|  |
| --- |
| The MITRE Corporation |
| The OVAL® Language Linux Component Model Specification |
| Version 5.10.1 |
|  |
| **Danny Haynes, Stelios Melachrinoudis** |
| **7/6/2012** |

|  |
| --- |
| The Open Vulnerability and Assessment Language (OVAL®) is an international, information security, community standard, being developed in collaboration with any and all interested parties, to promote open and publicly available security content, and to standardize the transfer of this information across the entire spectrum of security tools and services. By standardizing the three main steps of the assessment process: representing configuration information of systems for testing; analyzing the system for the presence of the specified machine state; and reporting the results of the assessment, the OVAL Language provides a common and structured format that facilitates collaboration and information sharing among the information security community as well as interoperability among tools. This document defines the Linux platform-specific data model for the OVAL Language. |

# Acknowledgements

# Trademark Information

OVAL and the OVAL logo are registered trademarks of The MITRE Corporation. All other trademarks are the property of their respective owners.

# Terms of Use

MITRE MAKES OVAL AVAILABLE ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE MITRE CORPORATION, ITS BOARD OF TRUSTEES, OFFICERS, AGENTS, AND EMPLOYEES, DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.[[1]](#footnote-1)

# Feedback

The MITRE Corporation welcomes any feedback regarding the OVAL Language Linux Component Model Specification. Please send any comments, questions, or suggestions to the public OVAL Developer's Forum at [oval-developer-list@lists.mitre.org](mailto:oval-developer-list@lists.mitre.org) or directly to the OVAL Moderator at [oval@mitre.org](mailto:oval@mitre.org)[[2]](#footnote-2).

Contents

[Acknowledgements 1](#_Toc322096712)

[Trademark Information 1](#_Toc322096713)

[Terms of Use 1](#_Toc322096714)

[Feedback 1](#_Toc322096715)

[1. Introduction 3](#_Toc322096716)

[1.1 Document Conventions 3](#_Toc322096717)

[1.2 Document Structure 4](#_Toc322096718)

[2. OVAL Language LINUX Component Model 4](#_Toc322096719)

[2.1 Data Model Conventions 4](#_Toc322096720)

[2.2 linux-def:rpminfo\_test 5](#_Toc322096721)

[2.2.1 Known Supported Platforms 5](#_Toc322096722)

[2.3 linux-def:rpminfo\_object 5](#_Toc322096723)

[2.4 unix-def:RpmInfoBehaviors 6](#_Toc322096724)

[2.5 unix-def:rpminfo\_state 7](#_Toc322096725)

[2.6 linux-sc:rpminfo\_item 11](#_Toc322096726)

[2.7. linux-def:dpkginfo\_test 16](#_Toc322096727)

[2.7.1. Known Supported Platforms 17](#_Toc322096728)

[2.8. linux-def:dpkginfo\_object 17](#_Toc322096729)

[2.9. linux-def:dpkginfo\_state 18](#_Toc322096730)

[2.10. linux-sc:dpkginfo\_item 20](#_Toc322096731)

[2.11 linux-def:partition\_test 23](#_Toc322096732)

[2.11.1 Known Supported Platforms 24](#_Toc322096733)

[2.12 linux-def:partition\_object 24](#_Toc322096734)

[2.13 linux-def:partition\_state 25](#_Toc322096735)

[2.14 linux-sc:partition\_item 27](#_Toc322096736)

[2.15 linux-def:inetlisteningservers\_test 30](#_Toc322096737)

[2.15.1 Known Supported Platforms 30](#_Toc322096738)

[2.16 linux-def:inetlisteningservers\_object 30](#_Toc322096739)

[2.17 linux-def:inetlisteningservers\_state 32](#_Toc322096740)

[2.18 linux-sc:inetlisteningservers\_item 34](#_Toc322096741)

[Appendix A – Normative References 37](#_Toc322096742)

[Appendix B - Change Log 37](#_Toc322096743)

[Appendix C – Terms and Acronyms 37](#_Toc322096744)

# Introduction

## 1.1 Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in *RFC 2119* [1].

The following font and font style conventions are used throughout the remainder of this document:

* The Courier New font without formatting is used for writing constructs in the OVAL Language Data Model. When the font is **boldfaced**, it indicates commands on the UNIX command line.

Examples: generator (OVAL Construct), **ls –al** (UNIX command)

* The *'italic, with single quotes'* font is used for noting values for OVAL Language properties.

Example: *'does not exist'*

* The bold font and the keyword **Default Value:** are used to indicate a property's default value.

Example: **Default Value: -1**

* The bold font and the keyword **xsi:nil="true":** are used to indicate the meaning of an entity when the xsi:nil property is set to true.

Example: **xsi:nil="true"** indicates that the file\_object MUST collect the set of directories specified by the path entity. In addition, a value, for the filename entity, MUST NOT be specified.

This document uses the concept of namespaces[[3]](#footnote-3) to logically group OVAL constructs throughout both the Data Model section of the document, as well as other parts of the specification. The format of these namespaces is prefix:element, where the prefix is the namespace component, and the element is the name of the qualified construct. The following table lists the namespaces used in this document:

|  |  |  |  |
| --- | --- | --- | --- |
| Data Model | Namespace | Description | Example |
| OVAL Definitions | oval-def | The OVAL Definitions data model that defines the core framework constructs for creating OVAL Definitions. This is defined in the OVAL Language Specification [2]. | oval-def:TestType |
| OVAL System Characteristics | oval-sc | The OVAL System Characteristics data model, which defines the constructs used to capture the data collected on a target system. This is defined in the OVAL Language Specification. | oval-sc:ItemType |
| Linux Definitions | linux-def | The Linux Definitions data model defines the platform-specific constructs used in OVAL Definitions to make assertions about the state of Linux systems. | linux-def:file\_test |
| Linux System Characteristics | linux-sc | The Linux System Characteristics data model defines the platform-specific constructs used in OVAL System Characteristics to represent the system state information collected from Linux systems. | linux-sc:file\_item |

Lastly, each OVAL Test will contain a section titled "Known Supported Platforms" that specifies which platforms the OVAL Test is known to work on. This section is provided for convenience only and should not be considered a comprehensive list. In addition, there may be further known support restrictions specified for behaviors or entities that supersede the "Known Supported Platforms" section for the OVAL Test.

