"> • 您只能寫入（推送）至這個暫存器；不能從中讀取。 • 推送至這個暫存器的資料一律是附加進去，而不會改寫之前的資料。 • 系統會隨著附加的資料來提供時間戳記和使用者 ID。如此可保留工作附註的審核追蹤。

專案暫存器

專案暫存器與一般暫存器不同。它們可以持續用於各項建置中，而您可以透過「管理主控台」介面來建立和檢視它們，因此很適合用來儲存某些類型的配置資訊。

例如，您可以將 IBM® Rational® ClearCase® 配置規格儲存為專案暫存器，然後讓步驟使用 `.pop -p` 指令，將規格解壓縮，並且將它搭配使用 `cleartool setcs` 指令，來配置您的建置。這樣，您就可以連同專案一起管理配置了。

即使專案暫存器名為 ALPHA，您也可以將內容完全不同的一般暫存器命名為 ALPHA。專案暫存器是另一組值。



您可以用兩種方式來建立和存取專案暫存器：

- 透過點指令（`.push` 和 `.pop`），並搭配 `-p` 選項。當您使用 `-p` 選項時，您的指令會參照專案暫存器，而不是一般暫存器。

例如，

```
.push -p ALPHA register.txt
```

指令會將 `register.txt` 檔的內容，置於名為 ALPHA 的專案暫存器中。

- 透過管理主控台介面。選取**專案**，然後按一下所需專案名稱旁的**編輯**圖示 。專案內容即會出現在畫面下半部；接著再按一下**暫存器**標籤，即會顯示專案的暫存器。此標籤會提供一個畫面，用來管理暫存器：
 - 如果要建立新的暫存器，請輸入名稱和內容，然後按一下**建立**。
 - 若要刪除暫存器，請在畫面右方的清單中，按一下該暫存器名稱旁的垃圾筒圖示。
 - 若要編輯暫存器，請在清單中按一下暫存器名稱旁的**編輯**圖示 。系統便會將暫存器的內容移入暫存器畫面中。完成變更之後，請按一下**儲存編輯的暫存器**按鈕。

只要有權存取專案的人，都可以檢視和編輯其專案暫存器。

註：基礎資料庫的區分大小寫性質可決定您建立暫存器名稱時可用的大小寫。

在步驟中將檔案複製到伺服器資源，以及從伺服器資源複製檔案

您可以使用點指令，將檔案從某個伺服器資源複製到另一個。這個主題說明如何使用 `.get` 與 `.put` 指令（針對單一檔案）以及 `.rget` 與 `.rput` 指令（用來複製整個目錄樹狀結構）。

重要：伺服器資源必須已啟用檔案複製功能。依預設，並不會啟用。請參閱『在伺服器資源上啟用檔案複製功能』。

在伺服器資源上啟用檔案複製功能

根據伺服器的預設值，並不容許使用點指令來複製檔案。如果要容許在專案與各伺服器資源之間複製檔案，請變更伺服器的**檔案**內容。

如果要變更設定，請執行下列動作：

1. 選取**伺服器** > <伺服器名稱>。
2. 在**檔案**內容中，選取「無」以外的值。您可以啟用從伺服器複製檔案，以及/或將檔案複製到伺服器中。

從伺服器取得檔案

如果要從伺服器取得檔案的副本，並放置在現行步驟工作目錄的相對目的地中，請使用 `.get` 指令。比方說，如果您有一個名為 `winbuildserver1` 的伺服器，且其 `config` 目錄中含有 `config.txt` 檔案，則您可以將下列步驟新增到您的專案中，以將檔案複製到現行伺服器的 `config` 目錄中：

```
.get winbuildserver1:./config/config.txt ./config/config.txt
```

如需相關資訊，請參閱：

- `.get` 的參照項目
- 第 359 頁的『`.rget`』的參照項目
- 第 384 頁的『工作的工作目錄』中有關如何解析步驟中之路徑的說明

將檔案放置在伺服器上

如果要將檔案從現行伺服器複製到不同的伺服器，請使用 `.put` 指令。下列範例步驟假設目前伺服器的 `config` 目錄中已有 `config.txt` 檔，並可從現行路徑存取：

```
.put ./config/config.txt winbuildserver1:./config/config.txt
```

如需相關資訊，請參閱：

- `.put` 的參照項目
- `.rput` 的參照項目
- 第 384 頁的『工作的工作目錄』中有關如何解析步驟中之路徑的說明

配置直接檔案傳輸

當 `_XSTREAM_PROTOCOL` 觸發程式變數位於工作環境中時，可啟用代理程式之間的直接檔案傳輸。傳送代理程式、接收代理程式及引擎都必須具有直接檔案傳輸的功能。有數個參數可控制編碼、網路及檔案傳輸性質。

請參閱 第 159 頁的『`bfagent.conf` 參照』參照中的第 276 頁的『觸發變數參照』及 `xstream_*` 參數。

疑難排解步驟的處理

如果您在步驟處理方面遇到問題，請檢閱這個主題中的資訊，以查看是否有令人滿意的暫行或徹底解決方法。

工作沒有處理 **ANT build** 指令後面的任何步驟指令

問題說明：

沒有處理 **ANT build** 指令後面的步驟指令。

在下列步驟範例中，**echo** 指令沒有執行。

```
<path to ant bin directory> ant -f <path to Java project>\build.xml build  
echo "Ant build complete"
```

說明：不論 **ANT build** 失敗或成功，**ANT build** 都傳回錯誤碼 1。

如果在步驟的「指令」內容中使用了多個指令，則只有執行的最後一個指令的結束狀態，才會影響步驟結果狀態。如果伺服器執行的步驟指令 **Script** 包含 **ANT build** 指令，則錯誤狀態 1 會導致 **ANT build** 後面的所有指令都失敗。

解決方案：

建立一項步驟日誌過濾器，來處理 **ANT build** 產生的步驟輸出。步驟日誌過濾器會設定步驟結果，且確保會處理工作中的下一個步驟。

1. **ANT build** 應該是步驟中的唯一指令，或是步驟中的最後一個指令。

如果沒有日誌過濾器，**ANT build** 會傳回錯誤碼 1，並將步驟結果設為失敗。

2. 建立一項日誌過濾器，以便在步驟輸出中搜尋適當的失敗字串 (**BUILD FAILED**)，並有效地控制步驟的處理。

若有找到字串，請使用「設定失敗」動作，將步驟結果設為失敗。當使用步驟日誌過濾器時，如果沒有找到該字串，步驟結果一律會設為通過。

如需設定步驟的日誌過濾器的詳細資料，請參閱第 299 頁的『日誌過濾器』。

沒有執行 Windows 批次指令後面的步驟指令

問題說明：

沒有處理 Windows 批次指令後面的步驟指令。

在下列的步驟範例中，第一個批次檔後面的兩個指令都沒有執行。

```
C:\script1.bat
C:\script2.bat
echo "Performed both batch commands"
```

說明： 步驟中的所有指令必須放在要由伺服器執行的 Windows 批次檔中。如果步驟指令含有批次檔的參照，在批次檔執行後，該步驟即會結束。批次檔參照後面的步驟指令則不會執行。

解決方案：

使用 `call` 指令來執行步驟中的批次檔。`call` 指令會在步驟批次檔內執行。

```
call C:\script1.bat
call C:\script2.bat
echo "Performed both batch commands"
```

點指令參照

您可以在步驟的「指令」欄位中，使用點指令。其提供系統中特殊能力及功能的存取權。

您可以將點指令與一般指令在步驟中混合使用，也可以在單一步驟中使用多個點指令。但是請不要在單一步驟中，使用多個 `.scan` 指令，假使您這麼做，系統無法準確地報告指令結果。

有另一份點指令的清單可供您搭配環境變數使用。請參閱第 273 頁的『在變數中使用點指令』。

點指令語法

語法規格：每一個點指令說明皆含有一項語法規格，所用的表示法如下所示：

- 使用者提供的值會顯示在角括弧中：`<value>`
- 選用文字會顯示在方括弧中：`[選用文字]`

除非另外特別註明，否則環境變數都可以當作指令參數使用。

如果點指令接受參數的環境變數，您也可以使用：

- backtick (```) 代表指令執行
- 方括弧 (`[` 和 `]`) 指出範圍（也稱為範圍表示法。）

請考量下列 `.set` 範例：

```
.set envgroupname "F00[1,3-5]=`cat foofile`"
```

在此案例中，代理程式會執行 `cat foofile`，從而列出 `foofile` 的內容。使用範圍表示法，會擷取第 1 行以及第 3 行到第 5 行。接著 `.set` 指令會使用這些行來更新名稱為 `envgroupname` 的環境群組內現有的變數 `'FOO'`。

.bom

```
.bom addcategory "category"
.bom setcolumn "category" "section" "column" [...]
.bom data category "section" "column=value" [...]
```

.bom 指令可新增資料至建置的「資料清單 (BOM)」。您可以用它來新增種類、區段以及資料。

種類 種類是列印在 BOM 中的標頭。請利用 addcategory 選項來指定它們。

```
.bom addcategory "category"
```

區段 區段定義了種類中的資料直欄。請利用 setcolumn 選項來指定區段及區段內的直欄。不會列印區段名稱。區段的直欄標頭集會列印在區段的開頭。您可以利用 -p 選項並識別母項區段來建立巢狀區段。

區段和直欄必須定義在試圖將資料加到其中的 data 選項之前。

```
.bom setcolumn "category" "section" "columnheader" [...]
.bom setcolumn "category" "section" -p parentsection "columnheader" [...]
```

資料 會將資料移入定義在區段中的直欄中。區段和直欄必須已利用 setcolumn 定義妥當。

```
.bom data category "section" "column=value" [...]
```

下列範例顯示指定種類、區段和直欄以及資料時必須遵循的順序。

```
.bom addcategory "Spaceships"
.bom setcolumn "Spaceships" "Section1" "ShipName" "WarpSpeed" "Tonnage"
.bom setcolumn "Spaceships" "Subsection1" -p "Section1" "ShippingDate" "ShippingManifest"
.bom data "Spaceships" "Section1" "ShipName=SpaceShipOne" "WarpSpeed=9" "Tonnage=10000"
.bom data "Spaceships" "Subsection1" "ShippingDate=123" "ShippingManifest=456"
.bom data "Spaceships" "Section1" "ShipName=Freighter" "WarpSpeed=6" "Tonnage=20000"
```

這個範例在 BOM 中顯示如下所示：

ShipName	WarpSpeed	Tonnage		
SpaceShipOne	9	10000		
			ShippingDate	ShippingManifest
			123	456
Freighter	6	20000		

至於其他點指令，您可以在指令中使用環境變數。像這個指令：

```
.bom data "Spaceships" "${SECTION}" "ShipName=${NAME}" "WarpSpeed=${SPEED}" "Tonnage=${TONNAGE}"
```

會以先前指令載入環境變數中的資料來植入 BOM。

您可以建立任何欄數，但是要在植入最後一個資料欄後，系統才會將資料行寫入 BOM。

如果您省略資料行中的某個直欄，系統就會使用上一列的值，如下列範例所示。

```
.bom addcategory "Spaceships"
.bom setcolumn "Spaceships" "Section1" "ShipName" "WarpSpeed" "Tonnage"
.bom data "Spaceships" "Section1" "ShipName=SpaceShipTwo" "WarpSpeed=3" "Tonnage=30000"
.bom data "Spaceships" "Section1" "ShipName=Tanker" "Tonnage=50000"
```

結果是 SpaceShipOne 中的 WarpSpeed 值會重複：

ShipName	WarpSpeed	Tonnage
SpaceShipTwo	3	30000
Tanker	3	50000

.bomexport

說明

.bomexport 點指令會將工作的 BOM 匯出至 XML 檔。 .bomexport 在收集 BOM 資訊之後，會將該資訊儲存至您所指定的檔案及位置。

路徑和檔名是選用的。依預設，Build Forge 會將 BOM 報告儲存至伺服器上的步驟工作目錄，並且使用標籤名稱作為檔名 (<build_tagname>.xml)。

請將 .bomexport 指令指定為專案的最後一個步驟。

註：在 Java MJC 中，bomexport 公用程式或 .bomexport 指令無法擷取工作的步驟結果及步驟日誌。然而，可以擷取伺服器資訊清單區段。由於 Java MJC 綱目與 Perl 綱目之間的差異，此問題不適用於 Perl 引擎上的工作。

語法

.bomexport [path_name][file_name]

選項

選項	說明
path_name	選用的路徑名稱。如有提供，此路徑必須是 Build Forge 伺服器上的步驟工作目錄的相對路徑。如果加以省略，檔案會儲存至步驟的工作目錄。
file_name	選用的檔名。工作的 BOM 會儲存至 XML 格式的檔案中。如果沒有提供檔名，則會採用建置標籤名稱和字串 _BOM 來建構檔名：<build_tagname>_BOM.xml。

範例

```
.bomexport
.bomexport myproj.xml
.bomexport path/to/myproj.xml
.bomexport /path/to/myproj.xml
```

.break

.break [<notification_group_name>]

使用 .break 指令可中止工作，直到您將它重新啟動。當系統發現含有 .break 指令的步驟時，執行會完成，且結果為「已停止」。使用**重新啟動**圖示來重新啟動工作，即可繼續進行 .break 步驟之後的步驟。

如果 .break 指令出現在鏈結的工作中，系統就會停止該鏈結的工作，而將控制權交回給呼叫端工作，繼續處理步驟。

您可以將存取群組當作選用引數來併入指令中；若您這麼做，當系統停止工作時，就會傳送電子郵件訊息至所指定的存取群組。

.bset

.bset env "<VarName>=<Value>" [...]


```
.bset selector <SelectorName> [(SelectorSnapshotName)]  
.bset server <ServerName>  
.bset buildserver <ServerName>
```

在工作期間，`.bset` 指令會暫時變更專案設定。

註：`.bset` 指令會影響之後使用該環境之步驟的基本建置環境。指定 `.bset` 指令時，該指令不要依賴相同步驟中的前一個 `.bset` 指令。例如，不要在同一個步驟中指定類似下列組合的步驟。

```
.bset env "VAR1=VALUE1"  
.bset env "VAR2=$VAR1"
```

該指令有下列選項：

- **env** 會針對執行中的工作，變更其一個以上專案環境變數的值。這項變更會在現行步驟中立即生效。如需以這種方式來使用該指令的相關資訊，請參閱第 338 頁的『在工作期間變更建置標籤』。您可以設定一個尚未存在的變數。以 `.bset` 指令設定的值，會寫入工作記錄中。如果您為一個同時定義在專案或步驟環境中的變數，設定新的值，該新值只會在工作期間生效。專案或步驟的環境不會變更。
- **selector** 會在工作期間變更專案選取元。新的選取元會在下個步驟中生效，其中可包括現行步驟的任何列入步驟。新的選取元只會用於沒有明確選取元設定的步驟。該步驟的選取元設為「專案預設值」。如果您在列入步驟中使用 **selector** 選項，此選項會影響列入專案中的所有後續步驟，包括列入步驟所列入的任何步驟。它不會影響呼叫端步驟，或是位於呼叫端步驟層次或其上方層次的任何其他步驟。

設有明確選取元的步驟不受影響。

使用選用的 `[(SelectorSnapshotName)]` 來指定所指定選取元的 Snapshot。

- **server** 會在工作期間變更預設專案伺服器。新的伺服器設定會在下個步驟中生效，其中不包括任何列入步驟。步驟只會使用指定的伺服器；它們沒有明確的選取元設定。

反引號語法：您可以使用反引號語法，將伺服器名稱設定成指令的輸出。例如，下列指定會執行 `SelectAServer.sh` Script，並提供其輸出作為 `.bset` 伺服器指令的伺服器名稱：

```
.bset server `SelectAServer.sh`
```

註：如果將 `.bset` 與反引號搭配使用，或將目標 Script 輸出與非零結束碼搭配使用，MJC 會報告錯誤訊息 `Exception communicating with endpoint server`。

- **buildserver** 會在工作期間變更預設專案伺服器。新的伺服器設定會在下個步驟中生效，其中包括任何列入步驟。在迴圈情況下套用 `.bset` 時，此設定也會在下一個迴圈中生效。它和 `server` 選項不同，因為它會套用至列入步驟、下一個迴圈步驟，以及專案層次的步驟。

註：針對 Perl 引擎，`.bset server` 和 `buildserver` 不會影響列入步驟。

重要：請避免在含執行緒作業的步驟中使用多個 `.bset` 指令。

.buildstatus

`.buildstatus <result>`

這個指令會在建置完成後，強制建置具有指定的 `<result>`，而不考慮建置步驟的結果。

`<result>` 的有效值如下所示：

- P** 將結果設為「通過」
- F** 將結果設為「失敗」
- W** 將結果設為「警告」
- B** 將結果設為「已停止」（如果已指定 `.break` 指令）

.date

`.date <conversion_specifier>`

使用 `.date` 指令，並以一個以上轉換指定元字元作為引數，可在專案執行時產生現行日期/時間資訊。

`.date` 指令和其引數必須定義成環境中的環境變數。之後您便可以將環境指派給專案或步驟。

例如，在環境中定義一個 `DayOfWeek` 環境變數，並指派 `.date %A` 值給它。然後再將環境指派給專案或步驟。如果專案是在星期三執行，工作會將「星期三」文字指派給 `DayOfWeek` 環境變數。

重要：在步驟的「指令」欄位中，無法直接參照 `.date` 指令。

註：`.date` 指令會重新評估每一個步驟。如果要保留特定的時間，使用 `.date` 時請搭配 `.bset env`。例如，使用 `$ORIGTIMESTAMP = .date %d-%b-%Y.%H:%M:%S` 然後使用 `.bset env "TIMESTAMP=`echo $ORIGTIMESTAMP`"`。

`.date` 指令建置在 POSIX `strftime` 函數上，可接受以 ANSI C89 標準識別的轉換指定元。下表提供轉換指定元的日期/時間值。

註：並非所有的轉換指定元都可以在各個語言環境與作業系統之間使用。請在您打算使用 `.date` 指令的伺服器作業系統與語言環境上，測試這個指令的結果。

日期轉換指定元	說明
<code>%a</code>	依現行語言環境而定的平日名稱縮寫。
<code>%A</code>	依現行語言環境而定的完整平日名稱。
<code>%b</code>	依現行語言環境而定的月份名稱縮寫。
<code>%B</code>	依現行語言環境而定的完整月份名稱。
<code>%c</code>	現行偏好的語言環境日期和時間表示法。

日期轉換指定元	說明
%d	一個月中的第幾日，以十進位數表示（範圍從 01 至 31）。
%H	使用 24 小時表示法的小時數，以十進位數表示（範圍從 00 至 23）。
%I	使用 12 小時表示法的小時數，以十進位數表示（範圍從 01 至 12）。
%j	一年中的第幾日，以十進位數表示（範圍從 001 至 366）。
%m	月份，以十進位數表示（範圍從 01 至 12）。
%M	分鐘，以十進位數表示（範圍從 00 至 59）。
%p	『AM』或『PM』，需根據所給定的時間值，或是現行語言環境的對應字串。正午視為『pm』，午夜則視為『am』。
%S	秒數，以十進位數表示（範圍從 00 至 61）。
%U	該年的第幾週，以十進位數表示，範圍從 00 至 53，以第一個星期日作為第 01 週的第一天。
%w	星期幾，以十進位數表示，範圍從 0 至 6，星期日 0。
%W	該年的第幾週，以十進位數表示，範圍從 00 至 53，以第一個星期一作為第 01 週的第一天。
%y	年份，以十進位數表示，不含世紀（範圍從 00 至 99）。
%Y	年份，以十進位數表示，包含世紀。
%Z	時區或名稱或縮寫。
%%	『%』字元。

.defect

說明 使用 .defect 指令，將問題追蹤應用程式的配接器新增至專案步驟。問題配接器為 Build Forge 物件；它是以問題追蹤應用程式的配接器範本為基礎。步驟的配接器程式碼會在專案執行時執行。

語法

```
.defect <adaptor_name> [entry_name]
```

<adaptor_name> 是必要的；這是在「管理主控台」中指派給配接器的名稱。
<adaptor_name> 的大小寫應該和主控台中所使用的大小寫相符。

如果配接器範本具有多個介面函數，請使用 *entry_name* 選項來指定所要執行的函數。*entry_name* 必須符合配接器範本中指定給 <interface> 元素的 *name* 屬性。如果 *entry_name* 中指定的 <interface> 元素不存在或找不到，則會改為執行預設 <interface> 元素。在下列範例中，項目名稱是 DefectFunction。

如果使用配接器鏈結，則會自動呼叫配接器，並執行配接器範本中的第一個介面函數。如果要執行不同的介面，請在配接器範本中針對您確實要執行的介面，將 *default* 屬性設為 *true* (*default="yes"*)。

範例

```
.defect MyClearCaseQuestAdaptor
.defect MyClearCaseQuestAdaptor DefectFunction
```

附註 如果要建立配接器或檢視配接器清單，請選取**專案 > 配接器**。

和 Build Forge 產品一起提供的配接器範本位於：

```
<bfinstall>/interface
```

.drill

```
.drill [through]
<"var1,var2,var3"|${EnvVar}|-r[p] Register>
[gr[ouped by] "{}"]
[sep[arated by] ","]
[exec] "Command $1 $2"
```

.drill 指令可讓您以迴圈方式執行指令，針對一系列值中的每個項目各執行一次指令。您可以在指令行上指定值，或是從環境變數或暫存器提取值。系統在執行 .drill 指令時，會使用 .drill 語法來建構一系列的指令行，並將其傳送至代理程式來執行。

例如，指令 .drill "A,B,C,D" "echo value \$1" 會建立下列指令：

```
echo value A
echo value B
echo value C
echo value D
```

分組

您可以將值分組，並使用 *\$n* 語法來參照每個群組中的多個值。*\$1* 是指群組中的第一個值，*\$2* 是指群組中的第二個值，依次類推。例如，.drill through "(A,B,C,D,E),(B,C,D,E,F),(C,D,E,F,G)" grouped by "(" separated by "," exec "echo 1[\$1] 2[\$2] 3[\$3] 4[\$4] 5[\$5]" 會建立下列指令：

```
echo 1[A] 2[B] 3[C] 4[D] 5[E]
echo 1[B] 2[C] 3[D] 4[E] 5[F]
echo 1[C] 2[D] 3[E] 4[F] 5[G]
```

註：沒有預設的分組字元。有預設的分隔字元，即逗點。如果沒有指定 *grouped by*，系統就會依據分隔字元的分隔方式來查閱所提供的值，並將每個這種字串視為單一值。例如，指令 .drill "(A,B),(C,D)" "echo \$1 \$2" 會解析為下列指令：

```
echo (A 2
echo B) 2
echo (C 2
echo D) 2
```

資料來源

您有數種選項來選擇 `.drill` 指令要從哪裡取得它要以迴圈方式執行的資料。指令的第一個參數是資料來源。您可以併入選用的指令字 "through" 來指出資料來源。

- 您可以在指令行中明確地列出資料，像下列指令一樣，以迴圈方式執行第一、第二及第三個值：

```
.drill through "one,two,three" exec "echo $1"
```

- 您可以從環境變數提取資料。下列指令假設環境變數 `FILENAMES` 是以逗點區隔的檔案清單，並使用 `DOS` 指令來刪除清單中的所有檔案：

```
.drill through $FILENAMES exec "del $1"
```

- 您可以從暫存器或專案暫存器提取資料。如果 `RegisterA` 包含以逗點區隔的檔名清單，發出至 Linux 系統的下列指令就會寫出每個檔案的內容：

```
.drill -r RegisterA exec "cat $1"
```

下列範例也會執行相同的動作，但使用的是專案暫存器：

```
.drill -rp ProjectRegisterA exec "cat $1"
```

.edit

```
.edit [<search_expression>/<replace_expression>/ [<relative_path>/]file [file ...]
```

使用 `.edit` 指令，在一個以上檔案中搜尋並取代文字字串。`.edit` 指令會取代每個指定檔案中，每一行的第一個字串實例 (`search_expression`)。除非您指定相對路徑，否則會假設檔案位於步驟的工作目錄中。

`.edit` 指令會實作標準 POSIX 正規表示式來進行比對及取代，包含使用取代型樣的 () 子字串選取及 \N 替代動作。

依預設，`.edit` 指令使用「POSIX 延伸正規表示式」語法。如果已編譯代理程式，並支援「Perl 相容正規表示式」，則可在替代表示式後面使用 "p" 字元來指定 PCRE 語法。

視代理主機上的作業系統而定，使用反斜線作為跳出字元將會需要不同的處理方法。

- 在 UNIX 及 Linux 上，代理程式處理程序會解譯表示式兩次，因此請使用四條反斜線來代表一條反斜線。例如，使用四條反斜線和句點，以符合文字句點字元。

```
\\\\.
```

- 在 Windows 上，只需兩條反斜線即可符合相同文字句點。

```
\\.
```

您無法跳出定界字元。在需要跳出定界字元的情況下，正確方法是使用替代定界字元。範例：您擁有的 `abc.txt` 檔案，僅包含 `abc` 字元。您要將它取代為 `a/c` 字串：

- 錯誤 - 在下列情況下，字串會保持不變，因為定界字元優先於跳出字元。

```
.edit /abc/a\c/ abc.txt
.edit /abc/a\\c/ abc.txt
.edit /abc/a\\\\c/ abc.txt
```

- 正確 - 變更定界字元將會容許替代 `a/c`。

```
.edit "abc"a/c" abc.txt
```

註：您必須直接列出一或數個檔名，且不能使用萬用字元。

範例：下列指令會取代 `drivermakefile` 檔案中 `winXPdriver` 和 `win2000driver` 之類的字串。

```
.edit /win.*driver/linuxdriver/ drivermakefile
```

`.edit` 指令類似 `.strsub` 指令；差別在於：

- 在大型文字檔或許多檔案中執行取代作業時，`.strsub` 指令會比 `.edit` 快。
- `.edit` 指令可以執行正規表示式的搜尋及取代作業。
- `.edit` 指令只會取代每個檔案中，每一行的第一個字串實例 (`search_expression`)。
- `.strsub` 指令會取代每個檔案中，每一行的每個字串實例（原始檔）。

.email

```
.email <recipient>
```

此指令使用 `step_email` 郵件範本，將電子郵件傳送至 *recipient*。理論上，指定的位址是 Build Forge 使用者，但是 `.email` 也會傳送至任何有效的電子郵件位址。

.expand

```
.expand <var1 var2 ... varn>
```

此指令會接受以空格區隔的變數、剖析伺服器（代理程式）上的變數，並顯示擴充變數，如下所示：

```
variable[1] = expanded value
variable[2] = expanded value
...
variable[n] = expanded value
```

.export

```
.export [path_name][file_name]
```

`.export` 指令會將呼叫端專案的專案定義，儲存至位於步驟工作目錄中的 XML 檔。XML 檔會說明專案及其步驟。它不會說明其他相關聯的物件（例如：伺服器）。

匯出的 XML 檔案可用來將專案定義匯入至「管理主控台」。

`.export` 指令可採用選用路徑與（或）檔名。路徑必須是相對路徑。它會套用自步驟的工作目錄。

如果沒有提供檔名，則會從現行專案標籤 `$BF_TAG.xml` 來建構檔名。

.get

```
.get server:[[<relative_path>/]file/]file [[[<relative_path>/]file/]file]
```

使用 `.get` 指令來將檔案從一部邏輯伺服器傳送至另一部邏輯伺服器。`.get` 作業會從現行伺服器路徑來執行，並從指定的伺服器路徑擷取檔案。目的地路徑名稱相對於步驟的現行工作目錄。來源路徑名稱相對於指定伺服器的基本路徑。伺服器必須指定可容許對檔案執行 `.get` 作業的邏輯伺服器。如需相關資訊，請參閱第 341 頁的『在伺服器資源上啟用檔案複製功能』。只能傳送單一檔案。

路徑規格可以併入環境變數。這項功能可讓您指定相對於特定工作所用路徑的檔案。請參閱第 384 頁的『工作的工作目錄』中，有關工作所用路徑的說明。

如果您要使用的伺服器名稱中有空格，請將伺服器名稱和檔案路徑用引號括住。例如：`.get [<relative_path>/]file server:[<relative_path>/file]`

傳送速度並不快，所以您可能會想要選擇不同的方法來傳送大型檔案。預期速度不會超過每秒 40 KB；70 MB 的檔案需要 45 分鐘至一小時的時間來傳送。

註：如果目的地檔案已存在，會逕自改寫而不警告。

由於 Windows 檔案路徑分隔字元（反斜線 \）與 Rational® Build Forge® 中使用的跳出字元相同，因此在剖析文件時，難以確定指令何時表示「跳出」。請使用下列字元來區分 Windows 檔案路徑分隔字元與跳出字元：

- 如果要啟用跳出，請輸入 /E。
- 如果要停用跳出，請輸入 /e。

下例顯示 .put 指令：

```
.put[/E/e] [<relative_path>/]file server:[<relative_path>/file]
```

如果 /E 與 /e 均未指定，系統將使用啟發方法，以查看點指令是否包含類似於 Windows 檔案路徑的內容。如果偵測到檔案路徑，則停用跳出；否則，啟用跳出。

.load

```
.load [-o] [-e] [-v] [-j] [<relative_path>/]<filename>
.load -r|-p <registername>
.load -s <command name>
```

在執行 .load 指令的步驟之後，.load 指令會從 XML 檔載入專案，並將載入專案的步驟新增至現行專案，讓專案能夠在執行時期動態建立及載入步驟。您可以使用選項來讓 .load 指令從暫存器或指令的輸出提取資料。

如果要撰寫 .load 指令的 XML 檔，請先使用現有專案的匯出檔案，以提供您適當的基本結構。您也可以在系統中建立專案，然後將其匯出，以用於 .load 指令。這個主題包含範例 XML 程式碼。

.load 指令所載入的步驟可包含列入或鏈結專案的參照。依預設，系統會在 XML 檔中尋找列入專案的定義，並載入其步驟；請參閱本主題後面的 -e 選項，以瞭解如何讓系統從資料庫取得列入專案定義。若為通過鏈專案或失敗鏈專案，系統一律會在資料庫中尋找專案定義。

註：如果是從 .load 來執行 JPO 步驟，else-inline 所識別의專案或程式庫，必須是系統中已存在的專案或程式庫，否則，就不會執行列入。

XML 檔中的多個專案

因為系統會將列入專案與其呼叫端專案一起匯出，所以 XML 檔可能會包含數個專案。 .load 指令會執行檔案中標示為主要的專案。這個專案的 <project> 元素上有 primary="1" 屬性。

指令的選項和參數

最簡單的指令形式是 `.load <filename>`。您可以選擇在檔名前面加上路徑名稱（工作目錄的相對路徑）。例如，指令

```
.load ../../project.xml
```

會從伺服器目錄（包含專案及工作目錄的目錄）載入 `project.xml` 檔，並假設該步驟的路徑內容為 `"/"`（預設值）。

註：當正常的步驟啟動列入專案時，系統會到資料庫取得該專案的現行定義；當 `.load` 指令所匯入的步驟啟動列入專案時，系統則會在 XML 檔中尋找列入專案的定義。請參閱下面 `-e` 選項的說明，以了解如何避免這種狀況。

註：從 `d from Build Forge 7.1` 版之前的版本產生的步驟 XML 不受支援且可能會失敗，因為這些元素是透過名稱而非 ID 來參照。

註：`.load` 會使用匯入公用程式，進而根據匯入公用程式的設定來套用存取群組所有權。請參閱第 420 頁的『如何將存取群組指派給匯入的物件』以取得進一步資訊。

該指令有下列選項：

-r 或 -p

這些選項會使系統從暫存器載入步驟。請搭配這些選項來使用指令行。

```
.load -r|-p <registername>
```

`-r` 選項會從原始暫存器載入步驟，而 `-p` 選項會從專案暫存器載入步驟。您可以在專案的前幾個步驟中，於暫存器內建置資料，然後使用此指令從暫存器載入步驟。

-s 這個選項會使系統執行某個指令，並利用這個指令的輸出作為要載入的資料。請使用下列指令行

```
.load -s `<command name>`
```

-e 當設定 `-e` 選項時，系統會從資料庫取得列入的專案，而不是從載入的 XML 檔取得。它會將 `chainID` 的值當作對資料庫中之專案 ID 的參照。這樣會使 XML 檔參照最新版的列入專案（而不是 XML 檔中的專案），或是參照沒有包含在 XML 檔中的專案。

-o 使用 `-o` 選項可停用 XML 檔中的列入專案。使用這個選項時，系統會忽略主要專案中的任何列入專案。步驟中若有包含對列入專案的參照，則會執行其指令，但會忽略其列入。

-j 如果 XML 檔中的最後一組步驟是採用執行緒，而且 `.load` 指令後面的步驟也採用執行緒，請使用 `-j` 選項。`-j` 選項會將最後一個含執行緒作業的步驟轉換成一個結合步驟。否則，含執行緒作業的步驟就會成為 `.load` 指令後面的執行緒步驟區塊一部分。

-v 傳送所要載入的 XML 檔內容，以便顯示終端機以供檢視。

範例 XML

下列範例顯示要搭配 `.load` 指令來使用的 XML 檔。該 XML 是藉由匯出一個名為 `HelloWorldPlusInline` 的專案而建立的。

請注意範例 XML 的下列詳細資料：

- XML 包含兩個 `<project>` 元素。
- XML 中的第一個專案是主要專案；其具有 `name="HelloWorldPlusInline"` 及 `primary="1"` 屬性。
- XML 中的第二個專案名為 `Sleepytime`，並具有 `primary="0"` 屬性，表示它不是主要專案。
- `HelloWorldPlusInline` 的第一個步驟是名為 `EchoHelloWorld` 的步驟，它包含 `echo` 指令和 `chainID` 屬性。`chainID` 屬性的值為 2，表示系統應列入含 ID 2 的專案，也就是 `Sleepytime` 專案。

註：請忽略步驟屬性 `inline`；這是已不再使用的淘汰屬性。所有步驟都有這個屬性，且值為 N。如果要判斷步驟中是否有列入專案，請尋找 `chainID` 屬性。`chainID` 的值會參照專案的 ID。依預設，系統會在 XML 檔中尋找列入專案，但如果您在 `.load` 指令中使用 `-e` 選項，系統就會將該值視為資料庫中的專案 ID。如此可讓您建立您自己的 `.load` 檔，而不需要在檔案中併入列入專案。

註：將 `.load` 與 Build Forge 7.0.x 版和更舊的版本所產生的 XML 檔搭配使用，不但受支援，而且還無法正常運作，因為舊的語法是透過名稱而非 UUID 來參照物件。

- 每個專案都有 `id` 屬性。這個 ID 值與資料庫中的專案 ID 相同。您可以從安裝目錄執行下列指令，來取得專案 ID 的清單：

```
bfexport -l
```

- `commandStore` 屬性包含 `<command>` 元素內容的副本，但某些字元會跳離 XML。例如，在 `<command>` 中可能有 `"、'、&、<、>` 或新行。這些字元在 `commandStore` 中是以 `"、'、&、<、>` 和 `
` 來表示。

```
<?xml version="1.0" encoding="UTF-8"?>

<buildforge schema="7.115014" comment="">
  <project access="6" active="Y" name="HelloWorldPlusInline" primary="1"
    selectorId="Choose_local" maxthread="0" increment="Y" tagsync="0"
    buildclass="Production" sticky="N" envId="0" tag="BUILD_$B" id="19" exclusive="0">
    <tagvar autoincrement="Y" name="B" id="1">2</tagvar>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="1" passwait="N" inline="N" threadable="N" chainId="2"
      access="6" active="Y" passnotify="0" description="EchoHelloWorld" onfail=" "
      failnotify="0" envId="0" commandStore="echo Hello World">
      <command>echo Hello World</command>
    </step>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="2" passwait="N" inline="N" threadable="N" access="6"
      active="Y" passnotify="0" description="export proj to build and server folders"
      onfail=" " failnotify="0" envId="0"
      commandStore=".export $BF_PROJECTNAME_PHYS.xml&#10;copy /Y $BF_PROJECTNAME_PHYS.xml ..\">
      <command>.export $BF_PROJECTNAME_PHYS.xml
copy /Y $BF_PROJECTNAME_PHYS.xml ..\"</command>
    </step>
  </project>
  <project access="6" active="Y" name="Sleepytime" primary="0"
    selectorId="Choose_local" maxthread="0" increment="Y" tagsync="0"
    buildclass="Production" sticky="N" envId="0" tag="SLEEP_$B" id="2" exclusive="0">
    <tagvar autoincrement="Y" name="B" id="1">21</tagvar>
    <step absolute="N" failwait="N" selectorId="" dir="/" broadcast="N"
      timeout="300" id="1" passwait="N" inline="N" threadable="N" access="6"
      active="Y" passnotify="0" description="Sleep, perchance to dream" onfail=" "
      failnotify="0" envId="0" commandStore=".sleep 0">
      <command>.sleep 0</command>
```

```

    </step>
</project>
<class maxdays="0" access="1" entranceprojectId="1" name="Production" keepfiles="B"
  deletechangedata="N" purgeprojectId="2" exitProjectId="5" candidates="AnyBuild "
  maxbuilds="0"></class>
  <selector operator="" required="" access="6" value="" name="Choose_local"
    selectorId="" property=""></selector>
</buildforge>

```

.lock

```
.lock
```

.lock 指令會使系統在指令完成之後，將工作鎖定。這可避免根據工作類別的內容，而自動將工作刪除；此外，鎖定的執行不會列在**工作 > 已完成**標籤上，而是出現在**已鎖定**標籤上。該指令不使用參數；它會鎖定在內部使用它的工作。

.mkdir

```
.mkdir <relative_path>
```

.mkdir 指令可建立目錄。**<relative_path>** 參數會解譯為現行步驟目錄的相對路徑。如果路徑名稱規格中的目錄不存在，則會加以建立。不容許絕對路徑和包含磁碟機代號（例如 C:\）的路徑。

.monitor

```
.monitor [-c] [-w] <interval> [<relative_path>/]<filename>
```

.monitor 指令可以讓系統在監視檔案以查看檔案大小何時停止變更時，中止專案。當步驟發出這個指令時，系統會檢查所指示的檔案；然後每隔 **<interval>** 秒，再重新檢查檔案。當檔案大小無法在二個間隔之間變更時，系統會繼續進行下一個步驟。

如果您使用 **-c** 選項，系統會在判定所監視的檔案已停止變更時，將其內容寫出至步驟日誌；然後再繼續下一個步驟。

如果檔案不存在，系統就不會等待，而會在第一個間隔之後，立即繼續進行。使用 **-w** 選項可強制系統在開始監視處理程序之前，等待建立檔案。

.pack

說明 使用 **.pack** 指令將包裝應用程式的配接器新增至專案步驟中。包裝配接器為 Build Forge 物件；它是以包裝應用程式的配接器範本為基礎。步驟的配接器程式碼會在專案執行時執行。

語法

```
.pack <adaptor_name> [entry_name]
```

<adaptor_name> 是必要的；這是在「管理主控台」中指派給配接器的名稱。**<adaptor_name>** 的大小寫應該和主控台中所使用的大小寫相符。

如果配接器範本具有多個介面函數，請使用 **entry_name** 選項來指定所要執行的函數。**entry_name** 必須符合配接器範本中指定給 **<interface>** 元素的 **name** 屬性。如果 **entry_name** 中指定的 **<interface>** 元素不存在或找不到，則會改為執行預設 **<interface>** 元素。在下列範例中，項目名稱是 PackageFunction。

如果使用配接器鏈結，則會自動呼叫配接器，並執行配接器範本中的第一個介面函數。如果要執行不同的介面，請在配接器範本中針對您確實要執行的介面，將 `default` 屬性設為 `true` (`default="yes"`)。

範例

```
.pack MyPackagingAdaptor
.pack MyPackagingAdaptor PackageFunction
```

附註： 如果要建立配接器或檢視配接器清單，請選取**專案 > 配接器**。

和 Build Forge 產品一起提供的配接器範本位於：

```
<bfinstall>/interface
```

.pop

```
.pop [-p] <register_name> [+] [<relative_pathname>|-]
.pop [-p] <register_name> [>|>>]<register_name>
```

`.pop` 指令會將暫存器的內容寫入檔案、步驟日誌或另一個暫存器。

選用性的 `-p` 參數可使指令參照專案暫存器。專案暫存器和一般暫存器分開，且專案暫存器會在工作結束後持續進行。

下列範例顯示 `.pop` 指令的各種用法：

- `.pop A data.txt`

將暫存器 A 寫入 `data.txt` 檔（位於步驟的工作目錄中）。

- `.pop ver +data.txt`

將暫存器 `ver` 的內容附加至 `data.txt` 檔。

- `.pop Alpha`

將暫存器 `Alpha` 的內容寫入步驟的日誌中。

- `.pop ALPHA > BETA`

使暫存器 `BETA` 的內容與暫存器 `ALPHA` 的內容相同。

- `.pop A >> B`

將暫存器 A 的內容附加至暫存器 B。

註： 取出暫存器資料並不會將其清空。如果要變更暫存器的內容，請使用 `.push` 指令來將新的值推送至暫存器中。

由於 Windows 檔案路徑分隔字元（反斜線 \）與 Rational® Build Forge® 中使用的跳出字元相同，因此在剖析文件時，難以確定指令何時表示「跳出」。請使用下列字元來區分 Windows 檔案路徑分隔字元與跳出字元：

- 如果要啟用跳出，請輸入 `/E`。
- 如果要停用跳出，請輸入 `/e`。

下例顯示 `.put` 指令：

```
.put[/E/e] [<relative_path>/]file server:[<relative_path>/]file]
```

如果 /E 與 /e 均未指定，系統將使用啓發方法，以查看點指令是否包含類似於 Windows 檔案路徑的內容。如果偵測到檔案路徑，則停用跳出；否則，啓用跳出。

.poptag

`.poptag [-p]<registername>`

`.poptag` 指令會變更現行標籤，以所指定的暫存器內容來將它取代。

選用性的 `-p` 參數可使指令參照專案暫存器。專案暫存器和一般暫存器分開，且專案暫存器會在工作結束後持續進行。

.purge

`.purge`

`.purge` 指令用來設定一個旗標，以便在工作完成之後，立即清除工作。在 `.purge` 指令之後執行的 `.lock` 指令，則會儲存工作。您可以使用此指令，建立唯有在順利完成所有步驟時才會儲存的工作。如果要建立此類工作，請以 `.purge` 指令作為專案中的第一個步驟，並以 `.lock` 指令作為最後一個步驟。

.push

`.push [-p] [+]<register_name> [<relative_pathname> | -]`

`.push` 指令會將 `<relative_pathname>` 的內容放進暫存器 `<register_name>`。 `<register_name>` 的現行內容會被取代。

若要附加而不是取代，請在 `<register_name>` 前面放上一個加號 (+)。

若要清除暫存器，請改用連字號來取代 `<relative_pathname>`。

選用性的 `-p` 參數可使指令參照專案暫存器。專案暫存器和一般暫存器分開，且專案暫存器會在工作結束後持續進行。

除非已啓用步驟的「絕對」內容，否則 `<relative_pathname>` 會相對於專案或標籤路徑。

下列範例假設沒有啓用步驟的「絕對」內容：

- `.push Alpha data.txt`

將步驟工作目錄中的 `data.txt` 檔的內容放在暫存器 Alpha 中。

- `.push +B ..\newdata.txt`

將步驟工作目錄的上層目錄中 `newdata.txt` 檔的內容附加到暫存器 B 中。

- `.push ALPHA -`

清除暫存器 ALPHA。

由於 Windows 檔案路徑分隔字元（反斜線 \）與 Rational® Build Forge® 中使用的跳出字元相同，因此在剖析文件時，難以確定指令何時表示「跳出」。請使用下列字元來區分 Windows 檔案路徑分隔字元與跳出字元：

- 如果要啓用跳出，請輸入 /E。
- 如果要停用跳出，請輸入 /e。

下例顯示 .put 指令：

```
.put[/E/e] [<relative_path>/]file server:[<relative_path>/]file]
```

如果 /E 與 /e 均未指定，系統將使用啓發方法，以查看點指令是否包含類似於 Windows 檔案路徑的內容。如果偵測到檔案路徑，則停用跳出；否則，啓用跳出。

.put

```
.put [<relative_path>/]file server:[<relative_path>/]file]
```

使用 .put 指令，可將檔案從某個邏輯伺服器傳送至另一個。 .put 作業從現行伺服器路徑執行，將指定的檔案傳送至遠端伺服器。目的地路徑名稱相對於目標伺服器的基本路徑。來源路徑名稱相對於步驟的現行工作目錄。遠端伺服器必須指定可容許對檔案執行 .put 作業的邏輯伺服器。如需相關資訊，請參閱第 341 頁的『在伺服器資源上啓用檔案複製功能』。只能傳送單一檔案。

路徑規格可以併入環境變數。這項功能可讓您指定相對於特定工作所用路徑的檔案。請參閱第 384 頁的『工作的工作目錄』中，有關工作所用路徑的說明。

如果您所使用的伺服器名稱中有空格，請用引號括住名稱。

傳送速度並不快，所以您可能會想要選擇不同的方法來傳送大型檔案。預期速度不會超過每秒 40 KB；70 MB 的檔案需要 45 分鐘至一小時的時間來傳送。

註：如果目的地檔案已存在，會逕自改寫而不警告。

由於 Windows 檔案路徑分隔字元（反斜線 \）與 Rational® Build Forge® 中使用的跳出字元相同，因此在剖析文件時，難以確定指令何時表示「跳出」。請使用下列字元來區分 Windows 檔案路徑分隔字元與跳出字元：

- 如果要啓用跳出，請輸入 /E。
- 如果要停用跳出，請輸入 /e。

下例顯示 .put 指令：

```
.put[/E/e] [<relative_path>/]file server:[<relative_path>/]file]
```

如果 /E 與 /e 均未指定，系統將使用啓發方法，以查看點指令是否包含類似於 Windows 檔案路徑的內容。如果偵測到檔案路徑，則停用跳出；否則，啓用跳出。

.rem

說明

.rem 點指令是作為註解用。它很適合用來對步驟中的指令加註。

範例

```
.rem This is some comment text here  
.rem all text after .rem is ignored.
```

.report

將您在「快速報告」中所建立報告的輸出，新增至工作 BOM 中。「快速報告」是 Rational® Build Forge® 中一個要另外授權的特性。

註：在 Build Forge® 8.0 中，Java 引擎上不支援 `.report` 指令。

說明 使用 `.report` 指令，可將報告新增至專案。

`.report` 指令會執行報告，並將其輸出新增至工作 BOM 中。每當執行工作時，BOM 就會根據現行資料顯示報告結果。

語法

```
.report <report_name>
```

`<report_name>` 的限制如下所示：

- `<report_name>` 是必要的。
 - 名稱會區分大小寫。
 - 如果名稱含有空格，請括上引號。例如 `"my_report"`。
- 您指定的 `<report_name>` 必須儲存成公用報告。
- 不支援 BOM 報告類型。

.retag

```
.retag <new_tag>
```

在步驟中使用 `.retag` 指令，可在工作期間變更工作的標籤。您可以使用變數或指令作為新標籤值。

.retry

```
.retry <count> <command>
```

`.retry` 指令可讓指令在失敗時重試。`.retry` 指令使用單一計數引數來指定重試指令的次數。所要執行的指令會被視為引數的其餘部分，所以 `.retry` 指令必須是步驟的最後一個點指令。例如，請考量下列指令：

```
.retry 3 myscript.sh arg1 arg2 arg3
```

它會執行『myscript.sh arg1 arg2 arg3』達 3 次，才會使步驟失敗。第一個傳回成功狀態的指令呼叫即停止重試程序。

.rget

```
.rget server:[<path>] [<path>]
```

`.rget` 指令的運作類似 `.get` 指令，但它會遞迴地複製整個目錄樹狀結構。您必須提供目錄名稱來作為參數。例如，指令

```
.rget winbuildserver1:config myconfig
```

會將 winbuildserver 伺服器上 config 目錄的內容複製到現行伺服器上的 myconfig 目錄中。

註：

- 請不要在此指令中使用環境變數。
- 請不要在此指令中使用符號鏈結。
- 指定的目錄必須存在，然後才能嘗試使用指令。如果指定的目錄不存在，指令將會失敗。

由於 Windows 檔案路徑分隔字元（反斜線 \）與 Rational® Build Forge® 中使用的跳出字元相同，因此在剖析文件時，難以確定指令何時表示「跳出」。請使用下列字元來區分 Windows 檔案路徑分隔字元與跳出字元：

- 如果要啟用跳出，請輸入 /E。
- 如果要停用跳出，請輸入 /e。

下例顯示 .put 指令：

```
.put[/E/e] [<relative_path>/]file server:[<relative_path>/]file]
```

如果 /E 與 /e 均未指定，系統將使用啟發方法，以查看點指令是否包含類似於 Windows 檔案路徑的內容。如果偵測到檔案路徑，則停用跳出；否則，啟用跳出。

.rmdir

```
.rmdir <relative_path>
```

.rmdir 指令會移除 <relative_path> 所指定的目錄。系統會移除該路徑名稱指定的基本目錄，包含所有的內容及其後代。

.rput

```
.rput [<relative_path>] server:[<relative_path>]
```

.rput 指令的運作類似 .put 指令，但它會遞迴地複製整個目錄樹狀結構。您提供的相對路徑必須是目錄，而不是檔案。例如，指令

```
.rput myconfig linuxserver5:feb2005
```

