

**Prerequisite Scanner**  
1.2 版

**使用手冊**

**IBM**



**Prerequisite Scanner**  
1.2 版

**使用手冊**

**IBM**

**附註**

使用本資訊及其支援的產品之前，請先閱讀第 147 頁的『注意事項』中的資訊。

# 目錄

圖 . . . . .	v
-------------	---

表 . . . . .	vii
-------------	-----

## 第 1 章 Prerequisite Scanner 概觀 . . . 1

Prerequisite Scanner 架構 . . . . .	1
必備項目內容 . . . . .	1
產品型號 . . . . .	12
Prerequisite Scanner 配置檔 . . . . .	13
Prerequisite Scanner 收集器 . . . . .	20
Prerequisite Scanner 評估器 . . . . .	22
輸出格式 . . . . .	23
Prerequisite Scanner Java Developer 工具箱 . . . . .	32
XML 結果檔案的 XML 綱目檔案 . . . . .	32
掃描處理程序 . . . . .	33
此版本更新事項 . . . . .	35

## 第 2 章 安裝 Prerequisite Scanner . . . 37

必備項目 . . . . .	37
安裝壓縮檔 . . . . .	38
解除安裝 Prerequisite Scanner . . . . .	38

## 第 3 章 延伸 Prerequisite Scanner . . . 39

執行 Prerequisite Scanner 之前 . . . . .	39
Windows 系統的必要檢查及延伸作業 . . . . .	39
UNIX 系統的必要檢查及延伸作業 . . . . .	40
新增產品型號 . . . . .	41
建立自訂配置檔 . . . . .	41
新增必備項目內容 . . . . .	43
編輯必備項目內容 . . . . .	44
建立 Windows 系統的自訂收集器 . . . . .	45
建立所有配置檔共用的自訂 VBScript 收集器 . . . . .	45
建立產品及產品版本特定的自訂 VBScript 收集器 . . . . .	47
建立 UNIX 系統的自訂收集器 . . . . .	49
編輯 UNIX 系統的套件測試 Script . . . . .	50
建立 Windows 系統的自訂收集器 . . . . .	51
建立 UNIX 系統的自訂評估器 . . . . .	55

## 第 4 章 正在執行 Prerequisite Scanner 57

prereq_checker . . . . .	57
從指令行執行 Prerequisite Scanner . . . . .	62
共用目錄位置 . . . . .	64

## 第 5 章 Prerequisite Scanner 疑難排解 65

Windows 系統疑難排解 . . . . .	65
UNIX 系統疑難排解 . . . . .	67
執行問題 . . . . .	69
回覆碼 . . . . .	70

## 附錄 A. 產品型號參照 . . . . . 73

## 附錄 B. 配置檔參照 . . . . . 77

## 附錄 C. 必備項目內容參照 . . . . . 81

共用資料內容 . . . . .	82
記憶體必備項目內容及 Tivoli Monitoring 代理程式的系統行為 . . . . .	84
Autonomic Deployment Engine 資料內容 . . . . .	85
連線功能資料內容 . . . . .	86
DB2 資料內容 . . . . .	86
MS SQL Server 資料內容 . . . . .	87
Internet Explorer 資料內容 . . . . .	87
網路資料內容 . . . . .	88
Oracle 資料內容 . . . . .	88
作業系統資料內容 . . . . .	89
已安裝軟體資料內容 . . . . .	99
使用者資料內容 . . . . .	99
Windows 網路資料內容 . . . . .	99
UNIX 網路資料內容 . . . . .	100
環境變數資料內容 . . . . .	100

## 附錄 D. UNIX 系統之預先定義的收集器 103

## 附錄 E. Windows 系統的共用函數 . . . 109

allFiles() . . . . .	110
arrayToString() . . . . .	110
bigthan() . . . . .	111
changeMG() . . . . .	111
checkItemToString() . . . . .	112
dictionaryToString() . . . . .	112
exeCommand() . . . . .	112
filterCommand() . . . . .	113
filterFile() . . . . .	113
findNewest() . . . . .	114
findSuitableFile() . . . . .	114
fmt() . . . . .	115
formatForDisplay() . . . . .	116
formatSizeForDisplay() . . . . .	116
getDecimalSeparator() . . . . .	116
getFirstMatch() . . . . .	117
isMatch() . . . . .	117
notInLatter() . . . . .	118
passOrFail() . . . . .	118
ppread() . . . . .	119
readFile() . . . . .	119
unitMGTOG() . . . . .	120
varToString() . . . . .	120

## 附錄 F. Windows 系統的記載公用程式子常式 . . . . . 123

**附錄 G. Windows 系統的檔案公用程式  
子常式 . . . . . 125**

**附錄 H. Windows 系統的其他共用函數  
及子常式. . . . . 127**

ffirstMatch() . . . . . 127  
getValue() . . . . . 128  
removeSpecialCharacters() . . . . . 128  
versionCompare() . . . . . 129

**附錄 I. UNIX 系統的共用函數 . . . . . 131**

changeMG() . . . . . 131  
AddMG() . . . . . 132  
compare() . . . . . 132  
cutdown() . . . . . 133  
mes4path() . . . . . 133  
mes4Path1() . . . . . 134  
findOSInfo() . . . . . 135  
telnetNFS() . . . . . 135  
NFScheck() . . . . . 136

**附錄 J. UNIX 系統的其他函數. . . . . 137**

formatSizeDisplay() . . . . . 137  
versionCompare() . . . . . 138  
checkHpux() . . . . . 139  
checkLinux() . . . . . 139  
checkSunOS() . . . . . 140  
getValue() . . . . . 140  
setValue() . . . . . 141  
copyValue() . . . . . 141  
getSystemId() . . . . . 141  
getClosestExistingParentDir() . . . . . 142  
parseDirParameter() . . . . . 142  
printDirSize() . . . . . 142

**附錄 K. UNIX 系統的記載公用程式函數 145**

**注意事項. . . . . 147**

**支援資訊及反饋 . . . . . 151**

**索引 . . . . . 153**



1. 在 Windows 系統上輸出至命令行介面 . . . . .	24	9. Prerequisite Scanner 架構及掃描處理程序	34
2. 在 UNIX 系統上輸出至命令行介面 . . . . .	25	10. 在 UNIX 系統上執行 Script 及設定 detail 參數	60
3. precheck.log 檔案 . . . . .	26	11. 在 Windows 系統上執行 Script 及設定 detail 參數 . . . . .	61
4. UNIX 系統上的 prs.debug 檔案 . . . . .	27	12. 具有除錯資料的 precheck.log 檔案 . . . . .	66
5. UNIX 系統上的 prs.trc 檔案 . . . . .	28	13. 不具有除錯資料的 precheck.log 檔案 . . . . .	67
6. Windows 系統上的 result.txt 檔案 . . . . .	29	14. UNIX 系統上的 prs.debug 檔案 . . . . .	68
7. UNIX 系統上的 result.txt 檔案 . . . . .	30	15. UNIX 系統上的 prs.trc 檔案 . . . . .	69
8. Windows 系統上的 result.XML 檔案 . . . . .	31		





---

## 表

1. 代表範圍類型的特殊字元 . . . . .	2	23. Oracle 資料內容 . . . . .	89
2. 必備項目內容範例 . . . . .	3	24. 作業系統資料內容 . . . . .	89
3. 基本必備項目內容種類 . . . . .	4	25. 已安裝軟體資料內容 . . . . .	99
4. 預先定義的子類型 . . . . .	6	26. 使用者資料內容 . . . . .	99
5. 預先定義的限定元 . . . . .	9	27. Windows 網路資料內容 . . . . .	99
6. 支援的資料類型種類及值 . . . . .	15	28. UNIX 網路資料內容 . . . . .	100
7. Windows 之配置檔已掃描的區段 . . . . .	18	29. 環境變數資料內容 . . . . .	101
8. UNIX 之配置檔已掃描的區段 . . . . .	19	30. UNIX 收集器 . . . . .	103
9. 新配置檔 . . . . .	35	31. common_function.vbs 中的函數 . . . . .	109
10. 使用 Windows 系統的配置檔之前的檢查及作業 . . . . .	39	32. 針對每個變數類型呼叫的函數 . . . . .	120
11. 使用 UNIX 系統配置檔之前的檢查及作業 . . . . .	40	33. 記載公用程式子常式 . . . . .	123
12. Prerequisite Scanner Script 的特殊字元圖註 . . . . .	57	34. 檔案公用程式子常式 . . . . .	125
13. 執行問題核對清單 . . . . .	69	35. 檔案公用程式函數 . . . . .	125
14. 預先定義的產品型號 . . . . .	73	36. Windows 系統的其他共用函數及子常式 . . . . .	127
15. 預先定義的配置檔 . . . . .	77	37. 呼叫 ffirstMatch() 的母項函數 . . . . .	127
16. 預先定義的必備項目內容種類 . . . . .	81	38. 使用 getValue() 的 Script . . . . .	128
17. 共用資料必備項目內容 . . . . .	82	39. 呼叫 versionCompare 的母項函數 . . . . .	129
18. Autonomic Deployment Engine 資料內容 . . . . .	85	40. common_function.sh 中的函數 . . . . .	131
19. DB2 資料內容 . . . . .	86	41. 多個檔案中的共用函數 . . . . .	137
20. MS SQL Server 資料內容 . . . . .	87	42. TAD722_impl.sh 中的共用函數 . . . . .	137
21. Internet Explorer 資料內容 . . . . .	87	43. 呼叫 versionCompare 的母項函數 . . . . .	138
22. 網路資料內容 . . . . .	88	44. UNIX 系統上的記載公用程式函數 . . . . .	145



---

## 第 1 章 Prerequisite Scanner 概觀

IBM® Prerequisite Scanner 是一個掃描工具，可在實際部署之前識別、檢查及驗證指定之軟體的必備項目。它會基於針對必備項目內容設定的值掃描軟硬體必備項目。Scanner 會在指令行介面中顯示掃描的結果，並會將結果儲存至文字檔以及 XML 檔案（可選擇）。它還會將參考資訊追蹤及除錯訊息寫入日誌檔中。

Prerequisite Scanner 可檢查機器的作業系統並驗證其是否是適用於指定軟體的正確版本。如果必備項目的任何個別檢查失敗，整體掃描就會失敗。

您可以在安裝之後或任何時間執行 Prerequisite Scanner，以確認您的現行環境。Prerequisite Scanner 不需要您執行要為其檢查必備項目之軟體的安裝程式。

您可以延伸 Prerequisite Scanner，以掃描並非 Scanner 隨附之必備項目檢查核心組一部分的必備項目。

Prerequisite Scanner 依據您的平台呼叫下列類型的 Script：

- Windows：VBScript 及批次
- UNIX：Shell

註：即使您在 Windows 機器上安裝了類似 UNIX 的環境（如 Cygwin），仍無法在 Windows 系統上執行 UNIX Script。

---

### Prerequisite Scanner 架構

IBM Prerequisite Scanner 包含下列主要元件：在指令行介面中執行的 Script、一組用於必備項目檢查的內容、必備項目內容配置檔、必備項目收集器，以及必備項目評估器。執行 Prerequisite Scanner 的結果可用於各種輸出格式。

#### 必備項目內容

必備項目內容是要安裝之產品或解決方案所需之不同軟硬體需求的期望值。必備項目內容範例包括機器上可用磁碟空間總計、機器上未處於使用中的埠組以及現行已安裝的應用程式組。

因為這些必備項目內容的值隨不同產品而變更，所以內容及其值以具有限定元的名稱/值配對表示。其位於必備項目內容配置檔中。每一行只有一個必備項目內容。

必備項目內容必須遵循下列格式：

```
[prefix_identifier.]property_name[.suffix_identifier]=  
[[qualifier_name:qualifier_value]]property_value
```

其中：

- *prefix\_identifier* 是預先定義的必備項目內容種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID。
- *property\_name* 是必備項目內容的名稱。
- *suffix\_identifier* 是必備項目內容子類型的選用 ID，如第 6 頁的表 4 所述。
- *qualifier\_name* 是必備項目內容的選用屬性。IBM Prerequisite Scanner 使用它來限定必備項目內容或要在必備項目內容上執行的檢查類型。

**註：**您可以具有多個限定元，以逗點分隔每一個。限定元組必須由 [] 方括弧括住。

- *qualifier\_value* 是選用屬性的值。每一個限定元及其值必須以 : 冒號分隔。
- *property\_value* 是必備項目內容的值，可以是字串或整數。

取決於資料類型及限定元，必備項目內容可以具有一或多個值，如下所示：

- 單一整數（例如 8080），以代表埠號的值。
- 整數範圍或整數群組（使用表 1 所述的特殊字元表示）。

表 1. 代表範圍類型的特殊字元

特殊字元	說明
*	識別多個值的位置保留元。例如，ports.* 可表示資料庫產品 ports.DB 及 IBM WebSphere® Application Server ports.WAS 的埠超集。
+	識別實際值必須至少符合預期的版本值。例如，os.versionNumber=5.0+ 表示版本必須是 5.0 或之後版本。
-	識別實際值必須至多符合預期版本的值。例如，os.versionNumber=5.0- 表示版本必須是 5.0 或之前版本。
.*	識別實際值可符合預期版本的任何萬用字元值。例如，os.versionNumber=5.*, means that the version can be 5.0, 5.0.1 or 5.5.

**限制：**在 Windows 系統上，僅在 OS Version 必備項目內容中的正規表示式中使用時才支援 \* 萬用字元。

- 可代表下列任何必備項目類型值的字串：
  - 具有單位的數值，例如，8GB 或 10MB
  - 應用程式、作業系統、架構或套件，例如，IBM Lotus Symphony, RedHat Enterprise Linux 5.4, 32-bit 或 ftp

**註：**字串也可能包含由逗點分隔的多個值，例如，應用程式清單。

- 選擇性值，由下列其中一項組合代表，如 True|False, Available|Unavailable 或 Enabled|Disabled

第 3 頁的表 2 說明必備項目內容的範例。

表 2. 必備項目內容範例

必備項目內容	說明
Disk=1GB	<p>可用的磁碟空間數量，其中：</p> <ul style="list-style-type: none"> <li>• <i>property_name</i> 是 Disk</li> <li>• <i>property_value</i> 是 1GB</li> </ul>
user.isAdmin=True	<p>已登入的使用者是否屬於管理者群組，其中：</p> <ul style="list-style-type: none"> <li>• <i>prefix_identifier</i> 是 user, 適用於使用者必要條件內容</li> <li>• <i>property_name</i> 是 isAdmin</li> <li>• <i>property_value</i> 是 True</li> </ul>
network.availablePorts.DB=60000-60005 network.availablePorts.WAS=8080 network.availablePorts.FTP=21	<p>檢查埠 60000-60005 是否可用於資料庫伺服器，埠 8080 是否可用於 WebSphere Application Server 以及埠 21 是否可用於 FTP，其中：</p> <ul style="list-style-type: none"> <li>• <i>prefix_identifier</i> 是 network, 適用於一般必要條件內容</li> <li>• <i>property_name</i> 是 availablePorts</li> <li>• <i>suffix_identifier</i> 是 DB (對於可用的資料庫埠)、WAS (對於可用的 WebSphere Application Server 埠) 以及 FTP 埠</li> <li>• <i>property_value</i> 是 60000-60005、8080 或 21</li> </ul>
os.dir.home=[dir:/home,type:permission]755+	<p>檢查起始目錄是否具有 drwxr-xr-x 權限，其中：</p> <ul style="list-style-type: none"> <li>• <i>prefix_identifier</i> 是 os (對於作業系統必備項目內容)</li> <li>• <i>property_name</i> 是 dir</li> <li>• <i>suffix_identifier</i> 是要檢查之目錄的 home</li> <li>• <i>qualifier_name</i> 是 dir 及 type 用於定義必要條件內容和類型 check</li> <li>• <i>qualifier_value</i> 是 home 及 permission, 限定元的值</li> <li>• <i>property_value</i> 是 755+, 即起始目錄之存取權限的八進位表示法</li> </ul>

對於要為其執行 Prerequisite Scanner 的每個產品，您可以新增或編輯預先定義的必備項目內容。您也可以建立自訂必備項目內容，並視需要使用 Prerequisite Scanner 收集器及評估器來掃描及比較必備項目內容。

#### 相關概念:

第 8 頁的『預先定義的必備項目內容限定元』

IBM Prerequisite Scanner 針對預先定義種類的某些必備項目內容提供一組基本限定元。限定元代表必備項目內容的屬性，Prerequisite Scanner 用於限定必備項目內容或要在該必備項目內容上執行的檢查類型。

#### 預先定義的必備項目內容種類

IBM Prerequisite Scanner 針對不同種類資料提供一組基本的必備項目內容：共用、已安裝軟體、作業系統、使用者、連線功能、Internet Explorer、資料庫伺服器、環境變數以及網路，其中包括 Windows 及 UNIX 的平台專用內容。

<*prefix\_identifier*> 是用於預先定義之必備項目內容種類的 ID。

第 4 頁的表 3 說明預先定義的軟硬體必備項目種類。

表 3. 基本必備項目內容種類

資料種類	說明	必要的字首 ID
共用	此種類檢查共用必備項目，如處理器速度、RAM、磁碟及暫存空間。此範例必備項目內容用於檢查作業系統： OS Version=RedHat Enterprise Linux 5.4	無
已安裝軟體	此種類檢查已安裝軟體必備項目，如 Windows 登錄中註冊的程式以及是否已安裝 cygwin 與 gskit。此範例必備項目內容用於掃描作業系統登錄以獲取已安裝程式及位置： installedSoftware=list_of_installed_programs	無
使用者	此種類檢查使用者必備項目，如已登入使用者是否具有管理權限或是否為 root 使用者。此範例必備項目內容用於檢查已登入使用者是否為管理者群組成員： user.isAdmin=True	user
作業系統	此種類檢查作業系統必備項目，如版本、架構、總記憶體、可用記憶體以及實體記憶體總數。此範例必備項目內容用於檢查遠端登錄服務是否在執行中： os.isServiceRunning.remoteRegistry=True	os
連線功能	此種類檢查連線功能必備項目，如 Telnet 是否在執行中並且 Scanner 可連接到哪些 IP 位址及埠。	無
網路	此種類檢查所有平台可用的網路必備項目，如是否存在可用的埠。此範例必備項目內容用於檢查 8080 埠是否可用於 IBM WebSphere Application Server： network.availablePorts.was=8080	network
Windows 網路	此種類檢查 Windows 網路必備項目，如機器上是否已啟用 NetBIOS 及 DHCP 以及 ping 內容。此範例必備項目內容用於檢查是否至少一個具有有效 IP 位址的配接卡已啟用 NetBIOS 作為通訊協定： network.netBIOSEnabled=True	network
UNIX 網路	此種類檢查 UNIX 網路必備項目，如機器上是否已啟用 NetBIOS 及 DHCP 以及 ping 內容。此範例必備項目內容用於檢查本端主機是否回應 ping 通訊協定： network.pingLocalhost=True	network
Internet Explorer	此種類檢查 Microsoft Internet Explorer 必備項目內容，如版本。此範例必備項目內容用於檢查 Internet Explorer 版本是否為 7.0： internetExplorer.version=7.0	internetExplorer
資料庫伺服器 DB2®	此種類檢查 DB2 必備項目，如版本。此範例必備項目內容用於檢查 DB2 版本是否至少為 9.5： DB2 Version=9.5.*	DB2
資料庫伺服器 Oracle	此種類檢查 Oracle 必備項目，如版本。此範例必備項目內容用於檢查 Oracle 用戶端版本是否至少為 9.2.0.8： oracle.Client=9.2.0.8+	Oracle
環境變數	此種類檢查環境變數必備項目，如是否已設定環境變數。此範例必備項目內容用於檢查類別路徑是否包含 Derby JAR 檔： env.classpath.derbyJAR=False	env

表 3. 基本必備項目內容種類 (繼續)

資料種類	說明	必要的字首 ID
Autonomic Deployment Engine	此種類檢查 Autonomic Deployment Engine 必備項目，如是否已安裝 Autonomic Deployment Engine 以及 Tivoli® Integrated Portal 的安裝單位。此範例必備項目內容用於檢查 Tivoli Integrated Portal 2.1.1.0 版或 2.1.1.1 版安裝單位是否已在 Windows 系統上安裝：  de.installationUnit=regex{.*C37109911C8A11D98E1700061BDE7AEA.* . *TIP 2.1.1.0.* . *TIP 2.1.1.1.*}	de
資料庫伺服器 MS SQL	此種類檢查 MS SQL 必備項目，如版本。此範例必備項目內容用於檢查 MS SQL Server 版本是否為 SQL Server 2008 R2 Developer Edition：  mssql.Server=10.50.1600.1	mssql

### 預先定義的必備項目內容子類型

IBM Prerequisite Scanner 針對預先定義種類的某些必備項目內容提供一組基本子類型。子類型將必備項目內容進一步分類，如依據應用程式、公用程式或服務子類型進行分類。

例如，您可以擁有針對可用網路埠的必備項目內容。您可以將該必備項目內容進一步分類，以檢查用於資料庫伺服器、應用程式伺服器或通訊協定的可用埠。

<suffix\_identifier> 是必備項目內容名稱中子類型的選用限定元。

表 4 說明針對不同必備項目內容種類之預先定義的子類型，包括 <suffix\_identifier>。

表 4. 預先定義的子類型

必備項目內容子類型	字尾 ID	平台	說明	子類型的有效值
<b>不依賴平台的網路種類</b>				
network.availablePorts. <i>app_type</i>	<i>app_type</i>	所有	使用此命名慣例，以檢查是否未接聽埠或埠範圍，或者埠或埠範圍可用於 <i>app_type</i> 應用程式類型。	代表 <i>app_type</i> 的字串，例如： <ul style="list-style-type: none"> <li>• DB2 檢查用於 DB2 資料庫伺服器的埠</li> <li>• WAS 檢查用於 WebSphere Application Server 的埠</li> <li>• ftp 檢查 FTP 埠</li> </ul>
network.portsInUse. <i>app_type</i>	<i>app_type</i>	所有	使用此命名慣例，以檢查是否未接聽埠或埠範圍，或者埠或埠範圍處於使用中（針對 <i>app_type</i> 應用程式類型）。	代表 <i>app_type</i> 的字串，例如： <ul style="list-style-type: none"> <li>• DB2 檢查用於 DB2 資料庫伺服器的埠</li> <li>• WAS 檢查用於 WebSphere Application Server 的埠</li> <li>• ftp 檢查 FTP 埠</li> </ul>
<b>作業系統種類</b>				
os.dir. <i>dir_name</i>	<i>dir_name</i>	UNIX	使用此命名慣例，以檢查 <i>dir_name</i> 檔案系統。必備項目內容的值使用預先定義的限定元。	代表 <i>dir_name</i> 的字串，例如： <ul style="list-style-type: none"> <li>• tmp</li> <li>• home</li> </ul>
os.file. <i>script_name</i>	<i>script_name</i>	UNIX	使用此命名慣例，以檢查 <i>script_name</i> Script 是否在機器上可用。	代表 <i>script_name</i> 的字串，例如： <ul style="list-style-type: none"> <li>• bash</li> <li>• expect</li> <li>• gzip</li> <li>• tar</li> </ul>
os. isService Running. <i>service_name</i>	<i>service_name</i>	Windows	使用此命名慣例，以檢查 <i>service_name</i> 服務是否在機器上處於執行中。	代表 <i>service_name</i> 的字串，例如： <ul style="list-style-type: none"> <li>• remoteRegistry</li> <li>• DNSClient</li> <li>• terminalServices</li> </ul>



表 4. 預先定義的子類型 (繼續)

必備項目內容子類型	字尾 ID	平台	說明	子類型的有效值
os.lib. <i>lib_name_version</i>	<i>lib_name</i> <i>_version</i>	UNIX	使用此命名慣例，以檢查受支援的 <i>lib_name_version</i> 程式庫版本是否已安裝在機器上。	<p>以粗體表示、代表 <i>lib_name_version</i> 的字串，例如：</p> <ul style="list-style-type: none"> <li>• 32 位元 <b>libstdc++.so.#</b> 程式庫</li> <li>• 64 位元 <b>libstdc++.so.#</b> 程式庫</li> <li>• 32 位元 <b>libXft.so.#</b> 程式庫</li> <li>• 32 位元 <b>libXtst.so.#</b> 程式庫</li> <li>• 64 位元 <b>libaio.so.#</b> 程式庫</li> <li>• 32 位元 <b>x1C.rte</b> XLC 執行時期層次</li> <li>• 32 位元 <b>x1C.aix50.rte</b> XLC 版本 (適用於 AIX® 5.3 版)</li> <li>• 32 位元 <b>x1C.aix61.rte</b> XLC 版本 (適用於 AIX 6.1 版)</li> <li>• AIX IOCP <b>bos.iocp.rte</b> 程式庫</li> <li>• <b>bos.loc.iso.en_us</b> : AIX 基本作業系統的 ISO 代碼檔案集</li> </ul> <p>regex {<i>str</i>} : 具有輸入參數 <i>str</i> 的正規表示式，代表程式庫名稱的搜尋型樣，例如： regex {.*libgcc.*}</p> <p>檢查針對該作業系統是否存在 GCC 低層次執行時期程式庫版本 libgcc。</p>

表 4. 預先定義的子類型 (繼續)

必備項目內容子類型	字尾 ID	平台	說明	子類型的有效值
os.package. <i>package_name</i>	<i>package_name</i>	UNIX	使用此命名慣例，以檢查受支援的 <i>package_name</i> 套件版本是否已安裝在機器上。	以粗體表示、代表 <i>package_name</i> 的字串，例如： <ul style="list-style-type: none"> <li>• <b>bash</b> shell</li> <li>• <b>expect</b> (適用於 TCL 延伸套件)</li> <li>• <b>libgcc</b> (適用於 GCC 低層次執行時期套件)</li> <li>• <b>openssh</b> (適用於開放程式碼安全 Shell)</li> <li>• <b>openssl</b> (適用於 SSL/TLS 的開放程式碼工具箱)</li> <li>• <b>perl</b> (適用於 Perl Scripting 套件)</li> <li>• <b>rpm</b> (適用於 RPM 或 RPM Build 套件)</li> <li>• <b>telnet</b> (適用於 Telnet 套件)</li> <li>• <b>wget</b> (適用於 GNU 檔案擷取套件)</li> </ul>
os.space. <i>dir_name</i>	<i>dir_name</i>	UNIX	使用此命名慣例，以檢查指定 <i>dir_name</i> 檔案系統的可用磁碟空間。必備項目內容的值使用預先定義的限定元。	代表 <i>dir_name</i> 的字串，例如： <ul style="list-style-type: none"> <li>• <b>usr</b></li> <li>• <b>home</b></li> <li>• <b>tmp</b></li> <li>• <b>var</b></li> </ul>

## 預先定義的必備項目內容限定元

IBM Prerequisite Scanner 針對預先定義種類的某些必備項目內容提供一組基本限定元。限定元代表必備項目內容的屬性，Prerequisite Scanner 用於限定必備項目內容或要在該必備項目內容上執行的檢查類型。

例如，您可以擁有針對檔案系統的必備項目內容。您可以限定基於該必備項目內容的檔案系統名稱及存取權限屬性，對其執行的檢查。您也可以基於檔案系統路徑及單位屬性檢查可用磁碟空間時要使用的單位類型。

限定元支援自訂作業，以符合您的環境需要並避免 Scanner 對多維度必備項目（如預設路徑及存取權限）的屬性作出隱含假設。您可以變更預先定義的限定元值，但是無法針對預先定義的必備項目內容將新限定元新增至現有的預先定義限定元集。

限定元必須遵循下列格式：

```
[qualifier_name:qualifier_value, qualifier_name:qualifier_value]
property_value
```

其中：

- *qualifier\_name* 是必備項目內容的選用屬性，IBM Prerequisite Scanner 用於限定必備項目內容或要在必備項目內容上執行的檢查類型。
- *qualifier\_value* 是選用屬性的值。

限定元的值也可以是名稱/值配對，以根據使用者類型支援多個有效值。例如，起始目錄的不同路徑取決於其是否為 root 使用者。

- *property\_value* 是必備項目內容的值，可以是字串或整數。

每一個限定元及其值必須以 `:` 冒號分隔。您可以具有多個限定元，以逗點分隔每一個。限定元組必須由 `[]` 方括弧括住。

表 5 說明針對不同必備項目內容種類之預先定義的限定元。部分必備項目內容還使用預先定義的子類型，以進一步分類必備項目內容。

**重要：** 您無法將預先定義的限定元搭配使用其他預先定義的必備項目內容。

表 5. 預先定義的限定元

必備項目內容	平台	說明	有效的限定元及值
<b>具有預先定義之子類型的作業系統種類</b>			
<code>os.dir.dir_name</code>	UNIX	基於下列資格屬性檢查 <code>dir_name</code> 檔案系統： <ul style="list-style-type: none"> <li>• <code>dir</code> 屬性，用於判定要檢查的檔案系統</li> <li>• <code>type</code> 屬性，用於檢查要檢查的檔案系統屬性，例如，該檔案系統之存取權限的 <code>&lt;octal_digits&gt;</code> 八進位表示法</li> </ul> <code>&lt;dir_name&gt;</code> 可代表如下內容： <ul style="list-style-type: none"> <li>• <code>tmp</code></li> <li>• <code>home</code></li> </ul>	具有下列限定元格式的字串： <pre>[dir:dir_name, type:permission] octal_digits+</pre> 例如，若要檢查起始目錄是否具有 <code>drwxr-xr-x</code> 權限： <pre>os.dir.home=[dir:/home, type:permission]755+</pre>

表 5. 預先定義的限定元 (繼續)

必備項目內容	平台	說明	有效的限定元及值
<p>os.space. <i>dir_name</i></p>	<p>UNIX</p>	<p>基於下列一或多個資格屬性檢查指定 <i>dir_name</i> 檔案系統的可用磁碟空間：</p> <ul style="list-style-type: none"> <li>• <i>dir</i> 屬性，用於判定要檢查之檔案系統的路徑</li> <li>• <i>unit</i> 屬性，用於判定要使用的磁碟空間單位</li> </ul> <p><i>dir</i> 屬性的值取決於已登入使用者；因此，該值為名稱/值配對，以代表使用者類型，即 <i>root</i> 或非 <i>root</i> 以及相關聯路徑。</p> <p><i>dir_name</i> 可代表如下內容：</p> <ul style="list-style-type: none"> <li>• <i>usr</i></li> <li>• <i>home</i></li> <li>• <i>tmp</i></li> <li>• <i>var</i></li> </ul>	<p>具有下列限定元格式的字串（適用於 <i>root</i> 使用者的檔案系統）：</p> <pre>[dir:root=<i>dir_path</i>, unit:<i>unit_name</i>] disk_space</pre> <p>例如：</p> <pre>os.space.usr= [dir:root=/usr/ibm/common/acsi, unit:GB]200</pre> <p>具有下列限定元格式的字串（適用於非 <i>root</i> 使用者的檔案系統）：</p> <pre>[dir:non_root=<i>dir_path</i>, unit:<i>unit_name</i>] disk_space</pre> <p>例如：</p> <pre>os.space.home= [dir:non_root=USERHOME/.acsi_HOST, unit:MB]200</pre> <p>具有下列限定元格式的字串（僅使用一個限定元）：</p> <pre>[dir:<i>dir_path</i>] disk_space MB</pre> <p>例如：</p> <pre>os.space.home=[dir:/home/sat]250MB</pre>
<p>不具有預先定義之子類型的作業系統種類</p>			
<p>os.mountcheck</p>	<p>UNIX</p>	<p>基於下列資格屬性檢查是否已裝載檔案系統：</p> <ul style="list-style-type: none"> <li>• <i>drive</i> 屬性，用於判定已裝載檔案系統的目錄</li> <li>• <i>nosuid</i> 屬性，用於判定如果檔案系統已裝載是否已設定裝載選項</li> </ul>	<p>具有下列限定元格式的字串：</p> <pre>[drive:<i>dir_name</i>, mount_option: false true] True False</pre> <p>例如，若要檢查是否已裝載 <i>/home</i> 目錄且未設定 <i>nosuid</i> 選項：</p> <pre>os.mountcheck=[drive:/home, nosuid:false]True</pre>

表 5. 預先定義的限定元 (繼續)

必備項目內容	平台	說明	有效的限定元及值
os.SELinux	Linux	基於下列資格屬性檢查 Security-Enhancement Linux 特性的強制執行狀態： <ul style="list-style-type: none"> <li>source 屬性，用於判定相關作業系統使用的指令</li> </ul>	<ul style="list-style-type: none"> <li>具有下列限定元格式的字串： [source:Command] Disabled Enabled</li> </ul> 例如，若要檢查 Red Hat 或 SUSE 作業系統上是否已停用功能或功能是否具有權限狀態： os.SELinux=[source:Command]Disabled <ul style="list-style-type: none"> <li>不具有限定元的字串，其中作業系統是一般 Linux 變式： os.SELinux=Disabled</li> </ul>
os.ulimit	UNIX	使用此命名慣例，以基於下列資格屬性檢查是否可執行不限數目的處理程序： <ul style="list-style-type: none"> <li>type 屬性，用於判定要檢查的其他限制，例如，filedescriptorlimit 會檢查處理程序可開啓的檔案描述子數目限制</li> </ul>	具有下列限定元格式的字串： [type:limit_name] limit_value, limited unlimited 例如，若要檢查檔案描述子限制是否大於 8192 且具有數目不限的處理程序： os.ulimit=[type:filedescriptorlimit] 8192+,unlimited 要檢查的有效限制類型，其中 limit_name 代表下列限制類型： <ul style="list-style-type: none"> <li>ALL：檢查所有限制</li> <li>corefilesizelimit</li> <li>datasegmentlimit</li> <li>filedescriptorlimit</li> <li>filesizelimit</li> <li>hardlimit</li> <li>processlimit</li> <li>maxmemorysizelimit</li> <li>maxprocesseslimit</li> <li>stacksizelimit</li> <li>threadlimit</li> </ul>
不具預先定義之子類型的共用種類			

表 5. 預先定義的限定元 (繼續)

必備項目內容	平台	說明	有效的限定元及值
Disk	Windows	可用的磁碟空間數量，具有下列選用資格屬性： <ul style="list-style-type: none"> <li>dir 屬性，用於判定要檢查之目錄的路徑</li> <li>unit 屬性，用於判定要使用的磁碟空間單位</li> </ul>	具有下列限定元格式的字串： <pre>[dir:dir_path, unit:unit_name] disk_space</pre> 例如： <pre>Disk= [dir:C:\Program Files\IBM\SQLLIB, unit:MB]1431</pre> 數字格式單位為 MB 或 GB： <pre>&lt;disk_space&gt;MB GB</pre> 例如： <pre>Disk=250MB</pre>

## 產品型號

IBM Prerequisite Scanner 在檔名及參數名稱中使用多字元代碼，以識別產品及元件，並判定要使用的配置檔類型。

### *product\_code*

它是 Windows 或 UNIX 系統上代表產品型號的變數。產品型號識別產品、個別平台（如 Windows、AIX、HP-UX、Linux 及 Solaris）以及該產品支援的作業系統版本（可選擇）。其儲存於 `codename.cfg` 檔案中。任何支援多個平台的產品都具有多個產品型號，每一個均識別產品、平台及作業系統版本（視需要）。

例如，COD、COK 及 COX 產品型號會識別 IBM Tivoli Provisioning Manager 受支援的部分作業系統及版本：

```
COD=Tivoli Provisioning Manager for AIX 6.1
COK=Tivoli Provisioning Manager for HP-UX
COX=Tivoli Provisioning Manager for Windows 2008
```

當執行 Prerequisite Scanner 時，您可以將產品型號及產品版本（選擇性地）作為輸入參數進行傳遞。Scanner 會檢查產品型號是否在 `codename.cfg` 檔中。在 UNIX 系統上，如果找不到型號，則 Scanner 會結束。在 Windows 系統上，如果找不到型號，則它不會結束。

Scanner 隨後會使用輸入參數，在 `ips_root/Windows|UNIX_Linux` 目錄中尋找配置檔。檔名包含與輸入參數相同的產品型號及產品版本。如果未傳遞選用產品版本參數，則 Scanner 會使用在此目錄中找到的最新版本配置檔。Prerequisite Scanner 然後會開始掃描。

註：僅限 Windows 系統，如果產品型號在 `codename.cfg` 檔中不存在，但名稱中具有該產品型號的配置檔存在，則 Prerequisite Scanner 會在輸出中顯示產品型號及版本號碼，而不針對產品名稱定義。

## Prerequisite Scanner 配置檔

個別平台的 IBM Prerequisite Scanner 配置檔包含產品所支援之每個平台的必備項目內容及其期望值。Prerequisite Scanner 提供了預先定義的一組可編輯的配置檔。您必須針對要支援的新產品及平台建立配置檔。

配置檔具有 .cfg 副檔名。您將收集器儲存在 *ips\_root/<OS>* 目錄中，其中 *<OS>* 是作業系統類型的名稱，例如 Windows 或 UNIX\_Linux。

配置檔必須遵循下列規則：

- 副檔名必須是 .cfg
- 檔名的命名慣例：

```
product_code[_<version>].cfg
```

其中：

– *product\_code*

它是 Windows 或 UNIX 系統上代表產品型號的變數。產品型號識別產品、個別平台（如 Windows、AIX、HP-UX、Linux 及 Solaris）以及該產品支援的作業系統版本（可選擇）。其儲存於 *codename.cfg* 檔案中。任何支援多個平台的產品都具有多個產品型號，每一個均識別產品、平台及作業系統版本（視需要）。

– *<version>* 是代表版本、版次、修正及層次的 8 位代碼，其中每兩位數表示代碼的每個部分，例如，7.3.21 為 07032100。

- 區段下的群組必備項目內容，必須遵循區段標題的命名慣例。
- 每個必備項目內容的標準格式是具有選用限定元的名稱/值配對，且每行只有一個內容：

```
[<prefix_identifier>.]<property_name>[.<suffix_identifier>]=  
[[<qualifier_name>:<qualifier_value>]]<property_value>
```

### 沒有區段的配置檔範例

此範例會檢查必備項目內容，但不會區分所需作業系統版本的不同必備項目內容。

```
os.space.var=[dir:root=/var/ibm/common/acsi,unit:MB]1.0  
os.space.usr=[dir:root=/usr/ibm/common/acsi,unit:MB]200  
os.space.home=[dir:non_root=USERHOME/.acsi_HOST,unit:MB]200  
os.space.tmp=30MB  
env.classpath.derbyJAR=False  
network.pingSelf=True  
network.pingLocalhost=True  
network.availablePorts.Derby=4130  
OS Version=RedHat Enterprise Linux 4.*,RedHat Enterprise Linux 5.*  
os.package.compat-libstdc++-33=compat_libstdc++_33  
os.package.libgcc=libgcc-3.4.3-9
```

#### 相關概念:

『配置檔中的區段』

可以將必備項目內容分組到配置檔中的區段組之下，每一個區段代表一個資料類型種類。區段是配置檔中的選用項目。

### 配置檔中的區段

可以將必備項目內容分組到配置檔中的區段組之下，每一個區段代表一個資料類型種類。區段是配置檔中的選用項目。

區段標題的命名慣例：

`[category_name:category_value]`

其中：

- `category_name` 是代表資料類型類型的多字元代碼
- `category_value` 是代表接受之種類值的多字元代碼

註：值可以使用第 2 頁的表 1 中所述的特殊字元。

每一個種類名稱及其值必須以下列方式表示：

: colon and enclosed by [] square brackets.

