Prerequisite Scanner
النسخة 1.2

# دليل المستخدم

Prerequisite Scanner
النسخة 1.2

# دليل المستخدم

**ملحوظة**

قبل استخدام هذه المعلومات والبرنامج الذي تدعمه، يجب قراءة المعلومات في "الملاحظات".

تنطبق هذه الطبعة على النسخة 1.2 الى IBM Prerequisite Scanner وكل الاصدارات والتعديلات التالية ما لم يتم الاشارة الى خلاف ذلك في الطبعات الجديدة.

**© Copyright IBM Corporation 2009, 2012.**

حقوق مقصورة على مستخدمي حكومة الولايات المتحدة - الاستخدام أو النسخ أو النشر محدد بواسطة عقد GSA ADP Schedule Contract مع شركة IBM Corp.

# المحتويات


**الأشكال** ..... **vii**

**الجداول** ..... **vii**

**الفصل الأول. مقدمة عن Prerequisite Scanner** ..... **ix**

أسلوب بناء Prerequisite Scanner ..... ١
خصائص المتطلبات الرئيسية ..... ١
أكواد البرنامج ..... ١٣
ملفات توصيف Prerequisite Scanner ..... ١٤
أدوات تجميع Prerequisite Scanner ..... ٢١
أدوات تقييم Prerequisite Scanner ..... ٢٤
نسق المخرجات ..... ٢٥
مجموعة أدوات Prerequisite Scanner Java Developer ..... ٣٣
ملف XML Schema لملف نتائج XML ..... ٣٣
عملية المسح ..... ٣٤
الجديد في هذا الاصدار ..... ٣٥

**الفصل الثاني. تركيب Prerequisite Scanner** ..... **٣٧**

المتطلبات الرئيسية ..... ٣٧
تركيب الملف المضغوط ..... ٣٨
الغاء تركيب Prerequisite Scanner ..... ٣٨

**الفصل الثالث. امتداد Prerequisite Scanner** ..... **٣٩**

قبل تشغيل Prerequisite Scanner ..... ٣٩
عمليات التحقق المطلوبة ومهام الامتداد لأنظمة Windows ..... ٣٩
عمليات التحقق المطلوبة ومهام الامتداد لأنظمة UNIX ..... ٤٠
اضافة أكواد البرنامج ..... ٤١
تكوين ملفات التوصيف المهيأة ..... ٤١
اضافة خصائص المتطلبات الرئيسية ..... ٤٣
تحرير خصائص المتطلبات الرئيسية ..... ٤٥
تكوين أدوات التجميع المهيأة لأنظمة Windows ..... ٤٥
تكوين أدوات تجميع VBScript المهيأة لكل ملفات التوصيف ..... ٤٥
تكوين أدوات تجميع VBScript مهيأة خاصة بمنتج ونسخة المنتج ..... ٤٧
تكوين أدوات التجميع المهيأة لأنظمة UNIX ..... ٤٩
تحرير البرنامج النصي لاختبار مجموعة البرامج لأنظمة UNIX ..... ٥١
تكوين أدوات التحقق المهيأة لأنظمة Windows ..... ٥٢
تكوين أدوات التحقق المهيأة لأنظمة UNIX ..... ٥٧

**الفصل الرابع. تشغيل Prerequisite Scanner** ..... **٥٩**

prereq_checker ..... ٥٩
تشغيل Prerequisite Scanner من سطر الأمر ..... ٦٤
أماكن الدليل المعتادة ..... ٦٥

**الفصل الخامس. تصحيح أخطاء Prerequisite Scanner** ..... **٦٧**

تصحيح الأخطاء على أنظمة Windows ..... ٦٧
تصحيح الأخطاء على أنظمة UNIX ..... ٦٨
مشاكل التنفيذ ..... ٧٠
الأكواد الناتجة ..... ٧١

**الملحق أ. مرجع أكواد البرنامج** ..... **٧٣**

**الملحق ب. مرجع ملفات التوصيف** ..... **٧٧**

**الملحق ج. مرجع خصائص المتطلبات الرئيسية ..... ٨١**

خصائص البيانات العامة ..... ٨٢
تصرف النظام لخاصية المتطلبات الرئيسية للذاكرة والأدوات البرمجية الى Tivoli Monitoring ..... ٨٦
خصائص بيانات Autonomic Deployment Engine ..... ٨٧
خصائص البيانات لامكانية الاتصال ..... ٨٨
خصائص بيانات DB2 ..... ٨٨
خصائص بيانات MS SQL Server ..... ٨٨
خصائص بيانات Internet Explorer ..... ٨٩
خصائص بيانات شبكة الاتصال ..... ٩٠
خصائص بيانات Oracle ..... ٩١
خصائص بيانات نظام التشغيل ..... ٩٢
خصائص بيانات البرامج التي تم تركيبها ..... ١٠٤
خصائص البيانات للمستخدم ..... ١٠٤
خصائص بيانات شبكة الاتصال لنظام Windows ..... ١٠٥
خصائص بيانات شبكة الاتصال لنظام UNIX ..... ١٠٦
خصائص بيانات متغير بيئة التشغيل ..... ١٠٦

**الملحق د. أدوات التجميع سابقة التعريف لأنظمة UNIX ..... ١٠٩**

**الملحق ه. الوظائف المعتادة لأنظمة Windows ..... ١١٥**

allFiles() ..... ١١٦
arrayToString() ..... ١١٦
bigthan() ..... ١١٧
changeMG() ..... ١١٧
checkItemToString() ..... ١١٨
dictionaryToString() ..... ١١٨
exeCommand() ..... ١١٩
filterCommand() ..... ١١٩
filterFile() ..... ١٢٠
findNewest() ..... ١٢٠
findSuitableFile() ..... ١٢١
fmt() ..... ١٢٢
formatForDisplay() ..... ١٢٣
formatSizeForDisplay() ..... ١٢٣
getDecimalSeparator() ..... ١٢٣
getFirstMatch() ..... ١٢٤
isMatch() ..... ١٢٤
notInLatter() ..... ١٢٥
passOrFail() ..... ١٢٦
ppread() ..... ١٢٦
readFile() ..... ١٢٧
unitMGTOG() ..... ١٢٧
varToString() ..... ١٢٨

**الملحق و. البرامج الفرعية لوظيفة التسجيل لأنظمة Windows ..... ١٢٩**

**الملحق ز. البرامج الفرعية لوظيفة الملفات لنظام Windows ..... ١٣١**

**الملحق ح. الوظائف المعتادة للبرامج الفرعية لأنظمة Windows ..... ١٣٣**

ffirstMatch() ..... ١٣٣
getValue() ..... ١٣٤
removeSpecialCharacters() ..... ١٣٥
versionCompare() ..... ١٣٥

**الملحق ط. الوظائف المعتادة لأنظمة UNIX ..... ١٣٧**

changeMG() ..... ١٣٧
AddMG() ..... ١٣٨
compare() ..... ١٣٨
cutdown() ..... ١٣٩
mes4path() ..... ١٤٠
mes4Path1() ..... ١٤١
findOSInfo() ..... ١٤١
telnetNFS() ..... ١٤٢
NFScheck() ..... ١٤٢

**الملحق ي. الوظائف الأخرى لأنظمة UNIX ..... ١٤٥**

formatSizeDisplay() ..... ١٤٦
versionCompare() ..... ١٤٦
checkHpux() ..... ١٤٨
checkLinux() ..... ١٤٨
checkSunOS() ..... ١٤٨
getValue() ..... ١٤٩
setValue() ..... ١٤٩
copyValue() ..... ١٤٩
getSystemId() ..... ١٥٠
getClosestExistingParentDir() ..... ١٥٠
parseDirParameter() ..... ١٥١
printDirSize() ..... ١٥١

**الملحق ك. وظائف تسجيل الوظيفة لأنظمة UNIX ..... ١٥٣**

**الملاحظات ..... ١٥٧**

**معلومات الدعم والمعلومات المرجعية ..... ١٦١**

# الأشكال


١. المخرجات لواجهة تعامل سطر الأمر بأنظمة Windows ..... ٢٦
٢. المخرجات لواجهة تعامل سطر الأمر بأنظمة UNIX ..... ٢٦
٣. ملف precheck.log ..... ٢٨
٤. ملف prs.debug على أنظمة UNIX ..... ٢٩
٥. ملف prs.trc على أنظمة UNIX ..... ٢٩
٦. ملف result.txt على أنظمة Windows ..... ٣٠
٧. ملف result.txt على أنظمة UNIX ..... ٣١
٨. ملف result.XML على أنظمة Windows ..... ٣٢
٩. هيكل بناء Prerequisite Scanner وعملية المسح ..... ٣٤
١٠. تشغيل البرنامج النصي وتحديد معامل التفاصيل في أنظمة UNIX ..... ٦١
١١. تشغيل البرنامج النصي بدون تحديد معامل التفاصيل في أنظمة Windows ..... ٦٢
١٢. ملف precheck.log مع بيانات تصحيح الأخطاء ..... ٦٧
١٣. ملف precheck.log بدون بيانات تصحيح الأخطاء ..... ٦٨
١٤. ملف prs.debug على أنظمة UNIX ..... ٦٩
١٥. ملف prs.trc على أنظمة UNIX ..... ٧٠

# الجداول


١. الحروف الخاصة لتمثيل أنواع المدى ..... ٢
٢. أمثلة لخصائص المتطلبات الرئيسية ..... ٣
٣. تصنيفات خصائص المتطلبات الرئيسية الأساسية ..... ٤
٤. الأنواع الفرعية المعرفة مسبقا ..... ٦
٥. المؤهلات المعرفة مسبقا ..... ١٠
٦. قيم وتصنيفات نوع البيانات المدعمة ..... ١٦
٧. الأقسام التي تم اجراء عملية مسح لها من ملف التوصيف لنظام Windows ..... ١٩
٨. الأقسام التي تم اجراء عملية مسح لها من ملف التوصيف لنظام UNIX ..... ٢١
٩. ملفات توصيف جديدة ..... ٣٥
١٠. عمليات التحقق والمهام قبل استخدام ملف التوصيف لأنظمة Windows ..... ٣٩
١١. عمليات التحقق والمهام قبل استخدام ملف التوصيف لأنظمة UNIX ..... ٤٠
١٢. وسيلة ايضاح الحروف الخاصة للبرنامج النصي الى Prerequisite Scanner ..... ٥٩
١٣. كشف مشاكل التنفيذ ..... ٧٠
١٤. أكواد المنتج المعرفة مسبقا ..... ٧٣
١٥. ملفات التوصيف سابقة التعريف ..... ٧٧
١٦. التصنيفات المعرفة مسبقا لخصائص المتطلبات الرئيسية ..... ٨١
١٧. خصائص المتطلبات الرئيسية للبيانات العامة ..... ٨٢
١٨. خصائص بيانات Autonomic Deployment Engine ..... ٨٧
١٩. خصائص بيانات DB2 ..... ٨٨
٢٠. خصائص بيانات MS SQL Server ..... ٨٩
٢١. خصائص بيانات Internet Explorer ..... ٨٩
٢٢. خصائص بيانات شبكة الاتصال ..... ٩٠
٢٣. خصائص بيانات Oracle ..... ٩١
٢٤. خصائص بيانات نظام التشغيل ..... ٩٢
٢٥. خصائص بيانات البرامج التي تم تركيبها ..... ١٠٤
٢٦. خصائص البيانات للمستخدم ..... ١٠٥
٢٧. خصائص بيانات شبكة الاتصال لنظام Windows ..... ١٠٥
٢٨. خصائص بيانات شبكة الاتصال لنظام UNIX ..... ١٠٦
٢٩. خصائص بيانات متغير بيئة التشغيل ..... ١٠٦
٣٠. أدوات التجميع لنظام UNIX ..... ١٠٩
٣١. الوظائف في common_function.vbs ..... ١١٥
٣٢. الوظيفة التي تم استدعائها لكل نوع متغير. ..... ١٢٨
٣٣. البرامج الفرعية لوظيفة التسجيل ..... ١٢٩
٣٤. البرامج الفرعية لوظيفة الملف. ..... ١٣١
٣٥. الوظائف العاملة للملفات ..... ١٣١
٣٦. الوظائف المعتادة الأخرى والبرامج الفرعية لأنظمة Windows ..... ١٣٣
٣٧. الوظائف الرئيسية التي تقوم باستدعاء ffirstMatch() ..... ١٣٣
٣٨. البرامج باستخدام getValue() ..... ١٣٤
٣٩. الوظائف الرئيسية التي تقوم باستدعاء versionCompare ..... ١٣٥
٤٠. الوظائف في common_function.sh ..... ١٣٧
٤١. الوظائف المعتادة في ملفات متعددة ..... ١٤٥
٤٢. الوظائف المعتادة في TAD722_impl.sh ..... ١٤٥
٤٣. الوظائف الرئيسية التي تقوم باستدعاء versionCompare ..... ١٤٦
٤٤. وظائف تسجيل الوظيفة لأنظمة UNIX ..... ١٥٣

# الفصل الأول. مقدمة عن Prerequisite Scanner

IBM® Prerequisite Scanner هي أداة المسح التي تقوم بتنفيذ عمليات تحديد الهوية والتحقق من المتطلبات الرئيسية للبرامج المحددة قبل اجراء عملية النشر الفعلية. حيث تقوم باجراء عملية مسح للمتطلبات الرئيسية للأجهزة والبرامج بناءا على القيم التي تم تحديدها لخصائص المتطلبات الرئيسية. يقوم الماسح الضوئي بعرض نتائج عملية المسح في واجهة تعامل سطر الأمر وحفظ النتائج في الملفات النصية وملفات XML الاختيارية. كما تقوم بكتابة الرسائل الاعلامية ورسائل التتبع ورسائل تصحيح الأخطاء في ملفات السجل.

تقوم أداة Prerequisite Scanner بالتحقق من نظام تشغيل الجهاز والتحقق مما اذا كان النسخة الصحيحة للبرنامج المحدد. اذا لم تتم أي من عمليات التحقق من المتطلبات الرئيسية المنفردة بنجاح، لن تتم عملية المسح الشامل بنجاح.

يمكنك تشغيل Prerequisite Scanner بعد عملية التركيب أو في أي وقت لتأكيد بيئة التشغيل الحالية الخاصة بك. لا تتطلب أداة Prerequisite Scanner تشغيل برنامج التركيب للبرامج التي تريد التحقق من المتطلبات الرئيسية لها.

يمكنك امتداد Prerequisite Scanner لاجراء مسح للمتطلبات الرئيسية التي لا تعد جزء من المجموعة الرئيسية لعمليات التحقق من المتطلبات الرئيسية المزودة مع الماسح الضوئي.

تقوم أداة Prerequisite Scanner باستدعاء أنواع البرامج النصية التالية بناءا على بيئة التشغيل الخاصة بك:

- نظام Windows: VBScript و batch
- نظام UNIX: shell

**ملاحظة:** لا يمكنك تشغيل البرامج النصية لنظام UNIX على أنظمة Windows حتى اذا كان قد تم تركيب بيئة تشغيل مماثلة لنظام UNIX على أجهزة Windows، على سبيل المثال، Cygwin.

## أسلوب بناء Prerequisite Scanner

تتضمن أداة IBM Prerequisite Scanner المكونات الرئيسية التالية: برنامج نصي ليتم تشغيله في واجهة تعامل سطر الأمر، مجموعة خصائص لعمليات التحقق من المتطلبات الرئيسية، ملفات توصيف خاصية المتطلبات الرئيسية، أدوات تجميع المتطلبات الرئيسية، أدوات تجميع المتطلبات الرئيسية. وتكون نتائج تشغيل Prerequisite Scanner متاحة بنسق مخرجات مختلفة.

### خصائص المتطلبات الرئيسية

خصائص المتطلبات الرئيسية هي القيم المتوقعة للمتطلبات الرئيسية المختلفة للبرامج والأجهزة، المطلوبة بواسطة المنتجات أو الحلول ليتم تركيبها. وتتضمن أمثلة خصائص المتطلبات الرئيسية اجمالي مساحة القرص المتاحة على الجهاز، مجموعة المنافذ غير المستخدمة على الجهاز، مجموعة التطبيقات التي تم تركيبها الحالية.

ولأن قيم خصائص المتطلبات الرئيسية هذه يمكن أن تتغير مع المنتجات المختلفة، يتم تمثيل الخصائص والقيم الخاصة بها كأزواج لقيم الاسم، مع مؤهلات اختيارية. ويتم تخزينها في ملفات توصيف خصائص المتطلبات الرئيسية. توجد خاصية للمتطلبات الرئيسية واحدة فقط بكل سطر.

تلتزم خصائص المتطلبات الرئيسية بالنسق التالي:

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

حيث:

- *prefix_identifier* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض التصنيفات سابقة التعريف.
- *property_name* هو اسم خاصية المتطلبات الرئيسية.
- *suffix_identifier* هو كود تعريف اختياري للنوع الفرعي لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٤.
- *qualifier_name* هو خاصية مميزة اختيارية لخاصية المتطلبات الرئيسية. يقوم IBM Prerequisite Scanner باستخدامه لتأهيل خاصية المتطلبات الرئيسية أو نوع التحقق الذي سيتم تنفيذه على خاصية المتطلبات الرئيسية.

**ملاحظة:** لديك مؤهلات متعددة، يتم الفصل بينهما بفاصلة. يجب تضمين مجموعة المؤهلات بأقواس مربعة [ ] .

- *qualifier_value* هي القيمة للخاصية الاختيارية. يجب الفصل بين كل من المؤهلات والقيم الخاصة بها بواسطة النقطتين : .
- *property_value* هي قيمة خاصية المتطلبات الرئيسية ويمكن أن تكون مجموعة حروف أو عدد صحيح.

يمكن أن تحتوي خاصية المتطلبات الرئيسية على قيمة أو أكثر بناءا على نوع البيانات والمؤهل كما يلي:

- رقم صحيح منفرد، على سبيل المثال، `8080` لتمثيل قيمة رقم المنفذ.
- مدى أو مجموعة من الأرقام الصحيحة يتم تمثيلها باستخدام الحروف الخاصة كما هو موضح في الجدول ١.

*الجدول ١. الحروف الخاصة لتمثيل أنواع المدى*

| الحرف الخاص | الوصف |
|---|---|
| * | يقوم بتعريف المكان المحتجز لعدة قيم. على سبيل المثال، يمكن أن يمثل `ports.*` مجموعة متميزة من المنافذ لكلا من منتج قاعدة البيانات، `ports.DB` و IBM WebSphere® Application Server، `ports.WAS`. |
| + | يقوم بتعريف أن النسخة الفعلية يجب أن تكون الأقل مطابقة لقيمة النسخة المتوقعة. على سبيل المثال، `os.versionNumber=5.0+`، يعني أن النسخة يجب أن تكون 5.0 أو أكثر. |
| - | يقوم بتعريف أن النسخة الفعلية يجب أن تكون الأكثر مطابقة لقيمة النسخة المتوقعة. على سبيل المثال، `os.versionNumber=5.0-`، يعني أن النسخة يجب أن تكون 5.0 أو أقل. |
| .* | يقوم بتعريف أن النسخة الفعلية يمكن أن تكون مطابقة لأية قيمة لحرف الابدال للنسخة المتوقعة. على سبيل المثال، `os.versionNumber=5.*`، يعني أن النسخة يمكن أن تكون 5.0 أو 5.0.1 أو 5.5. |

**قيد:** على أنظمة Windows، يكون حرف الابدال * مدعما فقط اذا تم استخدامه في التعبير المعتاد في خاصية المتطلبات الرئيسية `OS Version`.

- مجموعة الحروف التي يمكن أن تمثل أي من القيم التالية لأنواع المتطلبات الرئيسية:
  - قيمة رقمية مع وحدة، على سبيل المثال، `8GB` أو `10MB`
  - تطبيق أو نظام تشغيل أو هيكل بناء أو مجموعة برامج، على سبيل المثال، `IBM Lotus Symphony` أو `RedHat Enterprise Linux 5.4` أو `32-bit` أو `ftp`

    **ملاحظة:** قد تشمل مجموعة الحروف عدة قيم أيضا مفصولة بفاصلة، على سبيل المثال، كشف التطبيقات.
  - كلاهما أو القيم التي تم تمثيلها بواسطة أحد المجموعات التالية، مثل `True|False` أو `Available|Unavailable` أو `Enabled|Disabled`

يوضح الجدول ٢ أمثلة لخصائص المتطلبات الرئيسية.

*الجدول ٢ . أمثلة لخصائص المتطلبات الرئيسية*

| خاصية المتطلبات الرئيسية | الشرح |
|---|---|
| `Disk=1GB` | حجم المساحة الخالية على قرص التخزين، حيث:<br>• *property_name* هو `Disk`<br>• *property_value* هي `1GB` |
| `user.isAdmin=True` | سواء كان المستخدم الذي قام ببدء الاتصال ينتمي الى مجموعة موجه النظام ، حيث:<br>• *prefix_identifier* هو `user`، لخصائص المتطلبات الرئيسية للمستخدم<br>• *property_name* هو `isAdmin`<br>• *property_value* هي `True` |
| `network.availablePorts.DB=60000-60005`<br>`network.availablePorts.WAS=8080`<br>`network.availablePorts.FTP=21` | يتم التحقق مما اذا كانت المنافذ 60000-60005 متاحة لوحدة خدمة قاعدة البيانات، وما اذا كان المنفذ 8080 متاحا لوحدة خدمة WebSphere Application Server، وما اذا كان المنفذ 21 متاحا الى FTP، حيث:<br>• *prefix_identifier* هو `network`، لخصائص المتطلبات الرئيسية العامة<br>• *property_name* هي `availablePorts`<br>• *suffix_identifier* هي `DB` لمنافذ قاعدة البيانات المتاحة، `WAS` لمنفذ WebSphere Application Server المتاح، و `FTP` لمنفذ FTP المتاح<br>• *property_value* هي `60000-60005` أو `8080` أو `21` |
| `os.dir.home=[dir:/home,type:permission]755+` | يتم التحقق مما اذا كان الدليل الرئيسي له تصاريح `drwxr-xr-x`، حيث<br>• *prefix_identifier* هو `os`، لخصائص المتطلبات الرئيسية لنظام التشغيل<br>• *property_name* هو `dir`<br>• *suffix_identifier* هو `home` للدليل للتحقق منه<br>• *qualifier_name* هو `dir` والنوع الذي يقوم بتأهيل خاصية المتطلبات الرئيسية ونوع عملية التحقق<br>• *qualifier_value* هي `home` والتصريح والقيم للمؤهلات<br>• *property_value* هي `+755`، الذي يعد تمثيل للرقم الثماني لتصاريح التوصل للدليل الرئيسي |

يمكنك اضافة أو تحرير خصائص المتطلبات الرئيسية المعرفة مسبقا لكل منتج تريد تشغيل Prerequisite Scanner له. يمكنك أيضا تكوين خصائص مهيأة للمتطلبات الرئيسية واستخدام أدوات تجميع وأدوات تقييم Prerequisite Scanner وفقا للمطلوب لاجراء عملية مسح لها ومقارنتها بخصائص المتطلبات الرئيسية.

**المفاهيم المتعلقة**:

"المؤهلات المعرفة مسبقا لخصائص المتطلبات الرئيسية"

تقوم أداة IBM Prerequisite Scanner مجموعة من المؤهلات الأساسية لبعض خصائص المتطلبات الرئيسية في تصنيف معرف مسبقا. تمثل المؤهلات الخصائص المميزة لخاصية المتطلبات الرئيسية التي تقوم Prerequisite Scanner باستخدامها لتأهيل خاصية المتطلبات الرئيسية أو نوع التحقق الذي سيتم تنفيذ على خاصية المتطلبات الرئيسية.

**التصنيفات المعرفة مسبقا لخصائص المتطلبات الرئيسية**

تقدم الأداة IBM Prerequisite Scanner مجموعة من خصائص المتطلبات الرئيسية الأساسية لتصنيفات البيانات المختلفة: عام، البرامج التي تم تركيبها، نظام التشغيل، المستخدم، امكانية الاتصال، Internet Explorer، وحدة خدمة قاعدة البيانات، متغيرات بيئة التشغيل، شبكة الاتصال متضمنة خصائص بيئة التشغيل لنظام Windows و UNIX.

*<prefix_identifier>* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية.

يوضح الجدول ٣ التصنيفات المعرفة مسبقا للمتطلبات الرئيسية للأجهزة والبرامج.

*الجدول ٣. تصنيفات خصائص المتطلبات الرئيسية الأساسية*

| تصنيف البيانات | الوصف | بادئة كود التعريف المطلوبة |
|---|---|---|
| عام | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية العامة مثل سرعة المشغل و RAM والقرص والمساحة المؤقتة. هذا المثال لخاصية المتطلبات الرئيسية للتحقق من نظام التشغيل:<br>`OS Version=RedHat Enterprise Linux 5.4` | لا شيء |
| البرنامج الذي تم تركيبه | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية للبرنامج الذي تم تركيبه مثل البرامج المسجلة في سجل Windows وما اذا كان قد تم تركيب cygwin و gskit. هذا المثال لخاصية المتطلبات الرئيسية لاجراء مسح لسجل نظام التشغيل للبحث عن البرامج التي تم تركيبها مع الأماكن:<br>`installedSoftware=`*`list_of_installed_programs`* | لا شيء |
| المستخدم | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية للمستخدم مثل ما اذا كان يتوافر لدى المستخدم الذي قام ببدء الاتصال صلاحيات ادارية أو يعد مستخدم رئيسي. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كان المستخدم الذي قام ببدء الاتصال أحد أعضاء مجموعة موجه النظام:<br>`user.isAdmin=True` | `user` |
| نظام التشغيل | يقوم هذا التصنيف بالتحقق من متطلبات نظام التشغيل مثل النسخة وهيكل البناء واجمالي الذاكرة والذاكرة المتاحة واجمالي الذاكرة الفعلية. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كان يتم تشغيل خدمة السجل عن بعد:<br>`os.isServiceRunning.remoteRegistry=True` | `os` |
| امكانية الاتصال | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية لامكانية الاتصال مثل ما اذا كان يتم تشغيل Telnet وعناوين IP والمنافذ التي يمكن توصيل الماسح الضوئي بها. | لا شيء |
| شبكة الاتصال | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية لشبكة الاتصال التي يمكن أن تكون عامة عبر كل بيئات التشغيل مثل ما اذا كان هناك منافذ متاحة. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كان المنفذ `8080` متاحا لوحدة خدمة IBM WebSphere Application Server:<br>`network.availablePorts.was=8080` | `network` |

| | | |
|---|---|---|
| شبكة اتصال Windows | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية لشبكة اتصال Windows مثل ما اذا كان قد تم اتاحة NetBIOS و DHCP على الجهاز وخصائص اختبار الاتصال. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كان هناك موفق واحد على الأقل ذو عنوان IP صحيح تم اتاحة NetBIOS به كبروتوكول:<br>`network.netBIOSEnabled=True` | `network` |
| شبكة اتصال UNIX | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية لشبكة اتصال UNIX مثل ما اذا كان قد تم اتاحة NetBIOS و DHCP على الجهاز وخصائص اختبار الاتصال. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كان النظام الرئيسي المحلي يستجيب لبروتوكول اختبار الاتصال:<br>`network.pingLocalhost=True` | `network` |
| Internet Explorer | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية الى Microsoft Internet Explorer مثل النسخة. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كانت نسخة Internet Explorer هي 7.0:<br>`internetExplorer.version=7.0` | `internetExplorer` |
| وحدة خدمة قاعدة البيانات، DB2® | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية الى DB2 مثل النسخة. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كانت نسخة DB2 هي على الأقل 9.5:<br>`DB2 Version=9.5.*` | `DB2` |
| وحدة خدمة قاعدة البيانات، Oracle | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية الى Oracle مثل النسخة. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كانت نسخة الوحدة التابعة الى Oracle هي على الأقل 9.2.0.8:<br>`oracle.Client=9.2.0.8+` | `Oracle` |
| متغير بيئة التشغيل | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية لمتغير بيئة التشغيل مثل ما اذا كان قد تم تحديد متغير بيئة التشغيل. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كان مسار الفئة يحتوي على ملف Derby JAR:<br>`env.classpath.derbyJAR=False` | `env` |
| Autonomic Deployment Engine | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية الى Autonomic Deployment Engine مثل ما اذا كان قد تم تركيب Autonomic Deployment Engine أو يعد وحدة تركيب Tivoli® Integrated Portal. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كان قد تم تركيب وحدة تركيب Tivoli Integrated Portal Version 2.1.1.0 أو 2.1.1.1 على نظام Windows:<br>`de.installationUnit=regex{.*C37109911C8A11D9 8E1700061BDE7AEA.* \|.*TIP 2.1.1.0.*\|.*TIP 2.1.1.1.*}` | `de` |
| وحدة خدمة قاعدة البيانات، MS SQL | يقوم هذا التصنيف بالتحقق من المتطلبات الرئيسية الى MS SQL مثل النسخة. هذا المثال لخاصية المتطلبات الرئيسية للتحقق مما اذا كانت نسخة MS SQL Server هي SQL Server 2008 R2 Developer Edition:<br>`mssql.Server=10.50.1600.1` | `mssql` |

**الأنواع الفرعية المعرفة مسبقا لخصائص المتطلبات الرئيسية**

تقدم أداة IBM Prerequisite Scanner مجموعة من الأنواع الفرعية الأساسية لبعض خصائص المتطلبات الرئيسية في تصنيف معرف مسبقا. وتقدم الأنواع الفرعية المزيد من التصنيف لخاصية المتطلبات الرئيسية مثل التصنيف وفقا للتطبيق أو الوظيفة أو النوع الفرعي للخدمة.

على سبيل المثال، يمكن أن يكون لديك خاصية متطلبات رئيسية لمنافذ شبكة الاتصال المتاحة. ويمكنك تصنيف خاصية المتطلبات الرئيسية هذه بشكل أكثر للتحقق من المنافذ المتاحة لوحدة خدمة قاعدة البيانات أو وحدة خدمة التطبيق أو البروتوكول.

*<suffix_identifier>* هو كود تعريف اختياري للنوع الفرعي في اسم خاصية المتطلبات الرئيسية.

يوضح الجدول ٤ الأنواع الفرعية المعرفة مسبقا لتصنيفات خصائص المتطلبات الرئيسية المختلفة متضمنة *<suffix_identifier>*.

*الجدول ٤. الأنواع الفرعية المعرفة مسبقا*

| النوع الفرعي لخاصية المتطلبات الرئيسية | بادئة كود التعريف | بيئة التشغيل | الوصف | القيم الصحيحة للنوع الفرعي |
|---|---|---|---|---|
| **تصنيف شبكة الاتصال المستقلة لبيئة التشغيل** | | | | |
| `network.availablePorts.`*`app_type`* | *app_type* | كل | استخدم مصطلح التسمية هذا للتحقق مما اذا كان المنفذ أو مدى المنافذ لا يتم الانصات من خلاله أو ما اذا كان متاحا لنوع التطبيق *app_type*. | مجموعة حروف لتمثيل *app_type*، على سبيل المثال:<br>• `DB2` تقوم بالتحقق من المنافذ لوحدة خدمة قاعدة بيانات DB2<br>• `WAS` تقوم بالتحقق من المنافذ لوحدة خدمة WebSphere Application Server<br>• `ftp` تقوم بالتحقق من منفذ FTP |
| `network.portsInUse.`*`app_type`* | *app_type* | كل | استخدم مصطلح التسمية هذا للتحقق مما اذا كان المنفذ أو مدى المنافذ يتم الانصات من خلاله أو يتم استخدامه لنوع التطبيق *app_type*. | مجموعة حروف لتمثيل *app_type*، على سبيل المثال:<br>• `DB2` تقوم بالتحقق من المنافذ لوحدة خدمة قاعدة بيانات DB2<br>• `WAS` تقوم بالتحقق من المنافذ لوحدة خدمة WebSphere Application Server<br>• `ftp` تقوم بالتحقق من منفذ FTP |
| **تصنيف نظام التشغيل** | | | | |
| `os.dir.`*`dir_name`* | *dir_name* | نظام UNIX | استخدم مصطلح التسمية هذا للتحقق من نظام ملفات *dir_name*. تقوم قيمة خاصية المتطلبات الرئيسية باستخدام المؤهلات المعرفة مسبقا. | مجموعة حروف لتمثيل *dir_name*، على سبيل المثال:<br>• `tmp`<br>• `home` |

| | | | | |
|---|---|---|---|---|
| os.file.<br>*script_name* | *script_*<br>*name* | نظام<br>UNIX | استخدم مصطلح التسمية هذا للتحقق مما اذا كان البرنامج النصي *script_name* متاحا على الجهاز. | مجموعة حروف لتمثيل *script_name*، على سبيل المثال:<br>• bash<br>• expect<br>• gzip<br>• tar |
| os.<br>isService<br>Running.<br>*service_name* | *service_*<br>*name* | نظام<br>Windows | استخدم مصطلح التسمية هذا للتحقق مما اذا كان يتم تشغيل خدمة *service_name* هذه على الجهاز. | مجموعة حروف لتمثيل *service_name*، على سبيل المثال:<br>• remoteRegistry<br>• DNSClient<br>• terminalServices |

| | | | | |
|---|---|---|---|---|
| `os.lib.`<br>*`lib_name_version`* | *lib_name*<br>*_version* | نظام UNIX | استخدم مصطلح التسمية هذا للتحقق مما اذا كان قد تم تركيب نسخة المكتبة *lib_name_version* المدعمة على الجهاز. | مجموعة حروف تمثل *lib_name_version*، على سبيل المثال، بخط عريض:<br>• مكتبة **libstdc++.so.#** ذات ٣٢-بت<br>• مكتبة **libstdc++.so.#** ذات ٦٤-بت<br>• مكتبة **libXft.so.#** ذات ٣٢-بت<br>• مكتبة **libXtst.so.#** ذات ٣٢-بت<br>• مكتبة **libaio.so.#** ذات ٦٤-بت<br>• مستوى بيئة تشغيل XLC **xlC.rte** ذات ٣٢-بت<br>• بيئة تشغيل **xlC.aix50.rte** XLC ذات ٣٢-بت الى AIX® النسخة 5.3<br>• بيئة تشغيل **xlC.aix61.rte** XLC ذات ٣٢-بت الى AIX النسخة 6.1<br>• مكتبة AIX IOCP **bos.iocp.rte**<br>• **bos.loc.iso.en_us**، فئة الملف لكود ISO لنظام التشغيل الأساسي AIX<br><br>`regex {`*`str`*`}`، التعبير المعتاد مع معامل المدخلات، *str*، يمثل نموذج البحث لاسم المكتبة، على سبيل المثال:<br>`regex {.*libgcc.*}`<br>يتم التحقق من وجود نسخة مكتبة التشغيل ذات المستوى المنخفض GCC، `libgcc`، لنظام التشغيل هذا. |

| | | | | |
|---|---|---|---|---|
| os.package.*package_name* | *package_name* | نظام UNIX | استخدم مصطلح التسمية هذا للتحقق مما اذا كان قد تم تركيب نسخة مجموعة برامج *package_name* المدعمة على الجهاز. | مجموعة حروف تمثل *package_name*، على سبيل المثال، بخط عريض:<br>• الوحدة الطرفية **bash**<br>• **expect** لمجموعة برامج امتداد TCL<br>• **libgcc** لمجموعة برامج بيئة التشغيل ذات المستوى المنخفض الى GCC<br>• **openssh** للوحدة الطرفية المؤمنة للمصدر المفتوح<br>• **openssl** لمجموعة أدوات المصدر المفتوح SSL/TLS<br>• **perl** لمجموعة البرامج النصي الى Perl<br>• **rpm** لمجموعات برامج بناء RPM أو RPM<br>• **telnet** لمجموعة برامج Telnet<br>• **wget** لمجموعة برامج استرجاع ملف GNU |
| os.space.*dir_name* | *dir_name* | نظام UNIX | استخدم مصطلح التسمية هذا للتحقق من مساحة القرص المتاحة لنظام الملفات *dir_name* المحدد. تقوم قيمة خاصية المتطلبات الرئيسية باستخدام المؤهلات المعرفة مسبقا. | مجموعة حروف لتمثيل *dir_name*، على سبيل المثال:<br>• usr<br>• home<br>• tmp<br>• var |

**المؤهلات المعرفة مسبقا لخصائص المتطلبات الرئيسية**

تقوم أداة IBM Prerequisite Scanner مجموعة من المؤهلات الأساسية لبعض خصائص المتطلبات الرئيسية في تصنيف معرف مسبقا. تمثل المؤهلات الخصائص المميزة لخاصية المتطلبات الرئيسية التي تقوم Prerequisite Scanner باستخدامها لتأهيل خاصية المتطلبات الرئيسية أو نوع التحقق الذي سيتم تنفيذ على خاصية المتطلبات الرئيسية.

على سبيل المثال، لديك خاصية للمتطلبات الرئيسية لنظام الملفات. يمكنك تأهيل عملية التحقق التي سيتم تنفيذها لخاصية المتطلبات الرئيسية هذه بناءا على الخصائص المميزة لاسم نظام الملفات وتصاريح التوصل الخاصة بها. ويمكنك أيضا تأهيل نوع الوحدات الذي سيتم استخدامه عند التحقق من مساحة القرص المتاحة بناءا على الخصائص المميزة لمسار ووحدة نظام الملفات.

تقوم المؤهلات بدعم خاصية التهيئة لتحقيق متطلبات بيئة التشغيل الخاصة بك ومنع الماسح الضوئي من تكوين افتراضات ضمنية عن الخصائص المميزة للمتطلبات الرئيسية متعددة الأبعاد مثل تصاريح المسار والتوصل المفترضة. يمكنك تغيير قيم المؤهلات المعرفة مسبقا، لكن لا يمكنك اضافة مؤهلات جديدة الى مجموعة المؤهلات المعرفة مسبقا الحالية لخاصية المتطلبات الرئيسية المعرفة مسبقا.

يجب أن تلتزم المؤهلات بالنسق التالي:

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

حيث:

- *qualifier_name* هي خاصية مميزة اختيارية لخاصية المتطلبات الرئيسية تقوم IBM Prerequisite Scanner باستخدامها لتأهيل خاصية المتطلبات الرئيسية أو نوع التحقق الذي سيتم تنفيذه على خاصية المتطلبات الرئيسية.
- *qualifier_value* هي القيمة للخاصية الاختيارية.

يمكن أن تكون قيمة المؤهل زوج قيم الاسم لدعم عدة قيم صحيحة بناءا على نوع المستخدم. على سبيل المثال، تعتمد المسارات المختلفة للدليل الرئيسي على ما اذا كان مستخدم رئيسي أو غير رئيسي.

- *property_value* هي قيمة خاصية المتطلبات الرئيسية ويمكن أن تكون مجموعة حروف أو عدد صحيح.

يجب الفصل بين كل من المؤهلات والقيمة الخاصة بها باستخدام النقطتين `:`. لديك مؤهلات متعددة، يتم الفصل بينهما بفاصلة. يجب تضمين مجموعة المؤهلات بأقواس مربعة `[]`.

يوضح الجدول ٥ المؤهلات المعرفة مسبقا لتصنيفات خصائص المتطلبات الرئيسية المختلفة. تقوم بعض خصائص المتطلبات الرئيسية باستخدام الأنواع الفرعية المعرفة مسبقا لمزيد من التصنيف لخاصية المتطلبات الرئيسية.

**هام:** لا يمكنك استخدام المؤهلات المعرفة مسبقا مع خصائص المتطلبات الرئيسية المعرفة مسبقا الأخرى.

*الجدول ٥. المؤهلات المعرفة مسبقا*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | المؤهلات والقيم الصحيحة |
|---|---|---|---|
| **تصنيف نظام التشغيل مع نوع فرعي معرف مسبقا** | | | |
| `os.dir.`*`dir_name`* | نظام UNIX | للتحقق من نظام الملفات *dir_name* بناء على الخواص المميزة المؤهلة التالية:<br>• الخاصية المميزة `dir`، لتحديد نظام الملفات الذي سيتم فحصه.<br>• الخاصية المميزة `type`، لتحديد الخاصية المميزة لنظام التشغيل التي سيتم التحقق منها، على سبيل المثال، التمثيل المكون من أرقام ثمانية `<octal_digits>` لتصاريح الوصول الى نظام الملفات هذا.<br><br>*<dir_name>* يمكن أن تمثل:<br>• `tmp`<br>• `home` | مجموعة حروف بالنسق المؤهل التالي:<br>`[dir:`*`dir_name`*`, type:permission] `*`octal_digits`*`+`<br><br>على سبيل المثال، للتحقق مما اذا كان الدليل الرئيسى له تصاريح `drwxr-xr-x`:<br>`os.dir.home=[dir:/home, type:permission]755+` |

| | | | |
|---|---|---|---|
| os.space.*dir_name* | نظام UNIX | يتم التحقق من مساحة القرص المتاحة لنظام ملفات *dir_name* المحدد بواحدة أو أكثر من الخصائص المميزة المؤهلة التالية:<br>• الخاصية المميزة dir، لتحديد المسار الذي سيتم فحصه من نظام الملفات<br>• الخاصية المميزة unit، لتحديد وحدات قرص التخزين التي سيتم استخدامها<br><br>تتوقف قيمة الخاصية المميزة dirعلى الأشخاص القائمين بالاتصال؛ وعلى ذلك، فان القيمة هي زوج قيمة الاسم التي تمثل نوع المستخدم، أي الدليل الرئيسي أو غير الدليل الرئيسي، والمسار المتعلق.<br><br>*dir_name* يمكن أن تمثل:<br>• usr<br>• home<br>• tmp<br>• var | مجموعة حروف بالنسق المؤهل التالي لنظام الملفات لمستخدم رئيسي:<br>[dir:**root**=*dir_path*, unit:*unit_name*] *disk_space*<br><br>على سبيل المثال:<br>os.space.usr= [dir:root=/usr/ibm/common /acsi, unit:GB]200<br><br>مجموعة حروف بالنسق المؤهل التالي لنظام الملفات لمستخدم غير رئيسي:<br>[dir:**non_root**=*dir_path*, unit:*unit_name*] disk_space<br><br>على سبيل المثال:<br>os.space.home= [dir:non_root=USERHOME/.acs i_HOST, unit:MB]200<br><br>مجموعة حروف بالنسق المؤهل التالي، تقوم باستخدام مؤهل واحد فقط:<br>[dir:*dir_path*] *disk_space* MB<br><br>على سبيل المثال:<br>os.space.home=[dir:/home /sat]250MB |
| **تصنيف نظام التشغيل بدون نوع فرعي معرف مسبقا** | | | |
| os.mountcheck | نظام UNIX | يتم التحقق مما اذا كان قد تم تركيب نظام الملفات بناءا على الخصائص المميزة المؤهلة التالية:<br>• الخاصية المميزة drive، لتحديد الدليل الذي يعد نظام الملفات الذي تم تركيبه<br>• الخاصية المميزة nosuid، لتحديد ما اذا كان اختيار التحميل قد تم تحديده أم لا عند تحديد نظام الملفات | مجموعة حروف بالنسق المؤهل التالي:<br>[drive:*dir_name*, *mount_option*: false\|true] True\|False<br><br>على سبيل المثال، للتحقق مما اذا كان الدليل home/ موجودا ومن عدم تحديد الاختيار nosuid:<br>os.mountcheck=[drive:/home, nosuid:false]True |

| | | | |
|---|---|---|---|
| os.SELinux | نظام Linux | للتحقق من حالة الفرض لخاصية Linux لتحسين السرية بناء على الخواص المميزة المؤهلة التالية:<br>• الخاصية المميزة source، لتحديد الأمر الذي سيتم استخدامه لنظام التشغيل المتعلق | • مجموعة حروف بالنسق المؤهل التالي:<br>[source:Command] Disabled\|Enabled<br>على سبيل المثال، للتحقق مما اذا كانت الخاصية ستكون غير متاحة أو يتم السماح بها على نظام تشغيل Red Hat أو SUSE:<br>os.SELinux=[source: Command]Disabled<br>• مجموعة حروف بدون مؤهل، حيث يكون نظام التشغيل Linux عام:<br>os.SELinux=Disabled |
| os.ulimit | نظام UNIX | استخدم مصطلح التسمية هذا للتحقق مما اذا كان يمكن تشغيل عدد العمليات غير المحدود بناءا على الخصائص المميزة المؤهلة التالية:<br>• الخاصية المميزة type، لتحديد الحد الاضافي الذي سيتم التحقق منه، على سبيل المثال، يقوم filedescriptorlimit بالتحقق من الحد لعدد موصفات الملف التي يمكن أن تقوم العملية بفتحها | مجموعة حروف بالنسق المؤهل التالي:<br>[type:*limit_name*] *limit_value*, limited\|unlimited<br>على سبيل المثال، للتحقق مما اذا كان حد موصف الملف أكبر من 8192، مع عدد غير محدود من العمليات:<br>os.ulimit= [type:filedescriptorlimit] 8192+,unlimited<br>الأنواع الصحيحة للحدود التي سيتم التحقق منها، حيث *limit_name* تمثل نوع الحد كما يلي:<br>• ALL، للتحقق من كل الحدود<br>• corefilesizelimit<br>• datasegmentlimit<br>• filedescriptorlimit<br>• filesizelimit<br>• hardlimit<br>• processlimit<br>• maxmemorysizelimit<br>• maxprocesseslimit<br>• stacksizelimit<br>• threadlimit |
| **تصنيف عام بدون نوع فرعي معرف مسبقا** | | | |

| | | | |
|---|---|---|---|
| Disk | نظام Windows | حجم مساحة القرص الخالية، مع الخواص المميزة المؤهلة الاختيارية:<br>• الخاصية المميزة dir، لتحديد المسار الذي سيتم فحصه من الدليل<br>• الخاصية المميزة unit، لتحديد وحدات قرص التخزين التي سيتم استخدامها | مجموعة حروف بالنسق المؤهل التالي:<br>[dir:*dir_path*, unit:*unit_name*] *disk_space*<br><br>على سبيل المثال:<br>Disk=<br><br>[dir:C:\Program Files\IBM\SQLLIB, unit:MB]1431<br><br>نسق رقمي بالجيجابايت أو الميجابايت:<br><disk_space>MB\|GB<br><br>على سبيل المثال:<br>Disk=250MB |

**أكواد البرنامج**

تقوم IBM Prerequisite Scanner باستخدام الأكواد متعددة الحروف في أسماء الملفات وأسماء المعاملات لتعريف المنتجات والمكونات وتحديد نوع ملف التوصيف الذي سيتم استخدامه.

***product_code***

يعد المتغير المستخدم لتمثيل كود المنتج على نظام Windows أو نظام UNIX. تقوم أكواد المنتج بتعريف المنتج، وبيئة تشغيل منفردة مثل Windows و AIX و HP-UX و Linux و Solaris، واختياريا نسخة نظام التشغيل المدعمة بواسطة هذا المنتج. ويتم تخزينها في الملف codename.cfg. يحتوي أي منتج يدعم عدة بيئات تشغيل على عدة أكواد للمنتج مع كلا منهما، حيث تقوم بتعريف المنتج وبيئة التشغيل ونسخة نظام التشغيل وفقا للمطلوب.

على سبيل المثال، تقوم أكواد المنتج COD و COK و COX بتعريف بعض أنظمة التشغيل المدعمة ونسخ IBM Tivoli Provisioning Manager:

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

عند تشغيل Prerequisite Scanner، تقوم بامرار كود المنتج واختياريا نسخة المنتج كمعاملات للمدخلات. يقوم الماسح الضوئي بالتحقق من وجود كود المنتج في الملف codename.cfg. بأنظمة UNIX، اذا لم يتم ايجاد الكود، سيقوم الماسح الضوئي بالخروج. وبأنظمة Windows، اذا لم يتم ايجاد الكود، لن يقوم الماسح الضوئي بالخروج.

عندئذ سيقوم الماسح الضوئي باستخدام معاملات المدخلات لايجاد ملف التوصيف في الدليل *ips_root*/Windows|UNIX_Linux. سيحتوي اسم الملف على نفس كود المنتج ونسخة المنتج كمعاملات المدخلات. اذا لم تقم بامرار معامل نسخة المنتج الاختياري، سيقوم الماسح الضوئي باستخدام أحدث نسخة لملف التوصيف يتم ايجادها في هذا الدليل. ثم تقوم أداة Prerequisite Scanner ببدء عملية المسح.

**ملاحظة:** بأنظمة Windows فقط: اذا لم يكن كود المنتج موجودا في الملف codename.cfg لكن ملف التوصيف ذو كود المنتج كان موجودا في الاسم، ستقوم أداة Prerequisite Scanner بعرض كود المنتج ورقم النسخة في المخرجات، مع عدم تعريف اسم المنتج.

## ملفات توصيف Prerequisite Scanner

تحتوي ملفات توصيف IBM Prerequisite Scanner لبيئات التشغيل المنفردة على خصائص المتطلبات الرئيسية والقيم المتوقعة لها لكل بيئة تشغيل مدعمة بواسطة المنتج. يتيح Prerequisite Scanner مجموعة سابقة التعريف من ملفات التوصيف يمكن تحريرها. يجب تكوين ملفات توصيف للمنتجات الجديدة وبيئات التشغيل التي سيتم دعمها.