會將 myconfig 目錄的內容從現行伺服器複製到 linuxserver5 伺服器上的 feb2005 目錄。

來源路徑相對於步驟的工作目錄，所以是否併入專案及標籤目錄，需依據步驟的「絕對」內容值。目的地路徑僅相對於目的地伺服器的「路徑」內容。如需系統如何構造路徑的相關資訊，請參閱第 384 頁的『工作的工作目錄』。

註：

- 請不要在此指令中使用環境變數。
- 請不要在此指令中使用符號鏈結。
- 指定的目錄必須存在，然後才能嘗試使用指令。如果指定的目錄不存在，指令將會失敗。

由於 Windows 檔案路徑分隔字元（反斜線 \）與 Rational® Build Forge® 中使用的跳出字元相同，因此在剖析文件時，難以確定指令何時表示「跳出」。請使用下列字元來區分 Windows 檔案路徑分隔字元與跳出字元：

- 如果要啟用跳出，請輸入 /E。
- 如果要停用跳出，請輸入 /e。

下例顯示 .put 指令：

```
.put[/E/e] [<relative_path>/]file server:[<relative_path>/]file]
```

如果 /E 與 /e 均未指定，系統將使用啟發方法，以查看點指令是否包含類似於 Windows 檔案路徑的內容。如果偵測到檔案路徑，則停用跳出；否則，啟用跳出。

.run 及 .runwait

註：在 Build Forge 8.0 中，Java 引擎上不支援 .run 和 .runwait 的條件 [-c<condition>]。若要根據條件而啟動，請改用條件式步驟。

```
.run [-c "<condition>"] "<ProjectName>" ["<ProjectSnapshotName>"]  
.runwait [-c "<condition>"] "<ProjectName>" ["<ProjectSnapshotName>"]
```

您可以利用 .run 和 .runwait 指令，從步驟指令啟動鏈結的專案。若要指定專案的 Snapshot，請使用選用的 <ProjectSnapshotName> 參數。

這二個指令的不同，在於其啟動專案之後的行為：

- .run 指令會遵照鏈結的專案的環境變數繼承規則，將指定的專案當作鏈來啟動。
- .runwait 指令會啟動指定的專案。啟動步驟會等待所啟動的專案完成。當啟動的專案完成時，系統會根據啟動之專案的完成狀態，將啟動步驟的結果值設為通過或失敗。

重要：如果專案包含的步驟含有 .runwait，則該專案執行時會耗用兩個工作屬性。如果可用的工作屬性不足，該步驟會失敗並顯示錯誤。

重要：透過 .run 或 .runwait 指令啟動的任何專案，都不會在「建置結果」頁面中產生鏈的鏈結圖示。

條件式啟動

您可以使用選用的 -c 參數，由條件來決定是否啟動。您可以在條件中使用環境變數。條件可為下列數種形式：

字串比較

您可以使用等於 (=) 或不等於 (!=) 運算子來對字串求值。如果比較項求值為 true，則會啟動鏈。

數值比較

您可以使用 <、>、<>、>< 或 = 運算子來比較二個數值。

指令成功

您可以使用以反引號括住的指令作為 -c 參數的值。系統會執行指令；如果成功，就會啟動鏈。

範例

```
.run "BuildWindowsDriver"
```

系統會啟動 BuildWindowsDriver 專案。執行啟動的專案會立即繼續進行下個步驟。

```
.runwait "BuildWindowsDriver"
```

系統會啟動 BuildWindowsDriver 專案。系統會在 .runwait 步驟暫停執行啟動的專案。當 BuildWindowsDriver 專案完成並通過時，.runwait 步驟的狀態便會設為 pass。

```
.run -c "$HOMEDRIVE=C:" "Simple Echo"
```

只有在 HOMEDRIVE 變數具有值 C 時，系統才會執行 Simple Echo 專案。

這個指令會產生如下所示的日誌輸出（位於步驟日誌的 EXEC 區段中）：

- 當 HOMEDRIVE 為 C：

```
.run Condition: 'C:' = 'C:' satisfied.
```

```
Queueing Project "Simple Echo" on server [WinBox].  
Queued Build 'BUILD_202' of project 'Simple Echo'.
```

- 當 HOMEDRIVE 不是 C :

```
.run -c "$HOMEDRIVE=C:" "Simple Echo"
```

```
.run Condition: 'D:' = 'C:' unsatisfied, no project queued.
```

如果字串包含數字，系統就會以數值來比較字串。例如，其處理下列案例的情況如下所示。

```
.runwait -c "a12b<c42d" "Simple Echo"
```

```
.run Condition: '12' < '42' satisfied.  
Queueing Project "Simple Echo" on server [WinBox].  
Waiting for .run build (4411) to complete.  
.run build is now running.  
.run build has finished.  
Build 'BUILD_203' of project 'Simple Echo' completed.
```

```
.runwait -c "f43g<>h43i" "Simple Echo"
```

```
.run Condition: '43' <> '43' unsatisfied, no project queued.
```

下列範例顯示如何使用指令來作為條件。請注意，指令必須以引號及反引號括住。

```
.run -c "`exit 1`" "Simple Echo"
```

```
Env .run encountered an error during variable expansion,  
parameter ['exit1`] expanded to [].  
Expansion returned non-zero exit, project will not be queued.
```

```
.run -c "`exit 0`" "Simple Echo"
```

```
Expansion returned zero exit, project will be queued.  
Queueing Project "Simple Echo" on server [WinBox].  
Queued Build 'BUILD_204' of project 'Simple Echo'.
```

當您使用 `.runwait` 且建置失敗時，日誌內容將類似下列所示。

```
.runwait "Fail Build"
```

```
Queueing Project "Fail Build" on server [WinBox].  
Waiting for .run build (4413) to complete.  
.run build is now running.  
.run build has finished.  
Build 'BUILD_3' of project 'Fail Build' Failed, setting step status to fail.
```

.scan

```
.scan [-v][-i <ignorepattern>] baseline | checkpoint
```

使用 `.scan` 指令，可加強儲存在 BOM 中的工作資料。它會追蹤步驟工作目錄中的檔案，以及各檔案的 MD5 值。

.scan baseline

將所有檔案的清單儲存在步驟的工作目錄中。系統會針對工作，在 BOM 中將清單顯示為種類。一個工作中可以有多個 `baseline` 指令，但是在指令執行時，每一個指令都會將清單重設為步驟工作目錄的狀態。最後一個 BOM 只會顯示一個 `baseline` 種類。

.scan checkpoint

儲存自前次在工作中執行 `.scan baseline` 或 `.scan checkpoint` 指令以來，所有新建、已變更及已刪除的檔案清單。系統會在 BOM 中顯示清單。每一個 `checkpoint` 指令都會在 BOM 中建立新種類。

您必須先使用 `.scan baseline` 指令，才能在工作中使用第一個 `.scan checkpoint` 指令。在 `.scan baseline` 指令之前的 `.scan checkpoint` 指令會被忽略。

指令選項：

- v** 將變更資訊的副本記錄在工作日誌中。
- i** 忽略符合所提供型樣的目錄。型樣可以符合路徑的開頭、結束或任何的目錄部分。您可以使用這個選項，來刪除變更清單中的來源控制目錄。

CVS 範例：

```
.scan -i CVS checkpoint
```

此範例指令會將 CVS 目錄從報告中剔除。

Subversion 範例：

```
.scan -i .svn baseline
```

如果搭配使用 `-v` 與 `-i`，系統會記載來源控制目錄的變更，但 BOM 中不會包含這些變更。

註：請勿在單一步驟中使用多個 `.scan` 指令。如果您在單一步驟中使用多個 `.scan` 指令，系統將無法提供精確的輸出。

如需使用這些指令的相關資訊，請參閱第 376 頁的『使用 `.scan` 指令來新增基準線及核對點』。

.semget

```
.semget <semaphore_name>
```

當步驟發出這個指令時，系統會檢查是否有所指名的號誌存在。

- 如果沒有這樣的號誌存在，系統會加以建立，並將它指派給步驟的工作。然後執行作業會繼續進行下個步驟。
- 如果有其他工作已要求這個號誌名稱，此工作就會在 `.semget` 步驟停滯，直到其他專案釋出該號誌。

如需使用此指令的相關資訊，請參閱第 534 頁的『號誌 (semaphore)』。

.semput

```
.semput <semaphore_name>
```

使用 `.semput` 指令，可釋放名稱為 `<semaphore_name>` 的號誌。如需使用此指令的相關資訊，請參閱第 534 頁的『號誌 (semaphore)』。

.set

```
.set env <EnvGroupName>[(SnapshotName)] "<VariableName>=<DesiredValue>" [...]
```

`.set` 指令會指派一個值給環境變數。您可以指定其他變數和值。用引號含括每個變數及變數值。這個指令會設定環境群組的預設 `Snapshot` 的值，除非您指定 `Snapshot`。如果您指定 `Snapshot`，請用括弧加以含括，同時與環境群組名稱及 `Snapshot` 名稱之間不要有空格。

註：這個指令所設定的變數必須已經存在。

使用 `.set` 指令，可變更環境的主要記錄。當系統執行專案時，會從主要記錄複製一份專案環境，將它儲存在工作記錄中，並用它來作為專案預設值。

當步驟執行時，它使用的是環境的工作副本，而非主要記錄。因此，使用 `.set` 具有下列效果：

- 當 `.set` 指令在步驟中執行時，後續的步驟只要使用預設步驟環境，就不會看到變更。系統會使用步驟的預設環境工作副本。
- 當 `.set` 指令在指定的環境中執行時，後續的步驟只要指定該環境，就會看到您所做的變更。當步驟指定環境時，系統會讀取環境的主要記錄。即使指定的步驟環境與專案預設值的環境相同，仍舊是如此。
- 工作結束之後，`.set` 指令所做的變更仍會持續。未來執行時，將會使用先前執行之 `.set` 指令所建立的值。

如需使用此指令的相關資訊，請參閱第 337 頁的『使用工作資料』。另請參閱類似指令：第 345 頁的『`.bset`』。

.sleep

`.sleep <seconds>`

使用 `.sleep` 指令，可指定步驟即將暫停的秒數。由於「管理主控台」會處理這個指令，所以不會建立遠端伺服器連線。您也可以使用『`.sleep 0`』指令來作為可在不同的平台上執行的 `null` 指令。

.snapshot

`.snapshot` 指令用來建立呼叫端專案的新實例，並將實例當成專案 `Snapshot` 儲存在資料庫中。專案 `Snapshot` 是一個可執行的專案。

說明 `.snapshot` 指令會建立專案的 `Snapshot`，以及針對您也選擇要建立 `Snapshot` 或複製的相關聯物件建立 `Snapshot`。

利用 `.snapshot` 選項，可指定要建立 `Snapshot` 或複製的物件；相關說明請見下表。如果您沒有指定任何選項，則只會併入專案定義、步驟和標籤。

需要提供 `Snapshot` 名稱，且在專案 `Snapshot` 集當中必須是唯一的。`Snapshot` 名稱也會指派給您要建立 `Snapshot` 的其他物件。

在執行 `.snapshot` 的專案完成後，專案 `Snapshot` 會在使用者介面中顯示成呼叫端專案的子項。其他的 `Snapshot` 物件也會在使用者介面中顯示成其基礎 `Snapshot` 或母項物件的子項。

語法

```
.snapshot -v <"snapshot_name"> [-c <"comment">][-e[f]][-s[f]][-pI][-pC]
[-a][-t][-r][-g]
```

選項	說明
-v <"snapshot_name">	必須提供專案 Snapshot 的名稱。Snapshot 名稱在專案中必須是唯一的。您必須用引號括住名稱。
-c <"comment">	將選用的註解儲存成 Snapshot 的一部分。您必須用引號括住註解。
-e	在建立專案 Snapshot 時，建立專案和步驟環境的 Snapshot。
-ef	加上 f 選項也會為 Snapshot 環境所併入的任何環境（利用「併入環境」變數類型），建立 Snapshot。
-s	在建立專案 Snapshot 時，建立專案和步驟選取元的 Snapshot。
-sf	加上 f 選項也會為 Snapshot 選取元所併入的任何選取元（利用「併入選取元」內容類型），建立 Snapshot。
-pl	在建立專案 Snapshot 時，建立列入專案（或程式庫）與其步驟的 Snapshot。 列入專案或程式庫是由步驟來觸發，並在步驟完成時執行。
-pC	在建立專案 Snapshot 時，建立鏈結的專案（或程式庫）與其步驟的 Snapshot。 專案或程式庫可由專案的通過/失敗條件觸發，或由步驟的通過/失敗條件觸發。這兩種類型的條件式鏈結的專案或程式庫，都會建立 Snapshot。
-a	在建立專案 Snapshot 時，複製配接器鏈結。 配接器鏈結會將配接器新增至專案，並執行配接器程式碼。
-t	針對在專案層次和步驟層次設定的通過與失敗通知事件，複製其通知範本。
-r	在建立專案 Snapshot 時，複製專案暫存器。
-g	複製專案標籤變數的標籤值。 標籤變數會自動複製，但其值不會。如果您沒有複製標籤值，會將它們重設為 1。

.source

說明 使用 .source 指令將原始碼應用程式的配接器新增至專案步驟。程式碼配接器為 Build Forge 物件；它是以程式碼應用程式的配接器範本為基礎。步驟的配接器程式碼會在專案執行時執行。

語法

```
.source <adaptor_name> [entry_name]
```

<adaptor_name> 是必要的；這是在「管理主控台」中指派給配接器的名稱。<adaptor_name> 的大小寫應該和主控台中所使用的大小寫相符。

如果配接器範本具有多個介面函數，請使用 entry_name 選項來指定所要執行的函數。entry_name 選項必須符合配接器範本中指定給 <interface> 元素的 name 屬性。如果 entry_name 中指定的 <interface> 元素不存在或找不到，則會改為執行預設 <interface> 元素。在下列範例中，項目名稱是 By Date。

如果使用配接器鏈結，則會自動呼叫配接器，並執行配接器範本中的第一個介面函數。如果要執行不同的介面，請在配接器範本中針對您確實要執行的介面，將 `default` 屬性設為 `true` (`default="yes"`)。

範例

```
.source MyClearCaseAdaptor
.source MyClearCaseAdaptor "By Date"
```

附註 如果要建立配接器或檢視配接器清單，請選取**專案 > 配接器**。

和 Build Forge 產品一起提供的配接器範本位於：

```
<bfinstall>/interface
```

.stop

```
.stop <state of build>
```

`.stop` 會強制建置立即停止處理。請使用這個指令來終止建置。建置的可能狀態為：通過(P)、失敗(F)、警告(W) 或中斷(B)。

.strsub

```
.strsub <source> <replacement> file [file ...]
```

在一個以上文字檔中使用 `.strsub` 指令來執行基本字串取代。系統會掃描 `<source>` 字串的目標檔；如果找到相符項，系統就會以 `<replacement>` 來取代 `<source>` 字串。`.strsub` 指令會取代每個檔案中，每一行的每個字串實例（原始檔）。

`.strsub` 指令可跨作業系統運作，且伺服器上不需要有任何可用的特定指令。

如果要取代 `about.c` 檔中的 `string _VERSION_`，請使用如下所示的指令：

```
.strsub _VERSION_ 2.34 about.c
```

您必須確實指定一或數個檔案名稱，不能使用萬用字元。例如，類似以下的指令將會失敗：

```
.strsub _VERSION_ 2.34 *.txt
```

不過，由於您可以在指令中使用變數，如果已經在環境中定義 `VERSION` 和 `FILENAME` 變數，類似以下的指令即可正常運作。

```
.strsub _VERSION_ ${VERSION} ${FILENAME}
```

註：使用空格來隔開指令中的參數。

`.strsub` 指令類似於 `.edit` 指令，但差別在於：

- 在大型文字檔或許多檔案中執行取代作業時，`.strsub` 指令會比 `.edit` 快。
- `.edit` 指令可以執行正規表示式的搜尋及取代作業。
- `.edit` 指令會取代每個檔案中，每一行的第一個字串實例 (`search_expression`)。
- `.strsub` 指令會取代每個檔案中，每一行的每個字串實例（原始檔）。

依預設，`.edit` 指令使用「POSIX 延伸正規表示式」語法。如果已編譯代理程式，並支援「Perl 相容正規表示式」，則可在替代表示式後面使用 `"p"` 字元（指出應改用 PCRE 語法）。

在這兩種情況中，代理程式處理程序會解譯表示式兩次。因此，應在正常使用單一反斜線之處，改用四條反斜線。例如：

```
Four slashes escape a literal period:  
\\\\. 
```

或者，您可以使用 `/x` 旗標來抑制反斜線：

```
.strsub/x
```

.test

說明 使用 `.test` 指令將測試應用程式的配接器新增至專案步驟中。測試配接器為 Build Forge 物件；它是以測試應用程式的配接器範本為基礎。步驟的配接器程式碼會在專案執行時執行。

語法

```
.test <adaptor_name> [entry_name]
```

`<adaptor_name>` 是必要的；這是在「管理主控台」中指派給配接器的名稱。`<adaptor_name>` 的大小寫應該和主控台中所使用的大小寫相符。

如果配接器範本具有多個介面函數，請使用 `entry_name` 選項來指定所要執行的函數。`entry_name` 必須符合配接器範本中指定給 `<interface>` 元素的 `name` 屬性。如果 `entry_name` 中指定的 `<interface>` 元素不存在或找不到，則會改為執行預設 `interface` 元素。在下列範例中，項目名稱是 `TestFunction`。

如果使用配接器鏈結，則會自動呼叫配接器，並執行配接器範本中的第一個介面函數。如果要在配接器範本中執行不同的介面，請在您要執行的介面上，將 `default` 屬性設為 `true` (`default="yes"`)。

範例

```
.test MyTestAdaptor  
.test MyTestAdaptor TestFunction
```

附註 如果要建立配接器或檢視配接器清單，請選取**專案 > 配接器**。
和 Build Forge 產品一起提供的配接器範本位於：
`<bfinstall>/interface`

.tset

```
.tset env "<VariableName>=<DesiredValue>" [...]
```

`.tset` 指令會在步驟期間，暫時變更專案設定。您可以使用 `.tset` 來設定還不存在的變數。

這個指令適用於現行步驟。它適用於步驟中的所有指令，以及指定給步驟的任何「列入項目」。不過，執行緒作業可能影響這個行為。範例：

1. 這是一種執行緒步驟，指定了「列入項目」。
2. 「列入項目」的第一個步驟也是一種執行緒步驟。

在這個範例中，`.tset` 指令適用於第一個步驟 (1)，但並不適用於「列入項目 (2)」的第一個步驟。這兩個步驟屬於相同的執行緒區塊。它們單獨執行。請參閱第 330 頁的『執行緒作業：平行執行步驟』。

.tset 指令所設定的值會寫入至工作記錄。它們不會更新環境的資料庫記錄。之後的工作不會受到這些變更的影響。

.unlock

.unlock

.unlock 會使得系統釋放原本使用 .lock 指令鎖定的工作。已解除鎖定的執行現在會列於透過選取工作 > 已完成所顯示的頁面上，並會從「已鎖定」頁面上的清單中移除。

第 21 章 使用工作

這個主題說明如何在「管理主控台」中執行、檢視和管理工作。

關於工作

工作是指正在執行或者已完成執行的專案。

您可以使用**起始**畫面及**工作**畫面來監視工作。此外，您還可以使用**工作**畫面來啟動工作、檢視工作結果及管理號誌。

關於「起始」畫面

起始畫面提供最近工作與系統訊息的相關資訊。

若要存取**起始**畫面，請在左側功能表中按一下**起始**。

作用中作業							
說明 ?							
顯示 1 - 1 (總共 1)							
第 1 頁 (共 1 頁)							
標籤	專案與程式庫	類別	狀態	結果	日期	執行時期	擁有者
BUILD_5	HelloWorld	暫存	正在啟動	----	2010/9/28 上午 4:49	0:00:00	Root User

使用**起始**畫面來檢視最近或現行工作。此外也可以用來檢視通知和系統訊息。請選取一個功能表項目，以檢視：

- 作用中的執行 - 您目前正在執行的工作
- 已完成的執行 - 您已完成的工作
- 系統訊息 - 系統訊息日誌（若要檢視完整訊息，請使用**管理 > 訊息**。）

關於「工作」畫面

使用**工作**畫面來處理執行中的工作及檢視工作結果。

若要存取**工作**畫面，請在左側功能表中按一下**工作**。

工作							
說明 ?							
全部 已完成 執行中 已保存 已鎖定							
顯示 1 - 5 (總共 5)							
第 1 頁 (共 1 頁)							
標籤	專案與程式庫	類別	狀態	結果	日期	執行時期	擁有者
<input type="checkbox"/> BUILD_5	HelloWorld	暫存	已完成	✓ 通過	2010/9/28 上午 4:49	0:00:02	Root User
<input type="checkbox"/> BUILD_4	HelloWorld	暫存	已完成	✓ 通過	2010/9/27 上午 4:34	0:00:00	Root User
<input type="checkbox"/> BUILD_3	HelloWorld	暫存	已完成	✓ 通過	2010/9/26 下午 8:33	0:00:00	Root User

「工作」標籤

您可以使用「工作」標籤，依下列狀態來檢視工作清單：「全部」、「已完成」、「執行中」、「已保存」或「已鎖定」。

請按一下標籤，以便依類型來檢視工作。每一個標籤各有不同的選項，供您管理清單中的工作。

全部 所有執行中的工作會最先呈現，並且依其開始時間排序。接下來的是所有已完成的工作，並且依其完成時間排序。勾選框呈灰色。您可以按一下任何鏈結。沒有特殊的動作可用。

已完成 呈現所有已完成的工作，並且依其開始時間排序。執行中、已保存和已鎖定的工作不會呈現。您可以清除或鎖定已完成的工作。請選取所要工作的勾選框，然後按一下**清除**或**鎖定**。當清除已完成的工作時，如果該工作未完全刪除，會將它移至**已保存**。

執行中 呈現所有執行中的工作，並且依其開始時間排序。您可以取消一個以上執行中的工作。請選取所要工作的勾選框，然後按一下**取消**。

已保存 呈現所有已保存的工作，並且依其開始時間排序。已保存的工作是指已刪除了某些資訊的工作，而這通常是被工作類別之類別定義中指定的自動清除所刪除。您可以清除保存的工作。請選取所要工作的勾選框，然後按一下**清除**。

已鎖定 呈現所有已鎖定的工作，並且依其開始時間排序。您可以解除鎖定一個以上已鎖定的工作。請選取所要工作的勾選框，然後按一下**解除鎖定**。

註：您無法直接刪除已鎖定的工作。如果要刪除這些工作，請先解除鎖定，再跳至**已完成**標籤，然後使用**清除**。

工作清單欄位

欄位顯示每一項工作的資訊。

在任何標籤中，工作清單都是依完成時間來呈現，最近的工作會排在第一位。在每一項工作中，會在直欄內顯示下列資訊：

標籤 建置標籤（顯示成鏈結）。按一下此項時，可查看工作步驟的日誌。

專案和程式庫

顯示成鏈結。按一下此項時，可查看用於工作的專案。

類別 顯示成鏈結。按一下此項時，可查看用於工作之類別的類別定義。

狀態 「等待中」、「執行中」、「已完成」、「已保存」或「已鎖定」其中一個。

- 「等待中」狀態表示工作尚未啟動，可能是因為「執行中」佇列已滿，或工作在等待號誌。「執行中」佇列大小由系統設定**執行佇列大小**來定義。
- 「執行中」佇列中的工作通常都在執行中。不過，如果「執行中」佇列含有相同專案的多個工作，目前有一個以上工作在執行中，則取決於專案的**執行限制**內容。請參閱第 289 頁的『變更專案內容』。

結果 「通過」、「失敗」、「失敗但繼續」、「警告」、「已停止」或「已取消」其中一個。

日期 工作啟動前：要求工作的日期和時間。

工作啟動後：啟動工作的日期和時間。

執行時期

若為執行中的工作，則表示目前已經歷的時間。若為已完成的工作，則顯示經歷的時間總計。

擁有者 啓動工作的使用者。

過濾和排序工作清單

您可以過濾和排序任何標籤中的工作清單。

透過過濾，可限制工作清單中只會顯示符合您指定字串的工作。透過排序，可依任一直欄中的值，來排序清單。

- **過濾**：您可以過濾工作清單，只檢視包含特定字串的工作。只要任何直欄中有該字串存在，就算是符合過濾器。
 - **新的過濾器**：在方框中鍵入字串，然後按一下**過濾器**。
 - **現有的過濾器**：按一下方框旁的箭頭，然後選取要使用的過濾字串。
- **排序**：您可以依照任何單一直欄中的值，來排序工作清單。請按一下直欄名稱旁的箭頭。如果您按下多次，排序就會在下列的排序順序中循環。
 1. 遞增（在直欄標頭中，向下箭頭呈強調顯示）
 2. 遞減（在直欄標頭中，向下箭頭呈強調顯示）
 3. 無過濾器

執行工作並檢視結果

您可以執行、排定、取消和重新啓動工作，並檢視工作結果。

啓動工作

有數種方式可以啓動工作。

開始之前

執行工作之前，專案必須已定義環境和選取元。

程序

- 從「工作」畫面：選取**工作 > 啓動 > 啓動專案**，然後按一下專案名稱。按一下**啓動專案**時，即會顯示「啓動專案」頁面。

工作詳細資料

工作步驟

專案參數

Snapshot: 基本 Snapshot
選取元: Selector1
類別: 暫存
標籤格式: BUILD_\$B
標籤範例: BUILD_4

專案環境

工作將以預設環境起始

儲存環境

☐

專案標籤

☐ 可編輯的標籤

B

4

- 從「專案」畫面：按一下**快速啟動**圖示 。


檢查的條件如下所示。如果檢查通過，會立即啟動專案。

- 專案含有一個以上步驟。
- 在專案的環境中，沒有任何變數在「於專案」變數內容中設為必須變更。

如果檢查失敗，則會停用**快速啟動**圖示：。

以這種方式來執行專案，使用選取元、類別、標籤及環境變數的預設值。

如果「啟用快速啟動」系統設定是設為是，則「專案」頁面會檢查所有專案。這可判斷它們是否適合進行「快速啟動」。如果定義了大量專案，「專案」頁面可能要花

很長的時間來顯示專案清單。另外還有一個圖示指出可以啟動專案：。在 7.1.1.1 版之前，這是預設行為。

- 從「步驟」畫面：當檢視專案的步驟時，請按一下**啟動專案**。此方法會顯示專案的「啟動專案」頁面，您可在其中變更專案參數、環境變數值，並選取要從作業中排除的步驟：
 - 選取新的專案參數值。
 - 編輯專案標籤變數值。
 - 編輯專案環境變數值。如果您要將變更儲存為這些變數的新預設值，請選取**儲存環境**勾選框。
 - 選取**工作步驟**標籤以顯示專案步驟清單。您可以按一下直欄標頭中的勾選框，以選取您要併入的步驟。如果您要排除步驟，請清除步驟的勾選框。
 - 如果步驟是條件類型（Java 引擎中的 IF 條件和 ELSE 條件），則兩者都會顯示，您可以選取要執行的條件（IF 或 ELSE）。請參閱上方的影像。
 - 如果您只選取 IF 條件，則不評估條件就會執行 IF。
 - 如果您只選取 ELSE 條件，則不評估條件就會執行 ELSE。
 - 如果您同時選取 IF 與 ELSE 條件，則使用條件評估來執行指令。
 - 如果 IF 與 ELSE 條件都取消選取，則對選取的步驟來執行條件。（如果 IF 與 ELSE 條件都取消選取，則只能選取一個條件（IF 或 ELSE）。）

選擇完成後，請按一下**執行**以啟動專案。

結果

正在執行專案時，請檢視工作 > 執行中頁面，檢查專案狀態。

如果要檢視工作結果，請選取工作 > 已完成來顯示已完成的工作。按一下「標籤名稱」即可存取用來檢視工作結果的選項。

檢視工作結果

關於這項作業

您可以在所有工作標籤上檢視工作結果。

程序

1. 按一下工作。已選取**已完成**標籤。
2. 按一下想要的標籤。
3. 按一下您要檢視的工作標籤。會顯示下列工作結果：
 - 主畫面中會顯示步驟清單。每一個步驟各有一些資訊直欄。
 - 步驟：步驟號碼。
 - 步驟名稱：步驟的名稱。步驟名稱前面的圖形指出步驟類型。會顯示下列步驟類型：
 - a. 步驟
 - b. 含執行緒作業
 - c. 廣播
 - d. 結合
 - 請按一下步驟名稱，查看步驟日誌。
 - 結果：步驟結果。會顯示下列步驟結果：
 - a. 通過
 - b. 失敗
 - c. 取消
 - 日誌版本：如果此步驟重新啟動，則下拉清單會顯示此步驟的日誌版本。日誌版本 1 表示起始執行的步驟日誌。重新啟動之後，日誌版本號碼會增加 1。
 - 伺服器（選取元）：指出伺服器何時執行之步驟的名稱。也會顯示用來選擇伺服器的選取元。
 - 執行時期：步驟的經歷時間，以小時、分鐘或秒為單位。
 - 鏈結：如果步驟已啟動另一項工作，則會顯示連至該工作的鏈結。
 - 左側功能表底端會出現功能表，並會顯示下列項目：
 - 結果：在主畫面中顯示步驟清單。
 - 資料清單：在主畫面中顯示資料清單 (BOM)。BOM 包含鏈結，其顯示下列各項：
 - a. 工作步驟
 - b. 步驟資訊清單
 - c. 檢查點（如果已使用檢查點的話）
 - 附註：顯示針對這項工作所輸入的附註。

- 步驟日誌（會開啓此項目，並顯示步驟名稱。）：顯示步驟清單。請按一下某個步驟，查看其步驟日誌。
- 鏈結：顯示當作鏈結啓動之任何工作的工作號碼。

註：使用 `.run`、`.runwait` 或兩者啓動的工作不會產生此圖示。

4. 從任何步驟清單中選擇一個步驟、選取步驟的日誌版本，然後按一下步驟名稱。一開始會選取所有的種類。

如果您是檢視「執行中」工作中的步驟，而該步驟尚未完成，則您會看到部分日誌，可能看不到所有的種類。若要更新執行中步驟的日誌視圖，請按一下建置標籤（位於頂端，顯示為**工作 >>** 標籤），然後按一下步驟。檢視步驟日誌時，請按一下**顯示全部**以顯示所有步驟。如果只要檢視幾個步驟，請在**顯示 1 -m of n** 控制項中輸入 *m* 的值，然後按 **Enter** 鍵。

5. 若要過濾步驟日誌：請勾選或清除種類，然後按一下**重新整理**。

```
1 7/8/10 4:43 PM STEP      Step using selector 'My selector'.
2 7/8/10 4:43 PM MANIFEST  BF_LAST_UPDATE=1278618783
3 7/8/10 4:43 PM MANIFEST  BF_NAME=rbf-14
4 7/8/10 4:43 PM MANIFEST  BF_LOADRATIO=0.3333333333333333
5 7/8/10 4:43 PM MANIFEST  BF_JOBS=1
6 7/8/10 4:43 PM MANIFEST  BF_AGENT_VERSION=version
7 7/8/10 4:43 PM MANIFEST  BF_LAST_REFRESH=1278618783
```

工作狀態的 RSS 資料資訊來源

您可以使用 RSS 資料資訊來源，來追蹤和過濾個別工作的狀態。工作的 Build Forge RSS 資料資訊來源所顯示的資訊，與 Build Forge「管理主控台」中的伺服器狀態相同。

如果要訂閱工作狀態的 RSS 資料資訊來源，請執行下列動作：

1. 在 Build Forge「管理主控台」中，選取**工作**。

Web 瀏覽器會偵測 RSS 資訊來源，並且在瀏覽器位址列顯示 RSS 圖示。

2. 在 RSS 聚集器工具中，載入 Build Forge RSS 資料資訊來源。

例如，複製 URL，將它新增至 RSS 資料資訊來源清單中。或者，您可以拖放 RSS 圖示，將 URL 新增至 RSS 資料資訊來源清單中。

3. 訂閱 RSS 資料資訊來源來儲存 URL，並且只要一有更新，就接獲通知。

註：

- 如需載入 URL 和訂閱 RSS 資料資訊來源的相關資訊，請參閱您 RSS 聚集器工具的相關文件。
- 若要透過非英文的 RSS 資料資訊來源來檢視 Build Forge 工作狀態、系統訊息或伺服器狀態，您的 RSS 聚集器工具必須支援 UTF-8 多位元組字元編碼。
- RSS 資訊來源會使用時間戳記來報告工作活動。時間戳記是以針對您的 Build Forge 使用者帳戶所設定的時區為基礎。如果您的瀏覽器是屬於另一個時區，請將它設為與您的 Build Forge 帳戶相同的時區，以接收準確的時間戳記。

重新啓動失敗的工作

如果工作失敗，您可以重新啓動它。當重新啓動時，會在相同標籤下啓動新的執行。它會從失敗點繼續執行。

若要重新啟動工作，請在**工作 > 已完成**標籤上，按一下建置清單中的工作標籤。系統會顯示建置的相關資訊，並在畫面頂端的附近加上**重新啟動工作**按鈕。

1. 按一下**重新啟動工作**。

即會顯示「重新啟動」頁面。

2. 選取選項。如果您希望系統在其步驟中從專案記錄接收到指令的更新，請設定**同步指令**內容。若未設定，則會按照工作起始時的相同方式來輸入指令。
3. 在「重新啟動」頁面上，按一下**重新啟動**按鈕。

當工作重新啟動與新的工作相比較時，會有下列顯著差異：

- 它會使用與失敗執行相同的標籤號碼，取代**已完成** 清單中的失敗執行。
- 依預設，它會從失敗的步驟起始，而且不會重複先前的任何執行步驟。不過，您可以在工作重新啟動時選擇您的執行步驟。
- 依預設，系統會提供您在上次執行時所提供的相同環境變數值。然而，您可以在重新啟動工作之前變更這些變數值。
- 系統只會根據重新啟動工作期間所執行的步驟，來對工作的成功與否求值。上次執行時所產生的失敗並不會影響重新啟動之工作的狀態。

如需工作重新啟動與配接器鏈結步驟的相關資訊，請參閱第 445 頁的『關於配接器鏈結』。

重新啟動廣播步驟

在執行時期，標示為**廣播** 步驟的步驟會顯示為執行該步驟的每部伺服器的一組步驟。在重新啟動及啟動頁面上，則會顯示為單一步驟。

如果有某個廣播步驟失敗，則所有步驟可能都會顯示為失敗。請在重新啟動之後執行下列動作：

1. 選取單一廣播步驟。
2. 選取**同步指令**內容。
3. 設定其他所需的選項。
4. 在「重新啟動」頁面上，按一下**重新啟動**。

它會在可用的廣播伺服器上重新執行。

重新啟動 While 迴圈

假設是類型為「While 迴圈」的步驟，就會根據 **BF_ITERATION** 系統變數的值，讓步驟從發生失敗所在的反覆運算重新啟動。工作流程的範例如下所示：

1. 工作進入類型為「While 迴圈」的步驟
2. 條件求值為 **true**
3. **BF_ITERATION** 設為 1
4. 順利執行「指令」與「列入」
5. 步驟迴圈
6. 條件求值為 **true**
7. **BF_ITERATION** 設為 2
8. 在執行「指令」或「列入」期間停止工作

如果重新啟動上述工作，它會從「反覆運算 2」重新啟動，並試圖執行步驟的「指令」與「列入」。

使用資料清單

系統在完成每一個工作之後，會產生一份資料清單 (BOM)。BOM 包含工作步驟的相關資訊，以及針對產生之檔案所做的變更。您可以提供 BOM 給工作的使用者（例如，品管部門），幫助他們瞭解新工作的內容。它可以作為建置和發行程序的審核解決方案。您可以借助 BOM，取得工作內容的完整文件。其內容包含結果、附註、環境、檔案清單以及程式碼變更。您可以用它來比較和彙總整個企業的建置狀態。

系統會自動為每一項工作產生一份 BOM，不過您也可以步驟中使用點指令，讓系統儲存其他有關建置前後檔案狀態的資訊。

顯示「資料清單 (BOM)」

檢視已完成的建置（工作 > 已完成）時，系統會依預設顯示步驟標籤。按一下 **BOM** 標籤，以顯示資料清單。

按一下要展開之種類旁的 +。您看到的實際種類，視專案以及系統配置而定：

- 「專案步驟」種類會顯示在每一項工作中，並提供針對此工作所執行之步驟的相關資訊。
- 只有在系統包含原始碼配接器，且專案有指向該配接器的鏈結時，才會顯示「原始檔變更」種類。如需詳細資料，請參閱第 445 頁的『配接器和工作結果』。在您配置配接器時，可以變更「原始檔變更」種類的格式（甚至名稱）。
- 只有在專案併入 .scan 指令時，才會顯示基礎線及檢查點區段。

使用 .scan 指令來新增基準線及核對點

您可以使用 .scan 指令來新增其他資訊至 BOM。當 .scan 指令執行時，系統會將檔案狀態的相關資訊儲存在步驟的工作目錄中。本節顯示其用法範例。另請參閱 第 362 頁的『.scan』中的參照資訊。

該指令有二種形式。

.scan baseline

將所有檔案的清單儲存在步驟的工作目錄樹狀結構中，並儲存各檔案的 MD5 值。系統會針對工作，在 BOM 中顯示清單。您可能會想在執行某些設定步驟以及移出適當的一組檔案之後，發出這個指令。一個專案可以有多個 baseline 指令，但是當 .baseline 指令執行時，每個 baseline 指令都會將清單重設為步驟的工作目錄狀態。

.scan checkpoint

儲存自前次在專案中執行 .scan baseline 或 .scan checkpoint 之後，所有新建、變更及刪除的檔案清單，包含各檔案的 MD5 值。若是使用 .scan baseline 指令，系統會在 BOM 中顯示清單。您必須先發出 .scan baseline 指令，才能發出專案中的第一個 .scan checkpoint 指令。在 .scan baseline 指令之前的 .scan checkpoint 指令會被忽略。

下列範例顯示 .scan baseline 及 checkpoint 指令如何共同運作：

號碼	步驟	步驟之後的檔案	BOM 資料
1	移出起始檔案	config.c execute.c	
2	.scan baseline	config.c execute.c	基礎線： config.c execute.c
3	新增資料檔	config.c execute.c data.txt	
4	.scan checkpoint	config.c execute.c data.txt	核對點 1： 新增的 data.txt
5	新增其他資料檔	config.c execute.c data.txt data2.txt data3.txt	
6	刪除 data.txt	config.c execute.c data2.txt data3.txt	
7	.scan checkpoint	config.c execute.c data2.txt data3.txt	核對點 2： 新增的 data2.txt、 data3.txt 刪除的 data.txt

將 BOM 匯出為 XML 檔

本主題說明 bfbomexport 指令的語法、用法和選項說明。

若為 Windows，Build Forge 指令位於 Build Forge 安裝目錄，若為 UNIX 和 Linux，則位於 <bfinstall>/Platform 目錄。

說明

使用 bfbomexport 指令將工作的「資料清單」(BOM) 匯出至 XML 檔案。收集了 BOM 資訊後，bfbomexport 會將該資訊儲存至您所指定的位置和檔名。

如果要指出您要儲存至 XML 檔案的 BOM，您必須指定工作的專案和建置。

bfbomexport 指令必須從「管理主控台」的 Build Forge 安裝目錄執行，若為 UNIX 或 Linux，則是從 /Platform 目錄執行。

語法

```
bfbomexport [-f filename] [-p projectID | -P projectName]
            [-b buildID | -t buildTag] [-L] [-H]
```

選項

選項	說明
-f filename	選用路徑和/或檔名。工作的 BOM 會以 XML 格式儲存。如果沒有提供檔名，會將 BOM 寫至標準輸出 (stdout)。如果沒有提供路徑名稱，則會使用現行工作目錄。
-p projectID	工作的專案 ID。(bfexport 指令加上 -l 選項可列出專案 ID。)
-P projectName	專案名稱。
-b buildID	建置 ID。
-t buildTag	建置標籤名稱。
-L	併入步驟日誌。
-H	說明訊息。

排定工作

使用**排程**畫面，排定在未來的某個時間執行專案，或依重複間隔定期執行專案。例如，您可以設定專案每小時或每天執行。

若要檢視工作排程，請按一下**排程**。這時會顯示現行月份的行事曆，底下會有一個畫面，用來建立及修改排程項目。

行事曆中的每一天都會顯示當天排定的專案數目。將滑鼠移到某一天，可以查看當天排定之工作的名稱和排程參數。

如果您排定了多個專案，系統會顯示一個下拉框和**過濾**按鈕，供您依專案來過濾行事曆。

使用者介面配置 主控台 報告 登出: Root User

排程 新增排程作業 說明 ?

排程清單 行事曆

過濾條件

沒有要顯示的頁

2010年 9 月

星期日	星期一	星期二	星期三	星期四	星期五	星期六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2

(新建項目) 儲存排程 複製排程 重新同步化環境 刪除排程

排程詳細資料 環境

說明: 存取: 安全人員 擁有者: Root User

專案: --類別清除排程-- 類別: 暫存 模式: 作用中

環境: --專案預設值-- 自動同步環境: 否

選取元: --專案預設值-- 分鐘: 0 小時: 0 日期: * 月: * 日: *

如果要將專案放入排程器中，請執行下列動作：

1. 按一下左側功能表中的**排程**。
2. 按一下**新增排定的執行**。
3. 在**說明**中，輸入排程項目的說明。
4. 從**專案清單**中，選取一個專案。
5. 現行的專案 Snapshot 顯示在**專案**下的欄位內。如果您想以不同的 Snapshot 來執行排定的工作，請從清單中選取一個 Snapshot。
6. 在**模式**中，選擇「作用中」（預設值）、「非作用中」或「一次」。如果設為「一次」，在下一次符合時間設定時只會執行工作一次。如果設為「作用中」，每當符合時間設定時都會執行工作。
7. 選用項目：您可以置換下列專案內容。您在這裡選擇的設定只適用於這個排定的工作實例。
 - **存取**：如果指定的話，排定的工作就會使用指出的「存取」內容。
 - **擁有者**：如果指定的話，則會比照由指出的擁有者以手動啟動方式來執行排定的工作。
 - **存取**：如果指定的話，排定的工作會使用指出的「類別」。
 - **環境**：針對這個排定的工作，無論將其明確設定或保留為「專案預設」，都會建立一個環境副本。如果指定了環境，您也可以設定變數的起始值。

重要：製作副本後，針對排定工作而製作的環境副本中，並不會自動更新原始環境及其變數的變更。您可以手動或自動更新副本，以反映對原始環境所做的變更。如果要進行手動更新，請按一下**重新同步化環境**。如果要進行自動更新，請將**自動同步化環境**設為「是」。如果選擇自動更新，則會在排定的工作每次執行時更新環境。但是請注意，排程的**自動同步化環境**必須設為「否」，才能使用「環境」標籤。

- **選取元**：如果指定的話，工作會使用指定的選取元。

- 指定時間設定。請填寫「分鐘」、「小時」、「日期」、「月份」和「日」的值。請參閱『排程參數』，以瞭解要用的值。
- 按一下**儲存排程**。

「下次執行」直欄會顯示「計算中」，然後顯示下次執行工作的排定時間。

建立排程之後，排程名稱即會出現在行事曆中。它旁邊的圖示會顯示它的「模式」。您可以按一下圖示來變更模式。每次按一下，都會變更。

- 綠圈：作用中
- 藍圈：一次
- 紅圈：非作用中

試圖執行排定的工作時，系統會检查工作佇列、**強迫執行限制**系統設定，也可能會檢查專案的**執行限制**內容。如果排程啟動時，已有一個以上專案工作在執行，系統行為會視**強迫執行限制**系統設定而定，如下列表格說明。

強迫執行限制值	系統檢查執行限制值嗎？	系統啟動工作嗎？
是	是	是，如果執行工作數目小於 執行限制值 否，如果執行工作數目等於 執行限制值
否	否	是

註：如果之前的執行尚未完成，請將**執行限制值**設為 1，將**強迫執行限制**設為「是」以跳過執行。

排程參數

本節說明可用來指定何時執行工作的參數。

您可以使用一系列欄位來指定工作的執行時間。

欄位	說明	範圍
分鐘	分鐘數。	0-59
小時	當天的小時。	0-23
日期（當月的日期）	月份日期。	1-31
月	當年的月份。	1-12
日（平日）	星期幾（星期日 = 0）。	0-6

您在欄位中輸入的數值，可使用下列方式來表示：

- 使用星號 (*) 來表示範圍內的所有有效值。星號之後可接著一個正斜線 (/) 和步驟值。例如，「小時」欄位中的 */2 值，表示每 2 小時執行一次工作。

註：星號在「日期」與「星期幾」中的解譯較為特殊。它會比對其在 UNIX/Linux Cron 機能中的意義。若有一個欄位含有文字值，而另一個欄位含有星號，則只會採用文字值的頻率。例如，如果「月」為 *，「日期」為 *，而「星期幾」為 1，則會每星期一執行。另請參閱以下的「星期幾/日期」範例。

- 使用以連字號區隔的數字範圍。例如，「小時」欄位中的 8-11 指定 8、9、10 和 11 點。範圍後面可接著一個正斜線 (/) 和步驟值。例如，「小時」欄位中的 0-23/2，表示每隔一小時執行一次工作。
- 使用以逗點區隔的一組數字（或範圍）清單。例如，1, 2, 3-5, 8。

排程是根據欄位的指定值建構而成。範例：

值					
想要的排程	分鐘	小時	日期	月	日
每天下午 5 點執行工作	0	17	*	*	*
每週的星期一下午 4:30 執行工作	30	16	*	*	1
在每個工作日中，每隔半小時執行工作（跳過週末）	*/30	*	*	*	1-5
每隔一天的上午 12:30 執行工作	30	0	*/2	*	*

「星期幾」與「日期」欄位以特殊的方式來使用星號。

- 如果其中一個含有星號值，但另一個含有文字值，則會以擁有文字設定的欄位為依據，來執行工作。
- 如果兩者皆非星號值，只要出現任一條件，就會執行工作。
- 如果兩者皆有星號值，則會每天執行工作。

「星期幾」/「日期」範例：

值					
想要的排程	分鐘	小時	日期	月	日
每月第一天的上午 1:01 執行工作。「日期」使用文字值；「星期幾」使用星號。	1	1	1	*	*

值					
想要的排程	分鐘	小時	日期	月	日
當月每星期一的上午 1:01 執行工作。「日期」使用星號；「星期幾」使用文字值。	1	1	*	*	1
當月每星期一的上午 1:01 和當月第一天（不論當天是否為星期一）執行工作。「日期」和「星期幾」皆使用文字值。	1	1	1	*	1
每天上午 1:01 執行工作。「日期」和「星期幾」皆使用星號值。	1	1	*	*	*

排定清除工作的類別

您可以針對工作類別建立排程，來控制系統進行舊工作的清除作業時機。建立這些排程就像您建立排程來啟動專案一樣，但要選取的是「類別清除排程」選項，而不是專案。當您這麼做時，系統會依照您在排程中選取的時間，檢查您所選類別中是否有工作要清除。系統會對符合的每一個工作建立清除工作，並將該工作置於等待佇列中。

依預設，系統會根據「清除檢查時間」系統設定所設定的間隔（預設值為每隔 15 分鐘），檢查是否有工作應該清除（根據定義自動刪除規則的類別內容）。此行為會導致清除工作與一般工作競爭系統資源。

如果您針對某個類別的工作建立排程，則系統只在排程啟動時檢查要清除的工作。如果沒有特定類別的排程存在，系統會對該類別的工作使用預設行為。如果想要限制所有清除於特定的時間執行，則至少必須為每個類別建立一個排程。

如果要定義清除排程，請執行下列動作：

1. 像往常一樣建立排程，但在「專案」欄位中選取「類別清除排程」。
2. 在「類別」欄位中選取類別。

管理工作

您可以鎖定、保存和刪除工作。

鎖定工作

您可以鎖定專案以防止被自動刪除。

如果要鎖定工作，請按一下工作畫面中的**已完成**標籤，再選取建置，然後按一下**鎖定**。

如果要解除鎖定工作，請按一下工作畫面中的**已鎖定**標籤，再選取已鎖定的建置，然後按一下**解除鎖定**。

系統不會自動清除已鎖定的工作（例如，當執行的類別清除內容呼叫以要求刪除工作時）。您仍然可以手動刪除已鎖定的工作。

按一下**已鎖定**標籤，檢視處於已鎖定狀態的工作。此類專案不會出現在**已完成**頁面中。

刪除工作

下列主題說明數種用來刪除工作的方法。

將工作從「已完成」標籤中刪除

您可以用手動的方式，從工作畫面的「已完成」標籤中刪除工作。

您可以用手動的方式從此清單中刪除一個以上工作。當您這麼做時，系統會以每次執行的類別為基礎，來啟動清除工作。這項程序就如同是類別內容觸發了自動刪除。系統會保存工作（如果尚未完全刪除的話），並刪除指定給該類別之工作的資料。請參閱第 304 頁的『類別』。

如果要刪除工作：

1. 選取**工作 > 已完成**頁面，以顯示已完成的執行清單。
2. 按一下表格右端的一個以上勾選框，選取要刪除的建置。
3. 從清單底端的清單框中，選取「清除」選項。
4. 按一下「執行」按鈕。