## 1.2 Document Structure

This document serves as the specification for the Linux extension of the OVAL Language Specification and defines the platform-specific data model. This document is organized into the following sections:

* Section 1 – Introduction
* Section 2 – OVAL Language Linux Component Model
* Appendix A – References
* Appendix B – Change Log
* Appendix C – Terms and Acronyms

# OVAL Language Linux Component Model

The OVAL Language Linux Component Data Model is the platform-specific extension of the OVAL Language Data Model for Linux operating systems.

## Data Model Conventions

This document follows the data model conventions described in Section 4.1 of the OVAL Language Specification.

## linux-def:rpminfo\_test

The rpminfo\_test is used to make assertions about the RPM header information for a given RPM package on a Linux system. Linux OSes that use the RPM package management system can be managed with the rpm program. The rpminfo\_test MUST reference one rpminfo\_object and zero or more rpminfo\_states.



### Known Supported Platforms

* Red Hat Enterprise Linux 5

## linux-def:rpminfo\_object

The rpminfo\_object construct defines the RPM whose associated information[[4]](#footnote-4) should be collected and represented as rpminfo\_items.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | false | Enables the expression of complex rpminfo\_objects that are the result of logically combining and filtering the rpminfo\_items that are identified by one or more rpminfo\_objects.  Please see the OVAL Language Specification [2] for additional information. |
| behaviors | linux-def:  RpmInfoBehaviors | 0..1 | false | Specifies the behaviors that direct how the rpminfo\_object collects rpminfo\_items from the system. |
| name | oval-def:  EntityObjectStringType | 0..1 | false | This is the package name to check[[5]](#footnote-5). |
| filter | oval-def:filter | 0..\* | false | Allows for the explicit inclusion or exclusion of rpminfo\_items from the set of rpminfo\_items collected by an rpminfo\_object.  Please see the OVAL Language Specification [2] for additional information. |

## linux-def:RpmInfoBehaviors

The RpmInfoBehaviors construct defines the behaviors that direct how the rpminfo\_object collects rpminfo\_items from the system. Note that using these behaviors may result in some unique results. For example, a double negative type condition might be created where an object entity says include everything except a specific item, but a behavior is used that might then add that item back in.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Type | Possible Values | Description |
| filepaths | boolean | *true*  *false* | Determines whether or not to collect all the filepaths (directories and files) associated with an RPM on a local system.  *true*: All filepaths (directories and files) associated with an RPM on the local system MUST be collected from the RPM database.  *false:* Filepaths (directories and files) associated with an RPM on the local system MUST NOT be collected from the RPM database.  **Default Value: false** |

## linux-def:rpminfo\_state

The rpminfo\_state construct is used by an rpminfo\_test to specify the RPM header information to check for RPM packages on Linux platforms[[6]](#footnote-6). RPMs follow the file naming format of name-version-release.architecture.rpm[[7]](#footnote-7).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| name | oval-def:EntityStateStringType | 0..1 | false | This is the RPM package name to check[[8]](#footnote-8).  The rpmtsInitIterator[[9]](#footnote-9) function and RPMTAG\_NAME[[10]](#footnote-10) can be used to retrieve the specified RPM header. |
| arch | oval-def:EntityStateStringType | 0..1 | false | This is the architecture for which the RPM package was built[[11]](#footnote-11).  The headerGetEntry[[12]](#footnote-12) function and RPMTAG\_ARCH[[13]](#footnote-13) can be used to retrieve the architecture of the RPM package. |
| epoch | oval-def:  EntityStateAnySimpleType | 0..1 | false | This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking.  For a null epoch (or '(none)' as returned by rpm), the string '(none)' SHOULD be used, not the value 0 that is outputted by the rpm program.  The datatype of this property MUST be string or int. Please see the SimpleDatatypeEnumeration in the OVAL Language Specification [2] for additional information.  **rpm -q --qf '%{EPOCH}\n' installed\_rpm** can be used to retrieve the epoch of an RPM package.  The headerGetEntry[[14]](#footnote-14) function and RPMTAG\_EPOCH[[15]](#footnote-15) can also be used to retrieve the epoch of the RPM package. |
| release | oval-def:  EntityStateAnySimpleType | 0..1 | false | This is the release number of the build, changed by the vendor/builder.  The datatype of this property MUST be string or version. Please see the SimpleDatatypeEnumeration in the OVAL Language Specification [2] for additional information.  The headerGetEntry[[16]](#footnote-16) function and RPMTAG\_RELEASE[[17]](#footnote-17) can be used to retrieve the release of the RPM package. |
| version | oval-def:  EntityStateAnySimpleType | 0..1 | false | This is the version number of the build.  The datatype of this property MUST be string or version. Please see the SimpleDatatypeEnumeration in the OVAL Language Specification [2] for additional information.  The headerGetEntry[[18]](#footnote-18) function and RPMTAG\_VERSION[[19]](#footnote-19) can be used to retrieve the version of the RPM package. |
| evr | oval-def:  EntityStateEVRStringType | 0..1 | false | This represents the epoch, version, and release fields as a single version string.  It has the form "EPOCH:VERSION-RELEASE".  A null epoch (or '(none)' as returned by rpm) is equivalent to and MUST be represented as a '0'. |
| signature\_keyid | oval-def:EntityStateStringType | 0..1 | false | This is the 64-bit PGP key ID that the RPM issuer (generally the original operating system vendor) uses to sign the key.  Note that the value SHOULD NOT contain a hyphen to separate the higher 32-bits from the lower 32-bits.  It should simply be a 16 character hex string. PGP is used to verify the authenticity and integrity of the RPM being considered.  Software packages and patches are signed cryptographically to allow administrators to allay concerns that the distribution mechanism has been compromised, whether that mechanism is web site, FTP server, or even a mirror controlled by a hostile party.  OVAL uses this field most of all to confirm that the package installed on the system is that shipped by the vendor, since comparing package version numbers against patch announcements is only programmatically valid if the installed package is known to contain the patched code.  To find this key ID, use the command **rpm –qi <package name>** and look at the Signature line where it says Key ID[[20]](#footnote-20). |
| extended\_name | oval-def:EntityStateStringType | 0..1 | false | This is the name, epoch, version, release, and architecture fields as a single version string. It has the form "NAME-EPOCH:VERSION-RELEASE.ARCHITECTURE".  A null epoch (or '(none)' as returned by rpm) is equivalent to and MUST be represented as a '0'. |
| filepath | oval-def:EntityStateStringType | 0..1 | false | This is the absolute path of a file or directory included in the RPM.  **rpm –ql installed\_rpm** can be used to retrieve the directories and files associated with an RPM package.  The rpmfiNew[[21]](#footnote-21) function and RPMTAG\_BASENAMES[[22]](#footnote-22) can also be used to retrieve the directories and files associated with an RPM package. |

