Tivoli. *Endpoint Manager*

Version 8.1

Linux Client Inspector Guide

IBM

Note: Before using this information and the product it supports, read the information in Notices.

Contents

Part One

Introduction

The ***Tivoli Endpoint Manager Linux Client Inspector Library*** is a guide to the ordinary phrases (known as Inspectors) of the **Relevance Language**™. As the name implies, these phrases are used to inspect the properties of those Tivoli Endpoint Manager Clients that run the Linux OS. Thousands of Inspectors have been created to expose the inner workings of Linux computers, from the hardware and peripherals to the file system and software.

In addition to these client-specific Inspectors, there are several cross-platform Core Inspectors that are always available to you. These have been included in the keyword section at the end of this guide to provide you with a complete lexicon for Relevance scripting. For more information on the Core Inspectors, see the ***Tivoli Endpoint Manager Core Inspector Guide***. Note that the name of the program has changed from *BigFix* to *Tivoli Endpoint Manager*, however most Inspectors still use the BigFix name when referring to the program.

This guide may look imposing, but it reflects a certain amount of redundancy designed to improve accessibility. Each Inspector object has a creation method, listed by type. But objects are also properties of other objects (or the world), so they may be listed twice. In addition, the keyword section echoes the objects yet again, while adding type information and the plural format.

Inspectors can be thought of as object-oriented representations of the underlying computer system. They let you write Relevance expressions to query thousands of aspects of any Tivoli Endpoint Manager Client, instantly and with minimal overhead. Inspectors are keywords in the Relevance Language, so called because it allows content to be targeted to just those computers where it is relevant and no others. Relevance statements non-invasively analyze the Client computer to see if proper conditions exist before attempting remediation. Relevance Expressions are embedded into Action Scripts in such a way as to guarantee that the issue you detect is the one you remediate. In addition, Inspectors can be used to collect properties of any Tivoli Endpoint Manager Client for your own custom analysis in the Tivoli Endpoint Manager Console or Web Reports program.

Relevance and Action scripts are bundled with human-readable content into **Fixlet**® Messages, which can be further grouped into Fixlet Sites and Domains that specific subsets of your network can subscribe to as needed.

The bulk of these Inspectors are multi-platform, allowing one expression to address all the operating systems encountered in a typical network. So, although this guide is explicitly aimed at a single platform, most of these Inspectors have equivalents on other platforms as well. The list of Inspectors grows day by day, as need dictates. For each Inspector, this guide lists (by platform) the version of Tivoli Endpoint Manager where it first debuted.

For more information on how to write Relevance expressions, see the ***Tivoli Endpoint Manager Console Operator's Guide*** and the ***Tivoli Endpoint Manager Relevance Language Reference***.

Audience

This guide is for IT managers, product support groups and other people who want to use Inspectors to write Fixlet messages and Tasks for Linux-based Tivoli Endpoint Manager Clients.

IT managers use the **Tivoli Endpoint Manager** to keep their network of computers up to date and running smoothly without interruption. QA and other support teams can produce customized Fixlet messages to keep their users updated and their support calls to a minimum.

This document describes Inspectors for the Linux Operating System. Contact your Tivoli Endpoint Manager marketing representative for information about Inspector Guides for other operating systems, including Windows, Solaris, HPUX and Macintosh.

Conventions Used in this manual

This document makes use of the following conventions and nomenclature:

Convention	Use
`Mono-space`	A mono-spaced font is used to indicate expressions in the Relevance Language.
{curly braces}	Braces are used to indicate the comparison {=, !=} or arithmetic operators {+, -} that are available for a binary operation.
<angle bracket>	Angle brackets are used to indicate a type, such as string or integer, that is the object of a key phrase. When this document says 'absolute value of <integer>' it indicates that in practice, you will substitute an integer value, as in 'absolute value of 5'.
Italics	Indicates an Inspector *Form*. Some Inspectors are simple keywords. Others are a keyword in combination with another Inspector. Still other forms allow iteration through object lists. Each form is defined below
Small print	The small print beneath the description of each Inspector notes the version when it debuted on every relevant operating system (see the following section on Versions).

Examples

Square bullets and a mono-spaced font denote examples of Inspectors as used in a Relevance Expression. If you have a color version of this file, these square bullets are also red:

■ `concatenation of "light" & "year"`

▶ Returns "lightyear"

Versions

Prior to version 8.1, the program was known as **BigFix** or the **BigFix Enterprise Suite (BES)**. Although the name is now **Tivoli Enterprise Manager**, you will still find many legacy Inspectors that refer to BigFix or BES.

Most Inspectors have equivalent implementations on other operating systems, allowing you to write cross-platform relevance expressions. BigFix/Tivoli Endpoint Manager works across all major computer platforms, including the following:

> **Win:** the Windows version of the Tivoli Endpoint Manager Client and the Tivoli Endpoint Manager Session evaluation context..

> **Lin:** the Red Hat and SUSE Linux version of the Tivoli Endpoint Manager Client.

> **Sol:** the SUN Solaris operating system version of the Tivoli Endpoint Manager Client.

> **HPUX:** the Hewlett-Packard UNIX version of the Tivoli Endpoint Manager Client.

> **AIX:** the AIX version of the Tivoli Endpoint Manager Client.

> **Mac:** the Macintosh version of the Tivoli Endpoint Manager Client.

> **Ubu:** the Ubuntu / Debian version of the Tivoli Endpoint Manager Client.

> **WM:** the Windows Mobile version of the Tivoli Endpoint Manager Client.

There are exceptions, of course. Some of the Inspectors were introduced in later versions of the program, and won't work on all versions of all platforms. To keep track of them, the debut version is listed at the end of the Inspector description, for example:

> Win:2.0, Lin:3.1, Sol:7.1, HPUX:5.0, AIX:8.0, WM:7.2

This means that the Inspector of interest debuted in version 2.0 on Windows, but not until version 3.1 on Linux. In fact, version 3.1 of BigFix/Tivoli Endpoint Manager was the first version to include Linux Inspectors. Similarly, the first version for Windows Mobile was 7.2. The Inspector therefore exists on all versions of those two platforms, so the version number is unnecessary and we can simplify the list:

> Win:2.0, Lin, Sol:7.1, HPUX:5.0, AIX:8.0, WM

To further streamline this information, the version number is eliminated if it is less than or equal to version 6.0, which is a minimum requirement for most deployments. So the simplified version becomes:

> Win, Lin, Sol:7.1, HPUX, AIX:8.0, WM

Forms

You will notice that many of the keywords of the language are not unique; they get their meaning from their context. Accordingly, their definitions often include a phrase to define the context of each Inspector. In the following pages, you will find tables defining the Inspectors of the relevance language. The Inspectors come in several **forms** depending upon their context:

Form	Syntax	Example
Plain	keyword of <object>	address of ip interface
Plain Global	keyword	drives
Named	keyword "*name*" of <object>	variable "PATH" of environment
Named Global	keyword "*name*"	primary internet connection
Numbered	keyword *number* of <object>	line 5 of file "/usr/lib/foobar"
Numbered Global	keyword *number*	month 9
Index<(list)>	keyword (list) of <object>	substring (1,2) of "abcdefg"
Index<(list)> Global	keyword (list)	integers in (2,-1)
Binary Operator	<object> {op, cmp} <object>	December – current month
Unary Operator	{op} <object>	-month
Cast	<object> as keyword	"4.5" as floating point

These differ from one another in their format and the syntax they require. Except for cast, binary, and unary operators, these forms can be used to access both single objects and *lists* of objects by using the plural form of the keyword. The plurals are all listed in the keyword section at the end of this document.

In the following pages, each Inspector is described in terms of the **methods** that are used to create the Inspector object, the **properties** of the object that are available for inspection, the mathematical (binary and unary) **operations** that that can be performed on them, as well as **casting** options to convert the various types.

Part Two

Inspectors

World Objects

These are the plain, named, numbered or indexed global objects. This list is the subset of World objects that return primitive types, such as string, integer, boolean and time.

World

All objects created without context are known as 'properties of the world' in the relevance language. Below is a list of the primitive global properties, sorted by key phrase.