تحتوي ملفات التوصيف على لاحقة الملف `.cfg`. يمكنك تخزينها في الدليل *ips_root*`/`*<OS>*، حيث *<OS>* هو اسم نوع نظام التشغيل، على سبيل المثال `Windows` أو `UNIX_Linux`.

يجب أن تلتزم ملفات التوصيف بالقواعد التالية:

- يجب أن تكون لاحقة الملف هي `.cfg`.
- طريقة التسمية لاسم الملف:

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

  حيث:

  – *product_code*

  يعد المتغير المستخدم لتمثيل كود المنتج على نظام Windows أو نظام UNIX. تقوم أكواد المنتج بتعريف المنتج، وبيئة تشغيل منفردة مثل Windows و AIX و HP-UX و Linux و Solaris، واختياريا نسخة نظام التشغيل المدعمة بواسطة هذا المنتج. ويتم تخزينها في الملف `codename.cfg`. يحتوي أي منتج يدعم عدة بيئات تشغيل على عدة أكواد للمنتج مع كلا منهما، حيث تقوم بتعريف المنتج وبيئة التشغيل ونسخة نظام التشغيل وفقا للمطلوب.

  – *<version>* هو كود مكون من ٨-أرقام يمثل النسخة، والاصدار، والتعديلات، والمستوى، مع رقمين لكل من أجزاء الكود؛ على سبيل المثال 7.3.21 هو 07032100.

- تجميع خصائص المتطلبات الرئيسية تحت الأقسام التي يجب أن تقوم باتباع مصطلح التسمية لعناوين الأقسام.
- النسق القياسي لكل خاصية للمتطلبات الرئيسية هو زوج قيم الاسم مع المؤهلات الاختيارية، وخاصية واحدة فقط بكل سطر:

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

### مثال لملف التوصيف بدون أقسام

يقوم هذا المثال بالتحقق من خصائص المتطلبات الرئيسية لكن لا يقوم بالتمييز بين خصائص المتطلبات الرئيسية المختلفة لنسخ نظام التشغيل المطلوبة.

```
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
```

**المفاهيم المتعلقة**:

"الأقسام في ملفات التوصيف"

يمكن تجميع خصائص المتطلبات الرئيسية تحت مجموعة من الأقسام في ملفات التوصيف، مع تمثيل كل قسم لتصنيف نوع بيانات. تكون الأقسام اختيارية في ملفات التوصيف.

**الأقسام في ملفات التوصيف**

يمكن تجميع خصائص المتطلبات الرئيسية تحت مجموعة من الأقسام في ملفات التوصيف، مع تمثيل كل قسم لتصنيف نوع بيانات. تكون الأقسام اختيارية في ملفات التوصيف.

مصطلحات التسمية لعنوان القسم هي:

`[category_name:category_value]`

حيث:

- *category_name* يعد كود متعدد الحروف يمثل تصنيف نوع البيانات
- *category_value* يعد كود متعدد الحروف يمثل قيمة مسموح بها للتصنيف

**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١.

يجب الفصل بين كل اسم تصنيف والقيمة الخاصة به بواسطة

`colon and enclosed by [] square brackets.`

يمكن أن يكون لديك عدة تصنيفات لنوع البيانات من خلال جمع عناوين الأقسام، مما يتم قصر خصائص المتطلبات الرئيسية هذه على فئة التصنيفات المحددة هذه فقط.

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

على سبيل المثال، لتحديد خصائص المتطلبات الرئيسية التي تنطبق على جهاز يقوم بتشغيل نظام تشغيل، SUSE Linux Enterprise Server النسخة 11، Itanium ذو ٣٢-بت:

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

بالنسبة لكل بيئات التشغيل، يمكنك استخدام رمز OR المنطقي | لاستخدام أي من أو تصنيفات نوع البيانات. على سبيل المثال، لتحديد أي من متغيرات بيئة التشغيل الى `True`، ستكون مجموعة عناوين القسم هي:

- أنظمة **UNIX**

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

- أنظمة **Windows**

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

**هام:** يختلف موضع رمز OR المنطقي | ما بين أنظمة Windows و UNIX. بالنسبة لأنظمة UNIX، يتم تضمين فئة عناوين القسم بواسطة فئة واحدة من الأقواس المربعة [] مع فصل كل عنوان للقسم بالرمز. بالنسبة لأنظمة Windows، يقوم الرمز بفصل كل عنوان كامل للقسم بالأقواس المربعة [] المرفقة.

بالنسبة لأنظمة Windows فقط، يمكنك استخدام رمز NOT المنطقي!لاستبعاد تصنيف نوع البيانات. على سبيل المثال، لاستبعاد اختلاف Windows Server 2003 R2، ستكون مجموعة عناوين القسم هي:
[OSType:Windows Server 2003][!OSType:Windows Server 2003 R2]

يوضح الجدول ٦ تصنيفات نوع البيانات المدعمة والقيم المسموح بها المرفقة.

*الجدول ٦ . قيم وتصنيفات نوع البيانات المدعمة*

| تصنيف نوع البيانات | الوصف | القيم المسموح بها |
|---|---|---|
| OSType | نوع نظام التشغيل | • نظام UNIX<br>يشير الى أن كل الخصائص التي توجد في هذا التصنيف عامة لكل بيئات تشغيل UNIX، بما في ذلك AIX و HP-UX و Linux و Solaris، على سبيل المثال:<br>[OSType:UNIX]<br>• نظام AIX<br>يشير الى أن كل الخصائص التي توجد في هذا التصنيف عامة لكل اختلافات نظام تشغيل AIX، على سبيل المثال:<br>[OSType:AIX]<br>• نظام HP-UX<br>يشير الى أن كل الخصائص في هذا التصنيف عامة لكل اختلافات نظام تشغيل HP-UX، على سبيل المثال:<br>[OSType:HP-UX]<br>• نظام LINUX<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام التشغيل Linux، على سبيل المثال:<br>[OSType:LINUX]<br>• نظام RedHat<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لاختلافات نظام التشغيل RedHat Linux، على سبيل المثال:<br>[OSType:RedHat]<br>• RedHatEnterpriseLinuxServer<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل RedHat Enterprise Linux Server، على سبيل المثال:<br>[OSType:RedHatEnterpriseLinuxServer]<br>• نظام SUSE<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل SUSE Linux، على سبيل المثال:<br>[OSType:SUSE]<br>• SUSELinuxEnterpriseServer<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام التشغيل SUSE Linux Enterprise Server، على سبيل المثال:<br>[OSType:SUSELinuxEnterpriseServer]<br>• نظام Solaris<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل Solaris، على سبيل المثال:<br>[OSType:Solaris] |

| | | |
|---|---|---|
| | | • نظام Windows<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل أنظمة تشغيل Windows، على سبيل المثال:<br>`[OSType:Windows]`<br>• Windows 2000 Workstation (النسخة *.5.0)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل Windows 2000، على سبيل المثال:<br>`[OSType:Windows 2000]`<br>• Windows XP Workstation (النسخة *.5.1)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل Windows XP Professional ذو ٣٢-بت، على سبيل المثال:<br>`[OSType:Windows XP]`<br>• Windows XP Workstation (النسخة *.5.2)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل Windows XP Professional ذو ٦٤-بت، على سبيل المثال:<br>`[OSType:Windows XP]`<br>• Windows Vista Workstation (النسخة *.6.0)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل Windows Vista، على سبيل المثال:<br>`[OSType:Windows Vista]`<br>• Windows 7 Workstation (النسخة *.6.1)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل Windows 7، على سبيل المثال:<br>`[OSType:Windows 7]`<br>• Windows 2000 Server (النسخة *.5.0)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لكل اختلافات نظام تشغيل Windows 2000 Server، على سبيل المثال:<br>`[OSType:Windows 2000]`<br>• Windows Server 2003 (النسخة *.5.2)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لاختلافات نظام تشغيل Windows Server 2003، على سبيل المثال:<br>`[OSType:Windows Server 2003]`<br>• Windows Server 2003 R2 (النسخة *.5.2 والنوع الآخر لوصف نظام التشغيل هو R2)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لاختلاف نظام تشغيل Windows Server 2003 R2 فقط، على سبيل المثال:<br>`[OSType:Windows Server 2003 R2]` |

| | | |
|---|---|---|
| | | • Windows Server 2008 (النسخة 6.0.*)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لاختلافات نظام تشغيل Windows Server 2008، على سبيل المثال:<br>[OSType:Windows Server 2008]<br>• Windows Server 2008 R2 (النسخة 6.1.*)<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لاختلاف نظام تشغيل Windows Server 2008 R2 فقط، على سبيل المثال:<br>[OSType:Windows Server 2008 R2]<br>• *<OS_Name_Version>*<br>يشير الى أن كل الخصائص في هذا التصنيف تكون عامة لنسخة نظام التشغيل هذه، على سبيل المثال:<br>[OSType:RedHatEnterpriseLinuxServer4.2]<br>**ملاحظة:** يسمح لحرف الابدال الخاص، *، بتحديد عدة نسخ. |
| OSArch | هيكل بناء نظام التشغيل | • ٣٢-بت، على سبيل المثال:<br>[OSArch:32-bit]<br>• ٦٤-بت، على سبيل المثال:<br>[OSArch:64-bit] |
| CPU | اسم مجموعة المشغل العام | Itanium، على سبيل المثال:<br>[CPU:Itanium] |
| CPUArch | هيكل بناء المشغل | هيكل بناء المشغلات PowerPC® و Power Architecture® ذات ٦٤-بت، وهي:<br>• ppc4<br>• POWER4<br>• POWER5<br>• POWER6®<br>• POWER7®<br><br>على سبيل المثال:<br>[CPUArch:ppc4] |
| *@<EnvVar_Name>* | متغير بيئة التشغيل للمنتج | يتم الالتزام بقواعد هذا المنتج، على سبيل المثال:<br>[@TPAE_DB_SERVER:True] |

### مثال لملف توصيف Windows الذي يستخدم الأقسام

يقوم هذا المثال باستخدام الأقسام لتصنيف خصائص المتطلبات الرئيسية لأي جهاز لنظام Windows ثم الأجهزة التي تقوم بتشغيل نسخة معينة لنظام Windows.

```
#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+
```

عند تشغيل الأداة Prerequisite Scanner، فانها تقوم باجراء عملية مسح والتحقق من خصائص المتطلبات الرئيسية المختلفة بناءا على نظام التشغيل والنسخة التي تم تركيبها على الجهاز.

على سبيل المثال، يوضح الجدول ٧ الأقسام المختلفة التي تحتوي على خصائص المتطلبات الرئيسية التي تم التحقق منها بناءا على المثال.

*الجدول ٧. الأقسام التي تم اجراء عملية مسح لها من ملف التوصيف لنظام Windows*

| بيئة التشغيل أو نظام التشغيل | الأقسام ذات خصائص المتطلبات الرئيسية |
|---|---|
| جهاز ذو نظام Windows XP وأعلى | [OSType:Windows] |
| جهاز ذو نظام Windows Vista فقط | [OSType:Windows]<br>[OSType:Windows Vista] |

## مثال لملف التوصيف لنظام UNIX الذي يستخدم الأقسام

يحتوي هذا المثال على خصائص المتطلبات الرئيسية لكل بيئات التشغيل وبيئات التشغيل المنفردة ونسخ أنظمة التشغيل لمنتج معين.

```
# 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.ia32el=ia32el-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:filesizelimit]unlimited
os.ulimit=[type:filedescriptorlimit]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+
```

عند تشغيل الأداة Prerequisite Scanner، فانها تقوم باجراء عملية مسح والتحقق من خصائص المتطلبات الرئيسية المختلفة بناءا على نظام التشغيل والنسخة التي تم تركيبها على الجهاز.

على سبيل المثال، يوضح الجدول ٨ الأقسام المختلفة التي تحتوي على خصائص المتطلبات الرئيسية التي تم التحقق منها بناءا على المثال.

*الجدول ٨. الأقسام التي تم اجراء عملية مسح لها من ملف التوصيف لنظام UNIX*

| أنظمة التشغيل والنسخ | الأقسام ذات خصائص المتطلبات الرئيسية |
|---|---|
| جهاز به SUSE Linux Enterprise Server 11 ذو ٦٤-بت | [OSType:UNIX]<br>[OSType:LINUX]<br>[OSType:LINUX] [CPUArch:ppc64]<br>[OSType:SUSE Linux Enterprise Server 11]<br>[OSArch:64-bit] |
| جهاز به Red Hat Enterprise Linux Server 6.3 | [OSType:UNIX]<br>[OSType:LINUX]<br>[OSType:RedHat]<br>[OSType:RedHatEnterpriseLinuxServer]<br>[OSType:RedHatEnterpriseLinuxServer6.*] |
| جهاز به SUSE Linux Enterprise Server 11 ومتغير بيئة التشغيل `@TPAE_DB_Server` محدد بالقيمة `true` | [OSType:UNIX]<br>[OSType:LINUX]<br>[OSType:SUSELinuxEnterpriseServer11] [@TPAE_DB_Server:True] |
| جهاز به AIX 5.3.0.0 ومتغيرات بيئة تشغيل `@TPAE_DB_FEATURE` أو `@TPAE_DIR_FEATURE` محددة بالقيمة `True` | [OSType:UNIX]<br>[OSType:AIX]<br>[OSType:AIX5.3.0.0] [@TPAE_DB_FEATURE:True\|@TPAE_DIR_FEATURE:True] |

### أدوات تجميع Prerequisite Scanner

تقوم أدوات تجميع IBM Prerequisite Scanner بتجميع البيانات الفعلية عن بيئة التشغيل الحالية بناءا على خصائص المتطلبات الرئيسية التي تم تحديدها للمنتجات التي سيتم تركيبها. تقوم أدوات التجميع بالحصول على البيانات من خلال الكود الأصلي. يمكن أن تكون البيانات بيانات عامة، مثل سرعة المشغل و RAM، بيانات البرامج التي تم تركيبها، بيانات نظام التشغيل، بيانات المستخدم، شبكة الاتصال، بيانات امكانية الاتصال. وتكون أدوات التجميع قابلة للامتداد أيضا، بحيث يمكنك تكوين أدوات تجميع مهيأة للحصول على القيم الفعلية لخصائص المتطلبات الرئيسية المهيأة.

تقوم أداة Prerequisite Scanner باستخدام أدوات التجميع باللغات التالية بناءا على بيئة التشغيل الخاصة بك:

- نظام Windows: VBScript باللاحقة `.vbs`.
- نظام UNIX: Shell باللاحقة `.sh` أو بدون لاحقة

**ملاحظة:** لا يمكنك تشغيل البرامج النصية لنظام UNIX على أنظمة Windows حتى اذا كان قد تم تركيب بيئات تشغيل مماثلة لنظام UNIX على أجهزة Windows، على سبيل المثال، Cygwin.

#### أدوات التجميع لأنظمة Windows

يتم تشغيل أدوات تجميع VBScript لأنظمة Windows في بيئة تشغيل Windows Script Host. وهي تقوم باستخدام Component Object Model للتوصل الى عناصر بيئة تشغيل Windows، على سبيل المثال، `FileSystemObject` و `TextStream`.

يقوم Prerequisite Scanner بتشغيل أدوات تجميع VBScript للحصول على القيم الفعلية لخصائص المتطلبات الرئيسية لبيئة تشغيل Windows. يمكن أن تحصل كل أداة تجميع على البيانات الخاصة بواحدة أو أكثر من خصائص المتطلبات الرئيسية.

بالنسبة لكل خاصية للمتطلبات الرئيسية في أداة التجميع VBScript، تقوم أداة التجميع بكتابة اسم خاصية المتطلبات الرئيسية والقيمة الفعلية الخاصة بها كمخرجات قياسية. تقوم Prerequisite Scanner بكتابة هذه المخرجات القياسية في ملف نصي مؤقت، الذي يمثل، `localhost_hw.txt`.

يمكنك تكوين أدوات تجميع VBScript العامة المهيأة لتجميع بيانات لخصائص المتطلبات الرئيسية التي يتم تطبيقها على أي من البرامج ونسخ البرامج. يمكنك أيضا تكوين خصائص مهيأة محددة للبرنامج لتجميع البيانات التي تنطبق على برنامج محدد ونسخ معينة للبرنامج.

عند تشغيل أداة Prerequisite Scanner، يتم تشغيل أدوات التجميع بالترتيب التالي: أدوات تجميع VBScript المعرفة مسبقا؛ أدوات تجميع VBScript العامة المهيأة في الدليل *ips_root*`/lib`؛ أدوات تجميع VBScript المهيأة الخاصة بالمنتج بالبحث عن الملف *product_code*[_<*version*>]`.vbs` في الدليل *ips_root*`/Windows`.

على سبيل المثال، يعد الملف `env.tcrhome.vbs` أداة تجميع مهيأة تقوم بالتحقق من متغير بيئة التشغيل للدليل الرئيسي الى Tivoli Common Reporting. ويتم تخزينه في الدليل *ips_root*`/lib`.

يجب أن تلتزم أدوات تجميع VBScript بالقواعد التالية:

- مصطلح التسمية لملف أداة تجميع VBScript العام المهيأ

  وهو يحتوي على خاصية المتطلبات الرئيسية التي سيتم اتاحتها لأي منتج ونسخة منتج، والذي يمثل، كل ملفات التوصيف:

  *prefix_identifier*`.]property_name.vbs`

  حيث:

  – *prefix_identifier* هو بادئة كود التعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض الفئات سابقة التعريف، على سبيل المثال، `env`.

  – *property_name* هو اسم الخاصية للمتطلبات الرئيسية، على سبيل المثال، `tcrhome`.

  يتم تخزين هذا النوع من أدوات تجميع VBScript في دليل *ips_root*`/lib`.

- مصطلح التسمية لملف أداة تجميع VBScript المهيأ الخاص بالمنتج

  وهو يحتوي على الخصائص التي سيتم اتاحتها لمنتج معين ونسخ المنتج، والذي يمثل، ملف توصيف واحد:

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

  حيث:

  – *product_code*

  يعد المتغير المستخدم لتمثيل كود المنتج على نظام Windows أو نظام UNIX. تقوم أكواد المنتج بتعريف المنتج، وبيئة تشغيل منفردة مثل Windows و AIX و HP-UX و Linux و Solaris، واختياريا نسخة نظام التشغيل المدعمة بواسطة هذا المنتج. ويتم تخزينها في الملف `codename.cfg`. يحتوي أي منتج يدعم عدة بيئات تشغيل على عدة أكواد للمنتج مع كلا منهما، حيث تقوم بتعريف المنتج وبيئة التشغيل ونسخة نظام التشغيل وفقا للمطلوب.

  – <*version*> هو كود مكون من ٨-أرقام يمثل النسخة، والاصدار، والتعديلات، والمستوى، مع رقمين لكل من أجزاء الكود؛ على سبيل المثال 7.3.21 هو 07032100.

  يتم تخزين هذا النوع من أدوات تجميع VBScript في دليل *ips_root*`/Windows`.

- المخرجات القياسية لكل خاصية للمتطلبات الرئيسية هي كما يلي:

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

  – *property_name* الذي يمثل خاصية المتطلبات الرئيسية كما تم كتابتها في ملف التوصيف، على سبيل المثال، `env.tcrhome`

  – *var_for_value*، أي أن المتغير VBScript للقيمة الفعلية التي ستقوم أداة التجميع بالحصول عليها لخاصية المتطلبات الرئيسية.

  على سبيل المثال، تقوم المخرجات القياسية التالية بكتابة خاصية المتطلبات الرئيسية لمتغير بيئة التشغيل الرئيسي Tivoli Common Reporting والقيمة الفعلية الخاصة به:

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

### أدوات التجميع لأنظمة UNIX

يتم تشغيل أدوات التجميع لأنظمة UNIX في بيئة تشغيل النظام الرئيسي Shell المتعلقة لأي من AIX أو HP-UX أو Linux أو Solaris. وهي تقوم باستخدام الأوامر والاختيارات الخاصة ببيئة التشغيل هذه للتوصل الى عناصر بيئة التشغيل الرئيسية.

تقوم كل أداة تجميع لنظام UNIX بالحصول على بيانات خاصية المتطلبات الرئيسية أو خاصية متطلبات رئيسية بأنواع فرعية معرفة مسبقا. تقوم أداة التجميع بكتابة نتيجة عملية التحقق لخاصية المتطلبات الرئيسية كمخرجات قياسية. وتقوم Prerequisite Scanner بكتابة هذه المخرجات القياسية في ملف نصي مؤقت.

يمكنك تكوين أدوات تجميع UNIX مهيأة لتجميع البيانات لخصائص متطلبات رئيسية مهيأة. يتم استدعاء كل أداة تجميع، معرفة مسبقا أو مهيأة، في الملف *ips_root*/UNIX_Linux/packageTest.sh.

عند تشغيل Prerequisite Scanner، يتم تشغيل أدوات التجميع بالترتيب التالي: أدوات تجميع معرفة مسبقا ذات plug_ في اسم الملف في الدليل *ips_root*/lib؛ أدوات تجميع معرفة مسبقا في الدليل *ips_root*/UNIX_Linux؛ وأدوات تجميع UNIX مهيأة في الدليل *ips_root*/UNIX_Linux.

على سبيل المثال، يعد الملف installedSoftware.TCR.version أداة تجميع مهيأة تقوم بالحصول على نسخة Tivoli Common Reporting التي تم تركيبها على الجهاز. ويتم تخزينها في الدليل *ips_root*/UNIX_Linux.

يجب أن تلتزم أدوات تجميع UNIX بالقواعد التالية:

- طريقة التسمية لملف أداة تجميع UNIX المهيأ بدون لاحقة الملف:

  [*prefix_identifier*.]*property_name*

  حيث:

  - *prefix_identifier* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض الفئات سابقة التعريف، على سبيل المثال، installedSoftware.
  - *property_name* هي اسم خاصية المتطلبات الرئيسية، على سبيل المثال، TCR.version.

  قم بتخزين أداة التجميع في الدليل *ips_root*/UNIX_Linux. وتأكد من أنها لا تحتوي على لاحقة الملف.

- المخرجات القياسية لخاصية المتطلبات الرئيسية التي تقوم بارجاع القيمة الفعلية لخاصية المتطلبات الرئيسية اذا كانت رقم صحيح أو مجموعة حروف؛ على سبيل المثال، نسخة البرامج أو مساحة القرص المتاحة لنظام الملفات الذي تم تركيبه. تبادليا، يمكن أن تقوم بارجاع "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
```

- الكود لاستدعاء وتشغيل أداة التجميع في البرنامج النصي *ips_root*/UNIX_Linux/packageTest.sh.

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

echo "\`wrlTrace "Starting" "installedSoftware.TCR.version"\`"
>>/tmp/prs.check
```

```
echo "\`wrlTrace "Executing" "installedSoftware.TCR.version"\`"
>>/tmp/prs.check
echo "\`wrlDebug "Starting" "installedSoftware.TCR.version"\`"
>>/tmp/prs.check
echo "\`wrlDebug "Expected" "ExpValue" \`" >>/tmp/prs.check

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

### أدوات تقييم Prerequisite Scanner

أدوات التقييم IBM Prerequisite Scanner هي البرامج النصية التي تقوم بمقارنة البيانات الفعلية من أدوات التجميع والبيانات المتوقعة لنفس الخصائص التي توجد في ملفات التوصيف. عمليات التقييم يمكن أن تكون: متعلقة ببيئة التشغيل؛ بناءا على معاملات بسيطة، مثل أقل من أو يساوي أو أكبر من؛ وبناءا على ما اذا كان قد تم تركيب الخاصية أو ما اذا كانت موجودة أو متاحة. ويمكنها أيضا التحقق مما اذا كان يتم استخدام المنافذ أو اذا كانت متاحة ومن حالة امكانية الاتصال بالجهاز. يمكنك تكوين أو تحرير أدوات التحقق.

تقوم أداة Prerequisite Scanner باستخدام اللغات التالية بناءا على بيئة التشغيل الخاصة بك:

- نظام Windows: VBScript مع اللاحقة vbs.
- نظام UNIX: shell مع اللاحقة sh.

**ملاحظة:** لا يمكنك تشغيل البرامج النصية لنظام UNIX على أنظمة Windows حتى اذا كان قد تم تركيب بيئة تشغيل مماثلة لنظام UNIX على أجهزة Windows، على سبيل المثال، Cygwin.

يمكنك تخزين أدوات التقييم في *ips_root*/*OS*، حيث *OS* هو اسم نظام التشغيل، على سبيل المثال، Windows أو UNIX_Linux.

يجب أن تلتزم ملفات أداة التقييم بالقواعد التالية:

- طريقة التسمية لاسم الملف:

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

  حيث:

  - *prefix_identifier* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض التصنيفات سابقة التعريف.
  - *property_name* هو اسم خاصية المتطلبات الرئيسية.
  - *suffix_identifier* هو كود تعريف اختياري للنوع الفرعي لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٤.

- يمكنك اختياريا امرار اثنين من معاملات المدخلات الى البرنامج النصي لعمليات التقييم المركبة:
  - *expected_value*، التي تمثل، القيمة المتوقعة لخاصية المتطلبات الرئيسية التي تم تحديدها في ملف التوصيف.
  - *actual_value*، التي تمثل، القيمة الفعلية التي تقوم أداة التجميع باكتشافها لخاصية المتطلبات الرئيسية على الجهاز.

- المخرجات المعتادة هي كما يلي:
  - "PASS" عندما تكون القيمة المتوقعة لخاصية المتطلبات الرئيسية أكبر من أو تساوي القيمة الفعلية للخاصية.
  - "FAIL" عندما تكون القيمة المتوقعة لخاصية المتطلبات الرئيسية لا تساوي القيمة الفعلية للخاصية.

### نسق المخرجات

يقدم IBM Prerequisite Scanner مخرجات للشاشة ونسق الملف القابلة للقراءة التالية: مخرجات لواجهة تعامل سطر الأمر، ملفات السجل الخاصة بتصحيح وتتبع الأخطاء، ملفات نصية و XML للنتائج.

تقوم أداة Prerequisite Scanner بحفظ نتائج عملية المسح وملفات السجل في الدليل *ips_output_dir*. يمكنك تحديد هذا الدليل باستخدام معامل المدخلات **outputDir** عند تشغيل الماسح الضوئي. اذا لم تقم بتحديد هذا المعامل، فان مكان المخرجات المفترض هو *ips_root*.

**ملاحظة:** يقوم Prerequisite Scanner بتكوين الملفات المؤقتة أثناء التنفيذ، ولكن يتم حذف هذه الملفات قبل اتمام الماسح الضوئي لتنفيذها. توجد هذه الملفات المؤقتة في الدليل الفرعي *ips_output_dir*/temp. كما يقوم الماسح الضوئي بحذف الدليل الفرعي *ips_output_dir*/temp، ما لم يكن الدليل الفرعي يتضمن ملفات التتبع وتصحيح الأخطاء التي تم تكوينها في أنظمة UNIX فقط.

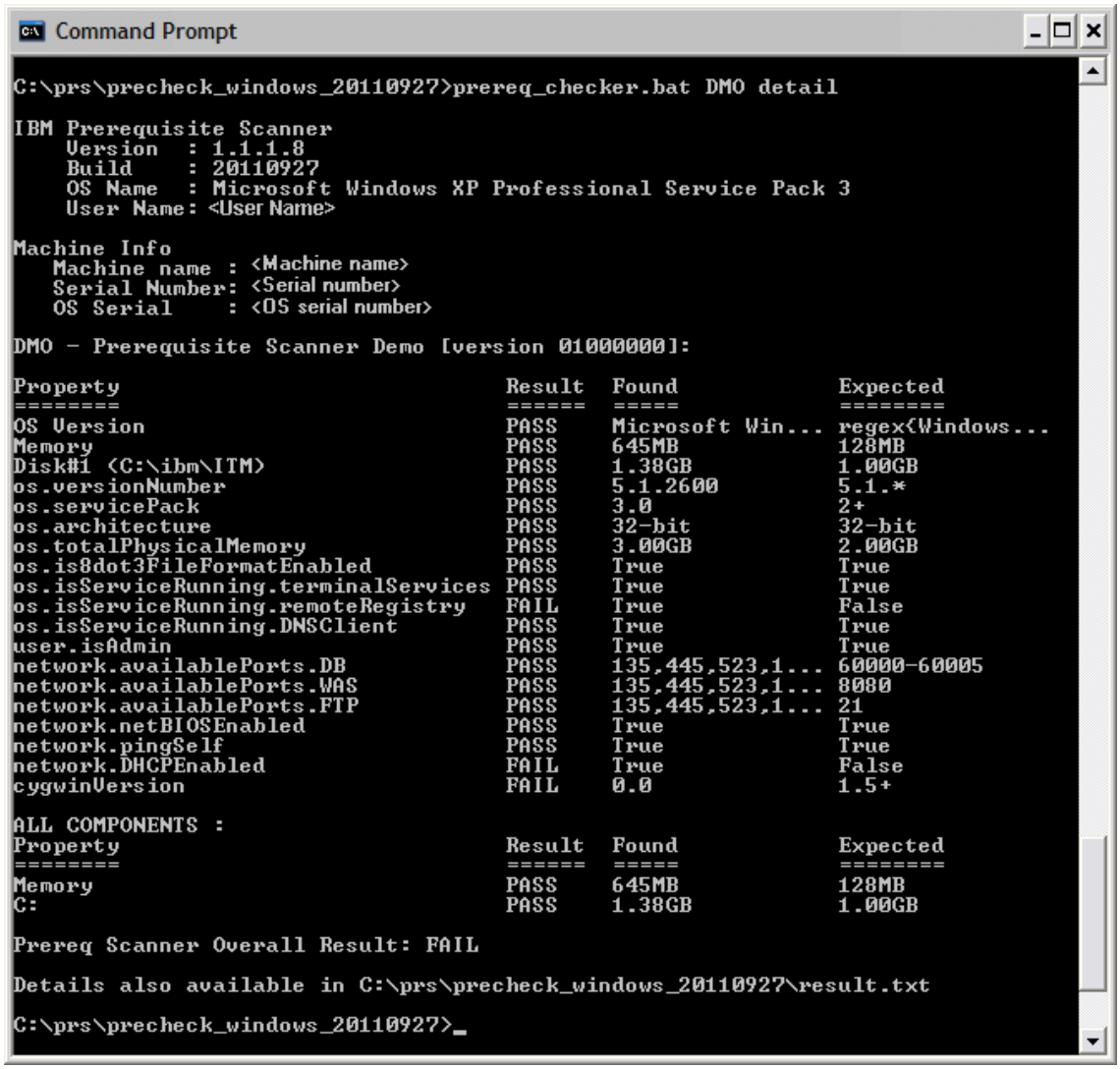
يمكنك أيضا استخدام المعامل لتحديد مكان، اذا قمت باختيار تشغيل Prerequisite Scanner من CD أو DVD أو مشغل شبكة اتصال للقراءة فقط.

**هام:** اذا كان دليل المخرجات غير موجود، ستقوم أداة Prerequisite Scanner بتكوين الدليل. يجب أن يكون لديك صلاحيات بالكتابة لتكوين أو الكتابة في دليل المخرجات الذي يقوم Prerequisite Scanner بحفظ الملفات به.

#### مخرجات واجهة تعامل سطر الأمر

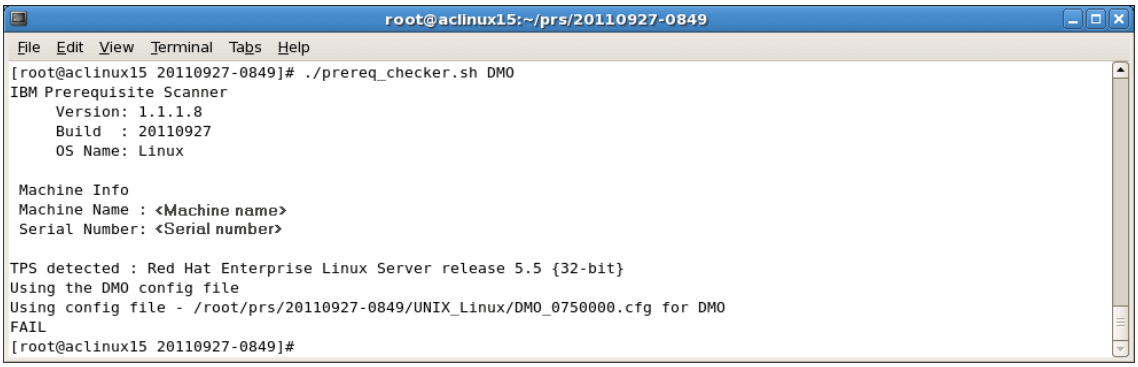
عند تشغيل البرنامج النصي Prerequisite Scanner وتحديد المعامل **detail** الاختياري، ستقوم أداة Prerequisite Scanner بعرض النتائج التفصيلية لعملية المسح في واجهة تعامل سطر الأمر. تحتوي النتائج التفصيلية على:

- نسخة Prerequisite Scanner
- نسخة نظام التشغيل التي تم تشغيل الماسح الضوئي عليها
- نوع عملية المسح والسيناريو
  - اجراء مسح للمتطلبات الرئيسية: `Scenario: Prerequisite Scan`
- اسم المنتجات أو المكونات التي تم تشغيل عمليات التحقق من المتطلبات الرئيسية عليها
- بالنسبة لكل خاصية للمتطلبات الرئيسية: سيتم التحقق من اسم الخاصية، نتيجة PASS أو FAIL، القيمة الفعلية، القيمة المتوقعة
- بالنسبة لكل المكونات: سيتم التحقق من اسم الخاصية العامة، نتيجة PASS أو FAIL، القيمة الفعلية، القيمة المتوقعة
- نتيجة PASS أو FAIL العامة، حيث سينتج عن حدوث أي فشل لعملية تحقق منفردة فشل عملية المسح بالكامل




الشكل ١. المخرجات لواجهة تعامل سطر الأمر بأنظمة Windows

اذا لم تقم بتحديد المعامل **detail**، سيقوم الماسح الضوئي بعرض نتيجة PASS أو FAIL بالشاشة.




الشكل ٢. المخرجات لواجهة تعامل سطر الأمر بأنظمة UNIX

Prerequisite Scanner تقوم بانتاج أكواد مرتجعة تتوقف على نتائج عملية المسح وما اذا كان يجب الخروج بسبب الأخطاء أم لا. يتم كتابة الأكواد المرتجعة هذه في ملفات السجل. على سبيل المثال، اذا فشل Prerequisite Scanner في تشغيل عملية التحقق لأنه لا يمكن قراءة ملف التوصيف، وستقوم بتكوين كود 2 الناتج.

**جمع خصائص المتطلبات الرئيسية للذاكرة ومساحة القرص**

يمكنك تشغيل Prerequisite Scanner للتحقق في وقت واحد من المتطلبات الرئيسية لواحد أو أكثر من المنتجات أو المكونات، عند تحديد عدة أكواد للمنتج كمعاملات للمدخلات. تقوم أداة Prerequisite Scanner بجمع نتائج عمليات التحقق من المتطلبات الرئيسية للذاكرة ومساحة القرص في الأقسام المجمعة التالية للمخرجات، اذا كان قد تم تحديد خصائص المتطلبات الرئيسية المرفقة في أي من ملفات التوصيف:

- على أنظمة UNIX، في قسم `TOTAL ALL SPECIFIED COMPONENTS`
- على أنظمة Windows، في قسم `ALL COMPONENTS`

خصائص المتطلبات الرئيسية العامة هي كما يلي:

- اجمالي حجم الذاكرة الفعلية المتاحة حاليا في بيئة التشغيل المستهدفة، والذي يمثل:

  `Memory`

- مساحة القرص لأنظمة الملفات لخصائص المتطلبات الرئيسية التالية:

| بيئة التشغيل | خصائص المتطلبات الرئيسية |
|---|---|
| نظام UNIX و Linux | • `os.space.home`<br>• `os.space.opt`<br>• `os.space.tmp`<br>• `os.space.usr`<br>• `os.space.var` |
| نظام Windows | `Disk` |

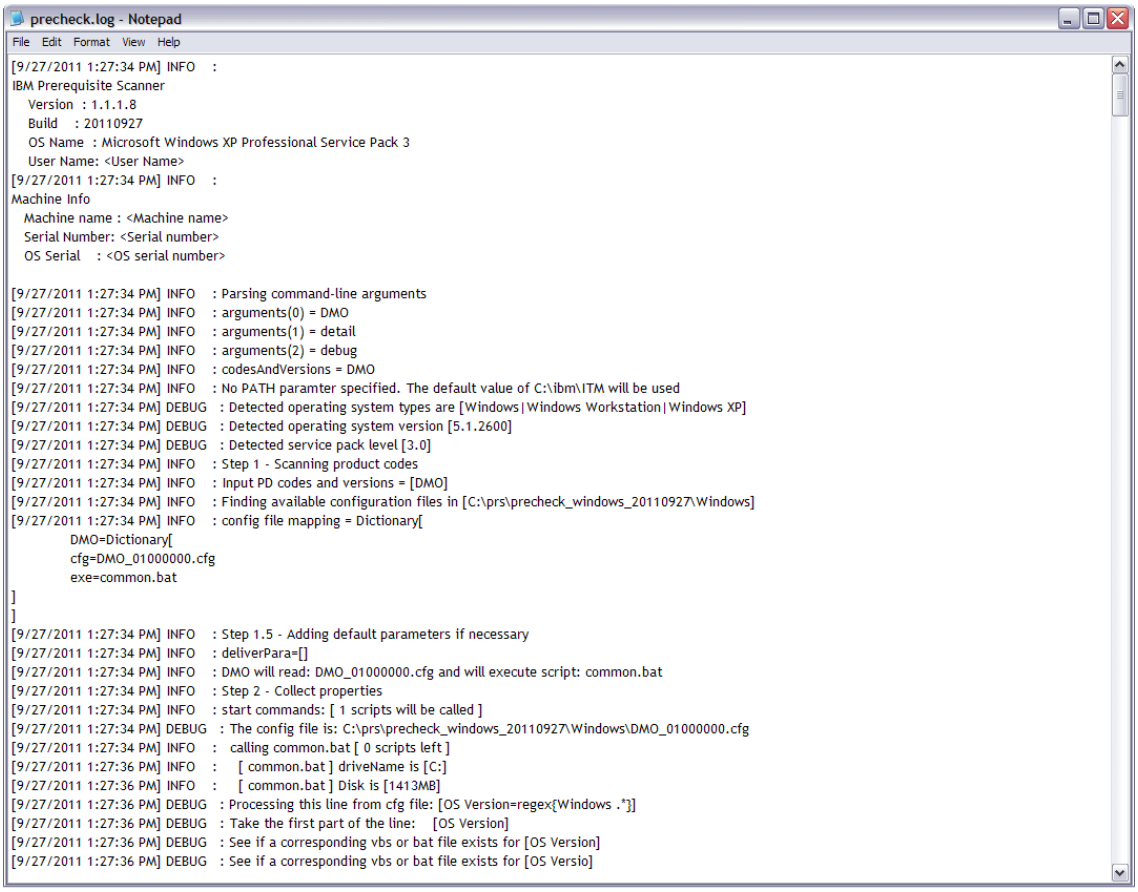
**ملاحظة:** اذا لم يتم تعريف `opt` و `usr` و `var` كأنظمة للملفات على الحاسب المستهدف، لن يقوم Prerequisite Scanner بعرض القيم المتوقعة والنتائج التي تم ارجاعها لخصائص المتطلبات الرئيسية هذه في القسم المجمع.

لن يقوم Prerequisite Scanner بعرض القسم المجمع، اذا لم تكن خصائص المتطلبات الرئيسية للذاكرة أو مساحة القرص موجودة في ملفات التوصيف.

ستقوم أداة Prerequisite Scanner بمعالجة المقارنة ويتم عرض قيم مساحة القرص في القسم المجمع من نتائج عملية المسح بشكل مختلف عن القسم الرئيسي. أنظر "وحدات القياس في المخرجات".

**مخرجات ملف سجل تصحيح الأخطاء بأنظمة Windows**

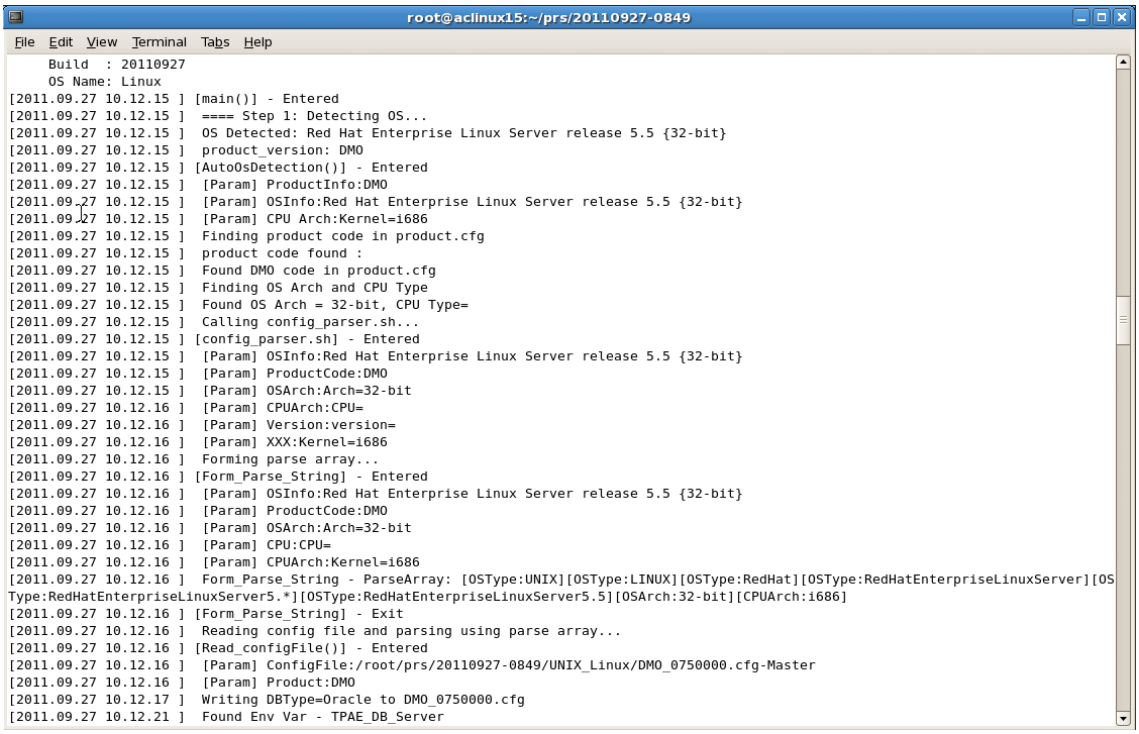
يقوم Prerequisite Scanner باخراج معلومات التشغيل، ورسائل الخطأ والتحذير، ونتائج عملية المسح في ملف *ips_output_dir*`/precheck.log`. عندما تقوم بتشغيل البرنامج Prerequisite Scanner وتحديد المعامل **`debug`** الاختياري، سيقوم Prerequisite Scanner باخراج المزيد من رسائل تصحيح الأخطاء في هذا الملف.




الشكل ٣. ملف precheck.log

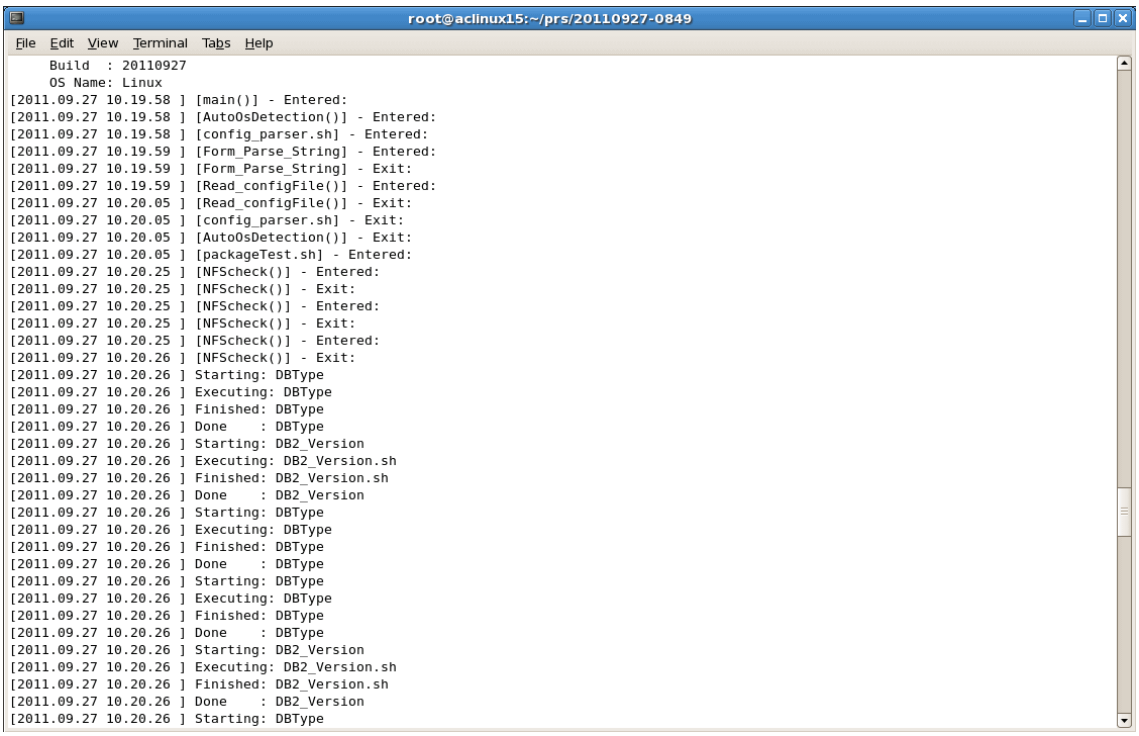
### مخرجات ملف سجل التتبع وتصحيح الأخطاء في أنظمة UNIX

عندما تقوم بتشغيل Prerequisite Scanner وتحديد المعامل **debug** الاختياري، ستقوم أداة Prerequisite Scanner باخراج معلومات التشغيل التفصيلية، ورسائل التحذير والخطأ، ونتائج عملية المسح في ملف *ips_output_dir*/temp/prs.debug.




الشكل ٤. ملف prs.debug على أنظمة UNIX

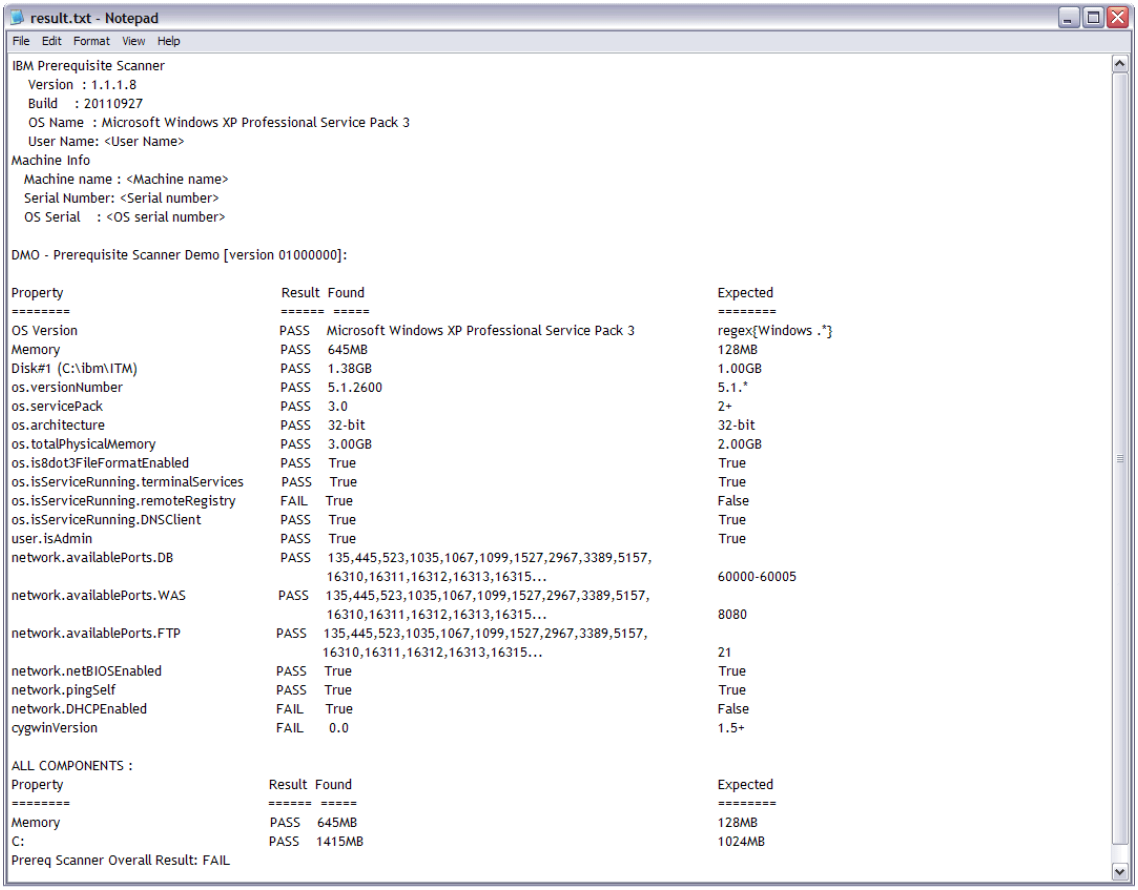
عندما تقوم بتشغيل Prerequisite Scanner وتحديد المعامل **`trace`** الاختياري، ستقوم أداة Prerequisite Scanner باخراج معلومات التتبع في ملف *ips_output_dir*`/temp/prs.trc`.




الشكل ٥. ملف prs.trc على أنظمة UNIX

### مخرجات الملف النصي

تقوم أداة Prerequisite Scanner باخراج نتائج عملية المسح التفصيلية في ملف `result.txt/`*ips_output_dir*. سيقوم بحفظ النتائج في الملف النصي بصرف النظر سواء قمت بتحديد المعامل **`detail`** أم لا.




الشكل ٦. ملف result.txt على أنظمة Windows

```
[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.*}"
                                                                                                regex{AIX.*}
                                                                                                regex{Solaris.*}
os.lib.libstdc++                        PASS            /usr/lib/gcc/i386-redhat-linux/4.1.1/libstdc++.solibstdc++

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"
                                                                                                unit:MB]1.0
os.space.usr                            PASS            "38GB"                                   "[dir:root=/usr/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
```

الشكل ٧. ملف result.txt على أنظمة UNIX

### مخرجات ملف XML

تقوم أداة Prerequisite Scanner باخراج نتائج عملية المسح التفصيلية في ملف *ips_output_dir*`/result.xml` عند تحديد معامل المدخلات **`xmlResult`** الاختياري. يمكنك استخدامه لتوجيه الأداة الى اخراج النتائج في ملف XML للنتائج بالاضافة الى ملف نتائج الاختبار العادي. سيقوم بحفظ النتائج في ملف XML بصرف النظر سواء قمت بتحديد المعامل **`detail`** أم لا.