藉由組合區段標題，您可以具有多個資料類型種類，從而將必備項目內容僅限於該組指定的種類。

`[category_name:category_value][category_name:category_value]`

例如，若要指定適用於執行 32 位元 SUSE Linux Enterprise Server 第 11 版 Itanium 作業系統之機器的必備項目內容：

`[OSType:SUSELinuxEnterpriseServer11][OSArch:64-bit][CPU:Itanium]`

對於所有平台，您可以使用 | 邏輯 OR 符號，以使用選擇性資料類型種類。例如，若要將任何環境變數設為 True，區段標題的組合為：

- **UNIX 系統**

`[@TPAE_DB_FEATURE:True|@TPAE_DIR_FEATURE:True|@TPAE_J2EE_FEATURE:True]`

- **Windows 系統**

`[@TPAE_DB_FEATURE:True]|[@TPAE_DIR_FEATURE:True]|[@TPAE_J2EE_FEATURE:True]`

**重要：** | 邏輯 OR 符號的位置在 Windows 及 UNIX 系統中是不同的。對於 UNIX 系統，區段標題組由一組 [] 方括弧括住，其中每一個區段標題由符號分隔。對於 Windows 系統，符號會使用相關聯的 [] 方括弧分隔每一個完整區段標題。

僅對於 Windows 系統，您可以使用 ! 邏輯 NOT 符號來排除資料類型種類。例如，若要排除 Windows Server 2003 R2 變式，區段標題組合為：`[OSType:Windows Server 2003][!OSType:Windows Server 2003 R2]`

第 15 頁的表 6 說明支援的資料類型種類及相關聯的接受值。



表 6. 支援的資料類型種類及值

資料類型種類	說明	接受值
OSType	作業系統類型	<ul style="list-style-type: none"> <li>• UNIX 指出此種類中的所有內容可用於所有 UNIX 平台，包括 AIX、HP-UX、Linux 及 Solaris，例如： [OSType:UNIX]</li> <li>• AIX 指出此種類中的所有內容可用於所有 AIX 作業系統變式，例如： [OSType:AIX]</li> <li>• HP-UX 指出此種類中的所有內容可用於所有 HP-UX 作業系統變式，例如： [OSType:HP-UX]</li> <li>• LINUX 指出此種類中的所有內容可用於所有 Linux 作業系統變式，例如： [OSType:LINUX]</li> <li>• RedHat 指出此種類中的所有內容可用於所有 RedHat Linux 作業系統變式，例如： [OSType:RedHat]</li> <li>• RedHatEnterpriseLinuxServer 指出此種類中的所有內容可用於所有 RedHat Enterprise Linux Server 作業系統變式，例如： [OSType:RedHatEnterpriseLinuxServer]</li> <li>• SUSE 指出此種類中的所有內容可用於所有 SUSE Linux 作業系統變式，例如： [OSType:SUSE]</li> <li>• SUSELinuxEnterpriseServer 指出此種類中的所有內容可用於所有 SUSE Linux Enterprise Server 作業系統變式，例如： [OSType:SUSELinuxEnterpriseServer]</li> <li>• Solaris 指出此種類中的所有內容可用於所有 Solaris 作業系統變式，例如： [OSType:Solaris]</li> </ul>

表 6. 支援的資料類型種類及值 (繼續)

資料類型種類	說明	接受值
		<ul style="list-style-type: none"> <li>• Windows 指出此種類中的所有內容可用於所有 Windows 作業系統，例如： [OSType:Windows]</li> <li>• Windows 2000 Workstation (5.0.* 版) 指出此種類中的所有內容可用於所有 Windows 2000 作業系統變式， 例如： [OSType:Windows 2000]</li> <li>• Windows XP Workstation (5.1.* 版) 指出此種類中的所有內容可用於所有 Windows XP Professional 32 位 元作業系統變式，例如： [OSType:Windows XP]</li> <li>• Windows XP Workstation (5.2.* 版) 指出此種類中的所有內容可用於所有 Windows XP Professional 64 位 元作業系統變式，例如： [OSType:Windows XP]</li> <li>• Windows Vista Workstation (6.0.* 版) 指出此種類中的所有內容可用於所有 Windows Vista 作業系統變式， 例如： [OSType:Windows Vista]</li> <li>• Windows 7 Workstation (6.1.* 版) 指出此種類中的所有內容可用於所有 Windows 7 作業系統變式，例 如： [OSType:Windows 7]</li> <li>• Windows 2000 Server (5.0.* 版) 指出此種類中的所有內容可用於所有 Windows 2000 Server 作業系統 變式，例如： [OSType:Windows 2000]</li> <li>• Windows Server 2003 (5.2.* 版) 指出此種類中的所有內容可用於 Windows Server 2003 作業系統變 式，例如： [OSType:Windows Server 2003]</li> <li>• Windows Server 2003 R2 (5.2.* 版及其他類型的 OS 說明為 R2) 指出此種類中的所有內容僅可用於 Windows Server 2003 R2 作業系 統變式，例如： [OSType:Windows Server 2003 R2]</li> </ul>

表 6. 支援的資料類型種類及值 (繼續)

資料類型種類	說明	接受值
		<ul style="list-style-type: none"> <li>Windows Server 2008 (6.0.* 版)</li> </ul> <p>指出此種類中的所有內容可用於 Windows Server 2008 作業系統變式，例如：</p> <p>[OSType:Windows Server 2008]</p> <ul style="list-style-type: none"> <li>Windows Server 2008 R2 (6.1.* 版)</li> </ul> <p>指出此種類中的所有內容僅可用於 Windows Server 2008 R2 作業系統變式，例如：</p> <p>[OSType:Windows Server 2008 R2]</p> <ul style="list-style-type: none"> <li>&lt;OS_Name_Version&gt;</li> </ul> <p>指出此種類中的所有內容可用於該版本的作業系統，例如：</p> <p>[OSType:RedHatEnterpriseLinuxServer4.2]</p> <p>註：容許特殊萬用字元 * 指定多個版本。</p>
OSArch	作業系統的架構	<ul style="list-style-type: none"> <li>32 位元，例如： [OSArch:32-bit]</li> <li>64 位元，例如： [OSArch:64-bit]</li> </ul>
CPU	一般處理器系列名稱	Itanium，例如： [CPU:Itanium]
CPUArch	處理器的架構	<p>64 位元 PowerPC® 及 Power Architecture® 處理器的架構，即：</p> <ul style="list-style-type: none"> <li>ppc4</li> <li>POWER4</li> <li>POWER5</li> <li>POWER6®</li> <li>POWER7®</li> </ul> <p>例如： [CPUArch:ppc4]</p>
@<EnvVar_Name>	產品的環境變數	<p>遵循該產品的規則，例如：</p> <p>[@TPAE_DB_SERVER:True]</p>

## 使用區段之 Windows 配置檔範例

此範例使用區段將任何 Windows 機器的必備項目內容分類，然後將執行特定版本 Windows 之機器的必備項目內容分類。

```
#Properties for all Windows operating systems, that is, Windows XP and above
[OSType:Windows]
os.versionNumber=5.1+
network.pingSelf=True
network.pingLocalhost=True
network.availablePorts.Derby=4130
env.CIT.homeExists=True
env.classpath.derbyJAR=False
# Disk space properties
commonPath=10MB
```

```
installPath=200MB
tempPath=30MB
```

```
[OSType:Windows Vista]
os.servicePack=2+
```

當執行 Prerequisite Scanner 時，它會依據機器上已安裝的作業系統及版本掃描並檢查不同的必備項目內容。

例如，表 7 說明包含基於範例檢查之必備項目內容的不同區段。

表 7. Windows 之配置檔已掃描的區段

平台或作業系統	具有必備項目內容的區段
具有 Windows XP 及以上版本的機器	[OSType:Windows]
僅具有 Windows Vista 的機器	[OSType:Windows] [OSType:Windows Vista]

## 使用區段之 UNIX 配置檔範例

此範例包含特定產品之所有平台、個別平台及作業系統版本的必備項目內容。

```
# Properties common to all UNIX platforms
[OSType:UNIX]
os.space.var=[dir:root=/var/ibm/common/acsi,unit:MB]1.0
os.space.usr=[dir:root=/usr/ibm/common/acsi,unit:MB]200
os.space.home=[dir:non_root=USERHOME/.acsi_HOST,unit:MB]200
os.space.tmp=30MB
env.classpath.derbyJAR=False
network.pingSelf=True

# Properties common to all Linux platforms
[OSType:Linux]
os.shell.default=bash
os.SELinux=[source:Command]Disabled
os.package.rpm=rpm

# Properties common to Linux platforms with the ppc64 CPU architecture
[OSType:Linux] [CPUArch:ppc64]
os.package.vacpp.rte=vacpp.rte-9.0.0-5+

# Properties common to all RedHat OS
[OSType:RedHat]
env.classpath.derbyJAR=False

# Properties common to all versions of Red Hat Enterprise
# Linux Server OS
[OSType: RedHatEnterpriseLinuxServer]
network.pingLocalhost=True

# Properties common to all Red Hat Enterprise Linux Server
# OS Version 6.x(6.1,6.2...)
[OSType: RedHatEnterpriseLinuxServer6.*]
os.package.compat-libstdc++-33=compat_libstdc++_33-3.2.3-68

[OSType:RedHatEnterpriseLinuxServer5.*]
os.package.compat-libstdc++-33=compat_libstdc++_33

# Properties common to all Red Hat Enterprise Linux Server
# Version 4.x(6.1,6.2...) OS and for Itanium family CPU
[OSType:RedHatEnterpriseLinuxServer4.*] [CPU:Itanium]
os.package.ia32e1=ia32e1-1.1-20

# Properties common to all Red Hat Enterprise Linux Server
```

```

# Version 4.x(6.1,6.2...) OS and for a 64-bit OS architecture
[OSType:RedHatEnterpriseLinuxServer4.*][OSArch:64-bit]
os.package.libgcc=libgcc-3.4.3-9

# Properties specific to RedHatEnterpriseLinuxServer5.2 OS
[OSType:RedHatEnterpriseLinuxServer5.2]
network.availablePorts.Derby=4130

# Properties specific to a 64 bit SUSE Linux Enterprise Server 11 OS
[OSType:SUSELinuxEnterpriseServer11][OSArch:64-bit]
os.package.libstdc++33-32bit=libstdc++33_32bit-3.3.3-11.9

# Properties specific to a 64 bit SUSE Linux Enterprise Server 11 OS
# and if the environment variable TPAE_DB_Server is set to 'True'
[OSType:SUSELinuxEnterpriseServer11][@TPAE_DB_Server:True]
os.package.libstdc++31-32bit=libstdc++31_32bit

# Properties specific to a 64 bit SUSE Linux Enterprise Server 11 OS
# and if the environment variables TPAE_DB_Server and TPAE_DIR_Server
# are set to 'True'
[OSType:SUSELinuxEnterpriseServer11][@TPAE_DB_Server:True]
[@TPAE_DIR_Server:True]
os.package.libstdc++34-32bit=libstdc++34_32bit

# Properties common to all AIX platforms
os.ulimit=[type:filesize:limit]unlimited
os.ulimit=[type:filedescriptor:limit]8192+,unlimited
os.FreePagingSpace=4GB+

# Properties specific to AIX 5.3.0.0 and
# if the environment variables TPAE_DB_FEATURE or TPAE_DIR_FEATURE
# are set to 'True'
[OSType:AIX5.3.0.0][@TPAE_DB_FEATURE:True|@TPAE_DIR_FEATURE:True]
os.lib.xlC.aix50.rte=xlC.aix50.rte.9.0.0.8+

```

當執行 Prerequisite Scanner 時，它會依據機器上已安裝的作業系統及版本掃描並檢查不同的必備項目內容。

例如，第 18 頁的表 7 說明包含基於範例檢查之必備項目內容的不同區段。

表 8. UNIX 之配置檔已掃描的區段

作業系統及版本	具有必備項目內容的區段
具有 64 位元 SUSE Linux Enterprise Server 11 的機器	[OSType:UNIX] [OSType:LINUX] [OSType:LINUX][CPUArch:ppc64] [OSType:SUSE Linux Enterprise Server 11] [OSArch:64-bit]
具有 Red Hat Enterprise Linux Server 6.3 的機器	[OSType:UNIX] [OSType:LINUX] [OSType:RedHat] [OSType:RedHatEnterpriseLinuxServer] [OSType:RedHatEnterpriseLinuxServer6.*]
具有 SUSE Linux Enterprise Server 11 的機器且環境變數 @TPAE_DB_Server 設為 true	[OSType:UNIX] [OSType:LINUX] [OSType:SUSELinuxEnterpriseServer11][@TPAE_DB_Server:True]
具有 AIX 5.3.0.0 的機器且環境變數 @TPAE_DB_FEATURE 或 @TPAE_DIR_FEATURE 設為 True	[OSType:UNIX] [OSType:AIX] [OSType:AIX5.3.0.0][@TPAE_DB_FEATURE:True @TPAE_DIR_FEATURE:True]

## Prerequisite Scanner 收集器

IBM Prerequisite Scanner 收集器會基於為要安裝之產品設定的必備項目內容，收集現行環境的實際資料。收集器透過原生程式碼取得資料。資料可以是共用資料（如處理器速度及 RAM）、已安裝軟體資料、作業系統資料、使用者資料及連線功能資料。收集器還可延伸，所以您可以建立自訂收集器，以取得自訂必備項目內容的實際值。

Prerequisite Scanner 依據您的平台使用下列語言的收集器：

- Windows：具有 .vbs 副檔名的 VBScript
- UNIX：具有 .sh 或沒有副檔名的 Shell

註：即使您在 Windows 機器上安裝了類似 UNIX 的環境（如 Cygwin），仍無法在 Windows 系統上執行 UNIX Script。

### Windows 系統的收集器

Windows 系統的 VBScript 收集器在 Windows Script 主機環境中執行。這些收集器使用「元件物件模型」，來存取 Windows 環境的元素，例如 FileSystemObject 及 TextStream。

Prerequisite Scanner 會執行 VBScript 收集器，以取得 Windows 環境之必備項目內容的實際值。每個收集器可取得一或多個必備項目內容的資料。

針對 VBScript 收集器中的每個必備項目內容，收集器會將必備項目內容的名稱及其實際值寫為標準輸出。Prerequisite Scanner 會將此標準輸出寫入暫存文字檔，即 localhost\_hw.txt。

您可以建立自訂共用 VBScript 收集器，以收集適用於任何產品及產品版本之必備項目內容的資料。您還可以建立產品特定的自訂收集器，以收集適用於特定產品及產品版本的資料。

當執行 Prerequisite Scanner 時，它會依照下列順序執行收集器：預先定義的 VBScript 收集器；*ips\_root/lib* 目錄中自訂共用 VBScript 收集器；產品特定的自訂 VBScript 收集器；方法是在 *ips\_root/Windows* 目錄中搜尋 *product\_code[<version>].vbs* 檔案。

例如，*env.tcrhome.vbs* 檔案是自訂收集器，可檢查 Tivoli Common Reporting 的起始目錄環境變數。它儲存於 *ips\_root/lib* 目錄中。

VBScript 收集器必須遵循下列規則：

- 自訂共用 VBScript 收集器檔案的命名慣例

其包含可用於任何產品及產品版本（即所有配置檔）的必備項目內容：

*prefix\_identifier.*]property\_name.vbs

其中：

- *prefix\_identifier* 是必備項目內容預先定義之種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID，例如 *env*。
- *property\_name* 是必備項目內容的名稱，例如 *tcrhome*。

將此類型的 VBScript 收集器儲存於 *ips\_root/lib* 目錄中。

- 產品特定自訂 VBScript 收集器檔案的命名慣例

其包含可用於特定產品及產品版本（即一個配置檔）的內容：

```
product_code[_<version>].vbs
```

其中：

– *product\_code*

它是 Windows 或 UNIX 系統上代表產品型號的變數。產品型號識別產品、個別平台（如 Windows、AIX、HP-UX、Linux 及 Solaris）以及該產品支援的作業系統版本（可選擇）。其儲存於 *codename.cfg* 檔案中。任何支援多個平台的產品都具有多個產品型號，每一個均識別產品、平台及作業系統版本（視需要）。

– *<version>* 是代表版本、版次、修正及層次的 8 位代碼，其中每兩位數表示代碼的每個部分，例如，7.3.21 為 07032100。

將此類型的 VBScript 收集器儲存於 *ips\_root/Windows* 目錄中。

- 每個必備項目內容的標準輸出如下：

```
WScript.Echo "property_name=" & <var_for_value>
```

– *property\_name*，代表配置檔中所寫入的必備項目內容，例如 *env.tcrhome*。

– *var\_for\_value*，即收集器為必備項目內容所取得之實際值的 VBScript 變數。

例如，下列標準輸出會寫入 Tivoli Common Reporting 起始目錄環境變數的必備項目內容及其實際值：

```
WScript.Echo "env.tcrhome=" & tcr_home
```

## UNIX 系統的收集器

UNIX 系統的收集器在用於 AIX、HP-UX、Linux 或 Solaris 的相關 Shell 主機環境中執行。它們使用該平台特定的指令及選項，以存取主機環境的元素。

每一個 UNIX 收集器均可取得必備項目內容的資料，或具有預先定義子類型之必備項目內容的資料。收集器會將必備項目內容的檢查結果寫為標準輸出。Prerequisite Scanner 會將此標準輸出寫入暫存文字檔。

您可以建立自訂 UNIX 收集器，以收集自訂必備項目內容的資料。每個預先定義或自訂收集器會在 *ips\_root/UNIX\_Linux/packageTest.sh* 檔案中進行呼叫。

當執行 Prerequisite Scanner 時，它會依照下列順序執行收集器：*ips\_root/lib* 目錄中檔名具有 *\_plug* 的預先定義收集器；*ips\_root/UNIX\_Linux* 目錄中的預先定義收集器；*ips\_root/UNIX\_Linux* 目錄中的自訂 UNIX 收集器。

例如，*installedSoftware.TCR.version* 檔案是自訂收集器，可取得機器上安裝的 Tivoli Common Reporting 版本。它儲存於 *ips\_root/UNIX\_Linux* 目錄中。

UNIX 收集器必須遵循下列規則：

- 沒有副檔名之自訂 UNIX 收集器檔案的命名慣例：

```
[prefix_identifier.]property_name
```

其中：

– *prefix\_identifier* 是必備項目內容預先定義之種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID，例如 *installedSoftware*。

– *property\_name* 是必備項目內容的名稱，例如 TCR.version。

將收集器儲存於 *ips\_root/UNIX\_Linux* 目錄中。請確保其沒有副檔名。

- 必備項目內容的標準輸出，其會傳回必備項目內容的實際值（如果是整數或字串）；例如，已裝載之檔案系統的軟體版本或可用磁碟空間量。或者，它可以傳回 "Unavailable"。

```
echo "True"|"False"
'If the scan checks for the existence of the prerequisite
'property
echo $res
'If the scan checks returns the value, for example, product version,
'of the prerequisite property
echo "Unavailable"
'If the scan returns no value for the prerequisite property
echo "Available"
'If the scan returns a valid check for the prerequisite property
```

- *ips\_root/UNIX\_Linux/packageTest.sh* Script 中呼叫及執行收集器的程式碼。

```
res=`echo $line | grep installedSoftware.TCR.version`
if [ $res ]; then
ExpValue=`echo $res | cut -d "=" -f2`

echo "\`wr\Trace "Starting" "installedSoftware.TCR.version"\`" >>/tmp/prs.check
echo "\`wr\Trace "Executing" "installedSoftware.TCR.version"\`" >>/tmp/prs.check
echo "\`wr\Debug "Starting" "installedSoftware.TCR.version"\`" >>/tmp/prs.check
echo "\`wr\Debug "Expected" "ExpValue" "\`" >>/tmp/prs.check

echo "ss=\.\/installedSoftware.TCR.version\`" >>/tmp/prs.check
echo "\`wr\Trace "Finished" "installedSoftware.TCR.version"\`" >>/tmp/prs.check
echo "echo \"os.userLimits=\$ss\" >>/tmp/prs.check
echo "\`wr\Debug "Finished" "installedSoftware.TCR.version"\`" >>/tmp/prs.check
echo "\`wr\Debug "OutPutValueIs" \$ss\`" >>/tmp/prs.check
echo "\`wr\Trace "Done" "installedSoftware.TCR.version"\`" >>/tmp/prs.check
fi
```

## Prerequisite Scanner 評估器

IBM Prerequisite Scanner 評估器是 Script，可比較收集器的實際資料與配置檔中相同內容的預期資料。評估可以是：平台專用；基於簡式運算符，如小於、等於或大於；基於內容已安裝、存在或已啟用。它們還可以驗證埠是否處於使用中還是可用以及機器的連線狀態。您可以建立或編輯評估器。

Prerequisite Scanner 依據您的平台使用下列語言的評估器：

- Windows：具有 .vbs 副檔名的 VBScript
- UNIX：具有 .sh 副檔名的 Shell

**註：**即使您在 Windows 機器上安裝了類似 UNIX 的環境（如 Cygwin），仍無法在 Windows 系統上執行 UNIX Script。

您將評估器儲存在 *ips\_root/OS* 中，其中 *OS* 是作業系統的名稱，如 Windows 或 UNIX\_Linux。

評估器必須遵循下列規則：

- 檔名命名慣例：

```
[prefix_identifier.]property_name[.suffix_identifier]_compare.vbs|sh
```

其中：



- *prefix\_identifier* 是預先定義的必備項目內容種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID。
- *property\_name* 是必備項目內容的名稱。
- *suffix\_identifier* 是必備項目內容子類型的選用 ID，如第 6 頁的表 4 所述。
- 或者，將兩個輸入參數傳遞到 Script，以進行複式評估：
  - *expected\_value* 即配置檔中設定之必備項目內容的期望值。
  - *actual\_value* 即收集器探索到的機器上必備項目內容的實際值。
- 標準輸出如下：
  - "PASS"：當必備項目內容的期望值等於或大於必備項目內容的實際值。
  - "FAIL"：當必備項目內容的期望值不等於必備項目內容的實際值。

## 輸出格式

IBM Prerequisite Scanner 會產生使用下列屏幕及易懂的檔案格式輸出：輸出至命令行介面、除錯及追蹤日誌以及文字與 XML 結果檔案。

Prerequisite Scanner 將掃描結果及日誌檔儲存至 *ips\_output\_dir* 目錄。您可在執行 Scanner 時使用 **outputDir** 輸入參數設定此目錄。如果未設定此參數，則預設輸出位置為 *ips\_root*。

**註：**Prerequisite Scanner 會在執行時建立暫存檔，但將在 Scanner 完成執行之前刪除這些檔案。這些暫存檔位於 *ips\_output\_dir/temp* 子目錄中。Scanner 也會刪除 *ips\_output\_dir/temp* 子目錄，除非該子目錄包含僅在 UNIX 系統上產生的除錯及追蹤檔案。

如果您選擇從 CD、DVD 或唯讀網路磁碟機執行 Prerequisite Scanner，也可使用此參數指定位置。

**重要：**如果輸出目錄不存在，Prerequisite Scanner 會建立該目錄。您必須具有寫入權限，以建立或寫入 Prerequisite Scanner 儲存檔案的輸出目錄。

## 命令行介面輸出

當執行 Prerequisite Scanner Script 並設定選用 **detail** 參數時，Prerequisite Scanner 會在命令行介面中顯示掃描的詳細結果。詳細結果包含：

- Prerequisite Scanner 的版本
- 執行 Scanner 之作業系統的版本
- 掃描及實務範例的類型
  - 必備項目掃描：實務範例：必備項目掃描
- 為其執行必備項目或性能檢查之產品或元件的名稱
- 針對每個必備項目內容：已檢查之內容的名稱、PASS 或 FAIL 結果、實際值以及期望值
- 針對所有元件：已檢查之一般內容的名稱、PASS 或 FAIL 結果、實際值以及期望值
- 整體 PASS 或 FAIL 結果，以及導致整體掃描失敗的個別檢查失敗

```

C:\prs\precheck_windows_20110927>prereq_checker.bat DMO detail

IBM Prerequisite Scanner
Version      : 1.1.1.8
Build       : 20110927
OS Name    : Microsoft Windows XP Professional Service Pack 3
User Name  : <User Name>

Machine Info
Machine name : <Machine name>
Serial Number: <Serial number>
OS Serial    : <OS serial number>

DMO - Prerequisite Scanner Demo [version 01000000]:

Property          Result Found Expected
=====
OS Version        PASS  Microsoft Win... regex<Windows...
Memory            PASS  645MB 128MB
Disk#1 (C:\ibm\ITM) PASS  1.38GB 1.00GB
os.versionNumber PASS  5.1.2600 5.1.*
os.servicePack    PASS  3.0 2+
os.architecture   PASS  32-bit 32-bit
os.totalPhysicalMemory PASS  3.00GB 2.00GB
os.is8dot3FileFormatEnabled PASS  True True
os.isServiceRunning.terminalServices PASS  True True
os.isServiceRunning.remoteRegistry FAIL  True False
os.isServiceRunning.DNSClient PASS  True True
user.isAdmin      PASS  True True
network.availablePorts.DB PASS  135,445,523,1... 60000-60005
network.availablePorts.WAS PASS  135,445,523,1... 8080
network.availablePorts.FTP PASS  135,445,523,1... 21
network.netBIOSEnabled PASS  True True
network.pingSelf  PASS  True True
network.DHCPEnabled FAIL  True False
cygwinVersion     FAIL  0.0 1.5+

ALL COMPONENTS :
Property          Result Found Expected
=====
Memory            PASS  645MB 128MB
C:                PASS  1.38GB 1.00GB

Prereq Scanner Overall Result: FAIL

Details also available in C:\prs\precheck_windows_20110927\result.txt
C:\prs\precheck_windows_20110927>_

```

圖 1. 在 Windows 系統上輸出至指令行介面

如果您未設定 **detail** 參數，Scanner 僅會在屏幕上顯示 PASS 或 FAIL 結果。

```

root@aqlinux15:~/prs/20110927-0849
File Edit View Terminal Tabs Help
[root@aqlinux15 20110927-0849]# ./prereq_checker.sh DM0
IBM Prerequisite Scanner
  Version: 1.1.1.8
  Build   : 20110927
  OS Name: Linux

Machine Info
Machine Name : <Machine name>
Serial Number: <Serial number>

TPS detected : Red Hat Enterprise Linux Server release 5.5 {32-bit}
Using the DM0 config file
Using config file - /root/prs/20110927-0849/UNIX_Linux/DM0_0750000.cfg for DM0
FAIL
[root@aqlinux15 20110927-0849]#

```

圖 2. 在 UNIX 系統上輸出至指令行介面

Prerequisite Scanner 會依據掃描的結果以及其是否由於錯誤而必須結束產生回覆碼。這些回覆碼會寫入日誌檔。例如，如果 Prerequisite Scanner 由於無法讀取配置檔而無法執行掃描，它會產生回覆碼 2。

### 聚合記憶體及磁碟空間必備項目內容

當您指定多個產品型號作為輸入參數時，您可以執行 Prerequisite Scanner，以同時檢查一或多產品或元件的必備項目。如果相關聯的必備項目內容已在任何配置檔中指定，則 Prerequisite Scanner 會在輸出的下列聚合區段中聚合記憶體及磁碟空間必備項目檢查的結果。

- 在 UNIX 系統上，位於 TOTAL ALL SPECIFIED COMPONENTS 區段
- 在 Windows 系統上，位於 ALL COMPONENTS 區段

一般必備項目內容如下：

- 目標環境中目前可用的實體記憶體總量，即：

#### Memory

- 下列必備項目內容之檔案系統的磁碟空間：

平台	必備項目內容
UNIX 及 Linux	<ul style="list-style-type: none"> <li>• os.space.home</li> <li>• os.space.opt</li> <li>• os.space.tmp</li> <li>• os.space.usr</li> <li>• os.space.var</li> </ul>
Windows	Disk

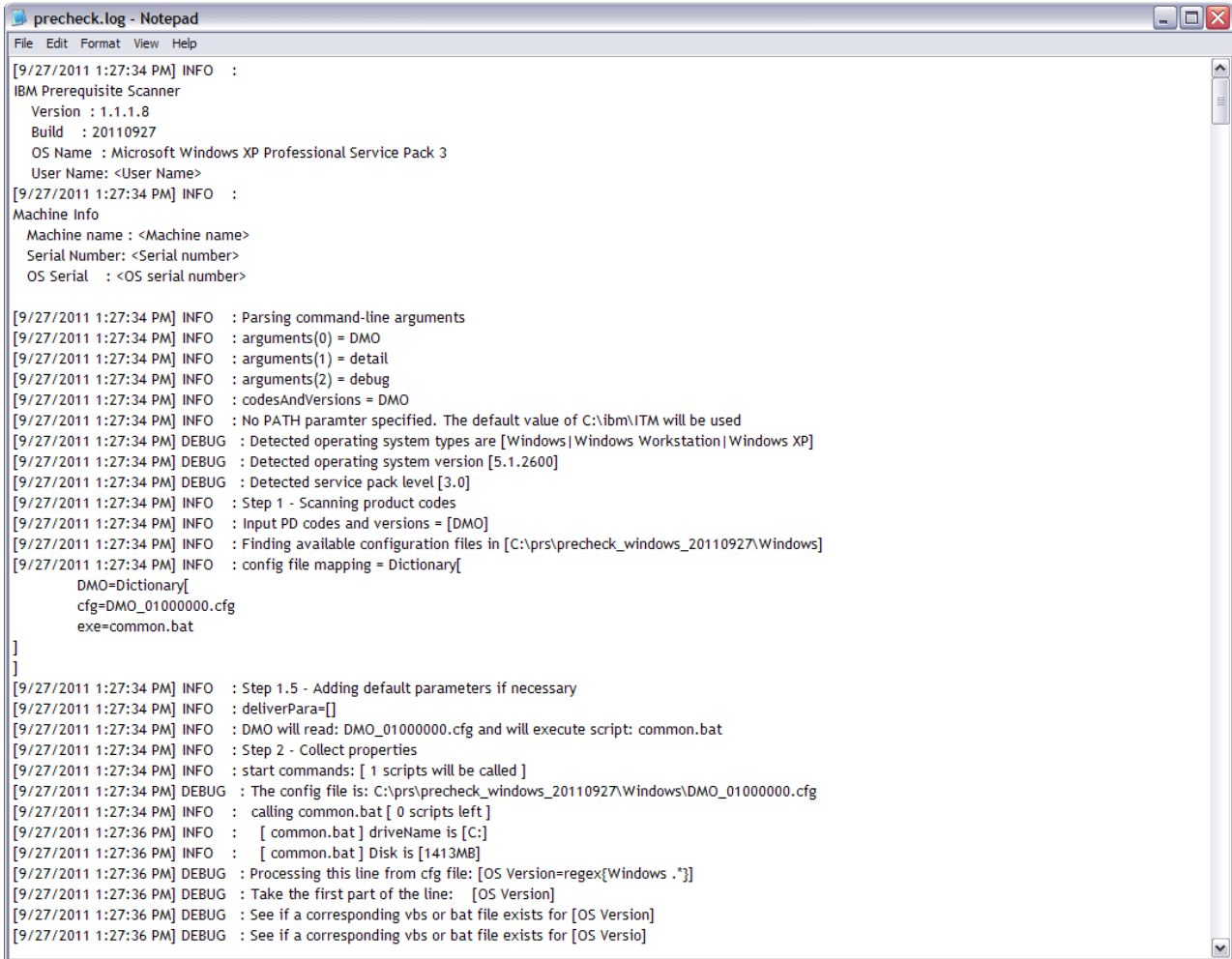
註：如果 opt、usr 及 var 尚未在目標電腦上定義為檔案系統，則 Prerequisite Scanner 不會在對照區段中顯示這些必備項目內容的期望值及傳回的結果。

如果配置檔中既不存在記憶體必備項目內容也不存在磁碟空間必備項目內容，Prerequisite Scanner 不會顯示聚合區段。

Prerequisite Scanner 在掃描結果的聚合區段中處理比較並顯示磁碟空間值，這與主區段不同。請參閱第 31 頁的『輸出中的度量單位』。

## Windows 系統上的除錯日誌檔輸出

Prerequisite Scanner 在 `ips_output_dir/precheck.log` 檔案中輸出處理資訊、警告與錯誤訊息以及掃描結果。當執行 Prerequisite Scanner Script 並設定選用 `debug` 參數時，Prerequisite Scanner 會在此檔案中輸出其他除錯訊息。



```
precheck.log - Notepad
File Edit Format View Help
[9/27/2011 1:27:34 PM] INFO :
IBM Prerequisite Scanner
  Version : 1.1.1.8
  Build : 20110927
  OS Name : Microsoft Windows XP Professional Service Pack 3
  User Name: <User Name>
[9/27/2011 1:27:34 PM] INFO :
Machine Info
  Machine name : <Machine name>
  Serial Number: <Serial number>
  OS Serial : <OS serial number>

[9/27/2011 1:27:34 PM] INFO : Parsing command-line arguments
[9/27/2011 1:27:34 PM] INFO : arguments(0) = DMO
[9/27/2011 1:27:34 PM] INFO : arguments(1) = detail
[9/27/2011 1:27:34 PM] INFO : arguments(2) = debug
[9/27/2011 1:27:34 PM] INFO : codesAndVersions = DMO
[9/27/2011 1:27:34 PM] INFO : No PATH paramter specified. The default value of C:\ibm\ITM will be used
[9/27/2011 1:27:34 PM] DEBUG : Detected operating system types are [Windows|Windows Workstation|Windows XP]
[9/27/2011 1:27:34 PM] DEBUG : Detected operating system version [5.1.2600]
[9/27/2011 1:27:34 PM] DEBUG : Detected service pack level [3.0]
[9/27/2011 1:27:34 PM] INFO : Step 1 - Scanning product codes
[9/27/2011 1:27:34 PM] INFO : Input PD codes and versions = [DMO]
[9/27/2011 1:27:34 PM] INFO : Finding available configuration files in [C:\prsr\precheck_windows_20110927\Windows]
[9/27/2011 1:27:34 PM] INFO : config file mapping = Dictionary[
  DMO=Dictionary[
    cfg=DMO_01000000.cfg
    exe=common.bat
  ]
]
[9/27/2011 1:27:34 PM] INFO : Step 1.5 - Adding default parameters if necessary
[9/27/2011 1:27:34 PM] INFO : deliverPara=[]
[9/27/2011 1:27:34 PM] INFO : DMO will read: DMO_01000000.cfg and will execute script: common.bat
[9/27/2011 1:27:34 PM] INFO : Step 2 - Collect properties
[9/27/2011 1:27:34 PM] INFO : start commands: [ 1 scripts will be called ]
[9/27/2011 1:27:34 PM] DEBUG : The config file is: C:\prsr\precheck_windows_20110927\Windows\DMO_01000000.cfg
[9/27/2011 1:27:34 PM] INFO : calling common.bat [ 0 scripts left ]
[9/27/2011 1:27:36 PM] INFO : [ common.bat ] driveName is [C:]
[9/27/2011 1:27:36 PM] INFO : [ common.bat ] Disk is [1413MB]
[9/27/2011 1:27:36 PM] DEBUG : Processing this line from cfg file: [OS Version=regex{Windows .*}]
[9/27/2011 1:27:36 PM] DEBUG : Take the first part of the line: [OS Version]
[9/27/2011 1:27:36 PM] DEBUG : See if a corresponding vbs or bat file exists for [OS Version]
[9/27/2011 1:27:36 PM] DEBUG : See if a corresponding vbs or bat file exists for [OS Versio]
```

圖 3. `precheck.log` 檔案

## UNIX 系統上的追蹤和除錯日誌檔輸出

當執行 Prerequisite Scanner Script 並設定選用 `debug` 參數時，Prerequisite Scanner 會在 `ips_output_dir/temp/prs.debug` 檔案中輸出詳細處理、警告與錯誤訊息以及掃描結果。

```
root@aclinux15:~/prs/20110927-0849
File Edit View Terminal Tabs Help
Build : 20110927
OS Name: Linux
[2011.09.27 10.12.15 ] [main()] - Entered
[2011.09.27 10.12.15 ] ==== Step 1: Detecting OS...
[2011.09.27 10.12.15 ] OS Detected: Red Hat Enterprise Linux Server release 5.5 {32-bit}
[2011.09.27 10.12.15 ] product_version: DMO
[2011.09.27 10.12.15 ] [AutoOsDetection()] - Entered
[2011.09.27 10.12.15 ] [Param] ProductInfo:DMO
[2011.09.27 10.12.15 ] [Param] OSInfo:Red Hat Enterprise Linux Server release 5.5 {32-bit}
[2011.09.27 10.12.15 ] [Param] CPU Arch:Kernel=i686
[2011.09.27 10.12.15 ] Finding product code in product.cfg
[2011.09.27 10.12.15 ] product code found :
[2011.09.27 10.12.15 ] Found DMO code in product.cfg
[2011.09.27 10.12.15 ] Finding OS Arch and CPU Type
[2011.09.27 10.12.15 ] Found OS Arch = 32-bit, CPU Type=
[2011.09.27 10.12.15 ] Calling config_parser.sh...
[2011.09.27 10.12.15 ] [config_parser.sh] - Entered
[2011.09.27 10.12.15 ] [Param] OSInfo:Red Hat Enterprise Linux Server release 5.5 {32-bit}
[2011.09.27 10.12.15 ] [Param] ProductCode:DMO
[2011.09.27 10.12.15 ] [Param] OSArch:Arch=32-bit
[2011.09.27 10.12.16 ] [Param] CPUArch:CPU=
[2011.09.27 10.12.16 ] [Param] Version:version=
[2011.09.27 10.12.16 ] [Param] XXX:Kernel=i686
[2011.09.27 10.12.16 ] Forming parse array...
[2011.09.27 10.12.16 ] [Form_Parse_String] - Entered
[2011.09.27 10.12.16 ] [Param] OSInfo:Red Hat Enterprise Linux Server release 5.5 {32-bit}
[2011.09.27 10.12.16 ] [Param] ProductCode:DMO
[2011.09.27 10.12.16 ] [Param] OSArch:Arch=32-bit
[2011.09.27 10.12.16 ] [Param] CPU:CPU=
[2011.09.27 10.12.16 ] [Param] CPUArch:Kernel=i686
[2011.09.27 10.12.16 ] Form_Parse_String - ParseArray: [0SType:UNIX][0SType:Linux][0SType:RedHat][0SType:RedHatEnterpriseLinuxServer][0S
Type:RedHatEnterpriseLinuxServer5.*][0SType:RedHatEnterpriseLinuxServer5.5][0SArch:32-bit][CPUArch:i686]
[2011.09.27 10.12.16 ] [Form_Parse_String] - Exit
[2011.09.27 10.12.16 ] Reading config file and parsing using parse array...
[2011.09.27 10.12.16 ] [Read_configFile()] - Entered
[2011.09.27 10.12.16 ] [Param] ConfigFile:/root/prs/20110927-0849/UNIX_Linux/DMO_0750000.cfg-Master
[2011.09.27 10.12.16 ] [Param] Product:DMO
[2011.09.27 10.12.17 ] Writing DBType=Oracle to DMO_0750000.cfg
[2011.09.27 10.12.21 ] Found Env Var - TPAE_DB_Server
```

圖 4. UNIX 系統上的 prs.debug 檔案

當執行 Prerequisite Scanner Script 並設定選用 **trace** 參數時，Prerequisite Scanner 會在 `ips_output_dir/temp/prs.trc` 檔案中輸出追蹤資訊。

```
root@aclinux15:~/prs/20110927-0849
File Edit View Terminal Tabs Help
Build : 20110927
OS Name: Linux
[2011.09.27 10.19.58 ] [main()] - Entered:
[2011.09.27 10.19.58 ] [AutoOsDetection()] - Entered:
[2011.09.27 10.19.58 ] [config_parser.sh] - Entered:
[2011.09.27 10.19.59 ] [Form_Parse_String] - Entered:
[2011.09.27 10.19.59 ] [Form_Parse_String] - Exit:
[2011.09.27 10.19.59 ] [Read_configFile()] - Entered:
[2011.09.27 10.20.05 ] [Read_configFile()] - Exit:
[2011.09.27 10.20.05 ] [config_parser.sh] - Exit:
[2011.09.27 10.20.05 ] [AutoOsDetection()] - Exit:
[2011.09.27 10.20.05 ] [packageTest.sh] - Entered:
[2011.09.27 10.20.25 ] [NFScheck()] - Entered:
[2011.09.27 10.20.25 ] [NFScheck()] - Exit:
[2011.09.27 10.20.25 ] [NFScheck()] - Entered:
[2011.09.27 10.20.25 ] [NFScheck()] - Exit:
[2011.09.27 10.20.25 ] [NFScheck()] - Entered:
[2011.09.27 10.20.26 ] [NFScheck()] - Exit:
[2011.09.27 10.20.26 ] Starting: DBType
[2011.09.27 10.20.26 ] Executing: DBType
[2011.09.27 10.20.26 ] Finished: DBType
[2011.09.27 10.20.26 ] Done : DBType
[2011.09.27 10.20.26 ] Starting: DB2_Version
[2011.09.27 10.20.26 ] Executing: DB2_Version.sh
[2011.09.27 10.20.26 ] Finished: DB2_Version.sh
[2011.09.27 10.20.26 ] Done : DB2_Version
[2011.09.27 10.20.26 ] Starting: DBType
[2011.09.27 10.20.26 ] Executing: DBType
[2011.09.27 10.20.26 ] Finished: DBType
[2011.09.27 10.20.26 ] Done : DBType
[2011.09.27 10.20.26 ] Starting: DBType
[2011.09.27 10.20.26 ] Executing: DBType
[2011.09.27 10.20.26 ] Finished: DBType
[2011.09.27 10.20.26 ] Done : DBType
[2011.09.27 10.20.26 ] Starting: DB2_Version
[2011.09.27 10.20.26 ] Executing: DB2_Version.sh
[2011.09.27 10.20.26 ] Finished: DB2_Version.sh
[2011.09.27 10.20.26 ] Done : DB2_Version
[2011.09.27 10.20.26 ] Starting: DBType
```

圖 5. UNIX 系統上的 prs.trc 檔案

### 文字檔輸出

Prerequisite Scanner 在 `ips_output_dir/result.txt` 檔案中輸出詳細掃描結果。無論您是否設定 **detail** 參數，它都會將結果儲存到文字檔。

```

result.txt - Notepad
File Edit Format View Help

IBM Prerequisite Scanner
Version : 1.1.1.8
Build : 20110927
OS Name : Microsoft Windows XP Professional Service Pack 3
User Name: <User Name>
Machine Info
Machine name : <Machine name>
Serial Number: <Serial number>
OS Serial : <OS serial number>

DMO - Prerequisite Scanner Demo [version 01000000]:

Property          Result Found          Expected
=====
OS Version        PASS  Microsoft Windows XP Professional Service Pack 3  regex[Windows .*]
Memory           PASS  645MB  128MB
Disk#1 (C:\ibm\ITM) PASS  1.38GB  1.00GB
os.versionNumber PASS  5.1.2600  5.1.*
os.servicePack   PASS  3.0  2+
os.architecture  PASS  32-bit  32-bit
os.totalPhysicalMemory PASS  3.00GB  2.00GB
os.is8dot3FileFormatEnabled PASS  True  True
os.isServiceRunning.terminalServices PASS  True  True
os.isServiceRunning.remoteRegistry FAIL  True  False
os.isServiceRunning.DNSClient PASS  True  True
user.isAdmin     PASS  True  True
network.availablePorts.DB PASS  135,445,523,1035,1067,1099,1527,2967,3389,5157,16310,16311,16312,16313,16315... 60000-60005
network.availablePorts.WAS PASS  135,445,523,1035,1067,1099,1527,2967,3389,5157,16310,16311,16312,16313,16315... 8080
network.availablePorts.FTP PASS  135,445,523,1035,1067,1099,1527,2967,3389,5157,16310,16311,16312,16313,16315... 21
network.netBIOSEnabled PASS  True  True
network.pingSelf PASS  True  True
network.DHCPEnabled FAIL  True  False
cygwinVersion    FAIL  0.0  1.5+

ALL COMPONENTS :
Property          Result Found          Expected
=====
Memory           PASS  645MB  128MB
C:               PASS  1415MB  1024MB

Prereq Scanner Overall Result: FAIL