如果將類別設為刪除輸出檔，但保留主控台資料，則從**已完成**清單中刪除工作時就會刪除輸出檔，並且會將工作的項目從**已完成**清單移至**已保存**清單。

將工作從保存清單中完全刪除

選取**工作 > 已保存**以顯示保存清單。這份清單會顯示檔案已被刪除之工作的相關資料。您可以在此刪除工作，也可以從**工作 > 已完成**清單中刪除工作。從保存清單刪除工作時，工作的所有追蹤資料會從資料庫移除，並從應用程式所提出的統計資料報告中去除。

自動刪除工作

當工作的類別內容判定應將工作刪除時，系統就會自動將其刪除。您可以利用這個特性來防止資料累積，以及刪除工作群組。

如果您為類別建立排程，當排程啟動時，系統只會檢查要清除的工作。請參閱第 382 頁的『排定清除工作的類別』。

當您希望刪除工作，並且要確定是否已刪除時，請在啟動或排定工作之前，先檢查下列設定：

1. 設定一或數個類別的刪除內容，讓系統在特定的天數之後，及/或在累計特定數量的工作之後，將工作刪除。
2. 將所使用之專案的**類別**內容設為適當的類別。

如果您產生數個工作，然後要將其刪除，您可以暫時變更相關類別的刪除內容。或者，您可以在**已完成**標籤上選取多個建置，並將其刪除（按一下每個工作旁的勾選框來選取，然後按一下**清除**按鈕）。

例如，如果您在前一天產生了許多「正式作業」工作，則可使用下列程序來將其刪除：

1. 請注意「正式作業」類別的現行設定。
2. 將「正式作業類別」的「星期幾」內容變更爲 1，然後按一下**儲存類別**按鈕。在延遲 15 分鐘之後，系統就會開始刪除存在一天以上的工作。
3. 刪除工作之後，再將「正式作業類別」的內容變回原來的設定。

工作的工作目錄

系統會爲每一個工作建構工作目錄，讓每一次都能在一個含標籤的隔離區執行。它以專爲伺服器路徑、專案名稱和標籤提供的值，爲工作目錄命名。

當它執行指令時，系統會在專爲步驟指定的目錄下啓動指令。依預設，該目錄是工作的工作目錄，不過您也可以指定某些相對於伺服器路徑內容的其他目錄。本節中的主題會說明如何建立路徑和目錄。

註：當它執行專案時，系統會建構專案目錄（如果此目錄還不存在的話）以及工作目錄。它不會建構伺服器目錄（在伺服器的路徑內容中指定），也不會建構步驟目錄（步驟的目錄內容）。

工作的工作目錄名稱

下列範例顯示系統如何使用數個值，針對在單一伺服器上運作的工作建構工作目錄：

系統值	所建立的目錄
伺服器的「路徑」欄位：C:\BuildForge 專案名稱：My Project 標籤：Job 5	C:\BuildForge\My_Project\Job_5\ 系統只會建立路徑的 粗體 部分。您必須在執行專案前建立伺服器目錄，否則它會失敗。

註：當系統建立專案目錄時，它會將無效的相對 **Dir** 字元系統設定中所指定的字元，變更爲底線字元。依預設，設定中包含空格和反引號字元，因此名爲 My Project 的專案會收到名爲 My_Project 的專案目錄。

如果工作是在多部伺服器上執行，系統會在每部伺服器上建立工作目錄。由於專案中的每一個步驟都能指定不同的伺服器，因此系統可能會建立多個目錄。下列範例說明使用兩部伺服器的專案：

系統值	所建立的目錄
專案伺服器：ServerA，路徑值爲 C:\BuildForge 專案的第三個步驟指定 ServerB，路徑值爲 C:\deployments。（其他所有步驟都使用預設（專案）伺服器，ServerA）。 專案名稱：My Project 標籤：Job_6	在 ServerA 上： C:\BuildForge\My_Project\Job_6\ 在 ServerB 上： C:\deployments\My_Project\Job_6\ 系統只會建立路徑的 粗體 部分。

在以上範例中，您可以預期從步驟產生的任何輸出檔，依預設都會建立在 C:\BuildForge\My_Project\Job_6 目錄中，除了使用 C:\deployments\My_Project\Job_6 目錄的第三個步驟之外。

建構步驟的目錄路徑

當系統執行步驟時，可以從系統為工作所建構的目錄開始；如果步驟的路徑選項設為「絕對」，則可以忽略專案和標籤目錄。

- 當步驟的路徑設為「絕對」時，系統會將伺服器路徑和步驟的目錄欄位加在一起，來建構步驟的路徑。目錄欄位中的值，是伺服器工作目錄的相對路徑。

適用於步驟的值	適用於指令的結果路徑
伺服器的路徑欄位：C:/BuildForge 步驟的目錄欄位：/bin 步驟的路徑設定：絕對	C:\BuildForge\bin 使用此表單來存取伺服器目錄上的目錄。
伺服器的路徑欄位：C:/BuildForge 步驟的目錄欄位：/（預設值） 步驟的路徑設定：絕對	C:\BuildForge
伺服器的路徑欄位：C:/BuildForge 步驟的目錄欄位：C:/temp 步驟的路徑設定：絕對	C:\BuildForge\C:\temp （此範例會造成錯誤；步驟將會失敗。）

註：您可以利用反斜線或正斜線輸入路徑值。系統會以反斜線儲存路徑值，並在 Windows[®] 電腦上依所需將它們變更為正斜線。

重要：將指令的路徑設為根目錄，會導致非預期的刪除或修改系統檔。如有可能，請在不同的目錄中執行指令。不過，如果您要將系統管理作業自動化，且需要路徑是根目錄，請將伺服器的路徑設為根目錄，將步驟的目錄設為 ./，並將步驟的路徑設為「絕對」。

- 當步驟的路徑選項設為「相對」時，系統會將伺服器路徑、專案名稱、標籤和步驟的目錄欄位加在一起，來建構步驟的路徑。目錄欄位中的值，會變成工作之工作目錄的相對路徑。

適用於步驟的值	適用於指令的結果路徑
伺服器的 路徑欄位 ：C:/BuildForge 專案名稱：My Project 標籤：Job_5 步驟的 目錄欄位 ：/bin 步驟的 路徑設定 ：相對	C:\BuildForge\My_Project\Job_5\bin 路徑的 粗體 部分如果原本不存在，則系統會加以建構。 註 ：當系統建立專案目錄時，它會將 無效的相對 Dir 字元 系統設定中所指定的字元，變更為底線字元。依預設，設定中包含空格和反引號字元，因此名為『My Project』的專案會收到名為『My_Project』的專案目錄。

如果伺服器的**路徑欄位**或步驟的**目錄欄位**中所指定的目錄不存在，則步驟會失敗；系統不會建立這些目錄。步驟的「目錄」欄位所指定的路徑部分，必須在專案期間由之前的步驟明確地建立。

一般來說，專案中稍早的步驟會將目錄樹狀結構從原始碼控制中移出，而下列步驟則會處理這些目錄。

當您新增步驟時，系統會記住您最後一次為**路徑**所選擇的設定（可為「相對」或「絕對」），並使用該設定作為新步驟的預設值。

路徑中的斜線

當您在**路徑**或**目錄欄位**中輸入路徑時，系統會將所有的反斜線全部轉換為正斜線。當系統建立實際路徑時，如果使用的伺服器為 Windows[®] 電腦，系統會將所有的正斜線全部轉換成反斜線。因此，您可以在路徑中使用任何一種類型的斜線。

註：系統不會在步驟的指令欄位中修改斜線。請視需要，對目標伺服器使用反斜線或正斜線。

號誌

號誌是系統中設定互斥（互斥旗標）資源的廣域信號旗號。請利用它們讓某些處理程序等待其他處理程序完成。

請使用**工作 > 號誌**，來檢視使用中的工作號誌。您也可以清除號誌，萬一當掉或是取消的工作無法釋放其號誌時，可能會需要執行這個動作。



您可以透過一對點指令來實作號誌：`.semget` 和 `.semput` 指令。請使用 `.semget` 指令「抓取」標籤：當步驟取得標籤之後，嘗試取得相同標籤的其他步驟（在任何專案中）必須等到原始要求程式釋放它為止。

註：含有 `.semget` 指令的步驟會等待到釋放號誌為止。如果工作失敗，而讓其號誌維持作用中，必須先手動清除號誌，任何使用該號誌的工作才能再次執行。

例如，假設您有一個建立印表機驅動程式的程式，您希望該程式一次只由一個處理程序使用。在每一個呼叫該程式的專案內，使用下列指令行設定三個步驟：

步驟	指令行
取得號誌	<code>.semget \$BF_PROJECTNAME_PHYS</code>
執行驅動程式建立程式	<code>printdrivermaker.exe windows</code>
釋放號誌	<code>.semput \$BF_PROJECTNAME_PHYS</code>

您可以為組織中的主要資源建立號誌，例如負載重的伺服器或具有單一使用者授權的軟體程式。每一個使用您所要保護之資源的步驟應該以 `.semget` 和 `.semput` 指令加以封裝。

號誌遵守下列規則：

- 使用的號誌點指令必須是該步驟的唯一點指令。
- 系統會依現狀使用標籤；請避免在標籤中使用特殊字元或尾端空白，因為這可能會讓標籤剖析器迷惑。
- 號誌是廣域的，除了執行緒區塊之外，可利用號誌使不同專案同步化。
- 第一次存取號誌時即建立號誌。
- 專案取得的號誌會在專案終止時自動釋放。
- 如果有兩個步驟同時要求一個號誌，則不保證先由哪一個步驟取得鎖定。

手動清除號誌

當專案終止時，系統會自動釋放專案所建立的任何號誌。不過，如果因專案異常終止而導致號誌沒有釋放，您可以手動清除它。

1. 選取**工作 > 號誌**，以顯示「號誌」清單。
2. 選取您要清除的號誌。
3. 按一下**清除**。

第 22 章 使用報告

這個主題說明 Rational Build Forge 所提供的內建報告。此外，亦說明如何使用「快速報告」（一種另行授權的報告工具）來建立和執行報告。

關於報告


報告的種類有下列兩種：一種是從「效能」與「查詢」畫面中存取的內建報告，一種是您使用「快速報告」工具建立的報告。

如果要著手執行，請按一下右上角的**報告**標籤，以顯示用來產生報告的選項：

- 選取**效能**畫面，可檢視標準報告。
- 選取**查詢**畫面，可檢視和執行預先定義的查詢報告。
- 如果您想建立自己的報告，請選取**快速報告**畫面。「快速報告」是 Rational Build Forge 的一個授權選項。

「效能」與「查詢」畫面

「效能」畫面使用 Build Forge 資料來建立數個標準報告。當您開啓效能報告時，會自動更新結果，因此您所查看的一律是現行資料。

「效能」畫面還有一個「查詢」畫面，內含預先定義的查詢報告。若要執行這些報告，請提供必要資料，然後按一下「快速啓動」圖示。執行查詢報告時，是依據現行資料。只需重新執行報告，即可查看以最新資料為基礎的結果。

「效能」與「查詢」畫面中的報告無法修改。依照設計，您無法修改欄位或變更報告的呈現方式。

「快速報告」工具

「快速報告」是一個以 BIRT（一種 Eclipse 型開放程式碼報告系統）為基礎的報告工具。透過「快速報告」，您可以建立自己的報告設計。您可以使用所提供的報告類型作為起始點，然後選擇您要的報告欄位。所有的報告類型皆使用 Rational Build Forge 資料庫中的資料作為其資料來源。

對於您在「快速報告」中所設計的任何報告，您均可選擇表格或圖表等數種報告格式。此外，您可以使用分組與過濾功能，來控制報告的呈現方式。

顯示報告輸出中資料的必備條件

所有的報告（包括範例報告在內）都是顯示來自 Build Forge 資料庫的資料。如果要在資料庫中移入資料，必須先建立專案並執行工作。

註：如果您是在資料庫包含資料之前，就建立和執行報告，報告結果會是空的。此外，您可能會遇到報告錯誤。

顯示報告中資料所需的存取權

執行報告之使用者的存取權，將決定報告結果中會包含以及可看見哪些專案資料。

使用者的專案存取權，將決定報告結果中所要包含的專案資料；如下所示：

- 若為「效能」畫面中的標準報告，則使用的許可權為開啓報告頁面之使用者的許可權。
- 若為「查詢」畫面中預先定義的報告，則使用的許可權為執行報告之使用者的許可權。
- 若為「快速報告」，則使用的許可權為執行報告之使用者的許可權，而非報告設計人員的許可權。

其他的存取權可讓使用者修改報告設計以及執行報告。請參閱第 396 頁的『「快速報告」的報告群組許可權』。

將內建報告的結果匯出至 CSV 檔

有數個內建報告可以將報告結果匯出成 CSV 檔。如果您想儲存結果，或者將結果匯入至另一個報告應用程式，請使用這項功能。

只要內建報告含有「儲存至 CSV」鏈結，就可以匯出報告結果。

註：若為「快速報告」，就沒有這項功能。

如果要將報告結果匯出成 CSV 檔格式，請執行下列動作：

1. 執行報告以產生結果：
 - 選取**分析報告**鏈結，即會開啓「步驟與伺服器效能」報告（**報告 > 效能**）。
 - 執行「查詢」畫面中的任何報告（**報告 > 查詢**）。
2. 從報告結果頁面中按一下**儲存至 CSV**。
3. 在「檔案下載」對話框中，按一下**開啓或儲存**：
 - 使用支援 CSV 檔格式的任何應用程式來開啓 CSV 檔案。例如，Microsoft Excel 或您喜好的文字型編輯器。
 - 將檔案儲存至您的本端工作站或網路上的另一個位置。

效能的標準報告

當您開啓標準報告頁面時，標準報告會根據現行資料來顯示結果。

「效能」畫面會使用現行專案和工作資料來建立下列數種標準報告：

- 針對所有專案建立工作效能報告
- 依專案建立工作持續時間報告
- 依專案建立步驟與伺服器分析報告

檢視專案的工作效能統計資料

「效能」畫面的主要特性是一份工作效能報告，此報告能彙總所有專案的高階工作統計資料。

- 在前次工作方面，報告會顯示完成的日期/時間以及工作的持續時間。
- 在工作總數方面，統計資料包含通過、失敗或通過但附帶警告的工作數量。

資料來源：Rational Build Forge 資料庫

報告格式：表格

顯示選項：建立過濾器以控制清單中所要顯示的專案。

欄位：

- 自信間隔：工作平均持續時間前後間隔 (+/-) 的值範圍（秒）。這個值範圍有 95% 可能包含下一個工作持續時間。例如，假設自信間隔是 5.88 秒，則不論工作平均持續時間的前後間隔，有 95% 可能包含下一個工作持續時間的值範圍為 5.88 秒。

如果要檢視工作效能統計資料報告，請執行下列動作：

1. 開啓報告標籤。
2. 按一下效能畫面。

所有專案的工作效能報告都是以表格來顯示結果。

檢視專案的工作持續時間

「效能」畫面含有一份工作持續時間報告，此報告會以圖形來顯示所有專案工作的持續時間。

資料來源：Rational Build Forge 資料庫

報告格式：圖形

如果要檢視專案的工作持續時間報告，請執行下列動作：

1. 開啓報告標籤。
2. 按一下效能畫面。
3. 按一下清單中的某個專案名稱。

該專案的工作持續時間報告會以圖形來顯示工作的時間。

依專案來檢視步驟和伺服器的效能

「效能」畫面含有一份依專案分類的步驟與伺服器分析報告。此外，也針對專案中的步驟併入一份重要路徑分析報告。

在每一個專案步驟方面，步驟與伺服器分析報告會列出執行該步驟的伺服器，以及列出步驟在不同次執行時（例如：最快或最慢一次的執行）所需的時間。步驟時間是按通過、失敗或警告來分類。種類可避免您不當比較時間，例如，通過時間和失敗時間相比較。

步驟的重要路徑分析報告列出已通過工作的一些重要測量值。

資料來源：Rational Build Forge 資料庫

報告格式：表格

索引鍵：-/- 索引鍵提供兩則步驟資訊：步驟結果與步驟持續時間（以秒為單位），如下列範例所示：

14/-/ 通過/警告/失敗	步驟通過，且執行時間總計為 14 秒。
-/-/ 通過/警告/失敗	步驟失敗，且執行時間總計為 1 秒。
-/1/ 通過/警告/失敗	步驟通過但出現警告，且執行時間總計為 1 秒。

欄位：

- 通過的步驟：步驟在所有工作中通過的次數。
- 平均執行時間：通過的工作的平均工作持續時間。
- 自信間隔：工作平均持續時間前後間隔 (+/-) 的值範圍（秒）。這個值範圍有 95% 可能包含下一個工作持續時間。例如，假設自信間隔是 5.88 秒，則不論工作平均持續時間的前後間隔，有 95% 可能包含下一個工作持續時間的值範圍為 5.88 秒。
- 最長持續時間的機率：下一項工作將等於已通過工作最長持續時間的可能性。
- 最短持續時間的機率：下一項工作將等於已通過工作最短持續時間的可能性。

如果要檢視分析報告，請執行下列動作：

1. 開啟**報告**標籤。
2. 按一下**效能**畫面。
3. 在清單中選擇專案，然後按一下**分析**。

步驟與伺服器的分析報告以及重要路徑分析報告會以表格來顯示結果。

查詢的預先定義報告

使用現行的專案與工作資料，提供一個日期範圍或其他必要資料，來執行預先定義的報告。

「查詢」畫面使用現行資料來建立數個預先定義的報告：

- 專案和步驟的選取元使用率歷程
- 所有伺服器的現行伺服器資訊清單
- 工作結果的通過/失敗/警告歷程
- 某個日期範圍中之工作的伺服器使用率歷程
- 根據 MD5 值來搜尋工作檔

檢視選取元使用率歷程

這份報告顯示所有專案的選取元使用率。


「查詢」畫面中含有一份報告，此報告會顯示所有專案的選取元使用率。利用這份報告，可查看指派給專案和步驟的選取元。如果您使用選取元，來動態選取要執行專案或步驟的伺服器，這份報告特別有用。

資料來源：Rational Build Forge 資料庫

報告格式：表格

顯示選項：選取**將報告輸出平面化**，可展開樹狀結構階層，而在清單中顯示所有的報告結果。

如果要檢視您步驟與工作的伺服器使用率報告，請執行下列動作：

1. 開啓**報告**標籤。
2. 按一下**查詢**畫面。
3. 按一下專案選取元的「快速啓動」圖示 ，然後進入伺服器報告。

選取元使用率報告會以表格來顯示結果。

依伺服器來檢視現行伺服器資訊清單

這份報告列出所有伺服器的伺服器資訊清單內容與值。

「查詢」畫面中含有一份報告，此報告會列出所有伺服器的伺服器資訊清單內容與值。這份報告可用來比較伺服器的內容、識別相同選取元可能選擇的伺服器，或識別伺服器所需的某個內容。


這份報告顯示資訊清單中的所有內容，而這些都是由指派給伺服器的收集器所指派給伺服器的內容。如果要併入自動指派給所有伺服器的特殊資訊清單內容，請選取**顯示 BF_properties**。

資料來源：Rational Build Forge 資料庫

報告格式：表格

顯示選項：選取**將報告輸出平面化**，可展開樹狀結構階層，而在清單中顯示所有的報告結果。

如果要檢視您伺服器的現行伺服器資訊清單報告，請執行下列動作：

1. 開啓**報告**標籤。
2. 按一下**查詢**畫面。
3. （選用）選取**顯示 BF_properties**，以便在報告結果中包含特殊的資訊清單內容。
4. 按一下現行伺服器資訊清單報告的「快速啓動」圖示 .

現行伺服器資訊清單報告會以表格來顯示結果。

檢視工作的通過/失敗/警告結果

這份報告列出在指定的日期/時間範圍內，所完成工作的通過/失敗/警告結果。

「查詢」畫面中含有一份報告，此報告會列出在指定的日期/時間範圍內所完成的工作的通過/失敗/警告結果。這份報告套用於所有專案的所有工作。利用這份報告，可取得已完成工作的快速性能狀態檢查。

資料來源：Rational Build Forge 資料庫

報告格式：表格

顯示選項：選取**將報告輸出平面化**，可展開樹狀結構階層，而在清單中顯示所有的報告結果。

如果要檢視您專案的工作結果報告，請執行下列動作：

1. 開啓**報告標籤**。
2. 按一下**查詢**畫面。
3. 使用預設時間範圍，或指定報告結果的時間範圍。使用 **UNIX** 時間格式來指定時間。
UNIX 時間格式是從 1970 年 1 月 1 日開始的秒數。您可以利用類似位於 http://www.onlineconversion.com/unix_time.htm 的工具，從傳統日期時間格式計算 **UNIX** 時間。

從當天的午夜 (00:00) 開始，並在隔天的午夜結束。比方說，如果要傳回在 2009 年 7 月 21 日完成的所有工作，請輸入下列 **UNIX** 時間：

從： 07/21/09 00:00 AM	1248134400
到： 07/22/09 00:00 AM	1248220800

4. 按一下工作結果報告的「快速啓動」圖示 。

工作結果報告會以表格來顯示結果。

檢視某個日期範圍中工作的伺服器使用率

這份伺服器使用率報告列出專案之所有工作的伺服器。

「查詢」畫面中含有一份伺服器使用率報告，此報告會列出專案的所有工作的伺服器。報告會針對每一個步驟，識別執行該步驟的伺服器，以及列出該步驟的持續時間。利用這份報告，可比較步驟在不同伺服器上的持續時間。

資料來源：Rational Build Forge 資料庫


報告格式：表格

顯示選項：選取**將報告輸出平面化**，可展開樹狀結構階層，而在清單中顯示所有的報告結果。

如果要檢視步驟的伺服器使用情況報告，請執行下列動作：

1. 開啓**報告標籤**。
2. 按一下**查詢**畫面。
3. 選取**開始和結束日期**。

從當天的午夜 (00:00) 開始，並在隔天的午夜結束。

4. 按一下伺服器使用率報告的「快速啓動」圖示 。

伺服器使用率報告會以表格來顯示結果。

利用工作檔的 MD5 值來搜尋工作檔

這份報告會在工作 BOM 中搜尋符合 MD5 值的檔案。MD5 雜湊值是檔案的數位指紋。

「查詢」畫面中含有一份報告，此報告會在工作 BOM 中搜尋符合 MD5 值的檔案，若有找到相符項，就會列出工作標籤和檔案位置。如果您需要搜尋檔案的特定版本，而且知道其 MD5 值，請使用這份報告。

不過先決條件是您的專案必須含有 .scan baseline 或 .scan checkpoint 指令，才能儲存 MD5 值。如需詳細資料，請參閱 第 362 頁的『.scan』中的參照資訊。.scan 指令會完成下列作業：


- 在步驟的工作目錄中建立一份檔案清單
- 針對工作目錄中的檔案產生 MD5 值，並將 MD5 值儲存在工作 BOM 中

資料來源：Rational Build Forge 資料庫

報告格式：表格

顯示選項：選取**將報告輸出平面化**，可展開樹狀結構階層，而在清單中顯示所有的報告結果。

如果要在工作 BOM 中搜尋 MD5 值，請執行下列動作：

1. 開啓**報告標籤**。
2. 按一下**查詢**畫面。
3. 在 **MD5 搜尋值**中，輸入您要尋找的 MD5 值。
4. 按一下 MD5 檔案報告的「快速啓動」圖示 。

MD5 檔案報告會以表格來顯示結果。

使用「快速報告」來建立報告

使用「快速報告」提供的報告類型，來建立您自己的報告設計。「快速報告」是 Rational® Build Forge® 的一個授權選項。

在「快速報告」中，您可以建立、儲存、編輯和執行報告設計。您可以將報告儲存至公用目錄，以供其他使用者共用。

「快速報告」中的報告類型會使用 Build Forge® 資料庫中的資料來建立報告。不同的報告類型包括不同的報告欄位。欄位名稱會說明其功能，例如：「容量」與「建置」。

建立報告時，您可以從數種報告格式（例如表格或圖表）中選擇。使用分組與排序功能來控制報告的呈現方式。

您可以選擇性地執行和檢視工作 BOM 檔中的報告結果。請參閱 第 358 頁的『.report』中的參照資訊。

如果要著手執行，請按一下**報告標籤**，以顯示用來產生報告的選項，然後選取**快速報告**。

註：8.0 的「快速報告」只會顯示 Perl 引擎所產生的報告。

「快速報告」的報告群組許可權

您可將報告儲存在公用或專用儲存庫中。這可讓您控制可執行、修改、儲存和刪除在「快速報告」中建立之報告的人員。

專用報告只可供建立與儲存報告的 Rational Build Forge 使用者存取。

只要使用者隸屬於具備適當許可權的存取群組，就可以存取公用報告。依預設，「建置工程師」群組中的使用者對於您已儲存成公用報告的報告，具備讀取、寫入、編輯或所有這三種存取權。

若要授與存取權，您可以將下列一個以上報告許可權指派給使用者存取群組：

讀取公用報告	此許可權容許執行「快速報告」的公用報告儲存庫中，所儲存的報告。
儲存、編輯或刪除報告	此許可權容許儲存、編輯或刪除「快速報告」的公用儲存庫中所儲存的報告。

下表列出「報告」許可權群組的許可權，並識別指派給存取群組的預設許可權。

若要在「主控台」使用者介面中管理許可權，請選取**管理 > 許可權**。

	建置工程師	開發人員	訪客	操作員	安全人員	系統管理員
讀取公用報告	是	是			是	是
儲存公用報告	是					
編輯公用報告	是					
刪除公用報告	是					

「快速報告」的報告類型參照

報告類型具有功能性的名稱，以說明其內容（例如，容量與建置）。這個主題會針對每一種報告類型，說明報告用途、報告欄位以及報告範例。

分析 **說明：**使用「分析」報告類型，可利用步驟持續時間與步驟執行順序，來報告步驟層次的工作效能。

報告範例：

- 建立一份表格報告，以顯示步驟持續時間，以及工作所含之每一個步驟的步驟順序。工作依專案名稱來分組。
- 建立一份折線圖報告，以顯示每一個建置標籤的步驟持續時間。建置標籤依步驟名稱來分組。

欄位說明：

欄位名稱	說明
建置標籤	工作標籤是一個以專案標籤格式為基礎的唯一 ID。
專案名稱	使用者指派的專案名稱。
步驟持續時間	步驟及其列入的任何步驟的執行時間總計（秒）。

欄位名稱	說明
步驟名稱	使用者指派的步驟名稱。
步驟順序	用來識別步驟執行順序的號碼。

建置 說明：使用建置報告類型，可報告專案層次的工作效能。

報告範例：

- 建立一份表格報告，以顯示每一個建置標籤的建置結果、開始時間以及持續時間。建置標籤依專案來分組。
- 建立一份長條圖報告，以顯示每一個專案的建置計數。依專案來分組。

欄位說明：

欄位名稱	說明
建置計數（聚集欄位）	工作數總計，包含已完成和失敗的工作。
建置持續時間	工作的執行時間總計（秒）。
建置結果	工作結果：通過、通過但附帶警告，或失敗。
建置開始時間	工作的開始時間。
建置狀態	建置狀態：執行中、完成、已保存或已鎖定。
建置標籤	工作標籤是一個以專案標籤格式為基礎的唯一 ID。
專案名稱	使用者指派的專案名稱。
選取元名稱	使用者指派的選取元名稱。
使用者登入	啟動工作之使用者的 Build Forge 登入或使用者名稱。
使用者名稱	啟動工作的使用者名稱。

容量 說明：使用「容量」報告類型，可依專案來報告工作效能。

報告範例：

- 建立一份表格報告，以顯示每一個建置標籤的建置開始時間、建置持續時間、建置平均持續時間，以及建置結果。建置標籤依專案來分組。
- 建立一份圖表報告，以顯示每一個專案的平均建置時間。專案依選取元來分組。

欄位說明：

欄位名稱	說明
建置平均持續時間（聚集欄位）	以工作數總計（包含已完成和失敗兩者）為基礎的工作平均執行時間。
建置持續時間	工作的執行時間總計（秒）。
建置結果	工作結果：通過、通過但附帶警告，或失敗。
建置開始時間	工作的開始時間。
建置標籤	工作標籤是一個以專案標籤格式為基礎的唯一 ID。
前次建置的持續時間（聚集欄位）	前次工作的執行時間。完成前次工作的時間總計（秒）。
專案名稱	使用者指派的專案名稱。
選取元名稱	使用者指派的選取元名稱。

專案 說明：使用「專案」報告，可依專案來報告伺服器、環境與步驟使用情形，以及依專案來報告步驟效能。

報告範例：

- 建立一份表格報告，以顯示專案、類別名稱與專案環境。專案依伺服器來分組。
- 建立一份表格報告，以顯示步驟、步驟結果，步驟順序、步驟環境以及伺服器。步驟依專案來分組，並依步驟順序來排序。
- 建立一份長條圖報告，以顯示每一個步驟的步驟計數。步驟依步驟結果來分組。

欄位說明：

欄位名稱	說明
類別名稱	使用者指派給專案的類別，例如：正式作業或測試。
失敗的步驟計數（聚集欄位）	您所選群組欄位（例如：專案、伺服器或其他欄位名稱）的失敗步驟數。
通過的步驟計數（聚集欄位）	您所選群組欄位（例如：專案、伺服器或其他欄位名稱）的通過步驟數。
專案環境名稱	用來定義專案環境變數的專案環境名稱。
專案層次	專案的使用者指派存取群組。
專案名稱	使用者指派的專案名稱。
伺服器名稱	使用者指派的伺服器名稱。
步驟計數（聚集欄位）	您所選群組欄位（例如：專案或伺服器）的步驟數總計。
步驟環境名稱	用來定義步驟環境變數的步驟環境名稱。
步驟層次	步驟的使用者指派存取群組。
步驟名稱	使用者指派的步驟名稱。
步驟結果	步驟結果：通過、通過但附帶警告，或失敗。
步驟順序	用來識別步驟執行順序的號碼。

步驟測量值

說明：使用「步驟測量值」報告，可依專案來報告步驟的成功和失敗統計資料。

報告範例：

- 建立一份表格報告，以顯示步驟名稱、步驟計數，以及通過和失敗的步驟百分比。步驟依專案來分組。
- 建立一份折線圖報告，並依建置標籤來顯示步驟的持續時間。建置標籤依步驟名稱來分組。

欄位說明：

欄位名稱	說明
建置標籤	工作標籤是一個以專案標籤格式為基礎的唯一 ID。
步驟失敗百分比（聚集欄位）	失敗的步驟數佔步驟數總計的百分比。步驟數總計是針對您選取的群組欄位，例如：專案。

欄位名稱	說明
步驟通過百分比（聚集欄位）	通過的步驟數佔步驟數總計的百分比。步驟數總計是針對您選取的群組欄位，例如：專案。
專案名稱	使用者指派的專案名稱。
伺服器名稱	使用者指派的伺服器名稱。
步驟平均持續時間（聚集欄位）	以步驟數總計（包含已完成和失敗兩者）為基礎的步驟平均執行時間。
步驟計數（聚集欄位）	步驟數總計，包含已完成和失敗的步驟。步驟數總計是針對您選取的群組欄位，例如：專案。
步驟持續時間	步驟及其列入的任何步驟的執行時間總計（秒）。
步驟名稱	使用者指派的步驟名稱。
步驟結果	步驟結果：通過、通過但附帶警告，或失敗。
步驟順序	用來識別步驟執行順序的號碼。
步驟開始時間	步驟的開始時間。

品質 **說明：**使用「品質」報告，可依專案來報告工作的成功和失敗統計資料。

報告範例：

- 建立一份表格報告，並依建置標籤來顯示建置結果。建置標籤依專案來分組。
- 建立一份表格報告，並依專案來顯示通過和失敗的所有建置的百分比。依專案來分組。
- 建立一份長條圖報告，並依專案來顯示建置計數。專案依建置結果來分組。

欄位說明：

欄位名稱	說明
建置計數（聚集欄位）	工作數總計，包含已完成和失敗的工作。
建置結果	工作的結果或狀態：通過、通過但附帶警告，或失敗。
建置開始時間	工作的開始時間。
建置標籤	工作標籤是一個以專案標籤格式為基礎的唯一 ID。
建置失敗百分比（聚集欄位）	失敗的建置佔建置數總計的百分比。建置數總計是針對您選取的群組欄位，例如：專案。
建置通過百分比（聚集欄位）	通過的建置數佔建置數總計的百分比。建置數總計是針對您選取的群組欄位，例如：專案。
專案名稱	使用者指派的專案名稱。

資源 **說明：**使用「資源」報告，可依專案和伺服器，來報告步驟與工作的效能。

報告範例：

- 建立一份表格報告，並依伺服器來顯示步驟的執行時間。選取步驟順序、步驟名稱、伺服器名稱、步驟開始時間，以及步驟持續時間。依步驟順序和開始時間來排序。

- 建立一份表格報告，並依伺服器來顯示工作的執行時間。選取建置標籤、伺服器名稱、建置開始時間、建置持續時間，以及建置結果。建置標籤依專案來分組，並依建置開始時間來排序。

欄位說明：

欄位名稱	說明
建置持續時間	工作的執行時間總計（秒）。
建置結果	工作結果：通過、通過但附帶警告，或失敗。
建置開始時間	工作的開始時間。
建置標籤	工作標籤是一個以專案標籤格式為基礎的唯一 ID。
專案名稱	使用者指派的專案名稱。
選取元名稱	使用者指派的選取元名稱。
伺服器名稱	使用者指派的伺服器名稱。
步驟持續時間	步驟及其列入的任何步驟的執行時間總計（秒）。
步驟順序	用來識別步驟執行順序的號碼。
步驟開始時間	步驟的開始時間。

BOM 說明：

使用 BOM 報告，可利用工作 BOM 中的資訊來建立一份報告。BOM 報告可針對下列任何資料集來執行：

- 資料庫中的所有專案（預設值）
- 單一專案
- 單一專案中的一或多項建置

所能選取的報告欄位，因資料集而異。報告欄位可能包括步驟資訊清單內容、**.scan** 指令的輸出、配接器記載的輸出，以及使用者定義的直欄（使用 **.bom** 指令指定）。

報告範例：

- 建立一份表格報告，以顯示涵蓋多個專案的 **.scan** 指令輸出。選取建置標籤、BOM 資料、BOM 路徑和 BOM 類型等欄位。建置標籤依專案來分組。
- 建立一份表格報告，以顯示過濾器動作在多個專案中的呼叫次數。選取步驟名稱、過濾器事件計數類型，以及過濾器計數。步驟依專案來分組。

欄位說明：

欄位名稱	說明
BOM 資料	如果您在專案中併入 .scan checkpoint 指令，就已掃描的檔案來說，「BOM 資料」欄位會顯示 MD5 值。
BOM 路徑	如果您在專案中併入 .scan checkpoint 指令，就已掃描的檔案來說，「BOM 路徑」欄位會顯示檔案路徑。
BOM 類型	如果您在專案中併入 .scan checkpoint 指令，就已掃描的檔案來說，「BOM 類型」欄位會指出路徑是 D（目錄）、F（檔案）還是 S（符號鏈結）。
建置標籤	工作標籤是一個以專案標籤格式為基礎的唯一 ID。
過濾事件計數	爲了在步驟輸出中尋找型樣相符項，過濾器已呼叫的過濾器動作次數。

欄位名稱	說明
過濾事件計數類型	當在步驟輸出中找到型樣相符項時，過濾器所呼叫的過濾器動作。
專案名稱	使用者指派的專案名稱。
結果說明	結果記錄中的說明欄位，通常含有一份步驟名稱副本。
步驟名稱	使用者指派的步驟名稱。
選取元名稱	使用者指派的選取元名稱。
伺服器名稱	使用者指派的伺服器名稱。
步驟持續時間	步驟及其列入的任何步驟的執行時間總計（秒）。
步驟結果	步驟結果：通過、通過但附帶警告，或失敗。

「快速報告」的報告格式與呈現參照

「快速報告」提供表格、長條圖、折線圖及圓餅圖等常用報告格式。

註：「快速報告」隨附的範例報告不包含任何資料。如果要查看報告資料，您必須先在「管理主控台」中建立專案並執行工作。

「表格」報告格式

範例

如果要查看範例表格報告，請在「快速報告」中選取 **SampleAnalytic-StepDuration** 報告。

若要查看呈現選項，請針對 **SampleAnalytic-StepDuration** 報告，選取編輯圖示



需求

在表格方面，請觀察以下的需求：

- 至少選取一個報告欄位。
- 如果您選取一個聚集報告欄位，就必須選取一個分組欄位。請參閱第 405 頁的『報告格式和聚集報告欄位的選項需求』。

報告欄位

在「報告」欄位中，選取報告的一個以上表格直欄。

註：選項清單中的欄位順序很重要。

第一個欄位會成為第一個表格直欄，最後一個欄位會成為最後一個表格直欄。第一個表格直欄也會用來分組報告結果。

若要試驗報告欄位，請針對 **SampleAnalytic-StepDuration** 報告，選取編輯圖示



，然後執行下列動作：

1. 在選項清單中，變更欄位順序。
2. 儲存您的選擇。
3. 執行報告，以查看表格直欄順序中的變更。

群組欄位



您可以選擇從「群組」欄位中選取欄位。您選取的欄位會當成樹狀控制項，新增至第一個表格直欄中，以便用來將報告結果分組。您可以選取多個群組欄位，也可以選取一個不會顯示成表格直欄的群組欄位。

若要試驗分組，請針對 **SampleAnalytic-StepDuration** 報告，選取編輯圖示，然後執行下列動作：

1. 選取「專案名稱」作為群組欄位。
2. 儲存您的選擇。
3. 執行報告。

因此，「專案名稱」的樹狀節點會新增至「建置標籤」直欄中。

排序欄位

您可以選擇在「排序」欄位中選取含有可排序的資料的表格直欄。使用箭頭來指定排序方向。向上箭頭 會以遞增順序最低到最高 來排序資料，向下箭頭 會以遞減順序最高到最低 來排序資料。

若要試驗排序，請針對 **SampleAnalytic-StepDuration** 報告，選取編輯圖示，然後執行下列動作：

1. 選取「專案名稱」作為排序欄位，並使用箭頭來指定排序方向。
2. 儲存您的選擇。
3. 執行報告。

「長條圖」報告格式

範例

若要在「快速報告」中查看範例長條圖，請選取 **SampleCapacity-RunTimeByProject** 報告。

若要查看呈現選項，請針對 **SampleCapacity-RunTimeByProject** 報告，選取編輯圖示。

垂直圖表元素（Y 軸）

針對一組資料元素（X 軸），選取一個欄位，以作為您要比較的資料值（Y 軸）。圖表的垂直軸或 Y 軸上會顯示欄位名稱和資料值單位。

註：只能選取一個欄位作為垂直圖表元素。

水平圖表元素（X 軸）

選取一個欄位，作為您要使用資料值（Y 軸）來比較的資料元素集（X 軸）。欄位名稱會顯示成 X 軸標籤。

資料值會顯示在長條直欄上，相等於長條直欄的高度。

註：只能選取一個欄位作為水平圖表元素。

群組欄位

（選擇性）您可以選取一個群組欄位，以便將圖表水平軸或 X 軸上的資料元素分組。

需求

在圖表方面，請觀察以下的需求：

- 同時選取 X 軸和 Y 軸報告欄位。
- 為 X 軸和 Y 軸選取不同的報告欄位。
- 為 X 軸資料選取一個分組欄位。
- 如果您選取一個聚集報告欄位，就必須選取一個分組欄位。請參閱第 405 頁的『報告格式和聚集報告欄位的選項需求』。

範例：

如果要試驗一下長條圖，請嘗試建立下列報告：

- 「步驟」報告類型：針對每一個「步驟名稱」X 軸，比較「步驟失敗百分比」Y 軸，並依「建置標籤」來將結果分組。
- 「容量」報告類型：針對每一個「專案」X 軸，比較「前次建置的持續時間」Y 軸，並依「專案名稱」來將結果分組。
- 「資源」報告類型：針對每一個「建置標籤」X 軸，比較「建置持續時間」Y 軸，並依「專案名稱」來將「建置標籤」分組。

「折線圖」報告格式


說明

折線圖顯示資料隨時間或一連串事件後的發展情況。預設分組欄位是 X 軸報告欄位。如果您選取多個分組欄位，則會在報告中顯示多條折線。

範例

若要查看範例折線圖，請在「快速報告」中選取 **SampleResource-DurationOverTime** 報告。

若要查看呈現選項，請針對 SampleResource-DurationOverTime 報告，選取編輯

圖示 。

垂直圖表元素 Y 軸

針對一組資料元素 X 軸，選取一個欄位，以作為您要比較的資料值 Y 軸。Y 軸圖表上會顯示欄位名稱和資料值單位。

註：只能選取一個欄位作為垂直圖表元素。

水平圖表元素 X 軸

選取一個欄位，作為您要使用資料值 Y 軸來比較的資料元素集 X 軸。欄位名稱會顯示成 X 軸圖表的標籤。

X 軸資料元素的資料值會顯示在點的旁邊，相等於它們在 Y 軸上的值。一條連續線條會連接各點，而形成折線圖。

註：只能選取一個欄位作為水平圖表元素。

群組欄位

（選擇性）您可以選取一個群組欄位，以便將水平軸或 X 軸圖表上的資料元素分組。預設分組欄位是 X 軸報告欄位。如果您選取多個分組欄位，則會在報告中顯示多條折線。

需求

在圖表方面，請觀察以下的準則：

- 同時選取 X 軸和 Y 軸報告欄位。
- 為 X 軸和 Y 軸選取不同的報告欄位。
- 為 X 軸資料選取一個分組欄位。
- 如果您選取一個聚集報告欄位，就必須選取一個分組欄位。請參閱第 405 頁的『報告格式和聚集報告欄位的選項需求』。

範例

如果要試驗一下折線圖，請嘗試建立下列報告：

- 「步驟」報告類型：針對每一個「建置標籤」X 軸，顯示「步驟持續時間」Y 軸，並依「步驟名稱」來將結果分組。
- 「建置」報告：針對每一個「建置開始時間」X 軸，顯示「持續時間」Y 軸，並依「專案名稱」來將結果分組。
- 「容量」報告類型：針對每一個「建置標籤」X 軸，顯示「建置持續時間」Y 軸，並依「伺服器名稱」來將結果分組。

「圓餅圖」報告格式

Y 數列元素

選取一個欄位，作為圓餅圖的資料值。楔形大小代表資料值。圓餅圖上顯示欄位名稱和資料值單位。

註：只能選取一個欄位作為 Y 數列元素。

X 數列元素

選取一個欄位，作為您要使用資料值 Y 數列 來求值的資料元素集 X 數列。楔形的數目代表資料元素的數目。

註：只能選取一個欄位作為 X 數列元素。

群組欄位

圓餅圖需要群組。X 數列元素會作為預設群組。選取不同的群組，會為資料值 Y 數列 產生另一個圓餅圖。

需求

在圖表方面，請觀察以下的準則：

- 同時選取 X 數列和 Y 數列報告欄位。
- 為 X 數列和 Y 數列選取不同的報告欄位。
- 為 X 軸資料選取一個分組欄位。
- 如果您選取一個聚集報告欄位，就必須選取一個分組欄位。請參閱第 405 頁的『報告格式和聚集報告欄位的選項需求』。

範例

如果要試驗一下圓餅圖，請嘗試建立下列報告：

- 「專案」報告類型：選取「失敗的步驟計數」Y 數列 和「步驟名稱」X 數列。
- 「容量」報告類型：選取「建置平均持續時間」Y 數列 和「專案名稱」X 數列。

- 「分析」報告類型：選取「步驟持續時間」Y 數列 和「步驟名稱」X 數列，並依「步驟名稱」來分組。

報告格式和聚集報告欄位的選項需求

為了產生有意義的報告，請觀察表格與圖表以及聚集報告欄位的一些基本需求。

聚集報告欄位

需求

如果您要在表格或圖表中包含聚集報告欄位，則必須包含一個分組欄位。

定義

聚集報告欄位所含的資料，是衍生自 Build Forge 資料庫中的一或多個原始資料欄位。聚集報告欄位的資料不會儲存在資料庫中。

下列的報告欄位為聚集欄位。

分析：前次建置的持續時間	專案：失敗的步驟計數	專案：通過的步驟計數
專案：步驟計數	步驟測量值：步驟失敗百分比	步驟測量值：步驟通過百分比
步驟測量值：步驟平均持續時間	步驟測量值：步驟計數	品質：建置計數
品質：建置失敗百分比	品質：建置通過百分比	建置：建置計數

「表格」報告格式

需求

在表格方面，請觀察以下的需求：

- 至少選取一個報告欄位。
- 如果您選取一個聚集報告欄位，就必須選取一個群組欄位。

「圖表」報告格式

需求

若為長條圖、折線圖與圓餅圖，請觀察以下的需求：

- 同時選取 X 數列和 Y 數列報告欄位。
- 為 X 數列和 Y 數列選取不同的報告欄位。
- 為 X 軸資料選取一個分組欄位。
- 如果您選取一個聚集報告欄位，就必須選取一個群組欄位。

範例報告參照

範例報告是指您可以使用系統提供的報告類型來建立的報告範例。

註：範例報告使用 Rational Build Forge 資料庫中的資料來建立報告輸出。只有在您已建立專案並執行工作時，才會顯示結果。

如果要顯示範例報告清單，請按一下**快速報告**。

若要查看某個範例報告的欄位與格式選項，請選取該報告的**編輯**圖示 。

若要複製範例報告，請選取報告的**編輯**圖示 ，然後按一下**複製報告**。

如果要執行範例報告，請選取該報告的名稱。

下表提供範例報告定義。

範例報告名稱	說明
SampleAnalytic-StepDuration	<p>說明：建立一份表格，以報告建置的步驟詳細資料</p> <p>報告類型：分析</p> <p>報告格式：表格</p> <p>報告欄位：建置標籤、專案名稱、步驟名稱、步驟持續時間</p> <p>報告選項：沒有分組或排序選項</p>
SampleBuild-BuildsByState	<p>說明：建立一份表格，並依專案來列出建置與建置狀態</p> <p>報告類型：建置</p> <p>報告格式：表格</p> <p>報告欄位：建置標籤、專案名稱、建置計數、建置狀態</p> <p>報告選項：依專案名稱分組</p>
SampleCapacity- RuntimeByProject	<p>說明：建立長條圖，並依專案顯示工作的平均建置持續時間</p> <p>報告類型：容量</p> <p>報告格式：長條圖</p> <p>報告欄位：建置平均持續時間（Y 軸）與專案名稱（X 軸）</p> <p>報告選項：依專案名稱分組</p>
SampleProject-TotalsByProject	<p>說明：建立一份表格，並依專案報告通過與失敗的步驟數</p> <p>報告類型：專案</p> <p>報告格式：表格</p> <p>報告欄位：專案名稱、通過的步驟計數、失敗的步驟計數、步驟計數</p> <p>報告選項：依專案名稱分組</p>
SampleQuality-PercentSuccess	<p>說明：建立一份表格，並依專案報告通過與失敗工作的百分比</p> <p>報告類型：品質</p> <p>報告格式：表格</p> <p>報告欄位：專案名稱、建置計數、建置失敗百分比、建置通過百分比</p> <p>報告選項：依專案名稱分組</p>

範例報告名稱	說明
SampleResource-DurationOverTime	<p>說明：建立一份圖表，以顯示每一項建置的建置開始時間與建置持續時間</p> <p>報告類型：資源</p> <p>報告格式：表格</p> <p>報告欄位：建置持續時間（Y 軸）與建置開始時間（X 軸）</p> <p>報告選項：遞增排序順序</p>

使用提供的報告類型來建立報告

如果要建立自己的報告，只需選取您要使用的報告類型、報告格式，以及報告欄位即可。

程序

1. 選取**報告標籤**，然後選取**快速報告**。
2. 在**報告名稱**中，輸入一個唯一名稱。報告名稱為必要的。報告名稱用來將報告儲存在資料庫中，並且必須是唯一的。
3. 在**報告標題**中，輸入報告的敘述性標題。報告標題為必要的。標題會顯示在報告的頂端。
4. 在**可見性**中，選取「公用」或「專用」。「專用」報告無法共用。「公用」報告可讓具備必要存取權的使用者共用。如需詳細資料，請參閱第 396 頁的『「快速報告」的報告群組許可權』。
5. 在**報告格式**中，選取一個表格或下列一種圖表格式：長條圖、折線圖或圓餅圖。如需報告格式的詳細資料，請參閱第 401 頁的『「快速報告」的報告格式與呈現參照』。
6. 在**報告類型**中，選擇一種報告類型。報告類型會決定您報告中所要併入的內容。如需詳細資料和範例，請參閱第 396 頁的『「快速報告」的報告類型參照』。
 - 選取要顯示在您的表格或圖表報告中的報告欄位。
 - 選取分組與排序選項，以控制報告的呈現。

重要：如果您選取 **BOM** 報告類型，則會顯示「專案」方框。如果要針對特定專案建立 **BOM** 報告，請先選取專案，以顯示完整的報告欄位清單，然後選取要包含在報告中的欄位。

7. （僅限 **BOM** 報告類型）在**專案**中，選取下列其中一個選項：
 - 選取「全部專案」，則只會顯示標準 **BOM** 報告欄位。（「全部專案」是預設值。）