## linux-sc:rpminfo\_item

The rpminfo\_item construct defines the RPM header information associated with RPM packages on Linux platforms[[23]](#footnote-23). RPMs follow the file naming format of name-version-release.architecture.rpm[[24]](#footnote-24).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| name | oval-sc:EntityItemStringType | 0..1 | false | This is the RPM package name to check[[25]](#footnote-25).  The rpmtsInitIterator[[26]](#footnote-26) function and RPMTAG\_NAME[[27]](#footnote-27) can be used to retrieve the specified RPM header. |
| arch | oval-sc:EntityItemStringType | 0..1 | false | This is the architecture for which the RPM package was built[[28]](#footnote-28).  The headerGetEntry[[29]](#footnote-29) function and RPMTAG\_ARCH[[30]](#footnote-30) can be used to retrieve the architecture of the RPM package. |
| epoch | oval-sc:  EntityItemAnySimpleType | 0..1 | false | This is the epoch number of the RPM, this is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking.  For a null epoch (or '(none)' as returned by **rpm**), the string '(none)' SHOULD be used, not the value 0 that is outputted by the **rpm** program.  The datatype of this property MUST be string or int. Please see the SimpleDatatypeEnumeration in the OVAL Language Specification [2] for additional information.  **rpm -q --qf '%{EPOCH}\n' installed\_rpm** can be used to retrieve the epoch of an RPM package.  The headerGetEntry[[31]](#footnote-31) function and RPMTAG\_EPOCH[[32]](#footnote-32) can also be used to retrieve the epoch of the RPM package. |
| release | oval-sc:  EntityItemAnySimpleType | 0..1 | false | This is the release number of the build, changed by the vendor/builder.  The datatype of this property MUST be string or version. Please see the SimpleDatatypeEnumeration in the OVAL Language Specification [2] for additional information.  The headerGetEntry[[33]](#footnote-33) function and RPMTAG\_RELEASE[[34]](#footnote-34) can be used to retrieve the release of the RPM package. |
| version | oval-sc:  EntityItemAnySimpleType | 0..1 | false | This is the version number of the build.  The datatype of this property MUST be string or version. Please see the SimpleDatatypeEnumeration in the OVAL Language Specification [2] for additional information.  The headerGetEntry[[35]](#footnote-35) function and RPMTAG\_VERSION[[36]](#footnote-36) can be used to retrieve the version of the RPM package. |
| evr | oval-sc:  EntityItemEVRStringType | 0..1 | false | This represents the epoch, version, and release fields as a single version string.  It has the form "EPOCH:VERSION-RELEASE".  A null epoch (or '(none)' as returned by rpm) is equivalent to and MUST be represented as a '0'. |
| signature\_keyid | oval-sc:EntityItemStringType | 0..1 | false | This is the 64-bit PGP key ID that the RPM issuer (generally the original operating system vendor) uses to sign the key.  Note that the value SHOULD NOT contain a hyphen to separate the higher 32-bits from the lower 32-bits.  It should simply be a 16 character hex string. PGP is used to verify the authenticity and integrity of the RPM being considered.  Software packages and patches are signed cryptographically to allow administrators to allay concerns that the distribution mechanism has been compromised, whether that mechanism is web site, FTP server, or even a mirror controlled by a hostile party.  OVAL uses this field most of all to confirm that the package installed on the system is that shipped by the vendor, since comparing package version numbers against patch announcements is only programmatically valid if the installed package is known to contain the patched code.  To find this key ID, use the command **rpm –qi <package name>** and look at the Signature line where it says Key ID[[37]](#footnote-37). |
| extended\_name | oval-sc:EntityItemStringType | 0..1 | false | This is the name, epoch, version, release, and architecture fields as a single version string. It has the form "NAME-EPOCH:VERSION-RELEASE.ARCHITECTURE".  A null epoch (or '(none)' as returned by rpm) is equivalent to and MUST be represented as a '0'. |
| filepath | oval-sc:EntityItemStringType | 0..1 | false | This is the absolute path of a file or directory included in the RPM.  **rpm –ql installed\_rpm** can be used to retrieve the directories and files associated with an RPM package.  The rpmfiNew[[38]](#footnote-38) function and RPMTAG\_BASENAMES[[39]](#footnote-39) can also be used to retrieve the directories and files associated with an RPM package. |