```

圖 6. Windows 系統上的 result.txt 檔案

```

[root@aclinux15 20110927-0849]# cat result.txt
IBM Prerequisite Scanner
  Version: 1.1.1.8
  Build : 20110927
  OS Name: Linux

Machine Info
Machine Name : <Machine name>
Serial Number: <Serial number>

DMO - Prerequisite Scanner Demo [0750000]:
Evaluation          PASS/FAIL          Result              Expected Result
DBType              FAIL               Unknown             Oracle
DBType              FAIL               Unknown             DB2
DBType              FAIL               Unknown             regex{.*Oracle.*}
DBType              FAIL               Unknown             regex{.*DB2.*}
DBTypeDetails      FAIL               Unknown             oracle
DBTypeDetails      FAIL               Unknown             DB2
DBTypeDetails      FAIL               Unknown             regex{.*Oracle.*}
DBTypeDetails      FAIL               Unknown             regex{.*DB2.*}
OS Version          PASS               "Red Hat Enterprise Linux Server release 5.5 (Tikanga)" "regex(R
ed Hat.*Tikanga.*)"
os.lib.libstdc++    PASS               /usr/lib/gcc/i386-redhat-linux/4.1.1/libstdc++.so libstdc++
os.lib.libgcc       PASS               /usr/lib/gcc/i386-redhat-linux/3.4.6/libgcc_s.so [CheckPackage:Tr
ue] regex{libgcc.*}
os.lib.libXp        PASS               /usr/lib/libXmu.so.6          regex{libX.*}
os.space.var        PASS               "38GB"                        "[dir:root=/var/ibm/
common/acsi]"
os.space.usr        PASS               "38GB"                        unit:MB]1.0
common/acsi"        "[dir:root=/usr/ibm/
os.space.tmp        PASS               36GB                          unit:MB]200
env.classpath.derbyJAR PASS               False                          30MB
network.pingSelf    PASS               True                            False
env.classpath.derbyJAR PASS               False                          True

```

圖 7. UNIX 系統上的 result.txt 檔案

## XML 檔案輸出

當指定選用 `xmlResult` 輸入參數時，Prerequisite Scanner 會在 `ips_output_dir/result.xml` 檔案中輸出詳細掃描結果。您可以使用它來將結果輸出至 XML 結果檔案（除純文字測試結果檔案以外）。無論您是否設定 `detail` 參數，它都會將結果儲存到 XML 檔案。



```

<PRSInfo>
<MachineInfo>
  <MachineName>my_machine_name</MachineName>
  <MachineSerialNumber>serial_number</MachineSerialNumber>
  <MachineOSSerial>os_serial_number</MachineOSSerial>
  <MachineOSName>Microsoft Windows XP Professional Service Pack 3</MachineOSName>
</MachineInfo>
<UserInfo>
<ProductInfo>
  <ProductElement>
    <ProductCode>DMO</ProductCode>
    <ProductName>Prerequisite Scanner Demo</ProductName>
    <ProductVersion>01000000</ProductVersion>
  </ProductElement>
</ProductInfo>
<DetailedResults>
  <DetailedProductResultsElement>
    <ProductCode>DMO</ProductCode>
    <ResultElement>
      <PropertyName>OS Version</PropertyName>
      <Result>FAIL</Result>
      <Found>Microsoft Windows XP Professional Service Pack 3</Found>
      <Expected>Windows 7 Ultimate</Expected>
    </ResultElement>
    <ResultElement>
      <PropertyName>Memory</PropertyName>
      <Result>PASS</Result>
      <Found>960MB</Found>
      <Expected>128MB</Expected>
    </ResultElement>
    <ResultElement>
      <PropertyName>Disk#1 (C:\ibm\ITM)</PropertyName>
      <Result>PASS</Result>
      <Found>22072MB</Found>
      <Expected>1GB</Expected>
    </ResultElement>
    <ResultElement>
      <PropertyName>os.versionNumber</PropertyName>
      <Result>FAIL</Result>
      <Found>5.1.2600</Found>
      <Expected>5.2.*</Expected>
    </ResultElement>
  </DetailedProductResultsElement>
</DetailedResults>

```

圖 8. Windows 系統上的 result.XML 檔案

開發人員可使用 Prerequisite Scanner Java™ Developer 工具箱，以剖析及讀取 XML 檔案。

### 輸出中的度量單位

Prerequisite Scanner 在掃描結果的聚合區段中處理比較並顯示磁碟空間值，這與主區段不同。

在掃描結果的主區段中，Prerequisite Scanner 會處理比較並顯示磁碟空間值，如下所示：

平台	必備項目內容
UNIX 及 Linux	如果實際值大於 1024 MB，Prerequisite Scanner 會轉換並傳回單位為 GB 的值；否則，它會傳回單位為 MB 的值。

平台	必備項目內容
Windows	Prerequisite Scanner 使用配置檔中期望值的單位作為比較及顯示單位。它會視需要將實際值轉換為此單位。

在掃描結果的聚合區段中，Prerequisite Scanner 會處理比較並顯示磁碟空間值，如下所示：

平台	必備項目內容
UNIX 及 Linux	如果實際值大於 1024 MB，Prerequisite Scanner 會轉換並傳回單位為 GB 的值；否則，它會傳回單位為 MB 的值。
Windows	Prerequisite Scanner 基於 MB 單位執行磁碟空間的比較。針對包含 Disk 必備項目內容的每一個配置檔，Prerequisite Scanner 會將實際值及期望值轉換為 MB 並執行比較。出於顯示之目的，如果聚合的實際值大於 1 GB，Prerequisite Scanner 會傳回單位為 GB、具有 2 位精準度的實際值；否則，會傳回單位為 MB 的值。

## Prerequisite Scanner Java Developer 工具箱

Prerequisite Scanner Java Developer 工具箱是一組 API，容許您作為開發人員根據需要以程式設計方式剖析及讀取結果 XML 檔案的內容；例如，剖析掃描結果以在安裝程式中使用。

工具箱提供下列套件：

- `com.ibm.prs.common.exception`

其中包含 `PRSApiException` 類別，可提供方法以擲出 XML 查詢 API 的異常狀況。

- `com.ibm.prs.common.reports.api`

其中包含 `PRSXmlResultReader` 介面，可定義 XML 結果檔案的 XML 查詢 API。

- `com.ibm.prs.common.reports.api.impl`

其中包含實作 `PRSXmlResultReader` 的 `PRSXmlResultReaderImpl` 類別。

Prerequisite Scanner 可根據 `ips_root/PRResults.xsd` XML 綱目檔案驗證格式及結構。

Javadoc 可用於 `ips_root/api/javadoc` 目錄中的工具箱。

## XML 結果檔案的 XML 綱目檔案

Prerequisite Scanner 提供 XML 綱目檔案，可以比對它來驗證結果 XML 檔案。

XML 綱目檔案包含下列代表一些區段的元素：

- `PRInfo` 管理 Prerequisite Scanner 詳細資料
- `MachineInfo` 管理在其上執行掃描之目標環境的相關資訊
- `UserInfo` 管理執行掃描之已登入使用者的相關資訊

- ScenarioInfo 管理掃描及實務範例類型的相關資訊
- ProductInfo 管理產品或元件及其配置檔的相關資訊
- DetailedResults 管理依據 DetailedProductResultsElement 分組之產品或元件的每一組必要內容的掃描結果
- AggregateResults 管理累計的磁碟空間和記憶體掃描結果
- OverallResult 管理掃描的整體 PASS 或 FAIL 結果

XML 綱目的名稱和位置為：*ips\_root/PRSResults.xsd*

身為開發人員或部署人員，您可以從查詢 XML API 呼叫方法來驗證結果 XML 檔案。Javadoc 可用於 *ips\_root/api/javadoc* 目錄中的工具箱。

---

## 掃描處理程序

當執行 IBM Prerequisite Scanner 時，它會在掃描處理程序的每個階段執行一組作業。使用者開啓命令行介面並使用一組包括產品型號的輸入參數執行 Prerequisite Scanner Script。

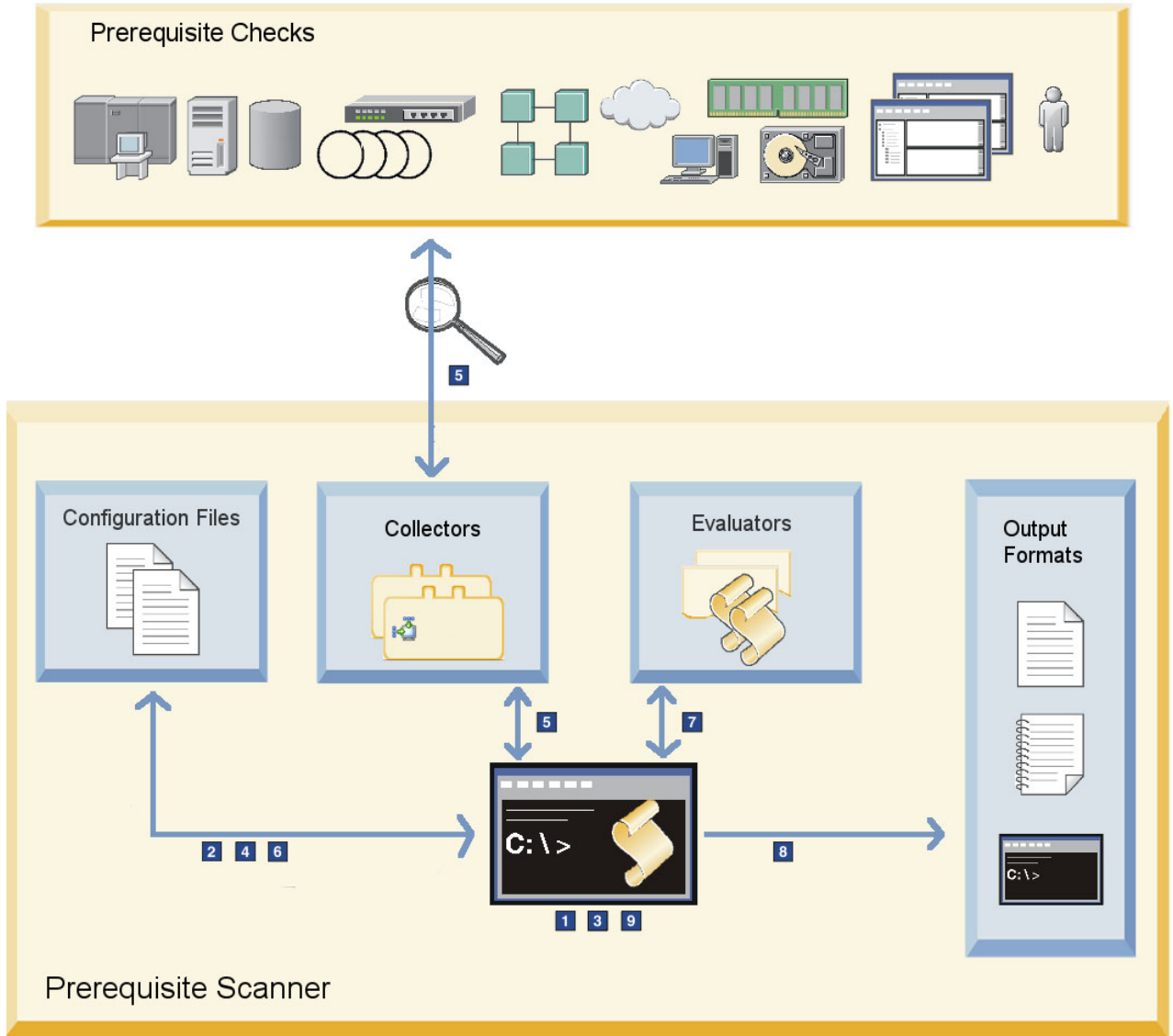


圖 9. Prerequisite Scanner 架構及掃描處理程序

圖 9 中掃描處理程序摘要如下：

1. Prerequisite Scanner 會驗證傳遞到 Scanner 之輸入參數的格式。
2. Scanner 會判定作為其中一個輸入參數的產品型號是否為 codename.cfg 檔案中的有效產品型號。
3. Scanner 會搜尋與產品型號相關聯的配置檔。如果未傳遞選用產品版本參數，Scanner 會使用在 `ips_root/Windows|UNIX_Linux` 目錄中找到的最新版本配置檔。
4. Scanner 會驗證機器的實際作業系統是否為支援的作業系統。Scanner 會將實際作業系統與配置檔中區段標題中之預期支援的作業系統比較，該配置檔的檔名包含相同的產品型號及產品版本作為輸入參數。
5. Scanner 使用 Prerequisite Scanner 收集器收集必備項目檢查的實際必備項目內容。
6. Scanner 會檢查與產品型號及產品版本相關聯之配置檔中的必備項目內容。

Scanner 會將實際作業系統與 OS 版本必備項目內容中預期支援的作業系統或配置檔中的區段標題相比較，該配置檔的檔名包含相同的產品型號及產品版本作為輸入參數。

7. Scanner 會讀取配置檔的必備項目內容，並分析必備項目內容的實際值與期望值以進行必備項目檢查。它會在必要時使用 Prerequisite Scanner 評估器。
8. Scanner 會將掃描的結果輸出至指令行介面、結果文字及 XML 檔案以及易懂的日誌檔。
9. Scanner 會清除及移除暫存檔與目錄。

---

## 此版本更新事項

IBM Prerequisite Scanner 1.2 版提供新的內容及加強功能。它還包含問題的修正程式。

### 此版本新增特性

可以剖析及讀取新掃描結果 XML 檔案。

Prerequisite Scanner Java Developer 工具箱是一組 API，容許開發人員根據需要以程式設計方式剖析及讀取結果 XML 檔案的內容；例如，剖析掃描結果以在安裝程式中使用。請參閱 第 32 頁的『Prerequisite Scanner Java Developer 工具箱』。

### 此版本新增配置檔

表 9 說明 Prerequisite Scanner 1.2 版隨附的新配置檔及產品型號

表 9. 新配置檔

產品或元件	產品型號	配置檔
Tivoli Composite Application Manager Agent for WebSphere MQ	KMQ	<i>ips_root/Windows UNIX_Linux/KMQ_07010000.cfg</i>
Tivoli Composite Application Manager Agent for WebSphere Message Broker	KQI	<i>ips_root/Windows UNIX_Linux/KQI_07010000.cfg</i>

### 此版本新增必備項目內容

已新增 `os.SeaMonkeyVersion` 內容來檢查機器上的 Mozilla SeaMonkey 版本。請參閱第 89 頁的『作業系統資料內容』。

已新增 `env.var.set.env_var_name` 內容來檢查是否已在機器上設定 `env_var_name` 所指定的環境變數。請參閱第 100 頁的『環境變數資料內容』。

### 此版本的加強功能

可以將掃描結果寫入 XML 檔案。

`ips_output_dir/result.xml` 是 XML 格式的新掃描結果檔案。依預設，工具僅會將結果輸出至純文字結果檔案。請參閱第 23 頁的『輸出格式』。

`xmlResult` 是 Prerequisite Scanner 1.2 版中 Prerequisite Scanner Script 的新選用輸入參數。您可以使用它來將結果輸出至 XML 結果檔案（除純文字測試結果檔案以外）。請參閱第 57 頁的『prereq\_checker』。

如果配置檔中既不存在記憶體必備項目內容也不存在磁碟空間必備項目內容，則移除結果中的聚合區段。

當配置檔中既不存在記憶體必備項目內容也不存在磁碟空間必備項目內容時，Prerequisite Scanner 不會在結果檔案中顯示聚合區段。請參閱第 23 頁的『輸出格式』。

#### 此版次淘汰的特性

無

#### 此版次修正的問題

若要取得本版本已修正問題報告的清單，請參閱當您解壓縮 Prerequisite Scanner 套裝軟體的內容時，*ips\_root* 目錄中的 *Readme.html* 檔。

#### 本修改中的文件變更

《Prerequisite Scanner 使用手冊》不再與 Prerequisite Scanner 的 Prerequisite Scanner 平台套件組合在一起。您可以使用 IBM Prerequisite Scanner 資訊中心。

## 第 2 章 安裝 Prerequisite Scanner

IBM Prerequisite Scanner 沒有安裝程式。當您擷取壓縮檔的內容時，核心檔案位於根目錄中，其具有下列子目錄：`/api` 用於 Prerequisite Scanner Java Developer 工具箱以支援查詢 XML API，`/lib` 用於收集器及共用 Script；`/Windows` 用於 Windows 上的評估器及配置檔；`/UNIX_Linux` 用於 UNIX 平台上的評估器及配置檔；`/licenses` 用於授權檔。

### 必備項目

IBM Prerequisite Scanner 可在 Windows 系統、Windows XP 或以上系統、32 位元或 64 位元上執行。它還可以在各種 AIX、HP-UX、Linux 及 Solaris 作業系統上執行。

請確保您的目標環境中已安裝或可用下列公用程式：

目標系統	必備項目
Windows	<ul style="list-style-type: none"><li>• Telnet 用戶端已啓用，以便預先定義之「連線功能」收集器中的連線功能檢查可正確運作。</li><li>• 請確保已安裝 Microsoft .Net Framework 1.0 或更新版本，以便 Prerequisite Scanner 可以正確運作。</li></ul>
UNIX	<ul style="list-style-type: none"><li>• 已安裝 Bash，以便 Prerequisite Scanner UNIX 收集器可正確運作。</li><li>• 對於非 root 使用者，<code>mount</code>、<code>swapinfo</code> 及 <code>psrinfo</code> 指令的位置必須在 PATH 環境變數中設定，以便指令可用於 Prerequisite Scanner。指令位於 <code>/usr/sbin</code> 目錄中；例如，如下所示設定 PATH 環境變數： <pre>export PATH=\$PATH:/usr/sbin/</pre></li><li>• 確保正確的存取權限指派給 <code>lscfg</code> 指令，包括 <code>setuid</code> 位元之類的存取權旗標設定的任何特定的權限。正確的存取權限表示 Prerequisite Scanner 可執行指令及擷取系統資訊。指令位於 <code>/usr/sbin</code> 目錄中；例如，若要設定 <code>lscfg</code> 的 <code>setuid</code> 位元，請如下所示執行 <code>chmod</code> 指令： <pre>chmod 4777 /usr/sbin/lscfg</pre></li></ul>

Prerequisite Scanner 支援為其執行 Prerequisite Scanner 之指定產品或 IBM 解決方案的所有硬體及作業系統。

---

## 安裝壓縮檔

您可以擷取 IBM Prerequisite Scanner 之壓縮檔的內容。您必須具有在其中擷取壓縮檔內容之根目錄的寫入權限。

### 程序

1. 開啓 Web 瀏覽器並輸入 IBM Fix Central 的 URL。確保您登入 IBM.com or IBM 支援中心入口網站。
2. 從**產品群組**清單選取 **Tivoli**。
3. 從**產品**清單選取 IBM Prerequisite Scanner。
4. 從**已安裝的版本**清單選取您要下載的版本。
5. 從**平台**清單選取要安裝 Prerequisite Scanner 的平台。
6. 按一下**繼續**。會開啓識別修正程式頁面。
7. 使用預設選項**瀏覽修正程式**，然後按一下**繼續**。
8. 在「選取修正程式」頁面上選取套件，然後按一下**繼續**。
9. 在「下載選項」頁面上選取下載選項，然後按一下**立即下載**。
10. 將壓縮檔的內容擷取到您偏好的位置，如 *ips\_root* 所指定。

### 下一步

請確保您已檢查產品的安裝文件或技術說明，以獲取執行 Prerequisite Scanner 之前必須執行的其他步驟。例如，您可能需要設定環境變數，以指示 Prerequisite Scanner 要在目標電腦上安裝哪些元件或特性，並且要檢查哪些必備項目。

---

## 解除安裝 Prerequisite Scanner

如果您想要安裝更新的版本、將它移至另一個環境，或是它是您不再需要的版本，請移除 IBM Prerequisite Scanner。

### 程序

1. 開啓 *ips\_root* 目錄。
2. 刪除目錄及其內容。



## 第 3 章 延伸 Prerequisite Scanner

IBM Prerequisite Scanner 提供一組基本收集器、評估器及配置，您可以用於執行工具及掃描必備項目。如果基本的檔案集、必備項目內容及值以及必備項目檢查不符合您的需求，則您可以延伸 Prerequisite Scanner。

### 執行 Prerequisite Scanner 之前

在執行 IBM Prerequisite Scanner 之前，請判定預先定義的必備項目內容、其期望值以及配置檔是否符合您的必備項目掃描需求。如果任一項不符合您的需求，則您可以執行一組必備項目作業，以配置或延伸 Prerequisite Scanner。必備項目檢查及作業集取決於平台及必備項目檢查的數量。

### Windows 系統的必要檢查及延伸作業

在執行 IBM Prerequisite Scanner 之前，您應該執行一組檢查及作業。這些檢查會判定您可以編輯及使用現有配置檔還是必須延伸 Prerequisite Scanner。

表 10 提供要執行之檢查及作業的清單。

表 10. 使用 Windows 系統的配置檔之前的檢查及作業

	檢查	作業
<input type="checkbox"/>	檢查產品、其支援的作業系統以及作業系統版本是否在 codename.cfg 檔案中列出。	<ul style="list-style-type: none"><li>如果是，則執行下一項檢查。</li><li>否則，將產品的產品型號、個別作業系統及作業系統版本新增至該檔案。如需相關資訊，請參閱第 41 頁的『新增產品型號』。</li></ul>
<input type="checkbox"/>	檢查與產品版本相關聯之產品型號的配置檔是否存在。	<ul style="list-style-type: none"><li>如果是，則執行下一項檢查。</li><li>否則，建立配置檔以包含該作業系統以及作業系統版本的必備項目內容。如需相關資訊，請參閱第 41 頁的『建立自訂配置檔』。</li></ul>
<input type="checkbox"/>	開啓配置檔並檢查其是否包含正確的必備項目內容。	<ul style="list-style-type: none"><li>如果是，則執行下一項檢查。</li><li>否則，新增必備項目內容。如需相關資訊，請參閱第 43 頁的『新增必備項目內容』。</li></ul>
<input type="checkbox"/>	檢查必備項目內容是否具有期望值。	<ul style="list-style-type: none"><li>如果是，則執行 Prerequisite Scanner。如需相關資訊，請參閱第 57 頁的第 4 章，『正在執行 Prerequisite Scanner』。</li><li>否則，編輯必備項目內容。如需相關資訊，請參閱第 44 頁的『編輯必備項目內容』。</li></ul>
<input type="checkbox"/>	針對任何新必備項目內容，檢查預先定義的收集器是否可收集必備項目內容的實際值。	<ul style="list-style-type: none"><li>如果是，則執行下一項檢查。</li><li>否則，建立自訂收集器。如需相關資訊，請參閱第 45 頁的『建立 Windows 系統的自訂收集器』。</li></ul>
<input type="checkbox"/>	針對任何新的或已編輯的必備項目內容，檢查預先定義的評估器是否可比較必備項目內容的期望值與實際值。	<ul style="list-style-type: none"><li>如果是，則執行下一項檢查。</li><li>否則，建立自訂評估器。如需相關資訊，請參閱第 51 頁的『建立 Windows 系統的自訂收集器』。</li></ul>

表 10. 使用 Windows 系統的配置檔之前的檢查及作業 (繼續)

	檢查	作業
<input type="checkbox"/>	請確保所有檔案均已在正確的目錄中儲存： <ul style="list-style-type: none"> <li>• 配置檔、任何產品特定的自訂收集器及相關聯批次檔以及任何自訂評估器檔案均位於 <i>ips_root/Windows</i> 目錄中</li> <li>• 自訂共用收集器位於 <i>ips_root/lib</i> 目錄中</li> </ul>	執行 Prerequisite Scanner。如需相關資訊，請參閱第 57 頁的第 4 章, 『正在執行 Prerequisite Scanner』。

## UNIX 系統的必要檢查及延伸作業

在執行 IBM Prerequisite Scanner 之前，您應該執行一組必備項目檢查及作業。這些檢查會判定您是可以編輯及使用現有配置檔還是必須延伸 Prerequisite Scanner。

表 11 提供要執行之必要檢查及作業的清單。

表 11. 使用 UNIX 系統配置檔之前的檢查及作業

	檢查	作業
<input type="checkbox"/>	檢查產品是否在 <i>codename.cfg</i> 檔案中列出。	<ul style="list-style-type: none"> <li>• 如果是，則執行下一項檢查。</li> <li>• 否則，將產品型號新增至 <i>codename.cfg</i> 檔案。如需相關資訊，請參閱第 41 頁的『新增產品型號』。</li> </ul>
<input type="checkbox"/>	檢查與產品相關聯之產品型號的配置檔是否存在。	<ul style="list-style-type: none"> <li>• 如果是，則執行下一項檢查。</li> <li>• 否則，建立配置檔以包含產品所支援之全部平台的必備項目內容。如需相關資訊，請參閱第 41 頁的『建立自訂配置檔』。</li> </ul>
<input type="checkbox"/>	開啟配置檔並檢查其是否包含正確的必備項目內容。	<ul style="list-style-type: none"> <li>• 如果是，則執行下一項檢查。</li> <li>• 否則，新增必備項目內容。如需相關資訊，請參閱第 43 頁的『新增必備項目內容』。</li> </ul>
<input type="checkbox"/>	檢查必備項目內容是否具有期望值。	<ul style="list-style-type: none"> <li>• 如果是，則執行 Prerequisite Scanner。如需相關資訊，請參閱第 57 頁的第 4 章, 『正在執行 Prerequisite Scanner』。</li> <li>• 否則，編輯必備項目內容。如需相關資訊，請參閱第 44 頁的『編輯必備項目內容』。</li> </ul>
<input type="checkbox"/>	針對任何新必備項目內容，檢查預先定義的收集器是否可收集必備項目內容的實際值。	<ul style="list-style-type: none"> <li>• 如果是，則執行下一項檢查。</li> <li>• 否則，建立自訂收集器。如需相關資訊，請參閱第 49 頁的『建立 UNIX 系統的自訂收集器』。</li> </ul>
<input type="checkbox"/>	針對任何新的或已編輯的必備項目內容，檢查評估器是否可比較必備項目內容的期望值與實際值。	<ul style="list-style-type: none"> <li>• 如果是，則執行下一項檢查。</li> <li>• 否則，建立自訂評估器。如需相關資訊，請參閱第 55 頁的『建立 UNIX 系統的自訂評估器』。</li> </ul>
<input type="checkbox"/>	針對任何新的或已編輯的必備項目內容，檢查呼叫及執行收集器的程式碼是否位於 <i>ips_root/UNIX_Linux/packageTest.sh</i> Script 中。	<ul style="list-style-type: none"> <li>• 如果是，則執行下一項檢查。</li> <li>• 否則，編輯主套件測試 Script。如需相關資訊，請參閱第 50 頁的『編輯 UNIX 系統的套件測試 Script』。</li> </ul>
<input type="checkbox"/>	確保所有檔案均已在正確的目錄中儲存： <ul style="list-style-type: none"> <li>• 配置檔、任何自訂收集器檔案以及任何自訂評估器檔案均位於 <i>ips_root/UNIX_Linux</i> 目錄中</li> </ul>	執行 Prerequisite Scanner。如需相關資訊，請參閱第 57 頁的第 4 章, 『正在執行 Prerequisite Scanner』。

---

## 新增產品型號

IBM Prerequisite Scanner 在 `codename.cfg` 檔案中提供了一組預先定義的產品版本代碼。針對產品版本、其支援的平台以及版本，如果檔案不包含產品型號，則您可以進行新增。

### 程序

1. 開啓 `ips_root/codename.cfg` 檔案。
2. 檢查檔案是否已包含產品版本的名稱/值配對。
3. 如果產品型號不存在，請新增並確保使用正確的格式，如下所示：

```
product_code=code_value
```

**限制：**IBM Tivoli Monitoring 及 Tivoli Composite Application Manager 具有 Prerequisite Scanner 視為保留項目的產品型號。這些型號不能用作 Prerequisite Scanner 產品型號，除非是指其關聯的 IBM Tivoli Monitoring 及 Tivoli Composite Application Manager 代理程式。如需產品型號的相關資訊，請參閱 ITM 6.X 產品型號 Technote。

**限制：**僅限 UNIX：在檔案中輸入產品型號的值時，請避免使用 `for`。它是保留字，可影響 Prerequisite Scanner 的執行方式。

例如，如果在所有 Windows 平台上新增 IBM Tivoli Monitoring for Energy Management，請將下行新增至檔案中：

```
MEA=IBM Tivoli Monitoring for Energy Management
```

---

## 建立自訂配置檔

如果預先定義的配置檔不符合您的必備項目內容需求，則可以根據範例配置檔建立自訂配置檔。在建立自訂配置檔之前，請確保您瞭解要新增的必備項目內容及其期望值。

### 關於這項作業

**重要：**您必須遵循控管自訂配置檔之建立及編輯的命名慣例及格式化規則。否則，Prerequisite Scanner 將無法順利使用此檔案執行掃描。

### 程序

1. 必要的話，將產品的產品型號新增至 `codename.cfg` 檔案。
2. 使用 `ips_root/OS` 目錄中的文字編輯器建立配置檔。請確保對檔名使用下列命名慣例：

```
product_code_version.cfg
```

其中：

- `product_code`

它是 Windows 或 UNIX 系統上代表產品型號的變數。產品型號識別產品、個別平台（如 Windows、AIX、HP-UX、Linux 及 Solaris）以及該產品支援的作業系統版本（可選擇）。其儲存於 `codename.cfg` 檔案中。任何支援多個平台的產品都具有多個產品型號，每一個均識別產品、平台及作業系統版本（視需要）。

- *version* 是代表版本、版次、修正及層次的 8 位代碼，其中每兩位數表示代碼的每個部分，例如，7.3.21 即是 07032100。
3. 檢閱 第 81 頁的附錄 C, 『必備項目內容參照』 所述的基本必備項目內容並判定要檢查哪些必備項目內容。
  4. 選擇性的： 新增區段並確保對區段標題使用下列命名慣例：

- **單一、預先定義的資料類型種類**

[*category\_name:category\_value*]

例如，若要建立可用於所有 Windows 平台的必備項目內容區段，請新增下列區段標題：

[OSType:Windows]

例如，若要建立可用於所有 RedHat Linux OS 變式，請新增下列區段標題：

[OSType:RedHat]

- **組合、預先定義的資料類型種類**

[*category\_name:category\_value*]

[*category\_name:category\_value*]

例如，若要建立用於 Windows Server 2003 變式（排除 Windows Server 2003 R2 變式）的必備項目內容區段，請新增下列組合區段標題：

[OSType:Windows Server 2003][!OSType:Windows Server 2003 R2]

例如，建立用於 SUSE Linux Enterprise Server 11 OS 的必備項目內容區段並且檢查環境變數 @TPAE\_DB\_SERVER 是否設為 true。請新增下列組合區段標題：

[OSType=SUSELinuxEnterpriseServer][@TPAE\_DB\_SERVER:true]

其中：

*category\_name* 是多字元代碼，代表第 15 頁的表 6 所述的資料類型種類

*category\_value* 是多字元代碼，代表第 15 頁的表 6 所述種類的接受值

5. 選擇性的： 針對每個區段，檢閱第 81 頁的附錄 C, 『必備項目內容參照』 所述的基本必備項目內容並判定要檢查哪些必備項目內容。
6. 針對您要新增的每個必備項目內容，按需要輸入具有選用限定元的名稱/值配對。請確保使用下列格式，每一行只有一個必備項目內容。

[*prefix\_identifier.property\_name[.suffix\_identifier]=qualifier\_name:qualifier\_value*]*property\_value*

其中：

- *prefix\_identifier* 是預先定義的必備項目內容種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID。
- *property\_name* 是必備項目內容的名稱。
- *suffix\_identifier* 是必備項目內容子類型的選用 ID，如第 6 頁的表 4 所述。
- *qualifier\_name* 是必備項目內容的選用屬性。IBM Prerequisite Scanner 使用它來限定必備項目內容或要在必備項目內容上執行的檢查類型，如第 8 頁的『預先定義的必備項目內容限定元』所述。

**註：**您可以具有多個限定元，以逗點分隔每一個。限定元組必須由 [] 方括弧括住。

- *qualifier\_value* 是選用屬性的值。每一個限定元及其值必須以 : 冒號分隔。
- *property\_value* 是必備項目內容的值，可以是字串或整數。

例如，使用者預先定義的必備項目內容種類具有 `user` 字首 ID。用於檢查登入的使用者是否屬於管理者使用者群組的必備項目內容為：`user.isAdmin=True`

7. 如果預先定義的種類中不存在必備項目內容，請新增自訂必備項目內容的名稱、值及選用限定元。然後，您必須建立下列檔案，以視需要檢查並比較自訂必備項目內容：可收集必備項目內容之實際值的自訂收集器，以及自訂評估器（當標準比較函數無法比較實際值與期望值時）。

---

## 新增必備項目內容

您可以從預先定義的必備項目內容種類將基本必備項目內容新增至配置檔。或者，可以新增自訂必備項目內容。

### 關於這項作業

**重要：**您必須遵循控管在配置檔中新增及編輯必備項目內容的格式化規則。否則，Prerequisite Scanner 將無法順利執行該必備項目內容掃描。

### 程序

1. 開啟配置檔。
2. 檢閱 第 81 頁的附錄 C, 『必備項目內容參照』 所述的基本必備項目內容並判定要檢查哪些必備項目內容。
3. 針對您要新增的每個必備項目內容，按需要輸入具有選用限定元的名稱/值配對。

例如，若要從共用預先定義的種類新增必備項目內容，請只輸入內容名稱及期望值。將下列必備項目內容新增至檔案：

```
Disk=1GB
OS Version=regex{Windows 200[3-8]}
```

例如，必備項目內容之網絡預先定義的種類具有 `network` 字首 ID，並且檢查可用埠的必備項目內容名稱為 `availablePorts`。您可以進一步依據應用程式子類型將可用的埠進行分類：DB2 用於 DB2 資料庫伺服器、WAS 用於 WebSphere Application Server、FTP 用於 FTP 通訊協定。將下列必備項目內容新增至檔案：

```
network.availablePorts.DB2=5000-5005
network.availablePorts.WAS=9080
network.availablePorts.FTP=21
```

例如，必備項目內容之作業系統預先定義的種類具有 `os` 字首 ID，並且檢查檔案系統之可用磁碟空間的必備項目內容名稱為 `space`。您可以進一步依據檔案系統子類型將檢查分類：`usr` 及 `home`。您可以指定 `dir` 及 `unit` 限定元的值。

將下列必備項目內容新增至檔案：

```
os.space.usr=[dir:root=/usr/ibm/common/acsi,unit:GB]2
os.space.home=[dir:non_root=USERHOME/.acsi_HOST,unit:MB]200
```

**重要：**您僅可將預先定義的限定元用於特定預先定義的內容，如第 9 頁的表 5 所述。

4. 如果預先定義的必備項目內容種類中不存在必備項目內容，請針對自訂必備項目內容及值新增具有選用限定元的名稱/值配對。請確保使用下列格式，每一行只有必備項目內容。

```
[prefix_identifier.]property_name[.suffix_identifier]=  
[[qualifier_name:qualifier_value]]property_value
```

其中：

- *prefix\_identifier* 是預先定義的必備項目內容種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID。
- *property\_name* 是必備項目內容的名稱。
- *suffix\_identifier* 是必備項目內容子類型的選用 ID，如第 6 頁的表 4 所述。
- *qualifier\_name* 是必備項目內容的選用屬性。IBM Prerequisite Scanner 使用它來限定必備項目內容或要在必備項目內容上執行的檢查類型，如第 8 頁的『預先定義的必備項目內容限定元』所述。

**註：**您可以有多個限定元，以逗點分隔其中每一個。限定元組必須由 [] 方括弧括住。

- *qualifier\_value* 是選用屬性的值。每一個限定元及其值必須以 : 冒號分隔。
- *property\_value* 是必備項目內容的值，可以是字串或整數。

例如，`env.tcrhome` 是自訂必備項目內容，它會檢查 Tivoli Common Reporting 的起始目錄環境變數且期望值應為 `True`：

```
env.tcrhome=True
```

`env.path.jar` 是自訂必備項目內容，它會檢查 `PATH` 環境變數中是否已設定 `JRE` 且期望值應為 `False`：

```
env.path.jar=False
```

**註：**然後，您必須建立下列檔案，以視需要檢查並比較自訂必備項目內容：可收集必備項目內容之實際值的自訂收集器，以及自訂評估器（僅當標準比較函數無法比較實際值與期望值時）。

---

## 編輯必備項目內容

您可以編輯必備項目內容、變更那些必備項目內容的期望值或變更限定元的關聯值。

### 開始之前

檢查新值是否為必備項目內容支援的有效值。例如，`Disk` 必備項目內容預期單位為 `MB` 或 `GB` 的數字格式。如果您要以 `TB` 為單位來檢查可用磁碟空間，則必須延伸比較 API 以支援 `TB` 比較。您還必須編輯相關配置檔中的 `Disk` 必備項目內容。

檢查必備項目內容之預先定義的限定元及有效值，如第 8 頁的『預先定義的必備項目內容限定元』中所述。

### 程序

1. 開啓配置檔。

2. 針對您要編輯的每一個必備項目內容，輸入新的期望值或變更限定元的值。例如，新的系統管理者是 root 使用者，因此 user.userID 必備項目內容的值必須變更。將值變更為新名稱：

```
user.userID=smithj
```

例如，os.ulimit 必備項目內容的 type 限定元目前值為 filedescriptorlimit，可檢查檔案描述子的限制。您可能要檢查其他限制，如堆疊大小。請變更必備項目內容的下列限定元值：

```
os.ulimit=[type:filedescriptorlimit]8192+,unlimited
```

變更為：

```
os.ulimit=[type:stacksize]512+,unlimited
```

**重要：**您僅可以將預先定義的限定元與特定預先定義的內容搭配使用，如第 9 頁的表 5 中所述。

---

## 建立 Windows 系統的自訂收集器

如果基本集收集器未收集要安裝之產品所需的必備項目內容值，則您可以建立自訂連接器。您可以建立自訂共用 VBScript 收集器，以收集適用於任何產品及產品版本之必備項目內容的資料。或者，您可以建立產品特定的自訂收集器，以收集適用於特定產品及產品版本的資料。雖然每種類型的自訂 VBScript 收集器均使用相同的方法收集資料，但是建立、儲存及執行的規則稍有不同。

### 建立所有配置檔共用的自訂 VBScript 收集器

建立自訂共用 VBScript 收集器時，檔名必須包含必備項目內容的名稱，並儲存於 /lib 子目錄中。收集器包含程式碼，可取得必備項目內容的實際值。它還可以使用共用函數及子常式，以在需要時取得那些值。

#### 開始之前

確保在建立收集器之前，檢閱下列附錄中的預先定義函數及子常式集。判定您是否可使用其中的任何函數來取得實際值：

- 第 109 頁的附錄 E, 『Windows 系統的共用函數』
- 第 125 頁的附錄 G, 『Windows 系統的檔案公用程式子常式』
- 第 123 頁的附錄 F, 『Windows 系統的記載公用程式子常式』
- 第 127 頁的附錄 H, 『Windows 系統的其他共用函數及子常式』

判定收集器是否必須檢查必備項目內容存在，且如果存在，必須收集何種資訊。每次檢查必須傳回值（指出是否存在必備項目內容）。例如：

- 檢查環境變數是否存在，如產品的起始目錄，例如 Tivoli Common Reporting 的 TCR\_HOME。
- 檢查環境變數是否包含 JAR 檔、二進位檔或路徑，如 PATH 環境變數中 JRE 的路徑。
- 檢查環境變數的實際值，如產品的起始目錄，例如 Tivoli Common Reporting 的 TCR\_HOME。
- 檢查是否已安裝產品。

- 檢查安裝了產品的哪個版本。

## 程序

1. 建立 VBScript 檔案。將檔案儲存至 *ips\_root/lib* 目錄，檔案命名慣例有如下不同：

`[prefix_identifier.]property_name.vbs`

其中：

- *prefix\_identifier* 是必備項目內容預先定義之種類的 ID，如第 4 頁的表 3 所述。
- *property\_name* 是必備項目內容名稱並在收集器名稱中使用。

例如，`mssqlVersion.vbs` 包含程式碼，可取得 Windows 機器上 MS SQL Server 必備項目內容的實際值。

2. 使用 VBScript 編輯器，新增程式碼，以取得必備項目內容的值。使用 VBScript COM 及函數，以存取 Windows 環境的元素，並在 Windows Script 主機環境中執行。請確保檢查傳回如下標準輸出：

```
WScript.Echo "property_name=" &#38; var_for_value
```

- *property\_name*，代表配置檔中所寫入的必備項目內容，例如 `env.tcrhome`。
- *var\_for\_value*，即收集器為必備項目內容所取得之實際值的 VBScript 變數。