報告輸出含有資料庫中所有專案的 **BOM** 資訊。

- 選取一個專案，除了標準 **BOM** 報告欄位之外，還會顯示其專案特有的 **BOM** 欄位。

報告輸出只含有單一專案的 **BOM** 資訊。

8. 按一下**儲存報告**，以儲存您的選擇。報告會顯示在報告清單中。

新增報告輸出至工作 BOM

您可以選擇性地使用 `.report` 指令，將報告結果新增至工作 BOM。

程序

1. 使用「快速報告」工具來建立報告。請參閱第 407 頁的『使用提供的報告類型來建立報告』。

註：BOM 報告類型或專用報告不支援這項特性。

2. 使用 `.report` 指令，將報告新增至工作 BOM。請參閱 第 358 頁的『`.report`』中的參照資訊。

在「快速報告」中修改與管理報告

在「快速報告」中，您可以執行報告、檢視報告結果，以及編輯您的報告設計。

報告可供編輯，以變更報告的設計。過濾器可以作為報告定義的一部分，並且可新增至任何報告中，藉以過濾輸出。

執行報告

如果要執行您在「快速報告」中所建立的任何報告，請按一下該報告名稱。

關於這項作業

報告結果會顯示在「快速報告」的結果視圖中。

結果

如果要回到報告清單，請按一下您 Web 瀏覽器的「上一頁」箭頭。

編輯報告

如果要編輯您在「快速報告」中所建立的任何報告，請按一下該報告名稱旁的**編輯**圖示。

關於這項作業

報告選項會顯示在「報告」詳細資料中。在您完成變更後，請按一下**儲存報告**，以儲存您的變更。

複製報告

當複製範例報告或其他報告時，會將其報告欄位和格式化選項複製到新的報告設計中，並指派一個唯一名稱給該報告。

開始之前

若要建立唯一的報告名稱，您可以在報告名稱中加上 `Copy`，例如：`<Report_Name> Copy`。

如果您要為報告建立多個副本，請使用下列語法來附加數字：`<Report_Name> Copy <Copy_Number>`。例如 `<Report_Name> Copy 2`。

程序

1. 針對所要複製的報告，按一下**編輯**圖示。
2. 按一下**複製報告**。

所複製的報告會顯示在報告清單中。

建立報告輸出的過濾器

您可以建立報告過濾器，以控制報告輸出中所要顯示的資訊。

開始之前

在建立過濾器之前，請先檢閱下列的需求和限制：

- 先建立報告；過濾功能會使用報告定義來提供過濾選項。
- 報告過濾器會儲存成報告定義的一部分，並且只套用至單一報告。
- 在建立報告過濾器後，每當執行該報告時就會套用它，直到您變更或刪除該過濾器為止。
- 若為 BOM 報告類型，您可以依專案和依建置來過濾，並針對特定的報告欄位指定過濾準則。
- 若為 BOM 以外的報告類型，則必須針對特定的報告欄位指定過濾準則。

程序

1. 在報告清單中選取一份報告，並選取報告的**編輯**圖示。
2. （只有 BOM 報告類型才需要）在**專案**中，選取下列一個選項：
 - 選取「全部專案」，則只會顯示標準 BOM 報告欄位。（「全部專案」是預設值。）

報告輸出含有資料庫中所有專案的 BOM 資訊。

- 選取一個專案，除了標準 BOM 報告欄位之外，還會顯示其專案特有的 BOM 欄位。

報告輸出只含有單一專案的 BOM 資訊。

3. 選取**過濾器**標籤。
4. （僅限 BOM 報告類型選用）按一下**顯示建置過濾器**，然後選取下列選項，以指定報告中所要併入的建置資訊：
 - 如果要包含所有現行與未來建置的相關資料，則不選取任何一項。
 - 如果只想包含所有現行建置的相關資料，請按一下**僅所有現行的建置**。
 - 如果只想包含所選建置的所有現行與未來建置的相關資料，請在清單中選取建置標籤。
5. （其他所有報告類型都需要）按一下**新增過濾器**，以便從您在**報告詳細資料**標籤中選取的報告類型，選取一個報告欄位來作為過濾器。

如果要建立過濾表示式，請執行下列動作：

- a. 在**過濾欄位**中，選取要用來過濾報告資料的報告欄位。
 - b. 在**過濾運算子**中，選取關係運算子。
 - c. 在**過濾值**中，輸入報告欄位值。
6. 按一下**儲存報告**，以儲存您的報告過濾器選擇。

疑難排解「快速報告」的一般問題

在您使用 Rational Software Analyzer 時，如果遇到問題，請檢閱這個主題中的資訊，以查看是否有令人滿意的暫行或徹底解決方法。

埠衝突 「快速報告」使用您在安裝期間指定的應用程式 Web 伺服器來顯示報告。如果您遇到埠衝突問題，必須為應用程式 Web 伺服器配置一個尚未指派的埠。

第 23 章 使用公用程式

這個主題說明如何設定和執行 Rational Build Forge 提供的指令行公用程式。

存取和執行指令行公用程式

指令行公用程式位於 Build Forge 安裝目錄，此目錄在 Windows 上是 `<bfinstall>`，而在 UNIX 及 Linux 上則是 `<bfinstall>/Platform`。

您的環境（或系統層次的環境）必須配置適當，引擎層次的 Shell 指令才能運作。例如，在 Oracle 及 UNIX 上，必須手動設定 `ORACLE_HOME`、`TNS_ADMIN` 及 `LD_LIBRARY_PATH`，否則公用程式將不會執行。

在您使用 `bfexport` 或 `bfimport` 等指令行公用程式時，指令必須能夠找到 `buildforge.conf` 檔以存取資料庫，因此您必須從安裝目錄來執行指令，或者將 `BF_CONFIG_FILE` 環境變數設為 `buildforge.conf` 檔的完整路徑。

指派隱藏移轉

您可以使用 `bfassignhiddenmigration` 參照指令，以移轉一批指派隱藏移轉值來進行加密。

如果您已在 Build Forge 7.x 或更早版本中建立指派隱藏移轉變數，則可以在 Build Forge 使用者介面「環境」頁面上編輯並重新儲存來加密這些變數。若要以批次模式來加密，請使用 `bfassignhiddenmigration` 公用程式。

bfassignhiddenmigration 參照

使用這個公用程式，可以批次模式加密先前的指派隱藏移轉值。

語法

```
bfassignhiddenmigration  
connection-options command-options
```

限制與考量

您可以執行下列檔案：

- Windows 系統：`bfassignhiddenmigration.bat`
- UNIX 及 Linux 系統：`bfassignhiddenmigration.sh`

需要有「編輯隱藏標籤」許可權才能進行此移轉。您有下列連線選項：

- `-h` | `--help`

顯示說明文字。

- `-S` | `--secure C /path/to/bfclient.conf` | `--config=/path/to/bfclient.conf`

此文字指定用戶端配置檔的位置。預設值為現行目錄中的 `bfclient.conf`。

- `-H hostname` | `--hostname=hostname`

連線的遠端主機名稱。預設值：*localhost*。

- `-P port | --port=port`

連線的通訊埠。預設值：*3966*。

- `-d domain | --domain=domain`

鑑別的網域或領域。預設值：無。

- `-u login | --user=login`

鑑別使用者名稱。預設值：*root*。

- `-p password | --password=password`

鑑別使用者密碼。預設值：*root*。

- `-E encoding | --encoding=encoding`

要用於輸出字集的編碼。如果指定無效的編碼，則會列出所有可用的編碼，且程式會結束。

註：以所要求的編碼來顯示文字的能力，視用戶端上所使用的終端機軟體功能而定。預設值會視您的作業系統和 **JVM** 支援而定。

指令選項：

- `-a | --all`

移轉系統中的所有指派隱藏環境變數。如果容許此選項，則會忽略 `'-e'` 及 `'-v'` 選項。

- `-e Environment1[,Environment2[,...]] | --environment=Environment1[,Environment2[,...]]`

移轉特定環境中的指派隱藏變數。如果未提供 `'-a'` 選項，則會移轉特定環境中的所有指派隱藏變數。您可以使用 Java 正規表示式來比對您的特定環境。

- `-v Variable1[,Variable2[,...]] | --variable=Variable1[,Variable2[,...]]`

移轉特定的指派隱藏環境變數。如果未提供 `'-a'` 選項，則會移轉系統中的所有特定指派隱藏變數。您可以使用 Java 正規表示式來比對您的特定變數。

範例

1. 此範例會移轉系統中的所有指派隱藏環境變數。

Windows

```
C:\> bfassignhiddenmigration.bat -H localhost -u root -p root -a
```

UNIX 和 Linux

```
$bfassignhiddenmigration.sh -H localhost -u root -p root -a
```

2. 此範例會移轉 *MyEnv* 環境的所有指派隱藏變數。

Windows

```
$ C:\> bfassignhiddenmigration.bat -H localhost -u root -p root -e MyEnv
```

UNIX 及 Linux

```
$ bfassignhiddenmigration.sh -H localhost -u root -p root -e MyEnv
```

3. 此範例會移轉開頭是 *bf8* 的所有指派隱藏變數。

Windows

```
$C:\> bfassignhiddenmigration.bat -H localhost -u root -p root -v bf8.*
```

UNIX 和 Linux

```
$ bfassignhiddenmigration.sh -H localhost -u root -p root -v bf8.*
```

4. 此範例會移轉包含 *bf8* 或結尾是 *MyEnv* 及 *Com* 隱藏環境的所有指派隱藏變數。

Windows

```
$C:\> bfassignhiddenmigration.bat -H localhost -u root -p root -e MyEnv,Com.* -v .*bf8.*,.*hid
```

UNIX 及 Linux

```
$ bfassignhiddenmigration.sh -H localhost -u root -p root -e MyEnv,Com.* -v .*bf8.*,.*hid
```

匯出專案

您可以使用 `bfexport` 指令或 `.export` 點指令，將專案和其他 Build Forge 物件匯出至 XML 檔。

匯出的專案會儲存在 XML 檔中。匯出的專案可以再匯入 Build Forge 中。

範例：您可以在專案結尾新增一個要執行 `bfexport` 的步驟，以儲存專案配置資料。您可以使用它作為專案定義的備份。此外，也可以用它來將專案移動至另一個 Build Forge 安裝架構中。

bfexport 參照

使用 `bfexport` 指令，可將專案資料匯出至具名的 XML 檔，或將專案資料傳送至顯示終端機（標準輸出）以供檢視。匯出檔含有單一專案或專案 Snapshot 的專案配置資料。

語法

`bfexport`

```
bfexport [-l]
bfexport [-l] <project_name>

bfexport [-c "<comment>"] [-f <file_name>] [-g] [-s] [-C] [-L ] [-n ]
<project_name> | <project_name> <snapshot_name> | <project_id>
```

用法

如果要完成一般的專案匯出作業，請使用下列的指令選項：

- 如果要顯示指令語法，請直接使用 `bfexport`（不含任何選項）。
- 如果要列出儲存在 Build Forge 資料庫中的專案名稱和專案 ID，請使用 `bfexport -l`。

在指令輸出中，專案名稱後面會附加 Snapshot 名稱：`<project_ID>: <project_name> - <snapshot_name>`

- 若要將專案資料傳送至 XML 檔，請使用 `bfexport -f <file_name>`。您必須指定 `-f <file_name>` 選項，才能產生一個可用來匯入專案資料的檔案。
- 在 z/Linux 平台上，您必須將指令執行成 `bfexport.pl`。而在其他所有平台上，指令不需要副檔名。

必備條件與限制

請在 Build Forge 安裝目錄中尋找 bfexport 公用程式。

匯出檔中不會包含伺服器的授權密碼；在匯入之後，您必須在使用者介面中手動輸入伺服器授權密碼。

bfexport 指令必須能夠找到 buildforge.conf 檔以及存取 Build Forge 資料庫。請從 buildforge.conf 所在的目錄執行 bfexport，此目錄在 Windows 上是 <bfinstall>，而在 UNIX 及 Linux 上則是 <bfinstall>/Platform。

範例

若要將輸出寫入至檔案，請使用 -f <file_name> 選項。在下列範例中，helloworld 為輸出檔名稱，並且使用的是專案 ID 而非專案名稱。

```
bfexport -c "Saving a copy of project before making changes"
-f helloworld 675B57CC-8366-11DD-B2E0-043C04E44E1A
```

若只要匯出預設專案 Snapshot，請使用 <project_name>。

```
bfexport -f helloworld test_project
```

若要匯出專案的一個 Snapshot，請使用 <project_name> <snapshot_name>。

```
bfexport -f helloworld test_project snapshot_1
```

如果母專案 Snapshot 不是預設專案，您必須指定 <project_name> 且後面跟著 parent 關鍵字，才能匯出母專案 Snapshot。

```
bfexport -f helloworld test_project parent
```

選項說明

選項	說明
<project_name>	要匯出的專案名稱。專案名稱或專案 ID 為必要的。如果專案名稱含有空格，則必須用引號括住名稱。 請在指令選項後面指定專案名稱。
<snapshot_name>	要匯出的專案 Snapshot 名稱；專案名稱為必要的，其語法如下： <project_name> <snapshot_name> 請在指令選項後面指定專案名稱和 Snapshot 名稱。如果專案或 Snapshot 名稱含有空格，則必須用引號括住名稱。
<project_id>	要匯出之專案的 ID。專案 ID 為一個 UUID。專案 ID 或專案名稱為必要的。 請在指令選項後面指定專案 ID。
-f <file_name>	bfexport 輸出用的 XML 檔名稱。如果您沒有提供路徑名稱，則會採用現行工作目錄。如果檔名包含空格，必須用引號括住名稱。 如果您沒有提供檔名，則會將 bfexport 輸出傳送至標準輸出。 註：標準輸出僅供檢視用。請勿將標準輸出重新導向至檔案；產生的檔案含有記載訊息，因此無法作為 bfimport 指令或使用者介面匯入公用程式的匯入檔。

選項	說明
parent	如果母專案 Snapshot 不是預設的專案 Snapshot，則必須使用這個關鍵字，來匯出母專案 Snapshot。 請在專案 Snapshot 名稱後面指定 parent 關鍵字： bfexport -f helloworld test_project parent
-l	依名稱和專案 ID，列出資料庫中的專案。-l 選項不能搭配其他選項使用。
-c "<comment>"	併入註解。您必須用引號括住註解（例如 "my project version 50"）。註解會新增至 <buildforge> XML 元素。
-g	將隸屬於指定來接收通知之存取群組的使用者，儲存至 XML 檔。使用者及其內容會列在 <user> XML 元素中。需要 -s。
-s	將定義在「管理主控台」中的伺服器，儲存至 XML 檔。伺服器及其內容，連同任何相關聯的 <auth> 和 <collector> 資訊，都會列在 <server> XML 元素中。
-L	將定義在使用者介面中的 LDAP 網域控制器，儲存至 XML 檔。LDAP 網域控制站及其內容會列在 <ldap> XML 元素中。
-n	將指派給專案和步驟的通知範本，儲存至 XML 檔。通知範本及其內容會列在 <mail-template> XML 元素中。
-C	將針對專案指派給伺服器的收集器，儲存至 XML 檔。收集器及其內容會列在 <collector> XML 元素中。需要 -s。

疑難排解

如果您遇到的是找不到報告檔案或程式庫的錯誤，請確定已正確地設定程式庫路徑。

- Linux 系統及 UNIX 系統上的 `LD_LIBRARY_PATH`。
- AIX 系統上的 `LIBPATH`。

正確的設定位於 `<bfinstall>/rc/buildforge` 中。請檢查程式庫路徑變數，以及在該檔案中設定的其他變數。視需要重設它們，然後匯出它們。

使用 .export

您可以使用 `.export` 從專案中的某個步驟匯出專案。

`.export` 指令不會提供選項讓您匯出其他任何的物件資料。如果要匯出其他的 Build Forge 物件，則必須使用 `bfexport` 指令。

請參閱 第 351 頁的『.export』。

匯入專案

您可以使用 `bfimport` 指令或主控台中的「匯入」機能，來匯入先前匯出的專案與其他的 Build Forge 物件。

系統提供一系列的選項，讓您選取要安裝的物件。

您可以選擇如何將存取群組套用至匯入的物件。這些是透過使用安全存取來匯入系統設定來設定的。

註：Build Forge 使用者介面可能會拒絕匯入大於 2M 的檔案。在此情況下，請直接使用 `bfimport` 指令行工具，或調整 `"upload_max_filesize"` PHP 環境變數。

使用「匯入」公用程式來匯入專案與其他物件

利用使用者介面「匯入」公用程式，可匯入已儲存至匯出檔的專案與其他物件的物件定義。您可以使用「匯入」公用程式來選取要從匯出檔匯入的物件。

註：如果您匯入的專案有通知群組，且該通知群組不是內建的群組（例如「建置工程師」、「開發人員」、「訪客」、「操作員」、「系統管理員」），則您必須在匯入之前建立存取群組。

1. 選取**管理** > **匯入**。

使用者介面配置

主控台

報告

登出: Root User

匯入

說明 ?

重設

匯入

專案 XML 檔:

瀏覽...

選取要處理的實體:

☒ 匯入專案

☐ 包含鏈接的專案

☒ 匯入伺服器定義

☒ 匯入環境變數

☒ 匯入專案類別

☐ 匯入使用者

☐ 匯入專案特定範本

☐ 匯入過濾條件

☐ 匯入收集器

☐ 匯入選取元

☐ 匯入 LDAP 網域設定

當名稱相同時:

☐ 取代實體

☒ 重新命名實體

重設

匯入

2. 若要尋找專案的匯出 XML 檔，請按一下**瀏覽**。

註：您必須使用 `bfexport` 指令或 `.export` 點指令來建立匯出檔。

3. 從匯出 XML 檔中，選取要匯入至使用者介面的專案和其他物件。

註：如果匯入伺服器物件，則在匯入之後，您必須在使用者介面中手動輸入其伺服器授權密碼。 **bfexport** 和 **.export** 指令不會將伺服器授權密碼儲存至匯出檔。

- 若要指定匯入公用程式是要取代或改寫同名的物件，還是要將它們重新命名，請選擇 **取代實體** 或 **重新命名實體**（預設選項）。

重要：若想瞭解如何重新命名與取代工作，請參閱第 420 頁的『匯入時重新命名與取代物件』。

bfimport 參照

使用 **bfimport** 公用程式可將專案和其他物件的定義，匯入至先前已匯出至 XML 檔的使用者介面中。您也可以使用「匯入」公用程式，從 XML 檔匯入選取的檔案。

本主題說明 **bfimport** 指令的語法，並提供用法詳細資料。

語法

bfimport

bfimport [-L] <file_name>

bfimport [-p -I -s -S -e -c -C -u -T -f -d -r] <file_name.xml>

bfimport [-L | [-p -I -s -S -e -c -C -u -T -f -d -r]] <file_name.xml>

限制與考量

匯出檔中不會包含伺服器的授權密碼；您必須手動輸入伺服器授權密碼。

依預設，在匯入時，如果有物件的名稱與所匯入的物件相同則會重新命名要匯入的物件，以防資料庫物件遭到改寫。或者，如果存在同名的物件，您可以選擇使用 **-r** 選項來取代物件。就 **Snapshot** 物件而言，將會依 **UUID** 來比較物件，而不是依名稱。

依預設，**bfimport** 會重新命名物件，並採用下列的命名慣例：

<object_name>_IMPORT_<number>

如果重新命名，則 **Snapshot** 物件會失去其 **Snapshot** 名稱，並匯入成新的基礎或母項層次的 **Snapshot**，即使該 **Snapshot** 物件是某個母項 **Snapshot** 的子項也一樣。

如需重新命名的詳細資料，請參閱第 420 頁的『匯入時重新命名與取代物件』。

如果要取代物件，您必須指定 **-r** 選項。取代選項會改寫現有的物件。若為 7.0.2 和更早版本的匯出檔，則不會取代 **Snapshot** 物件。而會使用 <object_name>_IMPORT_<number> 慣例將它們重新命名。若為 7.1 匯出檔，則會在 **UUID** 相符時取代 **Snapshot** 物件。

必備條件

使用 **bfimport** 公用程式之前，請確定您符合下列必要條件：

- 以 **bfexport** 指令或 **.export** 點指令建立的匯出 XML 檔。
- 請在 **Build Forge** 安裝目錄中尋找 **bfimport** 公用程式。
- bfimport** 指令必須能夠找到 **buildforge.conf** 檔，以及存取 **Build Forge** 資料庫。請從 **buildforge.conf** 所在的目錄執行 **bfimport**，此目錄在 **Windows** 上是 <bfinstall>，而在 **UNIX** 及 **Linux** 上則是 <bfinstall>/Platform。

註：如果您匯入的專案有通知群組，且該通知群組不是內建的群組（例如「建置工程師」、「開發人員」、「訪客」、「操作員」、「系統管理員」），則您必須在匯入之前建立存取群組。

用法

如果要完成一般的匯入作業，請使用下列的指令選項：

- 若要顯示指令語法，請使用 **bfimport**（不含任何選項）。
- 若要顯示 XML 檔中 Build Forge 物件的摘要清單與其名稱，請使用 **bfimport -L <file_name.xml>**。
- 在 zLinux 平台上，您必須將指令執行成 **bfexport.pl**。若為其他所有平台，指令不需要副檔名。
- 如果您沒有指定選項，則不會匯入任何物件。必須指定選項，以匯入個別的物件。
- 如果您針對不存在的物件指定選項，則匯入公用程式會跳過不存在於 XML 檔中的物件，並匯入位於檔案中的物件。

範例

若要列出 XML 檔中的 Build Forge® 物件，請只指定 -L 選項和 XML 檔名。下列範例顯示局部的指令輸出。

```
C:\Program Files\IBM\Build Forge>bfimport -L samples\projects\basic.xml
10/07/2008 5:31:55 PM: Import: 7624: CRRBF20081I: Importing export
file from a 7.0.10025 version console.
Project : [Basic Sample]
Tag Variable : [MAJ]
Tag Variable : [MIN]
Step : [Checkout Source]
Step : [Update Applet Version]
Step : [Create Baseline]
Environment : [Basic Environment]
Class : [Production]
Filter : []
Selector : [Web Server]
Selector : [Local Server]
```

如果要匯入 XML 檔中的所有物件，請按照以下範例所示，針對要匯入的物件指定選項。物件會在匯入時重新命名。成功陳述式顯示匯入是否成功。

```
C:\Program Files\IBM\Build Forge>bfimport -p -I -s -S -e -c -C -u -T -d -f
"samples\projects\basic.xml"
```

選項說明

選項	說明
<file_name.xml>	包含要匯入之 Build Forge® 物件的匯出 XML 檔名稱。XML 檔必須是使用 bfexport 指令或 .export 點指令所建立。XML 檔名是必要的，且如果 XML 檔不在現行目錄中（即發出 bfexport 指令的目錄），您必須提供路徑名稱。 如果檔名包含空格，必須用引號括住名稱。

選項	說明
-L	列出匯出 XML 檔中的物件和其物件名稱。請單獨使用這個選項；請不要與其他 bfimport 選項一起指定。-L 選項的輸出可傳送至標準輸出，或重新導向至 XML 或文字檔。
-p	從 XML 檔匯入專案配置資料。專案配置資料內含步驟與專案定義資料，包含標籤變數在內。 在重新命名時，會將專案名稱匯入至使用者介面成為 <code><project_name>_IMPORT_<number></code> 。
-I	匯入專案或步驟層次所參照的鏈結專案或程式庫。 在重新命名時，會在使用者介面中，將鏈結的專案或程式庫匯入至使用者介面成為 <code><project_or_library_name>_IMPORT_<number></code> 。
-S	匯入定義在使用者介面中的選取元物件。 在重新命名時，會將選取元匯入至使用者介面成為 <code><selector_name>_IMPORT_<number></code> 。
-s	如果在 bfexport 中指定了 -s 選項，則會匯入定義在「管理主控台」中的伺服器物件。 在重新命名時，會將伺服器匯入至使用者介面成為 <code><server_name>_IMPORT_<number></code> 。 匯出 XML 檔中不會包含伺服器的授權密碼；您必須手動輸入伺服器授權密碼。
-e	匯入專案或步驟層次所參照的環境和其變數。 在重新命名時，會將環境匯入為 <code><environment_name>_IMPORT_<number></code> 。
-c	匯入專案所參照的類別。 在重新命名時，會將類別匯入為 <code><class_name>_IMPORT_<number></code> 。
-C	如果在 bfexport 中指定了 -C 選項，則會匯入針對專案而指派給伺服器的收集器。 在重新命名時，會將收集器匯入為 <code><collector_name>>_IMPORT_<number></code> 。
-u	如果在 bfexport 中指定了 -g 選項，則會匯入屬於指定來接收電子郵件通知之存取群組成員的使用者。 在重新命名時，會將使用者的資訊匯入為 <code><users>_IMPORT_<number></code>
-T	如果在 bfexport 中指定了 -n 選項，則會匯入使用者所建立指派給專案和步驟的通知範本。 在重新命名時，會將通知範本匯入為 <code><template_name>_IMPORT_<number></code> 。
-f	如果在 bfexport 中指定了 -n 選項，則會匯入指派給專案步驟的日誌過濾器。 在重新命名時，會將日誌過濾器匯入為 <code><filter_name>_IMPORT_<number></code> 。

選項	說明
-d	<p>如果在 <code>bfexport</code> 中指定 <code>-L</code> 選項，則匯入使用者介面中定義的 LDAP 網域控制器。</p> <p>在重新命名時，會將日誌過濾器匯入為 <code><LDAP_domain_controller>_IMPORT_<number></code>。</p>
-r	<p>取代所匯入的物件，而不是將它們重新命名。依預設，匯入的物件會重新命名，並採用下列的命名慣例：</p> <p><code><object_name>_IMPORT_<number></code></p> <p>若為 7.1 物件，如果您指定取代選項，則 <code>bfimport</code> 指令會針對 7.1 物件，改寫使用者介面中的物件。</p> <p>若為 7.0.2 和更早的物件，則不會取代 Snapshot 物件。而會使用下列命名慣例將它們重新命名：</p> <p><code><object_name>_IMPORT_<number></code></p> <p>重要：在評估是否要取代 Snapshot 物件時會以 UUID 為依據，而不是名稱。</p> <p>如需詳細資料，請參閱『匯入時重新命名與取代物件』。</p>

如何將存取群組指派給匯入的物件

使用安全存取來匯入系統設定，用來控制如何將存取群組指派給匯入的物件。

- 如果使用安全存取來匯入設為「是」（預設值），匯入預設安全存取群組設定會指定存取群組。預設值為「建置工程師」。
- 如果使用安全存取來匯入設為「否」，便以匯入不安全預設存取群組設定來指定存取群組。預設值為「開發人員」。

如果未指定匯入不安全預設存取群組的存取群組，便會使用最近建立的存取群組。如果未建立任何存取群組，就使用「預設」存取群組。

存取群組指派和安全

Build Forge 採取一項安全措施，不容許直接匯出或匯入存取群組。如果容許的話，有權存取主控台主機的使用者，便可以直接操作已匯出物件的 XML 檔，然後將其匯入。在匯入時指派存取群組，可確保只由已授權的使用者來指派存取群組。

匯入時重新命名與取代物件

如果資料庫中已存在同名的物件，`bfimport` 指令與「匯入」公用程式會將匯入的物件重新命名。匯入時重新命名物件，是預設的行為。

如果要變更這項行為，而在匯入時取代現有的物件，您必須在 `bfimport` 中指定 `-r` 選項，或者在使用者介面中選取「取代實體」選項。

下列主題說明 `bfimport` 指令與「匯入」公用程式在重新命名與取代匯入的物件時，所使用的命名慣例。

如果 Snapshot 物件（專案、選取元和環境）有 Snapshot 名稱的話，則會予以保留；否則，便會在匯入時指派一個預設的 Snapshot 名稱給它們。

重新命名 7.1 或更早版本的匯出檔中的物件

對於 7.1 和更早版本的匯出檔中的物件，bfimport 指令與「匯入」公用程式會使用下列的命名慣例，來重新命名使用者介面中的物件。

物件狀態	使用者介面中的物件名稱	使用者介面中的 Snapshot 名稱（僅適用於 Snapshot 物件）
新建，不在資料庫中	<new_object_name>	<snapshot_name>
資料庫中現有的	<existing_object_name>_IMPORT_<n>	<snapshot_name> 基礎 Snapshot

Snapshot 物件會匯入成新的母項層次 Snapshot，即使它是某個母項 Snapshot 的子項也一樣。只有專案、選取元和環境才能作為 Snapshot 物件。如果該 Snapshot 名稱已存在，則會保留。否則，便會指派預設的「基礎 Snapshot」名稱，如表格中所示。

取代 7.1 以前之匯出檔中的物件

對於 7.1 以前之匯出檔中的物件，bfimport 指令與「匯入」公用程式會使用下列的命名慣例，來取代使用者介面中的物件。請注意，在評估是否要取代 Snapshot 物件時會以 UUID 為依據，而不是名稱。

物件狀態	使用者介面中的物件名稱	使用者介面中的 Snapshot 名稱（僅適用於 Snapshot 物件）
新建，不在資料庫中	<new_object_name>	基礎 Snapshot
資料庫中現有的非 Snapshot 物件	<existing_object_name>	不適用
資料庫中現有的 Snapshot 物件	<existing_object_name>_IMPORT_<n>	基礎 Snapshot

就 Snapshot 物件方面，Snapshot 物件會匯入成新的母項層次的 Snapshot，即使該 Snapshot 物件是某個母項 Snapshot 的子項。只有專案、選取元和環境才能作為 Snapshot 物件。7.1 以前的匯出檔無法包含具有 Snapshot 名稱的物件，因此會指派預設的「基礎 Snapshot」名稱，如表格中所示。

取代 7.1 匯出檔中的物件

對於 7.1 匯出檔中的物件，bfimport 指令與「匯入」公用程式會使用下列的命名慣例，來取代使用者介面中的物件。

物件狀態	使用者介面中的物件名稱	使用者介面中的 Snapshot 名稱（僅適用於 Snapshot 物件）
新建，不在資料庫中	<new_object_name>	<snapshot_name> 基礎 Snapshot
資料庫中現有的	<new_object_name>	<snapshot_name> 基礎 Snapshot

在評估是否要取代 Snapshot 物件時會以 UUID 為依據，而不是名稱。Snapshot 物件會匯入成新的母項層次 Snapshot，即使它是某個母項 Snapshot 的子項也一樣。只有專案、選取元和環境才能作為 Snapshot 物件。所有的 7.1 物件會有一個唯一的 Snapshot 名稱，或是使用預設的「基礎 Snapshot」名稱。

Snapshot 匯入與取代的實務範例

在評估是否要取代 Snapshot 物件時會使用其 UUID，而不是名稱。若要多多次匯入 Snapshot 物件多次，請多加注意。

以下實務範例將說明其行為。

1. Snapshot 物件從系統 A 匯出，然後匯入至空系統 B。在系統 A 上匯入的物件具有與系統 B 上相同的 UUID。
2. 在系統 A 上修改 Snapshot 物件屬性，包括名稱在內。
3. 此 Snapshot 物件又重新從系統 A 匯出，並使用取代選項匯入至系統 B。由於 UUID 相符，因此取代了此 Snapshot 物件。系統 B 上取代後的物件此時會具有該物件在系統 A 上所有變更過的屬性，包括已變更的名稱在內。

移轉舊式 Perl 工作控制器專案

若要使用「Java 主要工作控制器」來執行在 n Build Forge 7.x 或更早版本中建立的舊式 Perl 工作控制器專案，您必須將 Perl 專案複製成 Java MJC 專案。

您可以在使用者介面上複製單一專案或程式庫。您也可以使用 `bfperlprojectmigration` 公用程式，一次將所有 Perl 專案和程式庫（包括鏈關係）複製到 Java MJC 專案和程式庫。

Bfperlprojectmigration 公用程式參照

使用 `bfperlprojectmigration` 公用程式，可一次將所有 Perl 專案和程式庫（包括鏈關係）複製到 Java MJC 專案和程式庫。

用法

`bfperlprojectmigration.sh` [連線選項] [指令選項]

連線選項

-h

--help 顯示這個說明文字。

-S

--secure

對伺服器要求安全的 SSL/TLS 連線。

預設值：使用一般 TCP 連線（不安全）

-C /path/to/bfclient.conf

--config=/path/to/bfclient.conf

置換要使用的用戶端配置檔的位置。

預設值：bfclient.conf（在現行目錄中）

-H Hostname

--hostname=Hostname

設定連線的遠端主機名稱。

預設值：localhost

-P Port

--port=Port

設定連線的通訊埠。

預設值：3966

-d Domain

--domain=Domain

設定鑑別的網域（領域）。

預設值：無

-u Login

--user=Login

設定鑑別的使用者登入。

預設值：root

-p Password

--password=Password

設定鑑別的使用者密碼。

預設值：root

-E Encoding

--encoding=Encoding

設定輸出字集編碼。如果提供無效的編碼，則會列出所有可用的編碼，且程式會結束。

註：是否能夠以所要求的編碼來顯示文字，視用戶端上所使用的終端機軟體功能而定。

預設值：UTF-8

指令選項

-a

--all 將系統中的所有舊式 Perl 專案和程式庫複製成 MJC 專案和程式庫。如果指定這個選項，則會忽略 "-o" 選項。

-s NameSuffix

--suffix=NameSuffix

複製的 MJC 專案或程式庫的名稱字尾。如果未提供，則依預設會使用字串 "Copy" 作為名稱字尾。

預設值：TimeStamp

註：如果多次使用相同的字尾，則系統會在名稱尾端附加索引號碼，以避免重複。

-o project1Name[, "project2 Name", projectUUID]

--one=project1Name[, "project2 Name", projectUUID]

目前不支援。遞迴地將給定的 Perl 專案或程式庫複製到 MJC 專案或程式庫，包括鏈結的專案和程式庫。

範例：

- UNIX/Linux : bfperlprojectmigration.sh -H localhost -u root -p root -a -s MJC
- Windows : bfperlprojectmigration.bat -H localhost -u root -p root -a -s MJC

將系統中的所有 Perl 專案和程式庫（包括它們之間的關係）複製成 MJC 專案和程式庫，並提供新的名稱，例如 origName_MJC。

第 24 章 在「使用者介面配置」標籤中鏈結至 Web 資源

使用**使用者介面配置**標籤，將標籤新增至 Build Forge 使用者介面。

您可以使用**使用者介面配置**標籤，將標籤新增至 Build Forge 使用者介面。每一個新標籤皆含有一個 URL。您可以使用這些標籤鏈結至外部資源，例如應用程式、作業系統、伺服器或使用者的相關資訊。

註：請勿建立鏈結至內部 Build Forge URL 的標籤。

如果要新增標籤，請執行下列動作：

1. 選取**使用者介面配置**標籤。
2. 按一下**新增標籤**。
3. 在**名稱**中，輸入標籤的標題或名稱。
4. 在**鏈結**中，您可以輸入下列任何項目：

選項	用法	範例
含通訊協定的 URL	<ul style="list-style-type: none">• 如果通訊協定不是 HTTP，則為必要• 如果是 HTTP，則為選用	http://www.ibm.com
不含通訊協定的 URL	使用預設 HTTP 通訊協定	www.ibm.com

5. 在**目標**中，選取下列一個選項：
 - 內部（預設）：選取此項時，會在現有瀏覽器視窗中開啓鏈結。
 - 外部（預設）：選取此項時，會在新瀏覽器視窗中開啓鏈結。
6. 在**已啓用**中，選取下列一個選項：
 - 已啓用（預設）：選取此項時，會啓用標籤。如果啓用，在您選取標籤時，會試圖連接 URL。
 - 已停用：選取此項時，會停用標籤。如果已停用，則會隱藏此標籤。
7. 在**可見**中，選取下列一個選項：
 - True（預設）：選取此項時，會在使用者介面中顯示標籤。
 - False：選取此項時，會在使用者介面中隱藏標籤。
8. 在**限 Root** 中，選取下列一個選項：
 - True（預設）：僅容許 root 使用者或具備 root 存取權的 Build Forge 使用者，才能選取標籤和連接 URL 鏈結。
 - False：容許任何 Build Forge 使用者選取標籤和連接 URL 鏈結。

註：在儲存新標籤之後，該標籤的標題會出現在清單中。如果標籤本身沒有出現，請重新整理頁面。

第 25 章 Build Catalyst

Build Catalyst 可以加速 make-based C 及 C++ 軟體建置。它會解譯並分析您現有的 make 檔並執行加速建置。加速建置可以使用平行建置及分散式建置等技術，減少整體的建置時期。

Build Catalyst 雖然和 IBM® Rational® Build Forge® 一起提供，但是需要另外安裝。如果要搭配 Rational Build Forge 使用 Build Catalyst，請在安裝了 Rational Build Forge 代理程式的主機上，安裝 Build Catalyst。Build Catalyst 加速建置安裝並配置好之後，就可由 Rational Build Forge 步驟加以呼叫。與 Rational Build Forge 專案整合可供使用者存取 Rational Build Forge 所提供更廣泛的自動化工具。例如，呼叫重構的 make-based 建置可能是包含其他建置的較大型建置實務範例的一部分。Rational Build Forge 專案也能執行預先建置和後建置的程序。

Build Catalyst 與專為 GNU Make 3.80 撰寫的 make 檔高度相容。如果您有為 GNU Make 3.80 撰寫的 make 檔，在大部分情況下，您無須變更 make 檔，就能使用 Build Catalyst 來建置您的原始碼。部分情況可能需要稍做變更。Build Catalyst 支援以下建置：

- **平行建置**：Build Catalyst 可以辨認非獨立目標並加以平行建置。平行建置可以更充份運用建置電腦中的資源，如多核心處理器。藉著有效率地使用資源，Build Catalyst 可以加速建置並減少建置時期。
- **分散式建置**：分散式建置和平行建置相似，然而其建置活動分散於多部電腦上。

Build Catalyst 說明文件適用於熟悉軟體建置概念的 Build Catalyst 新手及有經驗的使用者。

支援的作業系統

Build Catalyst 可以用於 Linux®、Solaris 及 Microsoft® Windows® 等多種作業系統上。

下列作業系統支援 Build Catalyst：

- Red Hat Enterprise Linux 4、5
- Solaris 9、10 (SPARC)
- Microsoft Windows XP Professional
- Microsoft Windows Server 2003

註：只有 Linux 及 Solaris 作業系統支援分散建置。

安裝概觀

您從保存檔將 Build Catalyst 安裝於 Linux®、Solaris 或 Microsoft® Windows® 作業系統上。

如需相關資訊，請參閱『支援的作業系統』。

Build Catalyst 具有獨立的安裝程式，與 IBM® Rational® Build Forge® 安裝程式不同。如果要搭配 Rational Build Forge 使用 Build Catalyst，請在安裝了 Rational Build Forge

代理程式的同一部電腦上，安裝 Build Catalyst。您可以將 Build Catalyst 安裝在您計劃作為建置主機的任何電腦上，只要電腦執行受支援的作業系統即可。如果您不計劃擁有 Rational Build Forge 工作（此工作會在指定電腦上使用 Build Catalyst），那麼電腦就不需要安裝 Rational Build Forge 代理程式。

IBM 以保存檔形式發佈 Build Catalyst，而此保存檔包含下列項目：

- 「堆存檔案」(dump file)，其中包含所有可安裝檔案（二進位檔、各種內建 make 檔等等）
- 安裝 Script

在 Linux 和 Solaris 作業系統上安裝 Build Catalyst

您可以使用 Build Catalyst 保存檔將 Build Catalyst 安裝於 Linux[®] 和 Solaris 作業系統上。

關於這項作業

若要使用 Build Catalyst 進行分散式建置，請在分散式建置中所使用的所有電腦上安裝 Build Catalyst。

程序

1. 從 <http://hostname/buildcatalyst> 下載 Build Catalyst 保存檔，其中的 *hostname* 是您的「管理主控台」主機。
2. 將保存檔解壓縮到任何目錄。
3. 如果您不是超級使用者，請輸入 `su`，成為超級使用者。
4. 跳至您解壓縮安裝程式的目錄，然後跳至包含 `install.sh` 檔案的新子目錄。
5. 在那個目錄中執行 `./install.sh`。
 - a. 指定安裝目錄。（預設目錄是 `/opt/rational/buildforge/buildcatalyst`。）

如果安裝已經存在（即使不是位於您所提供的目錄），安裝程式會警告，如果您繼續安裝，現有安裝將被改寫或變成無法使用。（不在預設目錄中的現有安裝會被視為無法使用，因為現行安裝程序會從預設安裝目錄，建立指向新指定安裝目錄的符號鏈結。）接著安裝程式會再次提示安裝目錄。在您提供安裝位置之後，安裝會將檔案寫入所提供的位置。

- b. 指定分散式建置要使用 `rsh` 或 `ssh` 二進位檔（預設程式是 `rsh`，如果能在路徑上找到的話。）

Build Catalyst 二進位檔會在安裝目錄的 `bin` 目錄中，安裝為 `rafmake`。安裝會建立下列項目：

- 從 `/opt/rational/buildforge/buildcatalyst` 到安裝目錄的符號鏈結。
- 從安裝目錄中的 `etc/rafbe_starter` 到 Shell 二進位檔的符號鏈結。Shell 二進位檔不是 `rsh` 就是 `ssh`。

結果

現在您可以從 IBM[®] Rational[®] Build Forge[®] 工作啟動 `rafmake`，以建立 C 或 C++ 軟體。您也可以 Rational Build Forge 工作之外，單獨啟動 `rafmake`。

將 Build Catalyst 安裝在 Windows 上

您可以使用 Build Catalyst 保存檔將 Build Catalyst 安裝在 Microsoft® Windows® 上。

開始之前

驗證電腦上已經安裝 Microsoft Visual C++ 2005 SP1 Redistributable Package version 8.0.50727.762 版。您可以使用「控制台」中的**新增或移除程式**項目，驗證所安裝的是哪個套件。

重要：Build Catalyst 不使用此套件的其他版本。您必須安裝第 8.0.50727.762 版。

如有需要，您可以從 www.microsoft.com/downloads/ 下載正確的版本。

如需 Windows 支援的相關資訊，請參閱第 427 頁的『支援的作業系統』。

程序

1. 從 <http://hostname/buildcatalyst> 下載 Build Catalyst 保存檔，其中的 *hostname* 是您的「管理主控台」主機。
2. 將保存檔解壓縮到任何資料夾。
3. 開啓該資料夾，然後開啓內含 `install.bat` 檔案的資料夾。
4. 執行 `install.bat`。Build Catalyst 二進位檔會安裝為 `rafmake.exe` 檔案。下列表格指出檔案會儲存到 `bin` 資料夾。

表 12. 安裝資料夾

作業系統類型	位置
32 位元作業系統	C:\Program Files\IBM\Build Forge\buildcatalyst
64 位元作業系統	C:\Program Files (x86)\IBM\Build Forge\buildcatalyst

結果

現在您可以從 IBM® Rational® Build Forge® 工作啓動 `rafmake`，以建立 C 或 C++ 軟體。您也可以 Rational Build Forge 工作之外，單獨啓動 `rafmake`。

Build Catalyst 範例

使用以下範例以更充分地瞭解，如何將 Build Catalyst 用於簡式建置、平行建置和分散式建置。

簡式建置

如果是簡式建置（不是平行建置或分散式建置），請依下列方式執行 **rafmake** 指令：

```
rafmake [ -f makefile ] [ options ] ... [ targets ] ...
```

其中

- *makefile* 是所要使用的 `make` 檔
- *options* 包含 第 432 頁的『`rafmake` 公用程式參照』所解釋的選項
- *targets* 為所要建置的目標檔

下列清單依作業系統提供範例。

- Linux 和 Solaris
 - 在特定 `make` 檔上，以其所有的相依關係無條件地建置預設目標：

```
% rafmake -u -f project.mk
```
 - 在特定 `make` 檔上，以詳細輸出建置特定的目標：

```
% rafmake -v -f project.mk a.out
```
- Windows
 - 在預設 `make` 檔上，以 `INCL_DIR` 巨集所設定的特定值建置預設目標：

```
Y:\> rafmake INCL_DIR=C:\src\include
```

平行建置

rafmake 指令支援平行建置，可將建置區隔為可同時建立的獨立元件，以減少整體建置時間。

用來執行平行建置的語法如下所示：

```
rafmake -J num [ -f makefile ] [ options ] ... [ targets ] ...
```

其中

- *num* 是同時建立獨立目標之建置工作的數目
- *makefile* 是所要使用的 `make` 檔
- *options* 包含 第 432 頁的『**rafmake** 公用程式參照』所解釋的選項
- *targets* 為所要建置的目標檔

執行平行建置能提供有效率的資源使用率，對多核心處理器更是如此。從平行建置獲得的效能增益視建置類型而定。例如，在 CPU 密集的建置案例中，同時發生之建置工作的最佳數目和處理器（或核心）的數目相同。如果您指定的數目超過處理器數目，可能就無法實現明顯的效能增益。然而若是 IO 密集的建置，增加平行工作的數目會減少建置時間。您可以嘗試建立幾種不同工作數目的建置，以判斷對您的資源而言，同時發生之建置工作的最佳數目。

下列清單依作業系統提供平行建置的範例。

- Linux 和 Solaris
 - 在特定 `make` 檔中，平行建立具有 3 個同時發生之建置工作的預設目標：

```
% rafmake -J 3 -f project.mk
```
- Windows
 - 在預設 `make` 檔中，以 `INCL_DIR` 巨集所設定的值，平行建立具有 5 個同時發生之工作的特定目標：

```
Y:\> rafmake -J 5 INCL_DIR=C:\src\include software.exe
```

分散式建置

平行建置只能在一部電腦上使用資源。分散式建置則較平行建置更往前邁進一步，將建置負載分散於多部電腦上。

註：若要使用 Build Catalyst 進行分散式建置，請在分散式建置中所使用的所有電腦上安裝 Build Catalyst。

如果要分散式建置，您的環境必須符合下列需求：

- 原始碼在不同電腦之間共用。

在您要分散式建置的所有電腦上，相同的原始碼必須透過相同的路徑提供存取。一般來說，在裝載了 NFS 的分享上具有原始碼（並將分享裝載於所有電腦的相同路徑上），可確保所有建置電腦都在相同路徑提供相同的原始檔目錄。

- 透過選擇的遠端 Shell（rsh 或 ssh）進行登入是配置為無需密碼。

當使用者從主機啟動分散式建置時，**rafmake** 指令會針對要分散式建置的所有電腦，從內部啟動建置引擎（rafbe）。建置引擎可透過 **ssh** 或 **rsh** 指令啟動，這是在安裝 Build Catalyst 時所配置的。`/opt/rational/buildforge/buildcatalyst/etc/rafbe_starter` 檔案是使用者在安裝期間所選擇之遠端 Shell 的符號鏈結。啟動建置的使用者必須能夠無密碼登入所有使用選擇的遠端 Shell 的電腦。在無密碼登入的情況下，建置引擎可以在多部電腦上無聲自動啟動。請查閱《**rsh** 或 **ssh** 指令手冊》，以設定無密碼登入。

- 建置 hosts 檔說明如何將建置分散於不同電腦上。

建置 hosts 檔通常包含數行文字，載明多部電腦的主機名稱或 IP 位址。這些就是要用來分散式建置的電腦。檔案同時包含數行用來說明其他選項的文字，如 `-idle num`。這類型的行說明下列電腦必須達到何種閒置程度，電腦上的建置才會進行排程。如果沒有明確提到，閒置程度通常假定為 50。以下是建置 hosts 檔範例：

```
### 建置 hosts 檔內容
host1.mydomain.com
-idle 25
host2.mydomain.com
host2.mydomain.com
192.168.0.200
-idle 75
host3.mydomain.com
### 建置 hosts 檔結束
```

在這個範例中，建置被分散到 4 部電腦：`host1`、`host2`、`host3` 及 `192.168.0.200`。`host1.mydomain.com` 電腦的閒置程度假定為 50，因為沒有明確提到。`host2` 和 `192.168.0.200` 電腦的閒置程度則指定為 25。這表示如果其中一部電腦的負載或忙碌程度在 75% 以下（或閒置程度至少為 25%），則會在該電腦上進行建置排程。如果電腦的閒置程度在 25% 以下，就不會在這些電腦上排程建置。另外，範例中列出 `host2` 電腦兩次。因此，可以在 `host2` 電腦啟動 2 個平行建置執行緒。您可以根據電腦的配置調整建置 hosts 檔，將資源充份運用於建置上。

在您建立建置 hosts 檔之後，就能使用下列指令語法啟動分散式建置：

```
rafmake -J num -B build_hosts_file [ -f makefile ]
        [ options ] ... [ targets ] ...
```