```
<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>
```

الشكل ٨. ملف result.XML على أنظمة Windows

يستطيع المطورين استخدم أداة Prerequisite Scanner Java™ Developer لاجراء تحليل لغوي وقراءة ملف XML.

**وحدات القياس في المخرجات**

ستقوم أداة Prerequisite Scanner بمعالجة المقارنة ويتم عرض قيم مساحة القرص في القسم المجمع من نتائج عملية المسح بشكل مختلف عن القسم الرئيسي.

في القسم الرئيسي لنتائج عملية المسح، تقوم أداة Prerequisite Scanner بالتعامل مع قيم مقارنة وعرض مساحة القرص كما يلي:

| بيئة التشغيل | خصائص المتطلبات الرئيسية |
|---|---|
| نظام UNIX و Linux | اذا كانت القيمة الفعلية أكبر من ١٠٢٤ ميجابايت، ستقوم أداة Prerequisite Scanner بتحويل وارجاع القيمة بالميجابايت؛ خلاف ذلك يتم ارجاع القيمة بالميجابايت. |
| نظام Windows | تقوم أداة Prerequisite Scanner باستخدام وحدة القيمة المتوقعة في ملف التوصيف كوحدة للمقارنة والعرض. حيث تقوم بتحويل القيمة الفعلية الى هذه الوحدة اذا تطلب الأمر. |

في القسم المجمع لنتائج المسح، تقوم أداة Prerequisite Scanner بالتعامل مع قيم مقارنة وعرض مساحة القرص كما يلي:

| بيئة التشغيل | خصائص المتطلبات الرئيسية |
|---|---|
| نظام UNIX و Linux | اذا كانت القيمة الفعلية أكبر من ١٠٢٤ ميجابايت، ستقوم أداة Prerequisite Scanner بتحويل وارجاع القيمة بالميجابايت؛ خلاف ذلك يتم ارجاع القيمة بالميجابايت. |
| نظام Windows | تقوم أداة Prerequisite Scanner بتنفيذ عملية مقارنة مساحة القرص بناءا على وحدات الميجابايت. بالنسبة لكل ملف توصيف يحتوي على خاصية المتطلبات الرئيسية `Disk`، ستقوم أداة Prerequisite Scanner بتحويل القيم الفعلية والمتوقعة الى ميجابايت وتنفيذ عملية المقارنة. لأغراض العرض، اذا كانت القيمة الفعلية المجمعة أكبر من ١ جيجابايت، ستقوم أداة Prerequisite Scanner بارجاع القيمة الفعلية بالجيجابايت مع تحديد الدقة ٢؛ خلاف ذلك، يتم ارجاع القيمة بالميجابايت. |

## مجموعة أدوات Prerequisite Scanner Java Developer

مجموعة أدوات Prerequisite Scanner Java Developer هي مجموعة من واجهات تعامل البرمجة التي تسمح لك، كمطور، باجراء تحليل لغوي وقراءة محتويات ملف XML للنتائج بشكل برمجي للاحتياجات الخاصة بك؛ على سبيل المثال، اجراء تحليل لغوي لنتائج عملية المسح لاستخدامها في برنامج التركيب الخاص بك.

تقدم مجموعة الأدوات مجموعات البرامج التالية:

- com.ibm.prs.common.exception

  حيث تحتوي على فئة `PRSApiException` التي تقدم طرق لتقديم الأخطاء الاستثنائية في واجهة تعامل البرمجة لاستعلام XML.

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

  حيث تحتوي على واجهة تعامل `PRSXmlResultReader` التي تقوم بتعريف واجهة تعامل برمجة استعلام XML لملف نتائج XML.

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

حيث تحتوي على فئة `PRSXmlResultReaderImpl` التي تقوم باعداد `PRSXmlResultReader`.

تستطيع أداة Prerequisite Scanner التحقق من صلاحية النسق والهيكل لملف XML Schema الى *ips_root*`/PRSResults.xsd`.

يكون Javadoc متاحا لمجموعة الأدوات في الدليل *ips_root*`/api/javadoc`.

## ملف XML Schema لملف نتائج XML

يقدم Prerequisite Scanner ملف XML Schema يمكن التحقق من صلاحية ملف نتائج XML له.

تحتوي ملفات XML Schema على العناصر التالية التي تمثل الأقسام:

- `PRSInfo` لادارة تفاصيل Prerequisite Scanner
- `MachineInfo` لادارة المعلومات عن بيئة التشغيل المستهدفة التي يتم اجراء عملية المسح عليها
- `UserInfo` لادارة المعلومات عن المستخدم الذي قام ببدء الاتصال والذي يقوم باجراء عملية المسح
- `ScenarioInfo` لادارة المعلومات عن نوع عملية المسح والسيناريو
- `ProductInfo` لادارة المعلومات عن المنتج أو المكون أو ملف التوصيف الخاص به

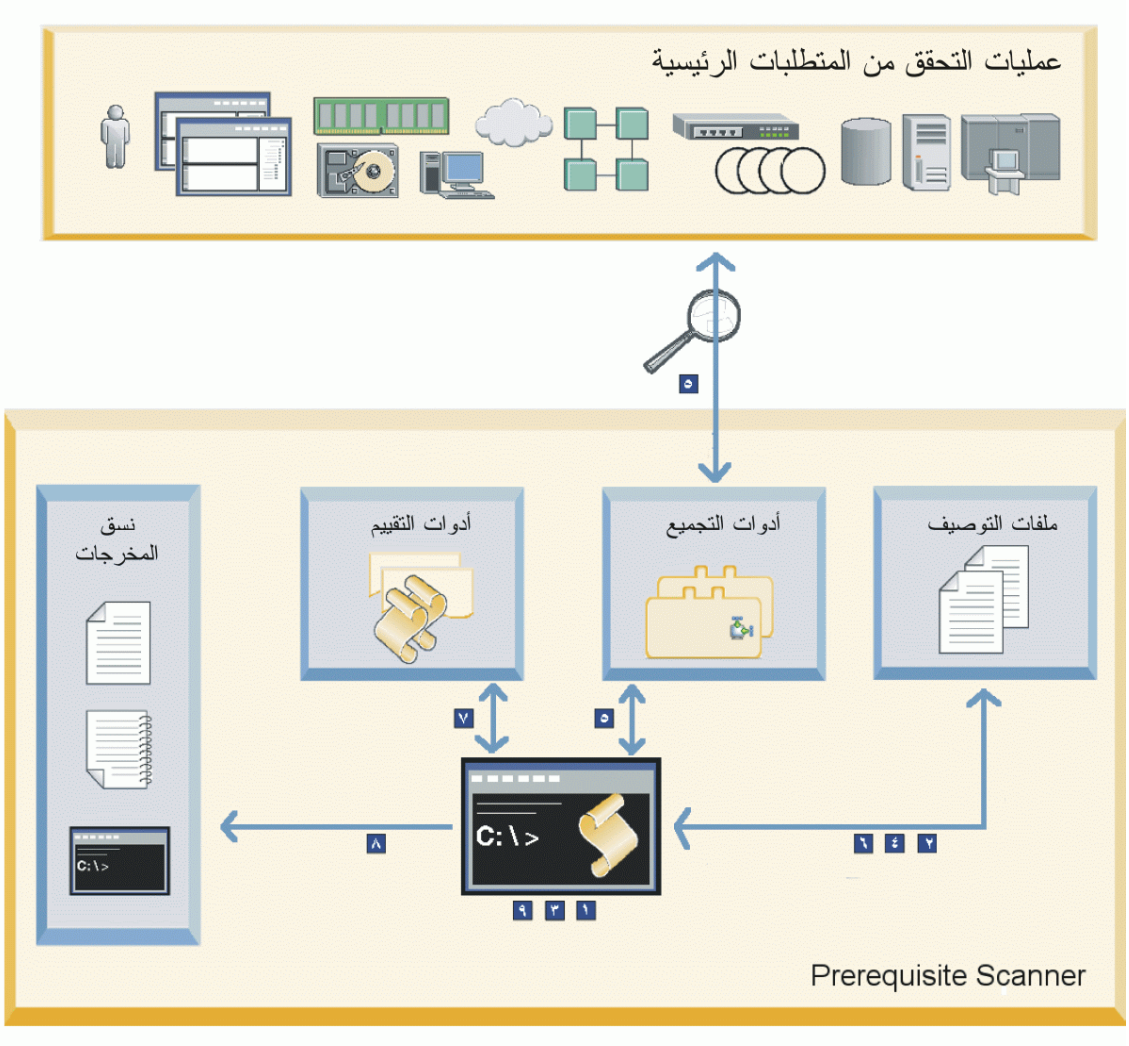
- `DetailedResults` لادارة نتائج عملية المسح لكل مجموعة من خصائص المتطلبات الرئيسية لمنتج أو مكون كما تم جمعها بواسطة `DetailedProductResultsElement`
- `AggregateResults` لادارة نتائج عملية المسح التي تم تجميعها لمساحة القرص والذاكرة
- `OverallResult` لادارة نتيجة PASS أو FAIL العامة للمسح الضوئي

اسم ومكان ملف XML Schema هو: *ips_root*`/PRSResults.xsd`

باعتبارك مطور أو ناشر، يمكنك استدعاء الطرق من واجهة تعامل برمجة XML للاستعلام للتحقق من صلاحية ملف XML للنتائج. يكون Javadoc متاحا لمجموعة الأدوات التي توجد في الدليل *ips_root*`/api/javadoc`.

## عملية المسح

عند تشغيل أداة IBM Prerequisite Scanner، فانها تقوم بتنفيذ مجموعة من المهام في كل مرحلة لعملية المسح. ويقوم المستخدم بفتح واجهة تعامل سطر الأمر وتشغيل البرنامج النصي Prerequisite Scanner مع مجموعة من معاملات المدخلات متضمنة كود المنتج.




الشكل ٩. هيكل بناء Prerequisite Scanner وعملية المسح

تم تلخيص عملية المسح في الشكل ٩ كما يلي:

١. تقوم أداة Prerequisite Scanner بالتحقق من نسق معاملات المدخلات التي تم امرارها الى الماسح الضوئي.
٢. يقوم الماسح الضوئي بتحديد ما اذا كان كود المنتج الذي تم امراره كأحد معاملات المدخلات يعد كود منتج صحيح في الملف `codename.cfg`.
٣. يقوم الماسح الضوئي بالبحث عن ملف التوصيف المرفق مع كود المنتج. اذا لم يتم امرار معامل نسخة المنتج الاختياري، سيقوم الماسح الضوئي باستخدام أحدث نسخة لملف التوصيف يتم ايجادها في الدليل *ips_root*`/Windows|UNIX_Linux`.
٤. يقوم الماسح الضوئي بالتحقق مما اذا كان نظام التشغيل الفعلي للجهاز هو نظام تشغيل مدعم. سيقوم الماسح الضوئي بالتحقق من نظام التشغيل الفعلي مقابل نظام التشغيل المدعم المتوقع في عناوين القسم في ملف التوصيف، والتي يتضمن اسم الملف نفس كود البرنامج ونسخة البرنامج الخاصة بمعاملات المدخلات.
٥. يقوم الماسح الضوئي بتجميع خصائص المتطلبات الرئيسية الفعلية لعمليات التحقق من المتطلبات الرئيسية باستخدام أدوات تجميع Prerequisite Scanner.
٦. يقوم الماسح الضوئي بالتحقق من خصائص المتطلبات الرئيسية في ملف التوصيف الذي تم ارفاقه مع كود المنتج ونسخة المنتج.

سيقوم الماسح الضوئي بالتحقق من نظام التشغيل الفعلي مقابل نظام التشغيل المدعم في خاصية المتطلبات الرئيسية لنسخة نظام التشغيل أو عناوين القسم في ملف التوصيف، والتي يتضمن اسم الملف بها نفس كود المنتج ونسخة المنتج كمعاملات المدخلات.

٧. سيقوم الماسح الضوئي بقراءة خصائص المتطلبات الرئيسية من ملف التوصيف وتحليل القيم الفعلية والمتوقعة لخصائص المتطلبات الرئيسية لعمليات التحقق من المتطلبات الرئيسية. ويتم استخدام أدوات تقييم Prerequisite Scanner اذا تطلب الأمر.
٨. سيقوم الماسح الضوئي باخراج نتائج عملية المسح في واجهة تعامل سطر الأمر، والملفات النصية وملفات XML للنتائج، وملفات السجل القابلة للقراءة.
٩. سيقوم الماسح الضوئي باعادة تنظيم وازالة الملفات والأدلة المؤقتة.

## الجديد في هذا الاصدار

يقدم IBM Prerequisite Scanner النسخة 1.2 خصائص وتحسينات جديدة. وهو يحتوي أيضا على تصحيحات للأخطاء.

**الخصائص الجديدة في هذه التعديلات**

امكانية اجراء تحليل لغوي وقراءة ملف XML لنتائج عملية المسح الجديدة.

مجموعة أدوات Prerequisite Scanner Java Developer هي مجموعة من واجهات تعامل البرمجة للسماح للمطورين باجراء تحليل لغوي وقراءة محتويات ملف XML للنتائج بشكل برمجي للاحتياجات الخاصة بهم؛ على سبيل المثال، اجراء تحليل لغوي لنتائج عملية المسح لاستخدامها في برنامج تركيب. أنظر "مجموعة أدوات Prerequisite Scanner Java Developer".

**ملفات توصيف جديدة في عملية التعديل هذه**

يوضح الجدول ٩ ملفات التوصيف الجديدة وأكواد المنتج التي تم شحنها مع Prerequisite Scanner النسخة 1.2

*الجدول ٩. ملفات توصيف جديدة*

| المنتج أو المكون | كود المنتج | ملف التوصيف |
|---|---|---|
| Tivoli Composite Application Manager Agent for WebSphere MQ | KMQ | *ips_root*`/Windows|UNIX_Linux/KMQ_07010000.cfg` |
| Tivoli Composite Application Manager Agent for WebSphere Message Broker | KQI | *ips_root*`/Windows|UNIX_Linux/KQI_07010000.cfg` |

**خصائص متطلبات رئيسية جديدة في عملية التعديل هذه**

تم اضافة خاصية `os.SeaMonkeyVersion` للتحقق من نسخة Mozilla SeaMonkey على الجهاز. أنظر "خصائص بيانات نظام التشغيل".

تم اضافة خاصية `env.var.set.`*`env_var_name`* لتحديد ما اذا كان قد تم تحديد متغير بيئة التشغيل كما تم تحديده بواسطة *env_var_name* على الجهاز. أنظر "خصائص بيانات متغير بيئة التشغيل".

**التحسينات في عملية التعديل هذه**

امكانية كتابة نتائج عملية المسح في ملف XML.

*ips_output_dir*`/result.xml` هو ملف نتائج عملية المسح الجديد بالنسق XML. بصفة مفترضة، تقوم الأداة باخراج النتائج في ملف نتائج نصي عادي فقط. أنظر "نسق المخرجات".

**`xmlResult`** هو معامل المدخلات الجديد والاختياري للبرنامج النصي Prerequisite Scanner في Prerequisite Scanner Version 1.2. يمكنك استخدامه لتوجيه الأداة الى اخراج النتائج في ملف XML للنتائج بالاضافة الى ملف نتائج الاختبار العادي. أنظر "prereq_checker".

ازالة القسم المجمع في النتائج اذا لم تكن خصائص المتطلبات الرئيسية للذاكرة أو مساحة القرص موجودة في ملفات التوصيف. ولن تقوم أداة Prerequisite Scanner بعرض الأقسام المجمعة في ملف النتائج عند عدم وجود أي من خصائص المتطلبات الرئيسية للذاكرة ومساحة القرص في ملفات التوصيف. أنظر "نسق المخرجات".

**خصائص متقادمة في عملية التعديل هذه**

لا شيء

**الأخطاء التي تم تصحيحها في عملية التعديل هذه**

بالنسبة الى كشف الأخطاء التي تم تصحيحها في هذا الاصدار، ارجع الى الملف `Readme.html` في الدليل *ips_root* عند استخراج محتويات مجموعات برامج Prerequisite Scanner.

**التغييرات في المطبوعات الفنية في هذا التعديل**

لم يعد يتم وضع دليل مستخدمي Prerequisite Scanner في حزمة البرامج مع مجموعات برامج بيئة التشغيل Prerequisite Scanner الى Prerequisite Scanner. يمكنك استخدام مركز معلومات IBM Prerequisite Scanner Information Center.

# الفصل الثاني. تركيب Prerequisite Scanner

لا يوجد برنامج تركيب الى IBM Prerequisite Scanner. عند استخراج محتويات الملف المضغوط، تكون الملفات الرئيسية في الدليل الرئيسي، مع الأدلة الفرعية التالية: api/ لمجموعة أدوات Prerequisite Scanner Java Developer لدعم واجهة تعامل XML للاستعلام، و lib/ لأدوات التجميع والبرامج النصية العامة؛ و Windows/لأدوات التقييم وملفات التوصيف على نظام Windows؛ و UNIX_Linux/لأدوات التقييم وملفات التوصيف ببيئات تشغيل UNIX؛ و licenses/ لملفات الترخيص.

## المتطلبات الرئيسية

IBM Prerequisite Scanner يمكن تشغيله على نظام Windows أو Windows XP أو ما بعده، ٣٢-بت أو ٦٤-بت. ويمكن تشغيله أيضا على أنظمة تشغيل AIX و HP-UX و Linux و Solaris المختلفة.

تأكد من أنه تم تركيب الوظائف التالية أو أنها متاحة في بيئات التشغيل المستهدفة:

| النظام المستهدف | المتطلبات الرئيسية |
|---|---|
| نظام Windows | • اتاحة الوحدة التابعة الى Telnet، بحيث تعمل عمليات التحقق من امكانية الاتصال في أداة تجميع Connectivity المعرفة مسبقا بطريقة صحيحة.<br>• تأكد من أنه تم تركيب Microsoft .Net Framework 1.0 أو الأحدث، بحيث يمكن تشغيل أداة Prerequisite Scanner بطريقة صحيحة. |
| نظام UNIX | • تم تركيب Bash، بحيث تعمل أدوات تجميع Prerequisite Scanner لنظام UNIX بطريقة صحيحة.<br>• بالنسبة للمستخدمين غير الرئيسيين، يجب تحديد مكان الأوامر **mount** و **swapinfo** و **psrinfo** في متغير بيئة تشغيل PATH، بحيث تكون الأوامر متاحة الى Prerequisite Scanner. توجد الأوامر في الدليل /usr/sbin؛ على سبيل المثال، يمكنك تحديد متغير بيئة تشغيل PATH كما يلي:<br>`export PATH=$PATH:/usr/sbin/`<br>• التأكد من أنه تم تخصيص تصاريح التوصل الصحيحة للأمر **lscfg**، بما في ذلك أية تصاريح تم تحديدها بواسطة شارات امكانية التوصل مثل setuid bit. وتعني تصاريح التوصل الصحيحة أن Prerequisite Scanner تستطيع تنفيذ الأمر واسترجاع معلومات النظام . يوجد الأمر في الدليل /usr/sbin؛ على سبيل المثال لتحديد setuid bit للأمر **lscfg**، قم بتنفيذ الأمر **chmod** كما يلي:<br>`chmod 4777 /usr/sbin/lscfg` |

يقوم Prerequisite Scanner بدعم كل الأجهزة وأنظمة التشغيل للمنتج المحدد أو حلول IBM التي تقوم بتشغيلها Prerequisite Scanner.

## تركيب الملف المضغوط

يمكنك استخراج محتويات الملف المضغوط الى IBM Prerequisite Scanner. يجب أن يتوافر لديك تصاريح الكتابة بالدليل الرئيسي الذي تقوم باستخراج محتويات الملف المضغوط به.

### الاجراء

١. قم بفتح برنامج استعراض الانترنت الخاص بك وأدخل عنوان URL الى IBM Fix Central. تأكد من أنك قمت ببدء الاتصال الى IBM.com أو IBM Support Portal.
٢. من كشف **Product Group**، حدد **Tivoli**.
٣. من كشف **Product**، حدد IBM Prerequisite Scanner.
٤. من كشف **Installed Version**، حدد النسخة التي تريد تحميلها.
٥. من كشف **Platform**، حدد بيئة التشغيل التي تريد تركيب Prerequisite Scanner عليها.
٦. اضغط **Continue**. سيتم فتح صفحة Identify Fixes.
٧. استخدم الاختيار المفترض، **Browse for fixes** واضغط **Continue**.
٨. في صفحة Select fixes، حدد مجموعة البرامج واضغط **Continue**.
٩. في صفحة Download option، حدد اختيار التحميل واضغط **Download now**.
١٠. قم باستخراج محتويات الملفات المضغوطة الى المكان المفضل الخاص بك كما هو محدد بواسطة *ips_root*.

### ما تريد القيام به بعد الآن

تأكد من أنه يمكنك الرجوع الى المطبوعات الفنية لتركيب البرنامج لأي من الخطوات الاضافية والتي يجب تنفيذها قبل تشغيل Prerequisite Scanner. على سبيل المثال، قد تحتاج الى تحديد متغير بيئة التشغيل التي تحدد الى Prerequisite Scanner ما هي المكونات أو الخصائص التي يتم تركيبها على الحاسب المستهدف وبالتالي المتطلبات الرئيسية التي سيتم التحقق منها.

## الغاء تركيب Prerequisite Scanner

قم بازالة IBM Prerequisite Scanner اذا كنت تريد تركيب نسخة جديدة، أو نقله الى بيئة تشغيل أخرى، أو تعد نسخة لم تعد تريدها.

### الاجراء

١. قم بفتح الدليل *ips_root*.
٢. قم بحذف الدليل والمحتويات الخاصة به.

# الفصل الثالث. امتداد Prerequisite Scanner

تقدم أداة IBM Prerequisite Scanner مجموعة أساسية من أدوات التجميع وأدوات التقييم والتوصيفات التي يمكنك استخدامها لتشغيل الأداة واجراء مسح للمتطلبات الرئيسية. اذا لم تحقق مجموعة الملفات الأساسية، وخصائص وقيم المتطلبات الرئيسية، عمليات التحقق من المتطلبات الرئيسية المتطلبات الخاصة بك، فانه يمكنك امتداد Prerequisite Scanner.

## قبل تشغيل Prerequisite Scanner

قبل تشغيل IBM Prerequisite Scanner، حدد ما اذا كانت خصائص المتطلبات الرئيسية المعرفة مسبقا والقيم المتوقعة الخاصة بها وملفات التوصيف تحقق المتطلبات الخاصة بك لاجراء عملية مسح للمتطلبات الرئيسية. اذا لم تحقق أي منهما المتطلبات الخاصة بك، يمكنك تنفيذ مجموعة من مهام المتطلبات الرئيسية لتوصيف أو امتداد Prerequisite Scanner. تعتمد مجموعة مهام وعمليات التحقق من المتطلبات الرئيسية على بيئة التشغيل وعدد عمليات التحقق من المتطلبات الرئيسية.

### عمليات التحقق المطلوبة ومهام الامتداد لأنظمة Windows

يجب أن تقوم بتنفيذ مجموعة من عمليات التحقق والمهام قبل تشغيل IBM Prerequisite Scanner. تحدد عمليات التحقق هذه ما اذا كان يمكنك تحرير واستخدام ملفات التوصيف الموجودة أو يجب أن تقوم بمد Prerequisite Scanner.

يقدم الجدول ١٠ كشف بعمليات التحقق والمهام المطلوب تنفيذها.

*الجدول ١٠. عمليات التحقق والمهام قبل استخدام ملف التوصيف لأنظمة Windows*

| | التحقق | المهمة |
|---|---|---|
| ☐ | تحقق مما اذا كان المنتج وأنظمة التشغيل المدعمة الخاصة به ونسخ نظام التشغيل معروضة في ملف `.codename.cfg`. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة لا، قم باضافة كود المنتج ونظام التشغيل المنفرد ونسخة نظام التشغيل الاختيارية الى الملف. للحصول على المزيد من المعلومات، أنظر "اضافة أكواد البرنامج". |
| ☐ | تحقق مما اذا كان ملف التوصيف موجودا لكود المنتج المرفق مع نسخة المنتج. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة لا، قم بتكوين ملف توصيف ليتضمن خصائص المتطلبات الرئيسية لنظام التشغيل هذا ونسخة نظام التشغيل. للحصول على المزيد من المعلومات، أنظر "تكوين ملفات التوصيف المهيأة". |
| ☐ | قم بفتح ملف التوصيف وتحقق مما اذا كان يتضمن خصائص المتطلبات الرئيسية الصحيحة أم لا. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة بلا، قم باضافة خصائص المتطلبات الرئيسية. للحصول على المزيد من المعلومات، أنظر "اضافة خصائص المتطلبات الرئيسية". |
| ☐ | تحقق مما اذا كانت خصائص المتطلبات الرئيسية لها القيم المتوقعة أم لا. | • اذا كانت الاجابة بنعم، قم بتشغيل Prerequisite Scanner. للحصول على المزيد من المعلومات، أنظر "الفصل الرابع. تشغيل Prerequisite Scanner".<br>• اذا كانت الاجابة بلا، قم بتحرير خصائص المتطلبات الرئيسية. للحصول على المزيد من المعلومات، أنظر "تحرير خصائص المتطلبات الرئيسية". |
| ☐ | بالنسبة الى أي من خصائص المتطلبات الرئيسية الجديدة، تحقق مما اذا كانت أدوات التجميع سابقة التعريف يمكن أن تقوم بتجميع القيم الفعلية لخصائص المتطلبات الرئيسية. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة بلا، قم بتكوين أدوات التجميع المهيأة. للحصول على المزيد من المعلومات، أنظر "تكوين أدوات التجميع المهيأة لأنظمة Windows". |

| | | |
|---|---|---|
| ☐ | بالنسبة الى أي من خصائص المتطلبات الرئيسية الجديدة أو التي تم تحريرها، تحقق مما اذا كانت أدوات التحقق سابقة التعريف يمكن أن تقوم بمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة بلا، قم بتكوين أدوات التحقق المهيأة.<br>للحصول على المزيد من المعلومات، أنظر "تكوين أدوات التحقق المهيأة لأنظمة Windows". |
| ☐ | تأكد من حفظ كل الملفات في الأدلة الصحيحة:<br>• ملفات التوصيف، أية أدوات تجميع مهيأة خاصة بالمنتج وملفات دفعية مرفقة، وأية ملفات مهيأة لأداة التقييم في الدليل *ips_root*/Windows<br>• أدوات التجميع العامة المهيأة في الدليل *ips_root*/lib | قم بتشغيل Prerequisite Scanner. للحصول على المزيد من المعلومات، أنظر "الفصل الرابع. تشغيل Prerequisite Scanner". |

## عمليات التحقق المطلوبة ومهام الامتداد لأنظمة UNIX

يجب أن تقوم بتنفيذ مجموعة من عمليات التحقق والمهام للمتطلبات الرئيسية قبل تشغيل IBM Prerequisite Scanner. تحدد عمليات التحقق هذه ما اذا كان يمكنك تحرير واستخدام ملفات التوصيف الموجودة أو يجب أن تقوم بمد Prerequisite Scanner.

يقدم الجدول ١١ كشف بعمليات التحقق والمهام المطلوب تنفيذها.

*الجدول ١١. عمليات التحقق والمهام قبل استخدام ملف التوصيف لأنظمة UNIX*

| | التحقق | المهمة |
|---|---|---|
| ☐ | يتم التحقق مما اذا كان المنتج معروض في الملف codename.cfg. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة لا، قم باضافة كود المنتج الى الملف codename.cfg. للحصول على المزيد من المعلومات، أنظر "اضافة أكواد البرنامج". |
| ☐ | تحقق مما اذا كان ملف التوصيف موجودا لكود المنتج المرفق مع المنتج. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة لا، قم بتكوين ملف توصيف ليتضمن خصائص المتطلبات الرئيسية لكل بيئات التشغيل المدعمة للمنتج. للحصول على المزيد من المعلومات، أنظر "تكوين ملفات التوصيف المهيأة". |
| ☐ | قم بفتح ملف التوصيف وتحقق مما اذا كان يتضمن خصائص المتطلبات الرئيسية الصحيحة أم لا. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة بلا، قم باضافة خصائص المتطلبات الرئيسية. للحصول على المزيد من المعلومات، أنظر "اضافة خصائص المتطلبات الرئيسية". |
| ☐ | تحقق مما اذا كانت خصائص المتطلبات الرئيسية لها القيم المتوقعة أم لا. | • اذا كانت الاجابة بنعم، قم بتشغيل Prerequisite Scanner. للحصول على المزيد من المعلومات، أنظر "الفصل الرابع. تشغيل Prerequisite Scanner".<br>• اذا كانت الاجابة بلا، قم بتحرير خصائص المتطلبات الرئيسية. للحصول على المزيد من المعلومات، أنظر "تحرير خصائص المتطلبات الرئيسية". |
| ☐ | بالنسبة الى أي من خصائص المتطلبات الرئيسية الجديدة، تحقق مما اذا كانت أدوات التجميع سابقة التعريف يمكن أن تقوم بتجميع القيم الفعلية لخصائص المتطلبات الرئيسية. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة بلا، قم بتكوين أدوات التجميع المهيأة.<br>للحصول على المزيد من المعلومات، أنظر "تكوين أدوات التجميع المهيأة لأنظمة UNIX". |

| | | |
|---|---|---|
| ☐ | بالنسبة الى أي من خصائص المتطلبات الرئيسية الجديدة أو التي تم تحريرها، تحقق مما اذا كانت أدوات التحقق يمكن أن تقوم بمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة بلا، قم بتكوين أدوات التحقق المهيأة. للحصول على المزيد من المعلومات، أنظر "تكوين أدوات التحقق المهيأة لأنظمة UNIX". |
| ☐ | بالنسبة لأية خصائص متطلبات رئيسية جديدة أو تم تحريرها، تحقق مما اذا كان الكود لاستدعاء وتشغيل أدوات التجميع يوجد في البرنامج النصي *ips_root*/UNIX_Linux/packageTest.sh. | • اذا كانت الاجابة نعم، قم بعملية التحقق التالية.<br>• اذا كانت الاجابة بلا، قم بتحرير البرنامج النصي للاختبار لمجموعة البرامج الرئيسية. للحصول على المزيد من المعلومات، أنظر "تحرير البرنامج النصي لاختبار مجموعة البرامج لأنظمة UNIX". |
| ☐ | تأكد من حفظ كل الملفات في الأدلة الصحيحة:<br>• ملفات التوصيف، أية ملفات أدوات تجميع مهيأة وأية ملفات أدوات تقييم مهيأة في الدليل *ips_root*/UNIX_Linux | قم بتشغيل Prerequisite Scanner. للحصول على المزيد من المعلومات، أنظر "الفصل الرابع. تشغيل Prerequisite Scanner". |

## اضافة أكواد البرنامج

تقدم أداة IBM Prerequisite Scanner مجموعة من أكواد نسخة المنتج المعرفة مسبقا في الملف codename.cfg. يمكنك اضافة أكواد المنتج اذا لم يحتوي الملف عليها لمعرفة نسخة المنتج وبيئات التشغيل المدعمة له ونسخ أنظمة التشغيل.

### الاجراء

١. قم بفتح الملف *ips_root*/codename.cfg.
٢. تأكد مما اذا كان الملف يتضمن أزواج قيمة الاسم لنسخ البرنامج بالفعل.
٣. اذا لم يكن كود المنتج متاحا، قم باضافة أحد الأكواد وتأكد من استخدام النسق الصحيح كما يلي:

*product_code=code_value*

**قيد:** IBM Tivoli Monitoring و Tivoli Composite Application Manager له أكواد برنامج سابقة التعريف يقوم Prerequisite Scanner باعتبارها محتجزة. لا يجب استخدام هذه الأكواد كأكواد لبرنامج Prerequisite Scanner ما لم تكن تقوم بالاشارة الى IBM Tivoli Monitoring و Tivoli Composite Application Manager الخاصة بها. للحصول على المزيد من المعلومات عن أكواد البرنامج، أنظر ITM 6.X Product Codes Technote.

**قيد:** بنظام UNIX فقط: عند ادخال قيمة لكود المنتج في الملف، تجنب استخدام for. حيث تعد كلمة محجوزة ويمكن أن تؤثر على كيفية تشغيل Prerequisite Scanner.

على سبيل المثال، لاضافة كود منتج الى IBM Tivoli Monitoring for Energy Management على كل أنظمة تشغيل Windows، قم باضافة السطر التالي الى الملف:

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

## تكوين ملفات التوصيف المهيأة

يمكنك تكوين ملفات التوصيف المهيأة من ملف توصيف نموذجي اذا لم تحقق ملفات التوصيف المعرفة مسبقا المتطلبات الخاصة بك لخصائص المتطلبات الرئيسية. قبل تكوين ملف التوصيف المهيأ، تأكد من خصائص المتطلبات الرئيسية التي تريد اضافتها والقيم المتوقعة الخاصة بها.

### نبذة عن هذه المهمة

**هام:** يجب الالتزام بمصطلحات التسمية وقواعد التنسيق التي تحكم عملية تكوين وتحرير ملف توصيف مهيأ. اذا لم يتم الالتزام، لن تستطيع أداة Prerequisite Scanner اجراء عملية المسح باستخدام هذا الملف.

### الاجراء

١. اذا تطلب الأمر، قم باضافة أكواد المنتجات الى ملف `codename.cfg`.
٢. قم بتكوين ملف التوصيف باستخدام برنامج تحرير نصي في الدليل *ips_root*`/OS`. تأكد من استخدام مصطلح التسمية التالي لاسم الملف:

`product_code_version.cfg`

حيث:

- *product_code*

يعد المتغير المستخدم لتمثيل كود المنتج على نظام Windows أو نظام UNIX. تقوم أكواد المنتج بتعريف المنتج، وبيئة تشغيل منفردة مثل Windows و AIX و HP-UX و Linux و Solaris، واختياريا نسخة نظام التشغيل المدعمة بواسطة هذا المنتج. ويتم تخزينها في الملف `codename.cfg`. يحتوي أي منتج يدعم عدة بيئات تشغيل على عدة أكواد للمنتج مع كلا منهما، حيث تقوم بتعريف المنتج وبيئة التشغيل ونسخة نظام التشغيل وفقا للمطلوب.

- *Version* هو كود مكون من ٨-أرقام يمثل النسخة، والاصدار، والتعديلات، والمستوى، مع رقمين لكل من أجزاء الكود؛ على سبيل المثال 7.3.21 هو 07032100.

٣. قم بمراجعة خصائص المتطلبات الرئيسية الموضحة في "الملحق ج. مرجع خصائص المتطلبات الرئيسية" وحدد الخصائص التي تريد التحقق منها.
٤. اختياري: قم باضافة قسم وتأكد من استخدام مصطلح التسمية التالي لعنوان القسم:

- **تصنيف نوع بيانات معرف مسبقا ومنفرد**

`[category_name:category_value]`

على سبيل المثال، لتكوين قسم عام لخصائص المتطلبات الرئيسية لكل بيئات تشغيل Windows، قم باضافة عنوان القسم التالي:

`[OSType:Windows]`

على سبيل المثال، لتكوين قسم عام لخصائص المتطلبات الرئيسية لكل اختلافات نظام تشغيل RedHat Linux OS، قم باضافة عنوان القسم التالي:

`[OSType:RedHat]`

- **تصنيفات نوع بيانات معرفة مسبقا ومجمعة**

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

على سبيل المثال، لتكوين قسم لخصائص المتطلبات الرئيسية لاختلافات Windows Server 2003 باستثناء Windows Server 2003 R2، قم باضافة عنوان قسم المجموعة التالي:

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

على سبيل المثال، قم بتكوين قسم لخصائص المتطلبات الرئيسية لنظام تشغيل SUSE Linux Enterprise Server 11 OS وحدد ما اذا كان قد تم تحديد متغير بيئة التشغيل `@TPAE_DB_SERVER` الى `true`. قم باضافة عنوان قسم المجموعة التالي:

`[OSType=SUSELinuxEnterpriseServer][@TPAE_DB_SERVER:true]`

حيث:

*category_name* هو كود متعدد الحروف يمثل تصنيف نوع البيانات كما هو موضح في الجدول ٦

*category_value* يعد كود متعدد الحروف يمثل قيمة مسموح بها للتصنيف كما هو موضح في الجدول ٦

٥. اختياري: في كل قسم، قم بمراجعة خصائص المتطلبات الرئيسية الأساسية الموضحة في "الملحق ج. مرجع خصائص المتطلبات الرئيسية" وحدد الخصائص التي تريد التحقق منها.
٦. قم بادخال زوج قيم الاسم مع المؤهلات الاختيارية كما يلزم، وذلك بالنسبة الى كل من خصائص المتطلبات الرئيسية التي تريد اضافتها. تأكد من استخدام النسق التالي، مع خاصية متطلبات رئيسية واحدة فقط بكل سطر:

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

حيث:

- *prefix_identifier* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض التصنيفات سابقة التعريف.
- *property_name* هو اسم خاصية المتطلبات الرئيسية.
- *suffix_identifier* هو كود تعريف اختياري للنوع الفرعي لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٤.
- *qualifier_name* هو خاصية مميزة اختيارية لخاصية المتطلبات الرئيسية. يقوم IBM Prerequisite Scanner باستخدامه لتأهيل خاصية المتطلبات الرئيسية أو نوع التحقق الذي سيتم تنفيذه على خاصية المتطلبات الرئيسية، كما هو موضح في "المؤهلات المعرفة مسبقا لخصائص المتطلبات الرئيسية".

  **ملاحظة:** لديك مؤهلات متعددة، يتم الفصل بينها بفاصلة. يجب تضمين مجموعة المؤهلات بأقواس مربعة [] .
- *qualifier_value* هي القيمة للخاصية الاختيارية. يجب الفصل بين كل من المؤهلات والقيمة الخاصة بها باستخدام النقطتين : .
- *property_value* هي قيمة خاصية المتطلبات الرئيسية ويمكن أن تكون مجموعة حروف أو عدد صحيح.

على سبيل المثال، يحتوي التصنيف المعرف مسبقا بواسطة المستخدم لخصائص المتطلبات الرئيسية على بادئة كود التعريف `user`. خاصية المتطلبات الرئيسية الخاصة بالتحقق مما اذا كان المستخدم الذي قام ببدء الاتصال ينتمي الى مجموعة موجه النظام هي: `user.isAdmin=True`

٧. اذا لم تكن خاصية المتطلبات الرئيسية موجودة في تصنيفات معرفة مسبقا، قم باضافة اسم خاصية المتطلبات الرئيسية المهيأة والقيمة الخاصة بها ومؤهلات اضافية. يجب عندئذ أن تقوم بتكوين الملفات التالية للتحقق من ومقارنة خاصية المتطلبات الرئيسية المهيأة وفقا للمطلوب: أداة تجميع مهيأة لتجميع القيم الفعلية لخاصية المتطلبات الرئيسية وأداة تقييم مهيأة اذا لم تستطع وظائف المقارنة القياسية مقارنة القيم الفعلية والمتوقعة.

---

## اضافة خصائص المتطلبات الرئيسية

يمكنك اضافة خصائص المتطلبات الرئيسية الأساسية من التصنيفات التي تم تعريفها مسبقا لخصائص المتطلبات الرئيسية الى ملفات التوصيف. أو، يمكنك اضافة خصائص المتطلبات الرئيسية.

### نبذة عن هذه المهمة

**هام:** يجب الالتزام بقواعد التنسيق التي تحكم عملية اضافة وتحرير خصائص المتطلبات الرئيسية بملف التوصيف. اذا لم يتم الالتزام، لن تستطيع أداة Prerequisite Scanner اجراء عملية المسح لخاصية المتطلبات الرئيسية هذه.

### الاجراء

١. قم بفتح ملف التوصيف.
٢. قم بمراجعة خصائص المتطلبات الرئيسية الموضحة في "الملحق ج. مرجع خصائص المتطلبات الرئيسية" وحدد الخصائص التي تريد التحقق منها.
٣. قم بادخال زوج قيم الاسم مع المؤهلات الاختيارية كما يلزم، وذلك بالنسبة الى كل من خصائص المتطلبات الرئيسية التي تريد اضافتها.

على سبيل المثال، لاضافة خصائص المتطلبات الرئيسية للتصنيف سابق التعريف العام، قم بادخال اسم الخاصية والقيمة المتوقعة فقط. واضافة معاملات المتطلبات الرئيسية التالية الى الملف:

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

على سبيل المثال، التصنيف سابق التعريف لشبكة الاتصال لخصائص المتطلبات الرئيسية له بادئة كود التعريف network واسم خاصية المتطلبات الرئيسية للتحقق من منافذ الاتصال المتاحة هو availablePorts. يمكنك اجراء المزيد من التصنيف على المنافذ المتاحة وفقا للأنواع الفرعية للتطبيق، DB2 لوحدة خدمة قاعدة بيانات DB2، WAS لوحدة خدمة WebSphere Application Server، FTP لبروتوكول FTP. واضافة معاملات المتطلبات الرئيسية التالية الى الملف:

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

على سبيل المثال، التصنيف سابق التعريف لنظام التشغيل لخصائص المتطلبات الرئيسية له بادئة كود التعريف os واسم خاصية المتطلبات الرئيسية للتحقق من مساحة القرص المتاحة هو space. يمكنك اجراء المزيد من التصنيف على عملية التحقق وفقا للأنواع الفرعية لنظام الملفات، usr و home. يمكنك تحديد القيم للمؤهلات dir و unit.

واضافة معاملات المتطلبات الرئيسية التالية الى الملف:

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

**هام:** يمكنك استخدام المؤهلات سابقة التعريف مع خصائص المتطلبات الرئيسية سابقة التعريف المحددة فقط، كما هو موضح في الجدول ٥.

٤. اذ لم تكن خاصية المتطلبات الرئيسية موجودة في التصنيفات المعرفة مسبقا لخصائص المتطلبات الرئيسية، قم باضافة زوج قيم الاسم مع المؤهل الاختياري لقيمة وخاصية المتطلبات الرئيسية المهيأة. تأكد من استخدام النسق التالي، مع خاصية المتطلبات الرئيسية فقط بكل سطر.

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

حيث:

- *prefix_identifier* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض التصنيفات سابقة التعريف.
- *property_name* هو اسم خاصية المتطلبات الرئيسية.
- *suffix_identifier* هو كود تعريف اختياري للنوع الفرعي لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٤.
- *qualifier_name* هو خاصية مميزة اختيارية لخاصية المتطلبات الرئيسية. يقوم IBM Prerequisite Scanner باستخدامه لتأهيل خاصية المتطلبات الرئيسية أو نوع التحقق الذي سيتم تنفيذه على خاصية المتطلبات الرئيسية، كما هو موضح في "المؤهلات المعرفة مسبقا لخصائص المتطلبات الرئيسية".

  **ملاحظة:** لديك مؤهلات متعددة، يتم الفصل بينها بفاصلة. يجب تضمين مجموعة المؤهلات بأقواس مربعة [].
- *qualifier_value* هي القيمة للخاصية الاختيارية. يجب الفصل بين كل من المؤهلات والقيمة الخاصة بها باستخدام النقطتين :.
- *property_value* هي قيمة خاصية المتطلبات الرئيسية ويمكن أن تكون مجموعة حروف أو عدد صحيح.

على سبيل المثال، env.tcrhome هي خاصية المتطلبات الرئيسية المهيأة التي تقوم بالتحقق من متغير بيئة تشغيل الدليل الرئيسي Tivoli Common Reporting، ويجب أن تكون القيمة المتوقعة هي True:

```
env.tcrhome=True
env.path.jar
```

هي خاصية المتطلبات الرئيسية المهيأة التي تقوم بالتحقق مما اذا كان قد تم تحديد JRE في متغير بيئة التشغيل PATH، ويجب أن تكون القيمة المتوقعة هي False:

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

**ملاحظة:** يجب عندئذ أن تقوم بتكوين الملفات التالية للتحقق من ومقارنة خاصية المتطلبات الرئيسية المهيأة وفقا للمطلوب:
أداة تجميع مهيأة لتجميع القيم الفعلية لخاصية المتطلبات الرئيسية وأداة تقييم مهيأة فقط اذا لم تستطع وظائف المقارنة القياسية مقارنة القيم الفعلية والمتوقعة.

## تحرير خصائص المتطلبات الرئيسية

يمكنك تحرير خصائص المتطلبات الرئيسية أو تغيير القيم المتوقعة لخصائص المتطلبات الرئيسية هذه أو تغيير القيم المرفقة للمؤهلات.

### قبل البدء

تحقق مما اذا كانت القيمة الجديدة هي قيمة صحيحة مدعمة بواسطة خاصية المتطلبات الرئيسية. على سبيل المثال، تتوقع خاصية المتطلبات الرئيسية `Disk`، نسق رقمي بالوحدة ميجابايت أو جيجابايت. اذا كنت تريد التحقق من مساحة القرص المتاحة بالتيرابايت (TB)، فانه يجب امتداد واجهة تعامل البرمجة للمقارنة لدعم مقارنات TB. يجب أن تقوم أيضا بتحرير خاصية المتطلبات الرئيسية `Disk` في ملفات التوصيف المتعلقة.

تحقق من المؤهلات المعرفة مسبقا والقيم الصحيحة لخاصية المتطلبات الرئيسية، كما هو موضح في "المؤهلات المعرفة مسبقا لخصائص المتطلبات الرئيسية".

### الاجراء

١. قم بفتح ملف التوصيف.
٢. بالنسبة لكل خاصية للمتطلبات الرئيسية تريد تحريرها، أدخل القيمة المتوقعة الجديدة أو قم بتغيير القيمة للمؤهل. على سبيل المثال، موجه النظام الجديد هو المستخدم الرئيسي، لذلك يجب تغيير قيمة خاصية المتطلبات الرئيسية `user.userID`. قم بتغيير القيمة الى الاسم الجديد:

```
user.userID=smithj
```

على سبيل المثال، يحتوي المؤهل `type` لخاصية المتطلبات الرئيسية `os.ulimit` على القيمة `filedescriptorlimit` للتحقق من الحد لموصفات الملف. قد تريد التحقق من الحد الآخر مثل حجم الرصة. قم بتغيير قيمة المؤهل التالية لخاصية المتطلبات الرئيسية من:

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

الى:

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

**هام:** يمكنك استخدام المؤهلات سابقة التعريف مع خصائص المتطلبات الرئيسية سابقة التعريف المحددة فقط، كما هو موضح في الجدول ٥.

## تكوين أدوات التجميع المهيأة لأنظمة Windows

يمكنك تكوين أدوات التجميع المهيأة اذا كانت أدوات التجميع الأساسية المحددة لا تقوم بتجميع قيم لخصائص المتطلبات الرئيسية المطلوبة حتى يمكن تركيب البرنامج. يمكنك تكوين أدوات تجميع VBScript العامة المهيأة لتجميع بيانات لخصائص المتطلبات الرئيسية التي يتم تطبيقها على أي من البرامج ونسخ البرامج. أو، يمكنك تكوين خصائص مهيأة محددة للبرنامج لتجميع البيانات التي تنطبق على برنامج محدد ونسخ معينة للبرنامج. بينما يقوم كل نوع لأداة تجميع VBScript المهيأة بتجميع بيانات باستخدام نفس الطرق، تكون قواعد التكوين والتخزين والتنفيذ مختلفة قليلا.

### تكوين أدوات تجميع VBScript المهيأة لكل ملفات التوصيف

عند تكوين أدوات تجميع VBScript العامة المهيأة، يجب أن يحتوي اسم الملف على اسم خاصية المتطلبات الرئيسية وتخزينه في الدليل الفرعي `lib/`. تحتوي أداة التجميع على كود للحصول على القيمة الفعلية لخاصية المتطلبات الرئيسية. ويمكن أيضا استخدام الوظائف العامة والبرامج الفرعية للحصول على القيمة اذا تطلب الأمر.

**قبل البدء**

تأكد من مراجعة مجموعة الوظائف المعرفة مسبقا والبرامج الفرعية في الملاحق التالية قبل تكوين أدوات التجميع. حدد ما اذا كان يمكنك استخدام أي منها للحصول على القيم الفعلية:

- "الملحق ه. الوظائف المعتادة لأنظمة Windows"
- "الملحق ز. البرامج الفرعية لوظيفة الملفات لنظام Windows"
- "الملحق و. البرامج الفرعية لوظيفة التسجيل لأنظمة Windows"
- "الملحق ح. الوظائف المعتادة للبرامج الفرعية لأنظمة Windows"

قم بتحديد ما اذا كانت أداة التجميع يجب أن تقوم بالتحقق من وجود خاصية المتطلبات الرئيسية، واذا كانت موجودة، ما هي المعلومات الأخرى التي يجب جمعها. يجب أن تقوم كل من أدوات التحقق بارجاع قيمة، وما اذا كان أي منها موجودا أم لا. على سبيل المثال:

- تحقق مما اذا كان متغير بيئة التشغيل موجودا، مثل الدليل الرئيسي للمنتج، على سبيل المثال، TCR_HOME الى Tivoli Common Reporting.
- تحقق مما اذا كان متغير بيئة التشغيل يحتوي على ملف JAR أو ملف ثنائي أو مسار، مثل المسار الى JRE في متغير بيئة التشغيل PATH.
- تحقق من القيمة الفعلية لمتغير بيئة التشغيل، مثل الدليل الرئيسي للمنتج، على سبيل المثال، TCR_HOME الى Tivoli Common Reporting.
- تحقق مما اذا كان قد تم تركيب البرنامج أم لا.
- تحقق من نسخة البرنامج التي تم تركيبها.

**الاجراء**

١. قم بتكوين ملف VBScript. قم بحفظ الملف بالدليل *ips_root*/lib، مع الاختلاف عن قواعد تسمية الملف التالية:

[*prefix_identifier*.]property_name.vbs

حيث:

- *prefix_identifier* هو بادئة كود التعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣.
- *property_name* هو اسم خاصية المتطلبات الرئيسية ويتم استخدامه في اسم أداة التجميع.

على سبيل المثال، يحتوي mssqlVersion.vbs على الكود للحصول على القيمة الفعلية لخاصية المتطلبات الرئيسية لوحدة خدمة MS SQL على جهاز Windows.

٢. باستخدام برامج تحرير VBScript، قم باضافة الكود للحصول على قيمة خاصية المتطلبات الرئيسية. استخدم VBScript COM والوظائف للتوصل الى عناصر بيئة تشغيل Windows وتشغيلها في بيئة تشغيل Windows Script Host. تأكد من أن عملية التحقق تقوم بارجاع مخرجات قياسية كما يلي:

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

- *property_name* الذي يمثل خاصية المتطلبات الرئيسية كما تم كتابتها في ملف التوصيف، على سبيل المثال، env.tcrhome.
- *var_for_value*، أي أن المتغير VBScript للقيمة الفعلية التي ستقوم أداة التجميع بالحصول عليها لخاصية المتطلبات الرئيسية.

للتحقق مما اذا كانت بيئة تشغيل TCR_HOME موجودة وارجاع القيمة الفعلية، حيث اسم خاصية المتطلبات الرئيسية هو env.tcrhome:

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

للتحقق مما اذا كان قد تم تحديد JRE في متغير PATH، حيث اسم خاصية المتطلبات الرئيسية هو env.path.jre:

```
Set wshShell = WScript.CreateObject("WScript.Shell")
path = WshShell.ExpandEnvironmentStrings("%PATH%")
```

```
Set objRegEx = new RegExp
objRegEx.Pattern = "(^|([:;\\/]))(C:\Program
Files\IBM\Java60\jre\bin)($|[:;])"
objRegEx.IgnoreCase = True
objRegEx.Global = True
Set matches = objRegEx.Execute(path)
WScript.Echo "env.path.jre=" &#38; (matches.Count > 0)
```

للتحقق من نسخة Tivoli Directory Integrator المركبة حاليا، حيث اسم خاصية المتطلبات الرئيسية هو `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 &amp; "\" &#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
```

٣. قم بتشغيل أداة التجميع VBScript للتأكد من أنه لا توجد أخطاء في التشغيل وتصحيح الأخطاء كما يلزم.
٤. قم بتكوين أدوات التحقق المهيأة اذا كانت وظائف المقارنة القياسية لا يمكنها أن تقوم بمقارنة القيم الفعلية والمتوقعة.