Properties

Key Phrase	Form	Return Type	Description
apparent registration server time	*PlainGlobal*	<time>	Shorthand for 'now of registration server'. When the client registers with the server, the server passes its current time back to the client. The client starts a stop watch at that time. The apparent registration server time is the time the server passed back to the client, plus the elapsed time on the stop watch. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
computer id	*PlainGlobal*	<integer>	This is a unique integer assigned to the computer by the BES system. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
computer name	*PlainGlobal*	<string>	Returns a string corresponding to the name of the computer as it appears on the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
custom site subscription effective date <string>	*NamedGlobal*	<time>	Returns the date the custom site (specified by <string>) was last subscribed or unsubscribed. It is used internally by BES to manage custom site subscriptions. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
dns name	*PlainGlobal*	<string>	Returns the DNS name of the computer. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
domain name	*PlainGlobal*	<string>	Returns the fully qualified domain name of the machine. Lin, Sol, HPUX, AIX, Ubu

Key Phrase	*Form*	Return Type	Description
domainname	*PlainGlobal*	\<string\>	Same as domain name. Lin, Sol, HPUX, AIX, Ubu
download path \<string\>	*NamedGlobal*	\<string\>	This inspector is available in relevance subsitution action processing. It returns a string corresponding to the download path of the specified file. This Inspector (along with download folder and download file) is designed to be used during the prefetch process of action execution. This is equivalent to '(pathname of download folder) & pathseparator & "myfile"'. Win:7.2, Lin:7.2, Sol:7.2, HPUX:7.2, AIX:7.2, Mac:7.2, WM, Ubu
host name	*PlainGlobal*	\<string\>	Returns the machine name (the same as the computer name or hostname on UNIX machines). Lin, Sol, HPUX, AIX, Ubu
hostname	*PlainGlobal*	\<string\>	Returns the standard host name, usually for the computer's network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
last relay select time	*PlainGlobal*	\<time\>	Returns the time when last relay selection took place. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
parameter \<string\>	*NamedGlobal*	\<string\>	This Inspector is a synonym for the parameter \<string\> of \<action\>. It looks up the value of the action parameter specified by \<string\>. This is used in conjunction with the parameter set command. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
pending login	*PlainGlobal*	\<boolean\>	Installers may leave values in the registry that the operating system will execute when the next user logs in. Pending login can detect these registry entries. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pending restart	*PlainGlobal*	\<boolean\>	Returns TRUE if the operating system indicates that a restart needs to occur. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pending restart \<string\>	*NamedGlobal*	\<boolean\>	Immediately after issuing a command like 'Action requires restart "PatchGroupX"', the expression 'Pending restart "PatchGroupX"' will be true until the next restart. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
pending restart name	*PlainGlobal*	<string>	This iterated Inspector returns the names of currently pending restarts. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
system language	*PlainGlobal*	<string>	Returns the language of the system as a string. It is identified using the GetSystemDefaultLangID() system call. See the language keyword of the application object for a list of possible language value. Win, Lin, Sol, HPUX, AIX, WM, Ubu
wake on lan subnet cidr string	*PlainGlobal*	<string>	Returns the subnet the client is in for Wake on Lan (WoL) purposes. The client sends information to the relay during registration that is used to decide which subnet the client is in. The relay returns the subnet to the client, which is the value this Inspector exposes. This value is used to send WoL commands to forwarders. To wake a machine by computer ID, the server looks up the mac address and subnet of that machine. It then tries to identify clients that have been configured as WoL forwarders within the same subnet and routes WoL commands to those forwarders, sending them the mac address of the machine that needs to be awoken. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Examples

- `wait "{download path"update.exe"}"`

▶ In an Action script, this line causes the BES Client to perform relevance substitution to compute the full path to the downloaded file (previously collected by a download command in the same Action script). After relevance substitution, the Client launches the specified executable and waits for it to complete before moving on to other Action lines.

- `host name`

▶ Returns a string like "localhost.localdomain" or "user.bigcorp".

Filesystem Objects

This chapter covers the keywords for extracting information from the file system, like files, drives, pathnames, folders, etc. It also includes the keywords needed to identify and compare version information of files and patches. For more information on special Windows folders, see the Resources section at the end of this guide.

Filesystem

The filesystem object can be used to inspect various aspects of mounted file systems, including the format of the file system. Here are some of the possible format types:
- affs
- ext, ext2, ext2_old
- hpfs
- iso
- minix, minix_30, minix2, minix2_30
- msdos
- ncp
- nfs
- proc
- smb
- xenix
- sysv4, sysv2
- coh
- ufs
- xia

Creation Methods

Key Phrase	Form	Description
drive	*PlainGlobal*	Iterates through all valid drives on the system. Typically used to return a list of the drives (volumes, filesystems) on the client computer. • On Windows computers, this returns a <drive> object. • On *nic computers, this returns a <filesystem> object. • Drives, volumes and filesystems are treated the same on the Macintosh and return a <volume> type. Lin, Sol, HPUX, AIX, , WM, Ubu

Key Phrase	Form	Description
drive <string>	*NamedGlobal*	Returns the drive associated with the pathname specified by <string>. • On Windows computers, this returns a <drive> object. • On Macintosh computers, this returns a <volume> object. • On *nix computers, this returns a <filesystem> object. Lin, Sol, HPUX, AIX, , WM, Ubu
drive of <device file>	*Plain*	Returns the drive associated with the specified device file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
drive of <fifo file>	*Plain*	Returns the drive associated with the specified FIFO (named pipe) file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
drive of <file>	*Plain*	Returns the drive associated with the specified file. • On Macintosh computers, this returns a <volume> object. • On *nix computers, this returns a <filesystem> object. Lin, Sol, HPUX, AIX, , Ubu
drive of <folder>	*Plain*	Returns the drive associated with the specified folder. • On Macintosh computers, this returns a <volume> object. • On *nix computers, this returns a <filesystem> object. Lin, Sol, HPUX, AIX, , Ubu
drive of <socket file>	*Plain*	Returns the drive associated with the specified socket file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
drive of <symlink>	*Plain*	Returns the drive associated with the specified symlink as a <filesystem> object. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
filesystem	*PlainGlobal*	Returns <filesystem> objects for all currently mounted file systems. Lin, Sol, HPUX, AIX, Ubu
filesystem <string>	*NamedGlobal*	Returns the filesystem object for the name provided. Lin, Sol, HPUX, AIX, Ubu
filesystem of <device file>	*Plain*	Returns the filesystem object corresponding to the specified device file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Key Phrase	*Form*	Description
filesystem of <fifo file>	*Plain*	Returns the filesystem object corresponding to the specified FIFO file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
filesystem of <file>	*Plain*	Returns the filesystem on which the file resides. Lin, Sol, HPUX, AIX, Ubu
filesystem of <folder>	*Plain*	Returns the filesystem on which the folder resides. Lin, Sol, HPUX, AIX, Ubu
filesystem of <socket file>	*Plain*	Returns the filesystem object corresponding to the specified socket file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
filesystem of <symlink>	*Plain*	Returns the filesystem on which the symlink resides. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
file count of <filesystem>	*Plain*	<integer>	Returns the total number of files that may be saved on this filesystem. Lin, Sol, HPUX, AIX, Ubu
free file count of <filesystem>	*Plain*	<integer>	Returns the number of files available on this filesystem. Lin, Sol, HPUX, AIX, Ubu
free percent of <filesystem>	*Plain*	<integer>	Returns the percentage of the file system currently available. Lin, Sol, HPUX, AIX, Ubu
free space of <filesystem>	*Plain*	<integer>	Returns the number of bytes on this filesystem. Lin, Sol, HPUX, AIX, Ubu
name of <filesystem>	*Plain*	<string>	Returns the mount point of the filesystem object. Lin, Sol, HPUX, AIX, Ubu
size of <filesystem>	*Plain*	<integer>	Returns the total number of bytes on this file system (same as total space). Lin, Sol, HPUX, AIX, Ubu
total space of <filesystem>	*Plain*	<integer>	Returns the total number of bytes on this file system (same as size). Lin, Sol, HPUX, AIX, Ubu

Key Phrase	Form	Return Type	Description
type of <filesystem>	*Plain*	<string>	Returns the filesystem format type, as listed in the type header. Lin, Sol, HPUX, AIX, Ubu
used file count of <filesystem>	*Plain*	<integer>	The number of files in use on this filesystem. Lin, Sol, HPUX, AIX, Ubu
used percent of <filesystem>	*Plain*	<integer>	Returns the percentage of the file system currently in use. Lin, Sol, HPUX, AIX, Ubu
used space of <filesystem>	*Plain*	<integer>	Returns the number of bytes on this filesystem currently in use. Lin, Sol, HPUX, AIX, Ubu

Examples

- ```
  names of drives
  ```
▶ Returns the names of the mounted drives.

- ```
  names of filesystems
  ```
▶ Returns the mount points of the file systems.

- ```
  (name of it, type of it) of filesystems
  ```
▶ Returns the names and types of all mounted file systems.