`-J` 選項的 `num` 值指定所有電腦的建置執行緒總計數目。

如果沒有 `-B` 選項，**rafmake** 指令會執行平行建置而非分散式建置。當您指定 `-B` 選項及有效的建置 hosts 檔時，建置引擎會在建置 hosts 檔所指定的不同電腦上，進行分散式建置的排程。

rafmake 公用程式參照

Build Catalyst 的介面是名為 **rafmake** 的 **make** 公用程式。此介面具有多種可控制其行為的選項。

指令的語法視作業系統而定，如下列表格所示。

表 13. 各作業系統的選項

作業系統	選項
Linux 和 Solaris (建置目標。)	<code>rafmake [- f <i>makefile</i>] ... [- u <i>kinservwdpqUN</i>] [- J <i>num</i>] [- B <i>bldhost-file</i>] [- c <i>compat-mode</i>] [- C <i>change-directory</i>] [- A <i>BOS-file</i>] ... [<i>macro=value</i> ...] [<i>target-name</i> ...]</code>
Windows (建置目標。)	<code>rafmake [- f <i>makefile</i>] ... [- u <i>kinservwdpqUN</i>] [- J <i>num</i>] [- c <i>compat-mode</i>] [- C <i>change-directory</i>] [- A <i>BOS-file</i>] ... [<i>macro=value</i> ...] [<i>target-name</i> ...]</code>
全部 (顯示 rafmake 公用程式的版本資訊。)	<code>rafmake { -ver/sion -VerAll }</code>

選項說明

rafmake 指令支援 GNU 3.80 版 **make** 指令大部分的一般選項。**rafmake** 指令也提供其他選項。

提示：您可以結合不帶引數的選項，例如，**-rNi**。

表 14. 選項說明

選項	說明
-f <i>makefile</i>	將 make 檔當作輸入檔使用。如果您省略這個選項， rafmake 指令會在現行工作目錄中，尋找名為 makefile 和 Makefile (依此順序) 的輸入檔。您可以指定 1 個以上的 -f <i>makefile</i> 引數配對。多個輸入檔是有效連接的。
-u	(無條件) 重建所有指定的目標和其所有相依關係，不論是否需要重建它們。(另請參閱 -U 。)
-k	現行項目上的工作如果失敗了，就予以放棄，但繼續執行其他不依賴該項目的目標。
-i	忽略指令傳回的錯誤代碼。
-n	(不執行) 針對需要重建的目標，列出其 make 檔中的指令行，包含開頭為 at 符號 (@) 的指令行，但不加以執行。異常狀況：包含字串 \$(MAKE) 的指令一律會執行。
-s	(無聲自動) 不在執行指令行前列示出來。
-e	環境變數置換 makefile 檔中的巨集指定。(然而，指令行或建置選項規格中的 macro=value 指定會置換環境變數。)
-r	(沒有規則) 不使用 buildcatalyst-home-dir/etc/builtin.mk 檔 (Linux 和 Solaris) 或是 buildcatalyst-home-dir\etc\builtin.mk 檔 (Windows) 中的內建規則。和 -c 選項一起使用時， -r 選項也會停用平台特定啟動檔案的讀取。如需相關資訊，請參閱 -c 選項。
-v	(詳細) 比預設的輸出模式略為詳細。
-w	(工作目錄) 列印包含執行 make 檔之前及之後的工作目錄的訊息。

表 14. 選項說明 (繼續)

選項	說明
-d	(除錯) 相當詳細，並包含 rafmake 指令在建置期間所讀取的环境變數的清單。只有在為 Make 檔進行除錯時，才使用這個選項。
-p	(列印) 列出全部的目標說明及巨集定義，包含目標特定的巨集定義及隱含的規則。不執行任何指令。
-q	(查詢) 對 make 檔目標求值，但不執行建置 Script。 rafmake 指令傳回下列其中一個回應： <ul style="list-style-type: none"> • 0，如果目標是最新的 • 1，如果有需要重建的任何目標
-U	僅無條件地重建主要目標 (goal target)。不建置子目標。如果您未在指令行指定任何目標，則預設目標就是主要目標。(-u 選項會無條件地建置主要目標和建置相依關係。)
-N	停用讀取一個以上建置選項規格 (BOS) 檔的預設程序。
-J num	啟用平行建置功能。同時發生的目標重建數目上限設為整數 <i>num</i> 。如果 <i>num</i> =0，平行建置就會停用。(這等於不指定 -J 選項。)或者，您可以將 <i>num</i> 指定為 RAFMAKE_CONC 環境變數的值。
-B bldhost-file	將 <i>bldhost-file</i> 檔當作平行建置的建置 hosts 檔使用。如果您不指定 -B 選項， rafmake 指令會使用您起始目錄中的 .bldhost.\$RAFMAKE_HOST_TYPE 檔。當您使用 -B 選項時，必須也使用 -J 選項或是設定 RAFMAKE_CONC 環境變數。
-c compat-mode	(相容性) 以相容模式呼叫 rafmake 指令。 (或者，您可以使用 RAFMAKE_COMPAT 環境變數來指定相容模式。) 您可以將 <i>compat-mode</i> 變數設為下列其中一個值： gnu 模擬「免費軟體基金會」的 Gnu make 程式。要定義內建的 make 規則時， rafmake 指令會讀取 gnubuiltin.mk ，而非 builtin.mk 。 這個模式是預設的相容模式。 std 在啟用 IBM Rational® ClearCase® clearmake 指令相容模式的情況下，執行 rafmake 指令。 <i>compat-mode</i> 變數也可以具有下列其中一個值 (僅適用於 Linux 和 Solaris 系統)。-c 選項可在不同的 Linux 和 Solaris 平台上執行。然而，有些模式會嘗試讀取系統特定的檔案；如果這些檔案不存在，則指令會失敗。 sun 模擬 SunOS 系統提供的標準 make(1) 指令。 aix 模擬 IBM AIX® 系統提供的標準 make(1) 指令。
-C dir	變更目錄。 rafmake 指令會在啟動建置前，變更至 <i>dir</i> 目錄。

表 14. 選項說明 (繼續)

選項	說明
-A <i>BOS-file</i>	<p>指定在依預設讀取的 BOS 檔案之後，接著立即要讀取的 BOS 檔案。您可以多次使用這個選項，指定多個 BOS 檔案。</p> <p>如果您對這個選項使用 -N，rafmake 會讀取指定的 BOS 檔，而非預設的 BOS 檔。</p> <p>或者，您也可以針對 BOS 檔案路徑名稱，指定以冒號區隔的清單（適用於 Linux 和 Solaris）或是以分號區隔的清單，作為環境變數 RAFMAKE_OPTS_SPECS 的值。</p>
-version	列印 rafmake 指令的版本資訊。
-VerAll	列印 rafmake 指令以及 rafmake 指令使用的程式庫（Linux 與 Solaris）或 DLL (Windows) 的版本資訊。

Build Catalyst 環境變數

Build Catalyst 支援數個環境變數以簡化使用情形。

在下列區段中，將變數依作業系統分組：

- 『Linux、Solaris 和 Microsoft Windows 共同的環境變數』
- 第 435 頁的『僅適用於 Linux 和 Solaris 的環境變數』
- 第 436 頁的『僅適用於 Windows 的環境變數』

Linux、Solaris 和 Microsoft Windows 共同的環境變數

RAFMAKE_CONC

設定 **rafmake** 建置中的並發層次。這項變數的值和 -J 選項相同。若您在使用 **rafmake** 指令時指定 -J 選項，則會置換此項變數的設定。

RAFMAKE_COMPAT

指定 **rafmake** 指令相容模式。這項變數的值和 -c 選項相同。若您在使用 **rafmake** 指令時指定 -c 選項，則會置換此項變數的設定。

預設值：無

RAFMAKE_MAKEFLAGS

為 **rafmake** 指令選項的指定，提供替代方案或增補機制。RAFMAKE_MAKEFLAGS 環境變數可以包含用於指令行選項的相同字串或索引鍵字母，除非選項帶有效參數。如果發生衝突，您在 **rafmake** 指令行指定的選項會置換此環境變數的設定。

預設值：無

RAFMAKE_OPTS_SPECS

提供以冒號（Linux 和 Solaris）或分號（Windows）區隔的路徑名稱清單，每個路徑均指定 **rafmake** 指令所讀取的建置選項規格 (BOS) 檔案。您可以使用這個變數，取代在具有一個以上 -A 選項的 **rafmake** 指令行指定 BOS 檔案。

預設值：Undefined

RAFMAKE_SHELL_FLAGS

指定要傳遞至執行建置 Script 指令之 Subshell 程式的 **rafmake** 指令選項。

預設值：

- Linux 和 Solaris：-e
- Windows：None

RAFMAKE_SHELL_REQUIRED

強迫 **rafmake** 指令在您以 SHELL 巨集指定的 Shell 程式中執行建置 Script。如果要使 **rafmake** 指令在 Shell 程式中執行建置 Script，請將這個變數設為 TRUE。如果要配置 **rafmake** 指令以直接執行建置 Script，請取消設定變數。

預設值：**rafmake** 指令直接執行建置 Script

RAFMAKE_VERBOSE

指定 **rafmake** 指令訊息日誌記錄層次，如下所示：

1 等於指令行上的 -v（詳細）

2 等於指令行上的 -d（除錯）

0 或 undefined

等於標準訊息日誌記錄層次

如果您也在指令行上指定 -v 或 -d，會以較高的值為準。

預設值：0

MAKEFLAGS

列出一個以上要傳遞至 **make** 指令的旗標。**rafmake** 指令會在啟動時，讀取 MAKEFLAGS 環境變數的內容。接著指令會修正變數，納入從指令行傳遞而來、且非 Build Catalyst 特定的旗標。Build Catalyst 特定的旗標會透過 RAFMAKE_MAKEFLAGS 環境變數，如果 **rafmake** 指令在 MAKEFLAGS 變數中偵測到這些旗標，會將它們移至 RAFMAKE_MAKEFLAGS 變數。

透過 MAKEFLAGS 變數的旗標包含：-I、-p、-N、-w、-e、-r、-i、-k、-n、-q、-s

透過 RAFMAKE_MAKEFLAGS 變數的旗標包含：-A、-B、-N、-v、-c、-U、-M、-u、-d

預設值：無

僅適用於 Linux 和 Solaris 的環境變數

RAFMAKE_BLD_HOSTS

指定一個以上可供 **rafmake** 指令建立目標的建置主機。

預設值：Undefined

RAFMAKE_HOST_TYPE

決定在平行建置期間，所要使用的建置 hosts 檔名稱（-J 選項）：您起始目錄中的 .blldhost.\$RAFMAKE_HOST_TYPE。（您可以檢查密碼資料庫以決定起始目錄為何。）

在指令行指定 -B 選項會置換此項變數的設定。

C Shell 使用者：在 .cshrc 檔案中，而非 .login 檔案中設定此項變數。平行建置設施會呼叫遠端 Shell，而此 Shell 並不會讀取 .login 檔。

您也可以將 RAFMAKE_HOST_TYPE 編碼為 **make** 巨集。

預設值：無

僅適用於 Windows 的環境變數

RAFMAKE_NO_ESC_PATT_CHARS

在 **rafmake** 指令的 GNU 相容模式中置換跳出字元 (\)。例如，**rafmake** 和 GNU **make** 指令都假設 \% 字串指示文字字元 %。這兩個指令並不將這項規則視為型樣規則。如果要避免 **rafmake** 指令使用跳出字元來指示文字字元，請將這個環境變數設為任何非空值。

RAFMAKE_PNAME_SEP

為 **rafmake** 指令所建構的路徑名稱設定路徑名稱分隔字元。這個變數可以在 **make** 檔、BOS 檔、指令行上設定，或是設為環境變數。

預設值：這個變數如果沒有設定，或是設為正斜線 (/) 或反斜線 (\) 以外的任何值，則 **rafmake** 指令會以反斜線 (\) 作為路徑名稱分隔字元。

第 26 章 Rational Build Forge 整合方法

Build Forge 使用下列方法與其他產品整合：

- 使用指令行整合。Rational Build Forge 步驟在代理主機上執行指令。
- 使用配接器整合。配接器是具有特殊特性的 XML 型程式，可協助與來源控制管理系統進行整合。
- 使用 IDE 外掛程式整合。這些外掛程式可讓您從 IDE 使用者介面存取主控台。
- 與 Websphere 產品整合。您可以使用 Websphere Application Server 作為 Web 應用程式伺服器，來取代 Apache Tomcat。您也可以使用 IBM HTTP Server 來取代 Apache HTTP Server。

使用指令行介面整合

Build Forge 很容易與具有指令行介面的應用程式整合。

如果要設定這類整合，請執行下列動作：

1. 在主機上安裝 Build Forge 主控台
2. 在應用程式主機（或是可存取應用程式的主機）上安裝 Build Forge 代理程式
3. 在主控台中，建立伺服器資源和伺服器鑑別。將伺服器資源配置為存取已安裝的 Build Forge 代理程式。
4. 依需求配置代理程式和主機環境，使得指令可以在應用程式中執行。這就像建立 Build Forge 要使用的使用者帳戶，接著確保執行中的代理程式已正確設定其 PATH 一樣簡單。其可能也需要安裝及配置用來在應用程式中執行指令的用戶端。例如，Rational ClearCase 和 ClearQuest® 需要使用用戶端應用程式來執行指令。

完成設定後，您在 Build Forge 中建立的專案即可包含執行應用程式指令的步驟。您可以根據步驟的通過或失敗狀態來控制專案執行情況，或設定日誌過濾器以掃描 Build Forge 日誌中由指令傳回的輸出型樣。應用程式可能還需要在專案邏輯方面進行其他設定。例如，與 ClearCase 的整合將需要您包含視圖的建立、啟動及移入。

您可以使用條件步驟及迴圈步驟，控制在特定指令成功或失敗時要執行的動作。

有關來源控制應用程式的一般用法是省略建置。系統會執行指令，以查詢自前次執行建置後是否移入原始碼更新項目：

- 如果沒有任何更新項目，則不會執行建置。
- 如果有更新項目，則會執行建置

模組相依關係可以使用專案步驟的執行邏輯表示。您可以控制特定模組的建置，是否會觸發其他模組或整個軟體專案的建置。

您可以使用通知範本，控制在建置成功或失敗時，如何通知專案成員群組。在連續整合環境中，請務必在失敗時發出通知，以便能修正有問題的程式碼。使用指令行整合時，只能通知整個專案成員群組。專案成員需要檢查專案的步驟日誌，以判斷導致失敗的程式碼。

使用配接器整合

透過配接器，可以將 Rational Build Forge 與外部應用程式整合。

範例配接器範本可供整合 IBM Rational ClearCase 與 IBM Rational ClearQuest。Adaptor Toolkit 這個另行授權的特性可讓您撰寫自訂配接器。

配接器的行為是由 XML 檔所定義，而 XML 檔的元素則是由內含的 DTD 所指定。Build Forge 會執行與專案步驟關聯的配接器。

配接器提供您用於整合型建置的其他工具：

- 內部條件邏輯：根據內部變數的設定
- 指令定義：應用程式指令可以在指令陳述式和變數之外建置。
- 回應掃描：您可以定義要掃描的型樣，以回應每一個指令。
- 動態通知群組：通知可以根據從應用程式收集的資料而定。例如，要通知的該組團隊成員可以建置為僅通知移入程式碼變更的成員。
- 執行 Script：配接器可以在 Build Forge 主機上執行 Script。這可以與應用程式主機上執行的 Script 和指令無關。

配接器需求

這個主題識別 Build Forge 配接器的安裝與配置需求及軟體需求。

檢查您的環境是否符合與 Build Forge 產品一起提供之任何配接器的使用需求：

- 在外部應用程式正在執行的電腦上，安裝 Build Forge 代理程式。部分產品有其他的設定需求。Rational Clear Case 和 Rational ClearQuest 需要其他的設定。請參閱這些產品在「整合」小節中的資訊。
- 如果您要使用 ClearCase 或 ClearQuest 以外的應用程式範本，請安裝 Build Forge Adaptor Toolkit 的授權碼。
- 請執行 Build Forge 產品所支援的外部應用程式版本。

表 15. 配接器有支援的應用程式版本

應用程式	版本
Rational ClearCase	6.0 或更新版本
Rational ClearQuest	7.0 或更新版本
CVS	1.1、1.2
Microsoft Visual SourceSafe	6.0
Perforce	2005.1
StarTeam	2005 版本 2
子版本	1.3.1 與以上的版本

配接器

這個主題提供一些關於配接器的一般資訊。另外也說明配接器如何與其他 Build Forge 物件和功能互動。

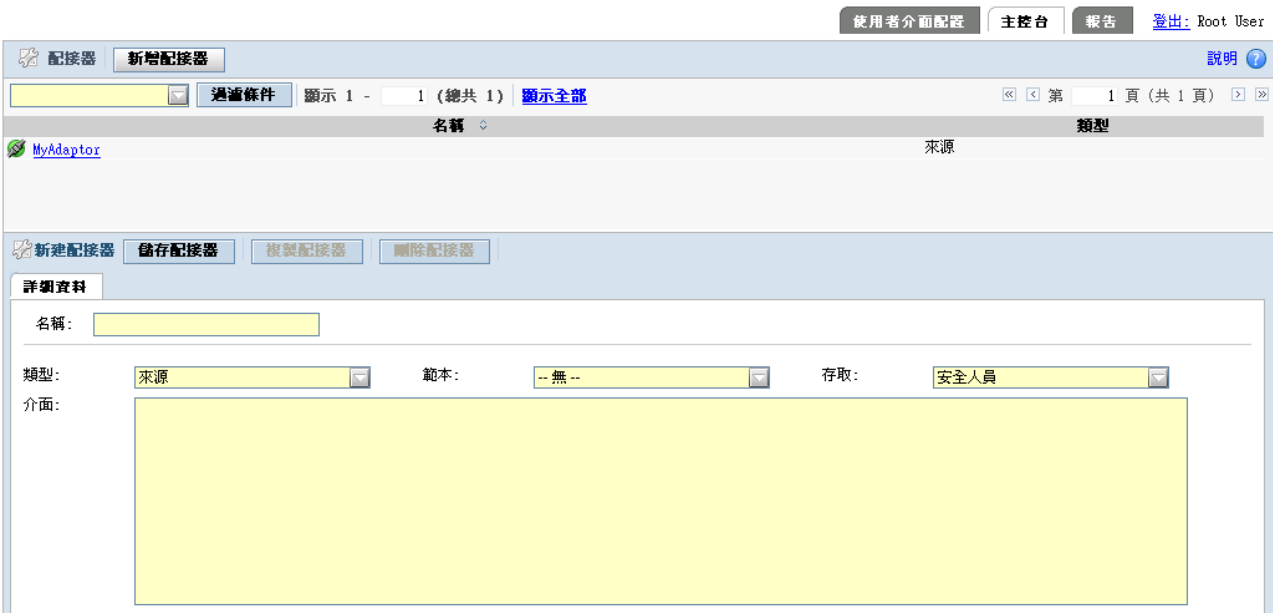
請檢閱本章節所提供的資訊，以便著手執行配接器。

關於配接器

配接器是外部應用程式的介面。配接器可以讓 Build Forge 專案與外部應用程式交換資訊以達成目標。

例如，原始碼應用程式的配接器會檢查程式碼變更的儲存庫，而這是執行 Build Forge 專案時的必備條件。如果原始碼檔案有變更，專案就會執行。如果沒有變更，專案就不會執行。

關於「配接器」畫面： 您可以使用「配接器」畫面來建立及編輯配接器。若要檢視畫面，請選取**專案 > 配接器**。



畫面具有下列欄位：

- 名稱** 不含空格的配接器唯一名稱
- 類型** 配接器類型
- 範本** Rational Build Forge 附有零或一個配接器範本
註：更新 Rational Build Forge 版本時，會清除這項設定。
- 存取** 存取群組會限制配接器檢視及編輯群組成員
- 介面** 由您或範本提供的 XML 介面

配接器範本

配接器為配接器範本的實例。當您建立配接器時，不僅會為其指定唯一名稱，還會將它與範本相關聯。

此範本為 XML 檔案。XML 所包含的應用程式指令可用來收集資訊、用來分析資訊的指令，以及在 BOM 報告中顯示結果的格式詳細資料。

The templates provided by Build Forge 所提供的範本原本是要在沒有修改的情況下使用。但您還是可以修改範本，或以範本為模型建立新的配接器範本。

配接器範本安裝在下列目錄中：

bfinstall\interface（執行 Microsoft Windows 作業系統的電腦）

bfinstall/Platform/interface（執行 UNIX 或 Linux 作業系統的電腦）

範例配接器範本

所提供的範例配接器範本適用於下列數種應用程式類型：原始碼、問題追蹤、測試和包裝。配接器依所支援的應用程式類型加以分類。

Build Forge 產品針對下表中的應用程式，提供範例配接器範本。ClearCase 和 ClearQuest 的範本不需要個別的授權金鑰。其他應用程式範本則透過 Build Forge Adaptor Toolkit 取得授權。

配接器範本 XML 檔安裝在下列目錄中，其副檔名為 .xml。

bfinstall\interface（執行 Microsoft Windows 作業系統的電腦）

bfinstall/Platform/interface（執行 UNIX 或 Linux 作業系統的電腦）

註：

如果執行專案時使用的是透過配接器鏈結附加的配接器，則當該配接器不在除錯模式且配接器步驟失敗時，配接器即會移除工作。

表 16. 配接器範本說明

配接器範本名稱	說明	類型
ClearCaseBaseline	在 ClearCase 視圖中掃描目錄。 將 ClearCase 報告的分支與版本資訊寫入 BOM 報告中。	原始檔
ClearCaseByBaselineActivities	從 ClearCase 視圖的內容建立新的基準線。 比較新的基準線與配接器前次執行時的基準線，以找出變更活動。 針對每次的變更活動，將下列資訊寫入 BOM 報告：活動、已變更的檔案、使用者、日期、註解與版本。 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。	原始檔
ClearCaseByBaselineVersions	從 ClearCase 視圖的內容建立新的基準線。 比較新的基準線與配接器前次執行時的基準線，以找出變更過的檔案。 針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、版本、日期、使用者與註解。 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。	原始檔

表 16. 配接器範本說明 (繼續)

配接器範本名稱	說明	類型
ClearCaseByDate	<p>在 ClearCase 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。</p> <p>針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、版本、日期、使用者與註解。</p> <p>針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔
ClearCaseByLabel	<p>建立新的標籤，並套用至 ClearCase 視圖的內容。</p> <p>比較新的標籤與配接器前次執行時的標籤，以找出變更過的檔案。</p> <p>針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、版本、日期、使用者與註解。</p> <p>針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔
ClearQuestBaseClearCaseByDate	<p>在 ClearCase 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。</p> <p>針對每個已變更的檔案，找出可用來識別 ClearQuest 變更 ID 的 CrmRequest 超鏈結屬性。透過新增工作資訊來試圖解析變更 ID，以解析 ClearQuest 中的問題記錄（如果 ClearQuest 狀態容許解析該問題記錄的話）。</p> <p>針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、問題 ID、問題狀態和 ClearQuest 的所有錯誤。</p>	問題
ClearQuestClearCaseByActivity	<p>找出與 ClearCase 活動清單相關聯的 ClearQuest 問題記錄。它會針對所找到的每個問題記錄新增工作資訊，以解析 ClearQuest 中的問題記錄（如果 ClearQuest 狀態容許解析該問題記錄的話）。</p> <p>將下列資訊寫入 BOM 報告：和 ClearCase 活動 ID 及 ClearQuest 問題狀態相關聯的檔案。</p>	問題
ClearQuestUCMClearCaseByDate	<p>在 ClearCase 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。</p> <p>請使用 Rational Unified Change Management (UCM) 來產生其結果。</p> <p>針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、問題 ID、問題狀態和 ClearQuest 的所有錯誤。</p>	問題
CVSv1Baseline	<p>掃描 Build Forge 代理程式上的 CVS 目錄，尋找已變更的檔案。</p> <p>將下列資訊寫入 BOM 報告：已變更的檔名、狀態、有效版本、儲存庫版本與固定標籤。</p>	原始檔

表 16. 配接器範本說明 (繼續)

配接器範本名稱	說明	類型
CVSv1ByDate	<p>在 CVS 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。</p> <p>將下列資訊寫入 BOM 報告：變更類型、日期、使用者名稱、版本和檔名。</p> <p>針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔
CVSv1ByTag	<p>將新的標籤套用至 CVS 模組。</p> <p>比較剛標示的模組，與在配接器前次執行期間標示之模組間的差異。</p> <p>將下列資訊寫入 BOM 報告：檔名、修訂、狀態、日期、時間、變更作者及提出註解。</p> <p>針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔
CVSv2ByDate	<p>在 CVS 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。</p> <p>將下列資訊寫入 BOM 報告：變更類型、日期、使用者名稱、版本和檔名。</p> <p>針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔
JazzJBE	用於 Rational Team Concert 3.x 以及更新版本的使用者。如需相關資訊，請參閱 Rational Team Concert 文件。	原始檔
JazzSCM	<p>查詢 Rational Team Concert 中是否有已變更的原始檔。</p> <p>將下列資訊寫入 BOM 報告中：已變更的原始檔、已變更之檔案的大小、已變更的元件，以及變更集。</p>	原始檔
JazzSCMv2	<p>查詢 Rational Team Concert 2.x 版是否有已變更的原始檔。</p> <p>將下列資訊寫入 BOM 報告中：已變更的原始檔、已變更的串流、已變更的元件、變更集、進行變更的使用者、移入註解及 Jazz 工作項目。</p>	原始檔
JUnit	<p>執行 JUnit 測試。</p> <p>將下列資訊寫入 BOM 報告：測試執行、失敗、錯誤和經歷時間。</p>	測試
PerforceByDate	<p>查詢 Perforce 用戶端於配接器執行後所發生的變更。</p> <p>將下列資訊寫入 BOM 報告：變更、日期、時間、使用者、Perforce 用戶端與註解。</p> <p>將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔

表 16. 配接器範本說明 (繼續)

配接器範本名稱	說明	類型
PerforceByRev	<p>查詢 Perforce 用戶端自前次儲存庫修訂後所發生的變更。</p> <p>將下列資訊寫入 BOM 報告：變更、日期、時間、使用者、Perforce 用戶端與註解。</p> <p>將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔
Quota	<p>查詢 Windows 資料夾，以判斷其任何子資料夾是否超過指定的臨界值大小。</p> <p>針對每個子資料夾，將下列資訊寫入 BOM 報告：資料夾大小、擁有者和前次修改日期。</p> <p>將超出臨界值大小的子資料夾清單寫入 BOM 報告。</p> <p>重要： 按照下列順序，將通知傳送給其目錄已超出大小的使用者：首先，依 Build Forge 系統中的名稱；接著，依使用者名稱。如果通知的使用者不存在，<notify> 將會失敗。</p>	原始檔
StarTeamBaseline	<p>查詢資料夾的 StarTeam 視圖，以收集檔案的相關資訊。</p> <p>將下列資訊寫入 BOM 報告：檔名、狀態、修訂和分支。</p>	原始檔
StarTeamByDate	<p>使用 StarTeam API 來查詢 StarTeam 視圖，以找出配接器前次執行時間與現行日期之間的變更。</p> <p>將下列資訊寫入 BOM 報告：已變更的檔案與目錄、使用者、版本、日期與變更註解。</p> <p>將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔
SubversionByDate	<p>查詢子版本在過去日期與現行日期之間所發生的儲存庫變更。</p> <p>將下列資訊寫入 BOM 報告：變更類型、修訂、使用者、檔案或目錄以及變更日期。</p> <p>將下列資訊寫入 BOM 報告：檔名、狀態、修訂和分支。</p>	原始檔
SubversionByRev	<p>查詢子版本在現行修訂與之前修訂之間所發生的儲存庫變更。</p> <p>針對每次的變更，將下列資訊寫入 BOM 報告：修訂、使用者、變更類型、檔案或目錄路徑以及變更日期。</p> <p>將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔
SynergyByDate	<p>在 Rational Synergy 中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。</p> <p>針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、版本、使用者、專案及日期。</p> <p>針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔

表 16. 配接器範本說明 (繼續)

配接器範本名稱	說明	類型
VSSByDate	<p>查詢 Visual Source Safe 目錄在之前日期與現行日期之間所發生的變更。</p> <p>將專案和檔案的變更資訊寫入 BOM 報告：專案或檔案、版本、使用者、日期、時間、專案活動、檔案專案與動作。</p> <p>將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。</p>	原始檔

配接器和專案

如果要執行配接器程式碼並與外部應用程式連接，必須將配接器新增至 Build Forge 專案。

配接器會以點指令或配接器鏈結新增至專案。

任何配接器都能透過適用於下列應用程式類型的點指令，來新增至專案中：
.source、.defect、.test 或 .pack。

只有原始碼配接器可以透過配接器鏈結新增至專案。會使用配接器鏈結而非 .source 指令，將配接器連接至專案。

配接器和環境變數

配接器需要環境變數才能執行應用程式指令。在配接器範本中，環境變數會列在 XML 檔之 <template> 區段的 <env> 元素中。

以 ClearCaseBaseline 配接器為例，下列環境變數會列在 ClearCaseBaseline.xml 檔中：

```
<template>
<!-- Template section, these are parsed out of the final xml.
Use the list below to help identify the variables needed to run this interface
if you are integrating it during a regular BuildForge step.
-->
<env name="VIEW" value="my_adaptor_view" />
<env name="VOB_PATH" value="\AdaptorVob" />
<env name="CCSERVER" value="BFServerName" /></template>
```

在 Build Forge 中，環境變數會儲存在環境中。建立配接器之前，請先為應用程式環境變數建立環境。

配接器和通知

大部分的配接器範本會傳送電子郵件通知給使用者。例如，ClearCaseByDate 配接器會在執行時，傳送「通過」電子郵件通知給變更原始碼檔案的使用者。如果沒有變更任何檔案，則會傳送「失敗」電子郵件通知。

您可以選擇性地修改配接器通知：

- 在配接器範本中複製 <adduser> 元素，將使用者新增至配接器通知群組。
- 在配接器範本中，使用 <notify> 元素來新增或刪除通知訊息。
- 在配接器專案方面，請設定專案層次的通知。
- 在配接器點指令步驟方面，請設定步驟層次的通知。

重要：如果 <notify> 指引失敗（例如，電子郵件所定址的使用者不存在），XML 將會失敗，且後續所有的通知也會失敗。

配接器和工作執行

當配接器專案使用配接器點指令時，可利用排定的工作，或是使用專案的任何手動啟動選項，來啟動配接器專案。

與配接器鏈的專案通常依排程執行。但是如果您完成某些額外設定，就能手動啟動與配接器鏈的專案。請參閱第 454 頁的『手動啟動配接器鏈的專案』。

配接器和工作結果

在配接器點指令專案方面，請在步驟日誌或 BOM 報告中檢視工作結果，如下所示：

- 選取**工作 > 已完成**。選取工作標籤，以檢視其步驟日誌。
- 選取**工作 > 已完成**。選取 BOM 標籤，依種類顯示 BOM 報告並檢視工作結果。

在與配接器鏈的專案方面，請在 BOM 報告的「原始檔變更」種類中檢視工作結果，如下所示：

- 選取**工作 > 已完成**。選取 BOM 標籤。在 BOM 報告中，找出「原始檔變更」種類的位置。

關於配接器鏈結

配接器鏈結會將配接器連接至專案，並將環境與配接器相關聯。

配接器鏈結只能與原始碼配接器一起運作。您可以使用點指令，將任何配接器類型與專案連接。

配接器鏈結具有下列特性：

- 將配接器程式碼新增至專案以作為步驟 0（執行其他專案步驟之前，對原始碼變更所做的測試）。
- 將應用程式環境變數自動移入環境中。
- 您可以選取「作用中」、「非作用中」、「除錯」的其中一種狀態，以控制是否執行配接器程式碼。
- 可讓您將配接器的運作當成手動啟動的工作來執行。

配接器鏈結具有下列限制：配接器鏈結僅針對一個配接器與一個專案所定義。如果要將同一個配接器用於其他專案，必須建立另一個配接器實例。

當您重新啟動工作時，重新啟動工作中任一配接器鏈結步驟的行為如下所示：

- 如果配接器鏈結的「狀態」為「非作用中」，則配接器鏈結步驟會失敗，並導致重新啟動執行失敗。
- 如果配接器鏈結的「狀態」為「作用中」且鏈結失敗，則配接器鏈結步驟會失敗，並導致重新啟動執行失敗。

這兩種情況都不會移除建置。不過，建置狀態會設為「失敗」。

關於「配接器鏈結」畫面： 您可以使用「配接器鏈結」畫面，將配接器連接到專案，並建立環境與配接器的關聯性。若要檢視畫面，請選取**專案 > 配接器鏈結**。



畫面具有下列欄位：

狀態 「作用中」、「除錯」或「非作用中」其中之一

註：這些狀態僅適用於一個配接器鏈結。您可以利用「鏈結除錯模式」系統設定，將所有配接器鏈結都設為「除錯」狀態。該狀態的優先順序高於系統配置設定。

配接器 鏈結到專案的配接器

進入點 XML 介面的進入點

專案 使用配接器的專案

環境 專案要使用的環境

配接器作業概觀

本主題涵蓋建立原始碼配接器、將它透過配接器鏈結連接至專案，並在測試模式中執行與配接器鏈的專案時，所需要的一切作業。

選取範本，以建立配接器

如果要選取範本來建立配接器，請執行下列動作：

1. 選取**專案** → **配接器**。
2. 按一下**新增配接器**。
3. 在「名稱」處，輸入配接器的唯一名稱。配接器名稱在整個配接器集中必須是唯一的，並且不能包含空格。
4. 在「類型」處，選取配接器類型。
5. 在「範本」處選取範本。清單會包含和 Build Forge 產品一起安裝的配接器範本。ClearCase 和 ClearQuest 配接器不需要個別的授權金鑰。其他配接器會經由 Adaptor Toolkit 個別授權。
6. 在「存取」處，選取存取群組。只有群組成員才能檢視或編輯配接器。
7. 按一下**儲存配接器**。

建立空的環境

如果要建立空的環境，請執行下列動作：

1. 選取**專案** → **環境**。
2. 按一下**新增環境**。
3. 在「名稱」處，輸入環境名稱。指派一個名稱，以說明環境的目的。
4. 在「存取」處，選取存取群組。只有這些群組成員才能檢視或編輯環境。
5. 按一下**儲存環境**。


將配接器新增至專案

如果要將配接器新增至專案，請執行下列動作：

1. 選取**專案** → **配接器鏈結**。
2. 按一下**新增配接器鏈結**。
3. 在「配接器」處，選取您所建立的配接器（和配接器範本）。
4. 在「專案」處，選取專案。清單會顯示尚未鏈結至配接器的專案。
5. 在「狀態」處，選取**作用中**。
6. 在「環境」處，選取您為配接器鏈結所建立的環境。
7. 在「將資料輸入環境」處，選取**是**。配接器範本中的應用程式環境變數會新增至環境。
8. 按一下**儲存**，將配接器鏈結至專案。配接器和專案會新增至配接器鏈結的清單中。

編輯環境變數

如果要編輯環境變數，請執行下列動作：

1. 選取**環境**。
2. 針對您所建立的環境，按一下**編輯圖示** 。此畫面會顯示自動新增至環境的配接器環境變數。
3. 檢視配接器範本所提供之環境變數的預設值。
4. 請視需要變更原始碼應用程式的預設值，以執行配接器專案。

condition 屬性

condition 屬性容許您使用 Perl 比較運算子，將條件套用至某些配接器內容。字串文字、數字或變數可在比較時使用。

condition 屬性的語法如下所示：

```
condition="true(<lvalue> <operator> <rvalue>)"
condition="false(<lvalue> <operator> <rvalue>)"
condition="hastext(variable)"
condition="isempty(variable)"
```

指定下列四種類型之一：

類型	說明
true	如果 lvalue 等於 rvalue，則求值成功。
false	如果 lvalue 不等於 rvalue，則求值成功。

類型	說明
hastext	如果值長度大於 0，則求值成功。
isempty	如果值長度為 0，則求值成功。

lvalue 和 rvalue 可以是字串、數字，或內含字串或數字的變數。條件運算子是指任何與 Perl 相容的條件運算子。以下是字串和數字條件運算子。您必須使用適當的運算子，否則，將會收到無法預期的結果。

字串運算子	數值運算子
eq	==
ne	!=
gt	>
lt	<
ge	>=
le	<=

數值運算子搭配字串使用時，不會傳回正確的結果，而在數值中使用字串比較運算子，也是同理可證。

例如：`condition="true("PASS"=="FAIL")`

上述條件一律傳回 `true`，這是不正確的。

條件範例：

- `condition="true($BF_SERVER eq "TEST_BOX")` - 只有在建置伺服器變數包含 `TEST_BOX` 時，才會執行項目。
- `condition="false($BF_BID <=141)` - 只有在建置標籤大於或等於 141 時，才會執行建置。

支援這些條件的配接器內容如下所示：

- `adduser`
- `bom`
- `run`
- `setenv`

仔細檢查 DTD 中 Build Forge 的現行安裝，以取得有關哪些內容支援 `condition` 屬性的最新資訊。

就支援條件的內容來說，您會看到類似下列的項目：

```
<!ATTLIST adduser condition CDATA #IMPLIED>
```

配接器 DTD 位於 `%BF_HOME%\`（若為 Windows）和 `$BF_HOME/Platform/interface`（若為 UNIX）中，其檔名為 `interface.dtd`。

在測試模式下執行配接器

如果要在測試模式下執行配接器，請執行下列動作：

1. 選取**管理** → **系統**。
2. 在系統配置參數清單中，選取**鏈結除錯模式**。
3. 在「鏈結除錯模式」中，選取**是**。
4. 按一下**儲存**。
5. 選取**工作** → **啟動**。
6. 在專案清單中，選取您在「啟動專案」頁面所建立的與配接器鏈的專案。
7. 按一下**執行**。

檢視工作狀態和日誌

如果要檢視配接器專案的工作狀態和日誌資訊：

1. 開啓**工作**。
2. 在專案清單中，找到與配接器鏈的專案，以檢視工作成功/失敗狀態。
3. 如果要檢視工作日誌：
 - 選取配接器專案的「標籤名稱」以存取工作日誌資訊。
 - 選取「資料清單」以存取 BOM 報告。

核心配接器作業

這個主題提供建立與配置配接器的相關資訊：

- **選取範本**：針對每個配接器應用程式範本加以說明。
- **建立環境**：說明可用來建立環境與配接器關聯性的選項。
- **建立配接器**：說明如何建立配接器，並建立配接器與範本的關聯性。
- **將配接器新增至專案**：說明可用來將配接器新增至專案的選項。
- **測試配接器**：說明如何僅測試配接器配置。

此外，這個主題也提供其他配接器作業的相關資訊：

- **設定配接器日誌層次**：說明如何控制配接器相關資訊的記載數量。
- **快速啟動配接器鏈的專案**：說明手動啟動的必要設定。
- **重設配接器範本**：說明何時需要重設，以更新範本資訊。

為配接器建立環境

配接器需要環境變數才能執行應用程式指令。在配接器範本中，環境變數會列在 XML 檔之 <template> 區段的 <env> 元素中。

請勿編輯配接器範本檔中的環境變數。在 Build Forge 產品中，您使用環境為配接器定義環境變數。

使用現有環境，或是建立配接器專用的環境。建議您建立新的環境，因為這樣可以為環境指派特定的用途和敘述性名稱，並能簡化疑難排解。

這個主題說明如何建立環境與配接器的關聯性：

- 經由配接器鏈結

- 在與配接器點指令相關聯的步驟中

為配接器鏈結建立環境： 如果您正使用配接器鏈結將原始碼配接器鏈結至專案，請使用這個方法。

1. 選取**專案 > 環境**。
2. 按一下**新增環境**。
3. 在「名稱」處，輸入環境名稱。指派一個名稱，以說明環境的目的。
4. 在「存取」處，選取存取群組。只有群組成員才能檢視或編輯環境。
5. 按一下**儲存環境**。這時請勿新增環境變數到群組中。這些環境變數會在您建立配接器鏈結時，從配接器範本自動填入。

為配接器點指令建立環境： 如果您正使用配接器點指令將配接器新增至專案，請使用這個方法。

如果要進行這項作業，您需要適用於配接器的環境變數。

在 `<adaptor_name>.xml` 檔中，外部環境變數會列在 `<template>/<env>` 元素中。

請在下列目錄中找出配接器範本：

`<bfinstall>\interface`

1. 選取**專案 > 環境**。
2. 按一下**新增環境**。
3. 在「名稱」處，輸入環境名稱。指派一個名稱，以說明環境的目的。
4. 在「存取」處，選取存取群組。只有群組成員才能檢視或編輯環境。
5. 按一下**儲存環境**。
6. 按一下**新增環境變數**。
7. 在「名稱」處，輸入要出現在 XML `<env>` 元素中的環境變數名稱。
8. 在「值」中，將 XML `<env>` 元素中的替代變數，變更為應用程式的實際值。（如果不知道正確值，可稍後再輸入。）
9. 在「動作」處選取**設定**。
10. 在「於專案」選取**一般**。

從範本來建立配接器

每個配接器都是以一個配接器範本為基礎。建立配接器之前，請先決定要使用哪個範本。

如果要建立配接器，請執行下列動作：

1. 選取**專案 > 配接器**。
2. 按一下**新增配接器**。
3. 在「名稱」處，輸入配接器的唯一名稱。配接器名稱在整個配接器集中必須是唯一的，並且不能包含空格。
4. 在「類型」處，選取配接器類型。
5. 在「範本」處選取範本。清單會包含和 Build Forge 產品一起安裝的配接器範本。ClearCase 和 ClearQuest 配接器不需要個別的授權金鑰。其他配接器會經由 Adaptor Toolkit 個別授權。

6. 在「存取」處，選取存取群組。只有群組成員才能檢視或編輯配接器。
7. 按一下**儲存配接器**。

新增配接器至專案

若要執行配接器程式碼，必須將配接器新增至專案。請建立新專案，或是將配接器新增至現有的專案。

本小節說明如何使用下列方法，將配接器新增至專案：

- 原始碼配接器可透過配接器鏈結新增至專案。
- 任何配接器（包含原始碼配接器）都能使用配接器點指令新增至專案。

使用配接器鏈結新增原始碼配接器： 配接器鏈結會將配接器連接至專案，並將應用程式環境變數連接至配接器。

啟動這項作業之前，請先為配接器建立專案和環境。

完成這項作業之後，請開啓環境並為應用程式環境變數提供真正的值（如果您還未執行這項作業的話）。

1. 選取**專案 > 配接器鏈結**。
2. 按一下**新增配接器鏈結**。
3. 在「狀態」處，選取狀態：

狀態	說明
作用中	在專案執行時，執行配接器程式碼。
非作用中	在專案執行時，跳過配接器程式碼。
除錯	僅執行配接器程式碼；在專案執行時，跳過其他步驟。

4. 在「配接器」處，選取配接器範本。清單會顯示和 Build Forge 產品一起安裝的配接器範本。
5. 在「專案」處，選取專案。清單會顯示尚未鏈結至配接器的專案。
6. 按一下**儲存**，將配接器鏈結至專案。配接器名稱會新增至清單。
7. 在「環境」處，選取配接器鏈結的環境。
8. 在「將資料輸入環境」處，選取**是**。配接器範本中的應用程式環境變數會新增至環境。
9. 按一下**儲存**以儲存配接器鏈結。

使用點指令來新增配接器： 任何配接器都能使用配接器點指令新增至專案。點指令會在步驟執行時呼叫 `<adaptor_name>.xml` 檔。

啟動這項作業之前，請先為配接器建立專案和環境。

完成這項作業之後，請開啓環境並為應用程式環境變數提供真正的值（如果您還未執行這項作業的話）。

如果要將配接器點指令新增至專案以作為步驟，請執行下列動作：

1. 選取**程式庫**。
2. 在清單中選取專案。

3. 按一下**新增**步驟。
4. 在「名稱」中，輸入步驟名稱。
5. 在「指令」中，輸入適用於應用程式類型的配接器點指令：
.source \.defect \.test \.pack。
6. 在「環境」中，選取為配接器建立的環境。
7. 按一下**儲存**步驟。

測試配接器

執行配接器專案以測試配接器配置。如果要檢查配接器是否能與外部應用程式互動並傳回預期結果，請在其他專案步驟之外，單獨執行配接器程式碼。

這個主題說明如何測試下列情況下的配接器：

- 透過配接器鏈結新增至專案
- 透過配接器點指令新增至專案

測試鏈結的配接器： 對於透過配接器鏈結來鏈結至專案的原始碼配接器，請使用這項程序來測試配接器配置。

一般程序如下所示：

1. 變更原始檔
2. 以鏈結的配接器執行 **Build Forge** 專案
3. 查看 **BOM** 報告以取得已變更的原始檔相關資訊
4. 查看是否收到通過或失敗通知的電子郵件

如果要測試鏈結的配接器，請執行下列動作：

1. 選取**專案 > 配接器鏈結**。
2. 在清單中選取鏈結的配接器與專案。
3. 在「狀態」處選取**除錯**。
4. 按一下**儲存**。
5. 在原始碼應用程式中，對一個以上的原始檔進行變更。送出變更以更新原始碼儲存庫。
6. 執行與配接器鏈的專案，如下所示：
 - a. 選取**管理 > 系統**。
 - b. 在清單中選取**鏈結除錯模式**。
 - c. 在「鏈結除錯模式」中，選取**是**。
 - d. 按一下**儲存**。
 - e. 選取**工作 > 啟動**。
 - f. 在專案清單中，從「啟動專案」頁面中選取與配接器鏈的專案。
 - g. 按一下**執行**。
7. 檢閱工作的 **BOM** 報告：
 - a. 開啓**工作**。
 - b. 選取**已完成**標籤，然後選取 **BOM** 標籤。

測試配接器點指令： 對於透過點指令新增至專案的配接器，請使用本選用程序來測試配接器配置。

一般程序如下所示：

1. 變更原始檔
2. 以配接器點指令執行 Build Forge 專案
3. 查看 BOM 報告以取得已變更的原始檔相關資訊
4. 查看是否收到通過或失敗通知的電子郵件

如果要測試配接器點指令，請執行下列動作：

1. 選取**工作 > 啟動**。
2. 在清單中選取專案。
3. 開啓**工作步驟**標籤。
4. 使用「步驟名稱」勾選框，清除配接器點指令以外的所有勾選項目。
5. 按一下**執行**以執行專案。
6. 檢閱工作的 BOM 報告：
 - a. 選取**工作**。
 - b. 選取**已完成**標籤，然後選取 **BOM** 標籤。

設定配接器日誌層次

如果要控制寫入配接器之步驟日誌的資訊量，請使用 `_InterfaceLoggingLevel` 環境變數。

1. 將 `_InterfaceLoggingLevel` 新增至配接器的環境。
 - 層次 8 記載的資訊量最多，層次 0 記載的資訊量最少。
 - 記載層次是內含的。例如，層次 2 包含層次 1 和 0 的資料。
 - 層次 7 為預設的日誌層次。

2. 指派一個日誌層次，作為 `_InterfaceLoggingLevel` 變數的值：

0：執行指令行加上伺服器連線錯誤，或是取消通知；沒有其他動作

1：已剖析的指令（就像要傳送至伺服器的指令）

2：未剖析的指令（設定區域變數之前的指令）

3：建置及環境變數 SET 行號

4：暫時及內部變數 SET 資料行號

5：環境求值、電子郵件群組新增、BOM 文字記載行

6：區塊及子區塊開始/結束行號

7：（預設日誌層次）將對照相符型樣來檢查的代理程式輸出，加上符合該型樣的行號

8：所有代理程式輸出

手動啟動配接器鏈的專案

如果您勾選「執行鏈結」勾選框，則配接器鏈的專案可依排程執行、使用快速啟動選項執行，或是以手動方式啟動。如果未勾選「執行鏈結」，則專案執行時不會使用配接器步驟。


1. 選取**工作 > 啟動**。
2. 在專案清單中，選取與配接器鏈的專案名稱。

這時會開啓「啟動專案」頁面。

3. 勾選**執行鏈結**勾選框。
4. 在「啟動專案」頁面上，按一下**執行**。

配接器鏈的專案的「快速啟動」

配接器鏈的專案可依排程執行，從「啟動專案」頁面（「工作」→「啟動」）選取專案名稱來手動啟動；如果您已設定「啓用快速啟動」系統配置設定，也可以使用「快速啟動」圖示來執行。下列步驟假設您已設定「啓用快速啟動」。

1. 選取**管理 > 系統**。
2. 在清單中，選取**鏈結手動工作**。
3. 在「鏈結手動工作」中，選取**是**。
4. 按一下**儲存**。
5. 選取**工作 > 啟動**。
6. 在專案清單中，選取配接器鏈結之專案的**快速啟動**圖示 ，即可立即執行該專案。
如果要驗證工作是否在執行中，請選取**工作 > 執行中**標籤。

重設配接器範本

關於這項作業

重設配接器範本時，會將最新的配接器範本從 `<bfinstall>\interface` 目錄複製到 Build Forge 資料庫。

每當您執行下列作業時，請重設配接器範本：

- 安裝維護版本或新的產品版本
- 透過在介面目錄（而非「管理主控台」）中編輯版本來修改範本
- 建立新配接器範本

程序

1. 選取**管理 > 系統**。
2. 在清單中，選取**重設配接器範本**。
3. 在「重設配接器範本」中，選取**是**。
4. 按一下**儲存**。

更新 ClearQuest 建置記錄

您可以將系統與 Rational ClearQuest 整合，以便自動建立並更新 ClearQuest 資料庫內部的建置記錄。

系統可以在 IBM Rational® ClearQuest® 資料庫中自動建立建置記錄，包含與日誌資料的鏈結。另外，當工作通過時，系統就能更新 ClearQuest 資料庫，以註明工作已經完成，並記錄結束時間以及所完成的步驟摘要。這項特性需要 Rational ClearQuest 7.0 版或更新版本。