## linux-def:dpkginfo\_test

The dpkginfo\_test is used to make assertions about the Debian package information on a Linux system[[40]](#footnote-40). Some Linux OSes, that use the Debian package management system, can be managed with the dpkg[[41]](#footnote-41) program. The dpkginfo\_test MUST reference one dpkginfo\_object and zero or more dpkginfo\_states.





### Known Supported Platforms

* Debian Sid

## linux-def:dpkginfo\_object

The dpkginfo\_object construct defines the deb header information[[42]](#footnote-42) that should be collected and represented as dpkginfo\_items.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | False | Enables the expression of complex dpkginfo\_objects that are the result of logically combining and filtering the dpkginfo\_items that are identified by one or more dpkginfo\_objects.  Please see the OVAL Language Specification [2] for additional information. |
| name | oval-def:  EntityObjectStringType | 0..1 | False | This is the Debian package name to check. |
| filter | oval-def:filter | 0..\* | False | Allows for the explicit inclusion or exclusion of dpkginfo\_items from the set of dpkginfo\_items collected by a dpkginfo\_object.  Please see the OVAL Language Specification [2] for additional information. |

## linux-def:dpkginfo\_state

The dpkginfo\_state construct is used by a dpkginfo\_test to specify deb package header information[[43]](#footnote-43) on LINUX platforms.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| name | oval-def:EntityStateStringType | 0..1 | false | This is the Debian package name to check. |
| arch | oval-def:EntityStateStringType | 0..1 | false | This is the architecture for which the Debian package was built.  The FindS function and the string "Architecture" can be used to retrieve the architecture from the pkgTagSection of the installed Debian package[[44]](#footnote-44). |
| epoch | oval-def:  EntityStateAnySimpleType | 0..1 | false | This is the epoch[[45]](#footnote-45) number of the deb package that is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking.  For a null epoch,the value 0 MUST be used[[46]](#footnote-46). |
| release | oval-def:  EntityStateAnySimpleType | 0..1 | false | This is the revision number of the build, changed by the vendor/builder. It is also known as the debian\_revision[[47]](#footnote-47). If it is not indicated, this value is zero.[[48]](#footnote-48). |
| version | oval-def:  EntityStateAnySimpleType | 0..1 | false | This is the version number of the build. It is also known as upstream\_version[[49]](#footnote-49).  The FindS function and the string "Version" can be used to retrieve the release from the pkgTagSection of the installed Debian package[[50]](#footnote-50). |
| evr | oval-def:  EntityStateEVRStringType | 0..1 | false | This represents the epoch, version, and release fields as a single version string. It has the form "EPOCH:VERSION-RELEASE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form 0:VERSION-RELEASE[[51]](#footnote-51). |

## linux-sc:dpkginfo\_item

The dpkginfo\_item construct specifies deb package header information on a LINUX system[[52]](#footnote-52).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| name | oval-sc:EntityItemStringType | 0..1 | false | This is the Debian package name to check. |
| arch | oval-sc:EntityItemStringType | 0..1 | false | This is the architecture for which the Debian package was built.  The FindS function and the string "Architecture" can be used to retrieve the architecture from the pkgTagSection of the installed Debian package[[53]](#footnote-53). |
| epoch | oval-sc:  EntityItemAnySimpleType | 0..1 | false | This is the epoch[[54]](#footnote-54) number of the deb package that is used as a kludge for version-release comparisons where the vendor has done some kind of re-numbering or version forking.  For a null epoch,the value 0 MUST be used[[55]](#footnote-55). |
| release | oval-sc:  EntityItemAnySimpleType | 0..1 | false | This is the revision number of the build, changed by the vendor/builder. It is also known as the debian\_revision[[56]](#footnote-56). If it is not indicated, this value is zero.[[57]](#footnote-57). |
| version | oval-sc:  EntityItemAnySimpleType | 0..1 | false | This is the version number of the build. It is also known as upstream\_version[[58]](#footnote-58).  The FindS function and the string "Version" can be used to retrieve the release from the pkgTagSection of the installed Debian package[[59]](#footnote-59). |
| evr | oval-sc:  EntityItemEVRStringType | 0..1 | false | This represents the epoch, version, and release fields as a single version string. It has the form "EPOCH:VERSION-RELEASE". Note that a null epoch (or '(none)' as returned by rpm) is equivalent to '0' and would hence have the form 0:VERSION-RELEASE[[60]](#footnote-60). |

## linux-def:partition\_test

The partition\_test is used to make assertions about partition information[[61]](#footnote-61) on Linux systems. The partition\_test MUST reference one partition\_object and zero or more partition\_states.