Filesystem Object

Properties

Key Phrase	Form	Return Type	Description
<filesystem object> as device file	*Cast*	<device file>	Returns a device file or nothing (if the filesystem object specified is not a device file). Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
<filesystem object> as fifo file	*Cast*	<fifo file>	Returns a FIFO file or nothing (if the filesystem object specified is not a FIFO file). Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
<filesystem object> as socket file	*Cast*	<socket file>	Casts a given filesystem object as a socket file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
<filesystem object> as string	*Cast*	<string>	Casts a filesystem object as a string. Win:8.0, Lin, Sol, HPUX, AIX, Mac, Ubu

Key Phrase	Form	Return Type	Description
<filesystem object> as symlink	*Cast*	<symlink>	Casts a link in the form of a file into a symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
accessed time of <filesystem object>	*Plain*	<time>	When the filesystem object (file or folder) was last accessed. Some file systems maintain this property. Win, Lin, Sol, HPUX, AIX, WM, Ubu
ancestor of <filesystem object>	*Plain*	<folder>	Returns all ancestor folders (recursive parent folders) of the given filesystem object (file or folder). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
gid of <filesystem object>	*Plain*	<integer>	Returns the group ID of the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
group execute of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the group execute flag is set for the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
group mask of <filesystem object>	*Plain*	<integer>	Returns the group permission mask of the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
group name of <filesystem object>	*Plain*	<string>	Returns the group name of the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
group read of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the group read flag is set for the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
group write of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the group write flag is set for the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
link count of <filesystem object>	*Plain*	<integer>	Returns an integer corresponding to the number of hard links attached to the specified filesystem object. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
location of <filesystem object>	*Plain*	<string>	Returns the name of the directory in which the file or folder (filesystem object) is located. Win, Lin, Sol, HPUX, AIX, WM, Ubu
mode of <filesystem object>	*Plain*	<mode>	Returns the permissions mode for the given filesystem object. Lin, Sol, HPUX, AIX, Ubu

Key Phrase	*Form*	Return Type	Description
modification time of <filesystem object>	*Plain*	<time>	The date and time of latest modification of the file. This corresponds to what is shown in the "Get Info" box. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
name of <filesystem object>	*Plain*	<string>	This returns the name of the file or folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
other execute of <filesystem object>	*Plain*	<boolean>	Returns TRUE if others (not in the group) have execute permissions on the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
other mask of <filesystem object>	*Plain*	<integer>	Returns the other (not in the group) mask as a 3-bit integer corresponding to rwx permissions for the specified filesystem object. Lin, Sol, HPUX, AIX, Ubu
other read of <filesystem object>	*Plain*	<boolean>	Returns TRUE if others (not in the group) have read permissions on the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
other write of <filesystem object>	*Plain*	<boolean>	Returns TRUE if others (not in the group) have write permissions on the given filesystem object. Lin, Sol, HPUX, AIX, Ubu
parent folder of <filesystem object>	*Plain*	<folder>	The folder containing the specified file or folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pathname of <filesystem object>	*Plain*	<string>	Returns the full pathname of the specified file or folder (filesystem object) as a string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
setgid of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the setgid (group ID) bit is set for the specified filesystem object. Lin, Sol, HPUX, AIX, Ubu
setuid of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the setuid (user ID) bit is set for the specified filesystem object. Lin, Sol, HPUX, AIX, Ubu
uid of <filesystem object>	*Plain*	<integer>	The user ID of the user who owns this filesystem object. Lin, Sol, HPUX, AIX, Ubu
user execute of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the owner (user) has execute permissions on the given filesystem object. Lin, Sol, HPUX, AIX, Ubu

Key Phrase	*Form*	Return Type	Description
user mask of <filesystem object>	*Plain*	<integer>	Integer representing user permissions (3 bit mask, RWX) on the specified filesystem object. Lin, Sol, HPUX, AIX, Ubu
user name of <filesystem object>	*Plain*	<string>	Returns the owner (user) name of the specified filesystem object. Lin, Sol, HPUX, AIX, Ubu
user read of <filesystem object>	*Plain*	<boolean>	Returns the owner (user) read permissions for the specified filesystem object. Lin, Sol, HPUX, AIX, Ubu
user write of <filesystem object>	*Plain*	<boolean>	Returns the owner (user) write permissions for the specified filesystem object. Lin, Sol, HPUX, AIX, Ubu

Examples

- `file "/example/link" as symlink`
  - Returns the link (in file format) as a symlink object ("/example/link").

File

For each file in the file system, you can create a corresponding file object and inspect its properties. Inspectors are also provided to look at version data of executable files.

• NOTE: File systems that do not maintain the creation or last accessed times will often return the last modification time when queried for the creation or last accessed times or files. Modification times are preserved when files are copied. Thus, it is not uncommon to see a file that appears to have been modified before it was created.

Type Derivation: This object type is derived from the <filesystem object> type and therefore shares the same properties as that type.

CAUTION: Some file content Inspectors can cause contention issues with other applications, regardless of the platform. These Inspectors open up the file for read access with maximal sharing with other applications. However, if other applications try to access the file with exclusive rights, they will fail. The set of Inspectors that hold a handle to the file are: 'lines of file', 'contents of file' and 'sha1 of file'.

Creation Methods

See application objects for additional creation methods

Key Phrase	Form	Description
<symlink> as file	*Cast*	Converts a symlink object into the file it points to. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
default web browser	*PlainGlobal*	Creates an object corresponding to the default web browser. This is a Windows and Macintosh inspector; it will fail gracefully on other operating systems instead of generating an error. • Note: This Inspector returns a <file> object on UNIX, an <application> on Windows and a <filesystem object> on the Mac. Lin, Sol, HPUX, AIX, Ubu
descendant of <folder>	*Plain*	Returns a list of all the descendant files of the specified folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
download file <string>	*NamedGlobal*	This inspector is available in relevance subsitution action processing. It returns a file object with the given name from the named folder or the download folder. This is equivalent to 'file "name" of download folder'. The file should exist or the result will not exist. Win:7.2, Lin:7.2, Sol:7.2, HPUX:7.2, AIX:7.2, Mac:7.2, WM, Ubu
file <string>	*NamedGlobal*	Returns a filesystem object corresponding to the full pathname provided in <string>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Description
file \<string> of \<folder>	*Named*	Creates the file objects corresponding to the named file within the folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
file \<symlink>	*Index\<symlink>Global*	Returns the file pointed to by the specified symlink. If the file doesn't exist, this Inspector will throw a 'non-existent object' error. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
file of \<folder>	*Plain*	Iterates through the files of a folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
find file \<string> of \<folder>	*Named*	Creates an object corresponding to the files of the folder that that match the wildcard \<string> provided. A wildcard string uses an asterisk to stand for any number of characters (incuding zero), and a question mark to stand for exactly one character. Thus A??.txt would match All.txt and AXE.txt but not all.txt or a.txt. Win, Lin, Sol, HPUX, AIX, Mac:8.0, WM, Ubu
masthead of \<site>	*Plain*	A copy of the masthead is maintained with the site data. This inspector returns a file object for the copy. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
byte \<integer> of \<file>	*Numbered*	\<integer>	Returns the numeric value of the byte located at the offset specified by number within the file. Byte 0 of the file is the first byte. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
content of \<file>	*Plain*	\<file content>	Returns an object that can be used to search for a string in the file. See content. CAUTION: This Inspector maintains a handle to the specified file, so during its operation it may block any other applications that attempt to open the file. Inspectors open files as with both read and write sharing, so apps that open with compatibleaccess will not block. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
drive of \<file>	*Plain*	\<filesystem>	Returns the drive associated with the specified file as a \<filesystem> object. Lin, Sol, HPUX, AIX, Ubu

Key Phrase	Form	Return Type	Description
filesystem of <file>	*Plain*	<filesystem>	Returns the UNIX filesystem flag for the given file. Lin, Sol, HPUX, AIX, Ubu
key <string> of <file>	*Named*	<string>	Returns a key and its value from the given structured text file. It iterates over lines that start with the key name (as specified by <string>) followed by an = or : character. When searching, white space is ignored. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
line <integer> of <file>	*Numbered*	<file line>	Returns the nth line (specified by <integer>) from the given file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line containing <string> of <file>	*Named*	<file line>	Returns all lines from the given file that contain the specified string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line of <file>	*Plain*	<file line>	Iterates over all the lines of the specified file. NOTE: lines are truncated to 1023 characters. CAUTION: This Inspector maintains a handle to the specified file, so during its operation it may block any other applications that attempt to open the file. Inspectors open files as with both read and write sharing, so apps that open with compatibleaccess will not block. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line starting with <string> of <file>	*Named*	<file line>	Same as line <string> of <file>, returns the lines of the given file that start with the specified string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
section <string> of <file>	*Named*	<file section>	Returns a named section of a file. Useful for locating sections of 'ini' files. Section names are delimited by square bracket characters '[section name]'. See examples below. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
sha1 of <file>	*Plain*	<string>	Returns the sha1 checksum of the file hex encoded as a 40 character long string. CAUTION: This Inspector maintains a handle to the specified file, so during its operation it may block any other applications that attempt to open the file. Inspectors open files as with both read and write sharing, so apps that open with compatibleaccess will not block. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
size of <file>	*Plain*	<integer>	Returns the size in bytes of a file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
variable of <file>	*Plain*	<string>	Returns the names of variables contained in an INF style file, in the format [section].name=value. Win, Lin, Sol, HPUX, AIX, WM, Ubu

NOTE:

Folder and file names may be case sensitive. Use "as uppercase" or "as lowercase" if you don't know the actual case when making comparisons. Iterating through folders with many files can be time consuming.