## تكوين أدوات تجميع VBScript مهيأة خاصة بمنتج ونسخة المنتج

عند تكوين أدوات تجميع VBScript خاصة بمنتج مهيأ، يجب أن يكون اسم الملف بنفس كود منتج ملف التوصيف وتم تخزينه في الدليل الفرعي `Windows/`. يمكن أن تحتوي أداة التجميع على كود لجمع القيم الفعلية لواحدة أو أكثر من خصائص المتطلبات الرئيسية. ويمكنها أيضا استخدام الوظائف العامة والبرامج الفرعية لجمع هذه القيم اذا تطلب الأمر.

### قبل البدء

تأكد من مراجعة مجموعة الوظائف والبرامج الفرعية في الملاحق التالية قبل تكوين أدوات التجميع. حدد ما اذا كان يمكنك استخدام أي منها للحصول على القيم الفعلية:

- "الملحق ه. الوظائف المعتادة لأنظمة Windows"
- "الملحق ز. البرامج الفرعية لوظيفة الملفات لنظام Windows"
- "الملحق و. البرامج الفرعية لوظيفة التسجيل لأنظمة Windows"
- "الملحق ح. الوظائف المعتادة للبرامج الفرعية لأنظمة Windows"

قم بتحديد ما اذا كانت أداة التجميع يجب أن تقوم بالتحقق من وجود خاصية المتطلبات الرئيسية، واذا كانت موجودة، ما هي المعلومات الأخرى التي يجب جمعها. يجب أن تقوم كل من أدوات التحقق بارجاع قيمة، اذا كان أي منها موجودا. على سبيل المثال:

- تحقق من وجود الدليل.
- تحقق من مساحة القرص المتاحة للدليل.
- تحقق مما اذا كان قد تم تركيب البرنامج أم لا.
- تحقق من نسخة البرنامج التي تم تركيبها.

**الاجراء**

١. قم بتكوين ملف VBScript. قم بحفظ الملف بالدليل `Windows/`*ips_root*، مع الاختلاف عن قواعد تسمية الملف التالية:

*product_code*`[_`*version*`].vbs`

حيث:

- *product_code*

يعد المتغير المستخدم لتمثيل كود المنتج على نظام Windows أو نظام UNIX. تقوم أكواد المنتج بتعريف المنتج، وبيئة تشغيل منفردة مثل Windows و AIX و HP-UX و Linux و Solaris، واختياريا نسخة نظام التشغيل المدعمة بواسطة هذا المنتج. ويتم تخزينها في الملف `codename.cfg`. يحتوي أي منتج يدعم عدة بيئات تشغيل على عدة أكواد للمنتج مع كلا منهما، حيث تقوم بتعريف المنتج وبيئة التشغيل ونسخة نظام التشغيل وفقا للمطلوب.

- *Version* هو كود مكون من ٨-أرقام يمثل النسخة، والاصدار، والتعديلات، والمستوى، مع رقمين لكل من أجزاء الكود؛ على سبيل المثال 7.3.21 هو 07032100.

٢. باستخدام برنامج التحرير VBScript، قم بفتح الملف وتضمين المسار الى `common_function.vbs` اذا كان يجب استخدام الوظائف العامة، كما يلي:

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

٣. اذا كان يجب استخدام قيم `PATH` وتم امرار شارة p– من Prerequisite Scanner، عندئذ استخدم `Wscript.Arguments()` حيث `Wscript.Arguments(0)` هو قيمة `PATH`.
`Wscript.Arguments(1)` هو شارة p– والقيم الخاصة بها.

٤. قم باضافة الكود للحصول على قيمة خاصية المتطلبات الرئيسية باستخدام VBScript COM والوظائف للتوصل الى عناصر بيئة تشغيل Windows. تشغيل في بيئة Windows Script Host. تأكد من أن عملية التحقق تقوم بارجاع مخرجات قياسية كما يلي:

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

- *property_name* الذي يمثل خاصية المتطلبات الرئيسية كما تم كتابتها في ملف التوصيف، على سبيل المثال، `env.tcrhome`.
- *var_for_value*، أي أن المتغير VBScript للقيمة الفعلية التي ستقوم أداة التجميع بالحصول عليها لخاصية المتطلبات الرئيسية.

للتحقق من مساحة القرص المتاحة لدليل التركيب لمنتج. على سبيل المثال، للتحقق من Tivoli Monitoring for Energy Management Reporting and Optimization باستخدام البرنامج الفرعي "getValue()"، حيث خاصية المتطلبات الرئيسية هي `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
```

٥. قم بتكوين ملف دفعي لاستدعاء أداة التجميع VBScript. يجب أن يحتوي الملف الدفعي على نفس اسم ملف التوصيف ولاحقة .bat، كما يلي: *product_code*[_*version*].bat

```
@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%
```

٦. قم بتشغيل أداة التجميع VBScript للتأكد من أنه لا توجد أخطاء في التشغيل وتصحيح الأخطاء كما يلزم.
٧. قم بتكوين أدوات التحقق المهيأة اذا كانت وظائف المقارنة القياسية لا يمكنها أن تقوم بمقارنة القيم الفعلية والمتوقعة.

## تكوين أدوات التجميع المهيأة لأنظمة UNIX

يمكنك تكوين أدوات التجميع المهيأة اذا كانت أدوات التجميع الأساسية المحددة لا تقوم بتجميع قيم لخصائص المتطلبات الرئيسية المطلوبة حتى يمكن تركيب البرنامج. عند تكوين أدوات تجميع مهيأة، يجب أن يكون اسم الملف مماثلا لخاصية المتطلبات الرئيسية بدون النوع الفرعي الذي يوجد بالاسم الخاص بها. يتم تخزين أداة التجميع في الدليل الفرعي UNIX_Linux/. يمكن أن تحتوي أداة التجميع على كود للحصول على القيم الفعلية لواحدة أو أكثر من خصائص المتطلبات الرئيسية. ويمكنها أيضا استخدام الوظائف العامة للحصول على هذه القيم اذا تطلب الأمر.

### قبل البدء

تأكد من مراجعة مجموعة الوظائف في الملاحق التالية قبل تكوين أدوات التجميع. حدد ما اذا كان يمكنك استخدام أي منها للحصول على القيم الفعلية:

- "الملحق ط. الوظائف المعتادة لأنظمة UNIX"
- "الملحق ي. الوظائف الأخرى لأنظمة UNIX"
- "الملحق ك. وظائف تسجيل الوظيفة لأنظمة UNIX"

قم بتحديد ما اذا كانت أداة التجميع يجب أن تقوم بالتحقق من وجود خاصية المتطلبات الرئيسية، واذا كانت موجودة، ما هي المعلومات الأخرى التي يجب جمعها. يجب أن تقوم كل من أدوات التحقق بارجاع قيمة، اذا كان أي منها موجودا. على سبيل المثال:

- تحقق مما اذا كان قد تم تركيب المنتج، على سبيل المثال، تم تركيب مجموعة برامج مع RPM.
- تحقق من نسخة البرنامج التي تم تركيبها.
- تحقق مما اذا كانت مساحة القرص المتاحة لنظام الملفات الذي تم تركيبه.

اذا كنت تريد استخدام الأنواع الفرعية، *suffix_identifier*، واجراء المزيد من التصنيف لخاصية المتطلبات الرئيسية وفقا للتطبيق أو الوظيفة أو النوع الفرعي للخدمة، فانه يمكنك تكوين أداة تجميع عامة. قم بامرار الفارق للنوع الفرعي *suffix_identifier*، الذي يمثل *differentiator_suffix_identifier*، الى أداة التجميع الخاصة به. على سبيل المثال، os.package هي أداة التجميع العامة للتحقق من وجود مجموعات البرامج. للتحقق من وجود openssh، قم بامرار اسم مجموعة البرامج عند استدعاء أداة تجميع os.package في ملف البرنامج النصي packageTest.sh، كما يلي:

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

حيث openssh هو اسم مجموعة البرامج، الذي يمثل، النوع الفرعي *suffix_identifier* والفارق *differentiator_suffix_identifier*.

**الاجراء**

١. قم بتكوين ملف البرنامج النصي shell. وحفظ الملف في الدليل *ips_root*/Unix_Linux، مع اختلاف مصطلح تسمية الملف التالي لكن بدون لاحقة الملف:

[*prefix_identifier.*]*property_name*

حيث:

- *prefix_identifier* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض الفئات سابقة التعريف، على سبيل المثال، env.
- *property_name* هي اسم خاصية المتطلبات الرئيسية، على سبيل المثال، path.jre

٢. باستخدام برنامج التحرير، قم بفتح الملف وتضمين المسار الى common_function.sh اذا كان يجب استخدام الوظائف العامة، كما يلي:

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

٣. قم باضافة الكود للحصول على قيمة خاصية المتطلبات الرئيسية باستخدام الأوامر والاختيارات الخاصة ببيئة التشغيل هذه للتوصل الى عناصر بيئة تشغيل النظام الرئيسي. على سبيل المثال، يتطلب أن تقوم خاصية المتطلبات الرئيسية env.path.jar المهيأة بالتحقق مما اذا كان قد تم تحديد JRE في المتغير PATH. يقوم الكود التالي بتشغيل الأمر env والبحث في المخرجات عن المتغير PATH ثم البحث في القيمة الخاصة به عن مسار JRE.

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

٤. تأكد من أن عملية التحقق تقوم بارجاع مخرجات قياسية:

```
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
```

في المثال، بناءا على قيمة المتغير $envJRE، تقوم عملية التحقق بارجاع True أو False:

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

٥. قم بتشغيل أداة التجميع المهيأة للتأكد من أنه لا توجد أخطاء في التشغيل وتصحيح الأخطاء كما يلزم.
٦. قم بتحرير *ips_root*/UNIX_Linux/packageTest.sh لاستدعاء وتشغيل أداة التجميع المهيأة.
٧. قم بتكوين أدوات التحقق المهيأة اذا كانت أداة التجميع المهيأة تقوم بارجاع قيم خلاف القيم البوولية.

## تحرير البرنامج النصي لاختبار مجموعة البرامج لأنظمة UNIX

يمكنك تحديث ملف البرنامج النصي `packageTest.sh` لاستدعاء أدوات تجميع مهيأ على أنظمة UNIX.

### قبل البدء

تأكد من معرفة أسماء أدوات التجميع المرفقة مع خصائص المتطلبات الرئيسية المعرفة مسبقا، كما هي موضحة في "الملحق د. أدوات التجميع سابقة التعريف لأنظمة UNIX". اذا كان قد تم اجراء المزيد من التصنيف على خاصية المتطلبات الرئيسية وفقا للتطبيق أو الوظيفة أو النوع الفرعي للخدمة، قم بامرار الفارق للنوع الفرعي *suffix_identifier*، الذي يمثل، *differentiator_suffix_identifier* الى أداة التجميع الخاصة به.

على سبيل المثال، `os.package` هي أداة التجميع العامة للتحقق من وجود مجموعات البرامج. للتحقق من وجود `openssh`، قم بامرار اسم مجموعة البرامج عند استدعاء أداة التجميع `os.package` في ملف البرنامج النصي `packageTest.sh`، كما يلي:

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

حيث `openssh` هو اسم مجموعة البرامج، الذي يمثل، النوع الفرعي *suffix_identifier* والفارق *differentiator_suffix_identifier*.

### الاجراء

١. استخدام برنامج تحرير، قم بفتح البرنامج النصي *ips_root*`/UNIX_Linux/packageTest.sh`.
٢. قم باضافة الكود لقراءة خاصية المتطلبات الرئيسية المهيأة من ملف التوصيف واجراء تحليل لغوي للقيمة الخاصة به.

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

على سبيل المثال، لقراءة خاصية المتطلبات الرئيسية `env.path.jar` المهيأة والتحقق مما اذا كان قد تم تحديد JRE في المتغير PATH:

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

في المثال:

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

٣. قم باستدعاء وظائف التسجيل لبيانات التتبع وتصحيح الأخطاء قبل استدعاء أداة التجميع المهيأة.

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

٤. قم باستدعاء أداة التجميع المهيأة.

**ملاحظة:** اذا كانت أداة التجميع المهيأة تحتوي على أنواع فرعية، التي تمثل، [*suffix_identifier*] في اسم الملف وكانت تتطلب عمليات تحقق اضافية بناءا على النوع الفرعي، قم بامرار الفارق [*differentiator_suffix_identifier*] للنوع الفرعي الى أداة التجميع المهيأة.

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

في المثال:

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

**ملاحظة:** أمثلة الفوارق للنوع الفرعي *script_name* لخاصية المتطلبات الرئيسية os.file.*script_name* هي المسارات للبرامج النصية التي تم امرارها الى أداة التجميع os.filepath:

```
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
```

٥. قم باستدعاء وظائف التسجيل لبيانات التتبع وتصحيح الأخطاء عند الخروج من أداة التجميع المهيأة.

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

في المثال:

```
 echo "ss=\`./env.path.jar\`" >>/tmp/prs.check
 echo "\`wrlTrace "Finished" "env.path.jar"\`"  >>/tmp/prs.check
 echo "echo \"env.path.jar=\$ss\"" >>/tmp/prs.check
 echo "\`wrlDebug "Finished" "env.path.jar"\`"  >>/tmp/prs.check
 echo "\`wrlDebug "OutPutValueIs" \$ss\`"  >>/tmp/prs.check
 echo "\`wrlTrace "Done" "env.path.jar"\`"  >>/tmp/prs.check
fi
```

٦. كرر الخطوات ٢ الى ٥ لكل من خصائص المتطلبات الرئيسية المهيأة.

## تكوين أدوات التحقق المهيأة لأنظمة Windows

يمكنك تكوين أدوات تقييم VBScript اذا كانت أدوات التقييم الأساسية لا تقوم بمقارنة القيم المتوقعة والفعلية لخصائص المتطلبات الرئيسية باستخدام معيار التقييم الصحيح. عند تكوين أدوات تقييم مهيأة، يجب أن ينتهي اسم الملف بواسطة _compare وتخزينه في الدليل الفرعي Windows/. يمكن أن تقوم أداة التقييم المهيأة باستخدام الوظائف العامة والبرامج المهيأة لمقارنة القيم اذا تطلب الأمر.

### قبل البدء

تأكد من مراجعة مجموعة الوظائف والبرامج الفرعية في الملاحق التالية قبل تكوين أدوات التقييم. حدد ما اذا كان يمكنك استخدام أي منها لمقارنة القيم:

- "الملحق ه. الوظائف المعتادة لأنظمة Windows"
- "الملحق ز. البرامج الفرعية لوظيفة الملفات لنظام Windows"
- "الملحق و. البرامج الفرعية لوظيفة التسجيل لأنظمة Windows"
- "الملحق ح. الوظائف المعتادة للبرامج الفرعية لأنظمة Windows"

**ملاحظة:** يمكن أن تقوم الوظيفة العامة، "passOrFail()"، بمقارنة القيم الفعلية والمتوقعة لأنواع البيانات التالية: رقم عام؛ الحجم بالميجابايت أو الجيجابايت؛ سرعة المشغل بالميجاهرتز أو الجيجاهرتز؛ قيمة بوولية؛ مجموعة حروف. يمكنك تكوين أداة تقييم مهيأة فقط اذا كان لا يمكن استخدام الوظيفة `passOrFail`.

**الاجراء**

١. قم بتكوين ملف VBScript. قم بحفظ الملف بالدليل *ips_root*`/Windows`، مع الاختلاف عن قواعد تسمية الملف التالية:

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

حيث:

- *prefix_identifier* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض التصنيفات سابقة التعريف.
- *property_name* هو اسم خاصية المتطلبات الرئيسية.
- *suffix_identifier* هو كود تعريف اختياري للنوع الفرعي لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٤.

٢. قم باضافة الكود لمقارنة القيم الفعلية والمتوقعة التي تم امرارها الى أداة التقييم كمتغيرات مستقلة باستخدام VBScript COM والوظائف المرفقة. تأكد من أن عملية المقارنة تقوم بارجاع مخرجات قياسية كما يلي:

- "PASS" عندما تكون القيمة المتوقعة لخاصية المتطلبات الرئيسية أكبر من أو تساوي القيمة الفعلية للخاصية
- "FAIL" عندما تكون القيمة المتوقعة لخاصية المتطلبات الرئيسية لا تساوي القيمة الفعلية للخاصية

٣. قم بتشغيل أداة التحقق المهيأة للتأكد من أنه لا توجد أخطاء في التشغيل وتصحيح الأخطاء كما يلزم.

**مثال**

تقوم أداة التقييم المهيأة هذه بالتحقق من القيم الفعلية والمتوقعة لنسخة Tivoli Directory Integrator. وتقوم باستخدام الوظيفة العامة، "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(expect,1),"-")=0)
Then
    if (Right(expect,1)="+" or Right(expect,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
' asstrings and not parsed into numeric form
'
' Returns
'       1 version1 > version2
'      -1 version1 &#60; 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 [" &#38; ver1 &#38; "] to [" &#38; ver2 &#38;
"]"

    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&#60;=ubound(ver1Parts) or i&#60;=ubound(ver2Parts))
    Dim v1, v2, v1Str, v2Str

    v1Str = UNASSIGNED
    v2Str = UNASSIGNED

    if (i&#60;=ubound(ver1Parts)) Then
        on error resume next
        v1 = Int(ver1Parts(i))
        if not Err=0 Then
            v1Str = ver1Parts(i)
            if (i&#60;=ubound(ver2Parts)) Then
                v2Str = ver2Parts(i)
            else
                v2Str = "0"
            end if
        end if
    else
        v1 = 0
    end if

    if (i&#60;=ubound(ver2Parts)) Then
        on error resume next
        v2 = Int(ver2Parts(i))
        if not Err=0 Then
            if (i&#60;=ubound(ver1Parts)) Then
```

```
                    v1Str = ver1Parts(i)
                else
                    v1Str = "0"
                end if
                v2Str = ver2Parts(i)
            end if
        else
            v2 = 0
        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
```

## تكوين أدوات التحقق المهيأة لأنظمة UNIX

يمكنك تكوين أدوات تجميع مهيأة اذا كانت أداة التجميع المهيأة لا تقوم بارجاع قيم بوولية، التي تمثل، `True` أو `False`. عند تكوين أدوات تقييم مهيأة، يجب أن ينتهي اسم الملف بواسطة `_compare` وتخزينه في الدليل الفرعي `./UNIX_Linux`. يمكن أن تقوم أداة التقييم المهيأة باستخدام الوظائف العامة لمقارنة القيم اذا تطلب الأمر.

### قبل البدء

تأكد من مراجعة مجموعة الوظائف في الملاحق التالية قبل تكوين أدوات التجميع المهيأة. حدد ما اذا كان يمكنك استخدام أي منها لمقارنة القيم الفعلية والمتوقعة:

- "الملحق ط. الوظائف المعتادة لأنظمة UNIX"
- "الملحق ي. الوظائف الأخرى لأنظمة UNIX"
- "الملحق ك. وظائف تسجيل الوظيفة لأنظمة UNIX"

يوجد اثنين من ملفات البرنامج النصي يمكنك استخدامها كنقطة بداية، التي تمثل، `._compare.sh` و `_compare.sh` في الدليل الفرعي `./Unix_Linux`.

**هام:** لا تقم بتكوين أدوات تقييم مهيأة اذا قامت أدوات التجميع المهيأة الخاصة بك بارجاع `True` أو `False`. تقوم IBM Prerequisite Scanner باستخدام أدوات تقييم معرفة مسبقا لأية أداة تجميع تقوم بارجاع قيم بوولية.

### الاجراء

١. قم بتكوين ملف shell. قم بحفظ الملف بالدليل `/UNIX_Linux` *ips_root*، مع الاختلاف عن قواعد تسمية الملف التالية:

`[`*prefix_identifier*`.]`*property_name*`[.`*suffix_identifier*`]_compare.sh`

حيث:

- *prefix_identifier* هو كود تعريف لتصنيف سابق التعريف لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٣. بادئة كود التعريف هذه تعتبر مطلوبة لبعض التصنيفات سابقة التعريف.
- *property_name* هو اسم خاصية المتطلبات الرئيسية.
- *suffix_identifier* هو كود تعريف اختياري للنوع الفرعي لخصائص المتطلبات الرئيسية كما هو موضح في الجدول ٤.

٢. قم باضافة الكود لمقارنة القيم الفعلية والمتوقعة التي تم امرارها الى أداة التقييم كمتغيرات مستقلة ووظائف مرفقة. تأكد من أن عملية المقارنة تقوم بارجاع مخرجات قياسية كما يلي:
   - `"PASS"` عندما تكون القيمة المتوقعة لخاصية المتطلبات الرئيسية أكبر من أو تساوي القيمة الفعلية للخاصية
   - `"FAIL"` عندما تكون القيمة المتوقعة لخاصية المتطلبات الرئيسية لا تساوي القيمة الفعلية للخاصية

٣. قم بتشغيل أداة التحقق المهيأة للتأكد من أنه لا توجد أخطاء في التشغيل وتصحيح الأخطاء كما يلزم.

# الفصل الرابع. تشغيل Prerequisite Scanner

يمكنك استخدام واجهة تعامل سطر الأمر لتشغيل IBM Prerequisite Scanner. يأخذ البرنامج النصي الى Prerequisite Scanner و prereq_checker، مجموعة من المعاملات المطلوبة والاختيارية وشارة الأمر للمعاملات الاضافية الاختيارية.

يوضح الجدول ١٢ الحروف الخاصة المستخدمة في صيغة البرنامج النصي الى Prerequisite Scanner.

*الجدول ١٢. وسيلة ايضاح الحروف الخاصة للبرنامج النصي الى Prerequisite Scanner*

| الحرف الخاص | الوصف |
|---|---|
| < > | يقوم بتعريف اسم المكان المحتجز. |
| [ ] | يقوم بتعريف معامل اختياري. المعاملات التي لا يتم تضمينها بين أقواس تكون مطلوبة. |
| . . . | يشير الى أنه يمكنك تحديد عدة قيم لمعامل. |
| \| | يشير الى المعاملات المتنافية. يمكنك تحديد المعامل يسار الفاصل أو المعامل يمين الفاصل، لكن ليس كلاهما. |
| { } | يقوم بتضمين مجموعة المعاملات المتنافية المفصولة بواسطة \| . |

## prereq_checker

يقوم البرنامج النصي prereq_checker بتشغيل IBM Prerequisite Scanner والتحقق من المتطلبات الرئيسية بناءا على مجموعة المعاملات التي تقوم بتحديدها عند تشغيل البرنامج النصي.

### الصيغة

```
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]
```

البرنامج prereq_checker له معامل مطلوب واحد والعديد من المعاملات الاختيارية.

**""*Product_Code* [*Product_Version*]|,*Product_CodeN* [*Product_VerN*]|...""**
معامل مطلوب

**"[detail]"**
معامل اختياري

**"[outputDir="*ips_output_dir*"]"**
معامل اختياري

**"[xmlResult]"**
معامل اختياري

**"[PATH="*product_root*"]"**

**"[-p *Product_Code.instance.parameter=value,...]"**
شارة اختيارية

**"[debug]"**
معامل اختياري

**"[trace]"**
معامل اختياري

***"Product_Code [Product_Version][,Product_CodeN [Product_VerN]]..."***

يجب تحديد معامل **`Product_Code`** واحد على الأقل لتعريف البرنامج أو المكون الذي سيتم تشغيل عملية التحقق من المتطلبات الرئيسية له وملف التوصيف المرفق. **`Product_Code`** هو كود البرنامج الذي تقوم بتحديده في الملف *ips_root*`/codename.cfg`.

على سبيل المثال، KMS هو كود البرنامج الى Tivoli Enterprise Monitoring Server في الملف `product.cfg`.
لتشغيل الماسح الضوئي، قم بادخال البرنامج التالي بكود البرنامج:

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

اذا قمت بتحديد معامل **`Product_Code`** الذي لا يحتوي على ملف توصيف مناظر، ستقوم أداة Prerequisite Scanner بتجاهله بدون حدوث خطأ. يحتوي ملف السجل على رسالة بأنه لم يتم ايجاد أي ملف توصيف.

يشير المعامل **`Product_Version`** للمعامل **`Product_Code`** المرفق الى نسخة البرنامج. وهو يعد كود من ٨-أرقام لتمثيل النسخة والاصدار والتعديلات والمستوى، مع رقمين لكل جزء من الكود؛ على سبيل المثال، 7.3.21 هو 07032100.
**`Product_Version`** هو معامل اختياري. اذا لم تقم بتحديده، ستقوم أداة Prerequisite Scanner بالتحقق من أحدث نسخة متاحة.

يمكنك تحديد واحد أو عدة معاملات **`Product_Code`** مع المعامل الاختياري **`Product_Version`**، مع فصل كلا منهما بفاصلة.

**هام:** عند تحديد أكثر من معامل **`Product_Code`** واحد باستخدام المعامل **`Product_Version`** الاختياري، قم بتضمين المعاملات بين علامات تنصيص. اذا لم تقم بذلك، لن تتم عمليات الماسح الضوئي بنجاح.

يقوم هذا المثال بالتحقق من المتطلبات الرئيسية لأحدث نسخة الى Tivoli Monitoring Operating System Agent for Windows و النسخة 6.2.1 الى Tivoli Monitoring Agent for DB2.

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

**[detail]**

هذا المعامل الاختياري يشير الى ما اذا كان سيتم عرض النتائج التفصيلية للمسح الضوئي في واجهة تعامل سطر الأمر.

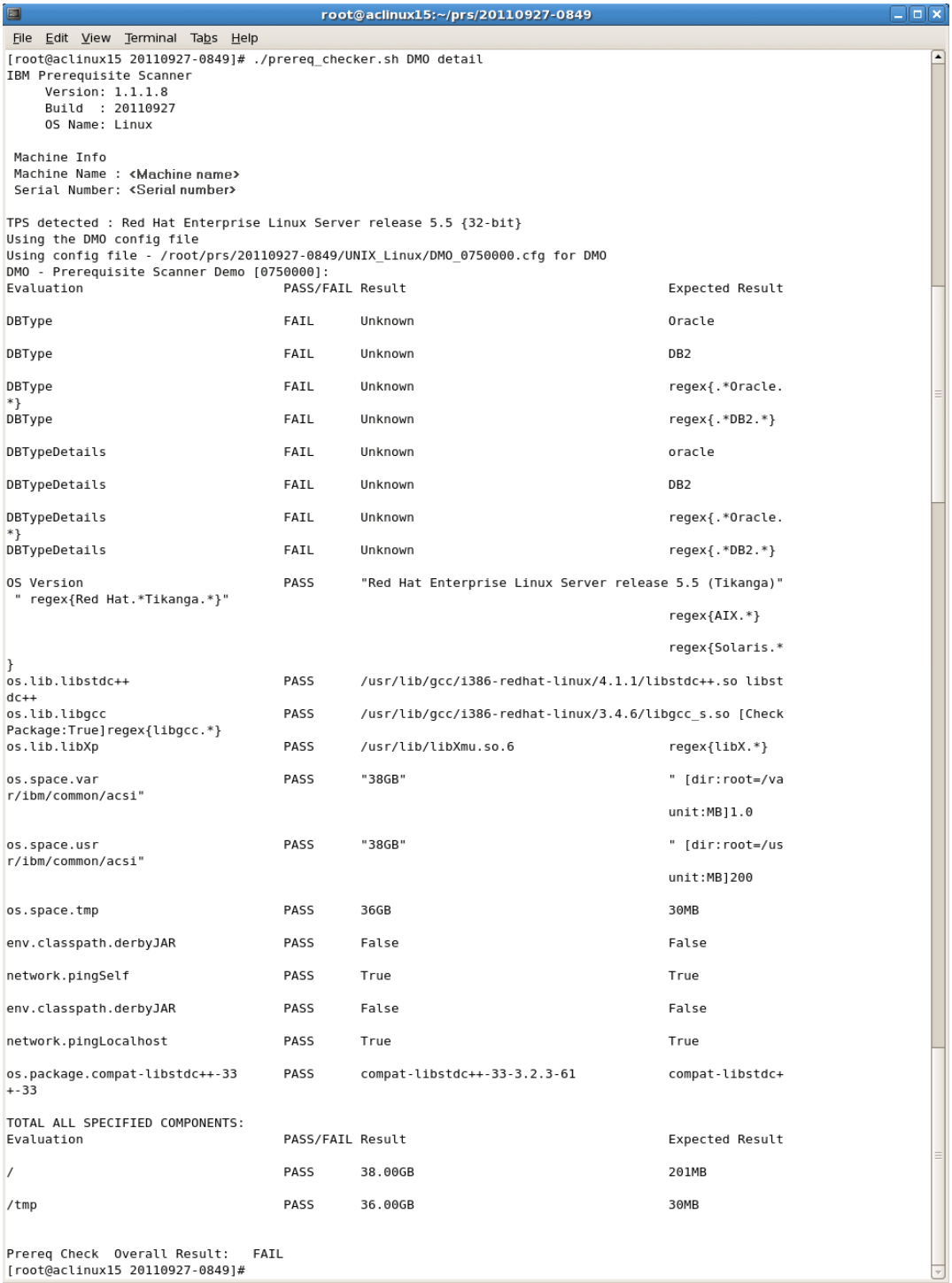
**هام:** لا تحاول تضمين هذا المعامل بين علامات تنصيص.

عندما تقوم بتحديد المعامل **`detail`**، فان النتائج التفصيلية تتضمن:

- نسخة Prerequisite Scanner
- نسخة نظام التشغيل التي تم تشغيل الماسح الضوئي عليها
- اسم البرامج أو المكونات التي سيتم تشغيل عمليات التحقق من المتطلبات الرئيسية عليها
- بالنسبة لكل خاصية للمتطلبات الرئيسية: سيتم التحقق من اسم خاصية المتطلبات الرئيسية، نتيجة PASS أو FAIL، القيمة الفعلية، القيمة المتوقعة

- بالنسبة لكل المكونات: سيتم التحقق من اسم خاصية المتطلبات الرئيسية العامة، نتيجة PASS أو FAIL، القيمة الفعلية، القيمة المتوقعة
- نتيجة PASS أو FAIL العامة

ستقوم Prerequisite Scanner أيضا بحفظ هذه النتائج في ملف *ips_output_dir*/result.txt. سيقوم بحفظ النتائج في الملف النصي بصرف النظر سواء قمت بتحديد المعامل **detail** أم لا.




الشكل ١٠. تشغيل البرنامج النصي وتحديد معامل التفاصيل في أنظمة UNIX

اذا لم تقم بتحديد المعامل **detail**، سيقوم الماسح الضوئي بعرض نتيجة PASS أو FAIL في واجهة تعامل سطر الأمر.

الشكل ١١. تشغيل البرنامج النصي بدون تحديد معامل التفاصيل في أنظمة Windows

### [outputDir="*ips_output_dir*"]

يشير هذا المعامل الاختياري الى أنك تريد تحديد دليل المخرجات لنتائج المسح وملفات السجل الى Prerequisite Scanner.

عند تشغيل البرنامج النصي Prerequisite Scanner وتحديد المعامل **outputDir** الاختياري، ستقوم أداة Prerequisite Scanner باخراج الملفات النصية وملفات XML وملفات السجل للنتائج الى الدليل المحدد بواسطة قيمة المعامل. تعرف هذه القيمة باسم *ips_output_dir* خلال المطبوعات الفنية.

اذا لم تقم بتحديد هذا المعامل، فان مكان المخرجات المفترض هو *ips_root*.

يجب استخدام المعامل لتحديد مكان، اذا قمت باختيار تشغيل Prerequisite Scanner من CD أو DVD أو مشغل شبكة اتصال للقراءة فقط. يجب أن تتوافر لديك صلاحيات للكتابة لتتمكن من الكتابة في *ips_output_dir*؛ والا فسيحدث فشل في Prerequisite Scanner.

**هام:** اذا كان دليل المخرجات غير موجود، ستقوم أداة Prerequisite Scanner بتكوين الدليل. يجب أن يكون لديك صلاحيات بالكتابة لتكوين أو الكتابة في دليل المخرجات الذي يقوم Prerequisite Scanner بحفظ الملفات به.

### [xmlResult]

هذا المعامل الاختياري يقوم بتحديد أنك تريد اخراج النتائج في ملف XML للنتائج بالاضافة الى ملف نتائج الاختبار العادي.

عندما تقوم بتشغيل Prerequisite Scanner وتحديد المعامل **xmlResult** الاختياري، ستقوم أداة Prerequisite Scanner باخراج النتائج الى ملف *ips_output_dir*/result.xml.

اذا لم تقم بتحديد هذا المعامل، سيتم اخراج النتائج الى ملف نصي عادي فقط.

### [PATH="*product_root*"]

هذا المعامل الاختياري يحدد أدلة التركيب للبرامج.

**هام:** في نظام Windows، لا تحاول تحديد المسار للحرف الخاص بقرص التشغيل فقط، حيث أنه C:. تأكد من أنك تقوم بتحديد مسار صحيح.

اذا لم تقم بتحديد المعامل **path**، سيقوم الماسح الضوئي بالتحقق من أدلة التركيب المفترضة لمنتجات IBM Tivoli:

- على أنظمة **UNIX**: /opt/ibm/itm
- على أنظمة **Windows**: C:\IBM\itm

**[-p *Product_Code.instance.parameter=value,...*]**

تشير شارة **-p** الاختيارية الى أنه يجب امرار معاملات الاستمرار الى ملف برنامج نصي لمزيد من عمليات التحقق من المتطلبات الرئيسية الاضافية. **<Product_Code>** هو كود المنتج. يتم امرار كل مجموعة من `instance.parameter=value` فقط الى البرنامج النصي. يمكنك امرار عدة مجموعات من المعاملات، مفصولة بفاصلة.

يتم تحديد البرنامج النصي الذي يتم امرار المعاملات اليه بواسطة الاختيارات التالية:

- باستخدام بادئة **Product_Code**، يتم امرار المعاملات الى البرنامج النصي مع **Product_Code** المرفق
- بدون البادئة **Product_Code**، يتم امرار المعاملات الى أدوات التجميع العامة.

مثال ١
`-p KUD.inst1.DB2_INST_OWNER=db2inst1, KUD.inst2.DB2_INST_OWNER=db2inst2`
قامت هذه الشارة ذات المعاملات بامرار `db2inst1.DB2_INST_OWNER=db2inst1` و `db2inst2.DB2_INST_OWNER=db2inst2` الى ملف البرنامج النصي `KUD.`**`Product_Version`**`.bat`.

مثال ٢

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

تم امرار هذه الشارة ذات المعاملات `SERVER=IP.PIPE://mymachine:1918` الى أداة التجميع العامة للتحقق من المنافذ.

**ملاحظة:** يقوم هذا البرنامج النصي بقبول المعاملات في **-p** على أنها `tacmd createNode`.

يمكنك تحديد المعاملات SERVER و PROTOCOL و PORT و BACKUP و BSERVER في *ips_root*`/lib/common_configuration`. تقوم أداة Prerequisite Scanner بتحديد أولوية المعاملات التي تم امرارها من واجهة تعامل سطر الأمر أعلى هذه المعاملات المحددة في الملف `common_configuration`.

**[debug]**

يشير هذا المعامل الاختياري الى أنك تريد تشغيل خاصية تصحيح الأخطاء أثناء تشغيل Prerequisite Scanner.

عند تشغيل البرنامج النصي Prerequisite Scanner وتحديد المعامل **debug** الاختياري، ستقوم Prerequisite Scanner باخراج معلومات التشغيل التفصيلية، رسائل التحذير والخطأ، نتائج المسح في ملف السجل. وهو ملف *ips_output_dir*`/prs.debug` على أنظمة UNIX وملف *ips_output_dir*`/precheck.log` على أنظمة Windows.

**هام:** يتم ايقاف تشغيل خاصية تصحيح أخطاء الماسح الضوئي بصفة مفترضة.

**[trace]**

(أنظمة UNIX فقط) يشير هذا المعامل الاختياري الى أنك تريد تشغيل خاصية تسجيل بيانات التتبع أثناء تشغيل Prerequisite Scanner.

عندما تقوم بتشغيل Prerequisite Scanner وتحديد المعامل **trace** الاختياري، ستقوم أداة Prerequisite Scanner باخراج معلومات التتبع في ملف *ips_output_dir*`/prs.trc`.

**هام:** يتم ايقاف خاصية تسجيل بيانات التتبع للماسح الضوئي بصفة مفترضة.

## تشغيل Prerequisite Scanner من سطر الأمر

يمكنك تشغيل IBM Prerequisite Scanner من واجهة تعامل سطر الأمر وادخال معاملات المدخلات المتعلقة للبرنامج النصي.

### قبل البدء

تأكد من أنه يمكنك الرجوع الى المطبوعات الفنية لتركيب البرنامج لأي من الخطوات الاضافية والتي يجب تنفيذها قبل تشغيل Prerequisite Scanner. على سبيل المثال، قد تحتاج الى تحديد متغير بيئة التشغيل التي تحدد الى Prerequisite Scanner ما هي المكونات أو الخصائص التي يتم تركيبها على الحاسب المستهدف وبالتالي المتطلبات الرئيسية التي سيتم التحقق منها.

### الاجراء

١. قم بفتح واجهة تعامل سطر الأمر وفتح الدليل *ips_root*.
٢. قم بتشغيل ملف البرنامج النصي Prerequisite Scanner، **prereq_checker**، كما يلي:

**نظام 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,...]
```

يقوم المثال التالي بتشغيل Prerequisite Scanner for Autonomic Deployment Engine باستخدام ملف التوصيف وكود المنتج المرفق الخاص به، ADE:

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

**نظام 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,...]
```

يقوم المثال التالي بتشغيل Prerequisite Scanner for Tivoli Provisioning Manager for Windows 2003 and 2008 باستخدام أكواد المنتج، COX و COY.

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

يقوم المثال التالي بتشغيل Prerequisite Scanner for Tivoli zEnterprise® Monitoring Agent باستخدام كود المنتج KZE. ويقوم أيضا بتحديد مكان النتائج وملفات السجلات الى *ips_output_dir* باستخدام معامل **outputDir** الاختياري.

**هام:** يجب استخدام المعامل **`outputDir`** لتحديد مكان، اذا قمت باختيار تشغيل Prerequisite Scanner من CD أو DVD أو مشغل شبكة اتصال للقراءة فقط. يجب أن تتوافر لديك صلاحيات للكتابة لتتمكن من الكتابة في *ips_output_dir*؛ والا فسيحدث فشل في Prerequisite Scanner.

**Windows**
```
prereq_checker.bat
"KZE 06230000"
outputDir="%TEMP%\ips"
```
**UNIX**
```
./prereq_checker.sh
"KZE 06230000"
outputDir="/tmp/ips"
```

يقوم الماسح الضوئي باخراج ملفات `result.txt file` و `precheck.log` الى الأماكن التالية:

- على أنظمة Windows: `D:\temp\ips` حيث `TEMP` هو متغير بيئة التشغيل للحافظة المؤقتة.
- على أنظمة UNIX: `/tmp/ips`

**هام:** اذا كان دليل المخرجات غير موجود، ستقوم أداة Prerequisite Scanner بتكوين الدليل. يجب أن يكون لديك صلاحيات بالكتابة لتكوين أو الكتابة في دليل المخرجات الذي يقوم Prerequisite Scanner بحفظ الملفات به.

---

## أماكن الدليل المعتادة

توجد متغيرات اسم المسار للأدلة العامة.

### أدلة تركيب IBM Prerequisite Scanner

يقدم *`ips_root`* شرح للمكان الذي تم تركيب Prerequisite Scanner عليه. يمكن تحديد هذا المكان أثناء عملية التركيب.

### دليل مخرجات Prerequisite Scanner

*`ips_output_dir`* توضح المكان الذي تم به حفظ نتائج عملية المسح وملفات السجل الى Prerequisite Scanner. هذا المكان يمكن تحديده من خلال استخدام معامل المدخلات **`outputDir`** عند تشغيل الماسح الضوئي. اذا لم تقم بتحديد هذا المعامل، فان مكان المخرجات المفترض هو *`ips_root`*.

**ملاحظة:** يقوم Prerequisite Scanner بتكوين الملفات المؤقتة أثناء التنفيذ، ولكن يتم حذف هذه الملفات قبل اتمام الماسح الضوئي لتنفيذها. هذه الملفات المؤقتة توجد في الدليل الفرعي *`ips_output_dir`*`/temp`. كما يقوم الماسح الضوئي بحذف الدليل الفرعي *ips_output_dir*`/temp`، ما لم يكن الدليل الفرعي يتضمن ملفات التتبع وتصحيح الأخطاء التي تم تكوينها في أنظمة UNIX فقط.

# الفصل الخامس. تصحيح أخطاء Prerequisite Scanner

يمكنك اكتشاف وتصحيح المشاكل التي توجد في IBM Prerequisite Scanner باستخدام ملفات السجل ووظائف التسجيل عند تكوين عمليات التحقق المهيأة من المتطلبات الرئيسية.

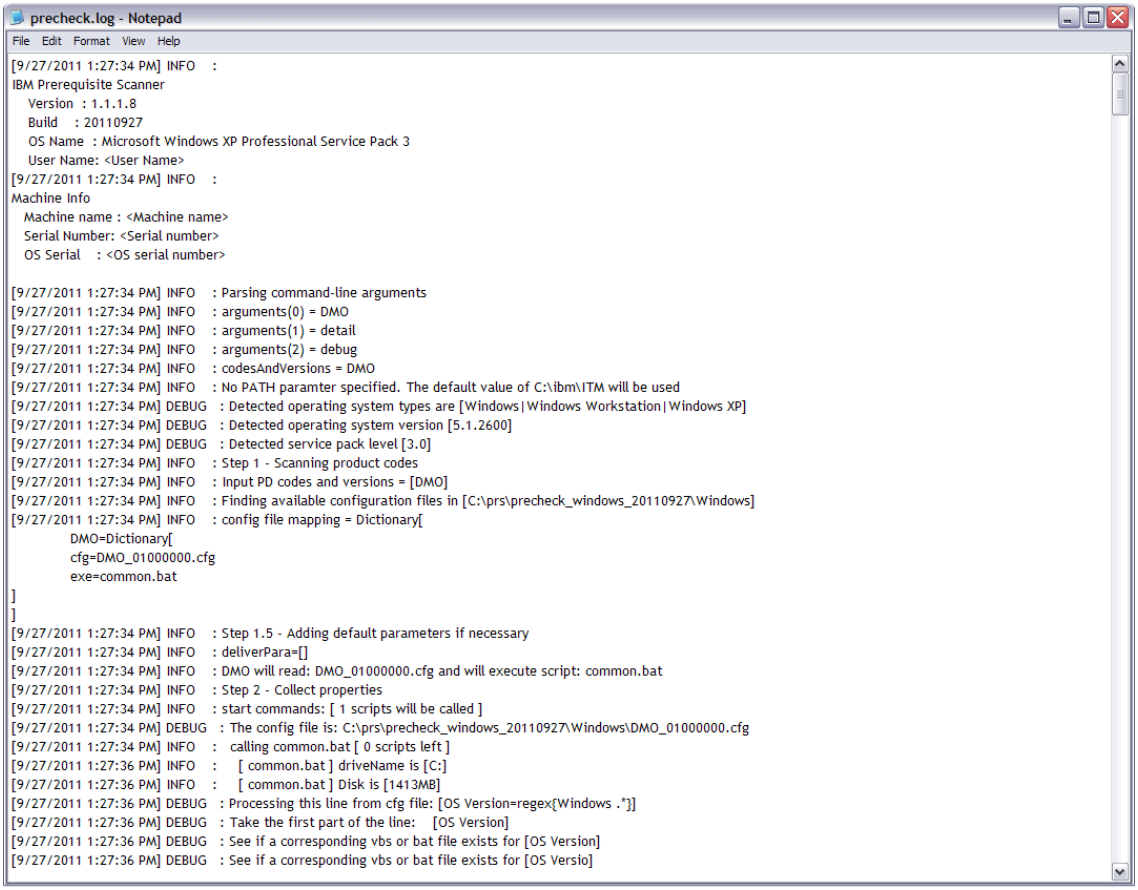
Prerequisite Scanner تقوم بانتاج أكواد مرتجعة تتوقف على نتائج عملية المسح وما اذا كان يجب الخروج بسبب الأخطاء أم لا. يتم كتابة الأكواد المرتجعة هذه في ملفات السجل. على سبيل المثال، اذا فشل Prerequisite Scanner في تشغيل عملية التحقق لأنه لا يمكن قراءة ملف التوصيف، وستقوم بتكوين كود 2 الناتج.