### Known Supported Platforms

* Red Hat Enterprise Linux 5

## linux-def:partition\_object

The partition\_object construct defines the set of partitions whose associated system state information should be collected and represented as partition\_items.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | false | Enables the expression of complex partition\_objects that are the result of logically combining and filtering the partition\_items that are identified by one or more partition\_objects.  Please see the OVAL Language Specification [2] for additional information. |
| mount\_point | oval-def:  EntityObjectStringType | 0..1 | false | This property specifies the mount points of the partitions on the system. A mount point is the root directory of the mounted file system[[62]](#footnote-62).  The mount points can be retrieved from /proc/mounts or via the dj command[[63]](#footnote-63). |
| filter | oval-def:filter | 0..\* | false | Allows for the explicit inclusion or exclusion of partition\_items from the set of partition\_items collected by a partition\_object.  Please see the OVAL Language Specification [2] for additional information. |

## linux-def:partition\_state

The partiton\_state construct is used by a partition\_test to specify the partition information to check about partitions on Linux platforms.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| mount\_point | oval-def:EntityStateStringType | 0..1 | false | This property specifies the mount points of the partitions on the system. A mount point is the root directory of the mounted file system[[64]](#footnote-64).  The mount points can be retrieved from /proc/mounts or via the dj command[[65]](#footnote-65). |
| device | oval-def:EntityStateStringType | 0..1 | false | The name of the device that is mounted at the specified mount point. This is the FileSystem column of the output of the df command[[66]](#footnote-66).  The device name can be retrieved from /proc/mounts[[67]](#footnote-67). |
| uuid | oval-def:EntityStateStringType | 0..1 | false | The universally unique identifier associated with a partition.  The blkid\_get\_tag\_value[[68]](#footnote-68)[[69]](#footnote-69) function or the blkid command[[70]](#footnote-70) can be used to retrieve the UUID information. |
| fs\_type | oval-def:EntityStateStringType | 0..1 | false | The type of filesystem on a partition.  Filesystem types are outputted via the df –T command[[71]](#footnote-71).  The file system type can be retrieved from /proc/mounts[[72]](#footnote-72). |
| mount\_options | oval-def:EntityStateStringType | 0..1 | false | The mount options associated with a partition.  The mount options for a partition can be retrieved from /proc/mounts[[73]](#footnote-73). |
| total\_space | oval-def:EntityStateIntType | 0..1 | false | An integer that represents the total number of blocks on a partition.  The statfs[[74]](#footnote-74) function in conjunction with the f\_blocks member of the statfs structure can be used to retrieve the total space of the partition. |
| space\_used | oval-def:EntityStateIntType | 0..1 | false | An integer that represents the number of blocks used on a partition.  The statfs[[75]](#footnote-75) function in conjunction with the f\_blocks and f\_bfree members of the statfs structure can be used to retrieve the total space of the partition. Specifically, f\_blocks – f\_bfree. |
| space\_left | oval-def:EntityStateIntType | 0..1 | false | An integer that represents the number of blocks left on a partition.  The statfs[[76]](#footnote-76) function in conjunction with the f\_bavail member of the statfs structure can be used to retrieve the space left in the partition. |

## linux-sc:partition\_item

The partition\_item construct defines the partition information associated with services with files and directories on file systems supported by the LINUX platform.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| mount\_point | oval-sc:EntityItemStringType | 0..1 | false | This property specifies the mount points of the partitions on the system. A mount point is the root directory of the mounted file system[[77]](#footnote-77).  The mount points can be retrieved from /proc/mounts or via the dj command[[78]](#footnote-78). |
| device | oval-sc:EntityItemStringType | 0..1 | false | The name of the device that is mounted at the specified mount point. This is the FileSystem column of the output of the df command[[79]](#footnote-79).  The device name can be retrieved from /proc/mounts[[80]](#footnote-80). |
| uuid | oval-sc:EntityItemStringType | 0..1 | false | The universally unique identifier associated with a partition.  The blkid\_get\_tag\_value[[81]](#footnote-81)[[82]](#footnote-82) function or the blkid command[[83]](#footnote-83) can be used to retrieve the UUID information. |
| fs\_type | oval-sc:EntityItemStringType | 0..1 | false | The type of filesystem on a partition.  Filesystem types are outputted via the df –T command[[84]](#footnote-84).  The file system type can be retrieved from /proc/mounts[[85]](#footnote-85). |
| mount\_options | oval-sc:EntityItemStringType | 0..1 | false | The mount options associated with a partition.  The mount options for a partition can be retrieved from /proc/mounts[[86]](#footnote-86). |
| total\_space | oval-sc:EntityItemIntType | 0..1 | false | An integer that represents the total number of blocks on a partition.  The statfs[[87]](#footnote-87) function in conjunction with the f\_blocks member of the statfs structure can be used to retrieve the total space of the partition. |
| space\_used | oval-sc:EntityItemIntType | 0..1 | false | An integer that represents the number of blocks used on a partition.  The statfs[[88]](#footnote-88) function in conjunction with the f\_blocks and f\_bfree members of the statfs structure can be used to retrieve the total space of the partition. Specifically, f\_blocks – f\_bfree. |
| space\_left | oval-sc:EntityItemIntType | 0..1 | false | An integer that represents the number of blocks left on a partition.  The statfs[[89]](#footnote-89) function in conjunction with the f\_bavail member of the statfs structure can be used to retrieve the space left in the partition. |

## linux-def:inetlisteningservers\_test

The inetlisteningservers\_test is used to make assertions about applications listening on the network. This is limited to applications using the TCP or UDP protocols with addresses represented as IPv4 or IPv6 addresses, corresponding to the AF\_INET and AF\_INET6 values, respectively. To acquire this information, an administrator can use the netstat –tuwlnpe command[[90]](#footnote-90). The inetlisteningservers\_test MUST reference one inetlisteningservers\_object and zero or more inetlisteningservers\_states.