NOTE:

The format of the string returned when casting a file using 'as string' is:
"<fileName>" "<version>" "<fileDesc>" "<fileVersion>" "<companyName>"
Where:

<fileName>	The name of the file
<version>	The 'Product Version' of the file.
<fileDesc>	The value 'FileDescription' of version block 1 of the file.
<fileVersion>	The value 'FileVersion' of version block 1 of the file.
<companyName>	The value 'CompanyName' of version block 1 of the file.

Examples

- ```
  wait "{pathname of download file "update.exe"}"
  ```
▶ In an Action script, this line causes the BES Client to perform relevance substitution to compute the full path to the downloaded file (previously collected by a download command in the same Action script). After relevance substitution, the Client launches the specified executable and waits for it to complete before moving on to other Action lines.

- ```
  Number of find files "siteico*.bmp" of client folder of current site = 3
  ```
▶ Returns TRUE if there are 3 files matching the wildcard pattern siteico*.bmp.

- ```
  modification time of masthead of current site < time "4 Aug 1997 01:00 pdt"
  ```
▶ TRUE if the masthead is older than the specified date.

Application

Application objects derive from file objects. Therefore, application objects inherit all of the properties of the file object. This means that you can inspect properties such as 'modification time' or 'Product Version' of an application just as you would a file. See the properties of a file object for a complete list of these. The real power and primary purpose of the application object is their creation. The creation methods are optimized in anticipation of the importance of this object.

- NOTE: See 'File' for a list of the Application properties. Folder and file names may be case sensitive. Use "as uppercase" or "as lowercase" if you don't know the actual case when making comparisons.
- For Windows versions of these Inspectors, a properly installed application will register itself with the operating system. It does this by creating a registry sub-key usually named after the executable. The regapp Inspector uses the default value of this sub-key as a string that is the full pathname of the executable.

Type Derivation: This object type is derived from the <file> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
application <string>	*NamedGlobal*	Creates an application object for the name provided. Win, Lin, Sol, HPUX, AIX, WM, Ubu
application <string> of <folder>	*Named*	As with the file object, you can create an application object by naming it relative to its parent folder. Win, Lin, Sol, HPUX, AIX, WM, Ubu
recent application	*PlainGlobal*	Iterates through the list of recently executed applications, creating application objects. This includes the list of all currently running applications. Win, Lin, Sol:8.1, WM, Ubu
recent application <string>	*NamedGlobal*	If named application has been executed recently, this inspector creates an application object. Only specify the last component of the filename. Win, Lin, Sol:8.1, WM, Ubu
running application	*PlainGlobal*	Iterates through the list of running applications. Win, Lin, Sol:8.1, WM, Ubu
running application <string>	*NamedGlobal*	If the named application is currently executing then this inspector creates an application object. Only specify the last component of the file name. Win, Lin, Sol:8.1, WM, Ubu

Examples

- ```
  exists application "notepad.exe" of the windows folder
  ```
  ▶ Using the application of folder creation method, this example locates the notepad application provided it exists in the Windows folder.

- ```
  exists running application whose (name of it as lowercase is "winword.exe")
  ```
  ▶ Returns TRUE if Microsoft Word is currently executing.

Folder

For every folder that exists in the file system, you can create a folder object. These Inspectors allow you to examine dozens of properties of folder objects. On the Macintosh, there are dozens of specialized folders; access to them depends on the domain. If the domain is not specified, it defaults to the system domain.

Type Derivation: This object type is derived from the <filesystem object> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
<symlink> as folder	Cast	Converts a symlink object into the folder it points to. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
ancestor of <filesystem object>	Plain	Returns all ancestor folders (recursive parent folders) of the given filesystem object (file or folder). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
ancestor of <symlink>	Plain	Returns all ancestor folders (recursive parent folders) of the given symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
client folder of <site>	Plain	Creates an object corresponding to the folder on the client where site data is gathered. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
descendant folder of <folder>	Plain	Returns the descendant folders, recursively, of the given folder. The folder equivalent of "descendants of <folder>". Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
find folder <string> of <folder>	Named	Finds the folder with the given wildcard name inside another folder. A wildcard string uses an asterisk to stand for any number of characters (incuding zero), and a question mark to stand for exactly one character. Thus A??.txt would match All.txt and AXE.txt but not all.txt or a.txt. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Key Phrase	Form	Description
folder <string>	*NamedGlobal*	Creates a folder object for the named folder. This is a global property. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
folder <string> of <folder>	*Named*	Creates a folder object for the named sub-folder. Trailing slashes should be omitted from the name. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
folder <symlink>	*Index<symlink>Global*	Returns the folder pointed to by the specified symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
folder of <folder>	*Plain*	Iterates through the sub-folders of the folder object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
parent folder of <filesystem object>	*Plain*	The folder containing the specified file or folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
parent folder of <symlink>	*Plain*	Creates a folder object corresponding to the parent folder of the given symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
root folder	*PlainGlobal*	Returns the folder corresponding to '/'. Lin, Sol, HPUX, AIX, Ubu

Properties

Key Phrase	Form	Return Type	Description
application <string> of <folder>	*Named*	<application>	Returns an application object for the named file located in the folder. See application. Win, Lin, Sol, HPUX, AIX, WM, Ubu
descendant folder of <folder>	*Plain*	<folder>	Returns the descendant folders, recursively, of the given folder. The folder equivalent of "descendants of <folder>". Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
descendant of <folder>	*Plain*	<file>	Returns a list of all the descendant files of the specified folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
device file <string> of <folder>	*Named*	<device file>	Returns the device file with the specified name in the given folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
device file of <folder>	*Plain*	<device file>	Returns the device files that exist in the specified folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Key Phrase	Form	Return Type	Description
drive of <folder>	*Plain*	<filesystem>	Creates the drive object corresponding to the folder location. Included for compatibility with Windows machines, this Inspector is the same as filesystem of <folder>. Lin, Sol, HPUX, AIX, Ubu
fifo file <string> of <folder>	*Named*	<fifo file>	Returns the FIFO file (named pipe) with the given name in the specified folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
fifo file of <folder>	*Plain*	<fifo file>	Returns the list of FIFO file in the given folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
file <string> of <folder>	*Named*	<file>	Returns a file object for the named file located in the folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
file of <folder>	*Plain*	<file>	Iterates through the files of a folder returning file objects. When combined with a whose clause you can select files with specific properties. See file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
filesystem of <folder>	*Plain*	<filesystem>	Returns the filesystem on which the folder resides. Lin, Sol, HPUX, AIX, Ubu
find file <string> of <folder>	*Named*	<file>	Iterates through the files of a folder returning file objects whose name matches the wildcard string provided in the name parameter. A wildcard string uses an asterisk to stand for any number of characters (incuding zero), and a question mark to stand for exactly one character. Thus A??.txt would match All.txt and AXE.txt but not all.txt or a.txt. See example below. Win, Lin, Sol, HPUX, AIX, Mac:8.0, WM, Ubu
find folder <string> of <folder>	*Named*	<folder>	Finds the folder with the given wildcard name inside another folder. A wildcard string uses an asterisk to stand for any number of characters (incuding zero), and a question mark to stand for exactly one character. Thus A??.txt would match All.txt and AXE.txt but not all.txt or a.txt. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
folder <string> of <folder>	*Named*	<folder>	Returns a folder object for the named sub-folder. Trailing slashes should be omitted from the name. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
folder of <folder>	*Plain*	<folder>	Iterates through the folders of a folder returning folder objects. When combined with a whose clause, you can select folders with specific properties. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
socket file <string> of <folder>	*Named*	<socket file>	Returns the socket file with the given name in the specified folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
socket file of <folder>	*Plain*	<socket file>	Returns the socket file(s) in the specified folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
symlink <string> of <folder>	*Named*	<symlink>	Returns the named symlink from the specified folder. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
symlink of <folder>	*Plain*	<symlink>	Returns the symlink(s) in the specified folder, whether or not they are broken. • NOTE: This behavior differs from looking for files in a folder. Although that returns links along with the files, it doesn't return broken links. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

NOTE: Folder and file names may be case sensitive. Use "as uppercase" or "as lowercase" if you don't know the actual case when making comparisons. Be careful not to iterate through folders that contain a large number of files. Counting files in such a folder can be slow. Always try to use the most efficient techniques to minimize the client overhead. Consider using the "find file" Inspector which allows you to filter sets of files by using the wildcard.