註：為了順利建立建置記錄，並在其中移入建置的相關資訊，您必須先開啓 ClearQuest 的「維護工具」，並設定通往資料庫的連線。

當您配置專案以更新 ClearQuest 資料庫時，系統會執行下列作業：

工作開始

當系統啓動工作時，系統就會建立 ClearQuest 建置記錄。建置記錄會是「已提交」狀態，並且包含工作日誌 URL、開始時間、版次名稱和 ID，以及指出「建置 XYZ 已啓動」的日誌項目。如果來源控制配接器取消工作（例如，因為找不到來源變更），則不會建立 ClearQuest 建置記錄。

註：如果專案與其他專案鏈結，則新專案會取得專屬的唯一 ClearQuest 建置 ID。

工作通過/失敗

當工作通過或失敗時，系統會將 ClearQuest 內的建置狀態變更為「已完成」或「失敗」，再設定建置的結束時間，然後將工作步驟的摘要儲存在 ClearQuest 建置日誌中。摘要包含每個步驟的名稱、結果狀態與伺服器。

工作重新啓動

當工作重新啓動時，系統會將 ClearQuest 內的建置狀態變更為「已提交」，並建立 ClearQuest 建置日誌項目，指出「建置 XYZ 已重新啓動。」

您可以透過特殊的環境變數來配置 ClearQuest 整合。若要將專案鏈結至 ClearQuest 資料庫，請確定下表內的變數已併入專案的環境中。

註：這些變數必須呈現在專案環境中。光是將它們新增至步驟是不夠的。但是，您可以使用設定為「併入」類型的變數，經由另一個環境來併入這些變數。此外，由於 CQ_RELEASE_NAME 值是唯一可能會依專案而有所不同的值，您可能會想建立一個內含其他變數的環境，然後使用「併入」類型的變數將該環境併入在專案環境中，而在專案環境中，您也可以將 CQ_RELEASE_NAME 指定為專案特有的環境變數。

表 17. 進行 Rational ClearQuest 整合所需的環境變數

變數	說明
CQ_DBNAME	您要更新的 ClearQuest 資料庫名稱。
CQ_INTERACTION	如果您的專案環境已定義了正確的環境變數，以允許建立 ClearQuest 建置記錄，但是您並不要建立建置記錄，請將此變數設為 OFF 以停止建立建置記錄。 如果要啓用建置記錄的建立，請將這個環境變數設為 ON。 註： 如果您正使用其中一個 ClearQuest 配接器，請將這個環境變數設為 OFF。
CQ_USER	登入 ClearQuest 資料庫時所要使用的使用者名稱。
CQ_RELEASE_NAME	在您要更新的 ClearQuest 資料庫中的版次名稱。

表 17. 進行 Rational ClearQuest 整合所需的環境變數 (繼續)

變數	說明
CQ_PASSWORD	登入 ClearQuest 資料庫時所要使用的密碼。不是必要的；預設為空白。
CQ_DBSET	ClearQuest 資料庫設定值。不是必要的；預設為空白。

進階配接器作業

您可以針對您要透過介面與 Build Forge 專案相連的外部應用程式，修改 Build Forge 配接器範本或建立新的配接器範本。

如果要執行像修改和建立範本這樣的進階作業，需要具備下列專業常識：

- XML 語言
- 外部應用程式的指令語言
- 正規表示式

進行進階作業之前，請先閱讀下列小節的資訊：

- 第 449 頁的『核心配接器作業』
- 第 462 頁的『配接器範本結構』
- 第 460 頁的『配接器參照』

本節提供下列作業的相關資訊：

- **修改配接器範本**：說明如何修改所有配接器的範本。
- **修改單一配接器的範本**：說明如何修改一個配接器的範本。
- **建立新的配接器範本**：說明建立新範本的一般程序。
- **範例：將使用者新增至配接器通知**：說明如何將存取群組新增至電子郵件通知。
- **範例：移除 BOM 報告的變更詳細資料**：說明如何移除 BOM 報告的變更詳細資料。

修改配接器範本

如果您要讓未來從配接器範本建立的所有配接器都能採用範本修改，請使用這個程序。

開始之前，請先確定您要修改的內容。例如，您可能要變更通知或修改 BOM 報告格式。

1. 使用 XML 編輯器，開啓您要修改的配接器範本。配接器範本位於下列目錄中：

`bfinstall\interface` (執行 Microsoft Windows 作業系統的電腦)

`bfinstall/Platform/interface` (執行 UNIX 或 Linux 作業系統的電腦)

2. 輸入範本修正內容。
3. 儲存配接器範本。**請勿**變更範本名稱。配接器和配接器範本的名稱必須是唯一的。

下次當您以修改過的配接器範本建立配接器時，新的配接器就會包含範本修正。

修改單一配接器的範本

使用這項程序可以只修改單一配接器實例的範本。

開始之前，請先確定您要修改的內容。例如，您可能要變更通知或修改 BOM 報告格式。

如果您要修改的配接器已經建立，請透過「管理主控台」來修改範本，如下所示：

1. 選取**專案 > 配接器**。
2. 請在文字方塊中，輸入範本的修改內容。
3. 按一下**儲存配接器**。

註：您所做的變更只會儲存至配接器範本的實例，而此配接器範本在 Build Forge 資料庫中與配接器相關聯。變更不會儲存至 `interface` 目錄中的配接器範本檔。

如果您要修改的配接器尚未建立，請在建立配接器之前修改範本，如下所示：

1. 使用 XML 編輯器開啓配接器範本。配接器範本位於下列目錄中：

`bfinstall\interface`（執行 Microsoft Windows 作業系統的電腦）

`bfinstall/Platform/interface`（執行 UNIX 或 Linux 作業系統的電腦）

2. 輸入範本的修正內容。
3. 變更配接器名稱，然後儲存配接器範本。

註：配接器和配接器範本的名稱必須是唯一的。

4. 重設配接器範本以採用修改過的範本，同時將它新增至「管理主控台」的可用範本清單中。請參閱第 454 頁的『重設配接器範本』。

建立新配接器範本

請使用這項程序，針對您要透過介面與 Build Forge 專案相連的外部應用程式，建立新的配接器範本。

1. 檢閱 Build Forge 產品所提供之配接器範本中的 XML 結構和元素。配接器範本位於下列目錄中：

`bfinstall\interface`（執行 Microsoft Windows 作業系統的電腦）

`bfinstall/Platform/interface`（執行 UNIX 或 Linux 作業系統的電腦）

2. 請規劃新配接器的用途：
 - 瞭解它會執行哪些指令
 - 判斷指令的剖析方式
 - 判斷要如何處理從剖析結果收集的資料
 - 瞭解需要哪些外部環境變數
3. 在介面目錄中選取一個配接器範本以當作模型使用。可能的話，請選取基於相同外部應用程式的配接器。或是選取功能和您正在建立之配接器相似的配接器。以模型中的 XML 階層、元素和元素屬性為指南，來建立新的範本。
4. 使用 XML 編輯器，開啓模型配接器範本。以新的名稱將範本儲存至介面目錄中。
5. 使用您的計劃來撰寫新配接器的 XML 程式碼。
6. 儲存新的配接器範本。
7. 使用介面目錄中的 `interface.dtd` 檔案來驗證配接器範本。

8. 重設配接器範本以採用新的範本，同時將它新增至「管理主控台」的可用範本清單中。請參閱第 454 頁的『重設配接器範本』。
9. 為配接器建立專案。
10. 為配接器建立環境。請參閱第 449 頁的『為配接器建立環境』。
11. 使用新的配接器範本建立配接器。請參閱第 450 頁的『從範本來建立配接器』。
12. 將配接器新增至專案中。請參閱第 451 頁的『新增配接器至專案』。
13. 執行配接器專案以測試配接器。請參閱第 452 頁的『測試配接器』。

建立多個進入點配接器

Build Forge 所提供的配接器範本是單一進入點配接器。

若為單一進入點配接器，您可以在「管理主控台」（**專案 > 配接器**）中，選取要執行的應用程式和函數的範本名稱。例如，ClearCaseBaseline 或 ClearCaseByDate。

您也可以建立一個 ClearCase 配接器範本，其中包含多個 ClearCase 介面或函數。若為多個進入點配接器，您可以依名稱（稱為進入點）來識別每一個介面。

如果要建立多個進入點配接器，請執行下列動作：

1. 建立配接器範本。如果要建立範本，您可以選擇使用提供的其中一個範本作為模型，並視需要來修改 XML。在範本中，您必須將 name 屬性新增至 <interface> 元素中，以識別新增至範本中的每一個介面的進入點。下列範例顯示 <interface> 元素的相關語法：

```
<interface name="By Date" default="true">
</interface>
```

2. 以不含空格的唯一名稱建立配接器，並將其與配接器範本相關聯。請參閱第 450 頁的『從範本來建立配接器』。
3. 使用配接器點指令或配接器鏈結，將配接器新增至專案。
 - 下列範例使用 .source 配接器指令，將配接器新增至專案步驟，該步驟會呼叫 ClearCase 配接器中的 By Date 介面函數：

```
.source ClearCase "By Date"
```
 - 如果要使用配接器鏈結來呼叫多個進入點配接器，請採取配接器範本內的下列其中一個動作，指定專案執行時所要執行的介面函數：
 - 放置要執行之函數的 <interface> 元素定義，作為範本檔中的第一個 <interface> 元素
 - 將要執行之函數的 <interface> 元素的 default 屬性設為 true (default="true")

範例：啓用電子郵件通知

您可以配置配接器範本，以便向在外部應用程式中造成變更的使用者傳送電子郵件通知。下列範例顯示如何設定兩種類型的通知：

- 對在現行建置移入檔案的所有使用者發出通知
- 對 Build Forge 存取群組的所有成員發出通知

下列程序會參照 ClearCaseByDate 範本中的元素。所有配接器範本都能藉由其元素來啓用通知功能。

對在現行建置移入檔案的所有使用者發出通知

您可以使用 `<adduser>` 指令，動態建置移入建置程式碼的使用者群組，然後使用 `<notify>` 指令將通知傳送至該群組。

`ClearCaseByDate` 範本會向 `ClearCase` 查詢是否有視圖含有兩個時間戳記之間的所有變更。預設的時間戳記適用於現行的配接器作業與前次的配接器作業。就實際而言，可轉換成自前次針對現行建置移入建置以來，所作的所有變更清單。

假設：您用來依使用者名稱發出通知的 SMTP 伺服器，能夠識別視圖中所有使用者的名稱。這表示 `ClearCase` 使用者名稱必須與電子郵件使用者名稱一致。

如果要啟用這項通知功能：

1. 在 XML 編輯器中開啓 `ClearCaseByDate` 配接器範本。
2. 尋找並編輯 `<adduser>` 以建立一組使用者，如下所示：

```
<adduser group="MyChangers" user="$4">
```

位置參數 `$4` 會參照 `ClearCase` 視圖（由 `ClearCaseByDate` 範本產生）所顯示的使用者名稱欄位。

3. 設定通知以將電子郵件傳至此群組。下列設定會在專案失敗及成功時傳送電子郵件。在部分環境中，您可能希望只有當建置失敗時才發出通知。

```
<!-- Set some notifications for when the build completes -->
<onproject result="fail">
  <notify group="MyChangers" subject="Build $BF_TAG ($CurDate) Failed." message="{Changing}{Changes}"/>
</onproject>
<onproject result="pass">
  <notify group="MyChangers" subject="Build $BF_TAG ($CurDate) Passed." message="{Changing}{Changes}"/>
</onproject>
```

4. 儲存配接器範本。

配接器在執行時，會從視圖中的使用者名稱建置 `MyChangers` 群組。電子郵件通知會在建置專案完成時，傳送至該群組。

對屬於 **Build Forge** 存取群組的所有使用者發出通知

在本範例中，您想通知 `Build Forge` 存取群組的所有成員。範例中使用的是 `ClearCaseByDate` 配接器範本。假設：`Build Forge` 中的所有使用者名稱都對應於 SMTP 伺服器中的電子郵件使用者名稱。

1. 在 XML 編輯器中開啓 `ClearCaseByDate` 配接器範本。
2. 尋找並編輯 `<adduser>`，從 `Build Forge` 存取群組建立一組使用者，如下所示：

```
<adduser group="Developer_Access_Group" user="Developer">
```

3. 設定通知以將電子郵件傳至此群組。下列設定會在專案失敗及成功時傳送電子郵件。在部分環境中，您可能希望只有當建置失敗時才發出通知。

```
<!-- Set some notifications for when the build completes -->
<onproject result="fail">
  <notify group="Developer_Access_Group"
    subject="Build $BF_TAG ($CurDate) Failed." message="{Changing}{Changes}"/>
</onproject>
<onproject result="pass">
  <notify group="Developer_Access_Group"
    subject="Build $BF_TAG ($CurDate) Passed." message="{Changing}{Changes}"/>
</onproject>
```

4. 儲存配接器範本。

配接器在執行時，會從屬於 `Developers` 存取群組的使用者名稱建置 `Developer_Access_Group` 群組。電子郵件通知會在建置專案完成時，傳送至該群組。

重要：如果 `<notify>` 指引失敗（例如，電子郵件所定址的使用者不存在），`.xml` 將會失敗，且後續所有的通知也會失敗。

範例：移除 BOM 報告中的變更詳細資料

大部分的配接器範本都會將變更詳細資料記錄到 BOM 報告中。（`diff` 指令即用來記錄變更詳細資料。）

下列步驟會參照 `ClearCaseByDate` 範本中的元素，但它們可用來移除任何配接器範本的變更詳細資料。

如果要移除 BOM 報告中的變更詳細資料，請執行下列動作：

1. 在 XML 編輯器中開啟配接器範本。
2. 找出呼叫 `diff` 指令的 `<run>` 元素。移除下列幾行：

```
<run command="cc_diff" params="$VIEW $1 $2" server="$CCSERVER" dir="/" timeout="360"/>
```

3. 找出 `diff` 指令的 `<command>` 元素。移除下列幾行：

```
<!-- The cc_diff command does a generic clearcase diff, logging the full output
of the diff in the BuildForge BOM -->
<command name="cc_diff">
<execute>
pushd \\view\%1 && cleartool diff -pred -diff_format "$2@@$3"
</execute>
<resultsblock>
<match pattern=".">
<bom category="Source" section="diff">
<field name="diff" text="$_" />
</bom>
</match>
</resultsblock/>
</command>
```

4. 尋找 `<bomformat>` 區段，然後尋找 `diff` 指令輸出的 `<section>` 元素。移除下列幾行：

```
<section name="diff">
<field name="diff" title="Change Details"/>
</section>
```

5. 儲存配接器範本。

配接器參照

配接器必須新增至 Build Forge 專案，並在未經修改的情況下執行。如果要修改配接器或建立新的配接器，您必須瞭解 Build Forge 配接器範本中使用的 XML 範本結構與元素。

註：本節並未說明 Build Forge 配接器範本中使用的外部應用程式指令。如需這些指令的相關資訊，請參考外部應用程式的文件。

隨 Build Forge 產品安裝的配接器範本位於下列目錄中：

`bfinstall\interface`（執行 Microsoft Windows 作業系統的電腦）

`bfinstall/Platform/interface`（執行 UNIX 或 Linux 作業系統的電腦）

本節提供下列參考資訊：

- **配接器需求**：說明配接器的一般使用需求，以及 ClearQuest 配接器範本的特定需求。
- **配接器的點指令**：說明配接器點指令的語法。
- **ClearCase 和 ClearQuest 的環境變數**：說明 ClearCase 和 ClearQuest 配接器所使用的環境變數。
- **Perforce 環境變數**：說明 Perforce 所需的一些其他環境變數。
- **配接器範本結構**：說明 Build Forge 配接器範本的一般結構。
- **配接器 XML 參照**：說明 Build Forge 配接器範本中所使用的 XML 元素。

適用於配接器的點指令

某些點指令可讓您將外部應用程式的配接器，新增至 Build Forge 專案以作為專案步驟。

- **.source**：將原始碼應用程式的配接器新增至專案步驟。
- **.defect**：將問題追蹤應用程式的配接器新增至專案步驟。
- **.test**：將測試應用程式的配接器新增至專案步驟。
- **.pack**：將包裝應用程式的配接器新增至專案步驟。

另請參閱第 343 頁的『點指令參照』。

Rational ClearCase 與 Rational ClearQuest 環境變數

Rational ClearCase 和 Rational ClearQuest 的配接器範本，會使用下表中的環境變數來執行 cleartool 指令

下表中的每一個環境變數，並非每一個 ClearCase 或 ClearQuest 配接器範本所必需。

在每一個配接器範本中，必要的環境變數會列在 <template> 區段的 <env> 元素中。

在執行 ClearCase 或 ClearQuest 配接器專案之前，請先提供必要變數的實際值，或者接受預設值。請編輯指派給配接器之環境中的變數值。

變數必須顯示於專案的環境中。這些變數不會於步驟層次上作用。

配接器範本位於下列目錄中：

bfinstall\interface（執行 Microsoft Windows 作業系統的電腦）

bfinstall/Platform/interface（執行 UNIX 或 Linux 作業系統的電腦）

表 18. ClearCase 和 ClearQuest 的環境變數

環境變數名稱	替代變數	說明
VIEW	value=my_adaptor_view	將此變數設為您要搭配配接器使用的 ClearCase 視圖名稱。
VOB_PATH	value=\c_vob	將這個值設為元件 VOB 的名稱及其子目錄（後者為選用）。
PROJECT_VOB	value=\ProjectVob	當您使用 ByBaseline 配接器時，請將此變數設為 VOB 專案的名稱（只能與 UCM ClearCase 搭配使用）。
CCSERVER	value=BFServerName	將此變數設為安裝和執行 ClearCase 用戶端的 Build Forge 伺服器名稱。
CurDate	value=.date %d-%b-%y.%H:%M:%S	使用 .date 指令產生 ClearCase 預期格式的日期，來提供現行日期給配接器。請勿變更此值。

表 18. ClearCase 和 ClearQuest 的環境變數 (繼續)

環境變數名稱	替代變數	說明
LAST_RUN	value=1-Jan-05.00:00:00	若為 ByDate 配接器，系統會使用此值來判斷是否發生任何變更；此值是前次順利執行的日期。您可以在測試配接器時操作此值，挑選在某些變更發生之前的日期，以強迫配接器執行。如果配接器容許繼續執行，它會自動將此值更新為現行日期。預設值為 1-Jan-05.00:00:00。
LABEL	value=BUILD_1	如果是使用 ByLabel 配接器，當您根據標籤以配接器產生差異時（使用 ByLabel 配接器），系統會將此值作為標籤使用。
BASELINE	value=BUILD_1	如果是使用 ByBaseline 配接器，當您根據基準線以配接器產生差異時，系統會將此值作為基準線使用。
ACTIVITIES	value=SAMPL0001@\ProjectVob	如果是使用 ClearQuestClearCaseByActivity 配接器，即為以空格區隔的活動 ID 集。

Perforce 環境變數

將下列 Perforce 環境變數新增至指派給 Perforce 配接器的 Build Forge 環境群組：

- P4USER
- P4PASSWD

Build Forge 需要有效的使用者名稱和密碼，才能存取 Perforce 伺服器。在步驟日誌中，Perforce 使用者名稱和密碼會以純文字撰寫。

您無法使用環境變數的「指派隱藏」內容，來加密 Perforce 鑑別資訊。

配接器範本結構

這個主題說明 Build Forge 配接器範本中的一般 XML 結構或元素階層。

配接器範本是由下列區段元素所構成：<template>、<interface>、<command> 和 <bomformat>。這些區段各包含一些子元素。

如需元素說明，請參閱第 463 頁的『配接器 XML 參照』。

```

<PROJECT_INTERFACE>
<template>
<env/>
</template>

<interface>
<setenv/>
<run/>
<ontempenv>
<step/>
</ontempenv>
<onproject>
<notify/>
</onproject>
</interface>

<command>
<execute> or <command>
command line
</execute> or </command>
<resultsblock>
<match>
<bom>

```

```

<field/>
</bom>
<adduser/>
<setenv/>
<run/>
</match>
</resultsblock>
</command>

<bomformat>
  <section>
    <field/>
  </section>
</bomformat>
</PROJECT_INTERFACE>

```

配接器 XML 參照

本節按字母順序列出配接器 XML 元素。它是配接器 XML 語言中所用元素的參照。說明中包含了一些範例和虛擬碼。

adduser

在 <interface> 元素或 <match> 元素中使用 <adduser> 元素，以根據變更指令的輸出，將使用者新增至暫時群組，讓配接器能夠傳送通知給造成變更的使用者。如果使用者已是群組成員，系統就不會將該使用者新增至群組，以避免有多重通知。<adduser> 元素是空元素。group 屬性會指定配接器邏輯執行期間所建立的暫時群組；您必須在 <notify> 元素中參照相同的群組，以傳送實際的通知。

```
<adduser group="MyChangers" user="$4"/>
```

請使用 condition="function" 屬性來控制 <adduser> 元素是否會將造成變更的使用者新增到暫時存取群組。function 的值是一個會求值為 true 或 false 的表示式。如果表示式的求值結果是 true，則會將使用者新增至暫時群組；否則就不會新增使用者。請參閱第 333 頁的『條件函數』。

系統支援下列電子郵件位址格式。使用者名稱 是電子郵件使用者名稱。名稱 是任意字串，通常是某形式的給定名稱。

```

使用者名稱@host.com
名稱 使用者名稱@host.com
名稱 "<使用者名稱@host.com>"
名稱 <使用者名稱@host.com>
名稱 "使用者名稱@host.com"

```

bom

<bom> 元素可定義要為工作記載於「資料清單 (BOM)」的資訊；其應包含在 <match> 元素中。<bom> 元素必須指定 BOM 中的種類及區段，並定義要將 <match> 元素所收集的哪些編號變數 (\$1...\$n) 轉換成 BOM 資料的欄位。

```

<bom category="Source" section="changesets" >
  <field name="Change" text="$1"/>
  <field name="Date" text="$2"/>
  <field name="User" text="$4"/>
</bom>

```

請使用 condition="function" 屬性來控制是否將 <bom> 元素寫入 BOM 報告。function 的值是一個會求值為 true 或 false 的表示式。如果表示式的求值為 true，就會將 <bom> 元素中的資訊寫入 BOM；否則不會寫入。請參閱第 333 頁的『條件函數』。

bomformat

`<bomformat>` 元素用來定義如何顯示先前 `<bom>` 元素中所收集的資料。它採用 `category` 屬性來指定 BOM 種類的邏輯名稱，並採用 `title` 屬性來指定種類的顯示名稱。包含 `<field>` 元素之 `<section>` 元素的結構定義了佈置，範例如下所示：

```
<bomformat category="Detail" title="Change Details">
  <section name="descriptions" parent="section name" expandable="yes">
    <field name="Description" title="Change Description"/>
  </section>
  <section name="diff">
    <field name="Diff" title="Differences"/>
  </section>
```

command

配接器 XML 檔可包含數個 `<command>` 元素；每個元素定義一個可由 `<run>` 元素在 `<interface>` 元素中參照的具名指令。`<command>` 元素是在 `<interface>` 元素之外指定，因此，XML 檔案內的多重介面可重複使用相同的指令。

指令如果要呼叫其他指令，可以在 `<resultsblock>` 元素的 `<match>` 元素中，內嵌 `<run>` 指令。

`<command>` 元素會包含 `<execute>` 和 `<resultsblock>` 元素的結構，如下所示：

```
<command name="p4_changes">
  <execute>
    command line
  </execute>
  <resultsblock>
    Has its own structure.
  </resultsblock>
</command>
```

或者，您可以在區塊中以 `<integrate>` 元素取代 `<execute>` 元素。

使用 `mode` 屬性來定義 `<command>` 元素的模式。會使用 `mode` 屬性 `exec`，以觸發伺服器啟動並立即處理指令。

註：在 Rational® Build Forge® 8.0 中，僅支援預設 `exec` mode。

env

`<env>` 元素可供元素內部用來定義環境變數（含起始值），而這些環境變數可複製到環境中，搭配專案鏈結一起使用。每個 `<env>` 元素都應包含 `name` 及 `value` 內容。`value` 提供變數的起始值。

```
<env name="FILESPEC" value="//depot..." />
```

execute

請在 `<command>` 元素內使用 `<execute>` 元素來指定指令。此元素的內容是要傳送至配接器所使用之伺服器的一或多行文字。在 `<execute>` 元素中不能使用點指令。當 `<run>` 元素呼叫 `<command>` 元素時，系統會以呼叫 `<run>` 元素中指定的參數，取代 `<execute>` 元素內容中的所有位置參數。第一個參數會取代 `<execute>` 元素內容中的 `$1` 參數，第二個參數取代 `$2` 參數，以此類推。

請使用 `condition="function"` 屬性來控制 `<execute>` 元素中指令的執行情況。`function` 的值是一個會求值為 `true` 或 `false` 的表示式。如果表示式求值為 `true`，系統就會執行指令；如果表示式求值為 `false`，則不會執行指令。請參閱第 333 頁的『條件函數』。


```
<execute>
p4 changes -s submitted -t -i $2@$1,@now
</execute>
```

field

<field>元素用於 <bom> 或 <section> 元素內，以用來指定欄位。

在 <bom> 元素中使用時，請指定名稱與文字；文字屬性定義用來將資料移入欄位的變數。

在 <bomformat> <section> 元素中使用時，請指定名稱及職稱。名稱指定邏輯名稱，而標題則用來顯示。如果 <section> 中有多個欄位，則併入 order 屬性。

```
<section name="changesets">
  <field order="1" name="Change" title="Change ID"/>
  <field order="2" name="Date" title="Date"/>
  <field order="3" name="Time" title="Time"/>
  <field order="4" name="User" title="User ID"/>
  <field order="5" name="Client" title="Client"/>
  <field order="6" name="Comment" title="Comment"/>
</section>
```

請使用 condition="function" 屬性來控制是否將 <field> 元素寫入 BOM 報告。function 的值是一個會求值為 true 或 false 的表示式。如果表示式的求值為 true，就會將 <field> 元素中的資訊寫入 BOM。如果表示式的求值結果是 false，就不會將資訊寫入 BOM 中。請參閱第 333 頁的『條件函數』。

請使用 template 屬性來定義 <field> 元素的文字格式。例如，如果文字是字串，範本值可能是 "Hello \$VALUE"。當欄位寫入 BOM 報告時，\$VALUE 會取代為欄位文字。

integrate

<integrate> 元素與 <execute> 元素類似。您可以使用 <integrate> 元素來取代 <execute> 元素。類似於 <execute> 元素，<integrate> 元素也會指定要執行的指令行。其差異性如下所示：

- 指令行在「管理主控台」系統上執行，並非在執行配接器的伺服器上執行。
- 指令行將 \integration 目錄（安裝目錄的子目錄）當成它的現行目錄使用。

如果要執行位於「管理主控台」電腦（特別是在 \integration 目錄）中的應用程式或 Script，<integrate> 元素非常有用。

當 <run> 元素呼叫包含 <integrate> 元素的 <command> 元素時，系統會以呼叫 <run> 元素中所指定的參數，取代 <integrate> 元素中的任何位置變數。第一個參數會取代 <integrate> 元素中的 \$1，第二個參數會取代 \$2，依此類推。

使用 <execute> 元素時，無法在 <integrate> 元素中使用點指令。

下列範例（來自 IBM Rational ClearQuest 配接器）會執行 CQperl 指令（執行 Perl 程式碼的 ClearQuest 程式），並提供它 \integration 目錄中 Perl Script 的名稱，進而將資料傳送到 ClearQuest。此範例假設 ClearQuest 已安裝在「管理主控台」系統上。

```
<integrate>
cqperl bfcqresolve.pl $2 Fixed "Fixed in build $BF_TAG"
</integrate>
```


interface

`<interface>` 元素是一個作為配接器進入點的儲存器。其中的元素是定義配接器的程式邏輯。它包含 `<setenv>`、`<run>`、`<ontempenv>`、`<onproject>` 和 `<adduser>` 元素。

配接器範本可有多個進入點。如果您要建立具有多個進入點的配接器範本，請執行下列動作：

- 使用 `name` 屬性來識別每一個進入點。
- 如果在呼叫配接器範本時沒有指定名稱，請使用 `default` 屬性來識別要執行的進入點。

如果要指定進入點，請在對範本的 `.source` 呼叫中，使用進入點名稱作為參數。

範例：建立一個名為 `MyAdaptorTemplate` 的配接器範本。在其中放置下列程式碼，以定義一個具名的進入點，並作為預設值：

```
<interface name="By Date" default="true">
</interface>
```

如果要使用名稱來呼叫這個介面，請在專案步驟中使用下列指令：

```
.source MyAdaptorTemplate "By Date"
```

屬性：

name 選用。這個介面的名稱。當呼叫配接器時，如果使用 `name` 作為參數，就會使用其名稱符合該參數的介面。

default

選用。可設為「是」或「否」。若為「是」，則當呼叫配接器時，如果沒有使用 `name` 參數，就會使用該介面。

match

`<match>` 元素用於 `<resultsblock>` 元素中，以處理輸出行。`<match>` 元素會使用定義相符資料行的 `pattern` 屬性。`pattern` 是正規表示式（在 Build Forge 8.0 中，是 Java 引擎的 Java 正規表示式，在 Build Forge 8.0 及所有 7.1.x 中，是 Perl 引擎的 Perl 正規表示式）。

相符型樣可包括弧表示式，儲存在 `$1...$n` 變數中。

```
<match pattern="^Change (\d+) on (.*) (.*) by (.*)@(.*?) '(.*)'$">
```

`<match>` 元素會使用 `<adduser>`、`<setenv>`、`<bom>` 和 `<field>` 作為子元素。請參閱第 468 頁的『`resultsblock`』的參照資訊，以查看更廣泛的範例。

notify

`<notify>` 元素會指定電子郵件和發佈清單。它通常是用於 `<onproject>` 元素中，會根據執行結果來指定通知。

屬性：

group 電子郵件的收件者。這是一個定義在配接器中的使用者群組。

message

電子郵件內文的內容。您將指定文字。

subject

電子郵件的主旨。您將指定文字。

onproject

<onproject> 元素會定義系統在執行專案步驟之後所執行的通知動作。該元素會選擇一個必要的 `result` 屬性，指定這些動作是針對通過還是失敗的工作來執行。配接器 XML 檔通常包含兩個 <onproject> 元素：一個用於通過的案例，一個用於失敗的案例。下列範例顯示一對 <onproject> 元素，這對元素會根據專案是通過還是失敗，使用 `notify` 元素來傳送不同的訊息：

```
<onproject result="fail">
  <notify group="MyChangers"
    subject="Run $BF_TAG ($CurDate) Failed." message="$Changing$Changes"/>
</onproject>

<onproject result="pass">
  <notify group="MyChangers"
    subject="Run $BF_TAG ($CurDate) Passed." message="$Changing$Changes"/>
</onproject>
```

ontempenv

<ontempenv> 元素會用於 <interface> 元素內，其行為類似 if-then 陳述式。請使用這個元素，將通過或失敗值傳給專案；如果通過，就表示系統應該繼續進行，並且執行專案剩下的部分，如果是失敗，則表示就此打住。其目的通常是指出介面是否有發現值得重新執行專案的相關變更。

當系統執行所有在 <run> 元素中所指定的指令之後，就會處理 <ontempenv> 元素。請使用此元素的 `name` 屬性來指定暫時環境變數，並使用 `state` 屬性來指定值。

<ontempenv> 會包住 <step> 元素，且只有在執行 <run> 元素指令之後，暫時環境變數 `name` 和 `state` 仍然存在時，才會執行。

```
<ontempenv name="Changes" state="empty">
  <step result="FAIL"/>
</ontempenv>
```

PROJECT_INTERFACE

<PROJECT_INTERFACE> 元素會將其他所有的標籤，包在配接器範本中。它會採用指出配接器類型的 `IFTYPE` 屬性。有效的類型包含 `Source`、`Test` 及 `Defect`。

```
<PROJECT_INTERFACE IFTYPE="Source">
...all other elements...
</PROJECT_INTERFACE>
```

relate

<relate> 元素會指定構件和使用者之間的關係。它將與定義在專案中且類型為「通知變更器」的日誌過濾器一起使用，以及在呼叫配接器之步驟的前一個步驟中的 `Result` 屬性中使用。「通知變更器」過濾器通常會指定一個預期的日誌行，指出成功或失敗。當執行步驟時，會將過濾器與 <relate> 元素中指定作為構件的文字相比較。如果找到相符項，就會將電子郵件傳送給和該構件相關聯的使用者。

另請參閱第 302 頁的『過濾器動作』中有關「通知變更器」過濾器的說明。

屬性：

artifact

用來和日誌過濾器所要搜尋的文字相比對的文字。

user 和構件相關聯的使用者。如果找到相符項，系統就會將電子郵件傳給這個使用者。

text 當日誌過濾器 and 這項關係之間存在相符項時，要記載至 BOM 的文字。

resultsblock

`<resultsblock>` 元素定義系統如何處理從相關 `<execute>` 元素執行之指令行的結果。
`<resultsblock>` 元素僅用在 `<command>` 元素中。可對 `<resultsblock>` 元素建立巢狀，以放置於分割區結果內。

`<resultsblock>` 元素可以有選用的 `beginpattern` 和 `endpattern` 屬性，這兩個屬性使用正規表示式（在 Build Forge 8.0 中，是 Java 引擎的 Java 正規表示式，在 Build Forge 8.0 及所有 7.1.x 中，是 Perl 引擎的 Perl 正規表示式）來定義要處理的輸出行範圍。然後，您可以使用不同的 `<resultsblock>` 元素來處理不同範圍。下列虛擬碼顯示 `<resultsblock>` 元素的結構。

```
<resultsblock startpattern="" endpattern="" >
  <match>
    <bom>
      <field/>
    </bom>
    <adduser/>
    <setenv/>
    <run/> (The <run> element can be used to run commands within other commands)
  </match>
  <setenv/>
</resultsblock>
```

下例顯示 `<resultsblock>`、`<match>` 和 `<bom>` 元素搭配使用的方式：

```
<resultsblock
beginpattern="^Change (\d+) by (.*)@(.*) on (.*) (.*)$"
endpattern="^Differences ...$">
  <match pattern="(?!^(:?!Differences ...).)*$.?)">
    <bom category="Detail" section="descriptions">
      <field name="Description" text="$_" />
    </bom>
  </match>
</resultsblock>
```

run

`<run>` 元素在 `<interface>` 元素內用來指定要執行的具名指令。稍後在相同的 XML 檔中定義此指令。`<run>` 元素是一個空元素。

屬性：

condition

選用，形式如下所示：`condition="function"`。 `function` 的值是一個會求值為 `true` 或 `false` 的表示式。如果求值為 `true`，指令就會執行。如果求值為 `false`，就不會執行。請參閱第 333 頁的『條件函數』。

command

（必要）。指定定義要執行之指令的名稱。指令是在 `<command>` 元素中命名和定義。

dir （必要）。指定要在其中執行指令的目錄。`dir` 會解譯成指令執行所在之伺服器資源中所設定的路徑副檔名。

mode 選用。指定 run 指令的執行模式。會使用 mode 屬性 *exec*，以觸發伺服器啟動並立即處理指令。

註：在 Rational® Build Forge® 8.0 中，僅支援預設 *exec* mode。

params

(必要)。指定要傳遞給指令的參數。請使用空格來區隔各參數。

server (必要)。指定指令執行所在的伺服器資源。如果設為空值，則指令會在執行配接器之步驟所用的伺服器上執行。使用 *server=""*，表示將伺服器設為空值。

timeout

(必要)。它指定在經過多少秒之後，指令即會逾時。

顯示所需屬性的範例：

```
<run command="UpdateEnv" params="" server="" dir="/" timeout="360"/>
```

範例：

```
<run command="p4_changes" params="$LAST_RUN $FILESPEC $LAST_VER"  
server="$P4CLIENT" dir="/" timeout="360"/>
```

section

<section> 元素用來定義如何顯示 BOM 種類中的某一部分。它使用 name 屬性。<section> 元素只能用在 <bomformat> 元素內。

setenv

請使用 <setenv> 元素來起始設定 <interface> 或 <match> 元素中的環境變數值。<setenv> 元素不會包含其他元素。

該元素的使用方法有三種：

- 當您指定群組名稱時，其運作方式和 .set 指令類似。它會將變數值設定在資料庫的主要記錄中（而不是在現行步驟使用的副本中）。正在現行步驟中執行的配接器不會看到這項變更。您無法以這種方式來建立新變數。
- 當您沒有指定群組名稱時，其運作方式和 .bset 指令類似。它會將變數值設定在執行中工作的環境中。工作中的所有步驟皆可使用這項變更。您使用採取這種方式來建立新變數。
- 當您沒有指定群組名稱，並且指定臨時變數 (type="temp")，則會設定一個僅作為配接器邏輯用的臨時變數。此變數在配接器步驟執行之後並不會持續保存。您使用採取這種方式來建立新變數。

屬性：

condition

選用，形式如下所示：condition="*function*"。function 的值是一個會求值為 true 或 false 的表示式。如果求值為 true，指令就會執行。如果求值為 false，就不會執行。請參閱第 333 頁的『條件函數』。

eval 選用。設為 True 或 False。若為 true，則配接器會試圖對 value 屬性表示式求值，並儲存結果。

group 選用。指定要在其中定義變數的 Build Forge 環境。當您指定環境名稱時，必須參照指定環境中的現有變數。

如果您指定 [ADAPTOR] 作為值，則會在執行時期設定值。它會設為呼叫配接器之步驟或配接器鏈結的環境。

Build Forge 容許同名的變數存在於多個環境中。環境繼承與環境併入的優先順序可能影響在執行時期如何決定將值指派給變數。

name （必要）。指定要設定之變數的名稱。值可以是一個變數。在這種情況下，在執行時期之前並不會設定變數名稱。

type 選用。指定變數的設定方法。其值可為下列之一：

- **append text**：將指定的值放在任何現有值之後。如果指定選用的文字，會將該文字放在兩值之間。
- **once**：變數必須尚未設定才能設定。
- **prepend 文字**：將值放在任何現有值之前。
- **temp**：變數只能設定在配接器的環境定義中。如果指定選用的文字，會將該文字放在兩值之間。請參閱下列範例。

value （必要）。這是設定在變數中的值。如果同時指定了 **eval** 屬性，則可以是所要求值的表示式。求值的結果會儲存成值。

範例：

下列範例是對 **value** 屬性求值的表示式，並將結果儲存在 **LAST_VER** 變數中。它將設為 **LAST_VER** 或 **\$1** 變數值中的較大者。

```
<setenv group="Adaptor" name="LAST_VER"
value="$LAST_VER>$1?$LAST_VER:$1" eval="true" />
```

下列範例是在附加到 **Changes** 變數後面的資料前面，插入一個換行字元 (**\n**)：

```
<setenv name="Changes" value="$4 - $1 - $6" type="temp append\n" />
```

下列範例是在附加到 **INFOPATH** 變數前面的值的後面插入一個冒號：

```
<setenv name="INFOPATH" value="/usr/local" type="temp prepend:" />
```

step

<step> 元素只使用於 **<ontempenv>** 元素內。它指定特殊配接器步驟的結果。它是空元素。下列範例顯示 **<step>** 元素的兩種形式。

```
<step result="FAIL"/>
```

```
<step result="pass"/>
```

第 27 章 Rational Build Forge 與 Rational ClearCase

BuildForge 可以搭配 Rational ClearCase 使用來管理原始碼。

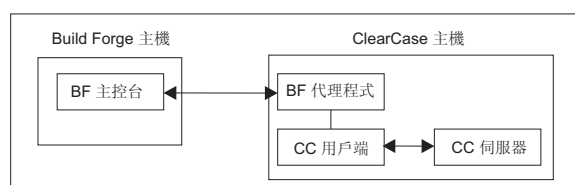
這個主題說明整合 Rational ClearCase 時額外的設定需求，並彙總所提供之範例配接器範本的性質。

ClearCase 配接器範本範例提供了分析基準線變更的方法。一般而言，變更分析是用於省略建置：如果基準線元件未變更，則不會重建

ClearCase 配接器額外的設定需求

請參閱第 438 頁的『配接器需求』以取得一般需求。除了滿足配接器的一般需求之外，請執行下列動作：

1. 在可以連接至 ClearCase 伺服器的主機上安裝 Build Forge 代理程式。
2. 在代理主機上安裝 ClearCase 完整用戶端。
3. 設定代理程式的環境，以便透過 ClearCase 用戶端來執行指令。
4. 決定所需 ClearCase 視圖的實作方式、啟動方式及啟動時間。提供的範本假設使用動態視圖，並且包含在執行 cleartool 指令時啟動該視圖。在代理主機上安裝 ClearCase 完整用戶端。



ClearCase 視圖

ClearCase 採用視圖作為供您存取資料的方法。您可以使用兩種類型的視圖：

- 動態視圖：動態視圖可讓您查看指定 VOB 中最新版本的元素，而無需將其複製到本端系統。必須啟動視圖。您還會指定要在視圖中裝載的 VOB。
- Snapshot 視圖：Snapshot 視圖是儲存庫副本。您可以指定要使用的物件版本。必須手動更新 Snapshot 視圖，以便取得已移入的最新項目。

動態視圖

只要您選擇查看的視圖環境定義中發生任何變更，動態視圖便會顯示持續更新的工作區。只有您移出的元素才會複製到工作區，且您必須將其移入，才能供其他使用者使用。使用動態視圖時，您會進行下列三項指定，以存取所需的資料：

- 視圖：定義要呈現的版本清單（透過配置規格）。必須使用指令來啟動視圖。在本端磁碟上，視圖表示為內容的根目錄。它也稱為起始視圖。常用的目錄如下所示：
 - Windows：磁碟機 M:，共用磁碟機
 - UNIX 或 Linux：/view，已裝載的檔案系統

- 視圖環境定義：透過視圖標籤來定義要使用的內容目錄。視圖標籤在 ClearCase 中定義。啟動視圖後，您即可使用此目錄。
- VOB（已版本化的物件庫）：定義視圖環境定義的子目錄。在 ClearCase 中可以定義 VOB。您會在視圖環境定義中明確裝載 VOB。此子目錄包含已版本化的構件（原始檔及其他構件）。

本端機器上的 VOB 路徑由上述三個項目組成：

- Windows 系統的範例路徑（使用預設值）：`M:\view_tag\vob_name`
- UNIX 及 Linux 系統的範例路徑（使用預設值）：`/view/view_tag/vob_name`

Snapshot 視圖

使用 Snapshot 視圖時，您會指定視圖環境定義，同時還必須載入資料。然後即會擁有您指定所有資料的本端副本。必須重新載入 Snapshot 視圖，以便取得自前次載入資料後移入的變更。

支援動態視圖的 Rational Build Forge 特性

您可以使用特殊環境變數，從 Build Forge 工作中定義及啟動動態視圖。

CLEARCASE_VIEW

啟動指定的 ClearCase 視圖。這個變數中指定的視圖必須存在，而且使用這個變數的步驟必須設為 "absolute"。在執行 Microsoft Windows 的系統中，此變數必須搭配 `bfagent.conf` 中代理程式的 `cc_suppress_server_root` 參數來使用。

_CLEARCASE_VIEWS

指定指令執行之前要啟動的 ClearCase 視圖清單。將此值設為以逗點區隔的視圖清單；例如，"`View1,View2,View3`"。

_CLEARCASE_VOBS

指定指令執行之前要裝載的 ClearCase VOB 清單。將此值設為以逗點區隔的 VOB 清單；例如，"`\Vob1,\Vob2,\Vob3`"。

您可能還需要使用代理程式參數，來管理代理主機上的視圖。如果您搭配 ClearCase 使用的代理程式是在 Windows 系統上執行，則必須設定代理程式參數。當您要透過此代理程式使用的起始視圖，與工作所設定的起始視圖不相同時，這些代理程式參數也會很有用。如果您針對多個代理程式使用此工作，並且需要在每個代理主機系統上進行不同的指定，則會發生此情況。

ccviewroot *root-path*

指定此主機的預設起始視圖。請參閱 ClearCase 文件，以取得 `init` 的詳細資訊。內部預設值如下所示：

- Windows 系統：`ccviewroot M:`
- UNIX 或 Linux 系統：`ccviewroot /view`

cc_suppress_server_root

若已設定，請檢視路徑是 `ccviewroot` 所設定的路徑。若未設定，則在伺服器定義中所設定的路徑就會附加至 `ccviewroot` 所設定的路徑。此設定不需要值。如果已在 `bfagent.conf` 中出現，則表示已設定。

ClearCase 配接器範本範例

所提供的配接器範本範例如下所示。

ClearCaseBaseline

1. 在 ClearCase 視圖中掃描目錄。
2. 將 ClearCase 報告的分支與版本資訊寫入 BOM 報告中。

使用的變數：

- INT_STREAM
- VIEW
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT

ClearCaseByBaselineActivities

1. 從 ClearCase 視圖的內容建立新的基準線。
2. 比較新的基準線與配接器前次執行時的基準線，以找出變更活動。
3. 針對每次的變更活動，將下列資訊寫入 BOM 報告：活動、已變更的檔案、使用者、日期、註解與版本。
4. 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

使用的變數：

- CurDate
- LAST_RUN
- BASELINE
- VIEW
- VOB_PATH
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT

ClearCaseByBaselineVersions

1. 從 ClearCase 視圖的內容建立新的基準線。
2. 比較新的基準線與配接器前次執行時的基準線，以找出變更過的檔案。
3. 針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、版本、日期、使用者與註解。
4. 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

使用的變數：

- CurDate
- LAST_RUN
- LABEL
- BASELINE

- VIEW
- VOB_PATH
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearCaseByDate

1. 在 ClearCase 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。
2. 針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、版本、日期、使用者與註解。
3. 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

使用的變數：

- CurDate
- LAST_RUN
- LABEL
- BASELINE
- VIEW
- VOB_PATH
- PROJECT_VOB
- CCSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearCaseByLabel

1. 建立新的標籤，並套用至 ClearCase 視圖的內容。
2. 比較新的標籤與配接器前次執行時的標籤，以找出變更過的檔案。
3. 針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、版本、日期、使用者與註解。
4. 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

使用的變數：

- CurDate
- LAST_RUN
- LABEL
- BASELINE
- VIEW
- VOB_PATH
- PROJECT_VOB
- CCSERVER

- UNIXCLIENT
- _CHAR_NATIVE

ClearCase 配接器變數

這份表格是配接器範本之變數清單的參照。

環境變數名稱	說明
BASLINE	如果是使用 ByBaseline 配接器，當您根據基準線以配接器產生差異時，系統會將此值作為基準線使用。
CCSERVER	將此變數設為已安裝 ClearCase 用戶端和 Build Forge 代理程式的主機名稱。
CurDate	使用 .date 指令產生 ClearCase 預期格式的日期，來提供現行日期給配接器。請勿變更此值。
LABEL	如果是使用 ByLabel 配接器，當您根據標籤以配接器產生差異時（使用 ByLabel 配接器），系統會將此值作為標籤使用。
LAST_RUN	若為 ByDate 配接器，系統會使用此值來判斷是否發生任何變更；此值是前次順利執行的日期。您可以在測試配接器時操作此值，挑選在某些變更發生之前的日期，以強迫配接器執行。如果配接器容許繼續執行，它會自動將此值更新為現行日期。預設值為 1-Jan-05.00:00:00。
PROJECT_VOB	當您使用 ByBaseline 配接器時，請將此變數設為 VOB 專案的名稱（只能與 UCM ClearCase 搭配使用）。範例：\ProjectVob
UNIXCLIENT	用來設定平台專用的資訊。如果用戶端是在 Windows 上執行，請設為 0。如果用戶端是在 UNIX 或 Linux 上執行，請設為 1。
VIEW	將此變數設為您要搭配配接器使用的 ClearCase 視圖名稱。
VOB_PATH	將這個值設為元件 VOB 的名稱及其子目錄（後者為選用）。請使用逗點區隔的清單來指定多個名稱。
_CHAR_NATIVE	供內部使用且一律設為 1。

下列觸發變數也可用來啟動視圖並裝載 VOB。不過，其與配接器變數中設定的視圖和 VOB 無關。

- CLEARCASE_VIEW
- _CLEARCASE_VIEWS
- _CLEARCASE_VOBS

請參閱第 276 頁的『觸發變數參照』。

第 28 章 Rational Build Forge 與 Rational ClearQuest

BuildForge 可以搭配 Rational ClearQuest 使用來管理原始碼。

Build Forge 與 ClearQuest 有兩項整合：

- 根據工作狀態自動建立或更新建置記錄。當專案環境中設定了所需的環境變數時，即會自動啟動這項功能。
- ClearQuest 配接器

這些功能彼此完全獨立。特別是，配接器會與針對其建立的環境相關聯。該環境中的變數與設定來啟動自動建置記錄的變數無關。

設定自動產生建置記錄

系統可以在 IBM Rational® ClearQuest® 資料庫中自動建立建置記錄，包含與日誌資料的鏈結。另外，當工作通過時，系統就能更新 ClearQuest 資料庫，以註明工作已經完成，並記錄結束時間以及所完成的步驟摘要。這項特性需要 Rational ClearQuest 7.0 版或更新版本。