### Known Supported Platforms

* Red Hat Enterprise Linux 5

## linux-def:inetlisteningservers\_object

The inetlisteningservers\_object construct defines the set of applications listening on the network whose associated information should be collected and represented as inetlisteningservers\_items.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| set | oval-def:set | 0..1 | false | Enables the expression of complex inetlisteningservers\_objects that are the result of logically combining and filtering the inetlisteningservers\_items that are identified by one or more inetlisteningservers\_objects.  Please see the OVAL Language Specification [2] for additional information. |
| protocol | oval-def:  EntityObjectStringType | 0..1 | false | Specifies the transport-layer protocol, in lowercase: tcp or udp, associated with the programs listening on the system. |
| local\_address | oval-def:  EntityObjectIPAddressStringType | 0..1 | false | Specifies the IP address of the network interface on which the program listens. The IP address can be either IPv4 or IPv6. |
| local\_port | oval-def:EntityObjectIntType | 0..1 | false | This is the TCP or UDP port number associated with the program listening on the system. |
| filter | oval-def:filter | 0..\* | false | Allows for the explicit inclusion or exclusion of process\_items from the set of process\_items collected by a process\_object.  Please see the OVAL Language Specification [2] for additional information. |

## linux-def:inetlisteningservers\_state

The inetlisteningservers\_state construct is used by an inetlisteningservers\_test to specify information about the specified program listening on Linux platforms. To get this information an administrator can use the netstat –tuwlnpe command[[91]](#footnote-91).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| protocol | oval-def:EntityStateStringType | 0..1 | false | Specifies the transport-layer protocol, in lowercase: tcp or udp, associated with the programs listening on the system. |
| local\_address | oval-def:  EntityStateIPAddress StringType | 0..1 | false | Specifies the IP address of the network interface on which the program listens. The IP address can be either IPv4 or IPv6. |
| local\_port | oval-def:EntityStateIntType | 0..1 | false | This is the TCP or UDP port number associated with the program listening on the system. |
| local\_full\_address | oval-def:EntityStateStringType | 0..1 | false | This is the IP address and network port number associated with the program listening on the system, equivalent to local\_address:local\_port. Note that the IP address can be IPv4 or IPv6. |
| program\_name | oval-def:EntityStateStringType | 0..1 | false | This is the name of the communicating program. |
| foreign\_address | oval-def:  EntityStateIPAddress StringType | 0..1 | false | This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening program. Note that the IP address can be IPv4 or IPv6. |
| foreign\_port | oval-def:EntityStateIntType | 0..1 | false | This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, this is usually a \* in Linux. Starting with version 5.10.1, the "\*" MUST be represented as 0. |
| foreign\_full\_address | oval-def:EntityStateStringType | 0..1 | false | This is the IP address and network port to which the program is communicating or will accept communications from, equivalent to foreign\_address:foreign\_port. Note that the IP address can be IPv4 or IPv6. |
| pid | oval-def:EntityStateIntType | 0..1 | false | The pid is the process ID of the process associated with the listening program. The netstat –tuwlnpe command must be run as root to get this information, or the output will be the dash '-' character, which is NOT an integer and does NOT represent an acceptable value for this property. |
| user\_id | oval-def:EntityStateIntType | 0..1 | false | The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program. |

## linux-sc:inetlisteningservers\_item

The inetlisteningservers\_item construct defines the information associated with listening programs supported by the Linux platform. To get this information an administrator can use the netstat –tuwlnpe command[[92]](#footnote-92).



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Type | Multiplicity | Nillable | Description |
| protocol | oval-sc:EntityItemStringType | 0..1 | false | Specifies the transport-layer protocol, in lowercase: tcp or udp, associated with the programs listening on the system. |
| local\_address | oval-sc:  EntityItemIPAddress StringType | 0..1 | false | Specifies the IP address of the network interface on which the program listens. The IP address can be either IPv4 or IPv6. |
| local\_port | oval-sc:EntityItemIntType | 0..1 | false | This is the TCP or UDP port number associated with the program listening on the system. |
| local\_full\_address | oval-sc:EntityItemStringType | 0..1 | false | This is the IP address and network port number associated with the program listening on the system, equivalent to local\_address:local\_port. Note that the IP address can be IPv4 or IPv6. |
| program\_name | oval-sc:EntityItemStringType | 0..1 | false | This is the name of the communicating program. |
| foreign\_address | oval-sc:  EntityItemIPAddress StringType | 0..1 | false | This is the IP address with which the program is communicating, or with which it will communicate, in the case of a listening program. Note that the IP address can be IPv4 or IPv6. |
| foreign\_port | oval-sc:EntityItemIntType | 0..1 | false | This is the TCP or UDP port to which the program communicates. In the case of a listening program accepting new connections, this is usually a \* in Linux. Starting with version 5.10.1, the "\*" MUST be represented as 0. |
| foreign\_full\_address | oval-sc:EntityItemStringType | 0..1 | false | This is the IP address and network port to which the program is communicating or will accept communications from, equivalent to foreign\_address:foreign\_port. Note that the IP address can be IPv4 or IPv6. |
| pid | oval-sc:EntityItemIntType | 0..1 | false | The pid is the process ID of the process associated with the listening program. The netstat –tuwlnpe command must be run as root to get this information, or the output will be the dash '-' character, which is NOT an integer and does NOT represent an acceptable value for this property. |
| user\_id | oval-sc:EntityItemIntType | 0..1 | false | The numeric user id, or uid, is the third column of each user's entry in /etc/passwd. It represents the owner, and thus privilege level, of the specified program. |