File Section

Many programs and utilities store their settings in 'ini' files. This object is designed to access these settings. An 'ini' file is composed of zero or more named sections, each with zero or more keys. Each key is identified by name and has a string value.

Creation Methods

Key Phrase	Form	Description
section <string> of <file>	*Named*	Creates a file section for the name given. A case-insensitive search is performed to locate the named section in the file. Searching through files for configuration data can be a slow process. Particularly for large ini files. In this case you may want to find another method that requires less computation. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

Key Phrase	Form	Return Type	Description
key <string> of <file section>	*Named*	<string>	Returns a string containing the value for the name provided. A case-insensitive search is performed through the section of the file. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

NOTE: Files with an extension of .ini are common in Windows systems, but rare in unix systems. However they are a handy cross-platform way of maintaining a collection of named variables.

File Content

Content objects can be constructed from file objects to inspect their contents.

Creation Methods

Key Phrase	Form	Description
<file content> as lowercase	*Cast*	Returns the contents of the file as lower case characters. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<file content> as uppercase	*Cast*	Returns the contents of the file as upper case characters. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
content of <file>	*Plain*	Creates a content object for a file. CAUTION: This Inspector maintains a handle to the specified file, so during its operation it may block any other applications that attempt to open the file. Inspectors open files as with both read and write sharing, so apps that open with compatibleaccess will not block. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
<file content> as lowercase	*Cast*	<file content>	Returns a lowercase version of the content provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<file content> as uppercase	*Cast*	<file content>	Returns an uppercase version of the content provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Operators

Key phrase	Return Type	Description
<file content> contains <string>	*<boolean>*	Returns TRUE if the string is located in the content provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

NOTE: See "file section" for a more detailed inspection of .ini files.

Examples

- `content of file "oeminfo.ini" of system folder as lowercase contains "emachines"`
  - ▶ Returns TRUE if either of the strings "emachines" or "eMachines" is found in the file.

Version

This is the numeric method of indicating the file version, which is compact, convenient and fast. It makes use of a short string to define the version number. Version types are available as both client and core Inspectors, so if you don't find what you want in one guide, please check the other.

Creation Methods

Key Phrase	*Form*	Description
version of <application usage summary instance>	*Plain*	Returns the version of the specified application instance. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
version of <client>	*Plain*	The product version of the BES application (BESClient or QnA). Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
version of <current relay>	*Plain*	Returns a version object that is the version of the server or relay that the client last registered with. This may be a BES Relay or the BES root server. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
version of <package>	*Plain*	Creates an object containing the version of the specified RPM package. Lin
version of <service>	*Plain*	This Inspector takes the specified service property and retrieves its version (file version). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

NOTE: Using the numeric version data is better than identifying an application based on version block strings. If you know the numeric version information and that the developer has identified each release of his application uniquely, then this is the way to proceed. It requires far less overhead than the other

method. Furthermore, if you know that the numeric version data is monotonically increasing then you can compare their values using the special comparison operators.

Examples

- `version of client as string`
▶ Returns a string like "4.0.3.7".

Strverscmp Version

Uses the OS supplied strverscmp function, which may be useful for certain older-style version comparisons. For more information, see the man page for strverscmp.

Type Derivation: This object type is derived from the <version> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	*Form*	Description
<string> as strverscmp version	*Cast*	Converts a string to strverscmp version, which is compared to other versions using the OS supplied strverscmp function. Lin, Ubu
strverscmp version <string>	*NamedGlobal*	Creates a strverscmp version, which is compared to other versions using the OS supplied strverscmp function. Lin, Ubu

Operators

Key phrase	Return Type	Description
<string> {cmp} <strverscmp version>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison (based on the OS supplied strverscmp function), where: • {cmp} is one of: =, !=, <, <=, >, >= . Lin, WM
<strverscmp version> {cmp} <strverscmp version>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison (based on the OS supplied strverscmp function), where: • {cmp} is one of: =, !=, <, <=, >, >= . Lin

Mode

The mode inpector returns file type information and permissions. These are the possible values of mode:
- S_IFMT 170000 bitmask for the file type bitfields
- S_IFSOCK 140000 socket
- S_IFLNK 120000 symbolic link
- S_IFREG 100000 regular file
- S_IFBLK 060000 block device
- S_IFDIR 040000 directory
- S_IFCHR 020000 character device
- S_IFIFO 010000 fifo
- S_ISUID 004000 set UID bit
- S_ISGID 002000 set GID bit
- S_ISVTX 001000 sticky bit
- S_IRWXU 000700 mask for file owner permissions
- S_IRUSR 000400 owner has read permission
- S_IWUSR 000200 owner has write permission
- S_IXUSR 000100 owner has execute permission
- S_IRWXG 000070 mask for group permissions
- S_IRGRP 000040 group has read permission
- S_IWGRP 000020 group has write permission
- S_IXGRP 000010 group has execute permission
- S_IRWXO 000007 mask for permissions for others (not in group)
- S_IROTH 000004 others have read permission
- S_IWOTH 000002 others have write permission
- S_IXOTH 000001 others have execute permission

Creation Methods

Key Phrase	Form	Description
mode of <filesystem object>	Plain	Returns the permissions mode for the given filesystem object. Lin, Sol, HPUX, AIX, Ubu

Properties

Key Phrase	Form	Return Type	Description
<mode> as octal string	Cast	<string>	Converts the mode to a string of octal numbers. Lin, Sol, HPUX, AIX, Ubu
<mode> as string	Cast	<string>	Converts the mode to a string. Lin, Sol, HPUX, AIX, Ubu

Key Phrase	Form	Return Type	Description
group mask of <mode>	*Plain*	<mode_mask>	Returns the mask for group permissions for the given mode. Lin, Sol, HPUX, AIX, Ubu
other mask of <mode>	*Plain*	<mode_mask>	Returns the mask for permissions for others (those not in the group) for the given mode. Lin, Sol, HPUX, AIX, Ubu
setgid of <mode>	*Plain*	<boolean>	Returns TRUE if setgid (the group ID flag) is set. Lin, Sol, HPUX, AIX, Ubu
setuid of <mode>	*Plain*	<boolean>	Returns TRUE if setuid (the user ID flag) is set. Lin, Sol, HPUX, AIX, Ubu
sticky of <mode>	*Plain*	<boolean>	The `sticky' bit (S_ISVTX) on a directory means that a file in that directory can be renamed or deleted only by the owner of the file, by the owner of the directory, and by root. This inspector returns TRUE if the sticky bit of the given mode is on. Lin, Sol, HPUX, AIX, Ubu
user mask of <mode>	*Plain*	<mode_mask>	Returns the mask for the user (file owner) permissions for the given mode. Lin, Sol, HPUX, AIX, Ubu

Mode_mask

The mode_mask Inspector is a differently formatted version of the mode, created by shifting the key information down to the low three bits.

Creation Methods

Key Phrase	Form	Description
group mask of <mode>	*Plain*	The rwx mask (shifted to the lower 3 bits) for group permissions for the given mode. Lin, Sol, HPUX, AIX, Ubu
other mask of <mode>	*Plain*	The rwx mask (shifted to the lower 3 bits) for permissions for others (those not in the group) for the given mode. Lin, Sol, HPUX, AIX, Ubu
user mask of <mode>	*Plain*	The rwx mask (shifted to the lower 3 bits) for permissions for the user (the file owner) for the given mode. Lin, Sol, HPUX, AIX, Ubu

Properties

Key Phrase	Form	Return Type	Description
<mode_mask> as integer	*Cast*	<integer>	Converts the mode mask to an integer, 1-4. Lin, Sol, HPUX, AIX, Ubu
<mode_mask> as string	*Cast*	<string>	Converts the mode mask to a string, for example "rwx". Lin, Sol, HPUX, AIX, Ubu
execute of <mode_mask>	*Plain*	<boolean>	Returns TRUE if the execute flag (x) of the rwx mode mask is on. (binary 001 = 1). Lin, Sol, HPUX, AIX, Ubu
read of <mode_mask>	*Plain*	<boolean>	Returns TRUE if the read flag of the rwx mode mask is on. (binary 100 = 4). Lin, Sol, HPUX, AIX, Ubu
write of <mode_mask>	*Plain*	<boolean>	Returns TRUE if the write flag (w) of the rwx mode mask is on. (binary 010 = 2). Lin, Sol, HPUX, AIX, Ubu

File Line

A file line is a string from a text file.