若要檢查 `TCR_HOME` 環境是否存在並傳回實際值（其中必備項目內容名稱為 `env.tcrhome`）：

```
set wshShell = WScript.CreateObject("WScript.Shell")
tcr_home=WshShell.ExpandEnvironmentStrings("%TCR_HOME%")
WScript.Echo "env.tcrhome=" &#38; tcr_home
```

若要檢查 `JRE` 是否在 `PATH` 變數中設定（其中必備項目名稱為 `env.path.jre`）：

```
Set wshShell = WScript.CreateObject("WScript.Shell")
path = WshShell.ExpandEnvironmentStrings("%PATH%")
Set objRegExp = new RegExp
objRegExp.Pattern = "(^[[:;\\/]]) (C:\Program Files\IBM\Java60\jre\bin)([[:;]])"
objRegExp.IgnoreCase = True
objRegExp.Global = True
Set matches = objRegExp.Execute(path)
WScript.Echo "env.path.jre=" &#38; (matches.Count > 0)
```

若要檢查已安裝之 Tivoli Directory Integrator 的版本（其中必備項目內容名稱為 `installedSoftware.TDI.version`）：

```
strComputer = "."
strKeyPath = "SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall"
regDisName = "DisplayName"
regDisVer = "DisplayVersion"

Set oReg = GetObject("winmgmts:{impersonationLevel=Impersonate}!\\" &#38; strComputer &#38; "\root\default:StdRegProv")

Set sftReg = new RegExp
sftReg.pattern = "Tivoli Directory Integrator"
sftReg.Global=False
oReg.EnumKey HKEY_LOCAL_MACHINE, strKeyPath, arrSubKeys
For Each subkey In arrSubKeys
    searchkey = strKeyPath & "\" &#38; subkey
    oReg.GetStringValue HKEY_LOCAL_MACHINE, searchkey, regDisName, strName
    oReg.GetStringValue HKEY_LOCAL_MACHINE, searchkey, regDisVer, strVersion
    If Not IsNull(strName) Then
        Set matches = sftReg.Execute(strName)
        If matches.Count > 0 Then
```



```

        Wscript.Echo "installedSoftware.TDI.version=" &#38; strVersion
    End If
End If
Next

```

3. 執行 VBScript 收集器，以確保不存在執行時期錯誤，並視需要除錯。
4. 僅當標準比較函數無法比較實際值與期望值時，才建立自訂評估器。

## 建立產品及產品版本特定的自訂 VBScript 收集器

建立產品特定的自訂 VBScript 收集器時，檔名必須與配置檔中的產品型號相同，且儲存於 /Windows 子目錄中。收集器可包含程式碼，以收集一或多個必備項目內容的實際值。它還可以使用共用函數及子常式，以在需要時收集那些值。

### 開始之前

確保在建立收集器之前，檢閱下列附錄中的函數及子常式集。判定您是否可使用其中的任何函數來取得實際值：

- 第 109 頁的附錄 E, 『Windows 系統的共用函數』
- 第 125 頁的附錄 G, 『Windows 系統的檔案公用程式子常式』
- 第 123 頁的附錄 F, 『Windows 系統的記載公用程式子常式』
- 第 127 頁的附錄 H, 『Windows 系統的其他共用函數及子常式』

判定收集器是否必須檢查必備項目內容存在，且如果存在，必須收集何種資訊。每次檢查必須傳回值（指出是否存在必備項目內容）。例如：

- 檢查目錄是否存在。
- 檢查目錄的可用磁碟空間。
- 檢查是否已安裝產品。
- 檢查安裝了產品的哪個版本。

### 程序

1. 建立 VBScript 檔案。將檔案儲存至 *ips\_root/Windows* 目錄，檔案命名慣例有如下不同：

```
product_code[version].vbs
```

其中：

- *product\_code*

它是 Windows 或 UNIX 系統上代表產品型號的變數。產品型號識別產品、個別平台（如 Windows、AIX、HP-UX、Linux 及 Solaris）以及該產品支援的作業系統版本（可選擇）。其儲存於 *codename.cfg* 檔案中。任何支援多個平台的產品都具有多個產品型號，每一個均識別產品、平台及作業系統版本（視需要）。

- *version* 是代表版本、版次、修正及層次的 8 位代碼，其中每兩位數表示代碼的每個部分，例如，7.3.21 為 07032100。
2. 如果必須使用共用函數，請使用 VBScript 編輯器開啓檔案，並將路徑併入 *common\_function.vbs*，如下所示：

```
Include("../lib\common_function.vbs")
```

3. 如果您必須使用 PATH 及 -p 旗標的值（從 Prerequisite Scanner 傳遞而來），請使用 Wscript.Arguments()，其中 Wscript.Arguments(0) 是 PATH 的值。Wscript.Arguments(1) 是 -p 旗標及其值。
4. 新增程式碼以取得必備項目內容的值，方法是使用 VBScript COM 及函數，以存取 Windows 環境的元素。在 Windows Script 主機環境中執行。請確保檢查傳回如下標準輸出：

```
WScript.Echo "property_name=" &#38; var_for_value
```

- *property\_name*，代表配置檔中所寫入的必備項目內容，例如 env.tcrhome。
- *var\_for\_value*，即收集器為必備項目內容所取得之實際值的 VBScript 變數。

若要檢查產品之安裝目錄的可用磁碟空間，例如，透過使用第 128 頁的『getValue()』子常式檢查 Tivoli Monitoring for Energy Management Reporting and Optimization（其中必備項目內容為 InstallDir）：

```
Set wshShell = WScript.CreateObject("WScript.Shell")
'Check the disk space for the installation path that is passed as
the value for the PATH argument
installPath = Wscript.Arguments(0)
sInstallPath= "InstallDir="
Wscript.Echo "installation path   : " & installPath
set fso = CreateObject("Scripting.FileSystemObject")
```

```
getValue fso, sInstallPath, installPath
```

```
'Common sub routine
Sub getValue(fso, sKey, drvPath)
    Wscript.Echo "getValue(" & sKey & ", " & drvPath & ")"
    If fso.driveExists(fso.getDriveName(drvPath)) then
        Set disk = fso.GetDrive(fso.getDriveName(drvPath))
        'Value returned is in bytes. Convert to MB
        cSize = CLng((disk.FreeSpace/1024)/1024) & "MB"
        WScript.Echo sKey & cSize
    Else
        Wscript.Echo " Disk for " & sKey & " -> " & drvPath & " does NOT exist"
    End If
End Sub
```

5. 建立批次檔，以呼叫 VBScript 收集器。批次檔名稱必須與配置檔相同，且具有 .bat 副檔名 *product\_code[\_version].bat*，如下所示：

```
@echo off

set CMD_LINE_ARGS=
:setArgs
if "%1"=="%" goto doneSetArgs
set CMD_LINE_ARGS=%CMD_LINE_ARGS% %1
shift
goto setArgs
:doneSetArgs

cscript.exe //nologo collector_file_name.vbs %CMD_LINE_ARGS%
```

6. 執行 VBScript 收集器，以確保不存在執行時期錯誤，並視需要除錯。
7. 僅當標準比較函數無法比較實際值與期望值時，才建立自訂評估器。

---

## 建立 UNIX 系統的自訂收集器

如果基本集收集器未收集要安裝之產品所需的必備項目內容值，則您可以建立自訂連接器。建立自訂收集器時，檔名必須與必備項目內容相同（雖然其名稱中沒有子類型）。收集器儲存於 `/UNIX_Linux` 子目錄中。收集器可以包含程式碼，以取得一或多個必備項目內容的實際值。它還可以使用共用函數，以在需要時取得那些值。

### 開始之前

確保在建立收集器之前，檢閱下列附錄中的函數集。判定您是否可使用其中的任何函數來取得實際值：

- 第 131 頁的附錄 I, 『UNIX 系統的共用函數』
- 第 137 頁的附錄 J, 『UNIX 系統的其他函數』
- 第 145 頁的附錄 K, 『UNIX 系統的記載公用程式函數』

判定收集器是否必須檢查必備項目內容存在，且如果存在，必須收集何種資訊。每次檢查必須傳回值（指出是否存在必備項目內容）。例如：

- 檢查是否已安裝產品，例如，隨 RMP 一起安裝的套件。
- 檢查安裝了產品的哪個版本。
- 檢查已裝載之檔案系統的可用磁碟空間

如果您要使用子類型 *suffix\_identifier*，並依據應用程式、公用程式或服務子類型進一步對必備項目內容分類，則您可以建立共用收集器。將 *suffix\_identifier* 子類型的區分元（即 *differentiator\_suffix\_identifier*）傳遞到其收集器。例如，`os.package` 是共用收集器，以檢查套件是否存在。若要檢查是否存在 `openssh`，請在呼叫 `packageTest.sh` Script 檔中的 `os.package` 收集器時傳遞套件的名稱，如下所示：

```
./os.package openssh
```

其中，`openssh` 是套件的名稱，即 *suffix\_identifier* 子類型及 *differentiator\_suffix\_identifier* 區分元。

### 程序

1. 建立 Shell Script 檔。將檔案儲存至 `ips_root/Unix_Linux` 目錄，檔案命名慣例有如下不同，但是仍不使用副檔名：

```
[prefix_identifier.]property_name
```

其中：

- *prefix\_identifier* 是必備項目內容預先定義之種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID，例如 `env`。
  - *property\_name* 是必備項目內容的名稱，例如 `path.jre`
2. 如果必須使用共用函數，請使用編輯器開啓檔案，並將路徑併入 `common_function.sh`，如下所示：

```
../lib/common_function.sh
```

3. 新增程式碼以取得必備項目內容的值，方法是使用該平台特定的指令及選項，以存取主機環境的元素。例如，自訂 `env.path.jar` 必備項目內容需要檢查 `PATH` 變數中是否已設定 `JRE`。下列程式碼執行 `env` 指令，搜尋 `PATH` 變數的輸出，然後搜尋 `JRE` 路徑的值。

```
envJRE=`env | grep "PATH" | grep -w "/opt/IBM/Java60/jre/bin"~`
```

4. 請確保檢查傳回標準輸出：

```
echo "True"|"False" 'If the scan checks for the existence of the prerequisite
property
echo $res 'If the scan checks returns the value, for example, product version,
'of the prerequisite property
echo "Unavailable" 'If the scan returns no value for the prerequisite property
echo "Available" 'If the scan returns a valid check for the prerequisite property
```

在範例中，基於 `$envJRE` 變數的值，檢查會傳回 `True` 或 `False`：

```
if [ $envJRE ]; then
  echo "True"
else
  echo "False"
fi
```

5. 執行自訂收集器，以確保不存在執行時期錯誤，並視需要除錯。
6. 編輯 `ips_root/UNIX_Linux/packageTest.sh` Script，以呼叫並執行自訂收集器。
7. 僅當自訂收集器傳回布林值以外的值時，才建立自訂評估器。

---

## 編輯 UNIX 系統的套件測試 Script

您可以更新 `packageTest.sh` Script 檔，以在 UNIX 系統上呼叫自訂收集器。

### 開始之前

請確保您瞭解與預先定義必備項目內容相關聯的收集器名稱，如第 103 頁的附錄 D, 『UNIX 系統之預先定義的收集器』中所述。如果必備項目內容由應用程式、公用程式或服務子類型進一步分類，請將 `suffix_identifier` 子類型的區分元（即 `differentiator_suffix_identifier`）傳遞到其收集器。

例如，`os.package` 是共用收集器，以檢查套件是否存在。若要檢查是否存在 `openssh`，請在呼叫 `packageTest.sh` Script 檔中的 `os.package` 收集器時傳遞套件的名稱，如下所示：

```
./os.package openssh
```

其中，`openssh` 是套件的名稱，即 `suffix_identifier` 子類型及 `differentiator_suffix_identifier` 區分元。

### 程序

1. 使用編輯器，開啓 `ips_root/UNIX_Linux/packageTest.sh` Script。
2. 新增程式碼，以讀取配置檔的自訂必備項目內容，並剖析其值。

```
res=`echo $line | grep [prefix_identifier.]property_name[.suffix_identifier]`
if [ $res ]; then
ExpValue=`echo $res | cut -d "=" -f2`
```

例如，若要讀取自訂 `env.path.jar` 必備項目內容，並檢查 JRE 是否已在 `PATH` 變數中設定：

```
res=`echo $line | grep env.path.jar`
if [ $res ]; then
ExpValue=`echo $res | cut -d "=" -f2`
```

在範例中：

```

echo "\`wr1Trace "Starting" "env.path.jar"\`" >>/tmp/prs.check
echo "\`wr1Trace "Executing" "env.path.jar"\`" >>/tmp/prs.check
echo "\`wr1Debug "Starting" "env.path.jar"\`" >>/tmp/prs.check
echo "\`wr1Debug "Expected" "ExpValue" "\`" >>/tmp/prs.check

```

3. 呼叫記載函數以用於追蹤和除錯資料，然後呼叫自訂收集器。

```

echo "\`wr1Trace "Starting" "[prefix_identifier.]property_name
[.suffix_identifier]" "\`" >>/tmp/prs.check
echo "\`wr1Trace "Executing" "[prefix_identifier.]property_name
[.suffix_identifier]" "\`" >>/tmp/prs.check
echo "\`wr1Debug "Starting" "[prefix_identifier.]property_name
[.suffix_identifier]" "\`" >>/tmp/prs.check
echo "\`wr1Debug "Expected" "ExpValue" "\`" >>/tmp/prs.check

```

4. 呼叫自訂收集器。

註：如果自訂收集器在檔名中具有子類型（即 `[suffix_identifier]`），並且需要基於子類型進行進一步檢查，請將子類型的 `[differentiator_suffix_identifier]` 區分元傳遞到自訂收集器。

```

echo "ss=\`./[prefix_identifier.]property_name[.suffix_identifier]
[differentiator_suffix_identifier]" "\`" >>/tmp/prs.check

```

在範例中：

```

echo "ss=\`./env.path.jar"\`" >>/tmp/prs.check

```

註：`os.file.script_name` 必備項目內容之 `script_name` 子類型的區分元範例是傳遞到 `os.filepath` 收集器之 `Script` 的路徑：

```

echo "ss=\`./os.filepath /usr/bin/expect"\`" >>/tmp/prs.check #os.file.expect
echo "ss=\`./os.filepath /usr/bin/tar"\`" >>/tmp/prs.check #os.file.tar
echo "ss=\`./os.filepath /usr/bin/gzip"\`" >>/tmp/prs.check #os.file.gzip

```

5. 結束自訂收集器時，呼叫記載函數以用於追蹤和除錯資料。

```

echo "\`wr1Trace "Finished" "[prefix_identifier.]property_name
[.suffix_identifier]" "\`" >/tmp/prs.check
echo "echo \"[prefix_identifier.]property_name
[.suffix_identifier]=\`ss\"" >>/tmp/prs.check
echo "\`wr1Debug "Finished" "[prefix_identifier.]property_name
[.suffix_identifier]" "\`" >>/tmp/prs.check
echo "\`wr1Debug "OutPutValueIs" "\`ss\"" >/tmp/prs.check
echo "\`wr1Trace "Done" "[prefix_identifier.]property_name
[.suffix_identifier]" "\`" >>/tmp/prs.check
fi

```

在範例中：

```

echo "ss=\`./env.path.jar"\`" >>/tmp/prs.check
echo "\`wr1Trace "Finished" "env.path.jar"\`" >>/tmp/prs.check
echo "echo \"env.path.jar=\`ss\"" >>/tmp/prs.check
echo "\`wr1Debug "Finished" "env.path.jar"\`" >>/tmp/prs.check
echo "\`wr1Debug "OutPutValueIs" "\`ss\"" >>/tmp/prs.check
echo "\`wr1Trace "Done" "env.path.jar"\`" >>/tmp/prs.check
fi

```

6. 針對每個自訂必備項目內容，重複步驟 2 到 5。

---

## 建立 Windows 系統的自訂收集器

如果基本評估器未使用正確的評估準則比較必備項目內容的期望值與實際值，您可以建立 VBScript 評估器。當建立自訂評估器時，檔名必須以 `_compare` 結尾且儲存於 `/Windows` 子目錄中。自訂評估器可使用共用函數及子常式，以在需要時比較值。

## 開始之前

確保在建立評估器之前檢閱了下列附錄中的函數及子常式組。判定您是否可使用其中的函數來比較值：

- 第 109 頁的附錄 E, 『Windows 系統的共用函數』
- 第 125 頁的附錄 G, 『Windows 系統的檔案公用程式子常式』
- 第 123 頁的附錄 F, 『Windows 系統的記載公用程式子常式』
- 第 127 頁的附錄 H, 『Windows 系統的其他共用函數及子常式』

**註：**共用函數第 118 頁的『passOrFail()』可比較下列類型的實際值與期望值：一般數字、大小（單位為 MB 或 GB）、處理器速度（單位為 MHz 或 GHz）、布林值或字串。僅當無法使用 passOrFail 函數時建立自訂評估器。

## 程序

1. 建立 VBScript 檔案。將檔案儲存至 *ips\_root/Windows* 目錄，命名慣例有如下不同：

```
[prefix_identifier.]property_name[.suffix_identifier]_compare.vbs
```

其中：

- *prefix\_identifier* 是預先定義的必備項目內容種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID。
  - *property\_name* 是必備項目內容的名稱。
  - *suffix\_identifier* 是必備項目內容子類型的選用 ID，如第 6 頁的表 4 所述。
2. 新增代碼，以比較使用 VBScript COM 作為引數傳遞到評估器及相關聯函數的實際值與期望值。請確保比較傳回如下標準輸出：
    - "PASS"：當必備項目內容的期望值等於或大於必備項目內容的實際值
    - "FAIL"：當必備項目內容的期望值不等於必備項目內容的實際值
  3. 執行自訂評估器，以確保不存在執行時期錯誤並視需要除錯。

## 範例

此自訂評估器會檢查 Tivoli Directory Integrator 版本的實際值與期望值。它會使用共用函數第 129 頁的『versionCompare()』。

```
wscript.echo "expect: " &#38; wscript.arguments(0)
wscript.echo "real value: " &#38; wscript.arguments(1)
wscript.echo tdiVersionCompare(wscript.arguments(0), wscript.arguments(1))

function tdiVersionCompare(expect, real)
    if len(real) = 0 then
        tdiVersionCompare = "FAIL"
        exit function
    end if

    expect = Trim(expect)
    real = Trim(real)

    Dim expectedVersion
    'if (StrComp(Right(expect,1),"+")=0 or StrComp(Right(real,1),"-")=0) Then
    if (Right(expect,1)="+ " or Right(real,1)="- ") Then
        expectedVersion = Left(expect,len(expect)-1)
    else
        expectedVersion = expect
    end if
```

```

Dim cmp
cmp = versionCompare(expectedVersion,real)

if (StrComp(Right(expect,1),"+")=0) Then
' Version must be at least expected value
if (cmp=0 or cmp=-1) Then
tdiVersionCompare = "PASS"
else
tdiVersionCompare = "FAIL"
end if
elseif (StrComp(Right(expect,1),"-")=0) Then
' Version must be less than or equal to expected value
if (cmp=0 or cmp=1) Then
tdiVersionCompare = "PASS"
else
tdiVersionCompare = "FAIL"
end if
elseif cmp=0 then
tdiVersionCompare = "PASS"
else
tdiVersionCompare = "FAIL"
end if
end function

' Generic function for comparing 2 version strings
'
' Parameters
' ver1 The first version string
' ver2 The second version string
'
' ver1 and ver2 are expected to be dot-separated version strings
' (e.g. 1.0.0.4, 2.3, 3.40.26.7800, 2.3.a)Version strings can have any
' number of parts. When comparing versions with different numbers of
' parts, missing parts of the shorter version string will be treated
' as if there was a zero there. If any non-numeric characters are
' included in a version part, those corresponding parts will be compared
' as strings and not parsed into numeric form
'
' Returns
' 1 version1 > version2
' -1 version1 < version2
' 0 version1 = version2
'
' Special cases:
' RESULT version 1 version 2
' 0 empty empty
' 1 validString empty
' -1 empty validString
'
' NOTE: This function should eventually move to common_functions.vbs

function versionCompare(ver1, ver2)
WScript.echo "Comparing [" & ver1 & "]" to [" & ver2 & "]"

Const UNASSIGNED = "*UNASSIGNED*"
Dim v1Default, v2Default

' Handle special cases:
if (IsEmpty(ver1) and IsEmpty(ver2)) Then
versionCompare = 0
exit function
end if
if (IsEmpty(ver1) and not IsEmpty(ver2)) Then
versionCompare = -1
exit function
end if
if (not IsEmpty(ver1) and IsEmpty(ver2)) Then

```

```

        versionCompare = 1
    exit function
end if

Dim ver1Parts, ver2Parts

' Versions are not empty. Break into parts and compare numbers
ver1Parts = Split(ver1, ".")
ver2Parts = Split(ver2, ".")

Dim v1Size, v2Size
v1Size = ubound(ver1Parts)
v2Size = ubound(ver2Parts)

' If last version part is "*", treat all missing parts as "*"
'(so 2.* matches 2.1.3, for example)
if (v1Size > v2Size) Then
    Redim Preserve ver2Parts(v1Size)
    if (ver2Parts(v2Size) = "*") Then
        for i = v2Size to v1Size
            ver2Parts(i) = "*"
        next end if
elseif (v2Size > v1Size) Then
    Redim Preserve ver1Parts(v2Size)
    if (ver1Parts(v1Size) = "*") Then
        for i = v1Size to v2Size
            ver1Parts(i) = "*"
        next end if
end if

Dim i
i = 0

Do While (i <= ubound(ver1Parts) or i <= ubound(ver2Parts))
    Dim v1, v2, v1Str, v2Str

    v1Str = UNASSIGNED
    v2Str = UNASSIGNED

    if (i <= ubound(ver1Parts)) Then
        on error resume next
        v1 = Int(ver1Parts(i))
        if not Err=0 Then
            v1Str = ver1Parts(i)
            if (i <= ubound(ver2Parts)) Then
                v2Str = ver2Parts(i)
            else
                v2Str = "0"
            end if
        end if
    else
        v1 = 0
    end if

    if (i <= ubound(ver2Parts)) Then
        on error resume next
        v2 = Int(ver2Parts(i))
        if not Err=0 Then
            if (i <= ubound(ver1Parts)) Then
                v1Str = ver1Parts(i)
            else
                v1Str = "0"
            end if
            v2Str = ver2Parts(i)
        end if
    else
        v2 = 0
    end if

```



```

end if

if (not v1Str=UNASSIGNED or not v2Str=UNASSIGNED) Then
  if (IsEmpty(v1Str)) Then
    v1Str = "0"
  end if
  if (IsEmpty(v2Str)) Then
    v2Str = "0"
  End if

  'WScript.echo "Comparing as strings: " &#38; v1Str &#38; " : " &#38; v2Str
  ' Compare as Strings if either part could not be converted to a number
  if (not v1Str="*" and not v2Str="*") Then
    if (not v1Str=v2Str) Then
      versionCompare = StrComp(v1Str,v2Str)
      exit function
    end if
  end if
else
  'WScript.echo "Comparing as numbers: " &#38; v1 &#38; " : " &#38; v2

  if (v1 > v2) Then
    versionCompare = 1
    exit function
  end if
  if (v2 > v1) Then
    versionCompare = -1
    exit function
  end if
end if

  i = i + 1
Loop

' If we got here, versions must be equal
versionCompare = 0

end function

```

---

## 建立 UNIX 系統的自訂評估器

如果自訂收集器未傳回布林值，即 True 或 False，您可以建立自訂評估器。當建立自訂評估器時，檔名必須以 `_compare` 結尾且儲存於 `/UNIX_Linux` 子目錄中。自訂評估器可使用共用函數，以在需要時比較值。

### 開始之前

確保在建立評估器之前檢閱了下列附錄中的函數組。判定您是否可使用其中的函數來比較實際值與期望值：

- 第 131 頁的附錄 I, 『UNIX 系統的共用函數』
- 第 137 頁的附錄 J, 『UNIX 系統的其他函數』
- 第 145 頁的附錄 K, 『UNIX 系統的記載公用程式函數』

您可以從兩個 Script 檔著手，即 `/Unix_Linux` 子目錄中的 `._compare.sh` 及 `_compare.sh`。

**重要：**如果您的自訂收集器傳回 True 或 False，請勿建立自訂評估器。IBM Prerequisite Scanner 會針對傳回布林值的任何收集器使用預先定義的評估器。

## 程序

1. 建立 Shell 檔案。將檔案儲存至 `ips_root/UNIX_Linux` 目錄，命名慣例有如下不同：  
`[prefix_identifier.]property_name[.suffix_identifier]_compare.sh`

其中：

- `prefix_identifier` 是預先定義的必備項目內容種類的 ID，如第 4 頁的表 3 所述。部分預先定義的種類需要此字首 ID。
  - `property_name` 是必備項目內容的名稱。
  - `suffix_identifier` 是必備項目內容子類型的選用 ID，如第 6 頁的表 4 所述。
2. 新增代碼，以比較作為引數傳遞到評估器及相關聯函數的實際值與期望值。請確保比較傳回如下標準輸出：
    - "PASS"：當必備項目內容的期望值等於或大於必備項目內容的實際值
    - "FAIL"：當必備項目內容的期望值不等於必備項目內容的實際值
  3. 執行自訂評估器，以確保不存在執行時期錯誤並視需要除錯。

---

## 第 4 章 正在執行 Prerequisite Scanner

您可以使用指令行介面，以執行 IBM Prerequisite Scanner。Prerequisite Scanner Script `prereq_checker` 接收一組必要及選用參數以及用於其他選用參數的指令旗標。

表 12 說明 Prerequisite Scanner Script 語法中使用的特殊字元。

表 12. Prerequisite Scanner Script 的特殊字元圖註

特殊字元	說明
<>	識別位置保留元名稱。
[]	識別選用參數。需要不以方括弧括住的參數。
...	指出您可為參數指定多個值。
	指出互斥的參數。指定分隔字元左側的參數或其右側的參數，但不能同時指定兩者。
{}	含括一組由   分隔的互斥參數。

---

### prereq\_checker

`prereq_checker` Script 會執行 IBM Prerequisite Scanner，並根據執行該 Script 時指定的一組參數，來檢查必備項目。

#### 語法

```
prereq_checker.bat|sh
  "Product_Code [Product_Version][,Product_CodeN [Product_VerN]]..."
  [detail]
  [outputDir="ips_output_dir"]
  [xmlResult]
  [PATH="product_root"]
  [-p Product_Code.instance.parameter=value,...]
  [debug]
  [trace]
```

`prereq_checker` Script 具有一個必要的參數及數個選用參數。

第 58 頁的『"Product\_Code [Product\_Version][,Product\_CodeN [Product\_VerN]]..."』

必要的參數

第 58 頁的『[detail]』

選用參數

第 61 頁的『[outputDir="ips\_output\_dir"]』

選用參數

第 61 頁的『[xmlResult]』

選用參數

第 61 頁的『[PATH="product\_root"]』

第 61 頁的『[-p Product\_Code.instance.parameter=value,...]』

選用旗標

第 62 頁的『[debug]』

選用參數

第 62 頁的『[trace]』

選用參數

**"Product\_Code [Product\_Version][,Product\_CodeN [Product\_VerN)]..."**

您必須至少設定一個 **Product\_Code** 參數，以識別為其執行必備項目檢查及相關配置檔的產品或元件。**Product\_Code** 是您在 *ips\_root/codename.cfg* 檔案中設定的產品型號。

例如，KMS 是 Tivoli Enterprise Monitoring Server 在 *product.cfg* 檔案中的產品型號。若要執行 Scanner，請輸入具有產品型號的下列 Script：

```
./prereq_checker.sh KMS
```

如果您設定沒有對應配置檔的 **Product\_Code** 參數，則 Prerequisite Scanner 會忽略該參數且不會傳回錯誤。日誌檔會包含一條訊息，說明找不到配置檔。

相關聯 **Product\_Code** 參數的 **Product\_Version** 參數會指出產品的版本。它是代表版本、版次、修正及層次的 8 位代碼，其中每兩位數表示代碼的每個部分，例如，7.3.21 為 07032100。**Product\_Version** 是選用參數。如果未設定該參數，則 Prerequisite Scanner 會檢查最新的可用版本。

如果您設定一或多個 **Product\_Code** 參數並搭配使用選用 **Product\_Version** 參數，請以逗點分隔每個參數。

**重要：**當設定多個 **Product\_Code** 參數並搭配使用選用 **Product\_Version** 參數時，請使用引號含括這些參數。如果不這樣做，Scanner 會失敗。

此範例會檢查 Tivoli Monitoring Operating System Agent for Windows 最新版本以及 Tivoli Monitoring Agent for DB2 6.2.1 版的必備項目。

```
prereq_checker.bat "KNT,KUD 06210000"
```

## [detail]

此選用參數指出是否在指令行介面中顯示掃描的詳細結果。

**重要：**請勿使用引號含括此參數。

設定 **detail** 參數時，詳細結果會包含：

- Prerequisite Scanner 的版本
- 執行 Scanner 之作業系統的版本
- 為其執行必備項目檢查之產品或元件的名稱
- 針對每個必備項目內容：已檢查之必備項目內容的名稱、PASS 或 FAIL 結果、實際值以及期望值
- 針對所有元件：已檢查之一般必備項目內容的名稱、PASS 或 FAIL 結果、實際值以及期望值
- 整體 PASS 或 FAIL 結果

Prerequisite Scanner 還會將這些結果儲存到 *ips\_output\_dir/result.txt* 檔案中。無論您是否設定 **detail** 參數，它都會將結果儲存到該文字檔。

```

root@aclinux15:~/prs/20110927-0849
File Edit View Terminal Tabs Help
[root@aclinux15 20110927-0849]# ./prereq_checker.sh DMO detail
IBM Prerequisite Scanner
  Version: 1.1.1.8
  Build : 20110927
  OS Name: Linux

Machine Info
Machine Name : <Machine name>
Serial Number: <Serial number>

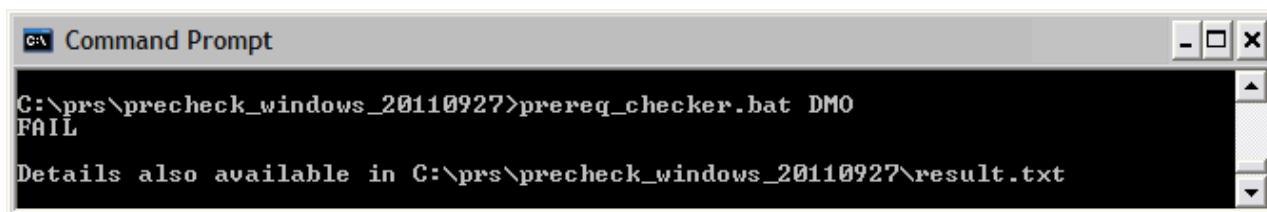
TPS detected : Red Hat Enterprise Linux Server release 5.5 {32-bit}
Using the DMO config file
Using config file - /root/prs/20110927-0849/UNIX_Linux/DMO_0750000.cfg for DMO
DMO - Prerequisite Scanner Demo [0750000]:
Evaluation          PASS/FAIL Result          Expected Result
DBType              FAIL    Unknown                   Oracle
DBType              FAIL    Unknown                   DB2
DBType              FAIL    Unknown                   regex{.*Oracle.*}
DBType              FAIL    Unknown                   regex{.*DB2.*}
DBTypeDetails      FAIL    Unknown                   oracle
DBTypeDetails      FAIL    Unknown                   DB2
DBTypeDetails      FAIL    Unknown                   regex{.*Oracle.*}
DBTypeDetails      FAIL    Unknown                   regex{.*DB2.*}
OS Version          PASS    "Red Hat Enterprise Linux Server release 5.5 (Tikanga)"
  " regex{Red Hat.*Tikanga.*}"
  regex{AIX.*}
  regex{Solaris.*}
}
os.lib.libstdc++    PASS    /usr/lib/gcc/i386-redhat-linux/4.1.1/libstdc++.so libst
dc++
os.lib.libgcc       PASS    /usr/lib/gcc/i386-redhat-linux/3.4.6/libgcc_s.so [Check
Package:True]regex{libgcc.*}
os.lib.libXp        PASS    /usr/lib/libXmu.so.6      regex{libX.*}
os.space.var        PASS    "38GB"                    " [dir:root=/va
r/ibm/common/acsi"
unit:MB]1.0
os.space.usr        PASS    "38GB"                    " [dir:root=/us
r/ibm/common/acsi"
unit:MB]200
os.space.tmp        PASS    36GB                      30MB
env.classpath.derbyJAR PASS    False                     False
network.pingSelf    PASS    True                       True
env.classpath.derbyJAR PASS    False                     False
network.pingLocalhost PASS    True                       True
os.package.compat-libstdc++-33 PASS    compat-libstdc++-33-3.2.3-61 compat-libstdc+
+-33
TOTAL ALL SPECIFIED COMPONENTS:
Evaluation          PASS/FAIL Result          Expected Result
/                   PASS    38.00GB                   201MB
/tmp                PASS    36.00GB                   30MB

Prereq Check Overall Result: FAIL
[root@aclinux15 20110927-0849]#

```

圖 10. 在 UNIX 系統上執行 Script 及設定 detail 參數

如果您未設定 **detail** 參數，則 Scanner 僅會在指令行介面中顯示 PASS 或 FAIL 結果。



```
C:\ Command Prompt
C:\prs\precheck_windows_20110927>prereq_checker.bat DMO
FAIL
Details also available in C:\prs\precheck_windows_20110927\result.txt
```

圖 11. 在 Windows 系統上執行 Script 及設定 detail 參數

### [outputDir="ips\_output\_dir"]

此選用參數指出您要設定 Prerequisite Scanner 之掃描結果及日誌檔的輸出目錄。

當執行 Prerequisite Scanner Script 並設定選用 **outputDir** 參數時，Prerequisite Scanner 會將結果文字、XML 及日誌檔輸出至參數值所指定的目錄中。此值即是文件中提到的 *ips\_output\_dir*。

如果未設定此參數，則預設輸出位置為 *ips\_root*。

如果您選擇從 CD、DVD 或唯讀網路磁碟機執行 Prerequisite Scanner，則必須使用此參數指定位置。您必須具有寫入權，才能寫入到 *ips\_output\_dir*；否則，Prerequisite Scanner 會失敗。

**重要：**如果輸出目錄不存在，則 Prerequisite Scanner 會建立該目錄。您必須具有寫入權，才能建立或寫入 Prerequisite Scanner 儲存檔案的輸出目錄。

### [xmlResult]

此選用參數指出您除了要將結果輸出至純文字測試結果檔案之外，還要將結果輸出至 XML 結果檔案。

當執行 Prerequisite Scanner Script 並設定選用 **xmlResult** 參數時，Prerequisite Scanner 會將結果輸出至 *ips\_output\_dir/result.xml* 檔案。

如果未設定此參數，則結果僅會輸出至純文字檔案。

### [PATH="product\_root"]

此選用參數指出產品的安裝目錄。

**重要：**在 Windows 上，請勿將路徑僅設為磁碟機字母，即 C:。請確保設定有效的路徑。

如果您未設定 **path** 參數，則 Scanner 會檢查 IBM Tivoli 產品的預設安裝目錄：

- 在 **UNIX** 系統上：*/opt/ibm/itm*
- 在 **Windows** 系統上：*C:\IBM\itm*

### [-p Product\_Code.instance.parameter=value,...]

選用 **-p** 旗標指出之前的參數必須傳遞到 Script 檔，以進行其他必備項目檢查。**<Product\_Code>** 是產品型號。只有每一組 *instance.parameter=value* 會傳遞到 Script。您可以傳遞多組參數，以逗點進行分隔。

向其傳遞參數的 Script 由下列選項判定：

- 如果有 **Product\_Code** 字首，則參數會傳遞到具有相關聯 **Product\_Code** 的 Script
- 如果沒有 **Product\_Code** 字首，則參數會傳遞到共用收集器。

範例 1-p KUD.inst1.DB2\_INST\_OWNER=db2inst1, KUD.inst2.DB2\_INST\_OWNER=db2inst2  
此具有參數的旗標會將 db2inst1.DB2\_INST\_OWNER=db2inst1 及 db2inst2.DB2\_INST\_OWNER=db2inst2 傳遞到 KUD.**Product\_Version**.bat Script 檔。

範例 2

```
-p SERVER=IP.PIPE://mymachine:1918
```

此具有參數的旗標會將 SERVER=IP.PIPE://mymachine:1918 傳遞到共用收集器，以檢查埠。

註：此 Script 接受 **-p** 中的參數作為 tacmd createNode。

您可以在 *ips\_root/lib/common\_configuration* 中設定 SERVER、PROTOCOL、PORT、BACKUP 及 BSERVER 參數。Prerequisite Scanner 會在 common\_configuration 檔案中設定的參數之前，優先處理指令行介面傳遞的參數。

## [debug]

此選用參數指出您要在執行 Prerequisite Scanner 時開啓除錯。

當執行 Prerequisite Scanner Script 並設定選用 **debug** 參數時，Prerequisite Scanner 會在日誌檔中輸出詳細處理資訊、警告與錯誤訊息以及掃描結果。該日誌檔即是 *ips\_output\_dir/prs.debug* 檔案（在 UNIX 系統上）及 *ips\_output\_dir/precheck.log* 檔案（在 Windows 系統上）。

**重要：**依預設會關閉 Scanner 除錯。

## [trace]

（僅限 UNIX 系統）此選用參數指出您要在執行 Prerequisite Scanner 時開啓追蹤記載。

當執行 Prerequisite Scanner Script 並設定選用 **trace** 參數時，Prerequisite Scanner 會在 *ips\_output\_dir/prs.trc* 檔案中輸出追蹤資訊。

**重要：**依預設會關閉 Scanner 追蹤記載。

---

## 從指令行執行 Prerequisite Scanner

您可以從指令行介面執行 IBM Prerequisite Scanner 並輸入 Script 的相關輸入參數。

### 開始之前

請確保您已檢查產品的安裝文件或技術說明，以獲取執行 Prerequisite Scanner 之前必須執行的其他步驟。例如，您可能需要設定環境變數，以指示 Prerequisite Scanner 要在目標電腦上安裝哪些元件或特性，並且要檢查哪些必備項目。

### 程序

1. 開啓指令行介面並開啓 *ips\_root* 目錄。
2. 執行 Prerequisite Scanner Script 檔 **prereq\_checker**，如下所示：



## UNIX

```
./prereq_checker.sh  
"Product_Code [Product_Version] [,Product_CodeN [Product_VerN]]..."  
[detail]  
[outputDir="ips_output_dir"]  
[xmlResult]  
[PATH="product_root"]  
[-p Product_Code.instance.parameter=value,...]
```

下列範例針對 Autonomic Deployment Engine 執行 Prerequisite Scanner，方法是使用配置檔及其相關聯的產品型號 ADE：

```
./prereq_checker.sh  
ADE 072000  
detail  
PATH=/opt/ibm/tivoli
```

## Windows

```
prereq_checker.bat  
"Product_Code [Product_Version] [,Product_CodeN [Product_VerN]]..."  
[detail]  
[outputDir="ips_output_dir"]  
[xmlResult]  
[PATH="product_root"]  
[-p Product_Code.instance.parameter=value,...]
```

下列範例使用產品型號 COX 及 COY 為 Tivoli Provisioning Manager for Windows 2003 及 2008 執行 Prerequisite Scanner。

```
prereq_checker.bat  
"COX, COY 07200000"  
detail  
PATH="D:\ibm\tivoli"  
-p SERVER=IP.PIPE://mytems:1234
```

下列範例使用產品型號 KZE 為 Tivoli zEnterprise® Monitoring Agent 執行 Prerequisite Scanner。它還會將結果及日誌檔的位置設為 *ips\_output\_dir*，方法是使用選用 **outputDir** 參數。

**重要：**如果您選擇從 CD、DVD 或唯讀網路磁碟機執行 Prerequisite Scanner，則必須使用 **outputDir** 參數指定位置。您必須具有寫入權限以寫入到 *ips\_output\_dir*；否則，Prerequisite Scanner 會失敗。

## Windows

```
prereq_checker.bat  
"KZE 06230000"  
outputDir="%TEMP%\ips"
```

## UNIX

```
./prereq_checker.sh  
"KZE 06230000"  
outputDir="/tmp/ips"
```

Scanner 會將 result.txt file 及 precheck.log 檔案輸出至下列位置：

- 在 Windows 系統上：D:\temp\ips，其中 TEMP 是暫存資料夾的環境變數。
- 在 UNIX 系統上：/tmp/ips

**重要：**如果輸出目錄不存在，Prerequisite Scanner 會建立該目錄。您必須具有寫入權限，以建立或寫入 Prerequisite Scanner 儲存檔案的輸出目錄。

---

## 共用目錄位置

共用目錄具有路徑名稱變數。

### IBM Prerequisite Scanner 安裝目錄

*ips\_root* 說明安裝 Prerequisite Scanner 的位置。此位置可在安裝期間指定。

### Prerequisite Scanner 輸出目錄

*ips\_output\_dir* 說明儲存 Prerequisite Scanner 之掃描結果及日誌檔的位置。執行 Scanner 時可使用 **outputDir** 輸入參數指定此位置。如果未設定此參數，則預設輸出位置為 *ips\_root*。

**註：**Prerequisite Scanner 會在執行時建立暫存檔，但將在 Scanner 完成執行之前刪除這些檔案。這些暫存檔位於 *ips\_output\_dir/temp* 子目錄中。Scanner 也會刪除 *ips\_output\_dir/temp* 子目錄，除非該子目錄包含僅在 UNIX 系統上產生的除錯及追蹤檔案。