# Appendix A – Normative References

[1] RFC 2119 – Key words for use in RFCs to Indicate Requirement Levels

<http://www.ietf.org/rfc/rfc2119.txt>

[2] The OVAL Language Specification

<http://oval.mitre.org/language/version5.10#specification>

# Appendix B - Change Log

# 

# Appendix C – Terms and Acronyms

1. For more information see <https://oval.mitre.org/about/termsofuse.html> [↑](#footnote-ref-1)
2. For more information see <https://oval.mitre.org/> [↑](#footnote-ref-2)
3. For more information see <http://en.wikipedia.org/wiki/Namespace_(computer_science)> [↑](#footnote-ref-3)
4. For more information see http://www.rpm.org/max-rpm/rpm.8.html. [↑](#footnote-ref-4)
5. For more information see http://www.rpm.org/max-rpm/ch-rpm-file-format.html. [↑](#footnote-ref-5)
6. For more information see <http://www.rpm.org/max-rpm/rpm.8.html>. [↑](#footnote-ref-6)
7. For more information see <http://www.rpm.org/max-rpm/ch-rpm-file-format.html>. [↑](#footnote-ref-7)
8. For more information see <http://www.rpm.org/max-rpm/rpm.8.html>. [↑](#footnote-ref-8)
9. For more information see <http://rpm.org/api/4.4.2.2/group__rpmts.html#g7268df6d90d55c033214f5feb743620f> [↑](#footnote-ref-9)
10. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591> [↑](#footnote-ref-10)
11. For more information see <http://www.rpm.org/max-rpm/rpm.8.html>. [↑](#footnote-ref-11)
12. For more information see <http://rpm.org/api/4.4.2.2/group__header.html#gddebc5eb6a9605829e11a5b356b36d33>. [↑](#footnote-ref-12)
13. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-13)
14. For more information see <http://rpm.org/api/4.4.2.2/group__header.html#gddebc5eb6a9605829e11a5b356b36d33>. [↑](#footnote-ref-14)
15. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-15)
16. For more information see <http://rpm.org/api/4.4.2.2/group__header.html#gddebc5eb6a9605829e11a5b356b36d33>. [↑](#footnote-ref-16)
17. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-17)
18. For more information see <http://rpm.org/api/4.4.2.2/group__header.html#gddebc5eb6a9605829e11a5b356b36d33>. [↑](#footnote-ref-18)
19. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-19)
20. For more information see <http://susefaq.sourceforge.net/articles/rpm.html> [↑](#footnote-ref-20)
21. For more information see <http://rpm.org/api/4.4.2.2/rpmfi_8c.html#7be3dd0af7bab1e4b324c891f76dc42b>. [↑](#footnote-ref-21)
22. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-22)
23. For more information see <http://distrowatch.com/dwres.php?resource=package-management>. To obtain all packages installed on the system, look for the task "List Installed packages" and utilize the appropriate command. [↑](#footnote-ref-23)
24. For more information see <http://www.rpm.org/max-rpm/ch-rpm-file-format.html> [↑](#footnote-ref-24)
25. For more information see <http://www.rpm.org/max-rpm/rpm.8.html>. [↑](#footnote-ref-25)
26. For more information see <http://rpm.org/api/4.4.2.2/group__rpmts.html#g7268df6d90d55c033214f5feb743620f> [↑](#footnote-ref-26)
27. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591> [↑](#footnote-ref-27)
28. For more information see <http://www.rpm.org/max-rpm/rpm.8.html>. [↑](#footnote-ref-28)
29. For more information see <http://rpm.org/api/4.4.2.2/group__header.html#gddebc5eb6a9605829e11a5b356b36d33>. [↑](#footnote-ref-29)
30. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-30)
31. For more information see <http://rpm.org/api/4.4.2.2/group__header.html#gddebc5eb6a9605829e11a5b356b36d33>. [↑](#footnote-ref-31)
32. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-32)
33. For more information see <http://rpm.org/api/4.4.2.2/group__header.html#gddebc5eb6a9605829e11a5b356b36d33>. [↑](#footnote-ref-33)
34. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-34)
35. For more information see <http://rpm.org/api/4.4.2.2/group__header.html#gddebc5eb6a9605829e11a5b356b36d33>. [↑](#footnote-ref-35)
36. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-36)
37. For more information see <http://susefaq.sourceforge.net/articles/rpm.html> [↑](#footnote-ref-37)
38. For more information see <http://rpm.org/api/4.4.2.2/rpmfi_8c.html#7be3dd0af7bab1e4b324c891f76dc42b>. [↑](#footnote-ref-38)
39. For more information see <http://rpm.org/api/4.4.2.2/rpmlib_8h.html#d1edbc08d3fcb1a7678eb956cfc33591>. [↑](#footnote-ref-39)
40. For more information see <http://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.html> [↑](#footnote-ref-40)
41. For more information see <http://manpages.debian.net/cgi-bin/man.cgi?query=dpkg&apropos=0&sektion=0&manpath=Debian+6.0+squeeze&format=html&locale=en> [↑](#footnote-ref-41)
42. For more information see <http://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.html> [↑](#footnote-ref-42)
43. For more information see <http://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.html> [↑](#footnote-ref-43)
44. For more information see <http://apt.sourcearchive.com/documentation/0.8.16~exp5ubuntu14.2/classpkgTagSection.html> [↑](#footnote-ref-44)
45. For more information see <http://manpages.debian.net/cgi-bin/man.cgi?query=deb-version&apropos=0&sektion=0&manpath=Debian+6.0+squeeze&format=html&locale=en> [↑](#footnote-ref-45)
46. For more information see <http://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version> [↑](#footnote-ref-46)