## تصحيح الأخطاء على أنظمة Windows

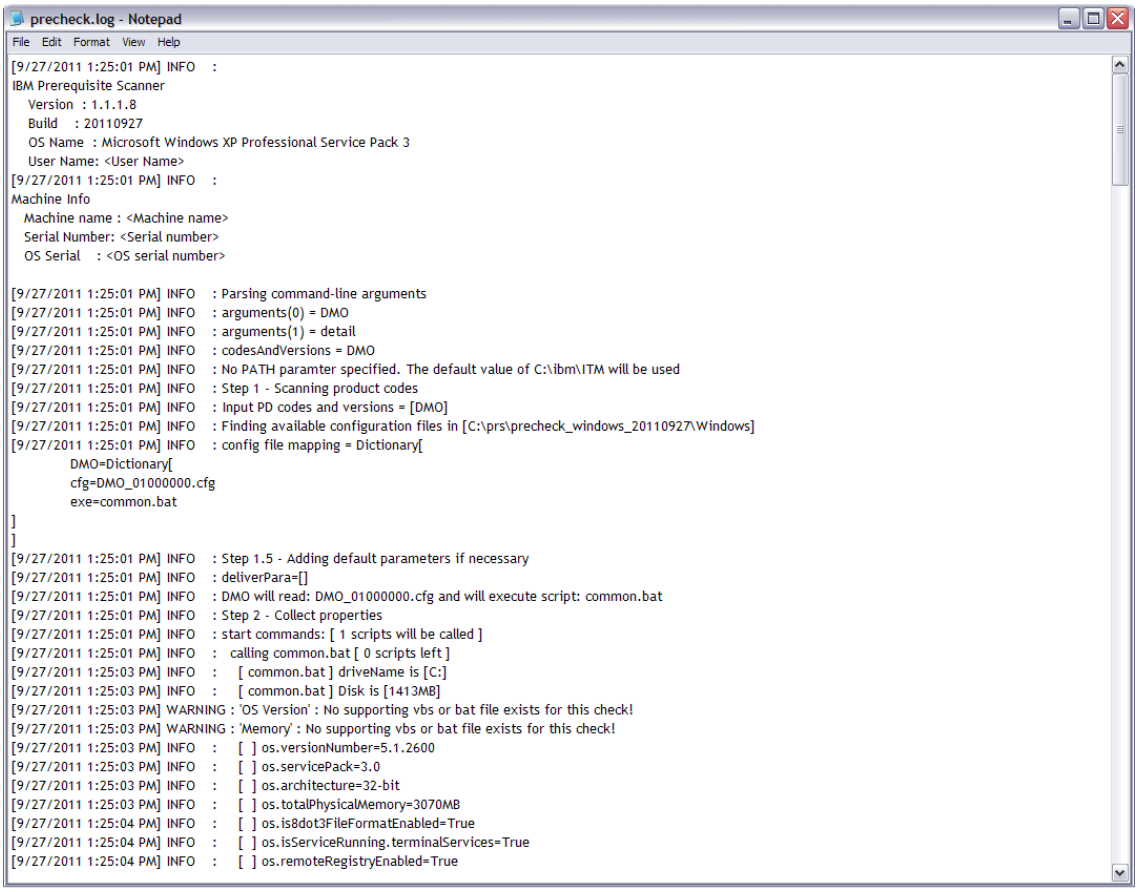
عند تشغيل IBM Prerequisite Scanner، يقوم بتكوين ملف السجل بصفة مفترضة. وتتضمن المعلومات المفصلة مع كل من الخطوات والوظائف التي يقوم الماسح الضوئي بتنفيذها بالترتيب. كما يتضمن الملف على خاتم الوقت، بما يتضمن أوقات البدء والانتهاء لكل من الوظائف والخطوات. يمكنك تصحيح أخطاء ومراجعة ملف السجل لتحديد مكان ووقت حدوث الخطأ.

يقوم Prerequisite Scanner باخراج معلومات التشغيل، ورسائل الخطأ والتحذير، ونتائج عملية المسح في ملف *ips_output_dir*/precheck.log. عندما تقوم بتشغيل البرنامج Prerequisite Scanner وتحديد المعامل **debug** الاختياري، سيقوم Prerequisite Scanner باخراج المزيد من رسائل تصحيح الأخطاء في هذا الملف.

يعرض الشكل ١٢ مثال لملف السجل عند تحديد معامل **debug** الاختياري ويعرض الشكل ١٣ ملف السجل عند عدم تحديد المعامل.




الشكل ١٢. ملف precheck.log مع بيانات تصحيح الأخطاء



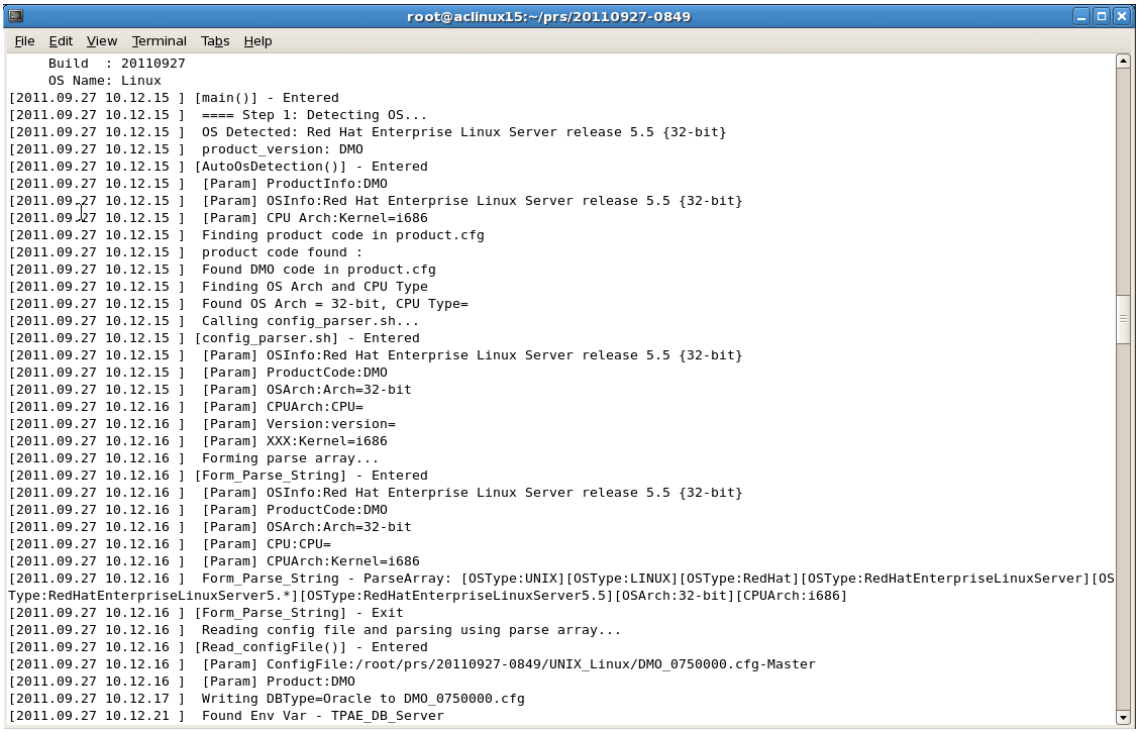

الشكل ١٣. ملف precheck.log بدون بيانات تصحيح الأخطاء

## تصحيح الأخطاء على أنظمة UNIX

يتم الغاء اتاحة خاصية كتابة الرسائل في ملفات السجل على أنظمة UNIX بصفة مفترضة. يمكنك اتاحة وظائف تصحيح الأخطاء أو التتبع باستخدام معاملات المدخلات **debug** و **trace**. يقوم الماسح الضوئي بكتابة بيانات تصحيح الأخطاء والتتبع في ملفات سجلات مختلفة واستخدام خاتم الوقت لتعليم أوقات بدء وانتهاء الخطوات أو الوظائف. يمكنك استخدام كلا من الملفين لربط وتصحيح أخطاء مشكلة أو وظيفة أو عملية تحقق من المتطلبات الرئيسية معينة.

### ملف السجل لتصحيح الأخطاء

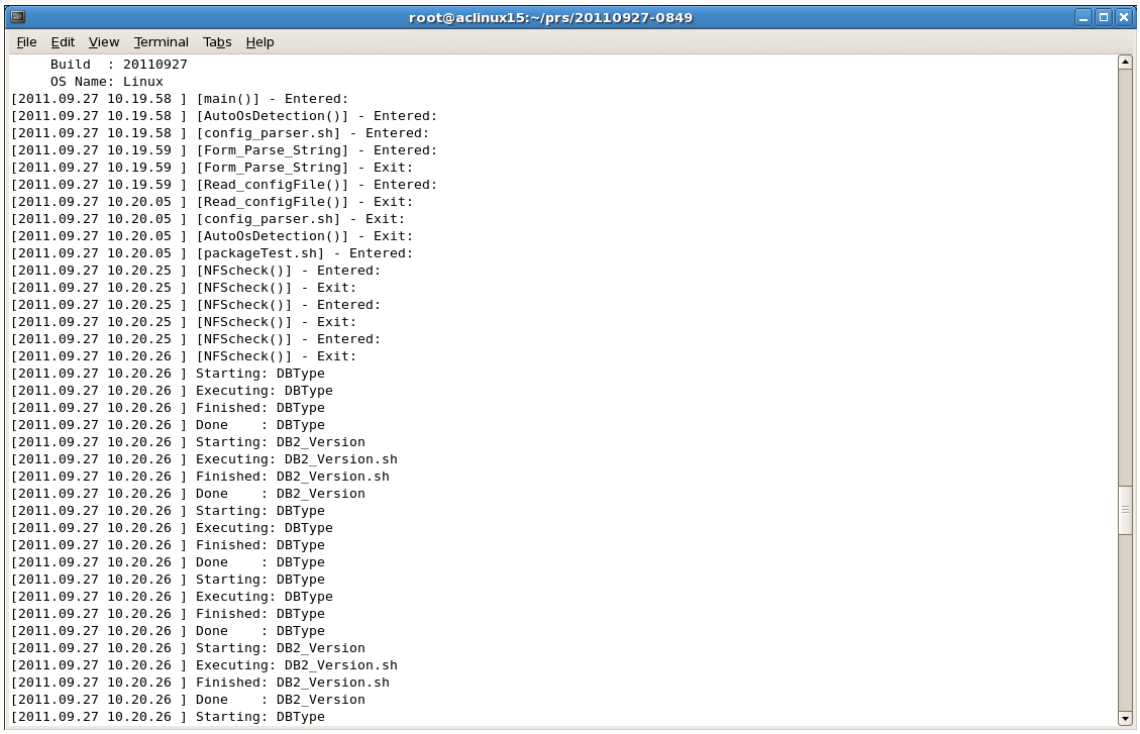
عندما تقوم بتشغيل Prerequisite Scanner وتحديد المعامل **debug** الاختياري، ستقوم أداة Prerequisite Scanner باخراج معلومات التشغيل التفصيلية، ورسائل التحذير والخطأ، ونتائج عملية المسح في ملف *ips_output_dir*`/temp/prs.debug`. وتتضمن المعلومات المفصلة مع كل من الخطوات والوظائف التي يقوم الماسح الضوئي بتنفيذها بالترتيب. كما يتضمن الملف على خاتم الوقت، بما يتضمن أوقات البدء والانتهاء لكل من الوظائف والخطوات.
يحتوي الدليل الفرعي *ips_output_dir*`/temp` أيضا على الملفات `result1.txt` و `result2.txt` المؤقتة التي تقدم المدخلات لملف *ips_output_dir*`/result.txt` النهائي. يمكنك استخدام هذه الملفات المؤقتة لتحديد المشاكل في النتائج لعمليات تحقق من المتطلبات الرئيسية معينة.




الشكل ١٤. ملف prs.debug على أنظمة UNIX

### ملف السجل للتتبع

عندما تقوم بتشغيل Prerequisite Scanner وتحديد المعامل **`trace`** الاختياري، ستقوم أداة Prerequisite Scanner باخراج معلومات التتبع في ملف *ips_output_dir*`/temp/prs.trc`. وتتضمن المعلومات مع كل من الوظائف التي يقوم الماسح الضوئي بتنفيذها بالترتيب. كما يتضمن الملف على خاتم الوقت، بما يتضمن أوقات البدء والانتهاء لكل من الوظائف.




الشكل ١٥. ملف prs.trc على أنظمة UNIX

## مشاكل التنفيذ

يمكنك استخدام كشف مشاكل التنفيذ لتصحيح الأخطاء التي قد تواجهها عند تشغيل Prerequisite Scanner.

قم بتشغيل البرنامج النصي لأداة Prerequisite Scanner باستخدام معاملات المدخلات الاختيارية **debug** و **trace** لمساعدتك في تصحيح الأخطاء.

*الجدول ١٣. كشف مشاكل التنفيذ*

| التحقق | البند |
|---|---|
| ☐ | عند تحديد المعامل **outputDir** الاختياري بسطر الأمر ولم يكن دليل المخرجات موجودا، ستقوم أداة Prerequisite Scanner بتكوين الدليل. يجب أن يكون لديك صلاحيات بالكتابة لتكوين أو الكتابة في دليل المخرجات الذي تقوم Prerequisite Scanner بحفظ الملفات به. اذا لم يكن يتوافر لديك تصاريح الكتابة، سيتم كتابة رسالة الخطأ التالية في واجهة تعامل سطر الأمر:<br>`ERROR: Cannot create files in output directory` *ips_output_dir*.<br>`Exit.` |
| ☐ | قبل تشغيل Prerequisite Scanner، تأكد من أن القرص الذي تريد تشغيل Prerequisite Scanner عليه وحفظ النتائج لدليل المخرجات به غير كامل؛ خلاف ذلك، سيتم كتابة رسالة الخطأ التالية في واجهة تعامل سطر الأمر:<br>`ERROR: Cannot create files in output directory` *ips_output_dir*.<br>`Exit.` |
| ☐ | اذا قامت Prerequisite Scanner بتكوين كود الناتج 2، قد يحدث خطأ في استخدام البرنامج النصي أو أداة التجميع. قم بمراجعة الأسباب المرفقة مع كود الخطأ هذا. اذا حدث خطأ في استخدام البرنامج النصي، أعد تشغيل Prerequisite Scanner باستخدام الصيغة الصحيحة. |

**المفاهيم المتعلقة:**

يتم الغاء اتاحة خاصية كتابة الرسائل في ملفات السجل على أنظمة UNIX بصفة مفترضة. يمكنك اتاحة وظائف تصحيح الأخطاء أو التتبع باستخدام معاملات المدخلات **debug** و **trace**. يقوم الماسح الضوئي بكتابة بيانات تصحيح الأخطاء والتتبع في ملفات سجلات مختلفة واستخدام خاتم الوقت لتعليم أوقات بدء وانتهاء الخطوات أو الوظائف. يمكنك استخدام كلا من الملفين لربط وتصحيح أخطاء مشكلة أو وظيفة أو عملية تحقق من المتطلبات الرئيسية معينة.

Prerequisite Scanner تقوم بانتاج أكواد مرتجعة تتوقف على نتائج عملية المسح وما اذا كان يجب الخروج بسبب الأخطاء أم لا. يتم كتابة الأكواد المرتجعة هذه في ملفات السجل.

يقوم البرنامج النصي `prereq_checker` بتشغيل IBM Prerequisite Scanner والتحقق من المتطلبات الرئيسية بناءا على مجموعة المعاملات التي تقوم بتحديدها عند تشغيل البرنامج النصي.

## الأكواد الناتجة

Prerequisite Scanner تقوم بانتاج أكواد مرتجعة تتوقف على نتائج عملية المسح وما اذا كان يجب الخروج بسبب الأخطاء أم لا. يتم كتابة الأكواد المرتجعة هذه في ملفات السجل.

تقوم أداة Prerequisite Scanner بتكوين الأكواد الناتجة بناءا على مجموعة النتائج التي تم تعريفها كما يلي:

| الكود الناتج | الوصف |
|---|---|
| 0 | يتم ارجاع 0 عند تشغيل Prerequisite Scanner بنجاح وعندما تكون كل نتائج عملية المسح هي PASS. |
| 1 | يتم ارجاع 1 عند تشغيل Prerequisite Scanner بنجاح، لكن واحدة أو أكثر من عمليات التحقق من المتطلبات الرئيسية قامت بارجاع FAIL. |
| 2 | يتم ارجاع 2 عند عدم تشغيل Prerequisite Scanner بنجاح، ويجب الخروج بسبب خطأ تم تصنيفه كما يلي:<br>• أخطاء في استخدام البرنامج النصي<br>• أخطاء في أداة التجميع<br>• أخطاء أخرى |

### أخطاء في استخدام البرنامج النصي

يمكن أن تقوم أداة Prerequisite Scanner بالخروج بسبب أي من أخطاء الاستخدام التالية عند تشغيل البرنامج النصي:

- معامل المدخلات **Product_Code** غير صحيح؛ على سبيل المثال، لم يتم ايجاده أو ليس نسق مدعم.
- نموذج معاملات المدخلات **Product_Code** و **Product_Version** غير صحيح؛ على سبيل المثال، تم ادخال أكثر من الكود والنسخة بين علامات تنصيص، أو لم يتم تضمين النموذج بين علامات تنصيص.
- معاملات المدخلات **Product_Version** غير صحيحة؛ على سبيل المثال، نسخة المنتج ليس حروف رقمية بالكامل.
- لم يتم ادخال أية معاملات مدخلات في واجهة تعامل سطر الأمر.
- الصيغة لم تكن صحيحة عند ادخالها في واجهة تعامل سطر الأمر؛ على سبيل المثال، تم ادخال متغير مستقل لسطر الأمر غير مدعم.
- لم يتم ادخال أي معامل مدخلات **Product_Code** مطلوب.

### أخطاء في أداة التجميع

يمكن أن تقوم أداة بالخروج من Prerequisite Scanner بسبب أي من أخطاء أداة التجميع التالية:

- لم يتم ايجاد ملف النتائج المؤقت لأداة التجميع في الدليل *ips_output_dir*`/temp`.
- لم يتم تشغيل ملف البرنامج النصي لأداة التجميع بطريقة صحيحة.

**أخطاء أخرى**

يمكن أن تقوم أداة Prerequisite Scanner بالخروج، بسبب عدم توافر تصريح الكتابة لدى المستخدم لدليل المخرجات *ips_output_dir*.

**المفاهيم المتعلقة:**

يقدم IBM Prerequisite Scanner مخرجات للشاشة ونسق الملف القابلة للقراءة التالية: مخرجات لواجهة تعامل سطر الأمر، ملفات السجل الخاصة بتصحيح وتتبع الأخطاء، ملفات نصية و XML للنتائج.

# الملحق أ. مرجع أكواد البرنامج

تقوم أداة IBM Prerequisite Scanner باستخدام كود متعدد الحروف، *product_code*، لتعريف المنتج وبيئة تشغيل مدعمة منفردة ونسخة نظام التشغيل. يحتوي الملف *ips_root*/codename.cfg على زوج قيم الاسم لتمثيل كود المنتج وبيئة التشغيل المدعمة الخاصة به ونسخة نظام التشغيل.

يوضح الجدول ١٤ مجموعة أكواد المنتج المعرفة مسبقا الحالية.

**قيد:** IBM Tivoli Monitoring و Tivoli Composite Application Manager له أكواد برنامج سابقة التعريف يقوم Prerequisite Scanner باعتبارها محتجزة. لا يجب استخدام هذه الأكواد كأكواد لبرنامج Prerequisite Scanner ما لم تكن تقوم بالاشارة الى IBM Tivoli Monitoring و Tivoli Composite Application Manager الخاصة بها. للحصول على المزيد من المعلومات عن أكواد البرنامج، أنظر ITM 6.X Product Codes Technote.

**قيد:** بنظام UNIX فقط: عند ادخال قيمة لكود المنتج في الملف، تجنب استخدام for. حيث تعد كلمة محجوزة ويمكن أن تؤثر على كيفية تشغيل Prerequisite Scanner.

*الجدول ١٤. أكواد المنتج المعرفة مسبقا*

| أكواد البرنامج سابقة التعريف | بيئة التشغيل | نسخة البرنامج، بيئة التشغيل، نظام التشغيل |
|---|---|---|
| ADE | كل | Autonomic Deployment Engine |
| BSM | كل | Tivoli Business Service Manager |
| CDB | كل | 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 bit} |
| 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 Version 7.2 for Red Hat Enterprise Linux 5 x86 64 bit |
| COH | Linux | Tivoli Provisioning Manager for Red Hat Enterprise Linux 5 System z® 64 bit |
| COI | Linux | Tivoli Provisioning Manager for SUSE 10 |
| COJ | Solaris | Tivoli Provisioning Manager Version 7.2 for Solaris |
| COK | HP-UX | Tivoli Provisioning Manager Version 7.2 for HP-UX |
| COL | Linux | Tivoli Provisioning Manager Version 7.2 for SUSE zSeries® 10 |
| COM | Linux | Tivoli Provisioning Manager Version 7.2 for SUSE 11 |
| CON | Linux | Tivoli Provisioning Manager Version 7.2 for SUSE zSeries 11 |
| COX | Windows | Tivoli Provisioning Manager Version 7.2 for Windows 2008 |

| | | |
|---|---|---|
| Tivoli Provisioning Manager Version 7.2 for Windows 2003 | Windows | COY |
| Tivoli Provisioning Manager Version 7.2 for Windows | Windows | COZ |
| Prerequisite Scanner demo | كل | DMO |
| IBM Tivoli Netcool® Performance Manager | UNIX | GYM |
| Tivoli Enterprise Portal Client | Windows | KCJ |
| Tivoli Enterprise Portal Client for UNIX | UNIX | |
| Tivoli Enterprise Portal Server | Windows | KCQ |
| Tivoli Enterprise Portal Server for UNIX | UNIX | |
| Warehouse Proxy Agent | كل | KHD |
| Warehouse Proxy Agent for UNIX | UNIX | KHE |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Internet Service Monitoring | UNIX | KIS |
| Tivoli Monitoring Operating System Agent for Linux | UNIX | KLZ |
| IBM Tivoli Composite Application Manager Agent for WebSphere MQ File Transfer Edition | Windows | KM6 |
| Tivoli Composite Application Manager Agent for WebSphere MQ | كل | KMQ |
| Tivoli Enterprise Monitoring Server | Windows | KMS |
| Tivoli Enterprise Monitoring Server for UNIX | UNIX | |
| Tivoli Monitoring Operating System Agent for Windows | Windows | KNT |
| Windows OS monitoring Agent for UNIX | UNIX | |
| Tivoli Monitoring Agent for Oracle | Windows | KOR |
| Tivoli Composite Application Manager Agent for WebSphere Message Broker | كل | KQI |
| Summarization and Pruning Agent | Windows | KSY |
| Summarization and Pruning Agent for UNIX | UNIX | |
| Tivoli Monitoring Agent for DB2 | Windows | KUD |
| Tivoli Monitoring Agent for DB2 | UNIX | |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Reporter | كل | KTO |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Collector | كل | KTU |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Application Management Console | كل | KT3 |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Client Response Time | كل | KT4 |

| | | |
|---|---|---|
| KT5 | كل | Tivoli Composite Application Manager (ITCAM) for Transactions: Web Response Time |
| KT6 | كل | Tivoli Composite Application Manager (ITCAM) for Transactions: Robotic Response Time |
| KZE | كل | Tivoli zEnterprise Monitoring Agent |
| LCM | Windows | Tivoli License Compliance Manager |
| | UNIX | Tivoli License Compliance Manager for UNIX |
| NCI | كل | Tivoli Netcool/Impact |
| NOC | كل | مكونات وحدة الخدمة ومكونات جهاز سطح المكتب الى Tivoli Netcool/OMNIbus |
| NOD | كل | مكون سطح المكتب الى Tivoli Netcool/OMNIbus |
| NOS | كل | مكونات وحدة الخدمة الى Tivoli Netcool/OMNIbus |
| PAE | كل | Tivoli Process Automation Engine |
| TAD | Windows | Tivoli Asset Discovery for Distributed |
| | UNIX | Tivoli Asset Discovery for Distributed for UNIX |
| TCR | كل | Tivoli Common Reporting |
| TPM | كل | Tivoli Provisioning Manager |

# الملحق ب. مرجع ملفات التوصيف

يتيح IBM Prerequisite Scanner مجموعة سابقة التعريف من ملفات التوصيف يمكن تحريرها. هذه الملفات توجد في *ips_root*/UNIX_Linux أو *ips_root*/Windows. الملفات يكون لها اللاحقة .cfg.

يعرض الجدول ١٥ ملفات التوصيف سابقة التعريف المدعمة حاليا.

*الجدول ١٥. ملفات التوصيف سابقة التعريف*

| ملف التوصيف | بيئة التشغيل | نسخة البرنامج، بيئة التشغيل، نظام التشغيل |
|---|---|---|
| ADE_01040000.cfg | كل | Autonomic Deployment Engine Version 1.4 |
| BSM_04210000.cfg | كل | Tivoli Business Service Manager Version 4.2.1 |
| BSM_06100000.cfg | كل | Tivoli Business Service Manager Version 6.1 |
| CDB_06220000.cfg | كل | Tivoli Composite Application Manager (ITCAM) for Applications: DB2 Version 6.2.2 |
| COA_07200000.cfg | UNIX | Tivoli Provisioning Manager Version 7.2 for UNIX |
| COB_07200000.cfg | AIX | Tivoli Provisioning Manager Version 7.2 for AIX |
| COC_07200000.cfg | AIX | Tivoli Provisioning Manager Version 7.2 for AIX V5.3.0.0 {64 bit} |
| COD_07200000.cfg | AIX | Tivoli Provisioning Manager Version 7.2 for AIX 6.1 |
| COE_07200000.cfg | Linux | Tivoli Provisioning Manager Version 7.2 for Linux |
| COF_07200000.cfg | Linux | Tivoli Provisioning Manager Version 7.2 for Red Hat Linux |
| COG_07200000.cfg | Linux | Tivoli Provisioning Manager Version 7.2 for Red Hat Enterprise Linux 5 x86 64 bit |
| COH_07200000.cfg | Linux | Tivoli Provisioning Manager Version 7.2 for Red Hat Enterprise Linux 5 System z 64 bit |
| COI_07200000.cfg | Linux | Tivoli Provisioning Manager Version 7.2 for SUSE 10 |
| COJ_07200000.cfg | Solaris | Tivoli Provisioning Manager Version 7.2 for Solaris |
| COK_07200000.cfg | HP-UX | Tivoli Provisioning Manager Version 7.2 for HP-UX |
| COL_07200000.cfg | Linux | Tivoli Provisioning Manager Version 7.2 for SUSE zSeries 10 |
| COM_07200000.cfg | Linux | Tivoli Provisioning Manager Version 7.2 for SUSE 11 |
| CON_07200000.cfg | Linux | Tivoli Provisioning Manager Version 7.2 for SUSE zSeries 11 |
| COX_07200000.cfg | Windows | Tivoli Provisioning Manager Version 7.2 for Windows 2008 |
| COY_07200000.cfg | Windows | Tivoli Provisioning Manager Version 7.2 for Windows 2003 |
| COZ_07200000.cfg | Windows | Tivoli Provisioning Manager Version 7.2 for Windows |
| DMO_00000000.cfg | كل | Prerequisite Scanner demo |
| DMO_01000000.cfg | كل | Prerequisite Scanner Version 1.0 demo |
| GYM_01030200.cfg | UNIX | IBM Tivoli Netcool Performance Manager Version 1.3.2 |
| KCJ_06200000.cfg | Windows | Tivoli Enterprise Portal Client Version 6.2 |

| | | |
|---|---|---|
| Tivoli Enterprise Portal Client Version 6.2.1 | UNIX | KCJ_06210000.cfg |
| Tivoli Enterprise Portal Client Version 6.2.2 | كل | KCJ_06220000.cfg |
| Tivoli Enterprise Portal Server Version 6.2 | Windows | KCQ_06200000.cfg |
| Tivoli Enterprise Portal Server Version 6.2.2 | UNIX | KCQ_06210000.cfg |
| Tivoli Enterprise Portal Server Version 6.2.2 | كل | KCQ_06220000.cfg |
| Warehouse Proxy Agent Version 6.2 | Windows | KHD_06200000.cfg |
| Warehouse Proxy Agent Version 6.2.1 | كل | KHD_06210000.cfg |
| Warehouse Proxy Agent Version 6.2.2 | كل | KHD_06220000.cfg |
| Warehouse Proxy Agent Version 6.2.2 | UNIX | KHE_06220000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Internet Service Monitoring Version 7.2 | كل | KIS_07200000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Internet Service Monitoring Version 7.3 | كل | KIS_07300000.cfg |
| Tivoli Monitoring Operating System Agent for Linux Version 6.2.1 | UNIX | KLZ_06210000.cfg |
| Tivoli Monitoring Operating System Agent for Linux Version 6.2.2 | UNIX | KLZ_06220000.cfg |
| Tivoli Composite Application Manager Agent for WebSphere MQ File Transfer Edition Version 7.1 | Windows | KM6_070100000.cfg |
| Tivoli Composite Application Manager Agent for WebSphere MQ Version 7.1 | كل | KMQ_070100000.cfg |
| Tivoli Enterprise Monitoring Server Version 6.2 | Windows | KMS_06200000.cfg |
| Tivoli Enterprise Monitoring Server Version 6.2.1 | كل | KMS_06210000.cfg |
| Tivoli Enterprise Monitoring Server Version 6.2.2 | كل | KMS_06220000.cfg |
| Tivoli Monitoring Operating System Agent for Windows Version 6.2 | Windows | KNT_06200000.cfg |
| Tivoli Monitoring Operating System Agent for Windows Version 6.2.1 | Windows | KNT_06210000.cfg |
| Tivoli Monitoring Operating System Agent for Windows Version 6.2.2 | Windows | KNT_06220000.cfg |
| Tivoli Monitoring Agent for Oracle Version 6.2.2 | Windows | KOR_06220000.cfg |
| Tivoli Composite Application Manager Agent for WebSphere Message Broker Version 7.1 | كل | KQI_07010000.cfg |
| Summarization and Pruning Agent Version 6.2 | Windows | KSY_06200000.cfg |
| Summarization and Pruning Agent Version 6.2.1 | كل | KSY_06210000.cfg |
| Summarization and Pruning Agent Version 6.2.2 | كل | KSY_06220000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Reporter Version 7.2 | UNIX | KTO_07200000.cfg |

| | | |
|---|---|---|
| Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Reporter Version 7.2.2 | Windows | KTO_07200200.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Reporter Version 7.3 | UNIX | KTO_07300000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Collector Version 7.2 | UNIX | KTU_07200000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Collector Version 7.2.2 | Windows | KTU_07200200.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Transaction Collector Version 7.3 | UNIX | KTU_07300000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Application Management Console Version 7.3 | كل | KT3_07300000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Client Response Time Version 7.3 | كل | KT4_07300000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Web Response Time Version 7.3 | كل | KT5_07300000.cfg |
| Tivoli Composite Application Manager (ITCAM) for Transactions: Robotic Response Time Version 7.3 | كل | KT6_07300000.cfg |
| Tivoli Monitoring Agent for DB2 Version 6.1 | Windows | KUD_06100000.cfg |
| Tivoli Monitoring Agent for DB2 Version 6.2 | كل | KUD_06200000.cfg |
| Tivoli Monitoring Agent for DB2 Version 6.2.1 | كل | KUD_06210000.cfg |
| Tivoli Monitoring Agent for DB2 Version 6.2.2 | كل | KUD_06220000.cfg |
| Tivoli zEnterprise Monitoring Agent Version 6.2.3 | كل | KZE_06020300.cfg |
| Tivoli License Compliance Manager Version 1.0 | كل | LCM_01000000.cfg |
| Tivoli License Compliance Manager Version 2.3 | كل | LCM_02300000.cfg |
| Tivoli Netcool/Impact Version 6.1 | كل | NCI_06100000.cfg |
| Tivoli Netcool/OMNIbus server components and desktop component Version 7.3.1 | كل | NOC_07310000.cfg |
| Tivoli Netcool/OMNIbus desktop component Version 7.3.1 | كل | NOD_07310000.cfg |
| Tivoli Netcool/OMNIbus server components Version 7.3.1 | كل | NOS_07310000.cfg |
| Tivoli Process Automation Engine | كل | PAE_07500000.cfg |
| Tivoli Asset Discovery for Distributed Version 7.2 | كل | TAD_07200000.cfg |
| Tivoli Asset Discovery for Distributed Version 7.2.2 | كل | TAD_07220000.cfg |
| Tivoli Common Reporting | كل | TCR_02010100.cfg |
| Tivoli Provisioning Manager Version 7.2.1 | كل | TPM_07210000.cfg |

# الملحق ج. مرجع خصائص المتطلبات الرئيسية

يوضح هذا المرجع خصائص المتطلبات الرئيسية الأساسية لكل تصنيف معرف مسبقا للمتطلبات الرئيسية للأجهزة والبرامج.

يوضح الجدول ١٦ التصنيفات المعرفة مسبقا للمتطلبات الرئيسية للأجهزة والبرامج.

*الجدول ١٦. التصنيفات المعرفة مسبقا لخصائص المتطلبات الرئيسية*

| تصنيف البيانات | الوصف | بادئة كود التعريف المطلوبة | المرجع |
|---|---|---|---|
| عام | تقوم خصائص البيانات العامة بالتحقق من المتطلبات الرئيسية العامة مثل سرعة المشغل و RAM والقرص والمساحة المؤقتة. | لا شيء | "خصائص البيانات العامة" |
| Autonomic Deployment Engine | تقوم خصائص بيانات Autonomic Deployment Engine بالتحقق من متطلبات Autonomic Deployment Engine الرئيسية مثل وحدة التركيب. | de | "خصائص بيانات Autonomic Deployment Engine" |
| البرنامج الذي تم تركيبه | تقوم خصائص بيانات البرامج التي تم تركيبها بالتحقق من المتطلبات الرئيسية للبرامج التي تم تركيبها مثل البرامج التي تم تسجيلها في سجل Windows وما اذا كان قد تم تركيب cygwin و gskit. | لا شيء | "خصائص بيانات البرامج التي تم تركيبها" |
| المستخدم | تقوم خصائص بيانات المستخدم بالتحقق من المتطلبات الرئيسية للمستخدم مثل ما اذا كان يتوافر لدى المستخدم الذي قام ببدء الاتصال صلاحية ادارية أو يعد مستخدم رئيسي. | user | "خصائص البيانات للمستخدم" |
| نظام التشغيل | تقوم خصائص بيانات نظام التشغيل بالتحقق من المتطلبات الرئيسية لنظام التشغيل مثل النسخة وهيكل البناء واجمالي الذاكرة والذاكرة المتاحة والذاكرة الفعلية الاجمالية. | os | "خصائص بيانات نظام التشغيل" |
| امكانية الاتصال | تقوم خصائص بيانات امكانية الاتصال بالتحقق من المتطلبات الرئيسية لامكانية الاتصال مثل ما اذا كان يتم تشغيل Telnet وعناوين IP والمنافذ التي يمكن توصيل الماسح الضوئي بها. | لا شيء | "خصائص البيانات لامكانية الاتصال" |
| شبكة الاتصال | خصائص بيانات شبكة الاتصال للتحقق من المتطلبات الرئيسية لشبكة الاتصال التي يمكن أن تكون عامة لكل بيئات التشغيل وما اذا كانت منافذ الاتصال متاحة أم لا. | network | "خصائص بيانات شبكة الاتصال" |
| شبكة اتصال Windows | تقوم خصائص بيانات شبكة اتصال Windows بالتحقق من المتطلبات الرئيسية لشبكة الاتصال مثل اتاحة NetBIOS و DHCP على الجهاز وخصائص اختبار الاتصال. | network | "خصائص بيانات شبكة الاتصال لنظام Windows" |

| | | | |
|---|---|---|---|
| شبكة اتصال UNIX | تقوم خصائص بيانات شبكة اتصال UNIX بالتحقق من المتطلبات الرئيسية لشبكة الاتصال مثل اتاحة NetBIOS و DHCP على الجهاز وخصائص اختبار الاتصال. | network | "خصائص بيانات شبكة الاتصال لنظام UNIX" |
| Internet Explorer | تقوم خصائص بيانات Microsoft Internet Explorer بالتحقق من متطلبات Internet Explorer الرئيسية مثل النسخة. | internetExplorer | "خصائص بيانات Internet Explorer" |
| وحدة خدمة قاعدة البيانات، DB2 | تقوم خصائص بيانات DB2 بالتحقق من متطلبات DB2 الرئيسية مثل النسخة. | DB2 | "خصائص بيانات DB2" |
| وحدة خدمة قاعدة البيانات، MS SQL | تقوم خصائص بيانات MS SQL Server بالتحقق من المتطلبات الرئيسية لوحدة خدمة MS SQL مثل النسخة. | mssql | "خصائص بيانات MS SQL Server" |
| وحدة خدمة قاعدة البيانات، Oracle | تقوم خصائص بيانات Oracle الرئيسية مثل النسخة. | Oracle | "خصائص بيانات Oracle" |
| متغير بيئة التشغيل | تقوم متغيرات بيئة التشغيل بالتحقق من المتطلبات الرئيسية لمتغير بيئة التشغيل مثل ما اذا كان قد تم تحديد متغير بيئة التشغيل. | env | "خصائص بيانات متغير بيئة التشغيل" |

## خصائص البيانات العامة

تقوم خصائص البيانات العامة بالتحقق من المتطلبات الرئيسية العامة مثل سرعة وحدة CPU و RAM والقرص والمساحة المؤقتة. بالنسبة لأنظمة Windows، يتم استخدام البرنامج النصي الأولي IBM Prerequisite Scanner. وبالنسبة لأنظمة UNIX، يتم استخدام البرنامج النصي الأولي Prerequisite Scanner وأداة التجميع العامة، *ips_root*/Unix_Linux/common.sh.

يوضح الجدول ١٧ خصائص المتطلبات الرئيسية للبيانات العامة. هذه الفئة من خصائص المتطلبات الرئيسية لا تتطلب بادئة كود التعريف.

*الجدول ١٧. خصائص المتطلبات الرئيسية للبيانات العامة*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| CPU Name | كل | اسم وحدة CPU والمستخدمة لأغراض العرض فقط في النتائج | غير منطبق |
| CpuArchitecture | نظام UNIX | هيكل بناء نظام التشغيل | مجموعة حروف، مع عدة قيم مدعمة مفصولة بفاصلة، على سبيل المثال:<br>x86_64,s390x,ppc64, AMD64 |

| | | | |
|---|---|---|---|
| DBType | كل | يتم التحقق من نوع وحدة خدمة قاعدة البيانات التي تم تركيبها على الجهاز.<br><br>بالنسبة الى Oracle على أنظمة UNIX فقط: تتوقع أداة التجميع تحديد متغيرات بيئة التشغيل ORACLE_BASE و ORACLE_HOME في الملف $HOME/.profile، على سبيل المثال:<br>`export ORACLE_BASE=/home/oracle/app/oracle/product/11.2.0/`<br>`export ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/dbhome_1`<br><br>حيث $HOME يجب أن تكون /home/oracle، والدليل الرئيسي لمستخدم Oracle. | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• مجموعة حروف تمثل أي نوع من وحدة خدمة قاعدة البيانات، على سبيل المثال:<br>`any`<br>• مجموعة حروف تمثل نوع وحدة خدمة قاعدة البيانات، على سبيل المثال:<br>`Oracle`<br>• regex{*str*}، تعبير معتاد مع معامل المدخلات، *str*، يمثل نموذج البحث لنوع وحدة خدمة قاعدة البيانات، على سبيل المثال:<br>`regex{.*MSSQL.*\|DB2.*}`<br>يتم التحقق مما اذا كان نوع وحدة خدمة قاعدة البيانات هو MS SQL أو DB2 على أنظمة Windows.<br>• مجموعة حروف لا تمثل أي نوع لوحدة خدمة قاعدة البيانات، على سبيل المثال:<br>`unknown` |

| | | | |
|---|---|---|---|
| DBTypeDetails | كل | أنوع وحدة خدمة قاعدة البيانات التي تم تركيبها على الجهاز.<br><br>بالنسبة الى Oracle على أنظمة UNIX فقط: تتوقع أداة التجميع تحديد متغيرات بيئة التشغيل ORACLE_BASE و ORACLE_HOME في الملف $HOME/.profile، على سبيل المثال:<br>`export ORACLE_BASE=/home/oracle/app/oracle/product/11.2.0/`<br>`export ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/dbhome_1`<br><br>حيث $HOME يجب أن تكون /home/oracle، والدليل الرئيسي لمستخدم Oracle.<br><br>تقوم خاصية المتطلبات الرئيسية بكتابة تفاصيل نوع وحدة خدمة قاعدة البيانات، وهي نوع وحدة خدمة قاعدة البيانات ومكان التركيب والنسخة لملف result.txt. يتم فصل تفاصيل أنواع وحدات خدمة قاعدة البيانات المتعددة بفواصل منقوطة. | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• مجموعة حروف تمثل أي نوع من وحدة خدمة قاعدة البيانات، على سبيل المثال:<br>`any`<br>• مجموعة حروف تمثل نوع واحد لوحدة خدمة قاعدة البيانات، على سبيل المثال:<br>`DB2`<br>• regex{*str*}، تعبير معتاد مع معامل المدخلات، *str*، يمثل نموذج البحث لنوع وحدة خدمة قاعدة البيانات، على سبيل المثال:<br>`regex{.*MSSQL.*\|DB2.*}`<br>يتم التحقق مما اذا كان نوع وحدة خدمة قاعدة البيانات هو MS SQL أو DB2 على أنظمة Windows. |
| Disk | نظام Windows | حجم مساحة القرص الخالية، مع الخواص المميزة المؤهلة الاختيارية:<br>• الخاصية المميزة dir، لتحديد المسار للدليل الذي سيتم فحصه<br>• الخاصية المميزة unit، لتحديد وحدات مساحة القرص التي سيتم استخدامها | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• مجموعة حروف بالنسق المؤهل التالي:<br>`[dir:`*dir_path*`, unit:`*unit_name*`]` *disk_space*<br>على سبيل المثال:<br>`Disk=[dir:C:\Program Files\IBM\SQLLIB, unit:MB]1431`<br>• نسق رقمي بالجيجابايت أو الميجابايت:<br>*disk_space*`MB\|GB`<br>على سبيل المثال:<br>`Disk=250MB` |
| Disk | نظام UNIX | حجم مساحة القرص الخالية | نسق رقمي بالجيجابايت أو الميجابايت، على سبيل المثال:<br>`2GB` |

| | | | |
|---|---|---|---|
| intel.cpu | كل | سرعة CPU لمشغل Intel | نسق رقمي بالجيجاهرتز، وبنظام Windows فقط يكون بالميجاهرتز أيضا، على سبيل المثال:<br>2GHz |
| Memory | كل | اجمالي حجم الذاكرة الفعلية المتاحة حاليا على الجهاز.<br><br>**تلميح:** بشكل منفصل، يتم التحقق من حجم الذاكرة الفعلية والظاهرية المتاحة باستخدام خصائص المتطلبات الرئيسية المعرفة مسبقا في تصنيف نظام التشغيل. | نسق رقمي بالجيجابايت أو الميجابايت، على سبيل المثال:<br>300MB |
| OS Version | كل | الاسم الكامل ونسخة نظام التشغيل التي يتم تشغيلها على الجهاز؛ تبادليا، يمكنك استخدام تعبير معتاد لامرار مجموعة حروف تمثل عدة اختلافات لنظام التشغيل.<br><br>**تلميح:** استخدم خاصية المتطلبات الرئيسية هذه مع os.servicePack و os.architecture للتحقق من مجموعة الخدمات الحالية وهيكل بناء النظام . | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• مجموعة حروف يمكن أن تمثل عدة نسخ، مع فصل كل نسخة بفاصلة، على سبيل المثال:<br>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<br>**قيد:** على أنظمة Windows، يكون حرف الابدال * مدعم فقط في التعبير المعتاد.<br>• regex{*str*}، تعبير معتاد مع معامل المدخلات، *str*، يمثل نموذج البحث للنسخة، على سبيل المثال:<br>regex{Windows 200[3-8]}<br>يتم التحقق مما اذا كان نظام التشغيل الفعلي يطابق أي نسخة من Windows 2003 الى Windows 2008.<br>regex{Red Hat*.*}<br>يتم التحقق مما اذا كان نظام التشغيل الفعلي يطابق اختلاف نظام التشغيل Red Hat Linux.<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |

| | | | |
|---|---|---|---|
| numCPU | كل | عدد الأجزاء المركزية أو المشغلات المنفصلة على الحاسب. اذا قامت الأداة باجراء عملية مسح للحاسب ولم تقم بايجاد أية أجزاء رئيسية أو قامت بايجاد مشغلات لا يعد أي منها جزء مركزي، ستقوم بارجاع النتيجة لم يتم ايجاد شيء. | رقم، على سبيل المثال، 4 |
| risc.cpu | نظام UNIX | سرعة CPU لمشغل RISC | نسق رقمي بالميجاهرتز، على سبيل المثال:<br>1.4GHz |
| Temp | نظام UNIX | مساحة القرص المتاحة لنظام الملفات *Temp* المحدد | نسق رقمي بالجيجابايت أو الميجابايت، على سبيل المثال:<br>300MB |

**المفاهيم المتعلقة**:

تقوم أداة Prerequisite Scanner بالتحقق من من خاصية المتطلبات الرئيسية الى Memory بشكل مختلف بناءا على ما اذا كان يتم تشغيل Tivoli Monitoring أو الأداة البرمجية Tivoli Composite Application Manager على الحاسب بالفعل.

**المرجع المتعلق:**

تقوم خصائص بيانات نظام التشغيل بالتحقق من المتطلبات الرئيسية لنظام التشغيل مثل النسخة وهيكل البناء واجمالي الذاكرة والذاكرة المتاحة والذاكرة الفعلية الاجمالية. بالنسبة لأنظمة Windows فقط، يتم استخدام أدوات تجميع VBScript لنظام التشغيل في الدليل *ips_root*/lib مع بادئة كود التعريف os في أسماء الملفات الخاصة بها. بالنسبة لأنظمة UNIX فقط، يتم استخدام أدوات تجميع نظام التشغيل UNIX في الدليل *ips_root*/UNIX_Linux مع بادئة كود التعريف os في أسماء الملفات الخاصة بها.

## تصرف النظام لخاصية المتطلبات الرئيسية للذاكرة والأدوات البرمجية الى Tivoli Monitoring

تقوم أداة Prerequisite Scanner بالتحقق من من خاصية المتطلبات الرئيسية الى Memory بشكل مختلف بناءا على ما اذا كان يتم تشغيل Tivoli Monitoring أو الأداة البرمجية Tivoli Composite Application Manager على الحاسب بالفعل.

اذا كان قد تم تركيب الأداة البرمجية بالفعل، ستقوم Prerequisite Scanner باستخدام قيمة متوقعة لخاصية المتطلبات الرئيسية الى Memory بناءا على قيمة ملفات التوصيف الجديدة والموجودة اذا كان ملف التوصيف الموجود مازال على الحاسب؛ خلاف ذلك، ستقوم بالتعامل مع القيمة المتوقعة بواسطة التصرف المفترض.

عند تشغيل Prerequisite Scanner للتحقق من المتطلبات الرئيسية للأداة البرمجية الى Tivoli التي يتم تطوير نسختها أو اعادة تركيبها، سيتم التحقق أولا مما اذا كان يتم تشغيل الأداة البرمجية على الحاسب بالفعل. اذا كان يتم تشغيل الأداة البرمجية، ستقوم أداة Prerequisite Scanner بالبحث عن ملف التوصيف المرفق مع نسخة الأداة البرمجية الحالية التي يتم تشغيلها. تحدث التصرفات التالية بناءا على نتائج عملية البحث هذه:

- اذا لم يمكن ايجاد ملف التوصيف، ستقوم Prerequisite Scanner بافتراض أنه لم يتم اجراء عملية مسح مسبقا لبيئة التشغيل المستهدفة؛ لذلك، تقوم Prerequisite Scanner باستخدام القيمة المتوقعة لخاصية المتطلبات الرئيسية Memory المحددة في ملف التوصيف الجديد، والتي تتبع التصرف المفترض. ستقوم Prerequisite Scanner بكتابة هذه القيمة المتوقعة في المخرجات الناتجة.
- اذا تم ايجاد ملف التوصيف، ستقوم Prerequisite Scanner بمقارنة القيمة المتوقعة لخاصية المتطلبات الرئيسية الى Memory للنسخة الحالية بالقيمة المتوقعة في ملف التوصيف للنسخة الجديدة. اذا كان هناك اختلاف بين القيم، وكانت القيمة الجديدة أكبر من القيمة المتوقعة الحالية، ستقوم Prerequisite Scanner بتحديد هذا الاختلاف على أنه القيمة المتوقعة. ستقوم Prerequisite Scanner بكتابة الاختلاف في القيمة المتوقعة في المخرجات الناتجة. على سبيل المثال، يقوم ملف التوصيف للأداة البرمجية بالنسخة 1 بتحديد ١ جيجابايت على أنها القيمة المتوقعة. ويقوم ملف التوصيف الجديد للأداة البرمجية بالنسخة 2، بتحديد القيمة ١,٥ جيجابايت على أنه القيمة المتوقعة؛ لذلك، ستقوم Prerequisite Scanner باستخدام وكتابة ٠,٥ جيجابايت على أنها الاختلاف في القيمة المتوقعة.