Type Derivation: This object type is derived from the <string> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
line <integer> of <file>	*Numbered*	Returns the nth line in a file. A file line is just a string, except that you can use the additional properties "next line" and "previous line". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line containing <string> of <file>	*Named*	Returns the line with the specified search string in the given file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line of <file>	*Plain*	Returns the lines of a specified file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line starting with <string> of <file>	*Named*	Returns a line from the given file beginning with the specified phrase. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Description
next line of <file line>	*Plain*	Returns the line after the specified line in a file (provided that it is not the last line). This Inspector can be chained indefinitely, eg., next line of next line of …. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
previous line of <file line>	*Plain*	Returns the line before the nth line in a file, provided n>1. You may repeat this command up to three times. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
line number of <file line>	*Plain*	<integer>	Returns the line number of a given line. Can be used to locate specific lines in a file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
next line of <file line>	*Plain*	<file line>	Returns the line after the specified line in a file (provided that it is not the last line). This Inspector can be chained indefinitely, eg., next line of next line of …. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
previous line of <file line>	*Plain*	<file line>	Returns the line before the nth line in a file, provided n>1. You may repeat this command up to three times. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Symlink

Symlinks, or symbolic links, are the Unix version of shortcut files (pointers to other files). Inspectors can analyze the file objects (files and folders) that symlinks point to by using the standard file Inspectors. The symlink Inspectors, on the other hand, allow you to analyze the properties of a symlink itself, not just the underlying file. There are many properties that symlinks have in common with files, such as name, pathname, parent folder, and more. In addition, a symlink has a value corresponding to the file object it points to. You can also determine if the file is available or not.

Creation Methods

Key Phrase	Form	Description
<filesystem object> as symlink	*Cast*	Casts a link in the form of a file into a symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

Key Phrase	Form	Description
<symlink> as symlink	*Cast*	Casts a symlink, provided for completeness. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
symlink <filesystem object>	*Index<filesystem object>Global*	Creates a symlink from the specified filesystem object. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
symlink <string>	*NamedGlobal*	Creates a symlink from the specified string. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
symlink <string> of <folder>	*Named*	Returns the named symlink from the specified folder. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
symlink <symlink>	*Index<symlink>Global*	Creates a symlink from the specified symlink path, even if the symlink is broken. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
symlink of <folder>	*Plain*	Returns the symlink(s) in the specified folder, whether or not they are broken. • NOTE: This behavior differs from looking for files in a folder. Although that returns links along with the files, it doesn't return broken links. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
<symlink> as device file	*Cast*	<device file>	Casts a symlink type as a device file type. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
<symlink> as fifo file	*Cast*	<fifo file>	Casts a symlink type as a FIFO (named pipe) file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
<symlink> as file	*Cast*	<file>	Converts a symlink object into the file it points to. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
<symlink> as folder	*Cast*	<folder>	Converts a symlink object into the folder it points to. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
<symlink> as socket file	*Cast*	<socket file>	Casts a symlink type as a socket file type. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
<symlink> as string	*Cast*	<string>	Casts a symlink object as a string. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
<symlink> as symlink	*Cast*	<symlink>	Casts a symlink, provided for completeness. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

Key Phrase	*Form*	Return Type	Description
accessed time of <symlink>	*Plain*	<time>	Returns the last accessed time of the specified symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
ancestor of <symlink>	*Plain*	<folder>	Returns all ancestor folders (recursive parent folders) of the given symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
change time of <symlink>	*Plain*	<time>	Returns the last time the specified symlink was 'changed' by either writing it or setting its inode information. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
drive of <symlink>	*Plain*	<filesystem>	Returns the drive associated with the specified symlink as a <filesystem> object. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
filesystem of <symlink>	*Plain*	<filesystem>	Returns the filesystem on which the symlink resides. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
gid of <symlink>	*Plain*	<integer>	Returns the group ID of the given symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
group name of <symlink>	*Plain*	<string>	Returns the group name of the specified symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
link count of <symlink>	*Plain*	<integer>	Returns an integer corresponding to the number of hard links attached to the specified symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
location of <symlink>	*Plain*	<string>	Returns a string corresponding to the directory in which the symlink is located. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
modification time of <symlink>	*Plain*	<time>	Returns the time corresponding to the modification time of the specified symlink, not the file it points to. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
name of <symlink>	*Plain*	<string>	Returns a string that is the full pathname of the specified symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
parent folder of <symlink>	*Plain*	<folder>	Creates a folder object corresponding to the parent folder of the given symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

Key Phrase	Form	Return Type	Description
pathname of <symlink>	*Plain*	<string>	Returns a string that contains the full pathname of the specified symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
uid of <symlink>	*Plain*	<integer>	The user ID of the user who owns this symlink. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
user name of <symlink>	*Plain*	<string>	Returns the symlink owner's (user's) name. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
value accessible of <symlink>	*Plain*	<boolean>	Returns TRUE if the pathname pointed to by the specified symlink is available. Returns FALSE if the file object is missing or unavailable with the current permissions. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
value of <symlink>	*Plain*	<string>	Returns the pathname that the symlink points to. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

Examples

- `file "/example/link" as symlink`
▶ Returns the link (in file format) as a symlink object ("/example/link").

- `symlinks of folder "/example"`
▶ Returns a list of the symlink in the specified folder, whether or not they are broken. Note that this behavior is different from looking for files in a folder -- that will return links as well as files, but won't return broken links.

- `modification time of symlink "/example/link"`
▶ Returns the time corresponding to the modification time of the given symlink, not the file it points to.

- `modification time of file "/example/link"`
▶ Returns the time corresponding to the file object that the symlink is pointing to.

Download Storage Folder

Before an Action executes, the download storage folder points to a temporary directory that holds the downloads for the Action. During execution of the Action, those downloads are moved to the standard BigFix __Download folder and the Inspector then points there. These Inspectors return information about the currently specified download storage folder, and can be used with relevance substitution in download and prefetch Action commands.

Type Derivation: This object type is derived from the <folder> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
download storage folder	*PlainGlobal*	This Inspector creates a pointer to the current download storage folder. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
total size of <download storage folder>	*Plain*	<integer>	Returns the amount of storage currently being used to store downloads (in bytes). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Device File

These Inspector types interrogate Unix-style device files, which contain device drivers or system resources. Unix identifies these resources by a major number and a minor number, both stored as part of a node structure. Typically, the major number identifies the device driver and the minor number identifies the particular device controlled by that driver.

Type Derivation: This object type is derived from the <filesystem object> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
<filesystem object> as device file	*Cast*	Returns a device file or nothing (if the filesystem object specified is not a device file). Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Key Phrase	Form	Description
<symlink> as device file	*Cast*	Casts a symlink type as a device file type. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
device file <filesystem object>	*Index<filesystem object>Global*	Returns the device file indicated by the specified filesystem object. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
device file <string>	*NamedGlobal*	Returns the device file with the specified name. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
device file <string> of <folder>	*Named*	Returns the device file with the specified name in the given folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
device file <symlink>	*Index<symlink>Global*	Returns the device file specified by the supplied symbolic link. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
device file of <folder>	*Plain*	Returns the device files that exist in the specified folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
device type of <device file>	*Plain*	<string>	Returns the device type corresponding to the give device file, as a string. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
drive of <device file>	*Plain*	<filesystem>	Returns the drive associated with the specified device file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
filesystem of <device file>	*Plain*	<filesystem>	Returns the filesystem object corresponding to the specified device file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
major of <device file>	*Plain*	<integer>	Returns the major number of the specified device file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
minor of <device file>	*Plain*	<integer>	Returns the minor number of the specified device file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Fifo File

In Unix systems, a FIFO file is a named pipe that uses the file system as a way to store the pipe name. These Inspectors provide access to these named pipes.

Type Derivation: This object type is derived from the <filesystem object> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
<filesystem object> as fifo file	Cast	Returns a FIFO file or nothing (if the filesystem object specified is not a FIFO file). Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
<symlink> as fifo file	Cast	Casts a symlink type as a FIFO (named pipe) file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
fifo file <filesystem object>	Index<filesystem object>Global	Returns the FIFO file (named pipe) described by the specified filesytem object. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
fifo file <string>	NamedGlobal	Returns the FIFO file (named pipe) with the given name. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
fifo file <string> of <folder>	Named	Returns the FIFO file (named pipe) with the given name in the specified folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
fifo file <symlink>	Index<symlink>Global	Returns the FIFO file (named pipe) described by the specified symbolic link (symlink). Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
fifo file of <folder>	Plain	Returns the list of FIFO file in the given folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
drive of <fifo file>	Plain	<filesystem>	Returns the drive associated with the specified FIFO (named pipe) file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
filesystem of <fifo file>	Plain	<filesystem>	Returns the filesystem object corresponding to the specified FIFO file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Socket File

These Inspectors allow you to interrogate socket files, which are representations of UNIX domain sockets identified by their pathname.