---

## 第 5 章 Prerequisite Scanner 疑難排解

藉由在建立自訂必備項目內容時使用日誌檔及記載功能，您可以對 IBM Prerequisite Scanner 中的問題進行疑難排解。

Prerequisite Scanner 會依據掃描的結果以及其是否由於錯誤而必須結束產生回覆碼。這些回覆碼會寫入日誌檔。例如，如果 Prerequisite Scanner 由於無法讀取配置檔而無法執行掃描，它會產生回覆碼 2。

---

### Windows 系統疑難排解

當執行 IBM Prerequisite Scanner 時，依預設它會建立日誌檔。其中包含 Scanner 依順序執行之每個步驟及函數的詳細資訊。檔案還包含時間戳記，包括每個函數及步驟的開始及結束時間。您可以除錯並檢查日誌檔，以判定發生錯誤的位置及時間。

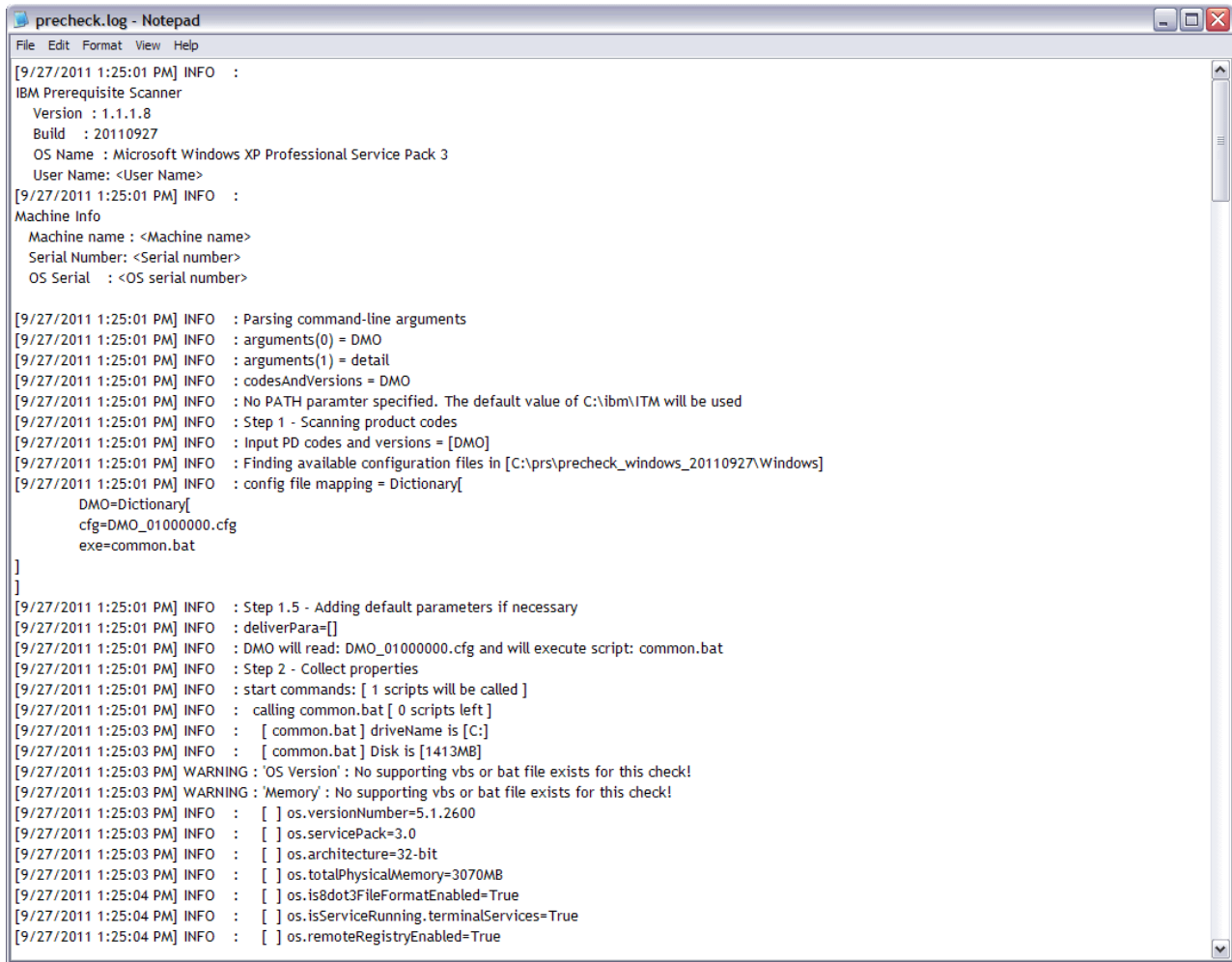
Prerequisite Scanner 在 *ips\_output\_dir/precheck.log* 檔案中輸出處理資訊、警告與錯誤訊息以及掃描結果。當執行 Prerequisite Scanner Script 並設定選用 **debug** 參數時，Prerequisite Scanner 會在此檔案中輸出其他除錯訊息。

第 66 頁的圖 12 顯示設定選用 **debug** 參數時的日誌檔範例，第 67 頁的圖 13 顯示未設定參數時的日誌檔。

```
precheck.log - Notepad
File Edit Format View Help
[9/27/2011 1:27:34 PM] INFO :
IBM Prerequisite Scanner
  Version : 1.1.1.8
  Build : 20110927
  OS Name : Microsoft Windows XP Professional Service Pack 3
  User Name: <User Name>
[9/27/2011 1:27:34 PM] INFO :
Machine Info
  Machine name : <Machine name>
  Serial Number: <Serial number>
  OS Serial : <OS serial number>

[9/27/2011 1:27:34 PM] INFO : Parsing command-line arguments
[9/27/2011 1:27:34 PM] INFO : arguments(0) = DMO
[9/27/2011 1:27:34 PM] INFO : arguments(1) = detail
[9/27/2011 1:27:34 PM] INFO : arguments(2) = debug
[9/27/2011 1:27:34 PM] INFO : codesAndVersions = DMO
[9/27/2011 1:27:34 PM] INFO : No PATH paramter specified. The default value of C:\ibm\ITM will be used
[9/27/2011 1:27:34 PM] DEBUG : Detected operating system types are [Windows|Windows Workstation|Windows XP]
[9/27/2011 1:27:34 PM] DEBUG : Detected operating system version [5.1.2600]
[9/27/2011 1:27:34 PM] DEBUG : Detected service pack level [3.0]
[9/27/2011 1:27:34 PM] INFO : Step 1 - Scanning product codes
[9/27/2011 1:27:34 PM] INFO : Input PD codes and versions = [DMO]
[9/27/2011 1:27:34 PM] INFO : Finding available configuration files in [C:\prs\precheck_windows_20110927\Windows]
[9/27/2011 1:27:34 PM] INFO : config file mapping = Dictionary[
  DMO=Dictionary[
    cfg=DMO_01000000.cfg
    exe=common.bat
  ]
]
[9/27/2011 1:27:34 PM] INFO : Step 1.5 - Adding default parameters if necessary
[9/27/2011 1:27:34 PM] INFO : deliverPara=[]
[9/27/2011 1:27:34 PM] INFO : DMO will read: DMO_01000000.cfg and will execute script: common.bat
[9/27/2011 1:27:34 PM] INFO : Step 2 - Collect properties
[9/27/2011 1:27:34 PM] INFO : start commands: [ 1 scripts will be called ]
[9/27/2011 1:27:34 PM] DEBUG : The config file is: C:\prs\precheck_windows_20110927\Windows\DMO_01000000.cfg
[9/27/2011 1:27:34 PM] INFO : calling common.bat [ 0 scripts left ]
[9/27/2011 1:27:36 PM] INFO : [ common.bat ] driveName is [C:]
[9/27/2011 1:27:36 PM] INFO : [ common.bat ] Disk is [1413MB]
[9/27/2011 1:27:36 PM] DEBUG : Processing this line from cfg file: [OS Version=regex{Windows .*}]
[9/27/2011 1:27:36 PM] DEBUG : Take the first part of the line: [OS Version]
[9/27/2011 1:27:36 PM] DEBUG : See if a corresponding vbs or bat file exists for [OS Version]
[9/27/2011 1:27:36 PM] DEBUG : See if a corresponding vbs or bat file exists for [OS Versio]
```

圖 12. 具有除錯資料的 precheck.log 檔案



```
precheck.log - Notepad
File Edit Format View Help
[9/27/2011 1:25:01 PM] INFO :
IBM Prerequisite Scanner
  Version : 1.1.1.8
  Build : 20110927
  OS Name : Microsoft Windows XP Professional Service Pack 3
  User Name : <User Name>
[9/27/2011 1:25:01 PM] INFO :
Machine Info
  Machine name : <Machine name>
  Serial Number : <Serial number>
  OS Serial : <OS serial number>

[9/27/2011 1:25:01 PM] INFO : Parsing command-line arguments
[9/27/2011 1:25:01 PM] INFO : arguments(0) = DMO
[9/27/2011 1:25:01 PM] INFO : arguments(1) = detail
[9/27/2011 1:25:01 PM] INFO : codesAndVersions = DMO
[9/27/2011 1:25:01 PM] INFO : No PATH paramter specified. The default value of C:\ibm\ITM will be used
[9/27/2011 1:25:01 PM] INFO : Step 1 - Scanning product codes
[9/27/2011 1:25:01 PM] INFO : Input PD codes and versions = [DMO]
[9/27/2011 1:25:01 PM] INFO : Finding available configuration files in [C:\prs\precheck_windows_20110927\Windows]
[9/27/2011 1:25:01 PM] INFO : config file mapping = Dictionary[
  DMO=Dictionary[
    cfg=DMO_01000000.cfg
    exe=common.bat
  ]
]

[9/27/2011 1:25:01 PM] INFO : Step 1.5 - Adding default parameters if necessary
[9/27/2011 1:25:01 PM] INFO : deliverPara=[]
[9/27/2011 1:25:01 PM] INFO : DMO will read: DMO_01000000.cfg and will execute script: common.bat
[9/27/2011 1:25:01 PM] INFO : Step 2 - Collect properties
[9/27/2011 1:25:01 PM] INFO : start commands: [ 1 scripts will be called ]
[9/27/2011 1:25:01 PM] INFO : calling common.bat [ 0 scripts left ]
[9/27/2011 1:25:03 PM] INFO : [ common.bat ] driveName is [C:]
[9/27/2011 1:25:03 PM] INFO : [ common.bat ] Disk is [1413MB]
[9/27/2011 1:25:03 PM] WARNING : 'OS Version': No supporting vbs or bat file exists for this check!
[9/27/2011 1:25:03 PM] WARNING : 'Memory': No supporting vbs or bat file exists for this check!
[9/27/2011 1:25:03 PM] INFO : [ ] os.versionNumber=5.1.2600
[9/27/2011 1:25:03 PM] INFO : [ ] os.servicePack=3.0
[9/27/2011 1:25:03 PM] INFO : [ ] os.architecture=32-bit
[9/27/2011 1:25:03 PM] INFO : [ ] os.totalPhysicalMemory=3070MB
[9/27/2011 1:25:04 PM] INFO : [ ] os.is8dot3FileFormatEnabled=True
[9/27/2011 1:25:04 PM] INFO : [ ] os.isServiceRunning_terminalServices=True
[9/27/2011 1:25:04 PM] INFO : [ ] os.remoteRegistryEnabled=True
```

圖 13. 不具有除錯資料的 *precheck.log* 檔案

## UNIX 系統疑難排解

在 UNIX 系統上依預設會停用訊息寫入到日誌檔。藉由使用 **debug** 及 **trace** 輸入參數，您可以啟用除錯或追蹤功能。Scanner 會將除錯及追蹤資料寫入不同的日誌檔，並使用時間戳記來標記步驟或函數的開始及結束時間。您可以使用這兩個檔案，以關聯及疑難排解特定問題、函數或必備項目檢查。

### 除錯日誌檔

當執行 Prerequisite Scanner Script 並設定選用 **debug** 參數時，Prerequisite Scanner 會在 *ips\_output\_dir/temp/prs.debug* 檔案中輸出詳細處理、警告與錯誤訊息以及掃描結果。其中包含 Scanner 依順序執行之每個步驟及函數的詳細資訊。檔案還包含時間戳記，包括每個函數及步驟的開始及結束時間。*ips\_output\_dir/temp* 子目錄還包含過渡期間 *result1.txt* 及 *result2.txt* 檔案，其為最終 *ips\_output\_dir/result.txt* 檔案提供輸入。您可以使用這些過渡期間檔案，以判定特定必備項目內容結果的問題。

```
root@aqlinux15:~/prs/20110927-0849
File Edit View Terminal Tabs Help
Build : 20110927
OS Name: Linux
[2011.09.27 10.12.15 ] [main()] - Entered
[2011.09.27 10.12.15 ] ==== Step 1: Detecting OS...
[2011.09.27 10.12.15 ] OS Detected: Red Hat Enterprise Linux Server release 5.5 {32-bit}
[2011.09.27 10.12.15 ] product_version: DMO
[2011.09.27 10.12.15 ] [AutoOsDetection()] - Entered
[2011.09.27 10.12.15 ] [Param] ProductInfo:DMO
[2011.09.27 10.12.15 ] [Param] OSInfo:Red Hat Enterprise Linux Server release 5.5 {32-bit}
[2011.09.27 10.12.15 ] [Param] CPU Arch:Kernel=i686
[2011.09.27 10.12.15 ] Finding product code in product.cfg
[2011.09.27 10.12.15 ] product code found :
[2011.09.27 10.12.15 ] Found DMO code in product.cfg
[2011.09.27 10.12.15 ] Finding OS Arch and CPU Type
[2011.09.27 10.12.15 ] Found OS Arch = 32-bit, CPU Type=
[2011.09.27 10.12.15 ] Calling config_parser.sh...
[2011.09.27 10.12.15 ] [config_parser.sh] - Entered
[2011.09.27 10.12.15 ] [Param] OSInfo:Red Hat Enterprise Linux Server release 5.5 {32-bit}
[2011.09.27 10.12.15 ] [Param] ProductCode:DMO
[2011.09.27 10.12.15 ] [Param] OSArch:Arch=32-bit
[2011.09.27 10.12.16 ] [Param] CPUArch:CPU=
[2011.09.27 10.12.16 ] [Param] Version:version=
[2011.09.27 10.12.16 ] [Param] XXX:Kernel=i686
[2011.09.27 10.12.16 ] Forming parse array...
[2011.09.27 10.12.16 ] [Form_Parse_String] - Entered
[2011.09.27 10.12.16 ] [Param] OSInfo:Red Hat Enterprise Linux Server release 5.5 {32-bit}
[2011.09.27 10.12.16 ] [Param] ProductCode:DMO
[2011.09.27 10.12.16 ] [Param] OSArch:Arch=32-bit
[2011.09.27 10.12.16 ] [Param] CPU:CPU=
[2011.09.27 10.12.16 ] [Param] CPUArch:Kernel=i686
[2011.09.27 10.12.16 ] Form_Parse_String - ParseArray: [OSType:UNIX][OSType:LINUX][OSType:RedHat][OSType:RedHatEnterpriseLinuxServer][OS
Type:RedHatEnterpriseLinuxServer5.*][OSType:RedHatEnterpriseLinuxServer5.5][OSArch:32-bit][CPUArch:i686]
[2011.09.27 10.12.16 ] [Form_Parse_String] - Exit
[2011.09.27 10.12.16 ] Reading config file and parsing using parse array...
[2011.09.27 10.12.16 ] [Read_configFile()] - Entered
[2011.09.27 10.12.16 ] [Param] ConfigFile:/root/prs/20110927-0849/UNIX_Linux/DMO_0750000.cfg-Master
[2011.09.27 10.12.16 ] [Param] Product:DMO
[2011.09.27 10.12.17 ] Writing DBType=Oracle to DMO_0750000.cfg
[2011.09.27 10.12.21 ] Found Env Var - TPAE_DB_Server
```

圖 14. UNIX 系統上的 prs.debug 檔案

### 追蹤日誌檔

當執行 Prerequisite Scanner Script 並設定選用 **trace** 參數時，Prerequisite Scanner 會在 *ips\_output\_dir/temp/prs.trc* 檔案中輸出追蹤資訊。其中包含 Scanner 依順序執行之每個函數的資訊。檔案還包含時間戳記，包括每個函數的開始及結束時間。

```

root@aqlinux15:~/prs/20110927-0849
File Edit View Terminal Tabs Help
Build : 20110927
OS Name: Linux
[2011.09.27 10.19.58 ] [main()] - Entered:
[2011.09.27 10.19.58 ] [AutoOsDetection()] - Entered:
[2011.09.27 10.19.58 ] [config_parser.sh] - Entered:
[2011.09.27 10.19.59 ] [Form_Parse_String] - Entered:
[2011.09.27 10.19.59 ] [Form_Parse_String] - Exit:
[2011.09.27 10.19.59 ] [Read_configFile()] - Entered:
[2011.09.27 10.20.05 ] [Read_configFile()] - Exit:
[2011.09.27 10.20.05 ] [config_parser.sh] - Exit:
[2011.09.27 10.20.05 ] [AutoOsDetection()] - Exit:
[2011.09.27 10.20.05 ] [packageTest.sh] - Entered:
[2011.09.27 10.20.25 ] [NFScheck()] - Entered:
[2011.09.27 10.20.25 ] [NFScheck()] - Exit:
[2011.09.27 10.20.25 ] [NFScheck()] - Entered:
[2011.09.27 10.20.25 ] [NFScheck()] - Exit:
[2011.09.27 10.20.25 ] [NFScheck()] - Entered:
[2011.09.27 10.20.26 ] [NFScheck()] - Exit:
[2011.09.27 10.20.26 ] Starting: DBType
[2011.09.27 10.20.26 ] Executing: DBType
[2011.09.27 10.20.26 ] Finished: DBType
[2011.09.27 10.20.26 ] Done : DBType
[2011.09.27 10.20.26 ] Starting: DB2_Version
[2011.09.27 10.20.26 ] Executing: DB2_Version.sh
[2011.09.27 10.20.26 ] Finished: DB2_Version.sh
[2011.09.27 10.20.26 ] Done : DB2_Version
[2011.09.27 10.20.26 ] Starting: DBType
[2011.09.27 10.20.26 ] Executing: DBType
[2011.09.27 10.20.26 ] Finished: DBType
[2011.09.27 10.20.26 ] Done : DBType
[2011.09.27 10.20.26 ] Starting: DBType
[2011.09.27 10.20.26 ] Executing: DBType
[2011.09.27 10.20.26 ] Finished: DBType
[2011.09.27 10.20.26 ] Done : DBType
[2011.09.27 10.20.26 ] Starting: DB2_Version
[2011.09.27 10.20.26 ] Executing: DB2_Version.sh
[2011.09.27 10.20.26 ] Finished: DB2_Version.sh
[2011.09.27 10.20.26 ] Done : DB2_Version
[2011.09.27 10.20.26 ] Starting: DBType

```

圖 15. UNIX 系統上的 prs.trc 檔案

## 執行問題

您可以使用執行問題核對清單，以對執行 Prerequisite Scanner 時可能遇到的問題進行疑難排解。

執行具有選用 **debug** 及 **trace** 輸入參數的 Prerequisite Scanner Script，以協助對問題進行除錯。

表 13. 執行問題核對清單

檢查	項目
<input type="checkbox"/>	當在指令行上設定選用 <b>outputDir</b> 參數且輸出目錄不存在時，Prerequisite Scanner 會建立目錄。您必須具有寫入權限，以建立或寫入 Prerequisite Scanner 儲存檔案的輸出目錄。如果您沒有寫入權限，會將下列錯誤訊息寫入指令行介面： ERROR: Cannot create files in output directory <i>ips_output_dir</i> . Exit.
<input type="checkbox"/>	在執行 Prerequisite Scanner 之前，請確保您要執行 Prerequisite Scanner 的磁碟以及儲存結果的輸出目錄空間未滿；否則，下列錯誤訊息會寫入指令行介面： ERROR: Cannot create files in output directory <i>ips_output_dir</i> . Exit.
<input type="checkbox"/>	如果 Prerequisite Scanner 產生回覆碼 2，可能會發生 Script 用法或收集器錯誤。請檢閱與此錯誤碼相關聯的原因。如果 Script 用法發生錯誤，請使用正確的語法傳回 Prerequisite Scanner。

### 相關概念:

在 UNIX 系統上依預設會停用訊息寫入到日誌檔。藉由使用 **debug** 及 **trace** 輸入參數，您可以啓用除錯或追蹤功能。Scanner 會將除錯及追蹤資料寫入不同的日誌檔，並使用時間戳記來標記步驟或函數的開始及結束時間。您可以使用這兩個檔案，以關聯及疑難排解特定問題、函數或必備項目檢查。

Prerequisite Scanner 會依據掃描的結果以及其是否由於錯誤而必須結束產生回覆碼。這些回覆碼會寫入日誌檔。

prereq\_checker Script 會執行 IBM Prerequisite Scanner，並根據執行該 Script 時指定的一組參數，來檢查必備項目。

---

## 回覆碼

Prerequisite Scanner 會依據掃描的結果以及其是否由於錯誤而必須結束產生回覆碼。這些回覆碼會寫入日誌檔。

Prerequisite Scanner 基於一組定義的輸出產生回覆碼，如下所示：

回覆碼	說明
0	當 Prerequisite Scanner 順利執行並且所有掃描結果為 PASS 時，傳回 0。
1	當 Prerequisite Scanner 順利執行但一或多個必備項目檢查傳回 FAIL 時，傳回 1。
2	當 Prerequisite Scanner 未順利執行且由於如下已分類的錯誤必須結束時，傳回 2： <ul style="list-style-type: none"><li>• Script 用法錯誤</li><li>• 收集器錯誤</li><li>• 其他錯誤</li></ul>

### Script 用法錯誤

Prerequisite Scanner 可在執行 Script 時由於下列任何用法錯誤而結束：

- **Product\_Code** 輸入參數無效；例如，找不到它或它不是支援的格式。
- **Product\_Code** 及 **Product\_Version** 輸入參數的型樣無效；例如，引號中包括了型號及版本以外的內容，或型樣未以引號括住。
- **Product\_Version** 輸入參數無效；例如，產品版本不全是數值字元。
- 指令行介面中未輸入輸入參數。
- 在指令行介面中輸入的語法不正確；例如，輸入了不支援的指令行引數。
- 未輸入必要的 **Product\_Code** 輸入參數。

### 收集器錯誤

Prerequisite Scanner 可由於下列任何收集器錯誤而結束：

- *ips\_output\_dir/temp* 目錄中找不到收集器的暫存結果檔案。
- 收集器的 Script 檔未正確執行。



## 其他錯誤

Prerequisite Scanner 可因為使用者不具有 *ips\_output\_dir* 輸出目錄的寫入權而結束。

### 相關概念:

IBM Prerequisite Scanner 會產生使用下列屏幕及易懂的檔案格式輸出：輸出至指令行介面、除錯及追蹤日誌以及文字與 XML 結果檔案。



## 附錄 A. 產品型號參照

IBM Prerequisite Scanner 使用多字元代碼 *product\_code*，以識別產品、個別支援的平台以及作業系統的版本。 *ips\_root/codename.cfg* 檔案包含名稱/值配對，以代表產品的產品型號、其支援的平台以及作業系統的版本。

表 14 說明一組現行預先定義的產品型號。

**限制：**IBM Tivoli Monitoring 及 Tivoli Composite Application Manager 具有 Prerequisite Scanner 視為保留項目的產品型號。這些型號不能用作 Prerequisite Scanner 產品型號，除非是指其關聯的 IBM Tivoli Monitoring 及 Tivoli Composite Application Manager 代理程式。如需產品型號的相關資訊，請參閱 ITM 6.X 產品型號 Technote。

**限制：**僅限 UNIX：在檔案中輸入產品型號的值時，請避免使用 `for`。它是保留字，可影響 Prerequisite Scanner 的執行方式。

表 14. 預先定義的產品型號

預先定義的產品型號	平台	產品版本、平台、作業系統
ADE	所有	Autonomic Deployment Engine
BSM	所有	Tivoli Business Service Manager
CDB	所有	Tivoli Composite Application Manager (ITCAM) for Applications : DB2
COA	UNIX	Tivoli Provisioning Manager for UNIX
COB	AIX	Tivoli Provisioning Manager for AIX
COC	AIX	Tivoli Provisioning Manager for AIX V5.3.0.0 {64 位元}
COD	AIX	Tivoli Provisioning Manager for AIX 6.1
COE	Linux	Tivoli Provisioning Manager for Linux
COF	Linux	Tivoli Provisioning Manager for Red Hat Linux
COG	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 Red Hat Enterprise Linux 5 x86 64 位元)
COH	Linux	Tivoli Provisioning Manager for Red Hat Enterprise Linux 5 System z® 64 位元
COI	Linux	Tivoli Provisioning Manager for SUSE 10
COJ	Solaris	Tivoli Provisioning Manager 7.2 版 (適用於 Solaris)
COK	HP-UX	Tivoli Provisioning Manager 7.2 版 (適用於 HP-UX)
COL	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 SUSE zSeries® 10)
COM	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 SUSE 11)
CON	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 SUSE zSeries 11)
COX	Windows	Tivoli Provisioning Manager 7.2 版 (適用於 Windows 2008)
COY	Windows	Tivoli Provisioning Manager 7.2 版 (適用於 Windows 2003)
COZ	Windows	Tivoli Provisioning Manager 7.2 版 (適用於 Windows)

表 14. 預先定義的產品型號 (繼續)

預先定義的產品型號	平台	產品版本、平台、作業系統
DMO	所有	Prerequisite Scanner 展示
GYM	UNIX	IBM Tivoli Netcool® Performance Manager
KCJ	Windows	Tivoli Enterprise Portal Client
	UNIX	Tivoli Enterprise Portal Client for UNIX
KCQ	Windows	Tivoli Enterprise Portal Server
	UNIX	Tivoli Enterprise Portal Server for UNIX
KHD	所有	Warehouse Proxy Agent
KHE	UNIX	Warehouse Proxy Agent for UNIX
KIS	UNIX	Tivoli Composite Application Manager (ITCAM) for Transactions: Internet Service Monitoring
KLZ	UNIX	Tivoli Monitoring Operating System Agent for Linux
KM6	Windows	IBM Tivoli Composite Application Manager Agent for WebSphere MQ File Transfer Edition
KMQ	所有	Tivoli Composite Application Manager Agent for WebSphere MQ
KMS	Windows	Tivoli Enterprise Monitoring Server
	UNIX	Tivoli Enterprise Monitoring Server for UNIX
KNT	Windows	Tivoli Monitoring Operating System Agent for Windows
	UNIX	Windows OS monitoring Agent for UNIX
KOR	Windows	Tivoli Monitoring Agent for Oracle
KQI	所有	Tivoli Composite Application Manager Agent for WebSphere Message Broker
KSY	Windows	摘要及刪改代理程式
	UNIX	摘要及刪改代理程式 for UNIX
KUD	Windows	Tivoli Monitoring Agent for DB2
	UNIX	Tivoli Monitoring Agent for DB2
KT0	所有	Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Reporter
KTU	所有	Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Collector
KT3	所有	Tivoli Composite Application Manager (ITCAM) for Transactions: Application Management Console
KT4	所有	Tivoli Composite Application Manager (ITCAM) for Transactions: Client Response Time
KT5	所有	Tivoli Composite Application Manager (ITCAM) for Transactions: Web Response Time
KT6	所有	Tivoli Composite Application Manager (ITCAM) for Transactions: Robotic Response Time
KZE	所有	Tivoli zEnterprise Monitoring Agent
LCM	Windows	Tivoli License Compliance Manager
	UNIX	Tivoli License Compliance Manager for UNIX

表 14. 預先定義的產品型號 (繼續)

預先定義的產品型號	平台	產品版本、平台、作業系統
NCI	所有	Tivoli Netcool/Impact
NOC	所有	Tivoli Netcool/OMNibus 伺服器元件及桌面元件
NOD	所有	Tivoli Netcool/OMNibus 桌面元件
NOS	所有	Tivoli Netcool/OMNibus 伺服器元件
PAE	所有	Tivoli Process Automation Engine
TAD	Windows	Tivoli Asset Discovery for Distributed
	UNIX	Tivoli Asset Discovery for Distributed for UNIX
TCR	所有	Tivoli Common Reporting
TPM	所有	Tivoli Provisioning Manager



## 附錄 B. 配置檔參照

IBM Prerequisite Scanner 提供了預先定義的一組可編輯的配置檔。這些檔案位於 *ips\_root/UNIX\_Linux* 或 *ips\_root/Windows* 中。檔案具有 .cfg 副檔名。

表 15 列出目前支援之預先定義的配置檔。

表 15. 預先定義的配置檔

配置檔	平台	產品版本、平台、作業系統
ADE_01040000.cfg	所有	Autonomic Deployment Engine 1.4 版
BSM_04210000.cfg	所有	Tivoli Business Service Manager 4.2.1 版
BSM_06100000.cfg	所有	Tivoli Business Service Manager 6.1 版
CDB_06220000.cfg	所有	Tivoli Composite Application Manager (ITCAM) for Applications : DB2 6.2.2 版
COA_07200000.cfg	UNIX	Tivoli Provisioning Manager 7.2 版 (適用於 UNIX)
COB_07200000.cfg	AIX	Tivoli Provisioning Manager 7.2 版 (適用於 AIX)
COC_07200000.cfg	AIX	Tivoli Provisioning Manager 7.2 版 (適用於 AIX 5.3.0.0 版 {64 位元})
COD_07200000.cfg	AIX	Tivoli Provisioning Manager 7.2 版 (適用於 AIX 6.1)
COE_07200000.cfg	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 Linux)
COF_07200000.cfg	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 Red Hat Linux)
COG_07200000.cfg	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 Red Hat Enterprise Linux 5 x86 64 位元)
COH_07200000.cfg	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 Red Hat Enterpris Linux 5 System z 64 位元)
COI_07200000.cfg	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 SUSE 10)
COJ_07200000.cfg	Solaris	Tivoli Provisioning Manager 7.2 版 (適用於 Solaris)
COK_07200000.cfg	HP-UX	Tivoli Provisioning Manager 7.2 版 (適用於 HP-UX)
COL_07200000.cfg	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 SUSE zSeries 10)
COM_07200000.cfg	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 SUSE 11)
CON_07200000.cfg	Linux	Tivoli Provisioning Manager 7.2 版 (適用於 SUSE zSeries 11)
COX_07200000.cfg	Windows	Tivoli Provisioning Manager 7.2 版 (適用於 Windows 2008)
COY_07200000.cfg	Windows	Tivoli Provisioning Manager 7.2 版 (適用於 Windows 2003)
COZ_07200000.cfg	Windows	Tivoli Provisioning Manager 7.2 版 (適用於 Windows)
DMO_00000000.cfg	所有	Prerequisite Scanner 展示
DMO_01000000.cfg	所有	Prerequisite Scanner 1.0 版展示
GYM_01030200.cfg	UNIX	IBM Tivoli Netcool Performance Manager 1.3.2 版
KCJ_06200000.cfg	Windows	Tivoli Enterprise Portal Client 6.2 版
KCJ_06210000.cfg	UNIX	Tivoli Enterprise Portal Client 6.2.1 版
KCJ_06220000.cfg	所有	Tivoli Enterprise Portal Client 6.2.2 版
KCQ_06200000.cfg	Windows	Tivoli Enterprise Portal Server 6.2 版
KCQ_06210000.cfg	UNIX	Tivoli Enterprise Portal Server 6.2.2 版

表 15. 預先定義的配置檔 (繼續)

配置檔	平台	產品版本、平台、作業系統
KCQ_06220000.cfg	所有	Tivoli Enterprise Portal Server 6.2.2 版
KHD_06200000.cfg	Windows	Warehouse Proxy Agent 6.2 版
KHD_06210000.cfg	所有	Warehouse Proxy Agent 6.2.1 版
KHD_06220000.cfg	所有	Warehouse Proxy Agent 6.2.2 版
KHE_06220000.cfg	UNIX	Warehouse Proxy Agent 6.2.2 版
KIS_07200000.cfg	所有	Tivoli Composite Application Manager (ITCAM) for Transactions : Internet Service Monitoring 7.2 版
KIS_07300000.cfg	所有	Tivoli Composite Application Manager (ITCAM) for Transactions : Internet Service Monitoring 7.3 版
KLZ_06210000.cfg	UNIX	Tivoli Monitoring Operating System Agent for Linux 6.2.1 版
KLZ_06220000.cfg	UNIX	Tivoli Monitoring Operating System Agent for Linux 6.2.2 版
KM6_0701000000.cfg	Windows	Tivoli Composite Application Manager Agent for WebSphere MQ File Transfer Edition 7.1 版
KMQ_0701000000.cfg	所有	Tivoli Composite Application Manager Agent for WebSphere MQ 7.1 版
KMS_06200000.cfg	Windows	Tivoli Enterprise Monitoring Server 6.2 版
KMS_06210000.cfg	所有	Tivoli Enterprise Monitoring Server 6.2.1 版
KMS_06220000.cfg	所有	Tivoli Enterprise Monitoring Server 6.2.2 版
KNT_06200000.cfg	Windows	Tivoli Monitoring Operating System Agent for Windows 6.2 版
KNT_06210000.cfg	Windows	Tivoli Monitoring Operating System Agent for Windows 6.2.1 版
KNT_06220000.cfg	Windows	Tivoli Monitoring Operating System Agent for Windows 6.2.2 版
KOR_06220000.cfg	Windows	Tivoli Monitoring Agent for Oracle 6.2.2 版
KQI_0701000000.cfg	所有	Tivoli Composite Application Manager Agent for WebSphere Message Broker 7.1 版
KSY_06200000.cfg	Windows	摘要及刪改代理程式 6.2 版
KSY_06210000.cfg	所有	摘要及刪改代理程式 6.2.1 版
KSY_06220000.cfg	所有	摘要及刪改代理程式 6.2.2 版
KTO_07200000.cfg	UNIX	Tivoli Composite Application Manager (ITCAM) for Transactions : Transaction Reporter 7.2 版
KTO_07200200.cfg	Windows	Tivoli Composite Application Manager (ITCAM) for Transactions : Transaction Reporter 7.2.2 版
KTO_07300000.cfg	UNIX	Tivoli Composite Application Manager (ITCAM) for Transactions : Transaction Reporter 7.3 版
KTU_07200000.cfg	UNIX	Tivoli Composite Application Manager (ITCAM) for Transactions : Transaction Collector 7.2 版
KTU_07200200.cfg	Windows	Tivoli Composite Application Manager (ITCAM) for Transactions : Transaction Collector 7.2.2 版
KTU_07300000.cfg	UNIX	Tivoli Composite Application Manager (ITCAM) for Transactions : Transaction Collector 7.3 版
KT3_07300000.cfg	所有	Tivoli Composite Application Manager (ITCAM) for Transactions : Application Management Console 7.3 版
KT4_07300000.cfg	所有	Tivoli Composite Application Manager (ITCAM) for Transactions : Client Response Time 7.3 版



表 15. 預先定義的配置檔 (繼續)

配置檔	平台	產品版本、平台、作業系統
KT5_07300000.cfg	所有	Tivoli Composite Application Manager (ITCAM) for Transactions : Web Response Time 7.3 版
KT6_07300000.cfg	所有	Tivoli Composite Application Manager (ITCAM) for Transactions : Robotic Response Time 7.3 版
KUD_06100000.cfg	Windows	Tivoli Monitoring Agent for DB2 6.1 版
KUD_06200000.cfg	所有	Tivoli Monitoring Agent for DB2 6.2 版
KUD_06210000.cfg	所有	Tivoli Monitoring Agent for DB2 6.2.1 版
KUD_06220000.cfg	所有	Tivoli Monitoring Agent for DB2 6.2.2 版
KZE_06020300.cfg	所有	Tivoli zEnterprise Monitoring Agent 6.2.3 版
LCM_01000000.cfg	所有	Tivoli License Compliance Manager 1.0 版
LCM_02300000.cfg	所有	Tivoli License Compliance Manager 2.3 版
NCI_06100000.cfg	所有	Tivoli Netcool/Impact 6.1 版
NOC_07310000.cfg	所有	Tivoli Netcool/OMNIBus 伺服器元件及桌面元件 7.3.1 版
NOD_07310000.cfg	所有	Tivoli Netcool/OMNIBus 桌面元件 7.3.1 版
NOS_07310000.cfg	所有	Tivoli Netcool/OMNIBus 伺服器元件 7.3.1 版
PAE_07500000.cfg	所有	Tivoli Process Automation Engine
TAD_07200000.cfg	所有	Tivoli Asset Discovery for Distributed 7.2 版
TAD_07220000.cfg	所有	Tivoli Asset Discovery for Distributed 7.2.2 版
TCR_02010100.cfg	所有	Tivoli Common Reporting
TPM_07210000.cfg	所有	Tivoli Provisioning Manager 7.2.1 版



## 附錄 C. 必備項目內容參照

此參照說明每一個預先定義之種類的軟硬體必備項目的基本必備項目內容。

表 16 說明預先定義的軟硬體必備項目種類。

表 16. 預先定義的必備項目內容種類

資料種類	說明	必要的字首 ID	參照
共用	共用資料內容檢查共用必備項目，如處理器速度、RAM、磁碟及暫存空間。	無	第 82 頁的『共用資料內容』
Autonomic Deployment Engine	Autonomic Deployment Engine 資料內容會檢查 Autonomic Deployment Engine 必備項目，如安裝單位。	de	第 85 頁的『Autonomic Deployment Engine 資料內容』
已安裝軟體	已安裝軟體資料內容會檢查已安裝軟體必備項目，如 Windows 登錄中註冊的程式以及是否已安裝 cygwin 與 gskit。	無	第 99 頁的『已安裝軟體資料內容』
使用者	使用者資料內容會檢查使用者必備項目，如已登入使用者是否具有管理權限或是否為 root 使用者。	user	第 99 頁的『使用者資料內容』
作業系統	作業系統資料內容會檢查作業系統必備項目，如版本、架構、總記憶體、可用記憶體以及實體記憶體總數。	os	第 89 頁的『作業系統資料內容』
連線功能	連線功能資料內容會檢查連線功能必備項目，如 Telnet 是否在執行中並且 Scanner 可連接到哪些 IP 位址及埠。	無	第 86 頁的『連線功能資料內容』
網路	網路資料內容會檢查所有平台可用的網路必備項目，如是否存在可用的埠。	network	第 88 頁的『網路資料內容』
Windows 網路	Windows 網路資料內容會檢查網路必備項目，如機器上是否已啓用 NetBIOS 及 DHCP 以及 ping 內容。	network	第 99 頁的『Windows 網路資料內容』
UNIX 網路	UNIX 網路資料內容會檢查網路必備項目，如機器上是否已啓用 NetBIOS 及 DHCP 以及 ping 內容。	network	第 100 頁的『UNIX 網路資料內容』
Internet Explorer	Microsoft Internet Explorer 資料內容會檢查 Internet Explorer 必備項目，如版本。	internetExplorer	第 87 頁的『Internet Explorer 資料內容』
資料庫伺服器 DB2	DB2 資料內容會檢查 DB2 必備項目，如版本。	DB2	第 86 頁的『DB2 資料內容』
資料庫伺服器 MS SQL	MS SQL Server 資料內容會檢查 MS SQL Server 必備項目，如版本。	mssql	第 87 頁的『MS SQL Server 資料內容』
資料庫伺服器 Oracle	Oracle 資料內容會檢查 Oracle 必備項目，如版本。	Oracle	第 88 頁的『Oracle 資料內容』
環境變數	環境變數會檢查環境變數必備項目，如是否已設定環境變數。	env	第 100 頁的『環境變數資料內容』

## 共用資料內容

共用資料內容檢查共用必備項目，如 CPU 速度、RAM、磁碟及暫存空間。針對 Windows 系統，它會使用主要 IBM Prerequisite Scanner Script。對於 UNIX 系統，它會使用主要 Prerequisite Scanner Script 及共用收集器 `ips_root/Unix_Linux/common.sh`。

表 17 說明共用資料必備項目內容。此種類的必備項目內容不需要字首 ID。

表 17. 共用資料必備項目內容

必備項目內容	平台	說明	有效值
CPU Name	所有	CPU 名稱且僅用於在結果中顯示	不適用
CpuArchitecture	UNIX	作業系統的架構	以逗點分隔多個受支援值的字串，例如： x86_64,s390x,ppc64,AMD64
DBType	所有	<p>檢查機器上安裝的資料庫伺服器類型。</p> <p>僅限於 UNIX 系統上的 Oracle：收集器預期 ORACLE_BASE 及 ORACLE_HOME 環境變數在 \$HOME/.profile 檔案中設定，例如：</p> <pre>export ORACLE_BASE=/home/oracle/app/oracle/product/11.2.0/ export ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/dbhome_1</pre> <p>其中，\$HOME 必須是 Oracle 使用者的起始目錄 /home/oracle。</p>	<p>值可以是下列其中任一類型：</p> <ul style="list-style-type: none"> <li>代表任意資料庫伺服器類型的字串，例如： any</li> <li>代表資料庫伺服器類型的字串，例如： Oracle</li> <li>regex{str} 具有輸入參數 str 的正規表示式，代表資料庫伺服器類型的搜尋型樣，例如： regex{.*MSSQL.* DB2.*}</li> </ul> <p>檢查 Windows 系統上資料庫伺服器類型是 MS SQL 還是 DB2。</p> <ul style="list-style-type: none"> <li>代表無資料庫伺服器類型的字串，例如： unknown</li> </ul>
DBTypeDetails	所有	<p>機器上已安裝的資料庫伺服器類型。</p> <p>僅限於 UNIX 系統上的 Oracle：收集器預期 ORACLE_BASE 及 ORACLE_HOME 環境變數在 \$HOME/.profile 檔案中設定，例如：</p> <pre>export ORACLE_BASE=/home/oracle/app/oracle/product/11.2.0/ export ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/dbhome_1</pre> <p>其中，\$HOME 必須是 Oracle 使用者的起始目錄 /home/oracle。</p> <p>必備項目內容會將資料庫伺服器類型的詳細資料（即資料庫伺服器類型、安裝位置及版本）寫入 result.txt 檔案。多個資料庫伺服器類型的詳細資料由冒號分隔</p>	<p>值可以是下列其中任一類型：</p> <ul style="list-style-type: none"> <li>代表任意資料庫伺服器類型的字串，例如： any</li> <li>代表一個資料庫伺服器類型的字串，例如： DB2</li> <li>regex{str} 具有輸入參數 str 的正規表示式，代表資料庫伺服器類型的搜尋型樣，例如： regex{.*MSSQL.* DB2.*}</li> </ul> <p>檢查 Windows 系統上資料庫伺服器類型是 MS SQL 還是 DB2。</p>