47. For more information see <http://manpages.debian.net/cgi-bin/man.cgi?query=deb-version&apropos=0&sektion=0&manpath=Debian+6.0+squeeze&format=html&locale=en> [↑](#footnote-ref-47)
48. For more information see <http://apt.sourcearchive.com/documentation/0.8.16~exp5ubuntu14.2/classpkgTagSection.html> [↑](#footnote-ref-48)
49. For more information see <http://manpages.debian.net/cgi-bin/man.cgi?query=deb-version&apropos=0&sektion=0&manpath=Debian+6.0+squeeze&format=html&locale=en> [↑](#footnote-ref-49)
50. For more information see <http://apt.sourcearchive.com/documentation/0.8.16~exp5ubuntu14.2/classpkgTagSection.html> [↑](#footnote-ref-50)
51. For more information see <http://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version> [↑](#footnote-ref-51)
52. For more information see <http://www.debian.org/doc/manuals/debian-faq/ch-pkg_basics.html> [↑](#footnote-ref-52)
53. For more information see <http://apt.sourcearchive.com/documentation/0.8.16~exp5ubuntu14.2/classpkgTagSection.html> [↑](#footnote-ref-53)
54. For more information see <http://manpages.debian.net/cgi-bin/man.cgi?query=deb-version&apropos=0&sektion=0&manpath=Debian+6.0+squeeze&format=html&locale=en> [↑](#footnote-ref-54)
55. For more information see <http://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version> [↑](#footnote-ref-55)
56. For more information see <http://manpages.debian.net/cgi-bin/man.cgi?query=deb-version&apropos=0&sektion=0&manpath=Debian+6.0+squeeze&format=html&locale=en> [↑](#footnote-ref-56)
57. For more information see <http://apt.sourcearchive.com/documentation/0.8.16~exp5ubuntu14.2/classpkgTagSection.html> [↑](#footnote-ref-57)
58. For more information see <http://manpages.debian.net/cgi-bin/man.cgi?query=deb-version&apropos=0&sektion=0&manpath=Debian+6.0+squeeze&format=html&locale=en> [↑](#footnote-ref-58)
59. For more information see <http://apt.sourcearchive.com/documentation/0.8.16~exp5ubuntu14.2/classpkgTagSection.html> [↑](#footnote-ref-59)
60. For more information see <http://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version> [↑](#footnote-ref-60)
61. For more information see <http://www.cyberciti.biz/tips/the-importance-of-linux-partitions.html>, <http://tech.zhenhua.info/2008/07/get-file-system-and-partition.html>, and <http://www.linfo.org/mount_point.html> [↑](#footnote-ref-61)
62. For more information see <http://www.linfo.org/mount_point.html> [↑](#footnote-ref-62)
63. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> and <http://linux.die.net/man/1/df> [↑](#footnote-ref-63)
64. For more information see <http://www.linfo.org/mount_point.html> [↑](#footnote-ref-64)
65. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> and <http://linux.die.net/man/1/df> [↑](#footnote-ref-65)
66. For more information see <http://linux.die.net/man/1/df> [↑](#footnote-ref-66)
67. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> [↑](#footnote-ref-67)
68. For more information see <http://linux.die.net/man/3/libblkid> [↑](#footnote-ref-68)
69. For more information see <http://www.kernel.org/pub/linux/utils/util-linux/v2.21/libblkid-docs/libblkid-Search-and-iterate.html#blkid-get-tag-value> [↑](#footnote-ref-69)
70. For more information see <http://linux.die.net/man/8/blkid> [↑](#footnote-ref-70)
71. For more information see <http://www.computerhope.com/unix/udf.htm> [↑](#footnote-ref-71)
72. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> [↑](#footnote-ref-72)
73. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> [↑](#footnote-ref-73)
74. For more information see <http://linux.die.net/man/2/statfs> [↑](#footnote-ref-74)
75. For more information see <http://linux.die.net/man/2/statfs> [↑](#footnote-ref-75)
76. For more information see <http://linux.die.net/man/2/statfs> [↑](#footnote-ref-76)
77. For more information see <http://www.linfo.org/mount_point.html> [↑](#footnote-ref-77)
78. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> and <http://linux.die.net/man/1/df> [↑](#footnote-ref-78)
79. For more information see <http://linux.die.net/man/1/df> [↑](#footnote-ref-79)
80. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> [↑](#footnote-ref-80)
81. For more information see <http://linux.die.net/man/3/libblkid> [↑](#footnote-ref-81)
82. For more information see <http://www.kernel.org/pub/linux/utils/util-linux/v2.21/libblkid-docs/libblkid-Search-and-iterate.html#blkid-get-tag-value> [↑](#footnote-ref-82)
83. For more information see <http://linux.die.net/man/8/blkid> [↑](#footnote-ref-83)
84. For more information see <http://www.computerhope.com/unix/udf.htm> [↑](#footnote-ref-84)
85. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> [↑](#footnote-ref-85)
86. For more information see <http://www.kernel.org/doc/man-pages/online/pages/man5/proc.5.html> [↑](#footnote-ref-86)
87. For more information see <http://linux.die.net/man/2/statfs> [↑](#footnote-ref-87)
88. For more information see <http://linux.die.net/man/2/statfs> [↑](#footnote-ref-88)
89. For more information see <http://linux.die.net/man/2/statfs> [↑](#footnote-ref-89)
90. For more information see <http://linux.die.net/man/8/netstat> [↑](#footnote-ref-90)
91. For more information see <http://linux.die.net/man/8/netstat> [↑](#footnote-ref-91)
92. For more information see <http://linux.die.net/man/8/netstat> [↑](#footnote-ref-92)