## خصائص بيانات Autonomic Deployment Engine

تقوم خصائص بيانات Autonomic Deployment Engine بالتحقق من متطلبات Autonomic Deployment Engine الرئيسية مثل وحدة التركيب. بالنسبة لأنظمة Windows فقط، يتم استخدام أدوات تجميع Autonomic Deployment Engine في الدليل *ips_root*/lib/، مع البادئة de في أسماء الملفات الخاصة بها. بالنسبة لأنظمة UNIX فقط، يتم استخدام أدوات تجميع UNIX Autonomic Deployment Engine في الدليل *ips_root*/UNIX_Linux، مع البادئة de في أسماء الملفات الخاصة بها.

يوضح الجدول ١٨ خصائص المتطلبات الرئيسية. يتطلب تصنيف خصائص المتطلبات الرئيسية هذا بادئة كود التعريف de.

*الجدول ١٨. خصائص بيانات Autonomic Deployment Engine*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| de.installed | كل | يتم التحقق مما اذا كان قد تركيب ما هو محدد | بوولي، على سبيل المثال:<br>true\|false |
| de.installationUnit | كل | يتم التحقق مما اذا كان قد تم تركيب وحدة التركيب المحددة باستخدام الأمر **listIU -v** | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• مجموعة حروف لتمثيل وحدة تركيب واحدة، على سبيل المثال، وحدة تركيب Tivoli Integrated Portal:<br>C37109911C8A11D98E17000 61BDE7AEA, B24209911C8A11D98E17000 61BDE7AEA<br>• مجموعة حروف لتمثيل عدة وحدات تركيب، على سبيل المثال:<br>5FFE79F918DF3BA0D67511 FD3F7C358E<br>• {str} regex، تعبير معتاد مع معامل المدخلات، str، لتمثيل نموذج البحث لوحدة التركيب والنسخة ومسار التركيب؛ على سبيل المثال، للتحقق من وحدة التركيب، تكون نسخة WebSphere Application Server ومسار تركيب Tivoli Integrated Portal ونموذج التركيب كما يلي:<br>regex{.*C00DA95AFD9B7E 0397153CD944B5A255. *6.1.0.2100.*SIU eWAS. *C:\\IBM\\tivoli\\tip.*}<br>**ملاحظة:** يمكنك أيضا استخدام متغير بيئة التشغيل لمسار التركيب؛ على سبيل المثال، باستبدال المسار بمتغير بيئة التشغيل TIPHOME، يكون نموذج البحث هو:<br>regex{.*C00DA95AFD9B7E 0397153CD944B5A255. *6.1.0.2100.*SIU eWAS. *%TIPHOME%.*} |

| | | | |
|---|---|---|---|
| | | | • عدة متغيرات مستقلة الى regex {str}لتمثيل عدة عمليات تحقق؛ على سبيل المثال:<br>regex{.*C37109911C8A11 D98E1700061BDE7AEA.*}, regex{.*B24209911C8A11 D98E1700061BDE7AEA.*} |

## خصائص البيانات لامكانية الاتصال

تقوم خصائص بيانات امكانية الاتصال بالتحقق من المتطلبات الرئيسية لامكانية الاتصال مثل ما اذا كان يتم تشغيل Telnet وعناوين IP والمنافذ التي يمكن توصيل الماسح الضوئي بها. بالنسبة لأنظمة Windows، يتم استخدام أداة التجميع لامكانية التوصل، *ips_root*/lib/connectivity_plug.vbs. وبالنسبة لأنظمة UNIX، يتم استخدام البرنامج النصي IBM Prerequisite Scanner الأساسي وأداة التجميع لامكانية الاتصال، *prs_root*/Unix_Linux/connectivity_plug.sh. يتم امرار المخرجات الى ملف سجل تصحيح الأخطاء فقط.

## خصائص بيانات DB2

تقوم خصائص بيانات DB2 بالتحقق من متطلبات DB2 الرئيسية مثل النسخة. بالنسبة لأنظمة Windows فقط، يتم استخدام أداة تجميع DB2، *ips_root*/lib/db2_version_plug.bat. وبالنسبة لأنظمة UNIX فقط، يتم استخدام أدوات تجميع DB2 UNIX في الدليل *ips_root*/UNIX_Linux، مع البادئة db2 في أسماء الملفات الخاصة بها.

يوضح الجدول ١٩ خصائص المتطلبات الرئيسية الى DB2. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف DB2.

*الجدول ١٩. خصائص بيانات DB2*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| DB2 Version | كل | نسخة DB2 التي تم تركيبها على الجهاز | مجموعة حروف، على سبيل المثال:<br>v9.5.100.179FP4 |
| db2.home.space | UNIX | مساحة القرص المتاحة لدليل DB2 الرئيسي | النسق الرقمي بالجيجابايت، على سبيل المثال:<br>8GB |

## خصائص بيانات MS SQL Server

تقوم خصائص بيانات MS SQL Server بالتحقق من المتطلبات الرئيسية لوحدة خدمة MS SQL Server مثل النسخة والمكان. بالنسبة لأنظمة Windows فقط، يتم استخدام أدوات تجميع MS SQL Server في الدليل *ips_root*/Windows، مع البادئة mssql في أسماء الملفات الخاصة بها.

يوضح الجدول ٢٠ خصائص المتطلبات الرئيسية الى MS SQL Server. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف `mssql`.

*الجدول ٢٠. خصائص بيانات MS SQL Server*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| `mssql.Client` | نظام Windows | يتم التحقق من نسخة MS SQL Client التي يتم تركيبها حاليا على الجهاز | القيمة الحرفية المتوقعة يمكن أن تكون نسخ متعددة، مفصولة بفاصلات، على سبيل المثال:<br>`10.50.1600.1`<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| `mssql.Server` | نظام Windows | يتم التحقق من نسخة MS SQL Server التي يتم تركيبها حاليا على الجهاز | القيمة الحرفية المتوقعة يمكن أن تكون نسخ متعددة، مفصولة بفاصلات، على سبيل المثال:<br>`10.50.1600.1`<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| `mssql.Server.Location` | نظام Windows | يتم التحقق من الدليل الرئيسي لوحدة خدمة قاعدة بيانات MS SQL | مجموعة حروف، على سبيل المثال:<br>`any` |

## خصائص بيانات Internet Explorer

تقوم خصائص بيانات Microsoft Internet Explorer بالتحقق من متطلبات Internet Explorer الرئيسية مثل النسخة. يتم استخدام أداة تجميع Internet Explorer، `/lib/internetExplorer_plug.vbs` *`ips_root`*.

يوضح الجدول ٢١ خصائص المتطلبات الرئيسية لبرنامج Internet Explorer. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف `internetExplorer`.

*الجدول ٢١. خصائص بيانات Internet Explorer*

| خاصية المتطلبات الرئيسية | الوصف | القيم الصالحة |
|---|---|---|
| `.internetExplorer version` | نسخة Internet Explorer التي تم تركيبها على الجهاز | النسق الرقمي، على سبيل المثال، `7.0+`<br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |

## خصائص بيانات شبكة الاتصال

خصائص بيانات شبكة الاتصال للتحقق من المتطلبات الرئيسية لشبكة الاتصال التي يمكن أن تكون عامة لكل بيئات التشغيل وما اذا كانت منافذ الاتصال متاحة أم لا. يتم استخدام أدوات تجميع شبكة الاتصال في الدليل *ips_root*/lib، مع بادئة كود التعريف network في أسماء الملفات الخاصة بها.

يوضح الجدول ٢٢ خصائص المتطلبات الرئيسية لشبكة الاتصال العامة عبر كل بيئات التشغيل. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف network.

*الجدول ٢٢. خصائص بيانات شبكة الاتصال*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| network.available Ports. *app_type* | كل | استخدم مصطلح التسمية هذا للتحقق مما اذا كان المنفذ أو مدى المنافذ متاحا لنوع التطبيق *app_type*. تحقق من المنافذ التي لا يتم الانصات اليها، على سبيل المثال:<br>• network.available Ports. تقوم DB2 بالتحقق من المنافذ لوحدة خدمة قاعدة بيانات DB2، حيث *app_type* هو DB2<br>• network.available Ports. تقوم WAS بالتحقق من المنافذ لوحدة خدمة WebSphere Application Server، حيث *app_type* هو WAS | الأرقام الصحيحة الموجبة، على سبيل المثال:<br>• network.available Ports. DB2 = 50000-50005<br>• network.available Ports. WAS = 8080<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| network.portsInUse. *app_type* | كل | استخدم مصطلح التسمية هذا للتحقق مما اذا كان المنفذ أو مدى المنافذ يتم استخدامه لنوع التطبيق *app_type*. تحقق من المنافذ التي يتم الانصات اليها، على سبيل المثال:<br>• network.available Ports. تقوم DB2 بالتحقق من المنافذ لوحدة خدمة قاعدة بيانات DB2، حيث *app_type* هو DB2<br>• network.available Ports. تقوم WAS بالتحقق من المنافذ لوحدة خدمة WebSphere Application Server، حيث *app_type* هو WAS | الأرقام الصحيحة الموجبة، على سبيل المثال:<br>• network.portsInUse. DB2 = 50900-50905<br>• network.portsInUse. WAS = 8080<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |

| | | | |
|---|---|---|---|
| network.validate HostsFile | نظام Windows | يتم التحقق مما اذا كانت كل أجهزة النظام الرئيسي معروضة في ملف الأنظمة الرئيسية بالنسق:<br>*IP_Address Host_Name Short_Name*<br><br>حيث:<br>• *IP_Address* هو عنوان IP للحاسب، على سبيل المثال، 127.0.0.1<br>• *Host_Name* هو اسم النظام الرئيسي المصنف بالكامل للحاسب، على سبيل المثال، localhost.local domain<br>• *Short_Name* هو الاسم المختصر للحاسب، على سبيل المثال، localhost | قيمة بوولية، على سبيل المثال True |

## خصائص بيانات Oracle

تقوم خصائص بيانات Oracle الرئيسية مثل النسخة. بالنسبة لأنظمة Windows فقط، يتم استخدام أداة تجميع Oracle. وبالنسبة لأنظمة UNIX فقط، يتم استخدام أدوات تجميع UNIX Oracle في الدليل *ips_root*/UNIX_Linux مع البادئة oracle في أسماء الملفات الخاصة بها. بالنسبة لأنظمة Windows فقط، يتم استخدام أدوات تجميع Windows Oracle في الدليل *ips_root*/lib مع البادئة oracle في أسماء الملفات الخاصة بها.

يوضح الجدول ٢٣ خصائص المتطلبات الرئيسية الى Oracle. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف oracle.

*الجدول ٢٣. خصائص بيانات Oracle*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| ORACLE Version | نظام Windows | يتم التحقق من نسخة Oracle التي تم تركيبها على الجهاز | القيمة الحرفية المتوقعة يمكن أن تكون نسخ متعددة، مفصولة بفاصلات، على سبيل المثال:<br>9.2, 10.1, 10.2 |
| oracle.Client | كل | يتم التحقق من نسخة الوحدة التابعة الى Oracle التي تم تركيبها على الجهاز | القيمة الحرفية المتوقعة يمكن أن تكون نسخ متعددة، مفصولة بفاصلات، على سبيل المثال:<br>9.2.0.8+<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| oracle.Client.Location | كل | يتم التحقق من الدليل الرئيسي للوحدة التابعة الى Oracle | مجموعة حروف، على سبيل المثال:<br>/opt/oracle/products /10.1.0/ client_1 |

| | | | |
|---|---|---|---|
| oracle.Server | كل | يتم التحقق من نسخة وحدة خدمة Oracle التي يتم تركيبها حاليا على الجهاز | القيمة الحرفية المتوقعة يمكن أن تكون نسخ متعددة، مفصولة بفاصلات، على سبيل المثال:<br>10.2.0.4g,11g R1<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| oracle.Server.Location | كل | يتم التحقق من الدليل الرئيسى لوحدة خدمة قاعدة بيانات Oracle | مجموعة حروف، على سبيل المثال:<br>/opt/oracle/product/10.1.0/Db_1 |

## خصائص بيانات نظام التشغيل

تقوم خصائص بيانات نظام التشغيل بالتحقق من المتطلبات الرئيسية لنظام التشغيل مثل النسخة وهيكل البناء واجمالي الذاكرة والذاكرة المتاحة والذاكرة الفعلية الاجمالية. بالنسبة لأنظمة Windows فقط، يتم استخدام أدوات تجميع VBScript لنظام التشغيل في الدليل *ips_root*/lib مع بادئة كود التعريف os في أسماء الملفات الخاصة بها. بالنسبة لأنظمة UNIX فقط، يتم استخدام أدوات تجميع نظام التشغيل UNIX في الدليل *ips_root*/UNIX_Linux مع بادئة كود التعريف os في أسماء الملفات الخاصة بها.

يوضح الجدول ٢٤ خصائص المتطلبات الرئيسية لنظام التشغيل. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف os.

*الجدول ٢٤. خصائص بيانات نظام التشغيل*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| os.architecture | كل | يتم التحقق من هيكل بناء النظام | 32-bit\|64-bit |
| os.automount | نظام UNIX | يتم التحقق مما اذا كانت خصائص automount تعمل | قيمة بوولية، على سبيل المثال:<br>True |
| os.autoUpdate Enabled | نظام Windows | يتم التحقق مما اذا كان قد تم اتاحة Windows Update آليا؛ يتم ارجاع True اذا تم اتاحته | قيمة بوولية، على سبيل المثال:<br>True |
| os.available Memory | نظام Windows | يتم التحقق من حجم الذاكرة الظاهرية المتاح حاليا لكن غير مستخدم بواسطة نظام التشغيل | النسق الرقمي بالميجابايت، على سبيل المثال:<br>900MB |

| | | | |
|---|---|---|---|
| os.dir.*dir_name* | نظام UNIX | يتم التحقق من نظام الملفات *dir_name* بناء على الخواص المميزة المؤهلة التالية:<br>• الخاصية المميزة dir، لتحديد نظام الملفات الذي سيتم فحصه.<br>• الخاصية المميزة type، لتحديد الخاصية المميزة لنظام التشغيل التي سيتم التحقق منها، على سبيل المثال، التمثيل المكون من أرقام ثمانية *octal_digits* لتصاريح التوصل الى نظام الملفات هذا<br><br>*dir_name* يمكن أن تمثل:<br>• tmp<br>• home | مجموعة حروف بالنسق المؤهل التالي:<br>[dir:*dir_name*, type:permission] *octal_digits*+<br><br>على سبيل المثال، للتحقق مما اذا كان الدليل الرئيسى له تصاريح drwxr-xr-x:<br>os.dir.home=[dir:/home, type:permission]755+ |
| os.diskquota | | يتم التحقق من حصة استخدام القرص للمستخدم الذي قام ببدء الاتصال؛ ويتم ارجاع القيمة للحصة بالكيلوبايت أو Unlimited | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• Number لتمثيل الكيلوبايت، على سبيل المثال، 414000<br>• String لتمثل حصة القرص غير المحدودة، على سبيل المثال، Unlimited |
| os.expectLink | نظام UNIX | يتم التحقق مما اذا كان امتداد Expect الى TCL متاحا على الجهاز؛ ويتم ارجاع Available اذا كان بالحالة متاح<br><br>**ملاحظة:** تقوم خاصية المتطلبات الرئيسية os.file.expect بالتحقق مما اذا كان قد تم تركيب امتداد Expect على الجهاز. | Available\|Unavailable |
| os.file.*script_name* | نظام UNIX | يتم التحقق مما اذا كان البرنامج النصي *script_name* متاحا على الجهاز.<br>*script_name* يمكن أن تمثل:<br>• bash<br>• expect<br>• gzip<br>• tar | قيمة بوولية، على سبيل المثال:<br>True |
| os.Firefox | نظام UNIX | يتم التحقق مما اذا كان قد تم تركيب Mozilla Firefox على الجهاز؛ يتم ارجاع Available اذا كان قد تم تركيبه | Available\|Unavailable |

| | | | |
|---|---|---|---|
| os.FreePaging Space | نظام UNIX | يتم التحقق من اجمالي حجم الذاكرة الوسيطة للصفحة المتاحة | نسق رقمي بالجيجابايت أو الميجابايت، على سبيل المثال:<br>4GB+<br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| os.ftpusers | نظام UNIX | يتم التحقق مما اذا كان المستخدم الرئيسي معروض في الملف ftpusers الذي يحدد المستخدمين الذين لا يسمح لهم بامتيازات بدء الاتصال الى ftp؛ يتم ارجاع Available اذا كان المستخدم غير معروض | Available\|Unavailable |
| os.gnu.tar | نظام UNIX | يتم التحقق مما اذا كانت وظيفة GNU tar متاحة على الجهاز؛ يتم ارجاع Available اذا كان قد تم تركيبها | Available\|Unavailable |
| os.hostformat | نظام UNIX | يتم التحقق مما اذا كانت الادخالات في /etc/host بالنسق الصحيح | قيمة بوولية، على سبيل المثال:<br>True |
| os.iodevice status | نظام AIX | يتم التحقق من حالة المدخلات/المخرجات الغير متزامنة (aio0)، حيث تعد عملية الجزء المركزي لنظام التشغيل لتحسين أداء عمليات المدخلات/المخرجات؛ يتم ارجاع Available اذا كان بالحالة متاح. | Available\|Unavailable |
| os.is8dot3File<br><br>FormatEnabled | نظام Windows | يتم التحقق مما اذا كان يتم تطبيق نسق اسم ملف 8.3 آليا؛ يتم ارجاع True اذا كان يتم تطبيقه | قيمة بوولية، على سبيل المثال:<br>True |
| os.localhost InHostsFile | كل | يتم التحقق مما اذا كان هناك ادخال في ملف الأنظمة الرئيسية يقوم بمناظرة النظام الرئيسي المحلي لعنوان IP 127.0.0.1، على سبيل المثال:<br>127.0.0.1 localhost | قيمة بوولية، على سبيل المثال:<br>True |
| os.isService Running.*service_name* | نظام Windows | استخدم مصطلح التسمية هذا للتحقق مما اذا كان يتم تشغيل *service_name* على الجهاز. *service_name* يمكن أن تمثل ما يلي:<br>• remoteRegistry الى Remote Registry Service<br>• DNSClient الى DNS Client Service<br>• terminalServices الى Remote Desktop Services أو الخدمات عن بعد | قيمة بوولية، على سبيل المثال:<br>True |

| | | | |
|---|---|---|---|
| os.kernelMode | نظام AIX | يتم التحقق من أن هيكل بناء وحدة CPU يدعم نمط الجزء المركزي لنظام التشغيل أو نمط غير مقيد | 32-bit\|64-bit |
| os.kernel Parameters | نظام Linux | يتم التحقق مما اذا كانت معاملات الجزء المركزي لنظام التشغيل متاحة لنظام التشغيل | Available\|Unavailable |
| os.kernelversion | نظام Linux | يتم التحقق من اصدار الجزء المركزي لنظام التشغيل لأنظمة تشغيل Linux | النسق الرقمي، على سبيل المثال، 2.6 |
| os.largeFile | نظام UNIX | يتم التحقق من دعم الملفات الكبيرة | قيمة بوولية، على سبيل المثال:<br>True |
| os.ldLibPath | نظام UNIX | يتم التحقق من وجود متغير بيئة التشغيل LD_LIBRARY_PATH وانتهائه بالعلامة :، بالشكل<br>os.ldLibPath=<br>[endsWith=:] | Available\|Unavailable |
| os.level | نظام AIX | يتم التحقق مما اذا كان نظام التشغيل AIX أكبر من المستوى ١٠ الى AIX النسخة 5.3 أو أكبر من المستوى ٣ الى AIX النسخة 6.1 | قيمة بوولية، على سبيل المثال:<br>True |

| | | | |
|---|---|---|---|
| os.lib.*lib_name_version* | نظام UNIX | يتم التحقق من أنه تم تركيب النسخة المدعمة للمكتبة *lib_name* على الجهاز. مجموعة الحروف أو التعبير المعتاد الذي يمثل *lib_name_version*، على سبيل المثال، بخط عريض:<br>• مكتبة **libstdc++.so.#** ذات ٣٢-بت<br>• مكتبة **libstdc++.so.#** ذات ٦٤-بت<br>• مكتبة **libXft.so.#** ذات ٣٢-بت<br>• مكتبة **libXtst.so.#** ذات ٣٢-بت<br>• مكتبة **libaio.so.#** ذات ٦٤-بت<br>• مستوى بيئة تشغيل **xlC.rte** XLC ذات ٣٢-بت<br>• بيئة تشغيل **xlC.aix50.rte** XLC ذات ٣٢-بت الى AIX النسخة 5.3<br>• بيئة تشغيل **xlC.aix61.rte** XLC ذات ٣٢-بت الى AIX النسخة 6.1<br>• مكتبة AIX IOCP **bos.iocp.rte**<br>• **bos.loc.iso.en_us**، فئة الملف لكود ISO لنظام التشغيل الأساسي AIX | القيمة يمكن أن تكون بأي من الأنواع التالية:<br><br>مجموعة حروف، على سبيل المثال:<br>• /usr/lib/libstdc++.so.# كقيمة للمكتبة libstdc++.so.# ذات ٣٢-بت<br>• /usr/lib64/libaio.so.# كقيمة للمكتبة libaio.so.# ذات ٦٤-بت<br>• xlC.aix50.rte.9.0.0.8+ كقيمة لبيئة التشغيل xlC.aix50.rte XLC ذات ٣٢-بت الى AIX النسخة 5.3<br>• bos.loc.iso.en_us لفئة ملف كود ISO<br><br>regex{*str*}، التعبير المعتاد مع معامل المدخلات، *str*، يمثل نموذج البحث لاسم المكتبة، على سبيل المثال:<br>regex{.*libgcc.*}<br><br>يتم التحقق من نسخة مكتبة التشغيل ذات المستوى المنخفض GCC، libgcc، التي يوجد لها نظام التشغيل هذا.<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| os.loginVariable | نظام UNIX | يتم التحقق مما اذا كان قد تم تحديد المسارات المفترضة للمستخدم الرئيسي في المتغيرات PATH و SUPATH؛ يتم ارجاع Available اذا كان قد تم تحديدها | Available\|Unavailable |
| os.maximo Directory | نظام UNIX | يتم التحقق مما اذا كان دليل /export/home/maximo متاحا | Available\|Unavailable |
| os.maximoDir Owner | نظام UNIX | يتم التحقق من مالك الدليل /export/home/maximo | maximo |
| os.maximum Processes | نظام UNIX | يتم التحقق من الحد الأقصى لعدد العمليات التي يمكن تشغيلها لكل مستخدم. | العدد، على سبيل المثال، 2048 |
| os.Mozilla Version | نظام UNIX | يتم التحقق من نسخة Mozilla Firefox على الجهاز بخلاف خاصية المتطلبات الرئيسية os.Firefox | النسق الرقمي، على سبيل المثال، الخاصية المميزة 3.0+<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |

| | | | |
|---|---|---|---|
| `os.mountcheck` | نظام UNIX | يتم التحقق مما اذا كان قد تم تركيب نظام الملفات بناءا على الخصائص المميزة المؤهلة التالية:<br>• `drive`، لتحديد الدليل الذي يعد نظام ملفات تم تركيبه<br>• الخاصية المميزة `nosuid`، لتحديد ما اذا كان اختيار التحميل قد تم تحديده أم لا عند تحديد نظام الملفات | مجموعة حروف بالنسق المؤهل التالي:<br>`[drive:`*`dir_name`*`, `*`mount_option`*`: false\|true] True\|False`<br><br>على سبيل المثال، للتحقق مما اذا كان الدليل `/home` موجودا ومن عدم تحديد الاختيار `nosuid`:<br>`os.mountcheck=[drive: /home, nosuid:false]True` |
| `os.package.`*`package_name`* | نظام UNIX | يتم التحقق من أنه تم تركيب النسخة المدعمة لمجموعة البرامج *package_name* على الجهاز.<br>مجموعة حروف تمثل *package_name*، على سبيل المثال، بخط عريض:<br>• الوحدة الطرفية **bash**<br>• **expect** لمجموعة برامج امتداد TCL<br>• **libgcc** لمجموعة برامج بيئة التشغيل ذات المستوى المنخفض الى GCC<br>• **openssh** للوحدة الطرفية المؤمنة للمصدر المفتوح<br>• **openssl** لمجموعة أدوات المصدر المفتوح SSL/TLS<br>• **perl** لمجموعة البرامج النصي الى Perl<br>• **rpm** لمجموعات برامج بناء RPM أو RPM<br>• **telnet** لمجموعة برامج Telnet<br>• **wget** لمجموعة برامج استرجاع ملف GNU | مجموعة حروف، على سبيل المثال:<br>• `bash-3.2+` الى bash shell<br>• `expect-1.2.0` الى Expect<br>• `libgcc-3.4.3-9` الى libgcc<br>• `openssh-5.0.0.5301-` الى openssh<br>• `openssl-4.2.0-` الى OpenSSL<br>• `perl-5.8.2` الى Perl<br>• `rpm`<br>• `telnet`<br>• `wget`<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| `os.pagesize` | نظام UNIX | يتم التحقق من حجم الصفحة للنظام . | النسق الرقمي بالكيلوبايت، على سبيل المثال:<br>`4KB`<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| `os.RAMSize` | كل | يتم التحقق من RAM للنظام | النسق الرقمي بالجيجابايت، على سبيل المثال، `8GB` |

| | | | |
|---|---|---|---|
| os.SeaMonkey Version | نظام UNIX | يتم التحقق من نسخة Mozilla SeaMonkey معينة على الحاسب لأنه يتم تحديد المسار الخاص بها في متغير بيئة التشغيل PATH | النسق الرقمي، على سبيل المثال، 2.10<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| os.SELinux | نظام Linux | للتحقق من حالة الفرض لخاصية Linux لتحسين السرية بناء على الخواص المميزة المؤهلة التالية:<br>• الخاصية المميزة source، لتحديد الأمر الذي سيتم استخدامه لنظام التشغيل المتعلق | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• مجموعة حروف بالنسق المؤهل التالي:<br>[source:Command] Disabled\|Enabled<br><br>على سبيل المثال، للتحقق مما اذا كانت الخاصية ستكون غير متاحة أو يتم السماح بها على نظام تشغيل Red Hat أو SUSE:<br>os.SELinux=[source: Command]Disabled<br>• مجموعة حروف بدون مؤهل، حيث يكون نظام التشغيل Linux عام:<br>os.SELinux=Disabled |
| os.servicePack | كل | يتم التحقق من النسخة الحالية لمجموعة الخدمات التي تم تركيبها | النسق الرقمي، ذو *majorVersion*. *minorVersion* أو نسخة *majorVersion* فقط<br><br>على سبيل المثال، للتحقق من أنه تم تركيب مجموعة الخدمات ٢ أو الأحدث، خصائص المتطلبات الرئيسية 2+<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| os.shell.default | نظام UNIX | يتم التحقق مما اذا كان قد تم تركيب البرنامج النصي للوحدة الطرفية | مجموعة حروف لتمثيل البرنامج النصي للوحدة الطرفية، على سبيل المثال، bash |

os.space.*dir_name*

تحتوي الأداة Prerequisite Scanner على ثلاثة أشكال مختلفة للخاصية os.space.*dir_name*:

- الخاصية os.space.*dir_name* التي تتحقق مما اذا كان هناك مساحة قرص متاحة وكافية لنظام الملفات المحدد بصرف النظر عن ما اذا كان المستخدم الذي قام ببدء الاتصال هو مستخدم رئيسي أو غير رئيسي دائما.

  استخدم اختلاف خاصية المتطلبات الرئيسية هذا عندما تريد التحقق من المسار المحدد لنظام الملفات، واذا كان لا يهم ما اذا كان المستخدم الذي قام ببدء الاتصال مستخدم رئيسي أو غير رئيسي دائما.

  **ملاحظة:** لا يمكنك استخدام هذا الاختلاف مرتين لنفس نظام الملفات لكن يمكن استخدامه لأنواع مختلفة من المستخدمين في ملف توصيف واحد؛ بدلا من ذلك، استخدم مجموعة مكونة من اثنين من الاختلافات الأخرى.

- الخاصية os.space.*dir_name*_nonroot التي تتحقق مما اذا كان هناك مساحة قرص متاحة لنظام الملفات المحدد لمستخدم غير رئيسي.

  استخدم اختلاف خاصية المتطلبات الرئيسية هذا عند بدء الاتصال كمستخدم غير رئيسي، وعندما تريد التحقق صراحة من المسار المحدد لنظام الملفات.

  **ملاحظة:** يجب أن يكون المستخدم غير الرئيسي هو نفس المستخدم الذي يقوم بتركيب المنتج على النظام المستهدف.

- الخاصية os.space.*dir_name*_root التي تتحقق مما اذا كان هناك مساحة قرص متاحة وكافية لنظام الملفات المحدد للمستخدم الرئيسي.

  استخدم اختلاف خاصية المتطلبات الرئيسية هذا عند بدء الاتصال كمستخدم رئيسي، وعندما تريد التحقق صراحة من المسار المحدد لنظام الملفات.

يمكنك تحديد الاختلافات os.space.*dir_name*_nonroot و os.space.*dir_name*_root في نفس ملف التوصيف. يقوم Prerequisite Scanner باخراج NOT_REQ_CHECK_ID في خانة النتائج الفعلية للاختلاف الذي لا يمكن تطبيقه. على سبيل المثال، اذا كان المستخدم الذي قام ببدء الاتصال هو مستخدم رئيسي، ستقوم أداة Prerequisite Scanner باخراج NOT_REQ_CHECK_ID للاختلاف os.space.*dir_name*_nonroot.

| | | | |
|---|---|---|---|
| os.space.*dir_name* | نظام UNIX | يتم التحقق من مساحة القرص المتاحة لنظام ملفات <*dir_name*> المحدد بواحدة أو أكثر من الخصائص المميزة المؤهلة التالية:<br>• الخاصية المميزة dir، لتحديد المسار الذي سيتم فحصه من نظام الملفات<br>• الخاصية المميزة unit، لتحديد وحدات قرص التخزين التي سيتم استخدامها<br><br>تتوقف قيمة الخاصية المميزة dir على الأشخاص القائمين بالاتصال؛ وعلى ذلك، فان القيمة هي زوج قيمة الاسم التي تمثل نوع المستخدم، أي الدليل الرئيسي أو غير الدليل الرئيسي، والمسار المتعلق.<br><br>*dir_name* يمكن أن تمثل:<br>• home<br>• opt<br>• tmp<br>• usr<br>• var<br>**ملاحظة:** لا يمكنك استخدام هذا الاختلاف مرتين لنفس نظام الملفات لكن يمكن استخدامه لأنواع مختلفة من المستخدمين في ملف توصيف منفرد. استخدم مجموعة مكونة من الاختلافات os.space.*dir_name*_nonroot و os.space.*dir_name*_root. | مجموعة حروف بالنسق المؤهل التالي لنظام الملفات لمستخدم رئيسي:<br>[dir:**root**=<*dir_path*>, unit:<*unit_name*>] <*disk_space*><br><br>على سبيل المثال:<br>os.space.usr= [dir:root=/usr/ibm/ common/acsi, unit:GB]200<br><br>مجموعة حروف بالنسق المؤهل التالي لنظام الملفات لمستخدم غير رئيسي:<br>[dir:**non_root**=<*dir_ path*>, unit:<*unit_name*>] <disk_space><br><br>على سبيل المثال:<br>os.space.home= [dir:non_root=USERHOME/ .acsi_HOST, unit:MB]200<br><br>مجموعة حروف بالنسق المؤهل التالي، تقوم باستخدام مؤهل واحد فقط:<br>[dir:<*dir_path*>] <*disk_space*> MB<br><br>على سبيل المثال:<br>os.space.home= [dir:/home/sat]250MB<br><br>نسق رقمي بالجيجابايت أو الميجابايت، على سبيل المثال:<br>os.space.opt=11GB |

| | | | |
|---|---|---|---|
| os.space.*dir_name*_ nonroot | نظام UNIX | يتم التحقق من مساحة القرص المتاحة لنظام ملفات *dir_name* لمستخدم غير رئيسي، بناءا على واحدة أو أكثر من الخصائص المميزة المؤهلة التالية:<br>• الخاصية المميزة dir، لتحديد المسار الذي سيتم فحصه من نظام الملفات<br>• الخاصية المميزة unit، لتحديد وحدات قرص التخزين التي سيتم استخدامها<br><br>*dir_name* يمكن أن تمثل:<br>• home<br>• opt<br>• tmp<br>• usr<br>• var | مجموعة حروف بالنسق المؤهل التالي لنظام الملفات لمستخدم غير رئيسي:<br>[dir:**non_root**=*dir_path*, unit:*unit_name*] disk_space<br><br>على سبيل المثال:<br>os.space.home_nonroot= [dir:non_root= USERHOME/.acsi_HOST, unit:MB]200<br><br>مجموعة حروف بالخاصية المميزة المؤهلة dir لنظام الملفات فقط لمستخدم غير رئيسي:<br>[dir:**non_root**=*dir_path*] *disk_space*GB\|MB<br><br>على سبيل المثال:<br>os.space.opt_nonroot= [dir:non_root=/opt/IBM/ ITM] 1024MB |
| os.space.*dir_name*_ root | نظام UNIX | يتم التحقق من مساحة القرص المتاحة لنظام الملفات *dir_name* للمستخدم الرئيسي، بناءا على واحدة أو أكثر من الخصائص المميزة المؤهلة التالية:<br>• الخاصية المميزة dir، لتحديد المسار الذي سيتم فحصه من نظام الملفات<br>• الخاصية المميزة unit، لتحديد وحدات قرص التخزين التي سيتم استخدامها<br><br>*dir_name* يمكن أن تمثل:<br>• home<br>• opt<br>• tmp<br>• usr<br>• var | مجموعة حروف بالنسق المؤهل التالي لنظام الملفات لمستخدم رئيسي:<br>[dir:**root**=*dir_path*, unit:*unit_name*] *disk_space*<br><br>على سبيل المثال:<br>os.space.usr_root= [dir:root= /usr/ibm/common/acsi, unit:GB]200<br><br>مجموعة حروف بالخاصية المميزة المؤهلة dir لنظام الملفات فقط لمستخدم رئيسي:<br>[dir:**root**=*dir_path*] *disk_space*GB\|MB<br><br>على سبيل المثال:<br>os.space.opt_root= [dir:root=/opt/IBM/ITM] 1024MB |
| os.sshdConfig | نظام UNIX | يتم التحقق مما اذا كان قد تم توصيف خاصية بدء اتصال المستخدم الرئيسي المصرح له لجلسات SSH daemon | Available\|Unavailable |

| | | | |
|---|---|---|---|
| os.swapSize | نظام UNIX | يتم التحقق مما اذا كانت الذاكرة الاضافية يجب أن تكون أكبر من حجم RAM أو اجمالي حجم الذاكرة الاضافية | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• قيمة بوولية، على سبيل المثال:<br>True<br>• نسق رقمي بالميجابايت أو الجيجابايت، على سبيل المثال:<br>2GB |
| os.tmpdir | نظام UNIX | يتم التحقق من تصاريح التوصل التي تم تخصيصها لنظام الملفات /tmp، بما في ذلك أية تصاريح خاصة تم تحديدها بواسطة شارات امكانية التوصل، على سبيل المثال sticky أو setuid أو setgid bits في الأرقام الثمانية. | رقم لتمثيل الأرقام الثمانية octal_digits لتصاريح التوصل<br><br>على سبيل المثال، للتحقق مما اذا كان الدليل المؤقت يحتوي على تصاريح drwxrwxrwt مع اتاحة تصريح sticky bit:<br>1777<br><br>كمثال آخر، للتحقق من أن الدليل المؤقت يحتوي على تصاريح drwxrwxrwx باستثناء sticky bit:<br>777 |
| os.totalMemory | نظام Windows | اجمالي حجم الذاكرة الظاهرية التي يستطيع نظام التشغيل التوصل اليها | نسق رقمي بالميجابايت أو الجيجابايت، على سبيل المثال: 4GB |
| os.totalPhysical Memory | نظام Windows | اجمالي حجم الذاكرة الفعلية التي يستطيع نظام التشغيل التوصل اليها، لكن لا يشير الى الحجم الصحيح للذاكرة الفعلية على الحاسب المستهدف | نسق رقم بالميجابايت أو الجيجابايت، على سبيل المثال: 2030MB |

| | | | |
|---|---|---|---|
| os.ulimit | نظام UNIX | يتم التحقق مما اذا كان يمكن تشغيل عدد العمليات غير المحدودة بناءا على الخصائص المميزة المؤهلة التالية:<br>• الخاصية المميزة type، لتحديد الحد الاضافي الذي سيتم التحقق منه، على سبيل المثال، يقوم filedescriptor limit بالتحقق من الحد لعدد موصفات الملف التي يمكن أن تقوم العملية بفتحها<br><br>تبادليا، يتم التحقق مما اذا كان قد تم تحديد الحدود التالية للنطاقات المحددة في الملف /etc/security/limits.conf:<br>root - stack unlimited<br>ctginst1 - stack unlimited<br>root - nofile 8192<br>tioadmin - nofile 32767 | القيمة يمكن أن تكون بأي من الأنواع التالية:<br>• مجموعة حروف بالنسق المؤهل التالي:<br>[type:*limit_name*] *limit_value*, limited\|unlimited<br><br>على سبيل المثال، للتحقق مما اذا كان حد موصف الملف أكبر من 8192، مع حدد غير محدود من العمليات:<br>os.ulimit=[type: filedescriptorlimit] 8192+, unlimited<br>الأنواع الصحيحة للحدود التي سيتم التحقق منها، حيث *limit_name* تمثل نوع الحد كما يلي:<br>– ALL، للتحقق من كل الحدود<br>– corefilesizelimit<br>– datasegmentlimit<br>– filedescriptor limit<br>– filesizelimit<br>– hardlimit<br>– processlimit<br>– maxmemorysize limit<br>– maxprocesseslimit<br>– stacksizelimit<br>– threadlimit<br>• Available\|Unavailable لتحديد ما اذا كانت النطاقات المتعلقة تحتوي على فئات الحد في الملف /etc/security/limits.conf. |
| os.umask | نظام UNIX | يتم التحقق من التصاريح لقناع تكوين نمط الملف | رقم لتمثيل الأرقام الثمانية *octal_digits* لتصاريح التوصل. على سبيل المثال، للتحقق من أن الملفات الجديدة قابلة للكتابة فقط للمالك، حدد الرقم الثماني ليكون 0022 |
| os.userLimits | نظام UNIX | يتم التحقق مما اذا كان أقصى حجم للرصة غير محدود؛ يتم ارجاع Available اذا كان غير محدود | Available\|Unavailable |

| | | | |
|---|---|---|---|
| `os.versionNumber` | نظام Windows | يتم التحقق من النسخة الحالية لنظام التشغيل التي تم تركيبها على الجهاز | النسق الرقمي، على سبيل المثال، `5.0+`<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| `os.windowManager` | نظام UNIX | يتم التحقق مما اذا كان GNOME أو KDE متاحا كسطح مكتب بياني | `Available\|Unavailable` |

## خصائص بيانات البرامج التي تم تركيبها

تقوم خصائص بيانات البرامج التي تم تركيبها بالتحقق من المتطلبات الرئيسية للبرامج التي تم تركيبها مثل البرامج التي تم تسجيلها في سجل Windows وما اذا كان قد تم تركيب cygwin و gskit. بالنسبة لأنظمة Windows فقط، يتم استخدام أدوات تجميع البرامج التي تم تركيبها في الدليل *ips_root*`/lib` مع بادئة كود التعريف `installedSoftware` أو `cygwin` أو `gskit` في أسماء الملفات الخاصة بها.

يوضح الجدول ٢٥ خصائص المتطلبات الرئيسية للبيانات العامة. هذه الفئة من خصائص المتطلبات الرئيسية لا تتطلب بادئة كود التعريف.

*الجدول ٢٥ . خصائص بيانات البرامج التي تم تركيبها*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| `installedSoftware` | نظام Windows | يتم اجراء مسح لسجل نظام التشغيل للبرامج التي تم تركيبها مع الأماكن | مجموعة حروف، مع عدة تطبيقات مفصولة بفاصلة. |
| `cygwinVersion` | نظام Windows | يتم التحقق مما اذا كان قد تم تركيب نسخة cygwin على الجهاز؛ يتم ارجاع 0.0 اذا لم يتم تركيب أية نسخة | رقم صحيح موجب، على سبيل المثال، `1.5`<br><br>**ملاحظة:** القيم يمكن أن تقوم باستخدام الحروف الخاصة كما هو موضح في الجدول ١. |
| `gskit7Version` | نظام Windows | يتم التحقق مما اذا كان قد تم تركيب gskit Version 7 على الجهاز؛ يتم ارجاع 0.0 اذا لم يكن قد تم تركيب النسخة 7 | رقم صحيح موجب، على سبيل المثال، `7.0` |
| `gskit8Version` | نظام Windows | يتم التحقق مما اذا كان قد تم تركيب gskit Version 8 على الجهاز؛ يتم ارجاع 0.0 اذا لم يكن قد تم تركيب النسخة 8 | رقم صحيح موجب، على سبيل المثال، `8.0` |

## خصائص البيانات للمستخدم

تقوم خصائص بيانات المستخدم بالتحقق من المتطلبات الرئيسية للمستخدم مثل ما اذا كان يتوافر لدى المستخدم الذي قام ببدء الاتصال صلاحيات ادارية أو يعد مستخدم رئيسي. بالنسبة لأنظمة Windows فقط، يتم استخدام أداة التجميع للمستخدم في الدليل *ips_root*`/lib`، مع بادئة كود التعريف `user` في أسماء الملفات الخاصة بها. بالنسبة لأنظمة UNIX فقط، يتم استخدام أداة التجميع للمستخدم في *ips_root*`/lib/packageTest.sh`.

يوضح الجدول ٢٦ خصائص المتطلبات الرئيسية للمستخدم. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف user.

*الجدول ٢٦ . خصائص البيانات للمستخدم*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| user.userID | نظام Windows | كود المستخدم الحالي الذي قام ببدء الاتصال | مجموعة حروف، على سبيل المثال، smithj |
| user.isAdmin | كل | يتم التحقق مما اذا كان المستخدم الذي قام ببدء الاتصال هو أحد أعضاء مجموعة موجه النظام | قيمة بوولية، على سبيل المثال True |

## خصائص بيانات شبكة الاتصال لنظام Windows

تقوم خصائص بيانات شبكة اتصال Windows بالتحقق من المتطلبات الرئيسية لشبكة الاتصال مثل اتاحة NetBIOS و DHCP على الجهاز وخصائص اختبار الاتصال. يتم استخدام أدوات تجميع شبكة الاتصال لنظام Windows في الدليل *ips_root*/lib، مع بادئة كود التعريف network في أسماء الملفات الخاصة بها. .

يوضح الجدول ٢٧ خصائص المتطلبات الرئيسية لشبكة الاتصال العامة عبر كل بيئات تشغيل Windows. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف network.

*الجدول ٢٧ . خصائص بيانات شبكة الاتصال لنظام Windows*

| خاصية المتطلبات الرئيسية | الوصف | القيم الصالحة |
|---|---|---|
| network.DHCPEnabled | يتم التحقق مما اذا كان هناك موفق واحد على الأقل ذو عنوان IP صحيح قام بالحصول على عنوان IP هذا باستخدام DHCP؛ يتم ارجاع True اذا كان هناك واحد على الأقل. | قيمة بوولية، على سبيل المثال False |
| network.netBIOSEnabled | يتم التحقق مما اذا كان هناك موفق واحد على الأقل ذو عنوان IP تم اتاحة NetBIOS له كبروتوكول؛ يتم ارجاع True اذا كان هناك واحد على الأقل. | قيمة بوولية، على سبيل المثال True |
| network.pingLocalhost | يتم التحقق مما اذا كان النظام الرئيسي المحلي يستجيب الى بروتوكول اختبار الاتصال؛ يتم ارجاع True اذا كان يستجيب. | قيمة بوولية، على سبيل المثال True |
| network.pingSelf | يتم التحقق مما اذا كان قد تم حل اسم الحاسب المحلي باستخدام DCHP وما اذا كان يمكن اختبار الاتصال به؛ يتم ارجاع True اذا أمكن اختبار الاتصال به. | قيمة بوولية، على سبيل المثال True |
| network.ValidateHostsFile | يتم التحقق مما اذا كانت الادخالات في C:\WINDOWS\system32\drivers\etc\hosts بنسق صحيح؛ يتم ارجاع True اذا كان النسق صحيح. | قيمة بوولية، على سبيل المثال True |

## خصائص بيانات شبكة الاتصال لنظام UNIX

تقوم خصائص بيانات شبكة اتصال UNIX بالتحقق من المتطلبات الرئيسية لشبكة الاتصال مثل اتاحة NetBIOS و DHCP على الجهاز وخصائص اختبار الاتصال. يتم استخدام أدوات تجميع شبكة الاتصال في الدليل *ips_root*/UNIX_Linux.

يوضح الجدول ٢٨ خصائص المتطلبات الرئيسية لشبكة الاتصال العامة عبر كل بيئات تشغيل UNIX. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف network.

*الجدول ٢٨ . خصائص بيانات شبكة الاتصال لنظام UNIX*

| خاصية المتطلبات الرئيسية | الوصف | القيم الصالحة |
|---|---|---|
| network.DHCPEnabled | يتم التحقق مما اذا كان تم هناك موفق واحد على الأقل ذو عنوان IP صحيح قام بالحصول على عنوان IP باستخدام DHCP | قيمة بوولية، على سبيل المثال False |
| network.dns | يتم التحقق مما اذا كان ادخال DNS لجهاز النظام الرئيسي صحيح | قيمة بوولية، على سبيل المثال True |
| network.fqdn | يتم التحقق مما اذا كان قد تم تحديد اسم النطاق المصنف بالكامل لجهاز النظام الرئيسي | قيمة بوولية، على سبيل المثال True |
| network.pingLocalhost | يتم التحقق مما اذا كان النظام الرئيسي المحلي يستجيب الى بروتوكول اختبار الاتصال | قيمة بوولية، على سبيل المثال True |
| network.pingSelf | يتم التحقق مما اذا كان قد تم حل اسم الحاسب المحلي باستخدام DCHP وما اذا كان يمكن اختبار الاتصال به | قيمة بوولية، على سبيل المثال True |

## خصائص بيانات متغير بيئة التشغيل

تقوم خصائص بيانات متغير بيئة التشغيل بالتحقق من المتطلبات الرئيسية لمتغير بيئة التشغيل والتي يمكن أن تكون عامة عبر كل بيئات التشغيل مثل ما اذا كان قد تم تحديد متغير بيئة التشغيل أو قيمة متغير بيئة التشغيل. بالنسبة لأنظمة Windows فقط، يتم استخدام أدوات تجميع متغير بيئة التشغيل في الدليل *ips_root*/lib، مع بادئة كود التعريف env في أسماء الملفات الخاصة بها. وبالنسبة لأنظمة UNIX فقط، يتم استخدام أدوات تجميع متغير بيئة التشغيل UNIX في الدليل *ips_root*/UNIX_Linux، مع بادئة كود التعريف env في أسماء الملفات الخاصة بها.

يوضح الجدول ٢٩ خصائص متغير بيئة التشغيل التي تكون عامة عبر كل بيئات التشغيل. هذه الفئة من خصائص المتطلبات الرئيسية تتطلب بادئة كود التعريف env.

*الجدول ٢٩ . خصائص بيانات متغير بيئة التشغيل*

| خاصية المتطلبات الرئيسية | بيئة التشغيل | الوصف | القيم الصالحة |
|---|---|---|---|
| env.var.set.*env_var_name* | نظام UNIX | استخدم مصطلح التسمية هذا للتحقق مما اذا كان تم تحديد متغير بيئة التشغيل *env_var_name* على الحاسب، على سبيل المثال:<br>• يقوم env.var.set.HOME بالتحقق مما اذا كان قد تم تحديد متغير بيئة التشغيل للدليل الرئيسي، حيث *env_var_name* هو اسم متغير بيئة تشغيل HOME. | قيمة بوولية، على سبيل المثال True |

| | | | |
|---|---|---|---|
| | | • يقوم `env.var.set.JAVA_HOME` بالتحقق مما اذا كان قد تم تحديد متغير بيئة التشغيل للدليل الرئيسي الى Java، حيث *env_var_name* هو اسم متغير بيئة التشغيل `JAVA_HOME` | |
| `env.var.set.`*`env_var_name`*`[type:`*`env_var_type`*`]` | نظام Windows | استخدم مصطلح التسمية هذا للتحقق مما اذا كان قد تم تحديد متغير بيئة التشغيل *env_var_name* لنوع متغير بيئة التشغيل المحدد *env_var_type*، على سبيل المثال:<br>• يقوم `env.var.set.HOME` بالتحقق مما اذا كان قد تم تحديد متغير بيئة التشغيل للدليل الرئيسي، حيث *env_var_name* هو اسم متغير بيئة تشغيل `HOME`.<br>• يقوم `env.var.set.JAVA_HOME[type:User]` بالتحقق مما اذا كان قد تم تحديد متغير بيئة التشغيل للدليل الرئيسي Java للمستخدم الذي قام ببدء الاتصال، حيث *env_var_name* هو اسم متغير بيئة التشغيل `JAVA_HOME` و *env_var_type* هو نوع `User` لمتغير بيئة التشغيل.<br><br>نوع متغير بيئة التشغيل *env_var_type* اختياري ويمثل أنواع متغيرات بيئة التشغيل المدعمة بواسطة نظام تشغيل Windows كما يلي:<br>• `Process`<br>• `System`<br>• `User`<br>• `Volatile`<br><br>يكون النوع المفترض اذا لم يتم تحديده هو `Process`. | قيمة بوولية، على سبيل المثال `True` |
| `env.classpath.derbyJAR` | كل | يتم التحقق مما اذا كان المسار الى ملف Derby JAR متضمن في متغير بيئة تشغيل مسار الفئة | قيمة بوولية، على سبيل المثال `True` |
| `env.CIT.homeExists` | نظام Windows | يتم التحقق من وجود كلا من متغيرات بيئة التشغيل `HOMEDRIVE` و `HOMEPATH` | قيمة بوولية، على سبيل المثال `True` |

# الملحق د. أدوات التجميع سابقة التعريف لأنظمة UNIX

توجد أدوات تجميع منفردة لعمليات التحقق من خصائص المتطلبات الرئيسية على أنظمة UNIX توجد في الدليل *ips_root*/lib. يمكنك مراجعة أدوات التجميع هذه ومعاملات المدخلات الخاصة بها قبل تكوين أدوات التجميع المهيأة.