Type Derivation: This object type is derived from the <filesystem object> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	*Form*	Description
<filesystem object> as socket file	*Cast*	Casts a given filesystem object as a socket file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
<symlink> as socket file	*Cast*	Casts a symlink type as a socket file type. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
socket file <filesystem object>	*Index<filesystem object>Global*	Returns the socket file(s) indicated by the supplied filesystem object. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
socket file <string>	*NamedGlobal*	Returns the named socket file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
socket file <string> of <folder>	*Named*	Returns the socket file with the given name in the specified folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
socket file <symlink>	*Index<symlink>Global*	Returns the socket file(s) indicated by the supplied symlink. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
socket file of <folder>	*Plain*	Returns the socket file(s) in the specified folder. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
drive of <socket file>	*Plain*	<filesystem>	Returns the drive associated with the specified socket file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
filesystem of <socket file>	*Plain*	<filesystem>	Returns the filesystem object corresponding to the specified socket file. Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

System Objects

These are the keywords available for querying various aspects of the system, including the name and version of the operating sytem. This chapter also covers the keywords used to describe the vendors and types of the various processors that coexist in a typical computer system. Some of these Inspectors are system-specific, but are included to provide cross-platform compatibility.

Bios

On Windows computers, this object returns strings that identify the version of the BIOS. On other computers, all bios expressions will fail gracefully, rather than generating an error.

Creation Methods

Key Phrase	Form	Description
bios	*PlainGlobal*	Returns the date of the bios if it exists, or <unknown> if it does not exist. This is a Windows-only command. On a non-Windows system, bios returns False. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

Properties

Key Phrase	Form	Return Type	Description
<bios> as string	*Cast*	<string>	This Windows-only Inspector returns a string that is the concatenation of the BIOS name and date. On a non-Windows operating system, it returns FALSE. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
date of <bios>	*Plain*	<string>	This Windows-only Inspector returns the date string stored in the bios. This string is formatted as MM/DD/YY. On a non-Windows operating system, it returns FALSE. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
version of <bios>	*Plain*	<string>	This Windows-only Inspector returns the first string of the multi-string version stored in the bios. This string may not exist. The format depends upon your BIOS manufacturer. On a non-Windows operating system, it returns FALSE. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

Operating System

The operating system object provides access to several important properties of the system.

Creation Methods

Key Phrase	*Form*	Description
operating system	*PlainGlobal*	Creates the global operating system object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<operating system> as string	*Cast*	<string>	Returns a string containing the name of the operating system concatenated with the release. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
architecture of <operating system>	*Plain*	<string>	Returns the architecture of the operating system. This is the value of the 'machine' element of the utsname structure obtained by calling uname. Lin, Sol, HPUX, AIX, Mac, Ubu
boot time of <operating system>	*Plain*	<time>	Returns the time of the last restart. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
build of <operating system>	*Plain*	<string>	Returns a string corresponding to the build number of the OS. Lin, Sol, HPUX, AIX, Mac, Ubu
codename of <operating system>	*Plain*	<string>	This *NIX Inspector returns a string that corresponds to the codename of the given release. For example, Ubuntu 8.04 has the codename of 'hardy'. Lin:8.1, Sol:8.1, HPUX:8.1, AIX:8.1, Ubu
mac of <operating system>	*Plain*	<boolean>	Returns TRUE if the client computer is a Macintosh. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
name of <operating system>	*Plain*	<string>	Returns the name of the operating system as a string. Names might include Win98, WinNT, etcetera. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
release of <operating system>	*Plain*	<string>	Information about the release of the operating system, formatted as a <version> on the Macintosh, but a <string> on UNIX and Windows. Win, Lin, Sol, HPUX, AIX, , WM, Ubu

Key Phrase	Form	Return Type	Description
unix of <operating system>	*Plain*	<boolean>	Returns TRUE if the local computer is a UNIX system. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
uptime of <operating system>	*Plain*	<time interval>	Returns a time interval that represents the elapsed time since the operating system was last booted. • Note: Depending on the notebook, this interval may not include time spent in hibernation. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
windows of <operating system>	*Plain*	<boolean>	Returns TRUE if the local computer is a Windows system. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Examples

- `now - boot time of operating system > week`
- ▶ Returns TRUE if the computer hasn't been rebooted for over a week.

- `name of operating system contains "Linux Red Hat"`
- ▶ Returns TRUE on a typical Linux Red Hat system.

Processor

The processor object is used to identify the number and properties of processors in the system. You can identify the manufacturer of the CPU as well as the speed and other features. Many operating systems provide for multiple processors. You can inspect any one of them by their ordinal number.
• For more information on Windows processors, see the Resource section at the end of this guide.

Creation Methods

Key Phrase	Form	Description
main processor	*PlainGlobal*	Creates the object associated with the 'Primary' processor. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
processor	*PlainGlobal*	Iterates through the processors in the system. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
processor <integer>	*NumberedGlobal*	Creates the processor object for the number specified. The first processor is processor number 1. Win, Lin, Sol, HPUX, AIX, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
bogomips of <processor>	*Plain*	<integer>	Returns the Bogomips of the specified processor, which provides a rough indication of its speed. Lin, Ubu
coma bug of <processor>	*Plain*	<boolean>	Returns TRUE if the specified processor is afflicted by the coma bug. The coma bug afflicts certain Cyrix CPUs, causing them to occasionally lock up. The affected CPUs include the Cyrix/IBM 6x86, 6x86L, and 6x86MX. Lin, Ubu
cpuid level of <processor>	*Plain*	<integer>	Returns an integer corresponding to the cpuid level for the given processor. Lin, Ubu
f00f bug of <processor>	*Plain*	<boolean>	Returns TRUE if the specified processor is afflicted by the f00f bug. This bug is named after the first two of a four-byte assembly language sequence that can freeze up x86 Pentium processors. Lin, Ubu
family name of <processor>	*Plain*	<string>	Returns the family name of the CPU, dependent on the type of client computer, for instance Pentium, Sparc, PowerPC G4, etcetera. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
family of <processor>	*Plain*	<integer>	Returns an integer representing the family of the CPU. See the notes for the meaning of these numbers. • Note: As of BES 6.0, this Inspector returns a string on Solaris and AIX computers. Win, Lin, , , WM, Ubu
fdiv bug of <processor>	*Plain*	<boolean>	Returns TRUE if the specified processor is afflicted by the fdiv bug, which occasionally causes incorrect floating point division answers. Lin, Ubu
flag list of <processor>	*Plain*	<string>	Returns a string containing the processor flags. To check if a processor has a particular flag, use the 'contains' operator. Lin, Ubu
fpu exception of <processor>	*Plain*	<boolean>	Returns TRUE if the fpu exception has occurred on this processor. Lin, Ubu

Key Phrase	Form	Return Type	Description
fpu of <processor>	*Plain*	<boolean>	Returns TRUE if the specified processor contains a floating point unit. Lin, Ubu
hlt bug of <processor>	*Plain*	<boolean>	Returns TRUE if the specified processor is afflicted by the hlt bug. Lin, Ubu
index of <processor>	*Plain*	<integer>	Returns the ordinal number of the processor on a multi processor machine. Lin, Sol, HPUX, AIX, Ubu
model name of <processor>	*Plain*	<string>	Returns a string containing the name of the processor model. Lin, Ubu
model of <processor>	*Plain*	<integer>	Returns the model number of the CPU. • Note: On Solaris, HPUX and AIX computers, this Inspector returns a <string> as of BES 6.0. Win, Lin, WM, Ubu
sep bug of <processor>	*Plain*	<boolean>	Returns TRUE if the given processor improperly reports the value of its SEP (Sysenter/sysexit Present) flag. Lin, Ubu
speed of <processor>	*Plain*	<hertz>	Returns the speed of the processor in Hertz. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
stepping of <processor>	*Plain*	<integer>	Returns the stepping number of the processor. This item can be helpful in identifying very specific processor features or limitations. Win, Lin, WM, Ubu
vendor name of <processor>	*Plain*	<string>	The manufacturer of the CPU. Names include: • GenuineIntel • AuthenticAMD • CyrixInstead • CentaurHauls • AmbiguousCPU. Win, Lin, WM, Ubu
wp of <processor>	*Plain*	<boolean>	Returns TRUE if the specified processor has the WP feature. Lin, Ubu

Examples

- `number of processors > 1`
▶ Returns TRUE if the computer is a multi-processor system.

- `speed of main processor < 2000 * MHz`
▶ Returns TRUE is the cpu is slower than 2Ghz.