表 17. 共用資料必備項目內容 (繼續)

必備項目內容	平台	說明	有效值
Disk	Windows	<p>可用的磁碟空間數量，具有下列選用資格屬性：</p> <ul style="list-style-type: none"> <li>dir 屬性，用於判定要檢查之目錄的路徑</li> <li>unit 屬性，用於判定要使用的磁碟空間單位</li> </ul>	<p>值可以是下列其中任一類型：</p> <ul style="list-style-type: none"> <li>具有下列限定元格式的字串：                     <pre>[dir:dir_path, unit:unit_name] disk_space</pre> <p>例如：</p> <pre>Disk= [dir:C:\Program Files\IBM\SQLLIB, unit:MB]1431</pre> </li> <li>數字格式單位為 MB 或 GB：                     <pre>disk_spaceMB GB</pre> <p>例如：</p> <pre>Disk=250MB</pre> </li> </ul>
Disk	UNIX	可用的磁碟空間數量	<p>數字格式單位為 GB 或 MB，例如：</p> <pre>2GB</pre>
intel.cpu	所有	Intel 處理器的 CPU 速度	<p>數字格式單位為 GHz，且僅在 Windows 上單位為 MHz，例如：</p> <pre>2GHz</pre>
Memory	所有	<p>機器上目前可用的實體記憶體總量。</p> <p><b>提示：</b>藉由使用作業系統種類中預先定義的必備項目內容，單獨檢查可用的實體及虛擬記憶體數量。</p>	<p>數字格式單位為 GB 或 MB，例如：</p> <pre>300MB</pre>

表 17. 共用資料必備項目內容 (繼續)

必備項目內容	平台	說明	有效值
OS Version	所有	機器上執行的作業系統完整名稱及版本； 或者，您可以使用正規表示式，以傳遞代表作業系統多個變式的字串。 <b>提示：</b> 使用此必備項目內容搭配 <code>os.servicePack</code> 及 <code>os.architecture</code> ，以檢查現行服務套件及系統架構。	<p>值可以是下列其中任一類型：</p> <ul style="list-style-type: none"> <li>代表多個版本的字串，其每一個版本由逗點分隔，例如： RedHat Enterprise Linux 6.*, SuSE Linux Enterprise Server 11, SuSE Linux Enterprise Server 10, SuSE Linux Enterprise Server 9, AIX V6.1,AIX V5.3</li> </ul> <p><b>限制：</b> 在 Windows 系統上，僅在正規表示式中支援 * 萬用字元。</p> <ul style="list-style-type: none"> <li><code>regex{str}</code> 具有輸入參數 <code>str</code> 的正規表示式，代表版本的搜尋型樣，例如： <code>regex{Windows 200[3-8]}</code></li> </ul> <p>檢查實際 OS 是否符合從 Windows 2003 到 Windows 2008 的任意版本。 <code>regex{Red Hat*.*}</code></p> <p>檢查實際 OS 是否符合 Red Hat Linux 的變式。</p> <p><b>註：</b> 值可以使用第 2 頁的表 1 中所述的特殊字元。</p>
numCPU	所有	電腦上的核心或獨立處理器數目。如果此工具掃描電腦，沒有發現任何核心，或是它發現的處理器沒有一個是核心，則它會傳回找不到結果。	數字，例如 4
risc.cpu	UNIX	RISC 處理器的 CPU 速度	數字格式單位為 GHz，例如： 1.4GHz
Temp	UNIX	指定 <i>Temp</i> 檔案系統的可用磁碟空間	數字格式單位為 GB 或 MB，例如： 300MB

**相關概念:**

Prerequisite Scanner 會依據電腦上是否已執行 Tivoli Monitoring 或 Tivoli Composite Application Manager 以不同方式處理 Memory 必備項目內容的檢查。

**相關參考:**

作業系統資料內容會檢查作業系統必備項目，如版本、架構、總記憶體、可用記憶體以及實體記憶體總數。僅對於 Windows 系統，它會使用 `ips_root/lib` 目錄中的作業系統 VBScript 收集器，其檔名具有 `os` 字首 ID。僅對於 UNIX 系統，它會使用 `ips_root/UNIX_Linux` 目錄中的 UNIX 作業系統收集器，且其檔名具有 `os` 字首 ID。

**記憶體必備項目內容及 Tivoli Monitoring 代理程式的系統行為**

Prerequisite Scanner 會依據電腦上是否已執行 Tivoli Monitoring 或 Tivoli Composite Application Manager 以不同方式處理 Memory 必備項目內容的檢查。

如果已安裝代理程式，Prerequisite Scanner 會基於新的及現有配置檔之期望值的差異（如果現有配置檔仍位於電腦上）使用 Memory 必備項目內容的期望值；否則，它會依預設行為處理期望值。

當執行 Prerequisite Scanner 以檢查要升級或重新安裝之 Tivoli Monitoring 代理程式的必備項目時，它首先會檢查電腦上是否已在執行代理程式。如果代理程式處於執行中，Prerequisite Scanner 會搜尋與執行中代理程式之現有版本相關聯的配置檔。依據該搜尋的結果會發生下列行為：

- 如果它找不到配置檔，Prerequisite Scanner 會假設之前尚未掃描目標環境；因此，Prerequisite Scanner 會使用新配置檔中指定之 Memory 必備項目內容的期望值，其會遵循預設行為。Prerequisite Scanner 會將此期望值寫入結果輸出。
- 如果它找到配置檔，Prerequisite Scanner 會將現有版本之 Memory 必備項目內容的期望值與新版本之配置檔中的期望值作比較。如果值之間有不同，且新值大於現有的期望值，Prerequisite Scanner 會將此不同設為期望值。Prerequisite Scanner 會將期望值差異寫入結果輸出。例如，代理程式第 1 版的配置檔指定 1 GB 作為期望值。代理程式第 2 版的新配置檔會指定 1.5 GB 作為期望值；因此 Prerequisite Scanner 會使用 0.5 GB 作為期望值差異。

---

## Autonomic Deployment Engine 資料內容

Autonomic Deployment Engine 資料內容會檢查 Autonomic Deployment Engine 必備項目，如安裝單位。僅對於 Windows 系統，它會使用 `ips_root/lib/` 目錄中的 Autonomic Deployment Engine 收集器，且其檔名具有 `de` 字首。僅對於 UNIX 系統，它會使用 `ips_root/UNIX_Linux` 目錄中的 UNIX Autonomic Deployment Engine 收集器，其檔名具有 `de` 字首。

表 18 說明必備項目內容。此種類的必備項目內容需要 `de` 字首 ID。

表 18. *Autonomic Deployment Engine* 資料內容

必備項目內容	平台	說明	有效值
<code>de.installed</code>	所有	檢查是否已安裝	布林，例如： <code>true false</code>

表 18. *Autonomic Deployment Engine* 資料內容 (繼續)

必備項目內容	平台	說明	有效值
de.installationUnit	所有	使用 <b>listIU - v</b> 指令檢查是否已安裝指定的安裝單位	<p>值可以是下列其中任一類型：</p> <ul style="list-style-type: none"> <li>代表單一安裝單位的字串，例如，Tivoli Integrated Portal 的安裝單位： C37109911C8A11D98E1700061BDE7AEA, B24209911C8A11D98E1700061BDE7AEA</li> <li>代表多個安裝單位的字串，例如： 5FFE79F918DF3BA0D67511FD3F7C358E</li> <li>regex {str}：具有輸入參數 str 的正規表示式，代表安裝單位、版本及安裝路徑的搜尋型樣；例如，若要檢查 WebSphere Application Server 的安裝單位、版本及 Tivoli Integrated Portal 的安裝路徑，搜尋型樣如下所示：</li> <li>regex{.*C00DA95AFD9B7E0397153CD944B5A255.*6.1.0.2100.*SIU eWAS.*C:\\IBM\\tivoli\\tip.*}</li> </ul> <p>註：您也可以使用環境變數代表安裝路徑；例如，藉由使用 TIPHOME 環境變數取代路徑，搜尋型樣為：</p> <pre>regex{.*C00DA95AFD9B7E0397153CD944B5A255.*6.1.0.2100.*SIU eWAS.*%TIPHOME%.*}</pre> <ul style="list-style-type: none"> <li>代表多個檢查的多個 regex {str} 引數，例如： regex{.*C37109911C8A11D98E1700061BDE7AEA.*}, regex{.*B24209911C8A11D98E1700061BDE7AEA.*}</li> </ul>

## 連線功能資料內容

連線功能資料內容會檢查連線功能必備項目，如 Telnet 是否在執行中並且 Scanner 可連接到哪些 IP 位址及埠。對於 Windows 系統，它會使用連線功能收集器 *ips\_root/lib/connectivity\_plug.vbs*。對於 UNIX 系統，它會使用主要 IBM Prerequisite Scanner Script 及連線功能收集器 *prs\_root/Unix\_Linux/connectivity\_plug.sh*。輸出僅會傳遞到除錯日誌檔。

## DB2 資料內容

DB2 資料內容會檢查 DB2 必備項目，如版本。僅對於 Windows 系統，它會使用 DB2 收集器 *ips\_root/lib/db2\_version\_plug.bat*。僅對於 UNIX 系統，它會使用 *ips\_root/Unix\_Linux* 目錄中的 UNIX DB2 收集器，其檔名具有 db2 字首。

表 19 說明 DB2 必備項目內容。此類型的必備項目內容需要 DB2 字首 ID。

表 19. *DB2* 資料內容

必備項目內容	平台	說明	有效值
DB2 Version	所有	機器上目前已安裝的 DB2 版本	<p>字串，例如：</p> <p>v9.5.100.179FP4</p>



表 19. DB2 資料內容 (繼續)

必備項目內容	平台	說明	有效值
db2.home.space	UNIX	DB2 起始目錄的可用磁碟空間	數字格式單位為 GB，例如： 8GB

## MS SQL Server 資料內容

MS SQL Server 資料內容會檢查 MS SQL Server 必備項目，如版本及位置。僅對於 Windows 系統，它會使用 *ips\_root/Windows* 目錄中的 MS SQL Server 收集器，且其檔名具有 *mssql* 字首。

表 20 說明 MS SQL Server 必備項目內容。此類型的必備項目內容需要 *mssql* 字首 ID。

表 20. MS SQL Server 資料內容

必備項目內容	平台	說明	有效值
mssql.Client	Windows	檢查機器上目前已安裝的 MS SQL 用戶端版本	預期字串值可以是以逗點分隔的多個版本，例如： 10.50.1600.1  註：值可以使用第 2 頁的表 1 中所述的特殊字元。
mssql.Server	Windows	檢查機器上目前已安裝的 MS SQL Server 版本	預期字串值可以是以逗點分隔的多個版本，例如： 10.50.1600.1  註：值可以使用第 2 頁的表 1 中所述的特殊字元。
mssql.Server.Location	Windows	檢查 MS SQL 資料庫伺服器的起始目錄	字串，例如： any

## Internet Explorer 資料內容

Microsoft Internet Explorer 資料內容會檢查 Internet Explorer 必備項目，如版本。它會使用 Internet Explorer 收集器 *ips\_root/lib/internetExplorer\_plug.vbs*。

表 21 說明 Internet Explorer 必備項目內容。此類型的必備項目內容需要 *internetExplorer* 字首 ID。

表 21. Internet Explorer 資料內容

必備項目內容	說明	有效值
internetExplorer.version	機器上已安裝的 Internet Explorer 版本	數字格式，例如 7.0+ 註：值可以使用第 2 頁的表 1 中所述的特殊字元。

## 網路資料內容

網路資料內容會檢查所有平台可用的網路必備項目，如是否存在可用的埠。它會使用 *ips\_root/lib* 目錄中的網路收集器，且其檔名中具有 *network* 字首 ID。

表 22 說明可用於所有平台的網路必備項目內容。此種類的必備項目內容需要 *network* 字首 ID。

表 22. 網路資料內容

必備項目內容	平台	說明	有效值
<code>network.availablePorts. app_type</code>	所有	使用此命名慣例，以檢查埠或埠範圍是否可用於 <i>app_type</i> 應用程式類型。檢查未接聽哪些埠，例如： <ul style="list-style-type: none"><li>• <code>network.availablePorts.DB2</code> 會檢查 DB2 資料庫伺服器的埠，其中 <i>app_type</i> 是 DB2</li><li>• <code>network.availablePorts.WAS</code> 會檢查 WebSphere Application Server 的埠，其中 <i>app_type</i> 是 WAS</li></ul>	正整數，例如： <ul style="list-style-type: none"><li>• <code>network.availablePorts.DB2 = 50000-50005</code></li><li>• <code>network.availablePorts.WAS = 8080</code></li></ul> 註：值可以使用第 2 頁的表 1 中所述的特殊字元。
<code>network.portsInUse. app_type</code>	所有	使用此命名慣例，以檢查埠或埠範圍是否處於使用中（針對 <i>app_type</i> 應用程式類型）。檢查正在接聽哪些埠，例如： <ul style="list-style-type: none"><li>• <code>network.availablePorts.DB2</code> 會檢查 DB2 資料庫伺服器的埠，其中 <i>app_type</i> 是 DB2</li><li>• <code>network.availablePorts.WAS</code> 會檢查 WebSphere Application Server 的埠，其中 <i>app_type</i> 是 WAS</li></ul>	正整數，例如： <ul style="list-style-type: none"><li>• <code>network.portsInUse.DB2 = 50900-50905</code></li><li>• <code>network.portsInUse.WAS = 8080</code></li></ul> 註：值可以使用第 2 頁的表 1 中所述的特殊字元。
<code>network.validate HostsFile</code>	Windows	檢查 <i>hosts</i> 檔中列出的所有主機的格式是否為： <i>IP_Address Host_Name Short_Name</i> 其中： <ul style="list-style-type: none"><li>• <i>IP_Address</i> 是電腦的 IP，例如 127.0.0.1</li><li>• <i>Host_Name</i> 是電腦的完整主機名稱，例如 localhost.localdomain</li><li>• <i>Short_Name</i> 是電腦的簡稱，例如 localhost</li></ul>	布林值，例如 True

## Oracle 資料內容

Oracle 資料內容會檢查 Oracle 必備項目，如版本。僅對於 Windows 系統，它會使用 Oracle 收集器。僅對於 UNIX 系統，它會使用 *ips\_root/UNIX\_Linux* 目錄中的 UNIX Oracle 收集器，其檔名具有 *oracle* 字首。僅對於 Window 系統，它會使用 *ips\_root/lib* 目錄中的 Windows Oracle 收集器，其檔名具有 *oracle* 字首。

表 23 說明 Oracle 必備項目內容。此類型的必備項目內容需要 oracle 字首 ID。

表 23. Oracle 資料內容

必備項目內容	平台	說明	有效值
ORACLE Version	Windows	檢查機器上目前已安裝的 Oracle 版本	預期字串值可以是以逗點分隔的多個版本，例如： 9.2, 10.1, 10.2
oracle.Client	所有	檢查機器上目前已安裝的 Oracle 用戶端版本	預期字串值可以是以逗點分隔的多個版本，例如： 9.2.0.8+  註：值可以使用第 2 頁的表 1 中所述的特殊字元。
oracle.Client.Location	所有	檢查 Oracle 用戶端的起始目錄	字串，例如： /opt/oracle/products/10.1.0/client_1
oracle.Server	所有	檢查機器上目前已安裝的 Oracle 伺服器版本	預期字串值可以是以逗點分隔的多個版本，例如： 10.2.0.4g,11g R1  註：值可以使用第 2 頁的表 1 中所述的特殊字元。
oracle.Server.Location	所有	檢查 Oracle 資料庫伺服器的起始目錄	字串，例如： /opt/oracle/product/10.1.0/Db_1

## 作業系統資料內容

作業系統資料內容會檢查作業系統必備項目，如版本、架構、總記憶體、可用記憶體以及實體記憶體總數。僅對於 Windows 系統，它會使用 *ips\_root/lib* 目錄中的作業系統 VBScript 收集器，其檔名具有 os 字首 ID。僅對於 UNIX 系統，它會使用 *ips\_root/UNIX\_Linux* 目錄中的 UNIX 作業系統收集器，且其檔名具有 os 字首 ID。

表 24 說明作業系統必備項目內容。此類型的必備項目內容需要 os 字首 ID。

表 24. 作業系統資料內容

必備項目內容	平台	說明	有效值
os.architecture	所有	檢查系統架構	32-bit 64-bit
os.automount	UNIX	檢查自動裝載特性是否正確運作	布林值，例如： True
os.autoUpdateEnabled	Windows	檢查 Windows Update 是否已自動啟用；如果已啟用，則傳回 True	布林值，例如： True
os.availableMemory	Windows	檢查目前可用但作業系統未用的虛擬記憶體數量	數字格式單位為 MB，例如： 900MB

表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.dir.dir_name	UNIX	<p>基於下列資格屬性檢查 <i>dir_name</i> 檔案系統：</p> <ul style="list-style-type: none"> <li>• <i>dir</i> 屬性，用於判定要檢查的檔案系統</li> <li>• <i>type</i> 屬性，用於檢查要檢查的檔案系統屬性，例如，該檔案系統之存取權限的 <i>octal_digits</i> 八進位表示法</li> </ul> <p><i>dir_name</i> 可代表如下內容：</p> <ul style="list-style-type: none"> <li>• tmp</li> <li>• home</li> </ul>	<p>具有下列限定元格式的字串：</p> <p>[dir:<i>dir_name</i>, type:permission] <i>octal_digits</i>+</p> <p>例如，若要檢查起始目錄是否具有 drwxr-xr-x 權限：</p> <p>os.dir.home=[dir:/home, type:permission]755+</p>
os.diskquota		<p>檢查已登入使用者的磁碟使用配額；傳回單位為 KB 的配額值或 Unlimited</p>	<p>值可以是下列其中任一類型：</p> <ul style="list-style-type: none"> <li>• 代表 KB 的數字，例如 414000</li> <li>• 代表無限制磁碟限額的字串，例如 Unlimited</li> </ul>
os.expectLink	UNIX	<p>檢查機器上是否可用 TCL 的 Expect 延伸；如果其為可用狀態，則傳回 Available</p> <p>註：os.file.expect 必備項目內容會檢查機器上是否已安裝 Expect 延伸。</p>	Available Unavailable
os.file.script_name	UNIX	<p>檢查機器上是否可用 <i>script_name</i> Script。 <i>script_name</i> 可代表如下內容：</p> <ul style="list-style-type: none"> <li>• bash</li> <li>• expect</li> <li>• gzip</li> <li>• tar</li> </ul>	<p>布林值，例如：</p> <p>True</p>
os.Firefox	UNIX	<p>檢查機器上是否已安裝 Mozilla Firefox；如果已安裝，則傳回 Available</p>	Available Unavailable
os.FreePagingSpace	UNIX	<p>檢查可用頁面快取的大小總計</p>	<p>數字格式單位為 MB 或 GB，例如：</p> <p>4GB+</p> <p>註：值可以使用第 2 頁的表 1 中所述的特殊字元。</p>
os.ftpusers	UNIX	<p>檢查 root 使用者是否在 ftpusers 檔案中列出，該檔案判定不容許其 FTP 登入專用權的使用者；如果未列出使用者，則傳回 Available</p>	Available Unavailable
os.gnu.tar	UNIX	<p>檢查機器上是否可用 GNU Tar 公用程式；如果已安裝，則傳回 Available</p>	Available Unavailable
os.hostformat	UNIX	<p>檢查 /etc/host 中的項目是否為正確格式</p>	<p>布林值，例如：</p> <p>True</p>

表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.iodevicestatus	AIX	檢查非同步 I/O (aio0) 的狀態，即用於加強 I/O 作業效能的核心處理程序；如果為可用狀態，則傳回 Available	Available Unavailable
os.is8dot3FileFormatEnabled	Windows	檢查是否已自動套用 8.3 檔名格式；如果已套用，則傳回 True	布林值，例如： True
os.localhostInHostsFile	所有	檢查 hosts 檔中是否有一個項目將本端主機對映至 127.0.0.1 IP 位址，例如： 127.0.0.1 localhost	布林值，例如： True
os.isServiceRunning.service_name	Windows	使用此命名慣例，以檢查 <i>service_name</i> 是否在機器上處於執行中。 <i>service_name</i> 可代表如下內容： <ul style="list-style-type: none"> <li>• 「遠端登錄服務」的 remoteRegistry</li> <li>• 「DNS 用戶端服務」的 DNSClient</li> <li>• 「遠端桌面服務」或「終端服務」的 terminalServices</li> </ul>	布林值，例如： True
os.kernelMode	AIX	檢查支援核心及未限定模式的 CPU 架構	32-bit 64-bit
os.kernelParameters	Linux	檢查核心參數是否可用於作業系統	Available Unavailable
os.kernelversion	Linux	檢查 Linux 作業系統之核心的版次	數字格式，例如 2.6
os.largeFile	UNIX	檢查大型檔案支援	布林值，例如： True
os.ldLibPath	UNIX	檢查是否存在 LD_LIBRARY_PATH 環境變數並以冒號結束，即 os.ldLibPath=[endsWith=:]	Available Unavailable
os.level	AIX	檢查 AIX 作業系統是否大於第 10 版（針對 AIX 5.3 版或更高版本）或大於第 3 版（針對 AIX 6.1 版）	布林值，例如： True

表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.lib.lib_name_version	UNIX	<p>檢查機器上已安裝支援的 <i>lib_name</i> 程式庫版本。以粗體表示、代表 <i>lib_name_version</i> 的字串或正規表示式，例如：</p> <ul style="list-style-type: none"> <li>• 32 位元 <b>libstdc++.so.#</b> 程式庫</li> <li>• 64 位元 <b>libstdc++.so.#</b> 程式庫</li> <li>• 32 位元 <b>libXft.so.#</b> 程式庫</li> <li>• 32 位元 <b>libXtst.so.#</b> 程式庫</li> <li>• 64 位元 <b>libaio.so.#</b> 程式庫</li> <li>• 32 位元 <b>x1C.rte</b> XLC 執行時期層次</li> <li>• 32 位元 <b>x1C.aix50.rte</b> XLC 執行時期 (適用於 AIX 5.3 版)</li> <li>• 32 位元 <b>x1C.aix61.rte</b> XLC 執行時期 (適用於 AIX 6.1 版)</li> <li>• AIX IOCP <b>bos.iocp.rte</b> 程式庫</li> <li>• <b>bos.loc.iso.en_us</b> : AIX 基本作業系統的 ISO 代碼檔案集</li> </ul>	<p>值可以是下列其中任一類型：字串，例如：</p> <ul style="list-style-type: none"> <li>• /usr/lib/libstdc++.so.# 作為 32 位元 libstdc++.so.# 程式庫的值</li> <li>• /usr/lib64/libaio.so.# 作為 64 位元 libaio.so.# 程式庫的值</li> <li>• x1C.aix50.rte.9.0.0.8+ 作為 32 位元 x1C.aix50.rte XLC 執行時期 (適用於 AIX 5.3 版) 的值</li> <li>• bos.loc.iso.en_us 表示 ISO 代碼檔案集</li> </ul> <p>regex {<i>str</i>} : 具有輸入參數 <i>str</i> 的正規表示式，代表程式庫名稱的搜尋型樣，例如： regex{.*libgcc.*}</p> <p>檢查針對該作業系統是否存在 GCC 低層次執行時期程式庫 libgcc 的版本。 註：值可以使用第 2 頁的表 1 中所述的特殊字元。</p>
os.loginVariable	UNIX	檢查 PATH 及 SUPATH 變數中是否已設定 root 使用者的預設路徑；如果已設定，則傳回 Available	Available Unavailable
os.maximoDirectory	UNIX	檢查 /export/home/maximo 目錄是否可用	Available Unavailable
os.maximoDirOwner	UNIX	檢查 /export/home/maximo 目錄的擁有者	maximo
os.maximumProcesses	UNIX	檢查可針對每位使用者執行的處理程序最大數目	數字，例如 2048
os.MozillaVersion	UNIX	檢查機器上與 os.Firefox 必備項目內容不同的 Mozilla Firefox 特定版本	<p>數字格式，例如 3.0+</p> <p>註：該值可使用在 第 2 頁的表 1 上概要的特殊字元。</p>

表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.mountcheck	UNIX	<p>檢查檔案系統是否已基於下列資格屬性裝載：</p> <ul style="list-style-type: none"> <li>• <b>drive</b> 屬性，用於判定作為已安裝檔案系統的目錄</li> <li>• <b>nosuid</b> 屬性，用於判定如果檔案系統已裝載是否已設定裝載選項</li> </ul>	<p>具有下列限定元格式的字串：</p> <p>[drive:dir_name, mount_option: false[true] True False</p> <p>例如，若要檢查是否已裝載 /home 目錄且未設定 nosuid 選項：</p> <p>os.mountcheck=[drive:/home, nosuid:false]True</p>
os.package.package_name	UNIX	<p>檢查機器上已安裝支援的 <i>package_name</i> 套件版本。以粗體表示、代表 <i>package_name</i> 的字串，例如：</p> <ul style="list-style-type: none"> <li>• <b>bash</b> shell</li> <li>• <b>expect</b> (適用於 TCL 延伸套件)</li> <li>• <b>libgcc</b> (適用於 GCC 低層次執行時期套件)</li> <li>• <b>openssh</b> (適用於開放程式碼安全 Shell)</li> <li>• <b>openssl</b> (適用於 SSL/TLS 的開放程式碼工具箱)</li> <li>• <b>perl</b> (適用於 Perl Scripting 套件)</li> <li>• <b>rpm</b> (適用於 RPM 或 RPM Build 套件)</li> <li>• <b>telnet</b> (適用於 Telnet 套件)</li> <li>• <b>wget</b> (適用於 GNU 檔案擷取套件)</li> </ul>	<p>字串，例如：</p> <ul style="list-style-type: none"> <li>• bash-3.2+ for bash shell</li> <li>• expect-1.2.0, 用於 Expect</li> <li>• libgcc-3.4.3-9, 用於 libgcc</li> <li>• openssh-5.0.0.5301-, 用於 openssh</li> <li>• openssl-4.2.0-, 用於 OpenSSL</li> <li>• perl-5.8.2, 用於 Perl</li> <li>• rpm</li> <li>• telnet</li> <li>• wget</li> </ul> <p>註：值可以使用第 2 頁的表 1 中所述的特殊字元。</p>
os.pagesize	UNIX	<p>檢查系統的分頁大小。</p>	<p>數字格式單位為 KB，例如：</p> <p>4KB</p> <p>註：值可以使用第 2 頁的表 1 中所述的特殊字元。</p>
os.RAMSize	所有	<p>檢查系統的 RAM</p>	<p>數字格式單位為 GB，例如 8GB</p>
os.SeaMonkeyVersion	UNIX	<p>檢查給定電腦上在 PATH 環境變數中設定了其路徑的特定 Mozilla SeaMonkey 版本</p>	<p>數字格式，例如 2.10</p> <p>註：值可以使用第 2 頁的表 1 中所述的特殊字元。</p>

表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.SELinux	Linux	基於下列資格屬性檢查 Security-Enhancement Linux 特性的強制執行狀態： <ul style="list-style-type: none"> <li>source 屬性，用於判定相關作業系統使用的指令</li> </ul>	值可以是下列其中任一類型： <ul style="list-style-type: none"> <li>具有下列限定元格式的字串：                      [source:Command] Disabled Enabled                       例如，若要檢查 Red Hat 或 SUSE 作業系統上是否已停用功能或功能是否具有權限狀態：                      os.SELinux=[source:Command]Disabled  </li> <li>不具有限定元的字串，其中作業系統是一般 Linux 變式：                      os.SELinux=Disabled</li> </ul>
os.servicePack	所有	檢查已安裝之服務套件的現行版本	數字格式，且具有 <i>majorVersion</i> 。 僅限 <i>minorVersion</i> 或 <i>majorVersion</i> 版本。  例如，若要檢查是否已安裝服務套件 2 或更新版本，2+ 註：該值可使用在 第 2 頁的表 1 中概要的特殊字元。
os.shell.default	UNIX	檢查是否已安裝了預設 shell script	代表 shell script 的字串，例如，bash
<p>os.space.dir_name 必備項目內容</p> <p>Prerequisite Scanner 具有三個 os.space.dir_name 內容變式：</p> <ul style="list-style-type: none"> <li>os.space.dir_name，可檢查無論已登入使用者始終是 root 還是非 root 使用者，指定的檔案系統是否具有足夠的可用磁碟空間。                       當您要檢查檔案系統的指定路徑時請使用此必備項目內容變式，但是這與已登入使用者是否始終為 root 使用者無關。                      註：您無法在單一配置檔中針對同一檔案系統（但使用者類型不同）兩次使用此變式；請使用其他兩個變式組合。</li> <li>os.space.dir_name_nonroot，可檢查非 root 使用者之指定檔案系統是否具有足夠的可用磁碟空間。                       當您以非 root 使用者身分登入且要明確檢查檔案系統的指定路徑時，請使用此必備項目內容變式。                      註：非 root 使用者應該是在目標系統上安裝產品的同一使用者。</li> <li>os.space.dir_name_root，可檢查 root 使用者之指定檔案系統是否具有足夠的可用磁碟空間。                       當您以 root 使用者身分登入且要明確檢查檔案系統的指定路徑時，請使用此必備項目內容變式。</li> </ul> <p>您可用在相同的配置檔中指定 os.space.dir_name_nonroot 及 os.space.dir_name_root 變式。Prerequisite Scanner 會在不適用之變式的實際結果資料格中輸出 NOT_REQ_CHECK_ID。例如，如果登入的是 root 使用者，Prerequisite Scanner 會針對 os.space.dir_name_nonroot 變式輸出 NOT_REQ_CHECK_ID。</p>			



表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.space.dir_name	UNIX	<p>基於下列一或多個資格屬性檢查指定 <code>&lt;dir_name&gt;</code> 檔案系統的可用磁碟空間：</p> <ul style="list-style-type: none"> <li>• <code>dir</code> 屬性，用於判定要檢查之檔案系統的路徑</li> <li>• <code>unit</code> 屬性，用於判定要使用的磁碟空間單位</li> </ul> <p><code>dir</code> 屬性的值取決於已登入使用者；因此，該值為名稱/值配對，以代表使用者類型，即 <code>root</code> 或非 <code>root</code> 以及相關聯路徑。</p> <p><code>dir_name</code> 可代表如下內容：</p> <ul style="list-style-type: none"> <li>• <code>home</code></li> <li>• <code>opt</code></li> <li>• <code>tmp</code></li> <li>• <code>usr</code></li> <li>• <code>var</code></li> </ul> <p>註：您無法在單一配置檔中針對同一檔案系統（但使用者類型不同）兩次使用此變式。請使用 <code>os.space.dir_name_nonroot</code> 及 <code>os.space.dir_name_root</code> 變式組合。</p>	<p>具有下列限定元格式的字串（適用於 <code>root</code> 使用者的檔案系統）：</p> <pre>[dir:root=&lt;dir_path&gt;, unit:&lt;unit_name&gt;] &lt;disk_space&gt;</pre> <p>例如：</p> <pre>os.space.usr= [dir:root=/usr/ibm/common/ acsi, unit:GB]200</pre> <p>具有下列限定元格式的字串（適用於非 <code>root</code> 使用者的檔案系統）：</p> <pre>[dir:non_root=&lt;dir_path&gt;, unit:&lt;unit_name&gt;] &lt;disk_space&gt;</pre> <p>例如：</p> <pre>os.space.home= [dir:non_root=USERHOME/ .acsi_HOST, unit:MB]200</pre> <p>具有下列限定元格式的字串（僅使用一個限定元）：</p> <pre>[dir:&lt;dir_path&gt;] &lt;disk_space&gt; MB</pre> <p>例如：</p> <pre>os.space.home= [dir:/home/sat]250MB</pre> <p>數字格式單位為 <code>MB</code> 或 <code>GB</code>，例如：</p> <pre>os.space.opt=11GB</pre>

表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.space.dir_name_nonroot	UNIX	<p>基於下列一或多個資格屬性檢查非 root 使用者之 <i>dir_name</i> 檔案系統的可用磁碟空間：</p> <ul style="list-style-type: none"> <li>dir 屬性，用於判定要檢查之檔案系統的路徑</li> <li>unit 屬性，用於判定要使用的磁碟空間單位</li> </ul> <p><i>dir_name</i> 可代表如下內容：</p> <ul style="list-style-type: none"> <li>home</li> <li>opt</li> <li>tmp</li> <li>usr</li> <li>var</li> </ul>	<p>具有下列限定元格式的字串（適用於非 root 使用者的檔案系統）：</p> <pre>[dir:non_root= dir_path, unit:unit_name] disk_space</pre> <p>例如：</p> <pre>os.space.home_nonroot= [dir:non_root= USERHOME/.acsi_HOST, unit:MB]200</pre> <p>具有 dir 資格屬性的字串（適用於非 root 使用者的檔案系統）：</p> <pre>[dir:non_root=dir_path] disk_spaceGB MB</pre> <p>例如：</p> <pre>os.space.opt_nonroot= [dir:non_root=/opt/IBM/ITM] 1024MB</pre>
os.space.dir_name_root	UNIX	<p>基於下列一或多個資格屬性檢查 root 使用者之 <i>dir_name</i> 檔案系統的可用磁碟空間：</p> <ul style="list-style-type: none"> <li>dir 屬性，用於判定要檢查之檔案系統的路徑</li> <li>unit 屬性，用於判定要使用的磁碟空間單位</li> </ul> <p><i>dir_name</i> 可代表如下內容：</p> <ul style="list-style-type: none"> <li>home</li> <li>opt</li> <li>tmp</li> <li>usr</li> <li>var</li> </ul>	<p>具有下列限定元格式的字串（適用於 root 使用者的檔案系統）：</p> <pre>[dir:root=dir_path, unit:unit_name] disk_space</pre> <p>例如：</p> <pre>os.space.usr_root= [dir:root= /usr/ibm/common/acsi, unit:GB]200</pre> <p>具有 dir 資格屬性的字串（適用於 root 使用者的檔案系統）：</p> <pre>[dir:root=dir_path] disk_spaceGB MB</pre> <p>例如：</p> <pre>os.space.opt_root= [dir:root=/opt/IBM/ITM] 1024MB</pre>
os.sshdConfig	UNIX	<p>檢查是否已針對 SSH 常駐程式階段作業配置允許的 root 登入</p>	<p>Available Unavailable</p>

表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.swapSize	UNIX	檢查交換空間是否必須大於 RAM 大小或交換空間總大小	值可以是下列其中任一類型： <ul style="list-style-type: none"> <li>布林值，例如： True</li> <li>數字格式單位為 MB 或 GB，例如：2GB</li> </ul>
os.tmpdir	UNIX	檢查指派給 /tmp 檔案系統的存取權限，包括存取權旗標設定的任何特定權限，例如八進位的 sticky、setuid 或 setgid 位元	代表存取權限之 octal_digits 八進位的數字  例如，若要檢查暫存目錄是否具有 drwxrwxrwt 權限，且已啓用組合位元權限： 1777  另一個範例為，若要檢查暫存目錄是否具有排除組合位元的 drwxrwxrwx 權限： 777
os.totalMemory	Windows	作業系統可存取的虛擬記憶體總量	數字格式單位為 MB 或 GB，例如：4GB
os.totalPhysicalMemory	Windows	作業系統可存取的實體記憶體總量，但它不會指出目標電腦上實體記憶體可用數量	數字格式單位為 MB 或 GB，例如：2030MB

表 24. 作業系統資料內容 (繼續)

必備項目內容	平台	說明	有效值
os.ulimit	UNIX	<p>檢查是否可基於下列資格屬性執行不限數目的處理程序：</p> <ul style="list-style-type: none"> <li>type 屬性，用於判定要檢查的其他限制，例如，filedescriptorlimit 會檢查處理程序可開啓的檔案描述子數目限制</li> </ul> <p>它還可以檢查在 /etc/security/limits.conf 檔案中是否已針對指定的網域設定下列限制：</p> <pre>root - stack unlimited ctginst1 - stack unlimited root - nofile 8192 tioadmin - nofile 32767</pre>	<p>值可以是下列其中任一類型：</p> <ul style="list-style-type: none"> <li>具有下列限定元格式的字串： [type:limit_name] limit_value, limited unlimited</li> </ul> <p>例如，若要檢查檔案描述子限制是否大於 8192 且具有數目不限制的處理程序：</p> <pre>os.ulimit=[type:filedescriptorlimit] 8192+, unlimited</pre> <p>要檢查的有效限制類型，其中 limit_name 代表下列限制類型：</p> <ul style="list-style-type: none"> <li>- ALL：檢查所有限制</li> <li>- corefilesizelimit</li> <li>- datasegmentlimit</li> <li>- filedescriptorlimit</li> <li>- filesizelimit</li> <li>- hardlimit</li> <li>- processlimit</li> <li>- maxmemorysizelimit</li> <li>- maxprocesseslimit</li> <li>- stacksizelimit</li> <li>- threadlimit</li> </ul> <ul style="list-style-type: none"> <li>Available Unavailable，用於指定相關網域在 /etc/security/limits.conf 檔案中是否具有限制集。</li> </ul>
os.umask	UNIX	檢查檔案模式建立遮罩的權限	代表存取權限之 octal_digits 八進位的數字。例如，若要檢查只有擁有者可寫入新檔案，請將八進位設為 0022
os.userLimits	UNIX	檢查堆疊大小上限是否無限制；如果無限制，則傳回 Available	Available Unavailable
os.versionNumber	Windows	檢查機器上已安裝之作業系統的現行版本	數字格式，例如 5.0+ 註：該值可使用在 第 2 頁的表 1 中概要的特殊字元。
os.windowManager	UNIX	檢查 GNOME 或 KDE 是否可用作圖形桌面	Available Unavailable

## 已安裝軟體資料內容

已安裝軟體資料內容會檢查已安裝軟體必備項目，如 Windows 登錄中註冊的程式以及是否已安裝 cygwin 與 gskit。僅對於 Windows 系統，它會使用 *ips\_root/lib* 目錄中的已安裝軟體收集器，且其檔名具有 *installedSoftware*, *cygwin*, or *gskit* 字首 ID。

表 25 說明共用資料必備項目內容。此種類的必備項目內容不需要字首 ID。

表 25. 已安裝軟體資料內容

必備項目內容	平台	說明	有效值
<i>installedSoftware</i>	Windows	掃描作業系統登錄以獲取已安裝程式及位置	以逗點分隔多個應用程式的字串。
<i>cygwinVersion</i>	Windows	檢查機器上已安裝的 cygwin 版本；如果未安裝任何版本，則傳回 0.0	正整數，例如 1.5 註：值可以使用第 2 頁的表 1 中所述的特殊字元。
<i>gskit7Version</i>	Windows	檢查機器上是否已安裝 gskit 第 7 版；如果未安裝第 7 版，則傳回 0.0	正整數，例如 7.0
<i>gskit8Version</i>	Windows	檢查機器上是否已安裝 gskit 第 8 版；如果未安裝第 8 版，則傳回 0.0	正整數，例如 8.0

## 使用者資料內容

使用者資料內容會檢查使用者必備項目，如已登入使用者是否具有管理權限或是否為 root 使用者。僅對於 Windows 系統，它會使用 *ips\_root/lib* 目錄中的使用者收集器，且其檔名具有 *user* 字首 ID。僅對於 UNIX 系統，它會使用 *ips\_root/lib/packageTest.sh* 中的使用者收集器。

表 26 說明使用者必備項目內容。此種類的必備項目內容需要 *user* 字首 ID。

表 26. 使用者資料內容

必備項目內容	平台	說明	有效值
<i>user.userID</i>	Windows	目前已登入使用者的 ID	字串，例如 smithj
<i>user.isAdmin</i>	所有	檢查已登入使用者是否為管理者群組成員	布林值，例如 True

## Windows 網路資料內容

Windows 網路資料內容會檢查網路必備項目，如機器上是否已啟用 NetBIOS 及 DHCP 以及 ping 內容。它會使用 *ips\_root/lib* 目錄中的 Windows 網路收集器，且其檔名中具有 *network* 字首 ID。

表 27 說明可用於所有 Windows 平台的網路必備項目內容。此種類的必備項目內容需要 *network* 字首 ID。

表 27. Windows 網路資料內容

必備項目內容	說明	有效值
<i>network.DHCPEnabled</i>	檢查是否至少一個具有有效 IP 位址的配接卡已使用 DHCP 取得該 IP 位址；如果存在至少一個，則傳回 True。	布林值，例如 False
<i>network.netBIOSEnabled</i>	檢查是否至少一個具有有效 IP 位址的配接卡已啟用 NetBIOS 作為通訊協定；如果存在至少一個，則傳回 True。	布林值，例如 True