يوضح الجدول ٣٠ أدوات التجميع المعرفة مسبقا لأنظمة UNIX.

*الجدول ٣٠. أدوات التجميع لنظام UNIX*

| أدوات التجميع | الخاصية المتطلبات الرئيسية | المدخلات |
|---|---|---|
| DB2_Version | DB2 Version | لا شيء |
| DBType | DBType | لا شيء |
| DBTypeDetails | DBTypeDetails | لا شيء |
| env.classpath.derbyJAR | env.classpath.derbyJAR | لا شيء |
| env.var.set | env.var.set*env_var_name*<br><br>*env_var_name* هو اسم متغير بيئة التشغيل الذي سيتم التحقق منه | *$env_var_name* |
| network.DHCPEnabled | network.DHCPEnabled | لا شيء |
| network.dns | network.dns | لا شيء |
| network.fqdn | network.fqdn | لا شيء |
| network.pingSelf | network.pingSelf | لا شيء |
| network.port | network.available Ports.*<br><br>network.portsInUse.* | $ports |
| oracle.Client | oracle.Client | لا شيء |
| oracle.Client.Location | oracle.Client.Location | لا شيء |
| oracle.Server | oracle.Server | لا شيء |
| oracle.Server.Location | oracle.Server.Location | لا شيء |
| os.architecture | os.architecture | 32 bit\|64 bit |
| os.automount | os.automount | لا شيء |
| os.cmd | os.lookup | nslookup |
| os.cmd | os.tar\|os.gnu.tar | tar\|gtar |

| | | |
|---|---|---|
| os.dir | os.dir.*dir_name*<br><br>*dir_name* يمكن أن تمثل ما يلي:<br>• tmp<br>• home | مجموعة حروف بالنسق التالي:<br>[dir:*dir_name*, type:permission] *octal_digits*+<br><br>على سبيل المثال، للتحقق مما اذا كان الدليل *dir_name*، الذي يمثل، الدليل الرئيسي يتوافر لديه تصاريح drwxr-xr-x:<br>[dir:/home, type:permission]755+ |
| os.diskquota | os.diskquota | لا شيء |
| os.expectLink | os.expectLink | لا شيء |
| os.filepath | os.file.*script_name*<br><br>*script_name* يمكن أن تمثل:<br>• bash<br>• expect<br>• gzip<br>• tar | المسار لملف البرنامج النصي، حيث أن المسار يمكن أن يكون:<br>• /bin/bash<br>• /usr/bin/expect<br>• /usr/bin/gzip<br>• /usr/bin/tar |
| os.Firefox | os.Firefox | لا شيء |
| os.FreePagingSpace | os.FreePagingSpace | لا شيء |
| os.ftpusers | os.ftpusers | لا شيء |
| os.hostformat | os.hostformat | لا شيء |
| os.iodevicestatus | os.iodevicestatus | لا شيء |
| os.kernelMode | os.kernelMode | لا شيء |
| os.kernelParameters | os.kernelParameters | لا شيء |
| os.kernelversion | os.kernelversion | لا شيء |
| os.largeFile | os.largeFile | لا شيء |
| os.ldLibPath | os.ldLibPath | البرنامج |
| os.level | os.level | لا شيء |

| | | |
|---|---|---|
| os.lib | os.lib.*lib_name_version*<br><br>مجموعة الحروف أو التعبير المعتاد الذي يمثل *lib_name_version*، على سبيل المثال، بخط عريض:<br>• مكتبة **libstdc++.so.#** ذات ٣٢-بت<br>• مكتبة **libstdc++.so.#** ذات ٦٤-بت<br>• مكتبة **libXft.so.#** ذات ٣٢-بت<br>• مكتبة **libXtst.so.#** ذات ٣٢-بت<br>• مكتبة **libaio.so.#** ذات ٦٤-بت<br>• مستوى بيئة تشغيل **xlC.rte** XLC ذات ٣٢-بت<br>• بيئة تشغيل **xlC.aix50.rte** XLC ذات ٣٢-بت الى AIX النسخة 5.3<br>• بيئة تشغيل **xlC.aix61.rte** XLC ذات ٣٢-بت الى AIX النسخة 6.1<br>• مكتبة AIX IOCP **bos.iocp.rte**<br>• **bos.loc.iso.en_us**، فئة الملف لكود ISO لنظام التشغيل الأساسي AIX<br><br>**regex{*str*}**، التعبير المعتاد مع معامل المدخلات، *str*، يمثل نموذج البحث لاسم المكتبة، على سبيل المثال، .*libgcc.* | *path_to_library*<br>أو *lib_name_version*<br><br>على سبيل المثال، للتحقق من القيمة الفعلية لخاصية المتطلبات الرئيسية os.lib.libstdc++.so، ستكون معاملات المدخلات هي /usr/lib/libstdc++.so.5 و libstdc++.so.:<br>os.lib /usr/lib/libstdc++.so.5<br>libstdc++.so<br><br>على سبيل المثال، للتحقق من وجود نسخة مكتبة التشغيل ذات المستوى المنخفض GCC، libgcc، على الجهاز، سيكون معامل المدخلات هو:<br>regex{.*libgcc.*} |
| os.loginVariable | os.loginVariable | لا شيء |
| os.maximoDirectory | os.maximoDirectory | لا شيء |
| os.maximoDirOwner | os.maximoDirOwner | لا شيء |
| os.maximumProcesses | os.maximumProcesses | لا شيء |
| os.MozillaVersion | os.MozillaVersion | لا شيء |

| | | |
|---|---|---|
| os.mountcheck | os.mountcheck | مجموعة حروف بالنسق المؤهل التالي:<br>[drive:*dir_name*, *mount_option*: false\|true] True\|False<br><br>على سبيل المثال، للتحقق مما اذا كان الدليل home/ موجودا ومن عدم تحديد الاختيار nosuid:<br>os.mountcheck=[drive: /home, nosuid:false]True |
| os.package | os.package.*package_name*<br><br>مجموعة حروف تمثل *package_name*، على سبيل المثال، بخط عريض:<br>• الوحدة الطرفية **bash**<br>• **expect** لمجموعة برامج امتداد TCL<br>• **libgcc** لمجموعة برامج بيئة التشغيل ذات المستوى المنخفض الى GCC<br>• **openssh** للوحدة الطرفية المؤمنة للمصدر المفتوح<br>• **openssl** لمجموعة أدوات المصدر المفتوح SSL/TLS<br>• **perl** لمجموعة البرامج النصي الى Perl<br>• **rpm** لمجموعات برامج بناء RPM أو RPM<br>• **telnet** لمجموعة برامج Telnet<br>• **wget** لمجموعة برامج استرجاع ملف GNU | *package_name*، على سبيل المثال، حيث *package_name* هو rpm:<br>os.package rpm |
| os.pagesize | os.pagesize | لا شيء |
| os.RAMSize | os.RAMSize | لا شيء |

| | | |
|---|---|---|
| os.SELinux | os.SELinux | • مجموعة حروف بالنسق المؤهل التالي:<br>[source:Command] Disabled\|Enabled<br><br>على سبيل المثال، للتحقق مما اذا كانت الخاصية ستكون غير متاحة أو يتم السماح بها على نظام تشغيل Red Hat أو SUSE:<br>os.SELinux=[source: Command]Disabled<br>• اذا لم يكن هناك مؤهل، لن يتم امرار أية قيمة الى أداة التجميع. |
| os.servicePack | os.servicePack | قيمة مجموعة الخدمات |
| os.shell.default | os.shell.default | القيمة المتوقعة لخاصية المتطلبات الرئيسية، على سبيل المثال، bash |
| os.space | os.space.*dir_name*<br><br>*dir_name* يمكن أن تمثل ما يلي:<br>• usr<br>• home<br>• tmp<br>• var | مجموعة حروف بالنسق المؤهل التالي لنظام الملفات لمستخدم رئيسي:<br>[dir:**root**=*dir_path*, unit:*unit_name*] *disk_space*<br><br>على سبيل المثال:<br>os.space.usr= [dir:root=/usr/ibm/ common/acsi, unit:GB]200<br><br>مجموعة حروف بالنسق المؤهل التالي لنظام الملفات لمستخدم غير رئيسي:<br>[dir:**non_root**=*dir_path*, unit:*unit_name*] disk_space<br><br>على سبيل المثال:<br>os.space.home= [dir:non_root=USERHOME/ .acsi_HOST, unit:MB]200<br><br>مجموعة حروف بالنسق المؤهل التالي، تقوم باستخدام مؤهل واحد فقط:<br>[dir:*dir_path*] *disk_space* MB<br><br>على سبيل المثال:<br>os.space.home= [dir:/home/sat]250MB |
| os.sshdConfig | os.sshdConfig | لا شيء |

| | | |
|---|---|---|
| os.swapSize | os.swapSize | لا شيء |
| os.tmpdir | os.tmpdir | لا شيء |
| os.ulimit | os.ulimit | مجموعة حروف بالنسق المؤهل التالي:<br>`[type:`*`limit_name`*`] `*`limit_value`*`, limited\|unlimited`<br><br>على سبيل المثال، للتحقق مما اذا كان حد موصف الملف أكبر من ٨١٩٢، مع حدد غير محدود من العمليات:<br>`os.ulimit= [type:filedescriptor limit] 8192+,unlimited`<br><br>الأنواع الصحيحة للحدود التي سيتم التحقق منها، حيث *limit_name* تمثل نوع الحد كما يلي:<br>• ALL، للتحقق من كل الحدود<br>• corefilesizelimit<br>• datasegmentlimit<br>• filedescriptorlimit<br>• filesizelimit<br>• hardlimit<br>• processlimit<br>• maxmemorysizelimit<br>• maxprocesseslimit<br>• stacksizelimit<br>• threadlimit |
| os.umask | os.umask | لا شيء |
| os.userLimits | os.userLimits | لا شيء |
| os.windowManager | os.windowManager | لا شيء |

# الملحق ه. الوظائف المعتادة لأنظمة Windows

Prerequisite Scanner له مجموعة من الوظائف المعتادة في الملف `/lib/common_function.vbs` لتشغيل عمليات تحقق على أنظمة Windows.

*الجدول ٣١. الوظائف في common_function.vbs*

| الوظيفة | الوصف |
|---|---|
| "allFiles()" | لقراءة أسماء الملف في الدليل المحدد بأحد المصفوفات. |
| "arrayToString()" | لتكوين تمثيل من مجموعة حروف للمصفوفة. |
| "bigthan()" | لاحتساب الفرق بين القيمة المتوقعة والفعلية لخاصية المتطلبات الرئيسية اذا حجم هذه الخاصية بالميجابايت أو الجيجابايت. |
| "changeMG()" | لتحويل معامل المدخلات الى الميجابايت أو الجيجابايت لمساحة القرص أو الذاكرة لخصائص المتطلبات الرئيسية. |
| "checkItemToString()" | لتكوين تمثيل من مجموعة حروف لعنصر `CheckItem`. |
| "dictionaryToString()" | لتكوين تمثيل من مجموعة حروف لعنصر قاموس البرمجة. |
| "exeCommand()" | لتنفيذ الأمر المحدد وارجاع نتائج تنفيذ الأمر. |
| "filterCommand()" | لتنفيذ الأمر المحدد وارجاع الأسطر من نتائج الأمر التي تطابق النموذج المحدد. |
| "filterFile()" | لقراءة وترشيح المحتويات الخاصة بالملف لعنصر قاموس البرمجة. |
| "findNewest()" | لايجاد أحدث ملف توصيف. |
| "findSuitableFile()" | لايجاد ملف التوصيف المتعلق لأحد البرامج أو النسخ. |
| "fmt()" | يقوم بتعديل مجموعة الحروف باضافة عدد الحروف المحدد من مجموعة حروف أخرى وتذييل مجموعة الحروف الأخرى بمسافات اذا كان طول مجموعة الحروف الأخرى قصير جدا أو يتم اقتطاع مجموعة الحروف الأخرى اذا كانت طويلة جدا. |
| "formatForDisplay()" | لتنسيق معامل المدخلات لجعله مقروءا. |
| "formatSizeForDisplay()" | يقوم بأخذ معامل المدخلات والحاق أو تقليل جزء الكسر من معامل المدخلات الى اثنين النقاط العشرية، على سبيل المثال، ١٢٣ ميجابايت الى ١٢٣,٠٠ ميجابايت أو ١٢,١٢٣ ميجابايت الى ١٢,١٢ ميجابايت. |
| "getDecimalSeparator()" | لتحديد الفاصل العشري الذي يتم استخدامه لمحددات اللغة الحالية. |
| "getFirstMatch()" | لاحضار أول مطابقة لعبارة البحث في المصفوفة. |
| "isMatch()" | للتحقق مما اذا كانت عبارة البحث موجودة في مجموعة الحروف. |
| "notInLatter()" | يتم ترشيح المصفوفة الأولى لتحديد ما اذا كانت المحتويات موجودة في المصفوفة الثانية. وبناء على قيمة معامل المدخلات `in_or_not`، ستقوم الوظيفة بارجاع محتويات أول مصفوفة بما يتضمن أو يستبعد ما هو مطابق للمصفوفة الثانية. |

| | |
|---|---|
| "passOrFail()" | يتم مقارنة القيم المتوقعة والفعلية لخاصية المتطلبات الرئيسية ويتم تحديد ما اذا كانت الخاصية ستتخطى عملية التحقق. معاملات المدخلات يمكن أن تكون أرقام عامة، أو بالميجابايت أو الجيجابايت، وبسرعة معالج بالميجاهيرتز أو الجياجاهيرتز، أو بوولية، أو مجموعة حروف. |
| "ppread()" | يتم قراءة محتويات الملف الى عنصر قاموس البرمجة، ويتم المزيد من التقسيم لكل من الأسطر في الملف من خلال معامل مدخلات الفاصل المحدد اذا كان الفاصل موجود في السطر. |
| "readFile()" | يتم قراءة كل من أسطر الملف في ادخال الفهرس للمصفوفة. |
| "unitMGTOG()" | يتم اضافة المحتويات الخاصة بالمصفوفة للحصول على العدد الاجمالي من الميجابايت. |
| "varToString()" | لتكوين تمثيل من مجموعة حروف للمتغير. المتغير الذي سيتم التحقق منه يمكن أن يكون من النوع string أو number أو scripting dictionary object أو array أو CheckItem. |

## allFiles()

لقراءة أسماء الملف في الدليل المحدد بأحد المصفوفات.

### الغرض

تقوم هذه الوظيفة بالحصول على كشف الملفات في معامل مدخلات الدليل واضافتها الى المصفوفة. تقوم بارجاع المصفوفة.

### الصيغة

```
allFiles(filepath)
```

### معاملات المدخلات

**String *filepath***

المسار الخاص بالدليل الذي يتضمن الملفات.

### القيم الناتجة

**Array *fileNames***

تقوم بارجاع المصفوفة التي تتضمن أسماء الملفات في الدليل المحدد.

## arrayToString()

لتكوين تمثيل من مجموعة حروف للمصفوفة.

### الغرض

تقوم هذه الوظيفة بأخذ المصفوفة التي يتم تمريرها كمعامل مدخلات ويتم ارجاع مجموعة حروف تمثل المحتويات الخاصة بالمصفوفة.

### الصيغة

```
arrayToString(arr)
```

**معاملات المدخلات**

**Array *arr***

يتضمن المصفوفة.

**القيم الناتجة**

**String *result***

لارجاع تمثيل من مجموعة حروف للمصفوفة، مع فصل كل من البنود بفاصلة.

---

## bigthan()

لاحتساب الفرق بين القيمة المتوقعة والفعلية لخاصية المتطلبات الرئيسية اذا حجم هذه الخاصية بالميجابايت أو الجيجابايت.

**الغرض**

تقوم هذه الوظيفة أولا باستدعاء الوظيفة "changeMG()" لتغيير القيم المتوقعة والفعلية لخاصية المتطلبات الرئيسية الى الميجابايت اذا لزم الأمر. وستقوم بعد ذلك بالتحقق مما اذا كانت القيمة المرتجعة من الوظائف بقيم خالية، وما اذا كانت أي من القيم خالية، وما اذا كانت القيمة المرتجعة للوظيفة هي بالميجابايت ومن وجود الوظيفة. وهي تقوم بالتحقق من الميجابايت أو الجيجابايت بأي من القيم، وتقوم بتحويل الميجابايت اذا لزم الأمر. وستقوم باحتساب الفرق بين القيم المنسقة النهائية وارجاع النتائج.

**الصيغة**

```
bigthan(expect,real)
```

**معاملات المدخلات**

**String *expect***

القيمة المتوقعة لخاصية المتطلبات الرئيسية.

**String *real***

القيم الفعلية لخاصية المتطلبات الرئيسية.

**القيم الناتجة**

**String *bigthan***

تقوم بارجاع الفرق في MBs أو 0MBs اذا لم يكن هناك اختلاف.

---

## changeMG()

لتنسيق معامل المدخلات لازالة أي من حروف تجميع الخانات الاضافية منها وتقوم بارجاع معامل منسق ما لم يتضمن معامل المدخلات على ميجابايت أو جيجابايت. واذا كانت كذلك، سيتم تحويل معامل المدخلات الى الجيجابايت أو الميجابايت على التوالي.

**الغرض**

تقوم هذه الوظيفة باستدعاء الوظيفة "getDecimalSeparator()" لتحديد الفاصل العشري لمحددات اللغة الحالية ثم يتم ازالة أي حروف لتجميع الخانات الاضافية لهذه اللغة من معامل مدخلات الرقم. ستقوم بعد ذلك باستدعاء الوظيفة "getFirstMatch()" لتحديد ما اذا كانت القيمة بالميجابايت أم بالجيجابايت ثم تقوم بتحويل القيمة الى الجيجابايت أو الميجابايت على التوالي.

**الصيغة**

```
changeMG(tochange)
```

**معاملات المدخلات**

**String *tochange***

يتضمن نسق القيمة ويتم تحويلها كما يلزم.

**القيم الناتجة**

**String *changeMG***

تقوم بارجاع رقم منسق بدون حروف لتجميع الخانة أو الرقم بالميجابايت أو الجيجابايت.

---

## checkItemToString()

لتكوين تمثيل من مجموعة حروف لعنصر CheckItem.

**الغرض**

تقوم هذه الوظيفة باحضار العنصر CheckItem الذي يتم تمريرها كمعامل مدخلات وتقوم بارجاع تمثيل من مجموعة حروف لمقارنة القيمة للخصائص المختلفة لهذه النسخة من العنصر CheckItem.

**الصيغة**

```
checkItemToString(var)
```

**معاملات المدخلات**

**CheckItem *var***

تتضمن نسخة من عنصر CheckItem.

**القيم الناتجة**

**String *result***

تقوم بارجاع تمثيل من مجموعة حروف لخصائص العنصر CheckItem كما يلي:

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

---

## dictionaryToString()

لتكوين تمثيل من مجموعة حروف لعنصر قاموس البرمجة.

**الغرض**

تقوم هذه الوظيفة بأخذ عنصر القاموس الذي يتم تمريره كمعامل مدخلات ويتم ارجاع مجموعة حروف تمثل المحتويات الخاصة بعنصر القاموس.

**الصيغة**

```
dictionaryToString(dic)
```

**معاملات المدخلات**

**Dictionary *dic***

تتضمن عنصر القاموس.

**القيم الناتجة**

**String *result***

لارجاع تمثيل من مجموعة حروف لعنصر القاموس، مع كل من المفاتيح والبنود وتكون مفصولة بعلامات يساوي.

---

## exeCommand()

لتنفيذ الأمر المحدد وارجاع نتائج تنفيذ الأمر.

**الغرض**

هذه الوظيفة تقوم بتنفيذ معامل مدخلات الأمر. اذا كانت هناك أي أخطاء، فستقوم باستدعاء البرنامج الفرعي `logWarning` لعرض الأخطاء؛ خلاف ذلك، ستقوم بارجاع النتيجة من تنفيذ الأمر.

**الصيغة**

```
exeCommand(cmd)
```

**معاملات المدخلات**

**String *cmd***

اسم الأمر الذي سيتم تنفيذه.

**القيم الناتجة**

**String *result***

تقوم بارجاع مجموعة حروف تتضمن النتيجة من تنفيذ الأمر.

---

## filterCommand()

لتنفيذ الأمر المحدد وارجاع الأسطر من نتائج الأمر التي تطابق النموذج المحدد.

**الغرض**

هذه الوظيفة تقوم بتنفيذ معامل مدخلات الأمر. وهي تقوم بتحليل نتائج تنفيذ الأمر لغويا ويتم التحقق مما اذا كان أي سطر من النتيجة يتطابق مع معامل المدخلات للسطر. واذا كان هناك تطابق، ستقوم باستدعاء الوظيفة "getFirstMatch()" لتحديد ما اذا كان هناك تطابق أيضا بين معامل مدخلات معلومات السطر ونتائج الأمر. واذا كان كذلك، ستقوم بعد ذلك باستخدام الوظيفة `Join` لارجاع المحتويات الى عنصر القاموس من الوظيفة `getFirstMatch()`.

**الصيغة**

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

**معاملات المدخلات**

**String *cmd***

اسم الأمر الذي سيتم تنفيذه.

**String *line_patt***

نموذج السطر الذي سيتم البحث طبقا لها في النتائج من تنفيذ الأمر.

**Number *after_line***

عدد الأسطر التي سيتوقف بعدها البحث عن المعلومات.

**String *info_patt***

النموذج أو العبارة التي سيتم البحث بها بكل من أسطر نتائج الأمر.

**القيم الناتجة**

**String *filterCommand***

تقوم بارجاع المحتويات لنموذج القاموس كمجموعة حروف واحدة.

---

## filterFile()

لقراءة وترشيح المحتويات الخاصة بالملف لعنصر قاموس البرمجة.

**الغرض**

تقوم هذه الوظيفة بقراءة كل من أسطر الملف وتمرير كل سطر يتضمن عبارة البحث الى الوظيفة "getFirstMatch()". واذا قامت بارجاع تطابق وكان السطر غير موجود بالفعل في عنصر القاموس، سيتم كتابة السطر في عنصر القاموس. وسيتم تشغيل الوظيفة حتى الوصول الى نهاية الملف وبعد ذلك يتم ارجاع عنصر القاموس.

**الصيغة**

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

**معاملات المدخلات**

**String *fileName***

الملف الذي سيتم ترشيحه.

**String *patt***

النموذج أو العبارة التي سيتم البحث بها بكل من أسطر الملف.

**القيم الناتجة**

**Dictionary *dic.keys***

يتم ارجاع عنصر القاموس *dic* بالأسطر المرشحة من الملف.

---

## findNewest()

لايجاد أحدث ملفات التوصيف في المصفوفة.

### الغرض

تقوم هذه الوظيفة بالمرور خلال المصفوفة وتحديد الملف الأحدث في المصفوفة. وهذا يقوم بارجاع اسم الملف.

### الصيغة

```
findNewest(arr)
```

### معاملات المدخلات

**Array *arr***

يتضمن مجموعة من ملفات التوصيف التي سيتم التحقق منها.

### القيم الناتجة

**String *result***

تقوم بارجاع اسم أحدث ملفات التوصيف.

---

## findSuitableFile()

لايجاد ملف التوصيف المتعلق لأحد البرامج أو النسخ.

### الغرض

تقوم هذه الوظيفة باستدعاء الوظيفة "getFirstMatch()" لاحضار مجموعة الملفات التي لها معامل مدخلات اللاحقة كلاحقة الملف من كشف الملفات المرتجعة من خلال الوظيفة "allFiles()". وستقوم بعد ذلك باستدعاء الوظيفة "getFirstMatch()" مرة أخرى لارجاع مجموعة الملفات التي تتضمن معامل المدخلات لكود البرنامج في اسم الملف. وستقوم بعد ذلك باستدعاء نفس الوظيفة لارجاع مجموعة الملفات التي تتضمن معامل المدخلات للنسخة في اسم الملف. اذا كانت الوظائف تقوم بايجاد واحد أو أكثر من الملفات المطابقة للنسخة، ستقوم باستدعاء الوظيفة "findNewest()" لاحضار أحدث نسخة من هذا الملف ويتم ارجاع اسم الملف؛ خلاف ذلك، تقوم بارجاع الملف `common.bat` أو تقوم باستخدام `logScreen` و `logWarning` قبل ارجاع أحدث نسخة من ملف التوصيف لكود البرنامج.

### الصيغة

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

### معاملات المدخلات

**String *pd***

كود البرنامج الخاص بالملف الذي يتم ايجاده، كما هو محدد في ملف كود البرنامج، الملف *`ips_root`*`/codename.cfg`.

**String *version***

نسخة البرنامج الخاصة بالملف الذي سيتم ايجاده. <*version*> هو كود مكون من ٨-أرقام يمثل النسخة، والاصدار، والتعديلات، والمستوى، مع رقمين لكل من أجزاء الكود؛ على سبيل المثال 7.3.21 هو 07032100.

**String *suf***

اللاحقة الخاصة بنوع الملف الذي سيتم ايجاده، على سبيل المثال، `cfg` أو `bat`.

**String *filepath***

المسار الخاص بالدليل الذي يتضمن الملف الذي سيتم ايجاده.

**القيم الناتجة**

**String *findSuitableFile***

تقوم بارجاع أسماء الملفات الحالية بناء على النتائج الخاصة بالوظائف التي يتم استدعائها:

- `pd_version.cfg`، أحدث نسخة من الملف لكود ونسخة البرنامج المصاحب.
- `common.bat` اذا كانت قيمة معامل المدخلات للاحقة الملف هي `bat`.
- `pd_version.cfg`، أحدث نسخة من ملف التوصيف العام للبرنامج، اذا كان لا يتم ايجاد الملف المتضمن لمعامل المدخلات للنسخة.

---

## fmt()

يقوم بتعديل مجموعة الحروف باضافة عدد الحروف المحدد من مجموعة حروف أخرى وتذييل مجموعة الحروف الأخرى بمسافات اذا كان طول مجموعة الحروف الأخرى قصير جدا أو يتم اقتطاع مجموعة الحروف الأخرى اذا كانت طويلة جدا.

**الغرض**

تقوم هذه الوظيفة بالبحث عن التعبير `s#%` داخل معامل المدخلات s لمجموعة حروف النوع. التعبير `s#%` يحدد # عدد الحروف من معامل المدخلات ***args*** التي يتم اضافتها الى مجموعة الحروف في موضع ما من التعبير. اذا كان # المحدد أكبر من هذا طول معامل المدخلات ***args***، فسيم تذييل الفرق بمسافات خالية. اذا كان # المحدد أقل من طول معامل المدخلات ***args*** هذا، فسيم قطع الطول بمقدار الفرق. اذا كان # المحدد يساوي ٠، ويتم اضافة الطول الكامل لمعامل المدخلات ***args*** الى مجموعة الحروف الأولى في المكان المناسب من مجموعة الحروف.

**الصيغة**

```
fmt(s, args)
```

**معاملات المدخلات**

**String *s***

تتضمن مجموعة الحروف التي سيتم تعديلها من خلال عدد الحروف # المحدد في التعبير `s#%` داخل مجموعة الحروف هذه.

**Array *args***

تتضمن مجموعة الحروف التي تقوم بتعديل معامل المدخلات *s*.

**القيم الناتجة**

**String *result***

تقوم بارجاع مجموعة الحروف المعدلة.

**مثال**

`fmt("Hello %5s!",array("Neo"))` تقوم بارجاع `"Hello Neo  !"` يتم تذييلها بالمزيد من المسافات

`fmt("Hello %5s!",array("Mr. Anderson"))` تقوم بارجاع `"Hello Mr. A!"` يتم قطعها لاضافة `"Mr. A"` فقط

`fmt("Hello %0s!",array("Mr. Anderson"))` تقوم بارجاع `"Hello Mr. Anderson!"`

## formatForDisplay()

لتنسيق معامل المدخلات لجعله مقروءا.

### الغرض

تقوم هذه الوظيفة باستدعاء الوظيفة "formatSizeForDisplay()" لتنسيق معامل المدخلات.

### الصيغة

```
formatForDisplay(val)
```

### معاملات المدخلات

**Variable *val***

المتغير الذي سيتم تنسيقه.

### القيم الناتجة

**String *vartoString***

تقوم بارجاع نتيجة استدعاء الوظيفة "formatSizeForDisplay()".

## formatSizeForDisplay()

يقوم بأخذ معامل المدخلات والحاق أو تقليل جزء الكسر من معامل المدخلات الى اثنين النقاط العشرية، على سبيل المثال، ١٢٣ ميجابايت الى ١٢٣,٠٠ ميجابايت أو ١٢,١٢٣ ميجابايت الى ١٢,١٢ ميجابايت.

### الغرض

تقوم هذه الوظيفة باحصاء عدد الحروف في معامل المدخلات، ويتم التحقق مما اذا كان عبارة عن رقم أم مجموعة حروف، ويتم فصل معامل المدخلات الى قيمة صحيحة وكسر. وبناء على الجزء الكسري، سيتم الحاق أو قطعه الى اثنين من الأماكن العشرية. تقوم بارجاع النتيجة.

### الصيغة

```
formatSizeForDisplay(size)
```

### معاملات المدخلات

**Integer *size***

يتم تقريب القيمة الى رقمين عشريين.

### القيم الناتجة

**Integer *val***

يتم ارجاع القيمة مقربة الى رقمين عشريين.

## getDecimalSeparator()

لتحديد الفاصل العشري الذي يتم استخدامه لمحددات اللغة الحالية.

**الغرض**

تقوم هذه الوظيفة بتكوين رقم وظيفي ثم استخدام الوظيفة `Mid()` لتحديد الفاصل العشري الذي يتم استخدامه في هذا الرقم الوظيفي.

**الصيغة**

```
getDecimalSeparator()
```

**معاملات المدخلات**

**None**

**القيم الناتجة**

**Character *sep***

لارجاع فاصل عشري، على سبيل المثال ، أو . لمحددات اللغة.

---

## getFirstMatch()

لاحضار أول مطابقة لعبارة البحث في المصفوفة.

**الغرض**

تقوم هذه الوظيفة باستخدام التعبير المعتاد للبحث عن النموذج، والتي يتم امرارها كمعامل مدخلات، في المصفوفة التي يتم تمريرها كمعامل مدخلات. وعندما تقوم بايجاد أول مطابقة للنموذج في المصفوفة، وتقوم باضافة القيمة من المصفوفة لبرمجة عنصر القاموس.

**الصيغة**

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

**معاملات المدخلات**

**String *patt***

يتضمن العبارة التي يتم البحث عنها.

**Array *arr***

تتضمن المصفوفة التي سيتم البحث عن عبارة البحث بها.

**القيم الناتجة**

**Dictionary *keys***

تقوم بارجاع المفاتيح الخاصة بعنصر القاموس.

---

## isMatch()

للتحقق مما اذا كانت عبارة البحث موجودة في مجموعة الحروف.

**الغرض**

تقوم هذه الوظيفة باستدعاء الوظيفة "getFirstMatch()"، مع امرار النموذج ومجموعة الحروف (المتضمنة في مصفوفة) كمدخلات الى هذه الوظيفة. وتقوم باستدعاء الوظيفة ubound للتحقق مما اذا كانت القيمة التي تم ارجاعها من الوظيفة getFirstMatch() أكبر من أو تساوي ٠. اذا كانت أكبر من أو تساوي ٠، فانه توجد مطابقة؛ خلاف ذلك، لا توجد مطابقة.

**الصيغة**

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

**معاملات المدخلات**

**String *patt***

يتضمن العبارة التي يتم البحث عنها.

**String *str***

تتضمن مجموعة الحروف التي سيتم البحث عن عبارة البحث بها.

**القيم الناتجة**

**Boolean *True|False***

تقوم بارجاع True اذا كان هناك تطابق؛ خلاف ذلك، يتم ارجاع False.

---

## notInLatter()

يتم ترشيح المصفوفة الأولى لتحديد ما اذا كانت المحتويات موجودة في المصفوفة الثانية. وبناء على قيمة معامل المدخلات in_or_not، ستقوم الوظيفة بارجاع محتويات أول مصفوفة بما يتضمن أو يستبعد ما هو مطابق للمصفوفة الثانية.

**الغرض**

**الصيغة**

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

**معاملات المدخلات**

**Array *arr1***

نسخ من مكان ما

**Array *arr2***

الى مكان آخر

**String *in_or_out***

يحتوي على "in" أو "not" بناءا على ما اذا كان يجب أن تقوم الوظيفة بارجاع محتويات المصفوفة الأولى التي تم ترشيحها لارجاع المحتويات المطابقة للمصفوفة الثانية ("in") فقط أو المحتويات التي لا تطابق المصفوفة الثانية ("not").

**القيم الناتجة**

**Dictionary *keys***

يتم ارجاع مفاتيح عنصر القاموس الذي يحتوي على المصفوفة الأولى التي تم ترشيحها لتتضمن المحتويات المطابقة للمصفوفة الثانية (in_or_not = "in") فقط أو المحتويات التي لا تطابق المصفوفة الثانية أو (in_or_not = "not").

## passOrFail()

يتم مقارنة القيم المتوقعة والفعلية لخاصية المتطلبات الرئيسية ويتم تحديد ما اذا كانت الخاصية ستتخطى عملية التحقق. معاملات المدخلات يمكن أن تكون أرقام عامة، أو بالميجابايت أو الجيجابايت، وبسرعة معالج بالميجاهيرتز أو الجياجاهيرتز، أو بوولية، أو مجموعة حروف.

### الغرض

تقوم هذه الوظيفة أولا باستدعاء الوظيفة "changeMG()" لتنسيق واذا لزم الأمر تحويل القيم الفعلية والمتوقعة. تقوم بالتحقق مما اذا كانت القيمة صفرية، واذا كانت صفرية، يتم ارجاع "FAIL" والخروج. اذا لم تكن القيم صفرية، ستقوم الوظيفة بالتحقق مما اذا كانت القيم بوولية أو رقمية أو حجم، بالميجابايت أو الجيجابايت، أو سرعة وحدة CPU، بالميجاهرتز (نظام Windows فقط) أو الجيجاهرتز، أو مجموعات حروف. عندئذ تقوم بمقارنة القيم وارجاع النتيجة.

### الصيغة

```
passOrFail(expect,real)
```

### معاملات المدخلات

**String *expect***

القيمة المتوقعة لخاصية المتطلبات الرئيسية.

**String *real***

القيم الفعلية لخاصية المتطلبات الرئيسية.

### القيم الناتجة

**String *passOrFail***

تقوم بارجاع "PASS" أو "FAIL" طبقا لما اذا كانت القيمة المتوقعة تساوي أو لا تساوي القيمة الفعلية.

## ppread()

يتم قراءة محتويات الملف الى عنصر قاموس البرمجة، ويتم المزيد من التقسيم لكل من الأسطر في الملف من خلال معامل مدخلات الفاصل المحدد اذا كان الفاصل موجود في السطر.

### الغرض

تقوم هذه الوظيفة بقراءة كل سطر للملف، وازالة أية مسافات بادئة أو لاحقة، والتحقق مما اذا كان يحتوي على فاصل. اذا كان يحتوي على فاصل، فانها تقوم بتقسيم السطر بواسطة الفاصل، واضافة كل قطعة كبند الى عنصر القاموس؛ خلاف ذلك، تقوم باضافة السطر الذي تم تعديله الى بند في عنصر القاموس. وتقوم بارجاع مصفوفة تحتوي على عنصر القاموس على أنه الفهرس الأول.

### الصيغة

```
ppread(fileName, sep)
```

### معاملات المدخلات

**String *fileName***

اسم الملف الذي سيتم قراءته في عنصر القاموس.

**Character *sep***

الحرف الذي يمثل الفاصل الذي يتم تقسيم السطر به في الملف.

**القيم الناتجة**

**Array *array(dic)***
يتم ارجاع مصفوفة بعنصر القاموس (`dic`) على أنه الفهرس الأول.

**مثال**

المثال الذي سيتم تقديمه.

---

## readFile()

يتم قراءة كل من أسطر الملف في ادخال الفهرس للمصفوفة.

**الغرض**

تقوم هذه الوظيفة بفتح الملف وقراءة كل سطر للملف في ادخال الفهرس للمصفوفة. تقوم بارجاع المصفوفة.

**الصيغة**

```
readFile(fileName)
```

**معاملات المدخلات**

**String *fileName***
اسم الملف الذي سيتم قراءته في المصفوفة.

**القيم الناتجة**

**Array *fileContents***
يتم ارجاع المصفوفة مع محتويات الملف.

---

## unitMGTOG()

يتم اضافة المحتويات الخاصة بالمصفوفة للحصول على العدد الاجمالي من الميجابايت.

**الغرض**

هذه الوظيفة تقوم بتحويل قيمة كل من قيم المصفوفة الى الميجابايت واضافتهم معا.

**الصيغة**

```
unitMGTOG(arr)
```

**معاملات المدخلات**

**Array *arr***
يتضمن المصفوفة.

**القيم الناتجة**

**String *unitMGTOG***
تقوم بارجاع اجمالي محتويات المصفوفة بالميجابايت والحاق اللاحقة "`MB`" الى المجموع.

## varToString()

لتكوين تمثيل من مجموعة حروف للمتغير. المتغير الذي سيتم التحقق منه يمكن أن يكون من النوع string أو number أو scripting dictionary object أو array أو `CheckItem`.

### الغرض

تقوم هذه الوظيفة بالتحقق من البيانات أو نوع العنصر للمتغير وتقوم باستدعاء الوظيفة المتعلقة لتكوين تمثيل لمجموعة الحروف لهذه البيانات أو نوع العنصر.

*الجدول ٣٢. الوظيفة التي تم استدعائها لكل نوع متغير.*

| نوع المتغير | الوظيفة التي يتم استدعائها |
|---|---|
| المصفوفة | "arrayToString()" |
| العنصر `CheckItem` | "checkItemToString()" |
| عنصر قاموس البرنامج | "dictionaryToString()" |

### الصيغة

```
varToString(var)
```

### معاملات المدخلات

**Variable *var***

المتغيرات المدعمة هي: string، number، scripting dictionary object، array، CheckItem object

### القيم الناتجة

**String *vartoString***

تقوم بارجاع مجموعة حروف تمثل المتغير بما يتضمن القيم المرتجعة من أي من الوظائف التي تم استرجاعها كما يلزم.

# الملحق و. البرامج الفرعية لوظيفة التسجيل لأنظمة Windows

تحتوي أداة IBM Prerequisite Scanner على مجموعة من البرامج الفرعية العامة للتسجيل في الملف `preq.vbs` لعرض الرسائل على الشاشة أو كتابتها في ملف السجل.

يوضح الجدول ٣٣ وظائف التسجيل.

*الجدول ٣٣. البرامج الفرعية لوظيفة التسجيل*

| البرنامج الفرعي | الوصف | معاملات المدخلات |
|---|---|---|
| `deleteLogFile` | لحذف ملف السجل اذا كان موجودا. | لا شيء |
| `log(level, msg)` | لكتابة رسالة في ملف السجل باستخدام الوظيفة "fmt()". كما يتضمن السجل التاريخ والوقت الحالي. | • `level`، سلسلة حروف تحدد نوع الرسالة مثل اخبارية أو تحذير<br>• `msg`، مجموعة حروف تمثل رسالة لملف السجل |
| `logDebug(msg)` | يقوم باستدعاء الوظيفة `()log`، بتمرير `"DEBUG"` كمعامل لمستوى المدخلات. | `msg`، مجموعة حروف تمثل رسالة لملف السجل |
| `logError(msg)` | يقوم باستدعاء الوظيفة `()log`، بتمرير `"ERROR"` كمعامل لمستوى المدخلات. | `msg`، مجموعة حروف تمثل رسالة لملف السجل |
| `logInfo(msg)` | يقوم باستدعاء الوظيفة `()log`، بتمرير `"INFO"` كمعامل لمستوى المدخلات. | `msg`، مجموعة حروف تمثل رسالة لملف السجل |
| `logScreen(msg)` | يقوم بكتابة الرسالة في الشاشة. | `msg`، مجموعة حروف تمثل الرسالة التي سيتم كتابتها على الشاشة |
| `logScreenWith Replacement (msg, replaceStr)` | يتم كتابة الرسالة بالشاشة، وامرار كود الرسالة ومجموعة الحروف كمعاملات مدخلات. | • `msg`، كود الرسالة الذي يمثل نص الرسالة التي سيتم كتابتها على الشاشة<br>• `replaceStr`، مجموعة الحروف التي تقوم باستبدال *%variable* في قيمة كود الرسالة |
| `logScreenWith MultiReplacements (msg, replaceStrArray)` | يتم كتابة الرسالة في الشاشة، وامرار كود الرسالة ومصفوفة مجموعة الحروف كمعاملات مدخلات. | • `msg`، كود الرسالة الذي يمثل نص الرسالة التي سيتم كتابتها على الشاشة<br>• `replaceStrArray`، مصفوفة مجموعة الحروف مع استبدال كل فهرس في المصفوفة الى *%variable* في قيمة كود الرسالة |
| `logWarning(msg)` | يقوم باستدعاء الوظيفة `()log`، بتمرير `"WARNING"` كمعامل لمستوى المدخلات. | `msg`، مجموعة حروف تمثل رسالة لملف السجل |

# الملحق ز. البرامج الفرعية لوظيفة الملفات لنظام Windows

Prerequisite Scanner له مجموعة من البرامج الفرعية للملفات العامة في الملف `/lib/common_function.vbs` لمعالجة الملفات. كما أن لها مجموعة من الوظائف لمعالجة الملفات.

يوضح الجدول ٣٤ وظائف الملف.

*الجدول ٣٤. البرامج الفرعية لوظيفة الملف.*

| البرنامج الفرعي | الوصف | معاملات المدخلات |
|---|---|---|
| `appendToFile(text, fileName)` | لالحاق النص في نهاية الملف المحدد. | • `text`، مجموعة حروف تتضمن النص الذي سيتم الحاقها الى الملف<br>• `filename`، مجموعة حروف تمثل اسم الملف الذي سيتم تعديله |
| `writeToFile(text, fileName)` | تقوم بكتابة النص في الملف المحدد، واحلال المحتويات الحالية اذا لزم الأمر. | • `text`، مجموعة حروف تتضمن النص الذي سيتم كتابته في الملف<br>• `filename`، مجموعة حروف تمثل اسم الملف الذي سيتم تعديله |

يوضح الجدول ٣٥ وظائف الملف التي ستقوم بمعالجة الملفات.

*الجدول ٣٥. الوظائف العاملة للملفات*

| الوظيفة | الوصف |
|---|---|
| "allFiles()" | لقراءة أسماء الملف في الدليل المحدد بأحد المصفوفات. |
| "filterFile()" | لقراءة وترشيح المحتويات الخاصة بالملف لعنصر قاموس البرمجة. |
| "findNewest()" | لايجاد أحدث ملف توصيف. |
| "findSuitableFile()" | لايجاد ملف التوصيف المتعلق لأحد البرامج أو النسخ. |
| "ppread()" | يتم قراءة محتويات الملف الى عنصر قاموس البرمجة، ويتم المزيد من التقسيم لكل من الأسطر في الملف من خلال معامل مدخلات الفاصل المحدد اذا كان الفاصل موجود في السطر. |
| "readFile()" | يتم قراءة كل من أسطر الملف في ادخال الفهرس للمصفوفة. |

# الملحق ح. الوظائف المعتادة للبرامج الفرعية لأنظمة Windows

Prerequisite Scanner له مجموعة من الوظائف المعتادة الأخرى والبرامج الفرعية التي يتم استخدامها في الملفات المختلفة.

يوضح الوظائف المعتادة الأخرى والبرامج الفرعية الأخرى.

*الجدول ٣٦. الوظائف المعتادة الأخرى والبرامج الفرعية لأنظمة Windows*

| الوظيفة أو البرنامج الفرعي | الوصف |
|---|---|
| "ffirstMatch()" | لاحضار أول مطابقة لعبارة البحث في المصفوفة. |
| "getValue()" | لاحضار مساحة القرص المتاحة للدليل المحدد. |
| "removeSpecialCharacters()" | لازالة العلامة التجارية أو الحروف الخاصة الأخرى لجعل المقارنة أسهل. |
| "versionCompare()" | للتحليل اللغوي لمعاملات المدخلات التي تمثل القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية ومقارنتها لتحديد ما اذا كانت خاصية المتطلبات الرئيسية ستجتاز عملية التحقق من المتطلبات الرئيسية. الوظيفة تتوقف صيغة نسخة بفواصل من النقط كمعاملات مدخلات، على سبيل المثال، 1.0.0.4 أو 2.3 أو 3.40.26.7800 أو *.2.3. |

## ffirstMatch()

لاحضار أول مطابقة لعبارة البحث في المصفوفة.

### الغرض

تقوم هذه الوظيفة باستخدام التعبير المعتاد للبحث عن النموذج، والتي يتم امرارها كمعامل مدخلات، في المصفوفة التي يتم تمريرها كمعامل مدخلات. وعندما تقوم بايجاد أول مطابقة للنموذج في المصفوفة، وتقوم باضافة القيمة من المصفوفة لبرمجة عنصر القاموس.

### الوظائف الرئيسية

*الجدول ٣٧. الوظائف الرئيسية التي تقوم باستدعاء ffirstMatch()*

| الوظيفة الرئيسية، البرنامج النصي | الوصف |
|---|---|
| `ud620db2level(expect, real)` في `DB2_Version_compare.vbs` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية لنسخة DB2. |
| `oslevelcompare(expect, real)` في `OS_Version_compare.vbs` | لمقارنة القيم الفعلية والمتوقعة لخاصية نسخة نظام التشغيل للمتطلبات الرئيسية. |

### الصيغة

```
ffirstmatch(patt,arr)
```

### معاملات المدخلات

**String *patt***

يتضمن العبارة التي يتم البحث عنها.

**Array *arr***

تتضمن المصفوفة التي سيتم البحث عن عبارة البحث بها.

**القيم الناتجة**

**Dictionary *keys***

تقوم بارجاع المفاتيح الخاصة بعنصر القاموس.

---

## getValue()

لاحضار مساحة القرص المتاحة للدليل المحدد.

**الغرض**

يقوم هذا البرنامج الفرعي باستخدام نسخة عنصر نظام الملفات لاستدعاء الوظيفة ( ) `getDriveName` لمعامل مدخلات المسار ثم استخدام الخاصية `freeSpace` لاحضار مساحة القرص المتاحة، والتي سيتم تحويلها فيما بعد الى الميجابايت. معامل المدخلات لخاصية المتطلبات الرئيسية ويتم كتابة القيمة الخاصة بها في ملف نصي مؤقت مرافق لملف البرنامج النصي.

**البرامج**

*الجدول ٣٨. البرامج باستخدام getValue()*

| البرنامج النصي | الوصف |
|---|---|
| `DEZ_01040000.vbs` | برنامج نصي لتجميع خصائص المتطلبات الرئيسية وجعلها متاحة لملف توصيف DEZ 01040000 فقط |
| `LCM_TAD_common.vbs` | برنامج نصي لتجميع خصائص المتطلبات الرئيسية وجعلها متاحة لملفات التوصيف LCM 02300000 و TAD 07200000 فقط |
| `TAD722_impl.vbs` | برنامج نصي لتجميع خصائص المتطلبات الرئيسية وجعلها متاحة لملف توصيف TAD 07220000 فقط |

**الصيغة**

`getValue fso، sKey، drvPath`

**معاملات المدخلات**

**عنصر نظام الملفات *fso***

نسخة عنصر نظام الملفات.

**String *sKey***

تتضمن مجموعة حروف بنفس خاصية المتطلبات الرئيسية وعلامة التساوي.

**String *drvPath***

تتضمن المسار الذي سيتم الحصول على مساحة القرص المتاحة له.

**القيم الناتجة**

**None**

## removeSpecialCharacters()

لازالة العلامة التجارية أو الحروف الخاصة الأخرى لجعل المقارنة أسهل. الوظيفة موجودة في الملف `./lib/common.vbs`.

### الغرض

تقوم هذه الوظيفة باستدعاء الوظيفة `Replace()` لاستبدال العلامة التجارية وعبارة حقوق النشر، والرموز المسجلة بالحروف " ".