當您配置專案以更新 ClearQuest 資料庫時，系統會執行建立或更新建置記錄，如下所示：

工作開始

當系統啟動工作時，系統就會建立 ClearQuest 建置記錄。建置記錄會是「已提交」狀態，並且包含工作日誌 URL、開始時間、版次名稱和 ID，以及指出「建置 XYZ 已啟動」的日誌項目。如果來源控制配接器取消工作（例如，因為找不到來源變更），則不會建立 ClearQuest 建置記錄。

註：如果專案與其他專案鏈結，則新專案會取得專屬的唯一 ClearQuest 建置 ID。

工作通過/失敗

當工作通過或失敗時，系統會將 ClearQuest 內的建置狀態變更為「已完成」或「失敗」，再設定建置的結束時間，然後將工作步驟的摘要儲存在 ClearQuest 建置日誌中。摘要包含每個步驟的名稱、結果狀態與伺服器。

工作重新啟動

當工作重新啟動時，系統會將 ClearQuest 內的建置狀態變更為「已提交」，並建立 ClearQuest 建置日誌項目，指出「建置 XYZ 已重新啟動。」

您會透過特殊環境變數來配置自動更新建置記錄。若要將專案鏈結至 ClearQuest 資料庫，請確定下表內的變數已併入專案的環境中。

註：這些變數必須呈現在專案環境中。光是將它們新增至步驟是不夠的。但是，您可以使用設定為「併入」類型的變數，經由另一個環境來併入這些變數。此外，由於 CQ_RELEASE_NAME 值是唯一可能會依專案而有所不同的值，您可能會想建立一個內含其他變數的環境，然後使用「併入」類型的變數將該環境併入在專案環境中，而在專案環境中，您也可以將 CQ_RELEASE_NAME 指定為專案特有的環境變數。

爲了啓動自動更新 Build Forge 工作的建置記錄，必須爲專案設定下列環境變數。這些變數不會於步驟層次上作用。

變數	說明
CQ_DBNAME	(必要)。您要更新的 ClearQuest 資料庫名稱。
CQ_DBSET	ClearQuest 資料庫設定值。非必要。預設值爲空白。
CQ_INTERACTION	如果您的專案環境已定義了正確的環境變數，以允許建立 ClearQuest 建置記錄，但是您並不要建立建置記錄，請將此變數設爲 OFF 以停止建立建置記錄。 如果要啓用建置記錄的建立，請將這個環境變數設爲 ON。 註： 如果您正使用其中一個 ClearQuest 配接器，請將這個環境變數設爲 OFF。配接器會與建置記錄直接互動。
CQ_PASSWORD	(必要)。登入 ClearQuest 資料庫時所要使用的密碼。不是必要的；預設爲空白。
CQ_RELEASE_NAME	(必要)。在您要更新的 ClearQuest 資料庫中的版次名稱。
CQ_USER	(必要)。登入 ClearQuest 資料庫時所要使用的使用者名稱。

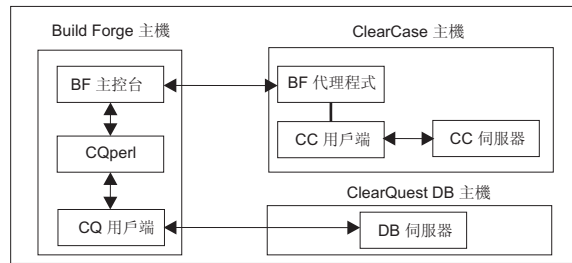
ClearQuest 配接器額外的設定需求

ClearQuest 配接器範本範例，提供在 ClearQuest 中掃描 ClearCase 及更新建置記錄的方法。這通常關係到在 Build Forge 中執行的建置是成功還是失敗。請參閱第 438 頁的『配接器需求』以取得一般需求。此外，還需要完成下列配置。

- 掃描原始檔所需的 ClearCase 存取權。在工作執行期間，配接器會透過代理程式及 ClearCase 用戶端來執行 cleartool 指令。配接器將使用 ClearQuest Perl API (cqperl) 執行這些指令。
- 更新建置記錄所需的 ClearQuest 存取權。在工作執行期間，配接器會直接在主控台主機上執行 cqperl Script。這些 Script 將由主控台主機上安裝的 Cqperl 公用程式解譯，並透過主控台主機上安裝的 ClearQuest 用戶端來執行。

請執行下列動作：

1. 在可以連接至 ClearCase 伺服器的主機上安裝 Build Forge 代理程式。
2. 在代理主機上安裝 ClearCase 完整用戶端。
3. 設定代理程式的環境，以便透過 ClearCase 用戶端來執行指令。
4. 在 Build Forge 主控台主機上安裝 ClearQuest 完整用戶端。
5. 將 cqperl (ClearQuest Perl API) 目錄新增至系統路徑。
6. 定義連線，Build Forge 主機上的 ClearQuest 用戶端使用此連線來存取 ClearQuest 資料庫。請在 ClearQuest 用戶端主機上執行這些動作。
 - a. 使用 cqreg 指令新增資料庫集 (cqreg add_dbset)。
 - b. 使用 CQ 的「維護工具」，設定通往 ClearQuest 資料庫的連線。
7. 決定所需 ClearCase 視圖的實作方式、啓動方式及啓動時間。



您不需要安裝代理程式。ClearQuest 配接器會透過用戶端，利用 ClearQuest Perl API 與 ClearQuest 直接通訊。

重要：ClearQuest 配接器只能在步驟中以點指令來呼叫。它不是來源配接器，因此不能使用配接器鏈結。

ClearQuest 配接器範本範例

所提供的配接器範本範例如下所示。

ClearQuestBaseClearCaseByDate

1. 在 ClearCase 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。
2. 針對每個已變更的檔案，找出可用來識別 ClearQuest 變更 ID 的 CrmRequest 超鏈結屬性。透過新增工作資訊來試圖解析變更 ID，以解析 ClearQuest 中的問題記錄（如果 ClearQuest 狀態容許解析該問題記錄的話）。
3. 針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、問題 ID、問題狀態和 ClearQuest 的所有錯誤。

配接器範本中定義的變數：

- CurDate
- LAST_RUN
- VIEW
- VOB_PATH
- CQ_USER
- CQ_PASSWORD
- BFSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearQuestClearCaseByActivity

1. 找出與 ClearCase 活動清單相關聯的 ClearQuest 問題記錄。
2. 它會針對所找到的每個問題記錄新增工作資訊，以解析 ClearQuest 中的問題記錄（如果 ClearQuest 狀態容許解析該問題記錄的話）。
3. 將下列資訊寫入 BOM 報告：和 ClearCase 活動 ID 及 ClearQuest 問題狀態相關聯的檔案。

配接器範本中定義的變數：

- CurDate
- VIEW

- VOB_PATH
- ACTIVITIES
- CQ_USER
- CQ_PASSWORD
- PROJECT_VOB
- BFSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearQuestUCMClearCaseByDate

1. 在 ClearCase 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。其使用 Rational Unified Change Management (UCM) 來產生結果。
2. 針對每個已變更的檔案，將下列資訊寫入 BOM 報告：檔名、問題 ID、問題狀態和 ClearQuest 的所有錯誤。

配接器範本中定義的變數：

- CurDate
- LAST_RUN
- VIEW
- VOB_PATH
- CQ_USER
- CQ_PASSWORD
- BFSERVER
- UNIXCLIENT
- _CHAR_NATIVE

ClearQuest 配接器變數

這份表格是配接器範本之變數清單的參照。

表 19. 進行 Rational ClearQuest 整合所需的環境變數

變數	說明
ACTIVITIES	如果是使用 ClearQuestClearCaseByActivity 配接器，即為以空格區隔的活動 ID 集。範例：SAMPL0001@\ProjectVob
BFSERVER	將此變數設為 Build Forge 主控台的主機名稱。
CQ_PASSWORD	(必要)。登入 ClearQuest 資料庫時所要使用的密碼。不是必要的；預設為空白。
CQ_USER	(必要)。登入 ClearQuest 資料庫時所要使用的使用者名稱。
CurDate	利用 .date 指令產生日期，來提供目前日期給配接器。請勿變更此值。
LAST_RUN	若為 ByDate 配接器，系統會使用此值來判斷是否發生任何變更；此值是前次順利執行的日期。您可以在測試配接器時操作此值，挑選在某些變更發生之前的日期，以強迫配接器執行。如果配接器容許繼續執行，它會自動將此值更新為現行日期。預設值為 1-Jan-05.00:00:00。

表 19. 進行 Rational ClearQuest 整合所需的環境變數 (繼續)

變數	說明
UNIXCLIENT	用來設定平台專用的資訊。如果用戶端是在 Windows 上執行，請設為 0。如果用戶端是在 UNIX 或 Linux 上執行，請設為 1。
VIEW	將此變數設為您要搭配配接器使用的 ClearCase 視圖名稱。
VOB_PATH	將這個值設為元件 VOB 的名稱及其子目錄（後者為選用）。請使用逗點區隔的清單來指定多個名稱。
_CHAR_NATIVE	供內部使用且一律設為 1。

重新啟動 ClearQuest-integrated 工作

一旦完成了 ClearQuest-integrated，通常無法在 Rational Build Forge 重新啟動它。有一個簡單的暫行解決方法，就是您可以將此工作當成新工作來啟動。

若要啟用重新啟動，您必須使用 ClearQuest 設計程式工具來編輯 ClearQuest 綱目。必須修改「建置」記錄的工作流程，才能將「已完成」的狀態轉移為「提交」。

第 29 章 Rational Build Forge 與 CVS

BuildForge 可以搭配 CVS 使用來管理原始碼。

這個主題說明整合 CVS 時額外的設定需求，並彙總所提供之範例配接器範本的性質。

CVS 配接器範本範例提供了分析基準線變更的方法。一般而言，變更分析是用於省略建置：如果基準線元件未變更，則不會重建

CVS 配接器額外的設定需求

請參閱第 438 頁的『配接器需求』以取得一般需求。CVS 沒有額外的需求。

CVS 配接器範本範例

所提供的配接器範本範例如下所示。

CVSv1Baseline

1. 掃描 Build Forge 代理程式上的 CVS 目錄，尋找已變更的檔案。
2. 將下列資訊寫入 BOM 報告：已變更的檔名、狀態、有效版本、儲存庫版本與固定標籤。在 ClearCase 視圖中掃描目錄。

使用的變數：

- CVSROOT
- MODULE
- CVSCLIENT

CVSv1ByDate

1. 在 CVS 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。
2. 將下列資訊寫入 BOM 報告：變更類型、日期、使用者名稱、版本和檔名。
3. 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

使用的變數：

- LAST_RUN
- CurDate
- CVSROOT
- MODULE
- BRANCH
- CVSCLIENT

CVSv1ByTag

1. 將新的標籤套用至 CVS 模組。
2. 比較剛標示的模組，與在配接器前次執行期間標示之模組間的差異。
3. 將下列資訊寫入 BOM 報告：檔名、修訂、狀態、日期、時間、變更作者及提出註解。

4. 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

使用的變數：

- LAST_RUN
- CurDate
- CVSROOT
- MODULE
- BRANCH
- LAST_TAG
- CVSCLIENT

CVSv2ByDate

1. 在 CVS 視圖中，查詢兩個日期之間的變更。預設日期為現行時間戳記與配接器前次執行的時間戳記。
2. 將下列資訊寫入 BOM 報告：變更類型、日期、使用者名稱、版本和檔名。
3. 針對每個已變更的檔案，將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

使用的變數：

- LAST_RUN
- CurDate
- CVSROOT
- MODULE
- CVSCLIENT

第 30 章 IDE 整合

透過外掛程式，可以將 Rational Build Forge 與整合開發環境 (IDE) 整合。

本節說明如何安裝和使用外掛程式，以便能夠從「整合開發環境 (IDE)」存取「管理主控台」的特性。

關於 IDE 整合

開發人員可以使用整合的整合開發環境 (IDE) 外掛程式，來連接「管理主控台」。

透過提供的外掛程式，開發人員可以直接從其 IDE 中完成下列作業：

- 檢視專案
- 執行工作
- 視察工作結果

其他的功能將因各外掛程式而異。

每一個開發人員在連接主控台時，將使用 Build Forge 可從其使用者清單中辨識的使用者名稱。對專案的存取是以存取群組中的使用者名稱成員資格來控制。對專案中步驟的存取也是以存取群組來控制。步驟可以明確指定一個存取群組。如果沒有指定，則會繼承專案的存取群組。

外掛程式不提供專案和步驟的編輯或刪除功能。

透過 IDE 來存取 Build Forge 的每一個使用者各耗用一份授權，就如同使用者透過瀏覽器用戶端階段作業耗用一份授權一樣。

外掛程式可提供給下列的 IDE：

- Eclipse™
- Rational® Application Developer，這是建置在 Eclipse™ 上的 IBM IDE
- Rational® Team Concert，這是 IBM 在 Jazz .net 中的發行套件

測試專案的特殊變數

當您使用外掛程式執行專案的測試建置時，可以使用一些特殊環境變數，指定在將系統中的檔案複製到伺服器之前和之後所要執行的指令。

所有的指令都在專案目錄下執行：

- 使用 PRECMD 變數，可針對從開發人員電腦複製到執行該建置之伺服器的目錄和檔案執行指令。該指令是在專案步驟之前執行。範例：您可以使用此指令，先將檔案從來源控制系統移出，然後再複製。
- 使用 POSTCMD 變數，可在專案步驟執行之後，對目錄和檔案執行指令。範例：您可以使用這個指令，釋出已移出的虛擬目錄（在採用這類概念的來源控制系統中，例如 Rational ClearCase）。

您執行指令的對象是，在「反映器」外掛程式中標示為「Build Forge 專案構件」的目錄和檔案。這些指令是在遍訪「反映器」的目錄樹狀結構時套用。

註：使用 PRECMD 指令時，目錄樹狀結構是以橫向優先方式向下遍訪，而 POSTCMD 指令則相反。目錄指令和檔案指令是在遍訪期間，尋找適當的時機執行。

_PRISM_DIR_PRECMD

指定一個指令，來執行在遍訪樹狀結構時遇到的目錄。每一個至少包含一個檔案的目錄，都要執行該指令一次。系統會將指令中的第一個 \$1 取代成目錄名稱。

_PRISM_FILE_PRECMD

指定一個指令，來執行在遍訪樹狀結構時所遇到的檔案。每一個檔案都要執行該指令一次。系統會將指令中的第一個 \$1 取代成檔名。

_PRISM_DIR_POSTCMD

指定一個指令，來執行在遍訪樹狀結構時遇到的目錄。每一個至少包含一個檔案的目錄，都要執行該指令一次。系統會將指令中的第一個 \$1 取代成目錄名稱。

_PRISM_FILE_POSTCMD

指定一個指令，來執行在遍訪樹狀結構時所遇到的檔案。每一個檔案都要執行該指令一次。系統會將指令中的第一個 \$1 取代成檔名。

適用於 Eclipse 及 Rational Application Developer 的外掛程式

外掛程式可讓您從 Eclipse™ 及 Rational® Application Developer IDE 中存取「管理主控台」特性。

下列外掛程式可用於 Eclipse 及 Rational Application Developer 環境：

頻率

透過「頻率」外掛程式，開發人員可以完成下列作業：

- 存取一個以上「管理主控台」，以檢視專案
- 啟動工作
- 檢視工作狀態
- 檢視執行中和已完成工作的建置日誌

反映器

「反映器」外掛程式使用本端環境中的檔案來執行工作。這些工作的執行通常是為了測試新的程式碼，以便之後再將程式碼移入來源控制中，供其他開發人員或正式作業建置使用。

Eclipse 外掛程式使用者可以選擇置換專案環境變數的值。當您啟動 Build Forge 專案時，會顯示「工作設定」蹦現畫面。對環境變數的變更只會套用至工作。專案的預設變數值並不會變更。

使用 Eclipse 或 Rational Application Developer 中的外掛程式

安裝外掛程式之後，您可以用下列方式來加以啟動：

- 如果要存取「管理主控台」來啟動工作及檢視專案日誌，請使用「頻率」外掛程式。在您的 IDE 中，選取視窗 > 開啟視景 > 其他。您的 IDE 會顯示含有視景類

型清單的對話框；請選取 **Build Forge** 視景。系統會顯示「主控台瀏覽器」、「建置資訊」及「建置日誌」視窗。用滑鼠右鍵按一下**主控台瀏覽器**，並選取**新建主控台**，以配置與「管理主控台」的連線。如需使用「頻率」的相關資訊，請參閱外掛程式隨附的線上說明。

註：如果您需要配置對 LDAP/Active Directory 網域的存取權，請務必為 LDAP 網域物件使用 **Build Forge** 系統名稱，而不是網域的實際名稱。

- 如果要執行測試建置，請使用反映器外掛程式。在您的 IDE 中，選取您的專案並按一下滑鼠右鍵，以配置「反映器」。從蹦現功能表選取**內容**。在**內容**對話框的內容選項清單中，選取 **Build Forge 專案構件**。透過您的專案所要搭配使用的 **Build Forge** 專案來配置對話框，並選取要上傳至系統的檔案。如需相關資訊，請參閱外掛程式隨附的線上說明。

註：「反映器」外掛程式需要「頻率」外掛程式。

安裝 Eclipse 或 Rational Application Developer 的外掛程式

從 **Build Forge** 伺服器來安裝 IDE 環境的外掛程式。

開始之前

必備條件：

- Eclipse 3.0.2 版（或更新版本）或 Rational Application Developer 7.0 版（或更新版本）
- Java 2 SE 5.0 版
- **Build Forge** 系統必須執行中

關於這項作業

如果要安裝外掛程式，請從 IDE 中完成此程序。

程序

1. 選取**說明 > 軟體更新 > 搜尋並安裝**。
2. 按一下**搜尋要安裝的新特性**圓鈕，然後按**下一步**。系統會顯示**要造訪的更新網站**對話框。
3. 按一下**新建遠端網站**按鈕。系統會顯示**新建遠端網站**對話框。
 - a. 在名稱欄位中輸入『**Build Forge Update Site**』。
 - b. 使用「管理主控台」電腦的主機名稱，在 **URL** 欄位中輸入下列更新網站 URL：`http://<console_host_name>/prism/eclipse/updateSite/site.xml`。
 - 如果執行 Eclipse 所在的系統與執行 **Build Forge** 所在的系統相同，您可以使用 `localhost` 作為主機名稱。
 - 如果主控台是在 80 以外的埠上執行，請併入埠號。例如：`http://myhostname:11812/prism/eclipse/updateSite/site.xml`
 - c. 按一下**確定**。
4. 在**要造訪的更新網站**對話框中，選取 **Build Forge 更新網站**勾選框，然後按一下**完成**。
5. 系統會在**搜尋結果**對話框中，顯示可用的外掛程式清單。選取所有提供的外掛程式，然後按**下一步**。

註：「反映器」外掛程式需要「頻率」外掛程式。如果是單獨安裝，則不會執行。

6. 閱讀授權合約，並選取**我接受授權合約中的條款**，然後按下一步。
7. 選取特性的安裝位置。如果要新增位置，請按一下**新建位置**，然後瀏覽至您要的位置。
8. 按一下**完成**。
9. 如果顯示**特性驗證**對話框，請按一下**安裝**。之所以會出現這個對話框，是因為外掛程式是未簽署的特性。除非您選取**全部安裝**，否則系統會針對每一項已安裝的特性都顯示一次對話框。
10. 系統會要求您重新啟動 Eclipse，讓變更生效。按一下**是**。

啓用 SSL 時的替代安裝

當在 Build Forge 系統上啓用 SSL 時，請使用替代安裝方法。

開始之前

現行版本的 Eclipse 和 Rational Application Developer 皆未啓用 SSL。因此，如果 Build Forge 系統有啓用 SSL，則無法從 Build Forge 系統來安裝 Eclipse 和 Rational Application Developer 外掛程式。暫行解決方法是在未受保護的 Web 伺服器上提供外掛程式的安裝檔案，或手動將這些檔案配送給使用者。

如果要包裝檔案，請執行下列動作：

1. 建立一個名為 prism 的目錄。
2. 在 prism 中，建立一個名為 eclipse 的目錄。
3. 在 eclipse 中，建立一個名為 updateSite 的目錄。

將下列檔案從 `<bfinstall>/webroot/public/prism/eclipse/updateSite` 複製到您建立的 updateSite 目錄：

- features 目錄
- plugins 目錄
- site.xml 檔

關於這項作業

在 prism 目錄可供使用之後，使用者可從其 IDE 中執行下列步驟：

程序

1. 選取**說明 > 軟體更新 > 搜尋並安裝**。
2. 按一下**搜尋要安裝的新特性圓鈕**，然後按下一步。系統會顯示**要造訪的更新網站**對話框。
3. 建立新的網站。選擇下列其中一個程序。
 - 從遠端伺服器取得檔案：
 - a. 按一下**新建遠端網站**按鈕。系統會顯示**新建遠端網站**對話框。
 - b. 在名稱欄位中輸入『Build Forge Update Site』。
 - c. 輸入檔案的位置：

`http://host/path/prism/eclipse/updateSite/site.xml`

host 是 Web 伺服器的主機名稱或 IP 位址。

path 是從伺服器根目錄至檔案放置所在的路徑。

d. 按一下**確定**。

• 從本端主機取得檔案：

a. 按一下**新建本端網站**按鈕。系統會顯示**新建本端網站**對話框。

b. 在名稱欄位中輸入『Build Forge Update Site』。

c. 輸入檔案的位置：

`file://path/prism/eclipse/updateSite/site.xml`

path 指定檔案的位置。

d. 按一下**確定**。

4. 在**要造訪的更新網站**對話框中，選取 **Build Forge 更新網站**勾選框，然後按一下**完成**。

5. 系統會在**搜尋結果**對話框中，顯示可用的外掛程式清單。選取所有提供的外掛程式，然後按**下一步**。

註：「反映器」外掛程式需要「頻率」外掛程式。如果是單獨安裝，則不會執行。

6. 閱讀授權合約，並選取**我接受授權合約中的條款**，然後按**下一步**。

7. 選取特性的安裝位置。如果要新增位置，請按一下**新建位置**，然後瀏覽至您要的位置。

8. 按一下**完成**。

9. 如果顯示**特性驗證**對話框，請按一下**安裝**。之所以會出現這個對話框，是因為外掛程式是未簽署的特性。除非您選取**全部安裝**，否則系統會針對每一項已安裝的特性都顯示一次對話框。

10. 系統會要求您重新啟動 Eclipse，讓變更生效。按一下**是**。

使用 Eclipse 及 Rational Application Developer 的外掛程式

如果要存取啟動工作並檢視專案日誌（「頻率」外掛程式），請執行下列動作：

1. 在您的 IDE 中，選取視窗 > 開啟視景 > 其他。

2. 選取 **Build Forge** 視景。視景包含下列視窗：

- 主控台瀏覽器
- 建置資訊
- 建置日誌

3. 用滑鼠右鍵按一下**主控台瀏覽器**，並選取**新建主控台**，以配置與「管理主控台」的連線。

4. 在 **Build Forge 服務層主機名稱**中輸入主機名稱或 IP 位址，驗證或編輯其他欄位，然後按一下**確定**。

在建立連線時，「建置資訊」視窗中會移入可供您執行的工作。如果要使用本端檔案來執行工作，請配置所要使用的工作，以及要使用的檔案。

1. 在您的 IDE 中，連接到 Build Forge。

2. 在**主控台瀏覽器**視窗中，用滑鼠右鍵按一下工作，然後選取**內容**。

3. 在**內容**中，選取 **Build Forge 專案構件**。
4. 在 **Build Forge 專案構件**中，選取要處理的專案，並選取要使用的本端檔案。

如需相關資訊，請參閱外掛程式隨附的線上說明。

Rational Team Concert 的外掛程式

Rational Team Concert 用戶端的外掛程式是 Rational Team Concert 與 Build Forge 整合中的一個元件。同時也會需要伺服器延伸與配接器範本。設定與 Rational Team Concert 的整合後，Rational Team Concert 使用者即可完成下列作業：

- 將 Build Forge 設為 RTC 建置伺服器
- 將 Build Forge 專案設為 RTC 建置定義
- 檢視專案、執行工作，以及檢視 RTC 用戶端的工作結果

如需設定整合的相關指示，請參閱 第 497 頁的第 33 章，『Rational Build Forge 和 Rational Team Concert』。

使用 Rational Team Concert 外掛程式

這些指示假設您已設定好 Rational Team Concert 整合，且已將外掛程式安裝在 Rational Team Concert 用戶端中。

如需設定整合的相關指示，請參閱 第 497 頁的第 33 章，『Rational Build Forge 和 Rational Team Concert』。

如果要設定建置定義並執行建置，請執行下列動作：

1. 設定建置定義。
 - a. 在「團隊構件」視圖中，展開專案資料夾。
 - b. 用滑鼠右鍵按一下**建置**，然後按一下**新建建置定義**。
 - c. 在「新建建置定義」中選取**建立新建置**，然後按**下一步**。
 - d. 在「一般資訊」中，輸入建置 ID 和說明。在「可用的範本」清單中，選取 **Rational Build Forge**。按**下一步**。
 - e. 在「其他配置」中，同時選取「一般」和「內容」，並按一下**完成**。這時會建立標籤，並以您輸入的「建置 ID」作為標籤名稱。
 - f. 按一下 **Build Forge** 標籤。
2. 為建置定義選取一個專案。
 - a. 按一下 Build Forge 標籤。輸入必要的資訊，連接到 Build Forge：
 - 主機名稱 - Build Forge 執行所在的主機名稱。若有設定**主控台 URL** 系統設定，則必須符合該項設定。如果您無法在**管理 > 系統**中存取該項設定，請聯絡管理者。
 - 埠 - 用來與 Build Forge 進行通訊的埠。預設埠為 3966。如果選取**安全連接 Build Forge**，依預設會顯埠 49150。如果安裝作業所用的埠號不同於預設埠，請輸入該埠號。若有設定**主控台 URL** 系統設定，則必須符合其中所設定的埠號。如果您無法在**管理 > 系統**中存取該項設定，請聯絡管理者。
 - 使用者名稱 - 連接 Build Forge 的使用者名稱。使用者必須已經存在於 Build Forge 中。
 - 密碼 - 使用者名稱的密碼

- 確認密碼 - 使用者名稱的密碼
- b. 按一下**取得專案**
- c. 在出現的 **Build Forge 專案**清單中，選取這項建置定義的專案。
- 3. 要求建置。
 - a. 用滑鼠右鍵按一下建置定義，然後選取**要求建置**。
 - b. 指定您要的「建置選項」和「建置內容」，然後按一下**提交**。
- 4. 檢查建置結果。
 - a. 在建置完成之後，請在**建置**標籤的清單中選取它。這時會顯示一個視窗。
 - b. 在「外部鏈結」下，按一下 **Build Forge 結果**鏈結。
 - c. 即會顯示 Build Forge 的登入畫面。登入。
 - d. 跳至**首頁 > 已完成的執行**，然後從清單中選取您的建置。
 - e. 這時會顯示建置步驟與結果。請按一下步驟鏈結，以查看步驟的日誌。
 - f. 當完成時，請按一下**登出**，並關閉視窗。

Rational Team Concert 外掛程式的疑難排解

本節說明已知問題以及其暫行解決方法。

處於「逾期」狀態的工作

如果工作停留在「逾期」狀態超過數分鐘，可能是已當掉。

暫行解決方法

停止再重新啟動 Build Forge 引擎。請參閱第 234 頁的『啟動和停止 Perl 引擎』。

第 31 章 Rational Build Forge 與 Perforce

BuildForge 可以搭配 Perforce 使用來管理原始碼。

這個主題說明整合 Perforce 時額外的設定需求，並彙總所提供之範例配接器範本的性質。

Perforce 配接器範本範例提供了分析基準線變更的方法。一般而言，變更分析是用於省略建置：如果基準線元件未變更，則不會重建

Perforce 配接器額外的設定需求

請參閱第 438 頁的『配接器需求』以取得一般需求。

1. 在與代理程式相同的主機上安裝 P4 指令行用戶端。
2. 設定 Build Forge 要使用的 Perforce 使用者帳戶。
3. 設定代理程式環境，使其能夠存取用戶端。

Perforce 配接器範本範例

所提供的配接器範本範例如下所示。

PerforceByDate

1. 查詢 Perforce 用戶端於配接器執行後所發生的變更。
2. 將下列資訊寫入 BOM 報告：變更、日期、時間、使用者、Perforce 用戶端與註解。
3. 將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

配接器中設定的變數：

- LAST_RUN
- CurDate
- P4PORT
- BFCLIENT
- P4CLIENT
- FILESPEC

環境所需的其他變數：

- P4USER
- P4PASSWD

這些變數不能使用「指派隱藏」內容。使用者和密碼會以明碼寫入步驟日誌。

PerforceByRev

1. 查詢 Perforce 用戶端自前次儲存庫修訂後所發生的變更。
2. 將下列資訊寫入 BOM 報告：變更、日期、時間、使用者、Perforce 用戶端與註解。
3. 將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

配接器中設定的變數：

- LAST_RUN
- CurDate
- P4PORT
- BFCLIENT
- P4CLIENT
- FILESPEC

環境所需的其他變數：

- P4USER
- P4PASSWD

這些變數不能使用「指派隱藏」內容。使用者和密碼會以明碼寫入步驟日誌。

第 32 章 Rational Build Forge 與 Rational Quality Manager

Rational Build Forge 可以搭配 Rational Quality Manager 使用

您可以將 Rational® Build Forge® 與 Rational Quality Manager 整合，以定義及執行從 Rational Quality Manager 建置和包裝軟體的專案。

在 Rational Quality Manager 中可以完成此整合。請參閱您所用 Rational Quality Manager 版本的資訊中心。如需相關鏈結，請參閱 Rational Quality Manager 資訊頁面。
for links.

- 若為 Rational Quality Manager 第 2 版，請參閱**整合 > Rational Quality Manager 以及軟體庫存和自動化工具 > 與 Rational Build Forge 整合**
- 若為 Rational Quality Manager 第 1 版，請參閱**整合 > 與軟體庫存和供應工具整合 > 與 Rational Build Forge 整合**

第 33 章 Rational Build Forge 和 Rational Team Concert

Build Forge 可以與 Rational Team Concert 整合。

設定好整合功能時，Rational Team Concert 的使用者可以執行下列動作：

- 將 Build Forge 設為 RTC 建置伺服器
- 檢視專案、執行工作，以及檢視 RTC 用戶端的工作結果
- 將 Build Forge 專案設為 RTC 建置定義

Rational Team Concert 3.x 版

Rational Team Concert 安裝隨附整合元件。

相較於與舊版 Rational Team Concert 的整合，Rational Team Concert 3.x 與 Rational Build Forge 7.1.1.3 版及更新版本之間的整合具有明顯優勢。Rational Team Concert 3.x 是與 Rational Build Forge 整合時的偏好平台。

安裝 Rational Build Forge 及 Rational Team Concert 後，您可以在 Rational Team Concert 中配置該整合。通常，這兩個產品會安裝在不同主機上。

若要配置該整合及進行疑難排解，請參閱 jazz.net 上的 Rational Build Forge Wiki，以取得相關指示。

Rational Team Concert 1.x 及 2.x 版

您需要安裝 Rational Build Forge 隨附的元件，以便與 Rational Team Concert 1.x 及 2.x 版整合

該整合需要您執行下列動作：

- 安裝 Rational Build Forge 的元件
- 配置 Rational Team Concert，使其與 Rational Build Forge 進行通訊

安裝元件

您將安裝下列元件：

- **Server Extension for Rational Team Concert Server**：此元件可在 Rational Team Concert 和 Build Forge 之間啟用通訊。伺服器延伸安裝在 RTC 伺服器上。第 498 頁的『安裝 Rational Team Concert 伺服器延伸』中含有安裝指示。您可以將 Build Forge 使用者配置給 Rational Team Concert 伺服器。
- **Client plug-in**：此元件是供 Rational Team Concert 用戶端使用的外掛程式。它容許使用者存取 Build Forge，以檢視專案、執行工作，以及檢視工作結果。用戶端外掛程式必須安裝在會存取 Build Forge 的每一個 RTC 用戶端上。第 499 頁的『安裝 Rational Team Concert 的用戶端外掛程式』中含有安裝指示。
- **Adaptor**：此元件是原始檔類型配接器，可讓專案存取 Rational Team Concert 儲存庫中的檔案。RTC 1.x 版與 RTC 2.x 版分別使用不同的配接器。請參閱第 501 頁的『配置 Rational Team Concert 配接器』。

配置 Rational Team Concert

若要配置該整合及進行疑難排解，請參閱 jazz.net 上的 Rational Build Forge Wiki，以取得相關指示。

安裝 Rational Team Concert 伺服器延伸

開始之前

必備條件：

- Rational Team Concert 必須是 1.x 或 2.x 版。請注意，程序中的指示和檔案會隨著版本而不同。
- Build Forge 系統必須執行中。

關於這項作業

請從「Rational Team Concert 伺服器」主機來執行下列步驟：

程序

1. 登入 Rational Team Concert 伺服器主機。
2. 導覽至 Build Forge 伺服器，並使用適用於 Rational Team Concert 版本的 URL，將伺服器延伸規格下載至暫存位置。
 - Rational Team Concert 1.x 版：
`http://<bf_console_hostname>/rtc-server/BuildForgeConnectorServer.zip`
 - Rational Team Concert 2.x 版：
`http://<bf_console_hostname>/rtc2-server/BuildForgeConnectorServer.zip`
 - 如果執行 Rational Team Concert 所在的系統與執行 Build Forge 所在的系統相同，您可以使用 localhost 作為主機名稱。
 - 如果主控台是在 80 以外的埠上執行，請併入埠號。範例：`http://myhostname:11812/clients`。
3. 將檔案內容解壓縮至 `RTC_install/jazz/server`。如果您下載了適用於 Rational Team Concert v1.x 版的延伸，便會新增下列檔案：

```
RTC_install/jazz/server/buildforgeconnector-update-site (目錄)
RTC_install/jazz/server/provision_profiles/buildforgeconnector-profile.ini
```

如果您下載了適用於 Rational Team Concert 2.x 版的延伸，便會新增下列檔案：

```
RTC_install/jazz/server/buildforgeconnector-update-site (目錄)
RTC_install/jazz/server/conf/provision_profiles/buildforgeconnector-profile.ini
```

若要在 WebSphere Application Server 中執行 RTC 伺服器，您必須編輯 `buildforgeconnector-profile.ini` 檔，使其具有 `buildforgeconnector-update-site` 目錄的完整路徑。

4. 重新啟動 Rational Team Concert 伺服器。

結果

Rational Team Concert 伺服器使用 Build Forge 使用者授權來與主控台通訊。

安裝 Rational Team Concert 的用戶端外掛程式

從 Build Forge 伺服器安裝 Rational Team Concert 的用戶端外掛程式。

開始之前

必備條件：

- Rational Team Concert 1.x 版或 2.x 版 - 各自需要不同的外掛程式。
- Build Forge Connector 伺服器延伸必須安裝在 Rational Team Concert 伺服器上。
- Build Forge 系統必須執行中。
- 每位使用者都應該要有兩個主控台登入 ID。Build Forge 限制一個使用者只能有一個登入階段作業。同一個使用者名稱不能同時透過「管理主控台」和 RTC 用戶端登入。當新的階段作業啟動時，任何現有的階段作業都會終止。RTC 用戶端使用者應該有兩個 ID：
 - UserRTC - 在配置建置定義來執行建置時使用。這個使用者名稱會用來存取 Build Forge 主控台以執行建置。
 - UserBF - 在檢查建置結果時使用。其需要在 RTC 用戶端的視窗內直接登入 Build Forge「管理主控台」。

關於這項作業

用戶端外掛程式必須安裝在會存取 Build Forge 的每一個 Rational Team Concert 用戶端上。

如果要安裝外掛程式，請從 Rational Team Concert 用戶端執行下列步驟。

程序

1. 選取說明 > 軟體更新 > 搜尋並安裝。
2. 選取搜尋要安裝的新特性，然後按下一步。系統會顯示要造訪的更新網站對話框。
3. 按一下新建遠端網站。系統會顯示新建遠端網站對話框。
 - a. 在名稱欄位中輸入Build Forge 連接器更新網站。
 - b. 在 URL 欄位中，輸入更新網站 URL。URL 會隨著所用的 Rational Team Concert 版本而不同。

Rational Team Concert version 1.x:
`http://<console_host_name>/rtc/site.xml`

Rational Team Concert version 2.x:
`http://<console_host_name>/rtc2/site.xml`

 - 如果執行 Rational Team Concert 所在的系統與執行 Build Forge 所在的系統相同，您可以使用 localhost 作為主機名稱。
 - 如果主控台是在 80 以外的埠上執行，請併入埠號。如果您使用 Rational Team Concert 1.x 版，且 Build Forge 是在 11812 埠上執行，下列範例便能夠運作：`http://myhostname:11812/rtc/site.xml`
 - c. 按一下確定。
4. 在要造訪的更新網站對話框中，選取 **Build Forge 連接器更新網站**勾選框，然後按一下完成。
5. 系統會在搜尋結果對話框中顯示更新網站內可用特性的清單。請選取所有提供的特性，然後按下一步。

6. 閱讀授權合約，並選取**我接受授權合約中的條款**，然後按**下一步**。
7. 選取特性的安裝位置。如果要新增位置，請按一下**變更位置**，然後瀏覽至您要的位置。
8. 按一下**完成**。
9. 如果顯示**特性驗證**對話框，請按一下**安裝**。之所以會出現這個對話框，是因為外掛程式是未簽署的特性。除非您選取**全部安裝**，否則系統會針對每一項已安裝的特性都顯示一次對話框。
10. 系統會要求您重新啟動 Rational Team Concert，讓變更生效。按一下**是**。
11. 重新啟動 Build Forge。

結果

整合完成之後，您可以利用外掛程式來執行工作以及檢查工作結果。您必須指定 RationalBuildForgeConnector 作為建置的「建置引擎」。

啓用 SSL 時的替代安裝

當在 Build Forge 系統上啓用 SSL 時，請使用替代安裝方法。

開始之前

現行版本的 Rational Team Concert 未啓用 SSL。因此，當 Build Forge 系統啓用 SSL 時，就不能從 Build Forge 系統安裝外掛程式。暫行解決方法是在未受保護的 Web 伺服器上提供外掛程式的安裝檔案，或手動將這些檔案配送給使用者。

在 `<bfinstall>/webroot/public/rtc`（若為 RTC 1.x 版）或 `<bfinstall>/webroot/public/rtc2`（若為 RTC 2.x 版）中，需要有下列檔案：

- features 目錄
- plugins 目錄
- site.xml 檔

關於這項作業

備妥 rtc 或 rtc2 目錄之後，使用者便在他們的 Rational Team Concert 用戶端內執行這些步驟：

程序

1. 選取**說明 > 軟體更新 > 搜尋並安裝**。
2. 按一下**搜尋要安裝的新特性**，然後按**下一步**。系統會顯示**要造訪的更新網站**對話框。
3. 建立新的網站。選擇下列其中一個程序。
 - 從遠端伺服器取得檔案：
 - a. 按一下**新建遠端網站**。系統會顯示**新建遠端網站**對話框。
 - b. 在名稱欄位中輸入『Build Forge Connector Update Site』。
 - c. 輸入檔案的位置：

`http://host/prism/eclipse/updateSite/site.xml`

`host` 是 Web 伺服器的主機名稱或 IP 位址。

path 是從伺服器根目錄至檔案放置所在的路徑。

d. 按一下**確定**。

• 從本端主機取得檔案：

a. 按一下**新建本端網站**。系統會顯示**新建本端網站**對話框。

b. 在名稱欄位中輸入 『Build Forge Connector Update Site』。

c. 輸入檔案的位置：

`file:///path/prism/eclipse/updateSite/site.xml`

path 指定檔案的位置。

d. 按一下**確定**。

4. 在**要造訪的更新網站**對話框中，選取 **Build Forge 連接器更新網站**勾選框，然後按一下**完成**。

5. 系統會在**搜尋結果**對話框中顯示可用特性的清單。請選取所有提供的特性，然後按**下一步**。

註：「反映器」外掛程式需要「頻率」外掛程式。如果是單獨安裝，則不會執行。

6. 閱讀授權合約，並選取**我接受授權合約中的條款**，然後按**下一步**。

7. 選取特性的安裝位置。如果要新增位置，請按一下**變更位置**，然後瀏覽至您要的位置。

8. 按一下**完成**。

9. 如果顯示**特性驗證**對話框，請按一下**安裝**。之所以會出現這個對話框，是因為外掛程式是未簽署的特性。除非您選取**全部安裝**，否則系統會針對每一項已安裝的特性都顯示一次對話框。

10. 系統會要求您重新啟動 Rational Team Concert，讓變更生效。按一下**是**。

11. 重新啟動 Build Forge。

配置 Rational Team Concert 配接器

關於這項作業

Build Forge 會自動安裝將 Rational Build Forge 連接至 Rational Team Concert 來源儲存庫的配接器。此配接器在 Rational Build Forge 主控台主機上執行，並可作為用戶端直接與 Rational Team Concert 通訊。您無需在 Rational Build Forge 主控台主機上安裝 Rational Team Concert 用戶端。您可以依下列方式配置個別的 Build Forge 專案，來存取 Rational Team Concert 來源。

程序

1. 定義「伺服器」資源。

• 輸入這個伺服器資源的**名稱**。

• 輸入**主機**：這是用來執行 Rational Team Concert 伺服器之主機的完整網域名稱。

註：此內容通常是指 Build Forge 代理程式。您無需在 Rational Team Server 主機上安裝 Build Forge 代理程式。此內容專門用來直接與 Rational Team Concert 通訊。

• 視狀況輸入其他內容的值。

註：RTC 工具 scm 必須位於使用者設定檔的 PATH 中，或是 RTC 伺服器的啟動設定檔中。配接器會使用 scm 來存取原始檔。

2. 針對使用 Rational Team Concert 儲存庫的每一個專案設定配接器鏈結。下列指示說明如何設定專案的配接器鏈結。
 - a. 在主控台中，跳至**專案 > 配接器鏈結**。
 - b. 按一下**新建配接器鏈結**。在**詳細資料**標籤中設定其內容，然後按一下**儲存**。
 - 狀態 - 選取**作用中**。
 - 名稱 - 輸入配接器鏈結的名稱。
 - 配接器
 - Rational Team Concert 1.x 版請選擇 JazzSCM
 - Rational Team Concert 2.x 版請選擇 JazSCMv2
 - 專案 - 選擇會套用配接器鏈的專案。
 - 環境 - 選擇要用於這個配接器鏈結的環境。其必須是現有的環境。
 - c. 請按一下您剛建立的配接器鏈結。
 - d. 按一下**移入環境**。這個步驟會將配接器中定義的變數 (JazzSCM.xml) 移入指定的環境。
 - e. 按一下**儲存**。
3. 更新配接器所提供的變數。請在您設定並移入的環境中，編輯配接器所提供的變數。

配接器中設定了四個環境變數：

- Current_Date - 設定現行日期。其會用來套用時間戳記。請勿變更這個定義。
- Last_Run - 由系統自動更新。請勿變更這個定義。
- Directory_Path - 設定從儲存庫擷取之原始檔的位置。依預設，其會設為 C:\temp。請將這個目錄變更為您要使用的暫存目錄。

在執行工作之後，依預設，不會清除這個目錄。請從已執行的工作中，刪除舊的目錄。

- Jazz_Server - 設定 Rational Team Concert 伺服器的位置。依預設，其會設為 \$BFServer。您必須將這項設定變更為指向 Rational Team Concert 伺服器之「伺服器」資源的**名稱**內容。
 - a. 在主控台中，移至**環境**。
 - b. 選取配接器鏈結所使用的環境。
 - c. 選取並編輯您要變更的變數。
 - d. 按一下**儲存**。

結果

您現在可以執行專案。每次執行時，其都會連接 Rational Team Concert 儲存庫。其會更新自前次執行專案以來，RTC 伺服器上已變更的原始檔。

其他資源：

- Rational Team Concert 1.x 版：JazzSCM 配接器位於 `<bfinstall>/interface/JazzSCM.xml`。您可以利用文字編輯器或 XML 讀取器來開啓它。

- Rational Team Concert 2.x 版：JazzSCMv2 配接器位於 *<bfinstall>/interface/JazzSCMv2.xml*。您可以利用文字編輯器或 XML 讀取器來開啓它。
- 在 Rational Team Concert 文件中，記載了配接器所用的 SCM 指令。當測試伺服器連線及測試配接器要用的指令時，請使用這些及其他指令。

第 34 章 Rational Build Forge 與 StarTeam

BuildForge 可以搭配 StarTeam 使用來管理原始碼。

這個主題說明整合 StarTeam 時額外的設定需求，並彙總所提供之範例配接器範本的性質。

StarTeam 配接器範本範例提供了分析基準線變更的方法。一般而言，變更分析是用於省略建置：如果基準線元件未變更，則不會重建

StarTeam 配接器額外的設定需求

請參閱第 438 頁的『配接器需求』以取得一般需求。

1. 在與代理程式相同的主機上安裝 StarTeam 指令行用戶端。
2. 設定 Build Forge 要使用的 StarTeam 使用者帳戶。
3. 設定代理程式環境，使其能夠存取用戶端。

StarTeam 配接器範本範例

所提供的配接器範本範例如下所示。

StarTeamBaseline

1. 查詢資料夾的 StarTeam 視圖，以收集檔案的相關資訊。
2. 將下列資訊寫入 BOM 報告：檔名、狀態、修訂和分支。

配接器中設定的變數：

- USER
- PASS
- HOST
- PORT
- PROJECT
- VIEW
- DIR
- BFCLIENT

環境所需的其他變數：

- P4USER
- P4PASSWD

這些變數不能使用「指派隱藏」內容。使用者和密碼會以明碼寫入步驟日誌。

StarTeamByDate

1. 使用 StarTeam API 來查詢 StarTeam 視圖，以找出配接器前次執行時間與現行日期之間的變更。
2. 將下列資訊寫入 BOM 報告：已變更的檔案與目錄、使用者、版本、日期與變更註解。

3. 將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告

配接器中設定的變數：

- STA_USER
- STA_PASS
- STA_HOST
- STA_PORT
- STA_PROJECT
- STA_VIEW
- STA_DIR
- BFCLIENT
- LASTRUN
- CURDATE
- EMAILCHANGES
- STARTEAM80JAR

環境所需的其他變數：

- P4USER
- P4PASSWD

這些變數不能使用「指派隱藏」內容。使用者和密碼會以明碼寫入步驟日誌。

第 35 章 Rational Build Forge 與 Subversion

BuildForge 可以搭配 Subversion (SVN) 使用來管理原始碼。

這個主題說明整合 SVN 時額外的設定需求，並彙總所提供之範例配接器範本的性質。

SVN 配接器範本範例提供了分析基準線變更的方法。一般而言，變更分析是用於省略建置：如果基準線元件未變更，則不會重建

SVN 配接器額外的設定需求

請參閱第 438 頁的『配接器需求』以取得一般需求。SVN 沒有額外的需求。

SVN 配接器範本範例

所提供的配接器範本範例如下所示。

SubversionByDate

1. 查詢子版本在過去日期與現行日期之間所發生的儲存庫變更。
2. 將下列資訊寫入 BOM 報告：變更類型、修訂、使用者、檔案或目錄以及變更日期。
3. 將下列資訊寫入 BOM 報告：檔名、狀態、修訂和分支。

使用的變數：

- SVN_CLIENT
- SVN_OPTS
- SVN_REPOSITORY
- SVN_LAST_REV
- SVN_LAST_DATE

SubversionByRev

說明如何設定此配接器的文章位於 jazz.net：<https://jazz.net/wiki/bin/view/Main/RationalBuildForge/SVNAdaptorExample>。

1. 查詢子版本在現行修訂與之前修訂之間所發生的儲存庫變更。
2. 針對每次的變更，將下列資訊寫入 BOM 報告：修訂、使用者、變更類型、檔案或目錄路徑以及變更日期。
3. 將變更詳細資料（從 diff 指令輸出）寫入 BOM 報告。

使用的變數：

- SVN_CLIENT
- SVN_OPTS
- SVN_REPOSITORY
- SVN_LAST_REV
- SVN_LAST_DATE

第 36 章 Rational Build Forge 與 WebSphere 產品

本節說明整合 Build Forge® 與 WebSphere 產品的方式：

- 使用 WebSphere Application Server 而非 Apache Tomcat 來執行 Build Forge 服務和「快速報告」
- 使用 IBM HTTP Server (IHS) 而非 Apache 作為 Web 應用程式伺服器

使用 WebSphere Application Server 來取代 Apache Tomcat

您可以使用 WebSphere Application Server 作為應用程式伺服器，來取代附帶提供的 Apache Tomcat 應用程式伺服器。您可以配置它來執行 Build Forge 服務與 Build Forge 說明應用程式。

在 Websphere Application Server 中，有三個配置區域：

- 您可以配置它來執行 Build Forge 服務與說明應用程式。這是必要的。
- 為 Build Forge 服務應用程式配置 Java 2 安全。如果您在 WebSphere Application Server 上執行 Java 2 安全，則必須執行此作業。
- 在 Build Forge 中配置 SSL 與密碼加密的支援。如果您已在 Build Forge 中配置這些特性，則必須執行此作業。