表 27. Windows 網路資料內容 (繼續)

必備項目內容	說明	有效值
network.pingLocalhost	檢查本端主機是否回應 ping 通訊協定；如果回應，則傳回 True。	布林值，例如 True
network.pingSelf	檢查是否已使用 DHCP 解析本端電腦名稱並且可以 ping 該電腦；如果可以 ping，則傳回 True。	布林值，例如 True
network.ValidateHostsFile	檢查 C:\WINDOWS\system32\drivers\etc\hosts 中的項目是否為正確格式；如果格式正確，則傳回 True。	布林值，例如 True

## UNIX 網路資料內容

UNIX 網路資料內容會檢查網路必備項目，如機器上是否已啓用 NetBIOS 及 DHCP 以及 ping 內容。它會使用 *ips\_root/UNIX\_Linux* 目錄中的網路收集器。

表 28 說明可用於所有 UNIX 平台的網路必備項目內容。此類型的必備項目內容需要 network 字首 ID。

表 28. UNIX 網路資料內容

必備項目內容	說明	有效值
network.DHCPEnabled	檢查是否至少一個具有有效 IP 位址的配接卡已使用 DHCP 取得該 IP 位址	布林值，例如 False
network.dns	檢查主機的 DNS 登錄是否正確	布林值，例如 True
network.fqdn	檢查是否已設定主機的完整網域名稱	布林值，例如 True
network.pingLocalhost	檢查本端主機是否回應 ping 通訊協定	布林值，例如 True
network.pingSelf	檢查是否已使用 DHCP 解析本端電腦名稱並且可以 ping 該電腦	布林值，例如 True

## 環境變數資料內容

環境變數資料內容會檢查所有平台共用的環境變數必備項目，如是否設定環境變數或環境變數的值。僅對於 Windows 系統，它會使用 *ips\_root/lib* 目錄中的環境變數收集器，且其檔名具有 env 字首 ID。僅對於 UNIX 系統，它會使用 *ips\_root/UNIX\_Linux* 目錄中的 UNIX 環境變數收集器，其檔名具有 env 字首 ID。

第 101 頁的表 29 說明可用於所有平台的環境變數必備項目內容。此類型的必備項目內容需要 env 字首 ID。

表 29. 環境變數資料內容

必備項目內容	平台	說明	有效值
env.var.set. env_var_name	UNIX	<p>使用此命名慣例，以檢查指定的 <i>env_var_name</i> 環境變數是否已在電腦上設定，例如：</p> <ul style="list-style-type: none"> <li>env.var.set.HOME 會檢查是否已設定起始目錄的環境變數，其中 <i>env_var_name</i> 是 HOME 環境變數的名稱</li> <li>env.var.set.JAVA_HOME 會檢查是否已設定 Java 起始目錄的環境變數，其中 <i>env_var_name</i> 是 JAVA_HOME 環境變數的名稱</li> </ul>	布林值，例如 True
env.var.set. env_var_name [type:env_var_type]	Windows	<p>使用此命名慣例，以檢查是否已針對指定的 <i>env_var_name</i> 環境變數類型設定指定的 <i>env_var_name</i> 環境變數，例如：</p> <ul style="list-style-type: none"> <li>env.var.set.HOME 會檢查是否已設定起始目錄的環境變數，其中 <i>env_var_name</i> 是 HOME 環境變數的名稱</li> <li>env.var.set.JAVA_HOME[type:User] 會檢查是否針對登入的使用者設定 Java 起始目錄的環境變數，其中 <i>env_var_name</i> 是 JAVA_HOME 環境變數的名稱，而 <i>env_var_type</i> 是環境變數的 User 類型</li> </ul> <p><i>env_var_type</i> 環境變數類型是選用的，代表 Windows 作業系統支援的環境變數類型，如下所示：</p> <ul style="list-style-type: none"> <li>Process</li> <li>System</li> <li>User</li> <li>Volatile</li> </ul> <p>若未指定，則預設類型為 Process。</p>	布林值，例如 True
env.classpath.derbyJAR	所有	檢查 Derby JAR 檔的路徑是否包含在類別路徑環境變數中	布林值，例如 True
env.CIT.homeExists	Windows	檢查是否同時存在 HOMEDRIVE 和 HOMEPATH 環境變數	布林值，例如 True





## 附錄 D. UNIX 系統之預先定義的收集器

在 UNIX 系統上存在必備項目內容檢查的個別收集器，位於 *ips\_root/lib* 目錄中。您可以檢閱這些收集器及其輸入參數，然後建立自訂收集器。

表 30 說明 UNIX 系統之預先定義的收集器。

表 30. UNIX 收集器

收集器	適用於必備項目內容	輸入
DB2_Version	DB2 Version	無
DBType	DBType	無
DBTypeDetails	DBTypeDetails	無
env.classpath.derbyJAR	env.classpath.derbyJAR	無
env.var.set	env.var.set <i>env_var_name</i> 是要檢查的環境變數名稱	<i>\$env_var_name</i>
network.DHCPEnabled	network.DHCPEnabled	無
network.dns	network.dns	無
network.fqdn	network.fqdn	無
network.pingSelf	network.pingSelf	無
network.port	network.availablePorts.* network.portsInUse.*	<i>\$ports</i>
oracle.Client	oracle.Client	無
oracle.Client.Location	oracle.Client.Location	無
oracle.Server	oracle.Server	無
oracle.Server.Location	oracle.Server.Location	無
os.architecture	os.architecture	32 bit 64 bit
os.automount	os.automount	無
os.cmd	os.lookup	nslookup
os.cmd	os.tar os.gnu.tar	tar gtar
os.dir	os.dir. <i>dir_name</i> <i>dir_name</i> 可代表如下內容： • tmp • home	下列格式的字串： [dir: <i>dir_name</i> , type:permission] <i>octal_digits</i> +  例如，若要檢查 <i>dir_name</i> 目錄（即起始目錄）是否具有 drwxr-xr-x 權限： [dir:/home, type:permission]755+
os.diskquota	os.diskquota	無
os.expectLink	os.expectLink	無

表 30. UNIX 收集器 (繼續)

收集器	適用於必備項目內容	輸入
os.filepath	<p><i>os.file.script_name</i></p> <p><i>script_name</i> 可代表如下內容：</p> <ul style="list-style-type: none"> <li>• bash</li> <li>• expect</li> <li>• gzip</li> <li>• tar</li> </ul>	<p>Script 檔的路徑，其中，路徑可以是：</p> <ul style="list-style-type: none"> <li>• /bin/bash</li> <li>• /usr/bin/expect</li> <li>• /usr/bin/gzip</li> <li>• /usr/bin/tar</li> </ul>
os.Firefox	os.Firefox	無
os.FreePagingSpace	os.FreePagingSpace	無
os.ftpusers	os.ftpusers	無
os.hostformat	os.hostformat	無
os.iodevicestatus	os.iodevicestatus	無
os.kernelMode	os.kernelMode	無
os.kernelParameters	os.kernelParameters	無
os.kernelversion	os.kernelversion	無
os.largeFile	os.largeFile	無
os.ldLibPath	os.ldLibPath	產品
os.level	os.level	無
os.lib	<p><i>os.lib.lib_name_version</i></p> <p>以粗體表示、代表 <i>lib_name_version</i> 的字串或正規表示式，例如：</p> <ul style="list-style-type: none"> <li>• 32 位元 <b>libstdc++.so.#</b> 程式庫</li> <li>• 64 位元 <b>libstdc++.so.#</b> 程式庫</li> <li>• 32 位元 <b>libXft.so.#</b> 程式庫</li> <li>• 32 位元 <b>libXtst.so.#</b> 程式庫</li> <li>• 64 位元 <b>libaio.so.#</b> 程式庫</li> <li>• 32 位元 <b>x1C.rte</b> XLC 執行時期層次</li> <li>• 32 位元 <b>x1C.aix50.rte</b> XLC 執行時期 (適用於 AIX 5.3 版)</li> <li>• 32 位元 <b>x1C.aix61.rte</b> XLC 執行時期 (適用於 AIX 6.1 版)</li> <li>• AIX IOCP <b>bos.iocp.rte</b> 程式庫</li> <li>• <b>bos.loc.iso.en_us</b> : AIX 基本作業系統的 ISO 代碼檔案集</li> <li>•</li> </ul> <p><b>regex{str}</b> : 具有輸入參數 <i>str</i> 的正規表示式，代表程式庫名稱的搜尋型樣，例如，<b>.*libgcc.*</b></p>	<p><i>path_to_library</i> 或 <i>lib_name_version</i></p> <p>例如，若要檢查 <b>os.lib.libstdc++.so</b> 必備項目內容的實際值，輸入參數是 <b>/usr/lib/libstdc++.so.5</b> 及 <b>libstdc++.so</b>：  <b>os.lib /usr/lib/libstdc++.so.5 libstdc++.so</b></p> <p>例如，若要檢查 GCC 低層次執行時期程式庫版本 <b>libgcc</b> 是否存在於機器上，輸入參數是：  <b>regex{.*libgcc.*}</b></p>
os.loginVariable	os.loginVariable	無
os.maximoDirectory	os.maximoDirectory	無

表 30. UNIX 收集器 (繼續)

收集器	適用於必備項目內容	輸入
os.maximoDirOwner	os.maximoDirOwner	無
os.maximumProcesses	os.maximumProcesses	無
os.MozillaVersion	os.MozillaVersion	無
os.mountcheck	os.mountcheck	<p>具有下列限定元格式的字串：</p> <pre>[drive:dir_name, mount_option: false true] True False</pre> <p>例如，若要檢查是否已裝載 /home 目錄且未設定 nosuid 選項：</p> <pre>os.mountcheck=[drive:/home, nosuid:false]True</pre>
os.package	<p>os.package.package_name</p> <p>以粗體表示、代表 package_name 的字串，例如：</p> <ul style="list-style-type: none"> <li>• <b>bash</b> shell</li> <li>• <b>expect</b> (適用於 TCL 延伸套件)</li> <li>• <b>libgcc</b> (適用於 GCC 低層次執行時期套件)</li> <li>• <b>openssh</b> (適用於開放程式碼安全 Shell)</li> <li>• <b>openssl</b> (適用於 SSL/TLS 的開放程式碼工具箱)</li> <li>• <b>perl</b> (適用於 Perl Scripting 套件)</li> <li>• <b>rpm</b> (適用於 RPM 或 RPM Build 套件)</li> <li>• <b>telnet</b> (適用於 Telnet 套件)</li> <li>• <b>wget</b> (適用於 GNU 檔案擷取套件)</li> </ul>	<p>例如，package_name，其中 package_name 為 rpm：</p> <pre>os.package rpm</pre>
os.pagesize	os.pagesize	無
os.RAMSize	os.RAMSize	無
os.SELinux	os.SELinux	<ul style="list-style-type: none"> <li>• 具有下列限定元格式的字串：</li> </ul> <pre>[source:Command] Disabled Enabled</pre> <p>例如，若要檢查 Red Hat 或 SUSE 作業系統上是否已停用功能或功能是否具有權限狀態：</p> <pre>os.SELinux=[source: Command]Disabled</pre> <ul style="list-style-type: none"> <li>• 如果沒有限定元，則不會將值傳遞到收集器。</li> </ul>
os.servicePack	os.servicePack	服務套件值

表 30. UNIX 收集器 (繼續)

收集器	適用於必備項目內容	輸入
os.shell.default	os.shell.default	必備項目內容的期望值，例如 bash
os.space	<p>os.space.dir_name</p> <p>dir_name 可代表如下內容：</p> <ul style="list-style-type: none"> <li>• usr</li> <li>• home</li> <li>• tmp</li> <li>• var</li> </ul>	<p>具有下列限定元格式的字串（適用於 root 使用者的檔案系統）：</p> <pre>[dir:root=dir_path, unit:unit_name] disk_space</pre> <p>例如：</p> <pre>os.space.usr= [dir:root=/usr/ibm/common/acsi, unit:GB]200</pre> <p>具有下列限定元格式的字串（適用於非 root 使用者的檔案系統）：</p> <pre>[dir:non_root=dir_path, unit:unit_name] disk_space</pre> <p>例如：</p> <pre>os.space.home= [dir:non_root=USERHOME/ .acsi_HOST, unit:MB]200</pre> <p>具有下列限定元格式的字串（僅使用一個限定元）：</p> <pre>[dir:dir_path] disk_space MB</pre> <p>例如：</p> <pre>os.space.home= [dir:/home/sat]250MB</pre>
os.sshdConfig	os.sshdConfig	無
os.swapSize	os.swapSize	無
os.tmpdir	os.tmpdir	無

表 30. UNIX 收集器 (繼續)

收集器	適用於必備項目內容	輸入
os.ulimit	os.ulimit	<p>具有下列限定元格式的字串：</p> <pre>[type:limit_name] limit_value, limited unlimited</pre> <p>例如，若要檢查檔案描述子限制是否大於 8192 且具有數目不限的處理程序：</p> <pre>os.ulimit= [type:filedescriptorlimit] 8192+,unlimited</pre> <p>要檢查的有效限制類型，其中 <i>limit_name</i> 代表下列限制類型：</p> <ul style="list-style-type: none"> <li>• ALL：檢查所有限制</li> <li>• corefilesizelimit</li> <li>• datasegmentlimit</li> <li>• filedescriptorlimit</li> <li>• filesizelimit</li> <li>• hardlimit</li> <li>• processlimit</li> <li>• maxmemorysizelimit</li> <li>• maxprocesseslimit</li> <li>• stacksizelimit</li> <li>• threadlimit</li> </ul>
os.umask	os.umask	無
os.userLimits	os.userLimits	無
os.windowManager	os.windowManager	無



## 附錄 E. Windows 系統的共用函數

Prerequisite Scanner 在 /lib/common\_function.vbs 檔案具有一組共用函數，可在 Windows 系統上執行檢查。

表 31. common\_function.vbs 中的函數

函數	說明
第 110 頁的『allFiles()』	將指定目錄中的檔名讀取到陣列中。
第 110 頁的『arrayToString()』	建立陣列的字串表示法。
第 111 頁的『bigthan()』	計算必備項目內容之期望值與實際值之間的差異（如果該必備項目內容為大小，且單位為 MB 或 GB）。
第 111 頁的『changeMG()』	針對磁碟空間或記憶體必備項目內容，將輸入參數轉換為 MB 或 GB。
第 112 頁的『checkItemToString()』	建立 CheckItem 物件的字串表示法。
第 112 頁的『dictionaryToString()』	建立 Scripting 字典物件的字串表示法。
第 112 頁的『exeCommand()』	執行指定的指令，並傳回執行該指令所產生的結果。
第 113 頁的『filterCommand()』	執行指定的指令，並傳回指令結果中符合指定型樣的行。
第 113 頁的『filterFile()』	將檔案內容讀取並過濾到 Scripting 字典物件。
第 114 頁的『findNewest()』	尋找最新的配置檔。
第 114 頁的『findSuitableFile()』	尋找產品及版本的相關配置檔。
第 115 頁的『fmt()』	依照下列方式修改字串：將另一個字串之指定數量的字元新增至此字串，並使用空格字元填補另一字串（如果該字串太短）或截斷另一字串（如果該字串太長）。
第 116 頁的『formatForDisplay()』	格式化輸入參數，以讓其可讀。
第 116 頁的『formatSizeForDisplay()』	接收輸入參數並增添或縮減輸入參數的小數部分為兩位小數，例如，123 MB 增添成 123.00MB，或 12.123MB 縮減成 12.12MB。
第 116 頁的『getDecimalSeparator()』	判定用於現行語言環境的小數分隔字元。
第 117 頁的『getFirstMatch()』	取得陣列中搜尋字串的第一個相符項目。
第 117 頁的『isMatch()』	檢查搜尋型樣是否位於字串中。
第 118 頁的『notInLatter()』	過濾第一個陣列，以判定內容是否處於第二個陣列中。根據 in_or_not 輸入參數的值，函數會傳回第一個陣列的內容，其中包含或排除與第二個陣列相符的項目。
第 118 頁的『passOrFail()』	比較必備項目內容的期望值與實際值，並判定必備項目內容是否通過檢查。輸入參數可以是一般數字、大小（單位為 MB 或 GB）、CPU 速度（單位為 MHz 或 GHz）、布林或字串。

表 31. *common\_function.vbs* 中的函數 (繼續)

函數	說明
第 119 頁的『 <code>ppread()</code> 』	將檔案的內容讀取到 <code>Scripting</code> 字典物件中，如果行中存在指定的分隔字元，則使用該分隔字元進一步分割檔案的每一行。
第 119 頁的『 <code>readFile()</code> 』	將檔案的每一行讀取到陣列的索引項目。
第 120 頁的『 <code>unitMGTOG()</code> 』	彙總陣列的所有內容，以取得總計大小 (MB)。
第 120 頁的『 <code>varToString()</code> 』	建立變數的字串表示法。要檢查的變數可以是字串、數字、 <code>Scripting</code> 字典物件、陣列或 <code>CheckItem</code> 物件。

---

## **allFiles()**

將指定目錄中的檔名讀取到陣列中。

### **用途**

此函數會取得目錄輸入參數中的檔案清單，並將其新增至陣列。它會傳回陣列。

### **語法**

`allFiles(filepath)`

### **輸入參數**

字串 *filepath*

包含檔案之目錄的路徑。

### **回覆值**

陣列 *fileNames*

傳回包含指定目錄中檔名的陣列。

---

## **arrayToString()**

建立陣列的字串表示法。

### **用途**

此函數會將傳遞的陣列作為輸入參數，並傳回該陣列內容的字串表示法。

### **語法**

`arrayToString(arr)`

### **輸入參數**

陣列 *arr*

包含該陣列。

### **回覆值**

字串 *result*

傳回陣列的字串表示法，每個項目均以逗點區隔。



---

## bigthan()

計算必備項目內容之期望值與實際值之間的差異（如果該必備項目內容為大小，且單位為 MB 或 GB）。

### 用途

此函數首先呼叫『changeMG()』函數，以將必備項目內容的期望值與實際值變更為 MB（如有必要）。然後，它會檢查函數的傳回值是否為空值，如果任一值為空值，則函數的傳回值為 0MB 且函數會結束。它會檢查值的單位是 MB 還是 GB，並轉換為 MB（如有必要）。它會計算最終格式化值之間的差異並傳回結果。

### 語法

bigthan(expect,real)

### 輸入參數

字串 *expect*

必備項目內容的期望值。

字串 *real*

必備項目內容的實際值。

### 回覆值

字串 *bigthan*

傳回差異（單位為 MB）或 0MB（如果沒有差異）。

---

## changeMG()

格式化輸入參數，以從中移除任何額外的位數分組字元，並傳回格式化的參數（除非輸入參數包含 MB 或 GB）。如果包含，則它會將輸入參數分別轉換為 GB 或 MB。

### 用途

此函數會呼叫第 116 頁的『getDecimalSeparator()』函數，以判定現行語言環境的小數分隔字元，然後從數字輸入參數移除該語言環境的任何其他位數分組字元。然後，它會呼叫第 117 頁的『getFirstMatch()』函數，以判定值的單位是 MB 還是 GB，然後將值分別轉換為 GB 或 MB。

### 語法

changeMG(tochange)

### 輸入參數

字串 *tochange*

包含值格式並進行必要轉換。

### 回覆值

字串 *changeMG*

傳回沒有位數分組字元的格式化數字，或單位為 MB 或 GB 的數字。

---

## checkItemToString()

建立 CheckItem 物件的字串表示法。

### 用途

此函數會將傳遞的 CheckItem 物件作為輸入參數，並傳回字串表示法，其中包含 CheckItem 物件實例之不同內容的值。

### 語法

```
checkItemToString(var)
```

### 輸入參數

**CheckItem** *var*

包含 CheckItem 物件的實例。

### 回覆值

字串 *result*

傳回 CheckItem 物件之內容的字串表示法，如下所示：

```
result = "CheckItem[pdCode[" & chkItem.pdCode & "],pdName[" & chkItem.pdName & _  
        "],itype[" & chkItem.itype & "],recommended[" & chkItem.recommended & _  
        "],realValue[" & chkItem.realValue & "],passOrFail[" &  
        chkItem.passOrFail & "]"
```

---

## dictionaryToString()

建立 Scripting 字典物件的字串表示法。

### 用途

此函數會將傳遞的字典物件作為輸入參數，並傳回該字典物件之內容的字串表示法。

### 語法

```
dictionaryToString(dic)
```

### 輸入參數

字典 *dic*

包含該字典物件。

### 回覆值

字串 *result*

傳回字典物件的字串表示法，其中每個索引鍵及項目均以等號區隔。

---

## exeCommand()

執行指定的指令，並傳回執行該指令所產生的結果。

### 用途

此函數會執行指令輸入參數。如果發生任何錯誤，它會呼叫 logWarning 子常式，以顯示錯誤；否則，會傳回該指令執行的結果。

## 語法

`exeCommand(cmd)`

## 輸入參數

字串 *cmd*

要執行的指令名稱。

## 回覆值

字串 *result*

傳回包含該指令執行結果的字串。

---

## filterCommand()

執行指定的指令，並傳回指令結果中符合指定型樣的行。

## 用途

此函數會執行指令輸入參數。它會剖析指令執行的結果，並檢查結果的任何行是否符合行型樣輸入參數。如果存在相符項目，則會呼叫第 117 頁的『getFirstMatch()』函數，以判定資訊行輸入參數與指令結果之間是否也存在相符項目。如果存在相符項目，則它會使用 Join 函數，以從 getFirstMatch() 函數傳回字典物件的內容。

## 語法

`filterCommand(cmd, line_patt, after_line, info_patt)`

## 輸入參數

字串 *cmd*

要執行的指令名稱。

字串 *line\_patt*

要在指令執行結果中搜尋的行型樣。

數字 *after\_line*

資訊型樣搜尋停止之前搜尋的行數。

字串 *info\_patt*

要在指令結果中每一行搜尋的資訊型樣。

## 回覆值

字串 *filterCommand*

將字典物件的內容作為單一字串傳回。

---

## filterFile()

將檔案內容讀取並過濾到 Scripting 字典物件。

## 用途

此函數會讀取檔案的每一行，並將具有搜尋型樣的每一行傳遞到第 117 頁的『getFirstMatch()』函數。如果傳回相符項目且該行在字典物件中並不存在，則會將該行寫入字典物件中。函數會迴圈，直到到達檔案的結尾，然後傳回字典物件。

## 語法

```
filterFile(fileName, patt)
```

## 輸入參數

字串 *fileName*

要過濾的檔案。

字串 *patt*

要在檔案中每一行搜尋的型樣。

## 回覆值

字典 *dic.keys*

傳回具有檔案中已過濾行的 *dic* 字典物件。

---

## findNewest()

尋找陣列中最新的配置檔。

## 用途

此函數會在陣列中迴圈，並判定陣列中最新的配置檔。它會傳回檔案的名稱。

## 語法

```
findNewest(arr)
```

## 輸入參數

陣列 *arr*

包含要檢查的配置檔集。

## 回覆值

字串 *result*

傳回最新配置檔的名稱。

---

## findSuitableFile()

尋找產品及版本的相關配置檔。

## 用途

此函數會呼叫第 117 頁的『getFirstMatch()』函數，以從第 110 頁的『allFiles()』函數傳回的檔案清單中，取得其副檔名為副檔名輸入參數的一組檔案。然後，它會再次呼叫第 117 頁的『getFirstMatch()』函數，以傳回在檔名中包含產品型號輸入參數的一組檔案。它會呼叫同一函數，以取得檔名中包含版本輸入參數的一組檔案。如果函數找到與版本相符的一或多個檔案，它會呼叫『findNewest()』函數，以取得該檔案的最新版本並傳回該檔名；否則，它會傳回 `common.bat` 檔案或使用 `logScreen` 及 `logWarning` 子常式，然後再傳回該產品型號的最新配置檔版本。

## 語法

```
findSuitableFile(pd,version,suf,filepath)
```

## 輸入參數

### 字串 *pd*

與要尋找之檔案相關聯的產品型號，如產品型號檔案 *ips\_root/codename.cfg* 檔案中所指定。

### 字串 *version*

與要尋找之檔案相關聯的產品版本。*<version>* 是代表版本、版次、修正及層次的 8 位代碼，其中每兩位數表示代碼的每個部分，例如，7.3.21 為 07032100。

### 字串 *suf*

要尋找之檔案類型的副檔名，如 *cfg* 或 *bat*。

### 字串 *filepath*

包含要尋找之檔案的目錄路徑。

## 回覆值

### 字串 *findSuitableFile*

根據所呼叫之函數的結果，傳回下列其中一個檔名：

- *pd\_version.cfg*：相關聯產品型號與版本之檔案的最新版本。
- *common.bat*（如果副檔名輸入參數的值為 *bat*）。
- *pd.cfg*：產品一般配置檔的最新版本（如果找不到包含版本輸入參數的檔案）。

---

## fmt()

依照下列方式修改字串：將另一個字串之指定數量的字元新增至此字串，並使用空格字元填補另一字串（如果該字串太短）或截斷另一字串（如果該字串太長）。

## 用途

此函數會在類型字串的 *s* 輸入參數中，搜尋 *%#s* 表示式。*%#s* 表示式會判定 *args* 輸入參數中的指定字元數目，以新增至該表示式之位置的第一個字串。如果指定的數目大於 *args* 輸入參數的長度，會以空格字元填補。如果指定的數目小於 *args* 輸入參數的長度，會依據差異截斷長度。如果指定的數目為 0，則會將 *args* 輸入參數的完整長度新增至字串中適當位置的第一個字串。

## 語法

*fmt(s, args)*

## 輸入參數

### 字串 *s*

包含要依據該字串之 *%#s* 表示式中指定的字元數目修改的字串。

### 陣列 *args*

包含修改 *s* 輸入參數的一組字元。

## 回覆值

### 字串 *result*

傳回已修改的字串。

## 範例

```
fmt("Hello %5s!",array("Neo")) returns "Hello Neo !" padded with extra space characters
fmt("Hello %5s!",array("Mr. Anderson")) returns "Hello Mr. A!" truncated to add only "Mr. A"
fmt("Hello %0s!",array("Mr. Anderson")) returns "Hello Mr. Anderson!"
```

---

## formatForDisplay()

格式化輸入參數，以讓其可讀。

### 用途

此函數會呼叫『formatSizeForDisplay()』函數，以格式化輸入參數。

### 語法

```
formatForDisplay(val)
```

### 輸入參數

變數 *val*

要格式化的變數。

### 回覆值

字串 *varToString*

傳回所呼叫之『formatSizeForDisplay()』函數的結果。

---

## formatSizeForDisplay()

接收輸入參數並增添或縮減輸入參數的小數部分為兩位小數，例如，123MB 增添成 123.00MB，或 12.123MB 縮減成 12.12MB。

### 用途

此函數會計算輸入參數中的字元數目，檢查其是數字還是字串，並將輸入參數分為整數與小數部分。視小數部分而定，會增添或縮減成兩位小數位數。此函數會傳回結果。

### 語法

```
formatSizeForDisplay(size)
```

### 輸入參數

整數 *size*

值會捨入成兩位小數。

### 回覆值

整數 *val*

傳回捨入成兩位小數的值。

---

## getDecimalSeparator()

判定用於現行語言環境的小數分隔字元。

## 用途

此函數會建立小數並使用 `Mid()` 函數，以判定該小數中所使用的小數分隔字元。

## 語法

```
getDecimalSeparator()
```

## 輸入參數

無

## 回覆值

字元 *sep*

傳回語言環境的小數分隔字元，例如 `,` 或 `.`。

---

## getFirstMatch()

取得陣列中搜尋字串的第一個相符項目。

## 用途

此函數使用正規表示式，以在作為輸入參數傳遞的陣列中搜尋型樣，該型樣亦作為輸入參數傳遞。當它找到陣列中第一個相符的型樣，會將值從陣列新增至 `Scripting` 字典物件。

## 語法

```
getFirstMatch(patt, arr)
```

## 輸入參數

字串 *patt*

包含要搜尋的型樣。

陣列 *arr*

包含要在其中對搜尋型樣進行搜尋的陣列。

## 回覆值

字典 *keys*

傳回 `Scripting` 字典物件的索引鍵。

---

## isMatch()

檢查搜尋型樣是否位於字串中。

## 用途

此函數會呼叫『`getFirstMatch()`』函數，以剖析作為此函數輸入參數的型樣與字串（包含在陣列中）。它會呼叫 `ubound` 函數，以檢查 `getFirstMatch()` 函數的傳回值是否大於或等於 0。如果是，則為相符；否則，沒有相符的項目。

## 語法

```
isMatch(patt,str)
```

## 輸入參數

字串 *patt*

包含要搜尋的型樣。

字串 *str*

包含要在其中對搜尋型樣進行搜尋的字串。

## 回覆值

布林 *True|False*

如果存在相符項目，則會傳回 `True`；否則，會傳回 `False`。

---

## notInLatter()

過濾第一個陣列，以判定內容是否處於第二個陣列中。根據 `in_or_not` 輸入參數的值，函數會傳回第一個陣列的內容，其中包含或排除與第二個陣列相符的項目。

## 用途

## 語法

```
notInLatter(arr1, arr2, in_or_not)
```

## 輸入參數

陣列 *arr1*

從某個位置複製

陣列 *arr2*

到另外一個位置

字串 *in\_or\_out*

包含 `"in"` 或 `"not"`（取決於函數應傳回已過濾之第一個陣列的內容，以僅傳回符合第二個陣列的內容，即 `"in"`；還是傳回不符合第二個陣列的內容，即 `"not"`）。

## 回覆值

字典 *keys*

傳回 `Scripting` 字典物件的索引鍵，其中包含已過濾之第一個陣列，以僅具有符合第二個陣列的內容 (`in_or_not = "in"`) 或具有不符合第二個陣列的內容 (`in_or_not = "not"`)。

---

## passOrFail()

比較必備項目內容的期望值與實際值，並判定必備項目內容是否通過檢查。輸入參數可以是一般數字、大小（單位為 `MB` 或 `GB`）、CPU 速度（單位為 `MHz` 或 `GHz`）、布林或字串。

## 用途

此函數首先呼叫第 111 頁的『`changeMG()`』函數，以進行格式化，並視需要轉換期望值與實際值。它會檢查值是否為 0，如果是，則傳回 `"FAIL"` 並結束。如果值不為 0，函數會檢查值是布林、數值、大小（單位為 `MB` 或 `GB`）、CPU 速度（單位為 `MHz`（僅限 `Windows`）或 `GHz`）還是字串。然後，它會比較值並傳回結果。



## 語法

`passOrFail(expect,real)`

## 輸入參數

字串 *expect*

必備項目內容的期望值。

字串 *real*

必備項目內容的實際值。

## 回覆值

字串 *passOrFail*

根據期望值是等於還是大於實際值，傳回 "PASS" 或 "FAIL"。

---

## ppread()

將檔案的內容讀取到 **Scripting** 字典物件中，如果行中存在指定的分隔字元，則使用該分隔字元進一步分割檔案的每一行。

## 用途

此函數會讀取檔案的每一行，移除任何前導或尾端空格，並檢查其是否包含分隔字元。如果其包含分隔字元，則會依據分隔字元分割該行，並將每個部分作為項目新增至字典物件；否則，它會將縮減的行新增至字典物件中的項目。它會傳回包含字典物件作為第一個索引的陣列。

## 語法

`ppread(fileName, sep)`

## 輸入參數

字串 *fileName*

要讀取到字典物件之檔案的名稱。

字元 *sep*

代表分隔字元的字元，據以分割檔案中的行。

## 回覆值

陣列 *array(dic)*

傳回字典物件 (*dic*) 為其第一個索引的陣列。

## 範例

範例尚待提供。

---

## readFile()

將檔案的每一行讀取到陣列的索引項目。

## 用途

此函數會開啓檔案，並將檔案的每一行讀取到陣列的索引項目。它會傳回陣列。

## 語法

`readFile(fileName)`

## 輸入參數

字串 *fileName*

要讀取到陣列之檔案的名稱。

## 回覆值

陣列 *fileContents*

傳回具有檔案內容的陣列。

---

## unitMGTOG()

彙總陣列的所有內容，以取得總計大小 (MB)。

## 用途

此函數會將陣列中每個索引的值轉換為 MB 並進行彙總。

## 語法

`unitMGTOG(arr)`

## 輸入參數

陣列 *arr*

包含該陣列。

## 回覆值

字串 *unitMGTOG*

傳回陣列中內容的大小總計 (MB)，並將 "MB" 附加至該大小總計。

---

## varToString()

建立變數的字串表示法。要檢查的變數可以是字串、數字、Scripting 字典物件、陣列或 CheckItem 物件。

## 用途

此函數會檢查變數的資料或物件類型，並呼叫相關函數，以建立該資料或物件類型的字串表示法。

表 32. 針對每個變數類型呼叫的函數。

變數類型	呼叫的函數
Array	第 110 頁的『arrayToString()』
CheckItem object	第 112 頁的『checkItemToString()』
Scripting dictionary object	第 112 頁的『dictionaryToString()』

## 語法

`varToString(var)`

## 輸入參數

### 變數 *var*

支援的變數包括：字串、數字、Scripting 字典物件、陣列或 CheckItem 物件

## 回覆值

### 字串 *varToString*

傳回變數的字串表示法，其中包括必要時從呼叫之任何函數傳回的值。



## 附錄 F. Windows 系統的記載公用程式子常式

IBM Prerequisite Scanner 在 `req.vbs` 檔案中具有一組共用記載子常式，可在畫面上顯示訊息或寫入日誌檔。

表 33 說明記載公用程式。

表 33. 記載公用程式子常式

子常式	說明	輸入參數
<code>deleteLogFile</code>	如果日誌檔存在則加以刪除。	無
<code>log(level, msg)</code>	透過使用第 115 頁的『 <code>fmt()</code> 』函數，將訊息寫入日誌檔。日誌還包括現行日期和時間。	<ul style="list-style-type: none"><li><code>level</code> 是設定訊息類型（如資訊或警告）的字串</li><li><code>msg</code> 是代表要記載之訊息的字串</li></ul>
<code>logDebug(msg)</code>	呼叫 <code>log()</code> 函數，將 "DEBUG" 作為層次輸入參數傳遞。	<code>msg</code> 是代表要記載之訊息的字串
<code>logError(msg)</code>	呼叫 <code>log()</code> 函數，將 "ERROR" 作為層次輸入參數傳遞。	<code>msg</code> 是代表要記載之訊息的字串
<code>logInfo(msg)</code>	呼叫 <code>log()</code> 函數，將 "INFO" 作為層次輸入參數傳遞。	<code>msg</code> 是代表要記載之訊息的字串
<code>logScreen(msg)</code>	將訊息寫入畫面。	<code>msg</code> 是代表要寫入畫面之訊息的字串
<code>logScreenWithReplacement(msg, replaceStr)</code>	將訊息寫入畫面，將訊息碼與字串作為輸入參數傳遞。	<ul style="list-style-type: none"><li><code>msg</code> 是代表要寫入畫面之訊息字串的訊息碼</li><li><code>replaceStr</code> 是在訊息碼值中取代 <code>%variable</code> 的字串</li></ul>
<code>logScreenWithMultiReplacements(msg, replaceStrArray)</code>	將訊息寫入畫面，將訊息碼與字串陣列作為輸入參數傳遞。	<ul style="list-style-type: none"><li><code>msg</code> 是代表要寫入畫面之訊息字串的訊息碼</li><li><code>replaceStrArray</code> 是字串陣列，其陣列中的每一個索引均會取代訊息碼值中的 <code>%variable</code></li></ul>
<code>logWarning(msg)</code>	呼叫 <code>log()</code> 函數，將 "WARNING" 作為層次輸入參數傳遞。	<code>msg</code> 是代表要記載之訊息的字串



## 附錄 G. Windows 系統的檔案公用程式子常式

Prerequisite Scanner 在 /lib/common\_function.vbs 檔案中有一組共用檔案子常式，可用於處理檔案。它還有一組可處理檔案的函數。

表 34 說明檔案公用程式。

表 34. 檔案公用程式子常式

子常式	說明	輸入參數
appendToFile(text, fileName)	將文字附加至指定檔案的結尾。	<ul style="list-style-type: none"><li>text 是包含要附加至檔案之文字的字串</li><li>filename 是代表要修改之檔案名稱的字串</li></ul>
writeToFile(text, fileName)	將文字寫入指定的檔案，改寫現有內容（必要的話）。	<ul style="list-style-type: none"><li>text 是包含要寫入檔案之文字的字串</li><li>filename 是代表要修改之檔案名稱的字串</li></ul>

表 35 說明處理檔案的檔案函數。

表 35. 檔案公用程式函數

函數	說明
第 110 頁的『allFiles()』	將指定目錄中的檔名讀取到陣列中。
第 113 頁的『filterFile()』	將檔案內容讀取並過濾到 Scripting 字典物件。
第 114 頁的『findNewest()』	尋找最新的配置檔。
第 114 頁的『findSuitableFile()』	尋找產品及版本的相關配置檔。
第 119 頁的『ppread()』	將檔案的內容讀取到 Scripting 字典物件中，如果行中存在指定的分隔字元，則使用該分隔字元進一步分割檔案的每一行。
第 119 頁的『readFile()』	將檔案的每一行讀取到陣列的索引項目。





---

## 附錄 H. Windows 系統的其他共用函數及子常式

Prerequisite Scanner 具有在多種檔案中使用的一組其他共用函數及子常式。

說明其他共用函數及子常式。

表 36. Windows 系統的其他共用函數及子常式

函數或子常式	說明
『ffirstMatch()』	取得陣列中搜尋字串的第一個相符項目。
第 128 頁的 『getValue()』	取得指定目錄的可用磁碟空間。
第 128 頁的 『removeSpecialCharacters()』	移除商標或其他特殊字元，以更易於比較。
第 129 頁的 『versionCompare()』	剖析代表必備項目內容之實際值與期望值的輸入參數，並進行比較以判定必備項目內容是否通過必備項目檢查。函數預期以點分隔的版本字串作為輸入參數，例如，1.0.0.4、2.3、3.40.26.7800 或 2.3.*。

---

### ffirstMatch()

取得陣列中搜尋字串的第一個相符項目。

#### 用途

此函數使用正規表示式，以在作為輸入參數傳遞的陣列中搜尋型樣，該型樣亦作為輸入參數傳遞。當它找到陣列中第一個相符的型樣，會將值從陣列新增至 Scripting 字典物件。

#### 母項函數

表 37. 呼叫 ffirstMatch() 的母項函數

母項函數、Script	說明
ud620db2level(expect, real) (位於 DB2_Version_compare.vbs 中)	比較 DB2 版本必備項目內容的實際值與期望值。
oslevelcompare(expect, real) (位於 OS_Version_compare.vbs 中)	比較 OS 版本必備項目內容的實際值與期望值。

#### 語法

ffirstmatch(patt,arr)

#### 輸入參數

字串 *patt*

包含要搜尋的型樣。

陣列 *arr*

包含要在其中對搜尋型樣進行搜尋的陣列。

## 回覆值

### 字典 *keys*

傳回 Scripting 字典物件的索引鍵。

---

## getValue()

取得指定目錄的可用磁碟空間。

### 用途

此子常式使用檔案系統物件的實例，以呼叫路徑輸入參數的 `getDriveName()` 函數，然後使用 `freeSpace` 內容，以取得可用磁碟空間（將隨後會轉換為 MB）。必備項目內容輸入參數及其值會寫入與 Script 檔相關聯的暫時文字檔。

### Script

表 38. 使用 `getValue()` 的 Script

Script	說明
DEZ_01040000.vbs	此 Script 會收集必備項目內容，並讓其僅可用於 DEZ 01040000 配置檔
LCM_TAD_common.vbs	此 Script 會收集必備項目內容，並讓其僅可用於 LCM 02300000 及 TAD 07200000 配置檔
TAD722_impl.vbs	此 Script 會收集必備項目內容，並讓其僅可用於 TAD 07220000 配置檔

### 語法

```
getValue fso, sKey, drvPath
```

### 輸入參數

#### 檔案系統物件 *fso*

檔案系統物件的實例。

#### 字串 *sKey*

包含具有必備項目內容之名稱及等號的字串。

#### 字串 *drvPath*

包含從中取得可用磁碟空間的路徑。

### 回覆值

無

---

## removeSpecialCharacters()

移除商標或其他特殊字元，以更易於比較。該函數位於 `/lib/common.vbs` 檔案中。

### 用途

此函數呼叫 `Replace()` 函數，以使用 "" 取代商標、著作權及註冊商標符號。

## 語法

removeSpecialCharacters(s)

## 輸入參數

字串 *s*

包含必須從中移除字元的字串

## 回覆值

字串 *s*

傳回沒有特殊字元的字串。

---

## versionCompare()

剖析代表必備項目內容之實際值與期望值的輸入參數，並進行比較以判定必備項目內容是否通過必備項目檢查。函數預期以點分隔的版本字串作為輸入參數，例如，1.0.0.4、2.3、3.40.26.7800 或 2.3.\*。

## 用途

此函數首先處理一個或兩個輸入參數為空白的特殊情況，然後傳回回覆碼以代表此類情況。它會將每個版本分割成以點分隔字元分隔的數個部分。如果版本的最後部分為 \* 萬用字元，則函數會將版本的所有遺漏部分視為萬用字元，例如，2.\* 與 2.1 或 2.3.\* 相符。然後，它會在每個版本的部分清單中迴圈並進行比較。根據期望值是小於、等於還是大於實際值，它會傳回回覆碼。