### الصيغة

```
removeSpecialCharacters(s)
```

### معاملات المدخلات

**String *s***

يتضمن مجموعة الحروف التي يجب نقل الحروف منها

### القيم الناتجة

**String *s***

تقوم بارجاع مجموعة الحروف بدون الحروف الخاصة.

## versionCompare()

للتحليل اللغوي لمعاملات المدخلات التي تمثل القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية ومقارنتها لتحديد ما اذا كانت خاصية المتطلبات الرئيسية ستجتاز عملية التحقق من المتطلبات الرئيسية. الوظيفة تتوقف صيغة نسخة بفواصل من النقط كمعاملات مدخلات، على سبيل المثال، 1.0.0.4، 2.3، 3.40.26.7800، *.2.3.

### الغرض

تقوم هذه الوظيفة في البداية بمعالجة الحالات الخاصة حيث تكون أي من أو كلا معاملات المدخلات خالية وتقوم بارجاع أكواد النتيجة لتمثل هذه الحالات. وتقوم بتقسيم كل من النسخ الى أجزاء متعددة من خلال فاصل النقطة. اذا كان آخر حرف من النسخة هو حرف الابدال *، فان الوظيفة تعتبر كل الأجزاء الغير موجودة من النسخة هي حرف الابدال، على سبيل المثال، *.2 تتطابق مع 2.1 أو *.2.3. وستقوم بعد ذلك بالبحث خلال كشف الأجزاء لكل من النسخ ومقارنتها معهم. وستقوم بعد ذلك بارجاع أكواد النتيجة بناء على ما اذا كانت القيم المتوقعة أقل من أو تساوي أو أكبر من القيمة الفعلية.

### الوظائف الرئيسية

*الجدول ٣٩. الوظائف الرئيسية التي تقوم باستدعاء versionCompare*

| الوظيفة الرئيسية، البرنامج النصي | الوصف |
|---|---|
| `cygwinVersion_compare.vbs` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية لنسخة cygwin. |
| `gskit7Version_compare.vbs` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية للنسخة 7 من gskit. |
| `gkit8Version_compare.vbs` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية للنسخة 8 من gskit. |

| | |
|---|---|
| internetExplorer.version_compare.vbs | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية لنسخة Internet Explorer. |
| os.servicePack_compare.vbs | لمقارنة القيم الفعلية والمتوقعة لخاصية نسخة نظام التشغيل للمتطلبات الرئيسية. |
| os.versionNumber_compare.vbs | لمقارنة القيم الفعلية والمتوقعة لخاصية نسخة نظام التشغيل للمتطلبات الرئيسية. |

**الصيغة**

```
versionCompare(ver1,ver2)
```

**معاملات المدخلات**

**String *ver1***

يتضمن القيمة المتوقعة لخاصية المتطلبات الرئيسية.

**String *ver2***

يتضمن النسخة الفعلية لخاصية المتطلبات الرئيسية.

**القيم الناتجة**

**Integer *0***

تقوم بارجاع كود النتيجة ٠ اذا كانت كلا معاملات المدخلات متساوية. الوظيفة الرئيسية تقوم بارجاع "PASS".

الحالة الخاصة: تقوم بارجاع كود النتيجة ٠ ويتم الخروج اذا كانت كلا معاملات مدخلات خالية.

**Integer *-1***

تقوم بارجاع كود النتيجة -١ اذا كان معامل المدخلات الأول أقل من معامل المدخلات الثاني. الوظيفة الرئيسية تقوم بارجاع "FAIL".

الحالة الخاصة: تقوم بارجاع كود النتيجة ١ ويتم الخروج اذا كان معامل المدخلات الأول خاليا.

**Integer *1***

تقوم بارجاع كود النتيجة ١ اذا كان معامل المدخلات الأول أكبر من معامل المدخلات الثاني. الوظيفة الرئيسية تقوم بارجاع "PASS".

الحالة الخاصة: تقوم بارجاع كود النتيجة ١ ويتم الخروج اذا كان معامل المدخلات الثاني خاليا.

# الملحق ط. الوظائف المعتادة لأنظمة UNIX

Prerequisite Scanner له مجموعة من الوظائف المعتادة في الملف lib/common_function.sh/ لتشغيل عمليات تحقق على الأنظمة على أساس UNIX.

*الجدول ٤٠. الوظائف في common_function.sh*

| الوظيفة | الوصف |
|---|---|
| "AddMG()" | يتم التحقق مما اذا كانت معاملات المدخلات موجودة بالميجابايت أو الجيجابايت ويتم اضافة المعاملات. |
| "changeMG()" | لتحويل معامل المدخلات الى الميجابايت أو الجيجابايت لمساحة القرص أو الذاكرة لخصائص المتطلبات الرئيسية. |
| "compare()" | للتحليل اللغوي لمعاملات المدخلات التي تمثل القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية ومقارنتها لتحديد ما اذا كانت القيمة الأولى (الحقيقية) أقل من القيمة الثانية (المتوقعة). |
| "cutdown()" | للتحليل اللغوي لمعاملات المدخلات التي تمثل القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية ومقارنتها لتحديد ما اذا كانت القيمة الأولى (الحقيقية) أقل من القيمة الثانية (المتوقعة). وبعد ذلك يتم طباعة الفرق بين القيمتين اذا كانت القيمة الأولى ليست أقل من القيمة الثانية. |
| "findOSInfo()" | لايجاد نسخة نظام التشغيل، ومستوى الاصدار والنسخة لنظام التشغيل، وبيانات تجهيز الجهاز الخاصة بالنظام . |
| "mes4path()" | لايجاد مساحة القرص الخالية لكل من أنظمة الملفات. |
| "mes4Path1()" | لايجاد مساحة القرص الخالية لكل من أنظمة الملفات الموجودة على نظام Solaris فقط. |
| "NFScheck()" | للتحقق من حالة NFS لما هو موجود على الأنظمة على أساس UNIX. |
| "telnetNFS()" | للتحقق من امكانية الاتصال مع عنوان IP لنظام الملفات الموجود على منفذ 2049 المفترض. |

## changeMG()

لتحويل معامل المدخلات الى الميجابايت أو الجيجابايت لمساحة القرص أو الذاكرة لخصائص المتطلبات الرئيسية.

### الغرض

تقوم هذه الوظيفة بالتحقق أولا من أن الوظيفة تقوم باستلام أحد معاملات المدخلات. اذا كانت تقوم باستلام أحد معاملات المدخلات، فانها تقوم بتحديد ما اذا كانت القيمة بالميجابايت أو الجيجابايت ثم تقوم بتحويل القيمة الى الجيجابايت أو الميجابايت على التوالي.

### الصيغة

```
changeMG val
```

**معاملات المدخلات**

**String *$val***

تتضمن القيمة لمساحة القرص أو الذاكرة بالميجابايت أو الجيجابايت.

**القيم الناتجة**

**Integer *1***

تقوم بارجاع ١ اذا كانت الوظيفة لا تقوم باستلام معامل المدخلات.

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

تقوم بارجاع القيمة بالميجابايت.

**String *printf "%.2fG%s",mm[1],mm[2];***

تقوم بارجاع القيمة بالجيجابايت.

---

## AddMG()

يتم التحقق مما اذا كانت معاملات المدخلات موجودة بالميجابايت أو الجيجابايت ويتم اضافة المعاملات.

**الغرض**

تقوم هذه الوظيفة بالتحقق أولا من أن الوظيفة تقوم باستلام معاملات المدخلات. اذا كانت تقوم باستلام أحد معاملات المدخلات، فانها تقوم بتحديد ما اذا كانت القيمة بالميجابايت أو الجيجابايت ثم باضافة القيم.

**الصيغة**

```
AddMG val1 val2
```

**معاملات المدخلات**

**String *$val1***

تتضمن القيمة لمساحة القرص أو الذاكرة بالميجابايت أو الجيجابايت ليتم اضافتها لمعامل المدخلات الآخر.

**String *$val2***

تتضمن القيمة لمساحة القرص أو الذاكرة بالميجابايت أو الجيجابايت ليتم اضافتها لمعامل المدخلات الآخر.

**القيم الناتجة**

**Integer *1***

لارجاع ١ اذا كانت الوظيفة لا تقوم باستلام اثنين من معاملات المدخلات.

**String *val***

لارجاع القيمة المضافة بالميجابايت أو الجيجابايت.

---

## compare()

للتحليل اللغوي لمعاملات المدخلات التي تمثل القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية ومقارنتها لتحديد ما اذا كانت القيمة الأولى (الحقيقية) أقل من القيمة الثانية (المتوقعة).

**الغرض**

تقوم هذه الوظيفة بالتحقق أولا من أن الوظيفة تقوم باستلام اثنين من معاملات المدخلات. اذا تم استلامها ولم تكن كلاها بالقيمة false، فانها تحدد ما اذا كانت القيم بالميجابايت أو بالجيجابايت ثم تقوم بمقارنتها مع اثنين من القيم للتحقق مما اذا كانت القيمة الأولى أقل من القيمة الثانية. واذا كانت كذلك، ستقوم بارجاع false؛ خلاف ذلك ستقوم بتمرير القيمة.

**الصيغة**

```
compare real expected
```

**معاملات المدخلات**

**String *$real***

يتضمن القيمة الفعلية لخاصية المتطلبات الرئيسية.

**String *$expected***

يتضمن القيمة المتوقعة لخاصية المتطلبات الرئيسية.

**القيم الناتجة**

**Integer *1***

لارجاع ١ اذا كانت الوظيفة لا تقوم باستلام اثنين من معاملات المدخلات.

**String *"FAIL|PASS"***

تقوم بارجاع مجموعة الحروف "FAIL" اذا كانت القيمة الفعلية أقل من القيمة المتوقعة؛ خلاف ذلك يتم ارجاع "PASS".

---

## cutdown()

للتحليل اللغوي لمعاملات المدخلات التي تمثل القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية ومقارنتها لتحديد ما اذا كانت القيمة الأولى (الحقيقية) أقل من القيمة الثانية (المتوقعة). وبعد ذلك يتم طباعة الفرق بين القيمتين اذا كانت القيمة الأولى ليست أقل من القيمة الثانية.

**الغرض**

تقوم هذه الوظيفة بالتحقق أولا من أن الوظيفة تقوم باستلام اثنين من معاملات المدخلات. اذا كانت تقوم باستلامها، فانها تحدد ما اذا كانت القيم بالميجابايت أم بالجيجابايت ثم تقوم بتحويلها الى الميجابايت اذا كانت بالجيجابايت. وستقوم بعد ذلك بمقارنة القيمتين للتحقق مما اذا كانت القيمة الأولى أقل من القيمة الثانية. واذا كانت كذلك، ستقوم بارجاع القيمة "٠ ميجابايت"؛ خلاف ذلك تقوم بارجاع الفارق بين القيمتين بالميجابايت.

**الصيغة**

```
cutdown real expected
```

**معاملات المدخلات**

**String *$real***

يتضمن القيمة الفعلية لخاصية المتطلبات الرئيسية.

**String *$expected***

يتضمن القيمة المتوقعة لخاصية المتطلبات الرئيسية.

**القيم الناتجة**

**Integer *1***

لارجاع ١ اذا كانت الوظيفة لا تقوم باستلام اثنين من معاملات المدخلات.

**String *"FAIL|PASS"***

تقوم بارجاع مجموعة الحروف "FAIL" اذا كانت القيمة الفعلية أقل من القيمة المتوقعة اذا لم تكن أي من القيم المتوقعة بالميجابايت أو الجيجابايت؛ خلاف ذلك يتم ارجاع "PASS".

**String *"OMB|Real-ExpectedMB"***

تقوم بارجاع مجموعة الحروف "OMB" اذا كانت القيمة الفعلية أقل من القيمة المتوقعة؛ خلاف ذلك تقوم بارجاع مجموعة حروف تمثل الفرق بين اثنين من القيم التي تم تحويلها بالميجابايت.

---

## mes4path()

لايجاد مساحة القرص الخالية لكل من أنظمة الملفات.

**الغرض**

هذه الوظيفة تأخذ مسار كأحد المدخلات، ويتم استدعاء الأمر **uname** لتحديد نظام التشغيل، ثم استدعاء الوظيفة NFScheck لتحديد ما اذا كان النظام يعمل وما يتم تشغيله. وستقوم بعد ذلك باستدعاء الأمر df لتحديد مساحة القرص الخالية لكل من البرامج على النظام . تقوم بارجاع قيمة مساحة القرص الخالية.

**الصيغة**

```
mes4Path path
```

**معاملات المدخلات**

**String *$path***

المسار الى النظام للتحقق من مساحة القرص المتاحة.

**القيم الناتجة**

**Integer *1***

لارجاع كود النتيجة ١ اذا كانت الوظيفة لا تقوم باستلام معامل المدخلات.

**Integer *2***

لارجاع كود النتيجة ٢ اذا كان معامل المدخلات ليس مسارا.

**String *$NF***

تقوم بارجاع مساحة القرص الخالية على كل ما يتم تركيبه.

**String *"$path Server NotAvailable Responding for $path"***

لارجاع رسالة لتحديد اتاحة وحدة الخدمة للمسار.

## mes4Path1()

لايجاد مساحة القرص الخالية لكل من أنظمة  الملفات الموجودة على نظام Solaris فقط.

**الغرض**

هذه الوظيفة تأخذ مسار كأحد المدخلات، ويتم استدعاء الأمر **uname** لتحديد أن نظام التشغيل هو Solaris.  وستقوم بعد ذلك باستدعاء الأمر df لتحديد مساحة القرص الخالية لكل من البرامج على النظام .  تقوم بارجاع قيمة مساحة القرص الخالية.

**الصيغة**

```
mes4Path1 path
```

**معاملات المدخلات**

**String *$path***

المسار الى النظام  للتحقق من مساحة القرص المتاحة.

**القيم الناتجة**

**Integer *1***

لارجاع كود النتيجة ١ اذا كانت الوظيفة لا تقوم باستلام معامل المدخلات.

**Integer *2***

لارجاع كود النتيجة ٢ اذا كان معامل المدخلات ليس مسارا.

**String *$NF***

تقوم بارجاع مساحة القرص الخالية على كل ما يتم تركيبه.

## findOSInfo()

لايجاد نسخة نظام التشغيل، ومستوى الاصدار والنسخة لنظام  التشغيل، وبيانات تجهيز الجهاز الخاصة بالنظام .

**الغرض**

تقوم هذه الوظيفة بتشغيل الأمر uname وتحليل المخرجات الخاصة بها لغويا لنسخة نظام التشغيل، مستوى ونسخة اصدرا نظام التشغيل، وبيانات تجهيز الجهاز للنظام .

**الصيغة**

```
findOSInfo
```

**معاملات المدخلات**

لا شيء

**القيم الناتجة**

**String *$oo***

المخرجات من **uname** بدون معلومات النظام  الأساسية.

**String *$kk***

نسخة نظام التشغيل

**String *$hh***

تجهيز الجهاز يتم تمثيله بالقيمة I للجهاز i386 أو Z للجهاز s390.

**String *$rr***

مستوى الاصدار لنظام التشغيل

**String *$vv***

نسخة مستوى الاصدار لنظام التشغيل

---

## telnetNFS()

للتحقق من امكانية الاتصال مع عنوان IP لنظام الملفات الموجود على منفذ 2049 المفترض.

### الغرض

تقوم هذه الوظيفة باتخاذ عنوان IP كمدخلات وتقوم باستدعاء الأمر **telnet** لاختبار ما اذا كانت الوصلة عن بعد ناجحة على منفذ اتصالات 2049 المفترض. وسيتم محاولة الاتصال عن بعد ١٠ مرات. اذا حدث فشل بأي من أوامر **telnet**، ستقوم الوظيفة بارجاع القيمة "FALSE"؛ وتقوم بارجاع القيمة "PASS".

### الصيغة

```
telnetNFS ipaddr
```

### معاملات المدخلات

**String *$ipaddr***

عنوان IP للتحقق مما اذا كان يمكن اجراء اتصال لاسلكي.

### القيم الناتجة

**String *"FALSE|TRUE"***

تقوم بارجاع نتائج عملية تحقق Telnet. وتقوم بارجاع "TRUE" اذا كانت عملية التحقق ناجحة؛ خلاف ذلك تقوم بارجاع "FALSE".

---

## NFScheck()

للتحقق من حالة NFS لما هو موجود على الأنظمة على أساس UNIX.

### الغرض

هذه الوظيفة تأخذ مسار كأحد المدخلات، ويتم استدعاء الأمر mount لاحضار كشف أنظمة الملفات الموجودة. وتقوم باستدعاء الأمر **uname** لتحديد نظام التشغيل. وستقوم بعد ذلك باستدعاء الأمر **ping** لاختبار الاتصال لكل من الأنظمة الموجودة وما اذا كان يمكن اختبارها، وستقوم بعد ذلك باستدعاء الوظيفة **telnetNFS** للتحقق مما اذا كان يمكن اقامة اتصال عن بعد. اذا حدث فشل في تصرفات اختبار الاتصال أو الاتصال اللاسلكي، فستقوم الوظيفة بارجاع القيمة "FALSE"؛ خلاف ذلك تقوم بارجاع القيمة "PASS".

**الصيغة**

`NFScheck path`

**معاملات المدخلات**

**String *$path***

تتخذ مسار صحيح للقاموس كمدخلات له.

**القيم الناتجة**

**Boolean value *TRUE* or *FALSE***

تقوم بارجاع TRUE اذا كان اختبار NFS ناجح، أي اذا قام باختبار الاتصال لعنوان IP الخاص به بنجاح أو ما اذا كان يمكن استخدام الاتصالات اللاسلكية للاتصال مع عنوان IP الخاص بكل من أنظمة الملفات؛ خلاف ذلك تقوم بارجاع FALSE.

**مثال**

هذا مثال للاستخدام من الوظيفة **`mes4Path()`**:

```
# 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
      ...
```

# الملحق ي. الوظائف الأخرى لأنظمة UNIX

Prerequisite Scanner له مجموعة من الوظائف المعتادة في ملفات مختلفة.

يوضح الجدول ٤١ مجموعة الوظائف في ملفات متعددة.

*الجدول ٤١. الوظائف المعتادة في ملفات متعددة*

| الوظيفة | الوصف |
|---|---|
| "formatSizeDisplay()" | يقوم بأخذ معامل المدخلات والحاق أو تقليل جزء الكسر من معامل المدخلات الى اثنين النقاط العشرية، على سبيل المثال، ١٢٣ ميجابايت الى ١٢٣,٠٠ ميجابايت أو ١٢,١٢٣ ميجابايت الى ١٢,١٢ ميجابايت. |
| "versionCompare()" | للتحليل اللغوي لمعاملات المدخلات التي تمثل القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية ومقارنة كل من الأجزاء لتحديد ما اذا كانت خاصية المتطلبات الرئيسية ستجتاز عملية التحقق من المتطلبات الرئيسية. |

يوضح الجدول ٤٢ مجموعة الوظائف في ملف `UNIX-Linux/TAD722_impl.sh` لتشغيل عمليات التحقق على Tivoli License Compliance Manager و Tivoli Asset Discovery for Distributed.

*الجدول ٤٢. الوظائف المعتادة في TAD722_impl.sh*

| الوظيفة | الوصف |
|---|---|
| "checkSunOS()" | للتحقق مما اذا كانت نسخة نظام التشغيل Solaris هي SPARC أو X86. |
| "checkHpux()" | للتحقق مما اذا كانت نسخة نظام التشغيل HP-UX هي IA64 أو PARISC. |
| "checkLinux()" | للتحقق مما اذا كانت نسخة نظام التشغيل Linux هي الى بيئات تشغيل System p® أو System z أو x86. |
| "getSystemId()" | يتم استدعاء وظائف نظام تشغيل مختلف للتحقق من بيئات التشغيل لنظام التشغيل المتعلق. |
| "getValue()" | لاحضار القيمة لمفتاح في ملف محدد اذا كان الملف موجود. |
| "setValue()" | لتحديد القيمة للمفتاح في الملف المحدد اذا كانت خاصية المتطلبات الرئيسية موجودة. |
| "copyValue()" | يتم احضار وتحديد القيمة لخاصية (key) بناء على البرنامج ونظام التشغيل. |
| "parseDirParameter()" | للتحليل اللغوي للمعامل من كشف المعاملات لشارة التعليم p- للماسح الضوئي ووضع القيمة الخاصة به في الكشف. |
| "getClosestExistingParentDir()" | لاحضار أقرب دليل رئيسي أو الدليل الرئيسي نفسه. |
| "printDirSize()" | للتحقق من حالة NFS لنظام الملفات الموجود ثم احضار مساحة القرص الخاصة بنظام الملفات أو الدليل الرئيسي لها. |

## formatSizeDisplay()

يقوم بأخذ معامل المدخلات والحاق أو تقليل جزء الكسر من معامل المدخلات الى اثنين النقاط العشرية، على سبيل المثال، ١٢٣ ميجابايت الى ١٢٣,٠٠ ميجابايت أو ١٢,١٢٣ ميجابايت الى ١٢,١٢ ميجابايت.

### الغرض

تقوم هذه الوظيفة باحصاء عدد الحروف في معامل المدخلات، ويتم التحقق مما اذا كان عبارة عن رقم أم مجموعة حروف، ويتم فصل الجزء الخاص بالمدخلات وأجزاء الكسر بالكامل. وبناء على الجزء الكسري، سيتم الحاق أو قطعه الى اثنين من الأماكن العشرية. تقوم بارجاع النتيجة.

### البرامج النصية الرئيسية

البرامج النصية التالية تتضمن الوظيفة:

- `./Unix-Linux/common.sh`
- `LCM_TAD_common.sh`

### الصيغة

```
formatSizeDisplay val
```

### معاملات المدخلات

**Integer *$val***

يتم تقريب القيمة الى رقمين عشريين.

### القيم الناتجة

**Integer *val***

يتم ارجاع القيمة مقربة الى رقمين عشريين.

## versionCompare()

للتحليل اللغوي لمعاملات المدخلات التي تمثل القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية ومقارنة كل جزء من النسخة لتحديد ما اذا كانت القيمة الأولى (الفعلية) أكبر من القيمة الثانية (المتوقعة).

### الغرض

تقوم هذه الوظيفة بالتحقق أولا من أن الوظيفة تقوم باستلام اثنين من النسخ كمعاملات مدخلات. وهي تقوم باستخدام awk للتحليل اللغوي وتقسيم كل من النسخ الى أجزاء، حيث "." هو الفاصل لتقسيم القيمة الى أجزاء. سيقوم بعد ذلك بتنفيذ عملية لمقارنة كل من أجزاء النسخة الأولى مع نفس الجزء من النسخة الثانية ومشاهدة ما اذا كانت متساوية أم لا.

### الوظائف الرئيسية

*الجدول ٤٣. الوظائف الرئيسية التي تقوم باستدعاء versionCompare*

| الوظيفة الرئيسية، البرنامج النصي | الوصف |
|---|---|
| `db2.home_compare.sh` | لمقارنة القيمة الفعلية والمتوقعة لمساحة القرص للخاصية DB2 HOME. |

| | |
|---|---|
| `oracle.Client_compare.sh` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية لبرنامج Oracle. |
| `os.locale_compare.sh` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية لنظام التشغيل المحلي. |
| `os.MozillaVersion_compare.sh` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية لبرنامج Mozilla Firefox. |
| `os.package.perl_compare.sh` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية لمجموعة برامج Perl. تقوم باستدعاء نفسها. |
| `os.RAMSize_compare.sh` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية الى RAM. |
| `os.space_compare.sh` | لمقارنة القيم الفعلية والمتوقعة لخاصية المتطلبات الرئيسية لمساحة القرص المتاحة. |
| `OS_Version_compare.sh` | لمقارنة القيم الفعلية والمتوقعة لخاصية نسخة نظام التشغيل للمتطلبات الرئيسية. |

**الصيغة**

```
versionCompare real expected
```

**معاملات المدخلات**

**String *$real***

يتضمن القيمة الفعلية لخاصية المتطلبات الرئيسية.

**String *$expected***

يتضمن القيمة المتوقعة لخاصية المتطلبات الرئيسية.

**القيم الناتجة**

**Integer *0***

تقوم بارجاع كود النتيجة ٠ اذا كانت القيم الفعلية والمتوقعة متساوية. الوظيفة الرئيسية تقوم بارجاع "PASS".

الحالة الخاصة: تقوم بارجاع كود النتيجة ٠ ويتم الخروج اذا كانت الوظيفة تقوم باستلام معاملات مدخلات خالية.

**Integer *-1***

تقوم بارجاع كود النتيجة -١ اذا كانت القيمة الحقيقية أقل من القيمة المتوقعة. الوظيفة الرئيسية تقوم بارجاع "FAIL".

تقوم بارجاع كود النتيجة -١ ويتم الخروج اذا كانت الوظيفة تقوم باستلام معامل المدخلات الثاني خاليا.

**Integer *1***

تقوم بارجاع كود النتيجة ١ اذا كانت القيمة الحقيقية أكبر من القيمة المتوقعة. الوظيفة الرئيسية تقوم بارجاع "PASS".

تقوم بارجاع كود النتيجة ١ ويتم الخروج اذا كانت الوظيفة تقوم باستلام معامل المدخلات الثاني خاليا.

## checkHpux()

للتحقق مما اذا كانت نسخة نظام التشغيل HP-UX هي IA64 أو PARISC.

**الغرض**

تقوم هذه الوظيفة باستخدام شارة التعليم m- للأمر **uname** لتحديد ما اذا كان نظام التشغيل HP-UX لبيئات تشغيل IA64 أو PARISC.

**الصيغة**

```
checkHpux
```

**القيم الناتجة**

**String *HPUXIA64|HPUXPARISC***

تقوم بارجاع "HPUXIA64" اذا كانت شارة التعليم m- هي "ia64"؛ خلاف ذلك تقوم بارجاع "HPUXPARISC".

## checkLinux()

للتحقق مما اذا كانت نسخة نظام التشغيل Linux هي الى بيئات تشغيل System p أو System z أو x86.

**الغرض**

تقوم هذه الوظيفة باستخدام شارة التعليم m- للأمر **uname** لتحديد ما اذا كان نظام التشغيل Linux هو لنظام System p أو System z أو x86.

**الصيغة**

```
checkLinux
```

**معاملات المدخلات**

**القيم الناتجة**

**String *LINUXPSERIES|LINUXZSERIES|LINUXX86***

تقوم بارجاع "LINUXPSERIES" اذا كانت شارة التعليم m- هي "ppc64" أم "ppc". وتقوم بارجاع "LINUXZSERIES" اذا كانت القيمة هي "s390x" أو "s390"؛ خلاف ذلك، ستقوم بارجاع "LINUXX86".

## checkSunOS()

للتحقق مما اذا كانت نسخة نظام التشغيل Solaris هي SPARC أو X86.

**الغرض**

تقوم هذه الوظيفة باستخدام شارة التعليم p– للأمر **uname** لتحديد ما اذا كان نظام التشغيل Solaris لبيئات تشغيل SPARC أو X86.

**الصيغة**

```
checkSunOS
```

**معاملات المدخلات**

**القيم الناتجة**

**String *SOLARISSPARC|SOLARISX86***

تقوم بارجاع "SOLARISSPARC" اذا كانت شارة التعليم p- هي "sparc"؛ خلاف ذلك تقوم بارجاع "SOLARISX86".

---

## getValue()

لاحضار القيمة لمفتاح في ملف محدد اذا كان الملف موجود.

**الغرض**

**الصيغة**

```
getValue key file
```

**معاملات المدخلات**

**String *$key***

يتضمن المفتاح الذي سيتم تحديده.

**String *$file***

يتضمن اسم الملف الذي يتضمن المفتاح.

---

## setValue()

لتحديد القيمة للمفتاح في الملف المحدد اذا كانت خاصية المتطلبات الرئيسية موجودة.

**الصيغة**

```
setValue key value file
```

**معاملات المدخلات**

**String *$key***

يتضمن خاصية المتطلبات الرئيسية التي سيتم تحديدها.

**String *$value***

يتضمن القيمة لخاصية المتطلبات الرئيسية.

**String *$file***

يتضمن اسم الملف الذي سيتضمن خاصية المتطلبات الرئيسية.

---

## copyValue()

يتم احضار وتحديد القيمة لخاصية (key) بناء على البرنامج ونظام التشغيل.

**الغرض**

تقوم هذه الوظيفة باستدعاء **`getValue()`** لاحضار القيمة لخاصية المتطلبات الرئيسية المحددة للبرنامج ونظام التشغيل. وستقوم بعد ذلك باستدعاء الوظيفة **`setValue()`** لتحديد القيمة لخاصية المتطلبات الرئيسية في الملف Prerequisite Scanner.

**الصيغة**

```
copyValue key file
```

**معاملات المدخلات**

**String *$key***

يتضمن المفتاح الذي سيتم احضاره وتحديده.

**String *$file***

يتضمن اسم الملف الذي يتضمن المفتاح.

**القيم الناتجة**

---

## getSystemId()

يتم استدعاء وظائف نظام تشغيل مختلف للتحقق من بيئات التشغيل لنظام التشغيل المتعلق.

**الغرض**

تقوم هذه الوظيفة باستدعاء وظائف مختلفة لنظام التشغيل لتحديد بيئات التشغيل لنظام التشغيل المتعلق.

**الصيغة**

```
getSystemId
```

**معاملات المدخلات**

**القيم الناتجة**

**String *AIX|Linux***

يتم ارجاع "AIX" أو "Linux" اذا كان البرنامج هو Tivoli License Compliance Manager ونظام التشغيل يكون AIX أو Linux أو "AIX" اذا كان البرنامج هو Tivoli Asset Discovery for Distributed ونظام التشغيل هو AIX.

---

## getClosestExistingParentDir()

لاحضار أقرب دليل رئيسي أو الدليل الرئيسي نفسه.

**الغرض**

**الصيغة**

```
getClosestExistingParentDir dirpath
```

**معاملات المدخلات**

**String *$dirpath***

يتضمن المسار لاحضار الدليل الرئيسي الخاص به أو الدليل بنفسه.

**القيم الناتجة**

**String *dirpath***

لارجاع الدليل الرئيسي أو الدليل نفسه.

---

## parseDirParameter()

للتحليل اللغوي للمعامل من كشف المعاملات لشارة التعليم p– للماسح الضوئي ووضع القيمة الخاصة به في الكشف.

**الغرض**

**الصيغة**

**معاملات المدخلات**

**String**

**القيم الناتجة**

---

## printDirSize()

للتحقق من حالة NFS لنظام  الملفات الموجود ثم احضار مساحة القرص الخاصة بنظام الملفات أو الدليل الرئيسي لها.

**الغرض**

تقوم هذه الوظيفة أولا باستدعاء **`NFScheck`** لتحديد حالة NFS للدليل.  واذا كانت الحالة هي true، فستقوم باستدعاء **`getClosestExistingParentDir`** لارجاع الدليل أو الدليل الرئيسي له، ثم استخدام الأمر **`df`** لاحضار مقدار مساحة القرص الخالية.  وستقوم في النهاية باستدعاء الوظيفة **`formatSizeDisplay`** لتقريب القيمة الى العلامة العشرية.

**الصيغة**

```
printDirSize dirpath
```

**معاملات المدخلات**

**String *$dirpath***

تتضمن المسار الخاص بالدليل الذي سيتم احضار مساحة القرص الخالية له.

**القيم الناتجة**

**Integer *dsize***

تقوم بارجاع مقدار مساحة القرص الخالية الى رقمين عشريين.

**String *"NFS_NOT_AVAILABLE"***

تقوم بارجاع أن نظام الملفات الموجود غير متاح.

# الملحق ك. وظائف تسجيل الوظيفة لأنظمة UNIX

Prerequisite Scannerله مجموعة من وظائف التسجيل العامة في ملف /lib/common_function.sh لتصحيح أخطاء الكتابة وبيانات التتبع في ملفات سجلات الأخطاء.

يوضح الجدول ٤٤ وظائف التسجيل.

*الجدول ٤٤. وظائف تسجيل الوظيفة لأنظمة UNIX*

| الوظيفة | الوصف | معاملات المدخلات |
|---|---|---|
| wrlTrace *log_str1* *log_str2* | تقوم بكتابة *log_str1* و *log_str2* في ملف التتبع، مع خاتم الوقت | *log_str1* و *log_str2*، مجموعات الحروف الخاصة بالتتبع التي تمثل التصرف وأداة التجميع التي يتم تنفيذها وما الذي سيتم تسجيله في ملف التتبع. على سبيل المثال:<br>**`wrlTrace Starting os.lib`**<br>**`wrlTrace Executing os.lib`**<br>`wrlDebug Starting os.lib`<br>`wrlDebug Expected libXp `<br>`ss=`./os.lib libXp libXp`<br>**`wrlTrace Finished os.lib`**<br>`echo "os.lib.libXp=$ss"`<br>`wrlDebug Finished os.lib`<br>`wrlDebug OutPutValueIs $ss`<br>**`wrlTrace Done os.lib`** |
| wrlTraceFuncStart *fn_name* | لتمرير الوظيفة *fn_name* الى wrlTrace() | *fn_name*، يقوم بتتبع مجموعات الحروف التي تمثل الوظيفة التي تم استدعائها الآن. على سبيل المثال:<br>`wrlTraceFuncStart "$1"` |
| wrlTraceFuncExit *fn_name* | لتمرير الوظيفة *fn_name* الى wrlTrace() | *fn_name*، يقوم بتتبع مجموعات الحروف التي تمثل الوظيفة التي تم اتمامها الآن. على سبيل المثال:<br>`wrlTraceFuncExit "$1"` |
| wrlDebug *log_str1* *log_str2* | لتمرير عبارات *log_str1* و *log_str2* الى wrlDebugGeneric() | *log_str1* و *log_str2*، تقوم بتصحيح أخطاء مجموعات الحروف التي تمثل التصرف وأداة التجميع التي يتم تنفيذها وما الذي سيتم تسجيله في ملف تسجيل الأخطاء. على سبيل المثال:<br>`wrlTrace Starting os.lib`<br>`wrlTrace Executing os.lib`<br>**`wrlDebug Starting os.lib`**<br>**`wrlDebug Expected libXp `**<br>`ss=`./os.lib libXp libXp`<br>`wrlTrace Finished os.lib`<br>`echo "os.lib.libXp=$ss"`<br>**`wrlDebug Finished os.lib`**<br>**`wrlDebug OutPutValueIs $ss`**<br>`wrlTrace Done os.lib` |

| | | |
|---|---|---|
| wrlDebugFuncStart *fn_name* | لتمرير الوظيفة *fn_name* الى wrlDebug() | *fn_name*، يقوم بتصحيح أخطاء مجموعات الحروف التي تمثل الوظيفة التي تم استدعائها الآن. على سبيل المثال:<br>`` `wrlDebugFuncStart "$1"` `` |
| wrlDebugFuncExit *fn_name* | لتمرير الوظيفة *fn_name* الى wrlDebug() | *fn_name*، يقوم بتصحيح مجموعات الحروف التي تمثل الوظيفة التي تم اتمامها الآن. على سبيل المثال:<br>`` `wrlDebugFuncExit "$1"` `` |
| wrlDebugFuncReturn *result_value* | تقوم بكتابة نتيجة *result_value* المرتجعة للوظيفة الى ملف سجل الأخطاء | *result_value*، يقوم بتصحيح أخطاء مجموعة الحروف التي تمثل القيمة المرتجعة من الوظيفة. على سبيل المثال:<br>`` `wrlDebugFuncReturn "$versionCompare"` `` |
| wrlDebugFuncParam *param1 param2* | لتمرير المعاملات *param1* و *param2* الى wrlDebugFunc() | *param1* و *param2*، لتصحيح أخطاء مجموعات الحروف التي تمثل عناوين قسم الذي تم تحليلها لغويا، والمؤهلات التي تم تحليلها لغويا أو متغيرات المدخلات التي تم تحليلها لغويا للوظائف التي تم استدعاءها. على سبيل المثال:<br>`` `wrlDebugFuncParam "OSArch" "$3"` `` |
| wrlDebugGeneric formatspec *log_str1 log_str2* | تقوم بكتابة مجموعات الحروف *log_str1* و *log_str2* في ملف تصحيح الأخطاء، ويتم تنسيقها من خلال *formatspec* | • *log_str1* و *log_str2*، التي تمثل البيانات المحددة التي سيتم تسجيلها في أحد سطور ملف تصحيح الأخطاء<br>• *formatspec*، المتغير الذي سيتم وضعه بعد خاتم الوقت ولكن قبل المتغير اليسار من مجموعات الحروف الخاصة بالسجل وتحكم السطر الجديد<br><br>على سبيل المثال:<br>`` `wrlDebugGeneric "" "$1" "$2"` `` |
| wrlDebugFunc *str* | تقوم بتمرير حرف tab ومعامل المدخلات *str* الى wrlDebugGeneric() | *str*، مجموعة حروف تمثل البيانات التي سيتم تسجيلها، والتي تمثل، حالة التحقق أو التصرف الذي سيتم تنفيذه. على سبيل المثال:<br>`` `wrlDebugFunc "Reading config file and parsing using parse array..." ` `` |
| wrlLogFuncStart *str* | تقوم بتمرير معامل المدخلات *str* الى wrlTraceFuncStart() wrlDebugFuncStart() | *str*، مجموعة حروف تمثل البيانات التي سيتم تسجيلها، والتي تمثل، اسم الوظيفة التي يتم استدعاءها. على سبيل المثال:<br>`` `wrlLogFuncStart "main()"` `` |

| | | |
|---|---|---|
| wrlLogFuncExit *str* | تقوم بتمرير معامل المدخلات *str* الى<br>wrlTraceFuncExit()<br>wrlDebugFuncExit() | *str*، مجموعة حروف تمثل البيانات التي سيتم تسجيلها، والتي تمثل، اسم الوظيفة التي يتم تنفيذها. على سبيل المثال:<br>`wrlLogFuncExit "main()"` |

# الملاحظات

تم تعديل هذه المعلومات للمنتجات والخدمات التي يتم تقديمها في الولايات المتحدة الأمريكية. قد لا تقوم شركة IBM بتقديم المنتجات أو الخدمات أو الخصائص المذكورة بهذه الوثيقة في البلاد الأخرى. يمكنكم الاتصال بممثل شركة IBM المحلي للحصول على معلومات عن المنتجات والخدمات المتاحة حاليا في بلدكم. ولا يعد أي ذكر لأي من منتجات أو برامج أو خدمات شركة IBM اشارة صريحة أو ضمنية الى أنه يجب استخدام منتجات أو برامج أو خدمات IBM فقط. ويمكن استخدام أي منتجات أو برامج أو خدمات مكافئة لها وظيفيا بحيث لا تتعارض مع أي من حقوق الملكية الفكرية الخاصة بشركة IBM بدلا منها. لكن، يقع على عاتق المستخدم مسئولية تقييم والتحقق من العمليات الخاصة بأي من المنتجات أو البرامج أو الخدمات الغير خاصة بشركة IBM.

قد يكون لشركة IBM براءات اختراع أو براءات اختراع مؤجلة لتطبيقات تغطي أحد الموضوعات التي يتم تناولها في هذه الوثيقة. لا يمنحك اقتناء هذه الوثيقة أي ترخيص لبراءات الاختراع هذه. يمكنك ارسال أي استفسارات عن التراخيص كتابة الى:

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

بالنسبة للاستفسارات الخاصة بالتراخيص التي تتعلق بالمعلومات ثنائية البايت (DBCS)، ا تصل بقسم IBM Intellectual Property Department في بلدك أو أرسل الاستفسارات كتابة الى:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

**لا تنطبق الفقرة التالية على المملكة المتحدة أو أي بلد أخرى حيث تتعارض هذه الشروط مع قوانينها المحلية:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

لا تسمح بعض الولايات بالتنازل الصريح أو الضمني عن الضمانات في بعض المعاملات، لذلك، قد لا تنطبق هذه العبارة عليك.

قد تتضمن هذه المعلومات أخطاء فنية أو مطبعية. يتم اجراء تغييرات دورية على المعلومات التي توجد بهذا الدليل؛ سيتم تضمين هذه التغييرات في الاصدارات الجديدة لهذه المطبوعات الفنية. يمكن أن تقوم شركة IBM باجراء تحسينات و/أو تغييرات على المنتج (المنتجات) و/أو البرنامج (البرامج) الموضحين بهذه المطبوعات الفنية في أي وقت بدون اشعار سابق.

أية اشارة في هذه المعلومات الى مواقع خلاف مواقع شركة IBM للعلم فقط ولا يمكن أن تشير بأي حال من الأحوال الى الاقرار بمواقع الانترنت هذه. ولا تعد المواد الخاصة بمواقع الانترنت هذه جزء من المواد الخاصة بمنتج IBM هذا وتقع مخاطر استخدام مواقع الانترنت هذه على مسئوليتك.

يمكن أن تقوم شركة IBM باستخدام أو توزيع أي من المعلومات التي تقوم بادخالها بأي طريقة تراها مناسبة دون أن يترتب على ذلك أي التزام تجاهك.

بالنسبة للأشخاص المرخص لهم بهذا البرنامج والذين يريدون الحصول على معلومات عنه بغرض: (i) تبادل المعلومات بين برامج تم تكوينها بطريقة مستقلة وبرامج أخرى (بما في ذلك هذا البرنامج) و (ii) الاستخدام المتبادل للمعلومات التي تم تبادلها، فانه يمكنهم الاتصال من خلال:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

يمكن اتاحة هذه المعلومات متاحة بموجب الشروط والأحكام المناسبة، بما في ذلك في بعض الحالات، دفع رسوم.

يتم تقديم البرنامج المرخص الموضح بهذه الوثيقة وكل المواد المرخصة المتاحة له بواسطة شركة IBM طبقا لشروط اتفاقية IBM Customer Agreement أو IBM International Program License Agreement أو أي اتفاقية مماثلة يتم عقدها بيننا.

تم تحديد أية بيانات أداء موجودة هنا في بيئة تشغيل يتم التحكم بها. لذلك، قد تختلف النتائج التي يتم الحصول عليها في بيئات تشغيل أخرى بشكل كبير. قد يتم اجراء بعض القياسات بأنظمة في مستوى التطوير ولا يوجد أي ضمان أن هذه القياسات ستكون متماثلة بالأنظمة المتاحة بصفة عامة. بالاضافة الى ذلك، سيتم تقدير بعض القياسات من خلال الاستنتاجات. ولكن قد تختلف النتائج الفعلية. يجب أن يقوم مستخدمي هذه الوثيقة بالتحقق من البيانات القابلة للتطبيق في بيئة التشغيل الخاصة بهم.

تم الحصول على معلومات عن المنتجات خلاف منتجات IBM من موردي هذه المنتجات أو التصريحات التي تم نشرها أو أي مصادر عامة أخرى متاحة. لم تقم IBM باختبار هذه المنتجات ولا يمكنها تأكيد دقة الأداء أو التوافق أو أي ادعاءات أخرى تتعلق بمنتجات خلاف منتجات IBM. يجب ارسال أية أسئلة تتعلق بامكانيات المنتجات خلاف منتجات IBM الى موردي هذه المنتجات.

تكون كل العبارات المتعلقة بالاتجاهات والنوايا المستقبلية لشركة IBM عرضة للتغيير أو الالغاء بدون أي اشعار، وهي تمثل الأهداف فقط.

تحتوي هذه المعلومات على أمثلة للبيانات والتقارير المستخدمة في عمليات الأعمال اليومية. لتوضيح هذه الأمثلة بقدر الامكان، تتضمن الأمثلة أسماء أشخاص وشركات وعلامات تجارية ومنتجات. تعد كل هذه الأسماء أسماء وهمية وأي تشابه في الأسماء والعناوين المستخدمة بواسطة مؤسسة الأعمال الفعلية يعد من قبيل الصدفة.

تراخيص حقوق النشر:

تتضمن هذه المعلومات عينات لبرامج التطبيق بلغة المصدر، والتي توضح تقنيات البرمجة ببيئات التشغيل المختلفة. يمكنك نسخ وتعديل وتوزيع عينات البرامج هذه بأية طريقة بدون دفع رسوم لشركة IBM، وذلك لأغراض التطوير أو الاستخدام أو التسويق أو التوزيع لبرامج التطبيقات بما يتفق مع واجهة تعامل البرمجة للتطبيقات لبيئات التشغيل التي تم كتابة عينات البرامج لها. لم يتم اختبار هذه الأمثلة تماما في ظل كل الظروف. وعلى ذلك فان شركة IBM، لا يمكنها ضمان أو الاشارة ضمنيا الى مدى مصداقية أو امكانيات أو وظيفية هذه البرامج. يمكنك نسخ وتعديل وتوزيع عينات البرامج هذه بأية طريقة بدون دفع أي رسوم لشركة IBM لأغراض التطوير أو الاستخدام أو التسويق أو التوزيع لبرامج التطبيقات بما يتفق مع واجهات تعامل برمجة تطبيقات شركة IBM.

اذا كان يتم عرض هذه المعلومات من خلال الشاشة، قد لا تظهر الصور والرسوم التوضيحية الملونة.

**العلامات التجارية**

تعد IBM وشعار IBM و ibm.com® علامات تجارية أو علامات تجارية مسجلة لشركة International Business Machines Corp.، مسجلة بجميع أنحاء العالم. قد تكون أسماء المنتجات والخدمات الأخرى علامات تجارية لشركة IBM أو شركات أخرى. يوجد الكشف الحالي للعلامات التجارية لشركة IBM على الانترنت "Copyright and trademark information" على www.ibm.com/legal/copytrade.shtml.

تعد Adobe و Acrobat و PostScript وكل العلامات التجارية على أساس Adobe علامات تجارية أو علامات تجارية مسجلة لشركة Adobe Systems Incorporated في الولايات المتحدة أو البلاد الأخرى أو كلاهما.

تعد Cell Broadband Engine و Cell/B.E. علامات تجارية لشركة Sony Computer Entertainment, Inc.، في الولايات المتحدة أو البلاد الأخرى أو كلاهما ويتم استخدامها وفقا للتراخيص الخاصة بها.

تعد Intel وشعار Intel و Intel Inside وشعار Intel Inside و Intel Centrino وشعار Intel Centrino و Celeron و Intel Xeon و Intel SpeedStep و Itanium و Pentium علامات تجارية أو علامات تجارية مسجلة لشركة Intel Corporation أو الشركات التابعة لها في الولايات المتحدة والبلاد الأخرى.

تعد IT Infrastructure Library علامة تجارية مسجلة لوكالة Central Computer and Telecommunications Agency التي تعد جزءا الآن من Office of Government Commerce.

تعد ITIL علامة تجارية مسجلة وعلامة تجارية مسجلة لمجموعة Office of Government Commerce، ومسجلة في U.S. Patent and Trademark Office.

تعد Linux علامة تجارية لشركة Linus Torvalds في الولايات المتحدة أو البلاد الأخرى أو كلاهما.

تعد Microsoft و Windows و Windows NT وشعار Windows علامات تجارية لشركة Microsoft Corporation في الولايات المتحدة أو البلاد الأخرى أو كلاهما.

تعد UNIX علامة تجارية مسجلة لشركة Open Group في الولايات المتحدة والبلاد الأخرى.

تعد Java وكل العلامات التجارية والشعارات المبنية على أساس Java علامات تجارية أو علامات تجارية مسجلة لشركة Sun Microsystems, Inc. في الولايات المتحدة أو البلاد الأخرى أو كلاهما.

قد تكون أسماء الشركات أو المنتجات أو الخدمات الأخرى علامات تجارية أو علامات خدمة لآخرين.

# معلومات الدعم والمعلومات المرجعية

اذا كان لديك مشكلة في برامج IBM الخاصة بك، ستكون في حاجة الى حلها سريعا. ولذلك تقدم IBM طرق مختلفة للحصول على الدعم الذي تريده، مثل الحصول عليه من خلال الانترنت أو IBM Support Assistant. يمكنك أيضا تقديم معلومات مرجعية أو احالة طلبات المنتج لاجراء تحسينات عليه.

**الاتصال من خلال الانترنت**

تحتوي المواقع التالية على معلومات عن اتباع الخطأ وتصحيحه:

- اذهب الى صفحة IBM Prerequisite Scanner على IBM Support Portal.
- اذهب الى موضوعات Prerequisite Scanner على Service Management Connect. لا تتردد في المساهمة في هذه الموضوعات.

استخدم المواقع التالية لتقديم معلومات مرجعية واحالة الطلبات أو اجراء مناقشة عن Prerequisite Scanner:

- اذهب الى موضوعات Prerequisite Scanner في Prerequisite Scanner at Service Management Connect ولا تتردد في المساهمة في هذه الموضوعات.
- استخدم Integrated Service Management Message Board في Service Management Connect.
- قم باحالة أو مراجعة التحسينات في طلب المنتج الى Prerequisite Scanner على Tivoli RFE Community.

**IBM Support Assistant**

يعد IBM Support Assistant (ISA) نطاق عمل مجاني لخدمة البرامج المحلية يساعدك على حل الأسئلة والمشاكل الخاصة بك في منتجات برامج IBM. حيث يقدم ISA امكانية توصل سريعة للمعلومات المتعلقة بالدعم وأدوات الخدمة لتحديد المشكلة. لتركيب برامج ISA، اذهب الى http://www.ibm.com/software/support/isa.

Printed in USA