必備條件

- 查閱特定產品的詳細系統需求網站。執行查詢來查出 Rational Build Forge 的版次與版本，以查明哪個版本的 WebSphere Application Server 是受支援的。
- 由於授權伺服器的限制，Build Forge 主控台和 WAS 必須在相同的作業系統和硬體平台上執行。
- 請先確定您已對 WebSphere Application Server 套用最新的修正套件之後，再嘗試部署 Build Forge 服務層。

重要：當您使用這項配置時，您必須先在 WebSphere Application Server 中啟動 Build Forge 服務和「Build Forge 說明」應用程式，才能啟動 Build Forge。不會自動啟動它們。

配置 Websphere Application Server 來執行 Build Forge® 應用程式

配置 WebSphere Application Server 來執行服務應用程式與說明應用程式。

關於這項作業

在 WebSphere Application Server 管理主控台上執行此程序中的步驟。

在 WebSphere Application Server 8.0 版中，WebSphere Application Server 管理主控台的使用者介面略有變更。以下是第 7 版與更早版本的相關指示。8.0 版中的下列差異會影響此程序：

- 若要存取應用程式，請按一下**應用程式 > 應用程式類型 > 企業應用程式**。
- 選取應用程式之後，設定環境定義根目錄。

註：您不會看見建立資源的 **MBean** 的勾選框。

程序

1. 開啓 WebSphere Application Server 管理主控台。以下是主控台的 URL：
 - `http://<was_host>:<was_port>/ibm/console`；預設埠是 9060。
 - `https://<was_host>:<was_port>/ibm/console`；預設埠是 9043。如果已啓用 WebSphere Application Server 管理安全，請使用這個 URL。
2. 配置 JDBC 驅動程式的支援：
 - a. 建立新的變數 `RBF_JDBC_DRIVER_PATH`。在 **環境 > WebSphere 變數** 中建立變數。其範圍應該是 WebSphere Application Server 節點和伺服器。請將值設為含有資料庫驅動程式 JAR 檔的目錄。
 - b. 將變更儲存至主要配置。
 - c. 停止並重新啓動 WebSphere Application Server，使新的變數可供使用。
 - d. 建立新的共用程式庫 `RBF_JDBC_LIBRARY`。

在 **環境 > 共用程式庫** 中建立程式庫。使用您剛建立的 `RBF_JDBC_DRIVER_PATH`，新增 JDBC 裝置驅動程式的 JAR 檔名稱。如果有多個 jar 檔要列出，請將每一個 jar 檔個別放在一行，如以下針對 DB2 的指示。

下列為針對 MySQL 資料庫驅動程式的範例：

```
${RBF_JDBC_DRIVER_PATH}\mysql-connector-java-5.0.5-bin.jar
```

此範例適用於 DB2：

```
${RBF_JDBC_DRIVER_PATH}\db2jcc.jar  
${RBF_JDBC_DRIVER_PATH}\db2jcc_license_cu.jar
```

對於 UNIX 或 Linux，在上面的範例中，使用斜線 (/) 來取代反斜線 (\)。

- e. 將變更儲存至主要配置。
3. 安裝 Build Forge[®] 應用程式 WAR 檔。
 - a. 開啓 **應用程式 > 企業應用程式**。
 - b. 按一下 **安裝**。
 - c. 瀏覽至 `jas.war` 檔。請使用 `jas` 作為環境定義根目錄。
 - Windows 系統：
`<bfinstall>\PrepForExternal\jas.war`
 - UNIX 及 Linux 系統：
`<bfinstall>/Platform/PrepForExternal/jas.war`
 - d. 按下一步，然後清除建立資源的 **MBean** 勾選框（如果已選取）。
 - e. 按下一步，然後按一下 **完成**。
 - f. 在安裝文字底端，按一下 **儲存**。
4. 安裝 Build Forge[®] 說明 WAR 檔。
 - a. 按一下 **應用程式 > 企業應用程式**。
 - b. 按一下 **安裝**。

- c. 瀏覽至 Build Forge[®] 安裝目錄中的 BuildForgeHelp.war 檔。請使用 /BuildForgeHelp 作為環境定義根目錄。
 - d. 按下一步，然後清除**建立資源的 MBean** 勾選框（如果已選取）：
 - e. 按下一步，直到看到**完成**按鈕為止，然後按一下**完成**。
 - f. 按一下安裝文字底端的**儲存**。
5. 將 RBF_JDBC_LIBRARY 設為共用程式庫參照。
 - a. 按一下**應用程式 > 企業應用程式**。
 - b. 按一下 **jas.war** 鏈結。
 - c. 按一下**共用程式庫參照**。
 - d. 選取**服務層登入 Servlet** 方框。
 - e. 按一下**參照共用程式庫**。
 - f. 將 RBF_JDBC_LIBRARY 加入清單中。
 6. 設定類別載入器順序：
 - a. 在**管理模組**下，選取**服務層登入 Servlet**，然後從下拉功能表中選取**類別載入器順序**。
 - b. 請將該值變更為**載入類別時先載入本端類別載入器（母項類別最後）**。
 7. 將變更儲存至主要配置。
 8. 檢查 Build Forge[®] 配置檔。確定檔案中的 services_url 項目有指向正確的 URL。如果您是使用 WebSphere Application Server，buildforge.conf 中的這個內容值看起來應該像是 https://<hostname>:9443/jas 或 https://<hostname>:9080/jas。buildforge.conf 檔位於下列位置：
 - Windows 系統
 <bfinstall>\
 - UNIX 及 Linux 系統
 <bfinstall>/Platform/
 9. 在 UNIX 和 Linux 系統中，新增對 jas 的 bin 目錄中所有檔案的執行許可權。

 在此範例中，是使用設定檔 AppSrv01。如果您的設定檔名稱不是 AppSrv01，請使用您的設定檔名稱。


```
chmod -R 755 <WAS-install>/AppServer/profiles/AppSrv01/installedApps/
<node>/jas_war.ear/jas.war/bin
```
 10. 更新程式庫 PATH 變數來包括下列目錄：
 - Windows 系統，PATH 包括：
 <bfinstall>\token_libs
 - AIX 系統，LIBPATH 包括：
 <bfinstall>/Platform/token_libs
 - Linux/Solaris，LD_LIBRARY_PATH 包括：
 <bfinstall>/Platform/token_libs
 11. 如果使用 64 位元的 Linux 或 WebSphere Application Server，請完成下列步驟來複製提供的 64 位元程式庫：
 - a. 重新命名程式庫檔案。

- Windows 系統：將 `<bfinstall>\token_libs\rcl_rational.dll` 重新命名為 `<bfinstall>\token_libs\rcl_rational.dll.old`。
 - UNIX 和 Linux 系統：將 `<bfinstall>/Platform/token_libs/librcl_rational.so` 重新命名為 `<bfinstall>/Platform/token_libs/librcl_rational.so.old`。
- b. 將 64 位元程式庫檔案自 `token_libs_64` 複製到 `token_libs`。請勿在指令中使用換行。此處的使用方式是為了可讀性。
- Windows 系統


```
copy <bfinstall>\token_libs_64\rcl_rational.dll
    <bfinstall>\token_libs\rcl_rational.dll
```
 - UNIX 及 Linux 系統


```
cp <bfinstall>/Platform/token_libs_64/librcl_rational.so
    <bfinstall>/Platform/token_libs/librcl_rational.so
```

在 Websphere Application Server 中為 Build Forge 配置 Java 2 安全

如果您執行的 WebSphere Application Server 已啟用 Java 2 安全，您必須配置 Build Forge 服務層來使用它。

關於這項作業

若要配置 WebSphere Application Server 管理主控台，請執行下列步驟。

程序

1. 開啓應用程式 > 企業應用程式。
2. 選取 **jas_war**，然後按一下**更新**。
3. 選取**置換或新增單一檔案**。
4. 在指定從已安裝的應用程式保存檔開始，到要置換或新增的檔案為止的路徑中，輸入 `META-INF/was.policy`。
5. 選取**本端檔案系統**，再瀏覽至作業系統的 `was.policy` 檔，然後按下一步。
 - Windows 系統：


```
<bfinstall>/samples/projects/was.policy
```
 - UNIX 及 Linux 系統：


```
<bfinstall>Platform/samples/projects/was.policy
```
6. 按一下**確定**。
7. 將變更儲存至主要配置，然後停止並啓動 Build Forge 服務應用程式。

在 Websphere Application Server 中為 Build Forge 配置 SSL 與密碼加密支援

在 WebSphere Application Server 中為 Build Forge 應用程式啓用 SSL 和（或）密碼加密

關於這項作業

若要支援 SSL 與密碼加密，還必須進行其他配置。您可以配置其中一項，或同時配置這兩項。您必須先在 WebSphere Application Server 上配置 SSL，才能對 Build Forge 啓用 SSL。

程序

1. 將 `<bfinstall>\keystore` 目錄複製到 `<WAS_INSTALL_DIR>\AppServer\profiles\AppSrv01`。如果您使用的設定檔不是 `AppSrv01`，請將金鑰儲存庫目錄複製到該設定檔目錄。

路徑在 UNIX 與 Linux 系統上均相同。此目錄包含應用程式伺服器設定檔。此目錄可能會依據您的 WebSphere Application Server 版本而有所不同。

2. 登入 WebSphere 主控台。使用主控台管理 WebSphere。
3. 移至「自訂內容」頁面。開啓伺服器 > 伺服器類型 > **WebSphere Application Server** > **server1** > **Java 與程序管理** > **程序定義** > **Java 虛擬機器** > **自訂內容**

預設伺服器名稱爲 **server1**。如果您的伺服器有不同的名稱，請使用該名稱。

4. 啓用 SSL。設定自訂內容以指向 `bfclient.conf` 檔案。

- Windows

`com.buildforge.client.config=<bfinstall>\bfclient.conf`

- UNIX 或 Linux

`com.buildforge.client.config=<bfinstall>/Platform/bfclient.conf`

5. 啓用密碼加密。設定自訂內容以指向 `bfpassword.conf` 檔案。

- Windows

`com.buildforge.password.encryption.file=<bfinstall>\bfpassword.conf`

- UNIX 或 Linux

`com.buildforge.password.encryption.file=<bfinstall>/Platform/bfpassword.conf`

6. 重新啓動 WebSphere Application Server。變更會在您重新啓動後生效。

使用 Websphere Application Server 來執行 Build Forge 應用程式的啓動順序

在啓動 Build Forge 之前，請先在 WebSphere Application Server 中啓動 Build Forge 應用程式。

關於這項作業

在啓動 Build Forge 之前，請先從 WebSphere Application Server 中啓動服務與說明應用程式。

程序

1. 開啓 WebSphere Application Server 管理主控台。
2. 選擇應用程式 > 企業應用程式。
3. 選取 **jas.war** 和 **BuildForgeHelp.war**。
4. 按一下啓動。
5. 啓動 Build Forge。

使用 IBM HTTP Server 來取代 Apache HTTP Server

您可以配置 IBM HTTP Server (IHS) 搭配「管理主控台」使用，而非使用依預設安裝的 Apache HTTP Server。

開始之前

- 由於授權伺服器的限制，Build Forge 主控台和 IBM HTTP Server 必須在相同的作業系統和硬體平台上執行。
- 您必須使用所提供的那些經過前置編譯的 **PHP 模組**。請不要嘗試使用您自行編譯的 PHP。查看版本注意事項，以確定實際上已併入您的作業系統模組。若未併入，則您必須與支援人員合作，來編譯模組。
- 如果您進行此設定時遇到困難，請聯絡 IBM 支援中心。
- 在範例中，會使用下列目錄作為 IBM HTTP Server 安裝目錄。如果它們不同，請換成您的安裝目錄。
 - Windows：C:\Program Files\IBM\HTTPServer
 - Unix 及 Linux：/opt/IBM/HTTPServer

關於這項作業

此主題說明如何執行下列作業：

1. 設定經過前置編譯的 PHP 檔。
2. 修改 IBM HTTP Server 配置檔以指向 Build Forge Web 應用程式。
3. 如果您使用 Proxy 伺服器來存取資料庫，請修改 PHP 為使用 Proxy 伺服器。
4. 如果您使用 SSL，請配置 IHS 為透過 SSL 運作 Build Forge。
5. Windows 系統：取代 IBM HTTP Server 安裝目錄中的 libeay32.dll。
6. AIX 系統：其他設定

設定經過前置編譯的 PHP 檔

關於這項作業

此作業說明如何找出所提供的 PHP 檔，並將它們複製到指定的位置。檔案位於下列目錄：

- Windows 系統
`<bfinstall>\ihs_modules`
- UNIX 及 Linux 系統
`<bfinstall>/Platform/ihs_modules`

程序

1. 複製 PHP 模組。必要時，請建立 modules 目錄。
 - Windows 系統

```
cd <bfinstall>\ihs_modules\modules\  
copy *.dll C:\Program Files\IBM\HTTPServer\modules
```
 - UNIX 及 Linux 系統

```
cd <bfinstall>/Platform/ihs_modules/modules  
cp libphp5.so /opt/IBM/HTTPServer/modules
```
2. 複製 PHP 延伸。必要時，請建立 extensions 目錄。
 - Windows 系統

```
cd <bfinstall>\ihs_modules\lib\php\extensions\  
copy *.dll C:\Program Files\IBM\HTTPServer\extensions
```
 - UNIX 及 Linux 系統

- ```
cd <bfinstall>/Platform/ihs_modules/lib/php/extensions/no-debug-zts-20090626
cp *.so /opt/IBM/HTTPServer/extensions
```
- 複製所提供的 `php.ini` 檔。必要時，請建立 `conf` 目錄。
    - Windows 系統

```
cd <bfinstall>\ihs_modules
copy php.ini C:\Program Files\IBM\HTTPServer\conf
```
    - UNIX 及 Linux 系統

```
cd <bfinstall>/Platform/ihs_modules
cp php.ini /opt/IBM/HTTPServer/conf
```
  - 編輯已複製到 `conf` 目錄的 `php.ini` 檔，以併入指向 `extensions` 目錄的指標。新增或取代下面這一行：
    - Windows 系統

```
extension_dir = C:\Program Files\IBM\HTTPServer\extensions
```
    - UNIX 及 Linux 系統

```
extension_dir = /opt/IBM/HTTPServer/extensions
```
  - 以您的系統資訊編輯 `php.ini` 檔。您可以使用提供給 Apache HTTP Server 作為引導的 `php.ini` 檔。於下列位置提供該檔案：
    - Windows 系統

```
<bfinstall>\Apache\php\php.ini
```
    - UNIX 及 Linux 系統

```
<bfinstall>/server/apache/conf/php.ini
```
  - 對 `php.ini` 中所使用的所有目錄新增寫入權。在 UNIX 與 Linux 系統上，對這些目錄執行下列指令：

```
chmod -R 777
```

## 編輯 IBM HTTP Server 的配置檔 程序

- 在伺服器安裝的 `conf` 目錄中，找出 IBM HTTP Server (IHS) 的 `httpd.conf` 檔。
- 修改 `DocumentRoot` 設定以指向 Build Forge Web 應用程式，如範例所示。在此範例中，Build Forge 安裝目錄是 `/opt/buildforge`。

```
<VirtualHost *:80>
 ServerAdmin build@yourdomain.com
 DocumentRoot /opt/buildforge/webroot/public
 ServerName ausbuild01.yourdomain.com
 ServerAlias build.yourdomain.com mc.yourdomain.com #optional server aliases
 ErrorLog logs/ausbuild.error_log
 CustomLog logs/ausbuild.access_log common
</VirtualHost>
```
- 新增 PHP 相關項目。
  - Windows 系統

```
LoadModule php5_module "C:\Program Files\IBM\HTTPServer\modules\php5apache2_2.dll"

<IfModule dir_module>
 DirectoryIndex index.html index.php
</IfModule>

AddType application/x-httpd-php .php
```

```
PHPIniDir 是 php.ini 檔的位置
PHPIniDir "C:\Program Files\IBM\HTTPServer\conf"

• UNIX 及 Linux 系統

LoadModule php5_module "/opt/IBM/HTTPServer/modules/libphp5.so"

<IfModule dir_module>
 DirectoryIndex index.html index.php
</IfModule>

AddType application/x-httpd-php .php

PHPIniDir 是 php.ini 檔的位置
PHPIniDir "/opt/IBM/HTTPServer/conf"
```

#### 4. 變更預設使用者的陳述式：

- 若為 AIX 系統，請在 <bfinstall>/webroot/templates\_c 資料夾中的 templates\_c 檔案上，將陳述式變更為 User daemon 和 Group staff。
- 若為 Linux 系統，請在 <bfinstall>/webroot/templates\_c 資料夾中的 templates\_c 檔案上，將陳述式變更為 User daemon 和 Group daemon。chown daemon:daemon /opt/buildforge/webroot/templates\_c 即為 Linux 系統上的一個例子。

#### 5. 如有需要，請變更 IHS 埠號。預設埠號是 80。對 httpd.conf 進行任何其他必要的變更。

## 在 PHP 中識別 Proxy 伺服器

### 關於這項作業

僅當管理主控台必須使用 Proxy 伺服器來存取其資料庫時，才執行此步驟。

### 程序

編輯 PHP 配置檔 php.ini。它位於 <php-install>/lib 中，例如 /usr/local/php-5.3.6。

新增下列項目：

```
bf_proxyHost=<your_proxy_server_hostname>
bf_proxyPath=<your_proxy_path>
bf_symlinkPath=<symlink_to_proxy_path>
```

## 配置 IHS 的 SSL

### 關於這項作業

除了 IHS 的一般 SSL 設定，若要使用 Build Forge<sup>®</sup>，另外還有其他需求。

### 程序

1. 將 Build Forge<sup>®</sup> 工具目錄併入路徑中。
  - Windows 系統：
    - *bfinstall\ibmjdk\bin*
    - *bfinstall\openssl*
  - Unix 或 Linux 系統：
    - *bfinstall/server/ibmjdk/bin*
2. 將 Build Forge<sup>®</sup> 工具目錄併入程式庫路徑中。

- AIX 系統 (LIBPATH) :
    - *bfinstall*/openssl
  - Unix 或 Linux 系統 (LD\_LIBRARY\_PATH) :
    - *bfinstall*/openssl
3. 將 Build Forge® 金鑰從 PKCS12 轉換成 CMS。使用最新版的 GSKIT 工具。在 gsk7\bin (Windows) 或 bin (Unix 或 Linux) 中，執行下列指令（換行是爲了便於閱讀）：
- ```
gsk7cmd -keydb
        -convert
        -db bfinstall\keystore\buildForgeKeyStore.p12
        -pw buildForgeKeyStore_password
        -old_format pkcs12
        -new_format cms
```
4. 將密碼儲存在隱藏檔中。在啓動期間，IHS 利用這個檔案來取得密碼。如果沒有的話，IHS 發出密碼提示。使用最新版的 GSKIT 工具。在 gsk7\bin (Windows) 或 bin (Unix 或 Linux) 中，執行下列指令（換行是爲了便於閱讀）：
- ```
gsk7cmd -keydb
 -stashpw
 -db bfinstall\keystore\buildForgeKeyStore.kdb
 -pw buildForgeKeyStore_password
```
5. 修改 httpd.conf。若爲 Windows，請併入下列項目。如果您要讓使用者僅利用 HTTPS 來存取，則必須註銷 Listen 80。

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
Listen 80
Listen 0.0.0.0:443
<VirtualHost *:443>
SSLEnable
SSLClientAuth None
SSLProtocolDisable SSLv2
SSLServerCert buildforge
KeyFile bfinstall\keystore\buildForgeKeyStore.kdb
SSLStashFile bfinstall\keystore\buildForgeKeyStore.sth
ErrorLog bfinstall\Apache\logs\ssl_error.log
TransferLog bfinstall\Apache\logs\transfer.log
</VirtualHost>
```

若爲 Unix 和 Linux，請併入下列項目。如果您要讓使用者僅利用 HTTPS 來存取，則必須註銷 Listen 80。

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
Listen 80
Listen 0.0.0.0:443
<VirtualHost *:443>
SSLEnable
SSLClientAuth None
SSLProtocolDisable SSLv2
SSLServerCert buildforge
KeyFile bfinstall/Platform/keystore/buildForgeKeyStore.kdb
SSLStashFile bfinstall/Platform/keystore/buildForgeKeyStore.sth
ErrorLog bfinstall/server/apache/logs/ssl_error.log
TransferLog bfinstall/server/apache/logs/transfer.log
</VirtualHost>
```

*bfinstall* 是安裝目錄的根目錄。針對上述步驟，在 Unix 和 Linux 上，您可能需要將 /Platform 附加到 *bfinstall* 後，才能到達正確的目錄。您也必須使用正斜線 (/) 而非反斜線 (\) 來區隔目錄。

## 結果

如需相關資訊，請參閱 IHS 文件上有關設定 SSL 的內容。

### Windows：取代 DLL 關於這項作業

如果您使用的是 8.0 版以前的 IBM HTTP Server 版本，則必須將 DLL 取代為 Rational® Build Forge® 隨附的 DLL。

#### 程序

在 IBM HTTP Server 安裝目錄的多個位置中找出並取代現有的 libeay32.dll 檔。

- 使用 <bfinstall>/openssl 中提供的 libeay32.dll。
- 搜尋該 DLL 的所有實例，並取代。Windows 2003 中的範例位置：

```
<ihs_install>\Plugins\gsk7\gsk7_32\icc\osslib
<ihs_install>\gsk7\icc\osslib
```

#### 結果

在您取代該 DLL 後，HTTP 伺服器即可正常啟動。若未取代，HTTP 伺服器將無法啟動。

### AIX 系統：其他設定

若為 AIX 系統，需要額外設定。

#### 程序

- AIX 6 系統：
  - 將 <bfinstall>/Platform/his\_modules/libxml2.a 複製到 /opt/IBM/HTTPServer/modules。
  - 將 /opt/IBM/HTTPServer/modules 新增至 LIBPATH。
- AIX 7 系統：
  - 將 <bfinstall>/Platform/his\_modules/libxml2.a 與 <bfinstall>/Platform/his\_modules/libxml2.so.2 複製到 /opt/IBM/HTTPServer/modules。
  - 將 /opt/IBM/HTTPServer/modules 新增至 LIBPATH。
  - 將下面一行新增至 httpd.conf：  
LoadFile "/opt/IBM/HTTPServer/modules/libxml2.so.2"

---

## 第 37 章 使用 API

Build Forge 提供 Java 用戶端 API 和 Perl 用戶端 API。

用戶端檔案會儲存在 `<bfinstall>/webroot/public/clients/` 中。

您可以在執行中的「管理主控台」上存取這些檔案。用戶端下載目錄位於下列 URL：`http://<hostname>:<portnumber>/clients/`

### API 存取 Build Forge

使用 API 的程式會直接和「服務層」通訊。「服務層」是 Apache Tomcat 上的一個應用程式。在安裝期間，會將 Apache Tomcat 伺服器配置成在特定埠上接聽。這些埠必須開啟，API 才能和 Build Forge 通訊。依預設，埠會設定成下列：

- 3966（未受保護的）
- 49150（安全的 - 已啟用 SSL）

API 用戶端的 SSL 配置放在 `bfclient.conf`，必須隨附於用戶端中。當啟用 SSL 時，用戶端必須具備金鑰儲存庫和憑證，才能和 Build Forge 通訊。如需相關資訊，請參閱第 105 頁的『為 API 用戶端（Perl 或 Java）啟用 SSL』。

---

### 建立 API 程式的 Build Forge 使用者

在「管理主控台」上建立使用者，供程式用於鑑別。

#### 關於這項作業

請為 API 程式建立一個用來登入「管理主控台」的使用者。登入以測試使用者是否能運作。

每次程式存取主控台時，它都必須以使用者認證來接受主控台鑑別。

鑑別之後，會為使用者階段作業產生一個階段作業 ID，並儲存在資料庫中。如果程式利用相同的登入作為現有的使用者，便會關閉這個使用者階段作業。

只有一個執行緒或處理程序可以運用使用者認證。如果有另一個執行緒或處理程序，試圖利用相同的使用者認證來建立階段作業，原始階段作業會因而終止。

**註：**請勿使用由 LDAP/Active Directory 鑑別提供的使用者。請在「管理主控台」中建立使用者。

---

### Java 用戶端 API

利用 Java 用戶端 API，可撰寫會存取「管理主控台」的 Java 程式。

以 Java 用戶端 API 建立的程式，是在用戶端主機上執行，並存取「管理主控台」上的資料。Java 用戶端 API 是由 `.jar` 檔組成，其中所含的類別會定義在這些物件上提供作業的「管理主控台」物件方法。

使用 Java 用戶端 API 時，需要搭配 Java SDK 1.5 或 1.6。

JavaDocs 中提供了相關文件。

註：必須在「管理主控台」上定義 Build Forge 使用者，供程式用於鑑別。

## 取得 Java 用戶端 API 套件

您可以從「管理主控台」主機下載 Java 用戶端軟體套件。

### 程序

如果要下載 Java API，請執行下列動作：

1. 存取用戶端下載目錄。在 Web 瀏覽器中，存取下列 URL：  
`http://<hostname>:<portnumber>/clients/`
2. 儲存 JAR 檔。用滑鼠右鍵按一下「Java 用戶端」下面的 **JAR 檔** 鏈結，然後選擇 **另存鏈結**。接著指定用來儲存 JAR 檔的位置。
3. 儲存 JavaDoc。用滑鼠右鍵按一下「Java 用戶端」下面的 **JavaDoc 參照 ZIP** 鏈結，然後選擇 **另存鏈結**。接著指定用來儲存 JAR 檔的位置。

### 結果

您可以透過「管理主控台」存取文件。在「用戶端下載目錄」頁面的「Java 用戶端」下，按一下 **JavaDoc 參照**。

## 設定 Java 用戶端 API

將 Java API 置於用戶端主機上，並設定 Java SDK 來使用它。

### 關於這項作業

該主機會作為「管理主控台」主機的用戶端。

註：啟用 TLS v1.1 或 TLS v1.2 之後，必須以 JDK v1.7 或更高版本來啟動 Java 用戶端，才能使用安全連線連接到 Build Forge<sup>®</sup>。

### 程序

1. 將 .jar 檔放在您要的位置。
2. 更新 CLASSPATH。將 CLASSPATH 設定成包含用來放置 rbf-services-client-java.jar 的目錄。

---

## Perl 用戶端 API

利用 Perl 用戶端 API，可撰寫會存取「管理主控台」的 Perl 程式。

「Perl 用戶端」是一組 Perl 模組，可用來存取「管理主控台」資料物件和方法的摘要。

「Perl 用戶端」模組的文件會以兩種形式包含在用戶端 API 套件中：

- 檔案：apidoc.txt
- 簡明文件 (Plain Old Documentation, POD) 格式的 Perl 文件。如需相關資訊，請參閱線上文件，網址是：<http://www.perl.org>。

如果要使用「Perl 用戶端」，您必須：

- 從「管理主控台」電腦取得「Perl 用戶端」套件。
- 安裝套件（如果尚未安裝，則與 Perl 一起安裝）。

註：必須在「管理主控台」上定義 Build Forge 使用者，供程式用於鑑別。

## 取得 Perl 用戶端 API 套件

您可以從「管理主控台」主機下載 Perl 用戶端 API。

### 程序

如果要下載 Perl 用戶端 API，請執行下列動作：

1. 存取用戶端下載目錄。在 Web 瀏覽器中，存取下列 URL：  
`http://<hostname>:<portnumber>/clients/`
2. 儲存 ZIP 檔。請用滑鼠右鍵按一下「Perl 用戶端」下面的 **ZIP 檔鏈結**，然後選擇 **另存鏈結**。接著指定用來儲存 ZIP 檔的位置。
3. 儲存文件。請用滑鼠右鍵按一下「Perl 用戶端」下面的 **PerlDoc 參照 tar.gz 鏈結**，然後選擇 **另存鏈結**。接著指定用來儲存 ZIP 檔的位置。請解開 zip 檔來存取每一個模組的文件。

### 結果

您可以透過「管理主控台」存取文件。請在用戶端下載目錄頁面上，用滑鼠右鍵按一下「Perl 用戶端」下面的 **PerlDoc 參照**。

## 設定 Perl 用戶端 API

如果要使用 Perl 用戶端 API，您必須在打算執行應用程式的主機上設定它。

### 關於這項作業

該主機會作為「管理主控台」主機的用戶端。

### 程序

1. 在用戶端主機上安裝 Perl 直譯器，例如 ActiveState 的 ActivePerl 5.8.4 版或更新版本。需要下列 Perl 必備模組（ActivePerl 版本 5.8.8 即包含這些模組）：
  - Exporter
  - LWP::UserAgent
  - HTTP::Request如需安裝 Perl 模組的相關資訊，請參閱 Perl 文件。
2. 將下載的 Perl 用戶端 API 套件解壓縮至一個暫存目錄。
3. 依照 apidoc.txt 檔的說明，將 Perl 用戶端 API 安裝成標準 Perl 發行套件。

在 Windows 上，您需要 nmake 1.5，其隨附在 Visual Studio 中，或可從 Microsoft 網站下載。它必須安裝在 PATH 環境變數可以找到的位置，例如 C:\Windows。在解壓縮「Perl 用戶端」套件的暫存目錄中，執行以下指令：

```
perl Makefile.PL
nmake
nmake install
```



在 UNIX 或 Linux 系統上（或是 Windows 的 Cygwin 環境中）：

```
perl Makefile.PL
make
make install
```

安裝完成後，最上層的 Perl 用戶端模組為 `BuildForge::Services::DB0`。如需相關資訊，請參閱 `PerlDoc` 的每一個模組。

---

## 第 38 章 判斷「管理主控台」的版本號碼

如果要找出您正在使用的「管理主控台」版本，請將滑鼠游標置於頁面左上角的 Rational Build Forge 標誌上。系統會在蹦現工具提示中顯示版本號碼。



---

## 第 39 章 診斷

本節協助您判斷 Build Forge 無法正常運作的原因。

請使用下列主題來協助診斷 Rational® Build Forge® 的問題。如果問題仍然存在，請洽詢 IBM® 軟體支援中心以取得協助。您也可以參閱以下的相關主題鏈結，以取得其他資訊。

### 相關概念:

第 178 頁的『日誌的位置與設定』

---

### Build Forge 服務層除錯診斷

這個診斷工具每隔五秒可以接收一次執行緒傾出，以顯示 Java 虛擬機器 (JVM) 中每一個執行緒的執行情形。這個診斷工具是很有用的除錯公用程式，尤其是在發生執行緒死鎖或任何非預期的 JVM 當機情況時。

#### 關於這項作業

**重要：**除非是基於除錯用途而需要執行緒傾出，否則請不要啓用這個診斷工具。這個工具會在極短時間內在檔案系統中產生很大的日誌。

若要啓用和使用服務層除錯診斷工具，請完成下列步驟。

#### 程序

1. 停止應用程式伺服器服務層。
2. 在服務層所參照的類別路徑（例如 tomcat/common/classes 或 tomcat/webapps/rbf-services/WEB-INF/classes）中，建立名為 enableBFSLThreadDump 的新檔案。
3. 啓動應用程式伺服器服務層。啓動時，在 enableBFSLThreadDump 檔所在的同一個目錄中，Build Forge 服務層會寫入名為 threadDump{Timestamp} 的檔案。
4. 若要取得新的執行緒傾出 Snapshot，請如下所示來更新 enableBFSLThreadDump 檔：
  - UNIX/Cygwin：輸入指令 touch enableBFSLThreadDump。
  - Windows：在文字編輯器中開啓檔案，更新檔案，並儲存檔案。

---

### Build Forge 預設日誌

本主題列出 Build Forge® 的預設日誌。請使用這些日誌來檢查各種錯誤和問題。

| 日誌                                                                                     | 說明                  |
|----------------------------------------------------------------------------------------|---------------------|
| <bfinstall>/logs                                                                       | 新建 Java 引擎日誌檔。      |
| <bfinstall>/db.log                                                                     | 資料庫和舊的 Perl 引擎日誌檔。  |
| <bfinstall>/server/apache/logs (UNIX/Linux) 或 <bfinstall>/Apache/logs (Windows)        | 應用程式伺服器 Apache 日誌檔。 |
| <bfinstall>/server/tomcat/logs (Linux/UNIX) 或 <bfinstall>/Apache/tomcat/logs (Windows) | 應用程式伺服器 Tomcat 日誌檔。 |

---

## 配置 Java 引擎日誌

您可以變更 Java 引擎日誌的預設值。

新的 Java 引擎日誌檔位於 `<bfinstall>/logs` 目錄中。每一個日誌檔的預設大小限制為 20 MB，檔案計數上限為五個檔案，預設記載層次為 **INFO**。

若要變更預設值，請使用 `com.ibm.jas-1.0.jar` 中的註解來更新 `logging.properties` 檔。這個 `jar` 檔安裝在應用程式伺服器目錄中。

---

## 啓用 Build Forge 代理程式活動記載

若要擷取 Build Forge 代理程式的資訊，您必須啓用記載功能。

### 開始之前

對代理程式啓用記載功能之前，請確定符合下列必要條件：

- 路徑必須存在於 `bfagent.conf` 中。
- 您具有寫入權可變更代理程式的預設值。

### 關於這項作業

記載資訊會附加到路徑所指定的檔案；檔案大小沒有限制。Build Forge 代理程式活動日誌設定的用途是，暫時用來除錯代理程式。其目的並非用來作為運作中之代理程式的永久日誌。必須手動刪除該日誌檔。

### 程序

1. 若要啓用 Build Forge 代理程式活動記載，請在 `bfagent.conf` 中解除註解含現有路徑的 `activity_log`。
2. 重新啓動代理程式。

---

## 診斷效能問題

如果 Build Forge 的效能有問題，請使用本主題中的資訊來協助診斷並解決問題。

- 資料庫效能 - 檢查資料庫儲存體及資源使用情況。
- 網路通訊 - 檢查 Build Forge 主控台與資料庫和代理程式之間的網路頻寬。
- 安全加強功能 - 檢查任何複雜的演算法。如果您在 Build Forge 主控台和代理程式上已啓用安全，則複雜的演算法可能會影響效能。
- 記載影響 - 檢查進階記載是否已啓用，這可能會影響效能。進階記載只是暫時用於除錯。
- 硬體影響 - 檢查硬體是否變更。變更硬體（例如 CPU、記憶體及儲存體）會影響效能。

---

## 第 40 章 效能和可調整性

系統是設計要在各種負載情況下運作。若要在高負載的情況下執行系統，將必須調整部分配置和資訊配置。

您可以參考下列文章來取得其他資源。另請參閱第 29 頁的第 5 章, 『安裝規劃』。

- Deploying the IBM Rational Build Forge Management Console: Getting results and performance：此文章說明規劃和部署階段，提供資源功能和視情況進行彈性調整的建議。
- Rational Build Forge Performance test results: Evaluating the improved performance in 7.1.2 relative to 7.1.1.4：此文章提供當負載增加時，各種系統功能會如何運作的一般資訊。它會針對不同作業系統和資料庫的組合來提供資訊。也會示範相較於 7.1.1.4 版，7.1.2 版提供了哪些改良的效能。





## 第 41 章 隨著產品一起安裝的可執行指令

下表說明 Rational® Build Forge® 所提供並使用的可執行指令。

在 Windows 上，指令檔位於 Build Forge® 安裝目錄，依預設為 C:\Program Files\IBM\Build Forge。

在 UNIX 和 Linux 上，指令檔位於 *<bfinstall>/Platform* 目錄，其中 *<bfinstall>* 依預設為 /opt/buildforge。

**註：**如果您是在 z/Linux 上執行「管理主控台」，您必須指定 .pl 副檔名才能執行指令。

若要顯示任何可執行指令的版本號碼，請使用 -v 選項。您必須從用來安裝可執行指令的目錄中執行指令。

bfproject -v

任何指令的 -v 選項都會顯示指令名稱和版本號碼，如下列範例所示：

bfproject.exe 8.0.0.0-0-0306

| 執行檔        | 服務？ | 說明                                            |
|------------|-----|-----------------------------------------------|
| bfdbdump   | 否   | bfdbdump.exe 會發出這個指令來清除資料庫。                   |
| bfproject  | 否   | buildforge.exe 會發出這個指令來啟動工作。                  |
| bfengine   | 是   | 這個指令會啟動 buildforge.exe 和 Web 伺服器。僅限於 Windows。 |
| bfexport   | 否   | 使用這個公用程式從資料庫匯出資料。                             |
| bfomexport | 否   | 使用這個公用程式從資料庫匯出 BOM。                           |
| bfimport   | 否   | 使用這個公用程式將專案資料匯入資料庫。                           |
| bfstepcmd  | 否   | bfproject 會針對長時間執行的步驟發出這個指令，以爲其建立個別的程序。       |
| buildforge | 否   | 這個指令會管理建置、清除和排程程序。                            |
| bfdispatch | 是   | 這個指令會啟動代理程式服務。僅限於 Windows。                    |
| bfpwncrypt | 否   | 用來加密密碼的公用程式                                   |



---

## 第 42 章 名詞解釋

這個主題提供整個系統所用的概念和詞彙的定義。

---

### 工作 (job)

執行中專案的實例。系統會儲存每一個已完成工作的資料，包含步驟日誌與 BOM 資料。

---

### 介面 (interface)

介面是配接器範本的實例。您必須建立一個介面（並且編輯它），才能使用配接器。原始配接器範本則維持不變。同時請注意，介面可以包含不只一個 <interface> 元素，每一個元素都是單獨執行的動作。

---

### 引擎 (engine)

系統的元件。引擎利用透過「管理主控台」輸入並儲存在資料庫中的資訊，控制專案的執行、傳送通知電子郵件，以及與代理程式（在伺服器上執行）通訊。

---

### 存取群組 (access group)

共用許可權、通知和 LDAP 群組內容的使用者集合。您可以將存取群組對映至 LDAP 群組。您可以建立巢狀群組。使用者會繼承他們所隸屬之群組的許可權。

---

### 外掛程式 (plug-in)

一種可另行安裝的軟體模組，可在現有的程式、應用程式或介面中新增功能。

---

### 代理程式 (agent)

Build Forge<sup>®</sup> 系統的元件。代理程式必須安裝在您要定義為系統中伺服器資源的任何電腦上。每一個代理程式都會和「管理主控台」通訊，並執行步驟中所定義的指令。代理程式也會組譯因執行步驟產生的輸出，並將它傳回至步驟日誌。

---

### 收集器 (collector)

這個物件用來決定要從伺服器資源收集哪些資訊，以及要指派哪些資訊給伺服器資源。資訊是透過收集器中的內容來指定的。指派給伺服器的收集器，如同伺服器資訊清單的一項規格。您在**伺服器 > 收集器**畫面中定義收集器。

---

## 伺服器 (server)

在 Build Forge 中，伺服器是一個與主機相關聯的物件。它也稱為*伺服器資源*。專案或步驟是在主機上執行。所要使用的伺服器，是由與專案或步驟相關聯的選取元來定義。

如果要在 Build Forge 中設定電腦以作為伺服器，您必須執行下列動作：

- 在電腦上安裝代理程式（請參閱第 131 頁的第 10 章，『安裝代理程式』以取得相關資訊）。
- 使用「管理主控台」建立伺服器資源。

您可以在**伺服器**畫面中定義伺服器資源。

---

## 步驟 (step)

步驟是專案或程式庫的一個元件。它包含一個或數個要執行的指令行。與步驟相關聯的選取元可決定所要使用的伺服器。如果都沒有指定，則會採用專案的選取元。步驟內容會決定如何執行步驟，以及如何處理輸出。您將在建立或編輯專案或程式庫時定義步驟。

---

## 步驟日誌 (step log)

已完成工作中之已完成步驟的相關資料清單。在檢視工作時，步驟日誌會顯示在**步驟**標籤中。每一個步驟的相關資訊會顯示在直欄中。當您選取**工作 > 工作名稱**時，會顯示步驟清單。請按一下某個步驟，以查看其步驟日誌。

---

## 使用者 (user)

系統中的登入。系統會維護它自己的使用者集和許可權設定。在正式作業安裝中，LDAP 用來管理使用者，而 LDAP 項目與群組會對映至系統中。使用者會有相關聯的存取群組，以授與他們特定的許可權來存取系統資源。

---

## 服務 (services)

系統的元件，亦稱為「服務層」，這是因為它可視同用戶端與資料庫之間的抽象層。「用戶端」包含系統本身，以及提供的 Java API 或 Perl API 所建構成的用戶端。

---

## 保存 (archive)

其輸出檔已經刪除但在資料庫中仍保有資料的工作清單。您可以在**工作**畫面中檢視這份清單。

---

## 除去 (clobber)

將專案和其所有相關聯的工作從資料庫中刪除。

---

## 信號交換 (handshake)

Secure Sockets Layer 階段作業啟動時的訊息交換，以便讓用戶端使用公開金鑰技術來鑑別伺服器（以及選擇性地讓伺服器鑑別用戶端），然後讓用戶端和伺服器合力建立對稱金鑰，以使用來加密、解密和偵測竄改情況。

---

## 通知範本 (notification templates)

通知範本可定義在發生特定事件時，傳送給存取群組的電子郵件內容和格式。系統會隨附許多預設範本。您可以編輯範本，或建立專案專用的新範本。

---

## 配接器 (adaptor)

配接器是一個附加程式，可讓 Build Forge 系統與外部系統（例如，來源控制系統、除錯資料庫或測試系統）進行互動。例如，原始碼配接器可讓系統監視和追蹤來源控制系統（例如，IBM® Rational® ClearCase®、Perforce、Visual SourceSafe 及 CVS）中的變更，並且根據這些變更來執行動作。您可以將配接器配置成收集「資料清單 (BOM)」中的儲存體資訊，也可以資訊推送回其他資訊系統。

---

## 執行緒作業 (threading)

各種相關的交易同時執行方式。

---

## 專案 (project)

專案是可執行的工作定義。它由一些步驟構成。其行為是透過專案內容來控制。專案可以有相關聯的選取元，以用來決定可執行它的一或多部伺服器。可為專案指派自己的環境。正在執行的專案就是一項工作。沒有指派選取元的專案就是一個程式庫。

---

## 動態 (dynamic)

在執行時期或處理期間發生的相關事件。

---

## 程式庫 (library)

程式庫是可執行的工作定義。它由一些步驟構成。其行為是透過內容來控制。它與專案的不同之處，在於它沒有選取元可決定它要在哪台伺服器上執行。程式庫是從專案內的一個步驟來加以呼叫。

---

## 資料庫 (database)

資料庫會儲存所有輸入到「管理主控台」的資訊。此外，資料庫還會儲存系統在執行專案或記錄使用者動作時，所建立的資料。

---

## 資訊清單 (manifest)

收集器已收集的伺服器相關資料清單。資訊清單資料供選取元用來選擇伺服器。伺服器的資訊清單會自動更新。您也可以手動更新它們。在**伺服器 > 伺服器名稱**中檢視伺服器時，使用**佇列資訊清單重新整理**按鈕即可進行更新。

---

## 號誌 (semaphore)

系統中的一個廣域旗標，用來避免活動同時進行。每一個號誌各是一個由系統管理的標籤。一般而言，需要專用資源的專案或步驟會取得號誌，以確保能專用資源。

您將使用 **.semget** 指令來設定步驟中的號誌。如果要在個別的步驟中釋放它，請使用 **.semput** 指令。取得號誌之後，其他步驟就無從取得它。試圖取得號誌的步驟，會等到它被釋放為止。

當專案完成時，系統會自動釋放專案所用的任何號誌。在某些情況下，例如，當工作因系統錯誤而結束，並不會釋放號誌。若遇到該情況，可採行手動釋放。

---

## 管理主控台 (Management Console)

安裝在單一電腦上，用以協調系統的系統元件。您可以登入「管理主控台」，來定義和執行專案，並檢視結果和報告。「管理主控台」會發出指示，協助代理程式完成工作。

---

## 輕量型目錄存取通訊協定 (Lightweight Directory Access Protocol, LDAP)

一種開放通訊協定，使用 TCP/IP 來提供支援 X.500 模型之目錄的存取，且不會因而需要較複雜「X.500 目錄存取通訊協定 (DAP)」的資源需求。例如，LDAP 可用來將人員、組織和其他資源定位在「網際網路」或企業內部網路的目錄中。

---

## 選取元 (selector)

此物件與專案或步驟相關聯，它會選取執行專案或步驟的伺服器。選取元中的內容決定如何選取伺服器。選取元可以使用靜態資訊。例如，選取元可以指定伺服器名稱。伺服器也可以使用動態資訊。例如，選取元可指定具有指定內容的伺服器，如 CPU 類型、磁碟大小或現行負載。在執行時期，系統會使用選取元來編譯相符的伺服器清單，並將專案或步驟指派給其中一部伺服器。您可以在**管理 > 選取元**畫面中定義選取元。必須至少先定義一個選取元，才能定義專案。

---

## 靜態 (static)

發生在預先決定或固定時間的相關作業。

---

## 環境 (environments)

環境是變數清單的儲存器。環境可以明確指派給伺服器、專案和步驟。步驟的環境是藉由依序套用伺服器環境、專案環境和步驟環境，建構而成的。如果變數出現在多個環境中，則會採用最後指定的那個值。

---

## 類別 (class)

具有廣域內容的專案的分組。內容用來管理已完成的工作，一般而言，會定期刪除它們，或啟動另一項工作來執行特定的清除作業。

---

## 攔截程式 (interceptor)

Web 服務用來鑑別送入訊息的一種處理程式。在 Build Forge 中，會提供攔截程式來實作單一登入。

---

## BOM

已完成之工作的相關資料清單。BOM 是「資料清單 (Bill of Materials)」的字首語。在檢視工作時，該工作會顯示在 BOM 標籤中，而個別步驟執行的相關資料，則會顯示在「步驟」標籤中。BOM 包含工作步驟的相關資訊，以及針對產生之檔案所做變更的相關資訊。在軟體建置中搭配原始碼配接器是一種常見的用法，在這個情況下，適合審核原始檔的變更。.scan 指令可用來設定原始檔變更的基準線，然後設定檢查點，以彙總自前次執行 .scan 指令以來的變更。

---

## Snapshot

某個時間點時的備份資料記錄。





---

## 附錄. 注意事項

本資訊是針對 IBM 在美國所提供的產品與服務所開發，IBM 在其他國家/地區不見得有提供本文件中所提的各項產品、服務或特性。請洽詢當地的 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。這份文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

這份文件所說明的主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以用書面方式查詢有關授權的問題，來信請寄：

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

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamoto-shi  
Kanagawa, 242-8502 Japan

下列段落若與該國之法律條款抵觸，即視為不適用：IBM 僅以「現狀」提供本書，而不提供任何明示或默示之保證（包括但不限於可售性或符合特定效用的保證）。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

這份資訊中所提及的任何非 IBM 站台只供參考，IBM 不為這些站台提供保證。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

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

如果本程式的已授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

Intellectual Property Dept. for Rational Software  
IBM Corporation  
5 Technology Park Drive  
Westford, MA 01886  
U.S.A.

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之「IBM 客戶合約」、「IBM 國際程式授權合約」或任何同等合約之條款，提供本文件中所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料是在控制環境中得出。因此，在其他作業環境中獲得的結果可能有明顯的差異。在開發層次的系統上可能有做過一些測量，但不保證這些測量在目前一般系統上有相同的結果。再者，有些測定可能是透過推測方式來評估。實際結果可能不同。本文件的使用者應驗證適用於其特定環境的資料。

本書所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

所有關於 IBM 未來方向或計劃的聲明僅代表目標與方針，可能隨時更改或撤消，恕不另行通知。

本資訊僅作規劃用途。在所述之產品上市之前，此處的資訊可能隨時更動。

這份資訊含有日常商業運作所使用的資料和報告範例。爲了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這裡的所有名稱全屬虛構，如與任何實際企業所用的名稱及地址雷同，純屬巧合。

版權：

這份資訊含有原始語言的範例應用程式，用以說明各作業平台中的程式設計技術。貴客戶可以爲了研發、使用、銷售或散佈符合範例應用式所適用的作業平台之應用程式介面的應用程式，以任何形式複製、修改及散佈這些範例程式，不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

這些範例程式的任何部分或任何衍生著作的每一份拷貝，都必須具有下列著作權聲明：

部份程式碼衍生自 IBM Corp. 範例程式。© Copyright IBM Corp. 2003, 2013.

若您是以電子檔檢視此資訊，則照片和彩色圖例可能不會出現。

---

## 商標

IBM、IBM 標誌及 [ibm.com](http://ibm.com) 是 International Business Machines Corp. 的商標或註冊商標，已在全球許多國家或司法管轄區註冊。其他產品及服務名稱可能是 IBM 或其他公司的商標。最新的 IBM 商標清單可從網站上取得，網址爲 [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)。

Adobe、Adobe 標誌、PostScript 及 PostScript 標誌是 Adobe Systems Incorporated 在美國及（或）其他國家或地區的註冊商標或商標。

Intel 是 Intel Corporation 或其子公司在美國及其他國家或地區的商標或註冊商標。

Windows 是 Microsoft Corporation 在美國及（或）其他國家或地區的註冊商標。

Java 及所有以 Java 為基礎的商標是 Sun Microsystems, Inc. 在美國及（或）其他國家或地區的商標。

Linux 是 Linus Torvalds 在美國及（或）其他國家或地區的註冊商標。

UNIX 是 The Open Group 在美國及其他國家或地區的註冊商標。

其他公司、產品或服務名稱，可能是第三者的商標或服務標誌。