## 母項函數

表 39. 呼叫 *versionCompare* 的母項函數

母項函數、Script	說明
cygwinVersion_compare.vbs	比較 cygwin 版本必備項目內容的實際值與期望值。
gskit7Version_compare.vbs	比較 gskit 第 7 版必備項目內容的實際值與期望值。
gkit8Version_compare.vbs	比較 gskit 第 8 版必備項目內容的實際值與期望值。
internetExplorer.version_compare.vbs	比較 Internet Explorer 版本必備項目內容的實際值與期望值。
os.servicePack_compare.vbs	比較 OS 服務套件必備項目內容的實際值與期望值。
os.versionNumber_compare.vbs	比較 OS 版本必備項目內容的實際值與期望值。

## 語法

versionCompare(ver1,ver2)

## 輸入參數

字串 *ver1*

包含必備項目內容的期望版本。

字串 *ver2*

包含必備項目內容的實際版本。

## 回覆值

### 整數 0

如果兩個輸入參數相等，則傳回 0 回覆碼。母項函數會傳回 "PASS"。

特殊情況：如果兩個輸入函數均為空白，則傳回 0 回覆碼並結束。

### 整數 -1

如果第一個輸入參數小於第二個輸入參數，則傳回 -1 回覆碼。母項函數會傳回 "FAIL"。

特殊情況：如果第一個輸入參數為空白，則傳回 -1 回覆碼並結束。

### 整數 1

如果第一個輸入參數大於第二個輸入參數，則傳回 1 回覆碼。母項函數會傳回 "PASS"。

特殊情況：如果第二個輸入參數為空白，則傳回 1 回覆碼並結束。

## 附錄 I. UNIX 系統的共用函數

Prerequisite Scanner 在 `/lib/common_function.sh` 檔案中有一組共用函數，可在 UNIX 型系統上執行檢查。

表 40. `common_function.sh` 中的函數

函數	說明
第 132 頁的『AddMG()』	檢查輸入參數單位是 MB 還是 GB 並新增參數。
『changeMG()』	針對磁碟空間或記憶體必備項目內容，將輸入參數轉換為 MB 或 GB。
第 132 頁的『compare()』	剖析代表必備項目內容之實際值與期望值的輸入參數，並進行比較以判定第一個值（實際）是否小於第二個值（預期）。
第 133 頁的『cutdown()』	剖析代表必備項目內容之實際值與期望值的輸入參數，並進行比較以判定第一個值（實際）是否小於第二個值（預期）。然後，如果第一個值不小於第二個值，則列印兩個值之間的差異。
第 135 頁的『findOSInfo()』	尋找系統的作業系統版本、OS 版次層次及版本，以及硬體實作資料。
第 133 頁的『mes4path()』	尋找每個已裝載檔案系統的可用磁碟空間。
第 134 頁的『mes4Path1()』	僅尋找 Solaris 系統上每個已裝載檔案系統的可用磁碟空間。
第 136 頁的『NFScheck()』	檢查 UNIX 型系統上裝載的 NFS 狀態。
第 135 頁的『telnetNFS()』	檢查是否可在預設埠 2049 上，Telnet 到已裝載之檔案系統的 IP 位址。

### changeMG()

針對磁碟空間或記憶體必備項目內容，將輸入參數轉換為 MB 或 GB。

#### 用途

此函數首先會檢查函數是否收到輸入參數。如果收到輸入參數，則會判定值的單位是 MB 還是 GB，然後將值分別轉換為 GB 或 MB。

#### 語法

```
changeMG val
```

#### 輸入參數

字串 `$val`

包含磁碟空間或記憶體的值（單位為 MB 或 GB）。

## 回覆值

### 整數 1

如果函數未收到輸入參數，則傳回 1。

字串 `printf "%.0fM%s",mm[1]*1024,mm[2];`

傳回單位為 MB 的值。

字串 `printf "%.2fG%s",mm[1],mm[2];`

傳回單位為 GB 的值。

---

## AddMG()

檢查輸入參數單位是 MB 還是 GB 並新增參數。

## 用途

此函數首先檢查函數是否收到輸入參數。如果收到輸入參數，則會判定值的單位是 MB 還是 GB，然後新增值。

## 語法

AddMG val1 val2

## 輸入參數

字串 *val1*

包含要新增至其他輸入參數之磁碟空間或記憶體的值（單位為 MB 或 GB）。

字串 *val2*

包含要新增至其他輸入參數之磁碟空間或記憶體的值（單位為 MB 或 GB）。

## 回覆值

### 整數 1

如果函數未收到兩個輸入參數，則傳回 1。

字串 *val*

傳回新增的值（單位為 MB 或 GB）。

---

## compare()

剖析代表必備項目內容之實際值與期望值的輸入參數，並進行比較以判定第一個值（實際）是否小於第二個值（預期）。

## 用途

此函數首先會檢查函數是否收到兩個輸入參數。如果收到兩個輸入參數且均不是 false，則它會判定值的單位是 MB 還是 GB，然後比較這兩個值，以檢查第一個值是否小於第二個值。如果小於，則會傳回 false 值；否則，會傳回 pass 值。

## 語法

compare real expected

## 輸入參數

字串 *\$real*

包含必備項目內容的實際值。

字串 *\$expected*

包含必備項目內容的期望值。

## 回覆值

整數 *1*

如果函數未收到兩個輸入參數，則傳回 1。

字串 "FAIL|PASS"

如果實際值小於期望值，則傳回字串 "FAIL"；否則，傳回字串 "PASS"。

---

## cutdown()

剖析代表必備項目內容之實際值與期望值的輸入參數，並進行比較以判定第一個值（實際）是否小於第二個值（預期）。然後，如果第一個值不小於第二個值，則列印兩個值之間的差異。

## 用途

此函數首先會檢查函數是否收到兩個輸入參數。如果收到兩個輸入參數，它會判定值的單位是 MB 還是 GB，然後將其轉換為 MB（如果單位是 GB）。然後，它會比較這兩個值，以檢查第一個值是否小於第二個值。如果小於，則會傳回 "OMB" 值；否則，傳回兩個值之間的差異（單位為 MB）。

## 語法

```
cutdown real expected
```

## 輸入參數

字串 *\$real*

包含必備項目內容的實際值。

字串 *\$expected*

包含必備項目內容的期望值。

## 回覆值

整數 *1*

如果函數未收到兩個輸入參數，則傳回 1。

字串 "FAIL|PASS"

如果實際值小於期望值（兩個值的單位既不是 MB 也不是 GB），則傳回字串 "FAIL"，否則，傳回字串 "PASS"。

字串 "OMB|Real-ExpectedMB"

如果實際值小於期望值，則傳回字串 "OMB"；否則，傳回兩個轉換值（單位為 MB）之間差異的字串表示法。

---

## mes4path()

尋找每個裝載的檔案系統的可用磁碟空間。

## 用途

此函數會將路徑用作輸入，呼叫 **uname** 指令以判定作業系統，然後呼叫 **NFScheck** 函數以判定系統是否已啟動及其裝載。然後，它會呼叫 **df** 指令，以判定系統上每個裝載的可用磁碟空間。它會傳回可用磁碟空間的值。

## 語法

```
mes4Path path
```

## 輸入參數

字串 *\$path*

系統的路徑，以檢查可用磁碟空間。

## 回覆值

整數 1

如果函數未收到輸入參數，則傳回回覆碼 1。

整數 2

如果輸入參數不是路徑，則傳回回覆碼 2。

字串 *\$NF*

傳回每個裝載的可用磁碟空間。

字串 *"\$path Server NotAvailable Responding for \$path"*

傳回訊息以表明該路徑的伺服器無法使用。

---

## mes4Path1()

僅尋找 Solaris 系統上每個已裝載檔案系統的可用磁碟空間。

## 用途

此函數會將路徑作為輸入，呼叫 **uname** 指令以判定作業系統是否為 Solaris。然後，它會呼叫 **df** 指令，以判定系統上每個裝載的可用磁碟空間。它會傳回可用磁碟空間的值。

## 語法

```
mes4Path1 path
```

## 輸入參數

字串 *\$path*

系統的路徑，以檢查可用磁碟空間。

## 回覆值

整數 1

如果函數未收到輸入參數，則傳回回覆碼 1。

整數 2

如果輸入參數不是路徑，則傳回回覆碼 2。

字串 *\$NF*

傳回每個裝載的可用磁碟空間。



---

## findOSInfo()

尋找系統的作業系統版本、OS 版次層次及版本，以及硬體實作資料。

### 用途

此函數執行 `uname` 指令，並針對系統的作業系統版本、OS 版次層次及版本，以及硬體實作資料，剖析其輸出。

### 語法

```
findOSInfo
```

### 輸入參數

無

### 回覆值

字串 *\$oo*

`uname` 的輸出，不含基本系統資訊。

字串 *\$kk*

作業系統版本

字串 *\$hh*

硬體實作，I 代表 i386 硬體、Z 代表 s390 硬體。

字串 *\$rr*

作業系統版次層次

字串 *\$vv*

作業系統版次層次版本

---

## telnetNFS()

檢查是否可在預設埠 2049 上，Telnet 到已裝載之檔案系統的 IP 位址。

### 用途

此函數將 IP 作為輸入，並呼叫 `telnet` 指令，以測試預設 Telnet 埠 2049 上的遠端連線是否成功。它會嘗試遠端連線 10 次。如果 `telnet` 指令失敗，則函數會傳回 "FALSE" 值；否則，會 "PASS" 值。

### 語法

```
telnetNFS ipaddr
```

### 輸入參數

字串 *\$ipaddr*

用於檢查是否可執行 Telnet 的 IP 位址。

### 回覆值

字串 "FALSE|TRUE"

傳回 Telnet 檢查的結果。如果檢查成功，則會傳回 "TRUE"；否則，會傳回 "FALSE"。

---

## NFScheck()

檢查 UNIX 型系統上裝載的 NFS 狀態。

### 用途

此函數會將路徑作為輸入，並呼叫 `mount` 指令以取得已裝載檔案系統的清單。它會呼叫 `uname` 指令，以判定作業系統。然後，它會呼叫 `ping` 指令以 `ping` 每個已裝載的系統，如果可以 `ping`，則它會呼叫 `telnetNFS` 函數，以檢查是否可以執行遠端連線。如果 `ping` 或 `telnet` 動作失敗，則函數會傳回 "FALSE" 值；否則，會傳回 "PASS" 值。

### 語法

`NFScheck path`

### 輸入參數

字串 *\$path*

將目錄的有效路徑作為輸入。

### 回覆值

布林值 *TRUE* 或 *FALSE*

如果 NFS 檢查成功（即如果針對每個檔案系統，可以順利 `ping` 相關聯的 IP 位址，或者可以使用 Telnet 連接到相關聯的 IP 位址），則會傳回 `TRUE`；否則，會傳回 `FALSE`。

### 範例

其用法的範例來自 `mes4Path()` 函數：

```
# check if it's a path
path=`echo "$1" | sed -n '/^\//p`
if [ -z "$path" ];then
    return 2;
else
    nfs_check_status=`NFScheck $path`
    if [ "$nfs_check_status" = "TRUE" ]; then
    case `uname` in
    ...
```

## 附錄 J. UNIX 系統的其他函數

Prerequisite Scanner 在多種檔案中具有一組共用函數。

表 41 說明多個檔案中的函數組。

表 41. 多個檔案中的共用函數

函數	說明
『formatSizeDisplay()』	接收輸入參數並增添或縮減輸入參數的小數部分為兩位小數，例如，123 MB 增添成 123.00MB，或 12.123MB 縮減成 12.12MB。
第 138 頁的 『versionCompare()』	剖析代表必備項目內容之實際值與期望值的輸入參數，並比較版本的每個部分，以判定必備項目內容是否通過必備項目檢查。

表 42 說明 UNIX-Linux/TAD722\_impl.sh 檔案中的函數組，以用於執行 Tivoli License Compliance Manager 及 Tivoli Asset Discovery for Distributed 檢查。

表 42. TAD722\_impl.sh 中的共用函數

函數	說明
第 140 頁的 『checkSunOS()』	檢查 Solaris 作業系統的版本是否適合於 SPARC 或 X86 平台。
第 139 頁的 『checkHpx()』	檢查 HP-UX 作業系統的版本是否適合於 IA64 或 PARISC 平台。
第 139 頁的 『checkLinux()』	檢查 Linux 作業系統的版本是否適合於 System p®、System z 或 x86 平台。
第 141 頁的 『getSystemId()』	呼叫不同的 OS 函數，以檢查相關作業系統的平台。
第 140 頁的 『getValue()』	取得指定檔案中索引鍵的值（如果該索引鍵存在的話）。
第 141 頁的 『setValue()』	設定指定檔案中索引鍵的值（如果必備項目內容存在的話）。
第 141 頁的 『copyValue()』	根據產品及作業系統，取得及設定必備項目內容（索引鍵）的值。
第 142 頁的 『parseDirParameter()』	針對掃描器的 -p 旗標，剖析參數清單中的參數，並將其值置於清單中。
第 142 頁的 『getClosestExistingParentDir()』	取得最近的上層目錄或目錄本身。
第 142 頁的 『printDirSize()』	檢查已裝載之檔案系統的 NFS 狀態，然後取得檔案系統或其上層目錄的磁碟空間

### formatSizeDisplay()

接收輸入參數並增添或縮減輸入參數的小數部分為兩位小數點，例如，123 MB 增添成 123.00 MB，12.123 MB 縮減成 12.12 MB

## 用途

此函數會對輸入參數中的字元數進行計數，檢查其是數字還是字串，並將輸入部分分為整數與小數部分。視小數部分而定，會增添或縮減成兩位小數位數。此函數會傳回結果。

## 母項 Script

下列 Script 包含函數：

- ./Unix-Linux/common.sh
- LCM\_TAD\_common.sh

## 語法

```
formatSizeDisplay val
```

## 輸入參數

整數 *val*

值會四捨五入成兩位小數點。

## 回覆值

整數 *val*

傳回四捨五入成兩位小數點的值。

---

## versionCompare()

剖析代表必備項目內容之實際與期望值的輸入參數，並比較版本的每個部分以判定第一個值（實際）是否大於第二個值（預期）。

## 用途

此函數首先會檢查函數接收到兩個版本作為輸入參數。它會使用 `awk` 來剖析每個版本並分割成各個部分，其中 "." 是將值分割成部分的定界字元。然後，它會執行迴圈以比較第一個版本的每個部分與第二個版本的相應部分，並檢查其是否相等。

## 母項函數

表 43. 呼叫 `versionCompare` 的母項函數

母項函數、Script	說明
db2.home_compare.sh	比較 DB2 HOME 必備項目內容之磁碟空間的實際與期望值。
oracle.Client_compare.sh	比較 Oracle 用戶端必備項目內容的實際與期望值。
os.locale_compare.sh	比較 OS 語言環境必備項目內容的實際與期望值。
os.MozillaVersion_compare.sh	比較 Mozilla Firefox 必備項目內容的實際與期望值。
os.package.perl_compare.sh	比較 Perl 套件必備項目內容的實際與期望值。會呼叫其本身。
os.RAMSize_compare.sh	比較 RAM 必備項目內容的實際與期望值。

表 43. 呼叫 `versionCompare` 的母項函數 (繼續)

母項函數、Script	說明
<code>os.space_compare.sh</code>	比較可用磁碟空間必備項目內容的實際與期望值。
<code>OS_Version_compare.sh</code>	比較 OS 版本必備項目內容的實際與期望值。

## 語法

```
versionCompare real expected
```

## 輸入參數

字串 *\$real*

包含必備項目內容的實際值。

字串 *\$expected*

包含必備項目內容的期望值。

## 回覆值

整數 *0*

如果實際與期望值相等，則傳回 0 回覆碼。母項函數傳回 "PASS"。

特殊情況：如果函數收到空白輸入參數，會傳回 0 回覆碼並結束。

整數 *-1*

如果實際值小於期望值，則傳回 -1 回覆碼。母項函數傳回 "FAIL"。

如果函數收到的第二個輸入參數為空白，會傳回 -1 回覆碼並結束。

整數 *1*

如果實際值大於期望值，則傳回 1 回覆碼。母項函數傳回 "PASS"。

如果函數收到的第一個輸入參數為空白，會傳回 1 回覆碼並結束。

## checkHpx()

檢查 HP-UX 作業系統的版本是否適合於 IA64 或 PARISC 平台。

## 用途

此函數使用 `uname` 指令的 `-m` 旗標，以判定 HP-UX 作業系統是否適合於 IA64 或 PARISC 平台。

## 語法

```
checkHpx
```

## 回覆值

字串 `HPUXIA64|HPUXPARISC`

如果 `-m` 旗標為 "ia64"，則會傳回 "HPUXIA64"；否則，會傳回 "HPUXPARISC"。

## checkLinux()

檢查 Linux 作業系統的版本是否適合於 System p、System z 或 x86 平台。

## 用途

此函數使用 `uname` 指令的 `-m` 旗標，以判定 Linux 作業系統是否適合於 System p、System z 或 x86 平台。

## 語法

```
checkLinux
```

## 輸入參數

## 回覆值

字串 `LINUXPSERIES|LINUXZSERIES|LINUXX86`

如果 `-m` 旗標為 "ppc64" 或 "ppc"，則會傳回 "LINUXPSERIES"。如果值為 "s390x" 或 "s390"，則會傳回 "LINUXZSERIES"；否則，會傳回 "LINUXX86"。

---

## checkSunOS()

檢查 Solaris 作業系統的版本是否適合於 SPARC 或 X86 平台。

## 用途

此函數使用 `uname` 指令的 `-p` 旗標，以判定 Solaris 作業系統是否適合於 SPARC 或 X86 平台。

## 語法

```
checkSunOS
```

## 輸入參數

## 回覆值

字串 `SOLARISSPARC|SOLARISX86`

如果 `-p` 旗標為 "sparc"，則會傳回 "SOLARISSPARC"；否則，會傳回 "SOLARISX86"。

---

## getValue()

取得指定檔案中索引鍵的值（如果該索引鍵存在的話）。

## 用途

## 語法

```
getValue key file
```

## 輸入參數

字串 `$key`

包含要設定的索引鍵。

字串 `$file`

包含其中包含索引鍵之檔案的名稱。

---

## setValue()

設定指定檔案中索引鍵的值（如果必備項目內容存在的話）。

### 語法

```
setValue key value file
```

### 輸入參數

字串 *\$key*

包含要設定的必備項目內容。

字串 *\$value*

包含必備項目內容的值。

字串 *\$file*

包含其中包含必備項目內容之檔案的名稱。

---

## copyValue()

根據產品及作業系統，取得及設定必備項目內容（索引鍵）的值。

### 用途

此函數會呼叫 **getValue()** 函數，以取得產品及作業系統之指定的必備項目內容的值。然後，它會呼叫 **setValue()** 函數，以設定 Prerequisite Scanner 檔案中必備項目內容的值。

### 語法

```
copyValue key file
```

### 輸入參數

字串 *\$key*

包含要取得及設定的索引鍵。

字串 *\$file*

包含其中包含索引鍵之檔案的名稱。

### 回覆值

---

## getSystemId()

呼叫不同的 OS 函數，以檢查相關作業系統的平台。

### 用途

此函數會呼叫各種 OS 函數，以判定相關作業系統的平台。

### 語法

```
getSystemId
```

### 輸入參數

## 回覆值

字串 *AIX|Linux*

如果產品為 Tivoli License Compliance Manager 且 OS 是 AIX 或 Linux，則會傳回 "AIX" 或 "Linux"；如果產品為 Tivoli Asset Discovery for Distributed 且 OS 為 AIX，則會傳回 "AIX"。

---

## getClosestExistingParentDir()

取得最近的上層目錄或目錄本身。

### 用途

### 語法

`getClosestExistingParentDir dirpath`

### 輸入參數

字串 *\$dirpath*

包含可取得其上層目錄或目錄本身的路徑。

### 回覆值

字串 *dirpath*

傳回上層目錄或目錄本身

---

## parseDirParameter()

針對掃描器的 `-p` 旗標，剖析參數清單中的參數，並將其值置於清單中。

### 用途

### 語法

### 輸入參數

字串

### 回覆值

---

## printDirSize()

檢查已裝載之檔案系統的 NFS 狀態，然後取得檔案系統或其上層目錄的磁碟空間。

### 用途

此函數首先呼叫 `NFScheck` 函數，以判定目錄的 NFS 狀態。如果狀態為 `true`，它會呼叫 `getClosestExistingParentDir` 函數，以傳回目錄或其上層目錄，然後使用 `df` 指令，以取得可用磁碟空間量。最後，它會呼叫 `formatSizeDisplay` 函數，以將值捨入到小數。

### 語法

`printDirSize dirpath`



## 輸入參數

字串 *\$dirpath*

包含要從中取得可用磁碟空間之目錄的路徑。

## 回覆值

整數 *dsize*

傳回具有兩位小數的可用磁碟空間量。

字串 *"NFS\_NOT\_AVAILABLE"*

傳回已裝載的檔案系統不可用。



## 附錄 K. UNIX 系統的記載公用程式函數

Prerequisite Scanner 在 `/lib/common_function.sh` 檔案中具有一組共用記載函數，可將除錯及追蹤資料寫入日誌檔。

表 44 說明記載公用程式。

表 44. UNIX 系統上的記載公用程式函數

函數	說明	輸入參數
<code>wrlTrace log_str1 log_str2</code>	將 <code>log_str1</code> 及 <code>log_str2</code> 字串寫入追蹤檔案且具有時間戳記	<code>log_str1</code> 及 <code>log_str2</code> 是追蹤字串，代表要執行的動作與收集器，以及要在追蹤檔案中記載的內容。例如：  <pre> <b>wrlTrace Starting os.lib</b> <b>wrlTrace Executing os.lib</b> wrlDebug Starting os.lib wrlDebug Expected libXp ss=./os.lib libXp libXp <b>wrlTrace Finished os.lib</b> echo "os.lib.libXp=\$ss" wrlDebug Finished os.lib wrlDebug OutPutValueIs \$ss <b>wrlTrace Done os.lib</b> </pre>
<code>wrlTraceFuncStart fn_name</code>	將 <code>fn_name</code> 函數傳遞到 <code>wrlTrace()</code>	<code>fn_name</code> 是追蹤字串，代表剛剛呼叫的函數。例如：  <pre> wrlTraceFuncStart "\$1" </pre>
<code>wrlTraceFuncExit fn_name</code>	將 <code>fn_name</code> 函數傳遞到 <code>wrlTrace()</code>	<code>fn_name</code> 是追蹤字串，代表剛剛完成的函數。例如：  <pre> wrlTraceFuncExit "\$1" </pre>
<code>wrlDebug log_str1 log_str2</code>	將 <code>log_str1</code> 及 <code>log_str2</code> 字串傳遞到 <code>wrlDebugGeneric()</code>	<code>log_str1</code> 及 <code>log_str2</code> 是除錯字串，代表要執行的動作與收集器，以及要在追蹤檔案中記載的內容。例如：  <pre> wrlTrace Starting os.lib wrlTrace Executing os.lib <b>wrlDebug Starting os.lib</b> <b>wrlDebug Expected libXp</b> ss=./os.lib libXp libXp wrlTrace Finished os.lib echo "os.lib.libXp=\$ss" <b>wrlDebug Finished os.lib</b> <b>wrlDebug OutPutValueIs \$ss</b> wrlTrace Done os.lib </pre>
<code>wrlDebugFuncStart fn_name</code>	將 <code>fn_name</code> 函數傳遞到 <code>wrlDebug()</code>	<code>fn_name</code> 是除錯字串，代表剛剛呼叫的函數。例如：  <pre> wrlDebugFuncStart "\$1" </pre>
<code>wrlDebugFuncExit fn_name</code>	將 <code>fn_name</code> 函數傳遞到 <code>wrlDebug()</code>	<code>fn_name</code> 是除錯字串，代表剛剛完成的函數。例如：  <pre> wrlDebugFuncExit "\$1" </pre>

表 44. UNIX 系統上的記載公用程式函數 (繼續)

函數	說明	輸入參數
wrlDebugFuncReturn <i>result_value</i>	將函數的傳回 <i>result_value</i> 結果寫入日誌檔	<i>result_value</i> 是除錯字串，代表函數的傳回值。例如： `wrlDebugFuncReturn "\$versionCompare"~`
wrlDebugFuncParam <i>param1</i> <i>param2</i>	將 <i>param1</i> 及 <i>param2</i> 參數傳遞到 wrlDebugFunc()	<i>param1</i> 及 <i>param2</i> 是除錯字串，代表被呼叫之函數的已剖析小節標題、已剖析限定元或已剖析輸入引數。例如： `wrlDebugFuncParam "OSArch" "\$3"~`
wrlDebugGeneric <i>formatspec</i> <i>log_str1</i> <i>log_str2</i>	將 <i>log_str1</i> 及 <i>log_str2</i> 字串寫入除錯檔案，且經 <i>formatspec</i> 字串引數格式化	<ul style="list-style-type: none"> <li><i>log_str1</i> 及 <i>log_str2</i> 是字串，代表要在除錯檔案中某一行上記載的特定資料</li> <li><i>formatspec</i> 是字串引數，要置於時間戳記之後，但在日誌字串及換行字元的向左對齊之前</li> </ul> 例如： `wrlDebugGeneric "" "\$1" "\$2"~`
wrlDebugFunc <i>str</i>	將定位點字元及 <i>str</i> 輸入參數傳遞到 wrlDebugGeneric()	<i>str</i> 是字串，代表要記載的資料，即檢查的狀態或所執行的動作。例如： `wrlDebugFunc "Reading config file and parsing using parse array..." ~`
wrlLogFuncStart <i>str</i>	將 <i>str</i> 輸入參數傳遞到 wrlTraceFuncStart() 及 wrlDebugFuncStart()	<i>str</i> 是字串，代表要記載的資料，即所呼叫之函數的名稱。例如： `wrlLogFuncStart "main()"~`
wrlLogFuncExit <i>str</i>	將 <i>str</i> 輸入參數傳遞到 wrlTraceFuncExit() 及 wrlDebugFuncExit()	<i>str</i> 是字串，代表要記載的資料，即所結束之函數的名稱。例如： `wrlLogFuncExit "main()"~`

---

## 注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發；而在其他國家中，IBM 不見得有提供本件中所提及的各項產品、服務或功能。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 之智慧財產權，任何功能相當之產品、程式或服務皆可取代 IBM 之產品、程式或服務。然而，評估及驗證非 IBM 的產品、程式及服務是使用者的責任。

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。本文件使用者並不享有前述專利之任何授權。您可以將特許權限查詢以書面方式寄到：

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, Yamato-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」提供本出版品，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）。

若有些地區在某些交易上並不允許排除上述保證，則該排除無效。

此資訊可能包含技術性的不精確或排版上的錯誤。會定期對此處的資訊進行變更；這些變更會併入到出版品的新版本中。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。

此資訊中提供之對非 IBM 網站的任何參照都只為您方便之用，並不以任何方式作為這些網站的背書。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。

如果本程式之獲授權人為了 (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：

IBM Corporation  
2Z4A/101  
11400 Burnet Road  
Austin, TX 78758 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 IBM 客戶合約、IBM 國際程式授權合約或雙方之任何同等合約的條款，提供本文件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，在其他作業環境下所取得的結果可能有所不同。在開發層次的系統上可能做過一些測量，但不保證在一般系統上有相同的結果。此外，部份測量是透過推斷法來預估。但實際結果可能並非如此。本文件的使用者應依自己的特定環境，查證適用的資料。

非 IBM 產品的相關資訊從這些產品的供應商、其發佈的公告或其他公開來源獲得。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。對於非 IBM 產品功能的疑問應洽詢這些產品的供應商。

一切關於 IBM 未來方針或目的之聲明，隨時可能更改或撤銷，不必另行通知，且僅代表目標與主旨。

此資訊包含日常企業運作所使用的資料和報告的範例。為求儘可能地完整說明，範例包括了個人、公司、品牌及產品的名稱。所有名稱純屬虛構，如有雷同純屬巧合。

版權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。貴客戶可以為了研發、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。這些範例尚未在所有狀況下進行測試。故 IBM 不保證或默示保證這些樣本程式之可靠性、服務性或功能。貴客戶可以為了研發、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。

如果您是以電子檔形式檢視此資訊，可能不會顯示照片及彩色圖例。

## 商標

IBM、IBM 標誌及 [ibm.com](http://ibm.com)<sup>®</sup> 是 International Business Machines Corp. 在世界許多管轄區註冊的商標或註冊商標。其他產品及服務名稱可能是 IBM 或其他公司的商標。如需 IBM 商標的現行清單，可於「版權與商標資訊」網站取得，網址為：[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)。

Adobe、Acrobat、PostScript 和所有以 Adobe 為基礎的商標是 Adobe Systems Incorporated 在美國及/或其他國家或地區的商標或註冊商標。

Cell Broadband Engine 及 Cell/B.E. 是 Sony Computer Entertainment, Inc. 在美國及/或其他國家或地區的商標，並已從此獲授權使用這個商標。

Intel、Intel 標誌、Intel Inside、Intel Inside 標誌、Intel Centrino、Intel Centrino 標誌、Celeron、Intel Xeon、Intel SpeedStep、Itanium 及 Pentium 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。

IT Infrastructure Library 是 Central Computer and Telecommunications Agency（現已納入 Office of Government Commerce）的註冊商標。

ITIL 是「政府商務辦公室」的註冊商標以及已註冊的社群商標，並已在「美國專利商標局 (U.S. Patent and Trademark Office)」註冊。

Linux 是 Linus Torvalds 在美國及 (或) 其他國家或地區的商標。

Microsoft、Windows、Windows NT 及 Windows 標誌是 Microsoft Corporation 在美國及 (或) 其他國家或地區的商標。

UNIX 是 The Open Group 在美國及 (或) 其他國家的商標。

Java 和所有以 Java 為基礎之商標與標誌是 Sun Microsystems, Inc. 在美國及/或其他國家或地區之商標或註冊商標。

其他公司、產品及服務名稱可能是其他公司的商標或服務標記。





---

## 支援資訊及反饋

如果您有 IBM 軟體方面的問題，您當然會想要盡快解決它。IBM 提供不同的方式讓您取得所需的支援，如線上或 IBM Support Assistant。您也可以提供反饋或提交產品加強功能要求。

**線上** 下列網站包含疑難排解資訊：

- 跳至 IBM Prerequisite Scanner 頁面，網址為 IBM 支援中心入口網站。
- 跳至 Prerequisite Scanner 主題，網址為 Service Management Connect。歡迎您對這些主題發佈意見與建議。

使用下列網站提供反饋、提交要求或討論 Prerequisite Scanner：

- 跳至 Prerequisite Scanner 主題，網址為 Prerequisite Scanner at Service Management Connect。歡迎您對這些主題發佈意見與建議。
- 使用 Service Management Connect 處的整合式服務管理訊息板。
- 提交或檢閱 Prerequisite Scanner 產品要求加強功能，網址為 Tivoli RFE 社群。

### **IBM Support Assistant**

IBM Support Assistant (ISA) 是免費的本端軟體服務工作台，可協助您解決 IBM 軟體產品的問題。ISA 可供快速存取與支援相關的資訊及服務功能工具，以進行問題判斷。若要安裝 ISA 軟體，請跳至 <http://www.ibm.com/software/support/isa>



## 索引

索引順序以中文字，英文字，及特殊符號之次序排列。

### 〔三劃〕

子類型

必備項目內容 1, 5

已安裝軟體種類

預先定義的必備項目內容 99

說明 3

### 〔四劃〕

文字檔

結果 23

輸出格式 23

results.txt 23

日誌檔

輸出格式 23

precheck.log 23, 65

prs.debug 23, 67

prs.trc 23, 67

### 〔五劃〕

加強功能 35

必備項目 37

必備項目內容

子類型 1, 43, 44

收集器 20, 21

更新, 自訂 44

更新, 限定元值 44

更新, 預先定義 44

命名慣例 1, 43, 44

限定元 1, 8

格式 1, 43, 44

配置檔 13, 41

參照 81

評估器 22

新增, 自訂 43

新增, 預先定義 43

種類 1, 3, 43, 44, 82, 85, 86, 87, 88,

89, 99, 100

說明 1

類型 1

目錄子類型

說明 5, 89

### 〔六劃〕

共用

收集器, UNIX 49

收集器, Windows 20, 45

評估器, UNIX 22

評估器, Windows 22

共用種類

預先定義的必備項目內容 82

說明 3

回覆碼 70

存取權限定元

說明 8, 89

安裝 37, 38

安裝目錄 37, 38, 64

收集器

說明 20

UNIX

位置 21

命名慣例 21

建立 21, 49

格式 21

規則 21

預先定義的 103

說明 21

標準輸出 21

輸入 103

packageTest.sh, 更新 21, 49, 50

shell 21

Windows

共用 20, 45

位置 20

命名慣例 20

建立 20, 45, 47

格式 20

產品特定 20, 47

規則 20, 45

說明 20

標準輸出 20

VBScript 20

### 〔七劃〕

位置

收集器, UNIX 21

收集器, Windows 20, 45

評估器, UNIX 22, 55

評估器, Windows 22, 52

作業系統種類

預先定義的必備項目內容 89

說明 3

更新

必備項目內容, 自訂 44

必備項目內容, 預先定義 44

限定元 8

限定元值 44

packageTest.sh 50

### 〔八劃〕

使用者種類

預先定義的必備項目內容 99

說明 3

命名慣例

必備項目內容 1

收集器, UNIX 21

收集器, Windows 20

配置檔 13, 41

區段 14

評估器, UNIX 22

評估器, Windows 22

延伸

作業, UNIX 40

作業, Windows 39

檢查, UNIX 40

檢查, Windows 39

服務子類型

說明 5, 89

### 〔九劃〕

建立

收集器, UNIX 21, 49

收集器, Windows 20

共用 45

產品特定 47

配置檔 41

評估器, UNIX 22, 55

評估器, Windows 22, 52

指令行介面

執行 Prerequisite Scanner 57, 62

輸出格式 23, 57

限定元

必備項目內容 1, 8

命名慣例 8

格式 8

規則 8

預先定義的 8, 89

## 〔十劃〕

- 套件子類型
  - 說明 5, 89
- 格式
  - 必備項目內容 1
  - 收集器, UNIX 21
  - 收集器, Windows 20
  - 配置檔 13, 41
  - 區段 14
  - 評估器, UNIX 22
  - 評估器, Windows 22
- 特殊字元
  - 必備項目內容 1
  - Prerequisite Scanner script 57
- 記載公用程式子常式
  - precheck.log 123
- 記載公用程式函數
  - prs.debug 145
  - prs.trc 145
- 記憶體 82
- 追蹤參數
  - 記載公用程式函數 145
  - 說明 57
  - prs.trc 23, 57, 67, 145
- 配置檔
  - 必備項目內容 13, 41
  - 位置 13, 41
  - 作業系統, 受支援 13, 41
  - 命名慣例 13, 41
  - 建立 41
  - 格式 13, 41
  - 副檔名, .cfg 13, 41
  - 區段 13, 14, 41
  - 產品版本 13, 41
  - 規則 13, 41
  - 預先定義的 77
  - 說明 13
  - 標準輸出 13, 41
  - 範例 13, 41
  - 檢查, UNIX 40
  - 檢查, Windows 39
- 除錯
  - 日誌檔 23, 65, 67
  - 除錯 23
  - Prerequisite Scanner 23, 65
- 除錯參數
  - 記載公用程式子常式 123
  - 記載公用程式函數 145
  - 說明 57
  - precheck.log 23, 57, 65, 123
  - prs.debug 23, 57, 67, 145

## 〔十一劃〕

- 區段
  - 命名慣例 14
  - 格式 14
  - 配置檔 13, 14, 41
  - 區段種類 14
  - 新增 43
  - 說明 14
- 執行
  - Prerequisite Scanner 57, 62
- 掃描處理程序 33
- 產品版本
  - 配置檔 13, 41
  - 參數 12, 57
  - 產品型號 12
  - Prerequisite Scanner Script 12, 57
- 產品型號
  - 配置檔 77
  - 參數 12, 57
  - 預先定義的 73
  - 說明 12
  - codename.cfg 12, 41, 73
  - Prerequisite Scanner Script 12, 57
- 產品特定
  - 收集器, Windows 20, 45, 47
- 規則
  - 收集器, UNIX 21
  - 收集器, Windows 20, 45
  - 配置檔 13, 41
  - 產品型號 41
  - 產品型號, 12
  - 評估器, UNIX 22, 55
  - 評估器, Windows 22, 52
- 軟體支援中心 151
- 連線功能種類
  - 說明 3, 86

## 〔十二劃〕

- 單位限定元
  - 說明 8, 89
- 程式庫子類型
  - 說明 5, 89
- 結果
  - 文字檔 23
  - 日誌檔 23
  - 指令行介面 23
- 評估器
  - UNIX
    - 位置 22
    - 命名慣例 22
    - 建立 22, 55
    - 格式 22
    - 規則 22, 55
    - 說明 22

評估器 (繼續)

- UNIX (繼續)
  - 標準輸出 22
  - shell 22, 55
- Windows
  - 共用 22
  - 位置 22
  - 命名慣例 22
  - 建立 22, 52
  - 格式 22
  - 規則 22, 52
  - 說明 22
  - 標準輸出 22
  - VBScript 22, 52

## 〔十三劃〕

- 新增
  - 必備項目內容, 自訂 43
  - 必備項目內容, 預先定義 43
  - 區段 43
  - 產品型號 41
- 詳細參數
  - 說明 57
  - 輸出格式 23, 57
- 路徑名稱 64
- 路徑參數
  - 說明 57

## 〔十四劃〕

- 磁碟 82
- 種類
  - 已安裝軟體 99
  - 必備項目內容 1, 3
  - 共用 82
  - 作業系統 89
  - 使用者 99
  - 連線功能 86
  - 網路 88
  - 環境變數 100
  - Autonomic Deployment Engine 85
  - DB2 86
  - Internet Explorer 87
  - MS SQL Server 87
  - Oracle 89
  - UNIX 網路 100
  - Windows 網路 99
- 網路種類
  - 預先定義的必備項目內容 88
  - 說明 3

## 〔十五劃〕

### 標準輸出

- 收集器, UNIX 21
- 收集器, Windows 20
- 配置檔 13, 41
- 評估器, UNIX 22
- 評估器, Windows 22

## 〔十六劃〕

### 輸出格式

- 文字檔 23
- 日誌檔 23
- 回覆碼 70
- 位置 23
- 指令行介面 23

## 〔十七劃〕

### 應用程式子類型

- 說明 5, 89

### 檔案系統限定元

- 說明 8, 89

### 環境變數區段

- 說明 14

### 環境變數種類

- 預先定義的必備項目內容 100
- 說明 3

## 〔十九劃〕

### 類型

- 必備項目內容 1
  - 收集器 20
  - 評估器 22
- ### 類型限定元
- 說明 8, 89

## C

### codename.cfg

- 更新 41
- 新增產品型號 41
- 說明 12

### CPU 名稱 82

### CPU 區段

- 說明 14

### CPUArch 區段

- 說明 14

## D

### DB2 種類

- 預先定義的必備項目內容 86

### DB2 種類 (繼續)

- 說明 3

### de 種類

- 預先定義的必備項目內容 85

## I

### IBM Support Assistant 151

### Internet Explorer 種類

- 預先定義的必備項目內容 87
- 說明 3

### ISA 151

## M

### MS SQL Server 種類

- 預先定義的必備項目內容 87

## O

### Oracle 種類

- 預先定義的必備項目內容 89
- 說明 3

### OS 版本版本 82

### os 種類

- 請參閱 作業系統種類

### OSArch 區段

- 說明 14

### OSType 區段

- 說明 13, 14

### outputDir 參數

- 說明 57

## P

### p 旗標

- 說明 57

### packageTest.sh

- 收集器, UNIX 21
- 更新 50

### precheck.log

- 記載公用程式子常式 123
- 除錯日誌檔 23, 65, 123
- 除錯參數 23, 57, 65, 123

### Prerequisite Checker Wiki 151

### Prerequisite Scanner

- 二進位 57
- 加強功能 35
- 必備項目 37
- 必備項目內容 1
- 回覆碼 70
- 安裝 37, 38
- 安裝目錄 37, 38, 64
- 收集器 20
- 批次 1

### Prerequisite Scanner (繼續)

- 延伸 39, 40

### 版本 35

### 架構 1, 33

### 根目錄 64

### 配置檔 77

### 除錯 23

### 執行 57, 62

### 掃描處理程序 33

### 產品型號 12, 41, 73

### 結果 23

### 新增特性 35

### 解除安裝 38

### 說明 1

### 輸出格式 23

### Script 語法 57

### shell 1

### VBScript 1

### prereq\_checker

- 參數 57, 62

### 執行 62

### 旗標 57, 62

### 語法 57, 62

### prs.debug

- 記載公用程式函數 145
- 除錯日誌檔 23, 67, 145
- 除錯參數 23, 57, 67, 145

### prs.trc

- 記載公用程式函數 145
- 追蹤日誌檔案 23, 67, 145
- 追蹤參數 23, 57, 67, 145

## S

### Script

- 批次 1

### shell 1

### VBScript 1

### Script 子類型

- 說明 5, 89

### Support Assistant 151

## U

### UNIX 網路種類

- 預先定義的必備項目內容 100

## V

### VBScript

- 收集器, Windows 20
- 評估器, Windows 22

## W

Windows Script 主機 20, 22

Windows 網路種類  
預先定義的必備項目內容 99

## X

xmlResult

XML 結果參數 57





Printed in Taiwan