- `vendor name of main processor = "GenuineIntel"`
▶ Returns TRUE for an Intel processor chip.

- `number of processors whose (vendor name of it = "AuthenticAMD" or vendor name of it = "CyrixInstead" or vendor name of it = "CentaurHauls") = 1`
▶ Returns TRUE for a single processor system with the given vendors.

Ram

The ram object is used to inspect properties of the computer's random access memory.

Creation Methods

Key Phrase	Form	Description
ram	*PlainGlobal*	Creates the object that can be accessed to inspect the amount of ram on the machine. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
random access memory	*PlainGlobal*	Same as 'ram'. Win, Lin, Sol, HPUX, AIX, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
available amount of <ram>	*Plain*	<integer>	Returns the amount of system ram that is currently available, in bytes. Lin, Ubu
buffered amount of <ram>	*Plain*	<integer>	Returns the amount of system RAM currently buffered, in bytes. Lin, Ubu
cached amount of <ram>	*Plain*	<integer>	Returns the amount of system RAM currently cached, in bytes. Lin, Ubu

Key Phrase	Form	Return Type	Description
free amount of \<ram\>	*Plain*	\<integer\>	Returns the amount of system RAM currently unused, in bytes. Lin, Sol, HPUX, AIX, Ubu
shared amount of \<ram\>	*Plain*	\<integer\>	Returns the amount of system RAM currently shared, in bytes. Lin, Ubu
size of \<ram\>	*Plain*	\<integer\>	Returns the number of bytes of random access memory on the current machine. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
total amount of \<ram\>	*Plain*	\<integer\>	Same as size of \<ram\>. Lin, Sol, HPUX, AIX, Ubu
unavailable amount of \<ram\>	*Plain*	\<integer\>	Returns the amount of system RAM currently unavailable, in bytes. Lin, Ubu
used amount of \<ram\>	*Plain*	\<integer\>	Returns the amount of system RAM currently used, in bytes. Lin, Sol, HPUX, AIX, Ubu

Examples

- ```
  size of ram / (1024 * 1024)
  ```
- Returns the size of RAM in megabytes.

Service

These Inspectors provide access to all services configured on Windows NT, 2K and XP systems. On a non-Windows system, expressions using these objects will fail gracefully instead of generating an error.

Creation Methods

Key Phrase	Form	Description
main gather service	*PlainGlobal*	Returns a service object for the main gathering service, typically located on the main server. • Note: On a Macintosh, returns <nothing>. Included for compatibility. Win, Lin, Sol, HPUX, AIX, , Ubu
relay service	*PlainGlobal*	Returns a service object for the relay component of BES. • Note: On a Macintosh, this returns <nothing>. Win, Lin, Sol, HPUX, AIX, , Ubu
running service <string>	*NamedGlobal*	Creates the running service object for the specified name. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
service <string>	*NamedGlobal*	Returns the service object matching the name provided, regardless of its running state. • On a Macintosh, returns a <dummy> type. Win, Lin, Sol, HPUX, AIX, , Ubu

Properties

Key Phrase	Form	Return Type	Description
state of <service>	*Plain*	<string>	Returns one of Continuing, Pausing, Paused, Running, Starting, Stopping, Stopped, Unknown. Win, Lin, Sol, HPUX, AIX, Ubu
version of <service>	*Plain*	<version>	This Inspector takes the specified service property and retrieves its version (file version). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

Examples

- ```
  version of service "BESClient"
  ```
- Returns a version number, such as '8.0.584.0'.

Process

Processes allocate the various resources needed to execute a program. Processes have a process identifier, a virtual address space, associated code, a priority class, security settings, environment variables, min and max working set sizes, and at least one executing thread. Processes are typically started with a single primary thread which in turn can spawn additional threads.

Creation Methods

Key Phrase	Form	Description
process	*PlainGlobal*	Returns all process objects currently running. Win:8.0, Lin, Sol, HPUX, AIX, Mac, Ubu
process <integer>	*NumberedGlobal*	Returns the process object corresponding to the given integer pid. Lin, Sol, HPUX, AIX, Mac, Ubu
process <string>	*NamedGlobal*	Returns the process object corresponding to the name specified by <string>. Lin, Sol, HPUX, AIX, Ubu

Properties

Key Phrase	Form	Return Type	Description
command line argument <integer> of <process>	*Numbered*	<string>	Returns the Nth command line argument of the specified process. Lin:8.0, Sol:8.0, AIX:8.0, Ubu
command line argument of <process>	*Plain*	<string>	Returns the command line arguments of the specified process. Lin:8.0, Sol:8.0, AIX:8.0, Ubu
environment of <process>	*Plain*	<environment>	Returns the environment of the specified process. Lin:8.0, Sol:8.0, Ubu
id of <process>	*Plain*	<integer>	Returns the integer ID of the specified process. Win:8.0, Lin, Sol, HPUX, AIX, Mac, Ubu
name of <process>	*Plain*	<string>	Returns the name (as a string) of the specified process. Win:8.0, Lin, Sol, HPUX, AIX, Mac, Ubu
pid of <process>	*Plain*	<integer>	Returns the integer process ID for the specified process. Lin, Sol, HPUX, AIX, Mac, Ubu

Key Phrase	*Form*	Return Type	Description
process id of <process>	*Plain*	<integer>	Returns the integer process ID for the specified process. Lin, Sol, HPUX, AIX, Mac, Ubu

Examples

- `names of processes whose (pid of it < 20)`
▶ Returns a list of all process with an ID less than 20.

Swap

This object lets you inspect the properties of your swap space.

Creation Methods

Key Phrase	*Form*	Description
swap	*PlainGlobal*	Creates an object containing information about the swap partition. Lin, HPUX, AIX, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
free amount of <swap>	*Plain*	<integer>	Returns the amount of the swap partition currently unused, in bytes. Lin, HPUX, AIX, Ubu
size of <swap>	*Plain*	<integer>	Returns the size, in bytes, of the swap partition. Lin, HPUX, AIX, Ubu
total amount of <swap>	*Plain*	<integer>	Same as size of <swap>. Lin, HPUX, AIX, Ubu
used amount of <swap>	*Plain*	<integer>	Returns the amount of the swap partition currently in use, in bytes. Lin, HPUX, AIX, Ubu

Language

A language is composed of a primary language (for example, Swiss) and a sub-language (for example, Swiss German).

Creation Methods

Key Phrase	*Form*	Description
system locale	*PlainGlobal*	Determines which bitmap fonts, and OEM, ANSI, and MAC code pages are defaults for the system. This only affects applications that are not fully Unicode. Win, Lin, Sol, HPUX, AIX, WM, Ubu
system ui language	*PlainGlobal*	Determines the default language of menus and dialogs, messages, INF files, and help files. Win, Lin, Sol, HPUX, AIX, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<language> as string	*Cast*	<string>	Returns the language of the system locale. Win, Lin, Sol, HPUX, AIX, WM, Ubu
platform id of <language>	*Plain*	<string>	Returns the string resulting from a call to setlocale(LC_TYPE, ""). This call examines the system environment and returns a string representing the language and character set for any text-related system function. The string is of the form "en_US.UTF-8". Lin, Sol, HPUX, AIX, Ubu
primary language of <language>	*Plain*	<primary language>	Extracts the primary language identifier from a language. Win, Lin, Sol, HPUX, AIX, WM, Ubu

Examples

- `system locale as string`
  ▶ Returns English (United States) if the system locale is US English.

Primary Language

A primary language identifier indicates the written/spoken language that is used by the system. However, to identify the language that is used in a country or region you must combine the primary language with a sub-language identifier to form language identifiers.

Creation Methods

Key Phrase	Form	Description
primary language of <language>	*Plain*	Extracts the primary language identifier from a language. Win, Lin, Sol, HPUX, AIX, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
<primary language> as string	*Cast*	<string>	Returns the primary language. Win, Lin, Sol, HPUX, AIX, WM, Ubu

Examples

- ■ `primary language of system locale`
- ▶ Returns "English" for an English-language system.

Runlevel

The runlevel Inspectors refer to a mode of operation in various Unix systems. Typically, when a computer enters runlevel zero, it halts and when it enters runlevel six, it reboots. The intermediate runlevels differ widely among operating systems.

Creation Methods

Key Phrase	Form	Description
runlevel	*PlainGlobal*	Returns the current runlevel of the local machine. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
<runlevel> as string	*Cast*	<string>	Casts a runlevel object as a string. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

Key Phrase	*Form*	Return Type	Description
effective time of <runlevel>	*Plain*	<time>	Returns the time at which the runlevel was set to its current value. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu
previous value of <runlevel>	*Plain*	<string>	Returns the previous value of the runlevel. In the case where the runlevel has not changed since the system booted, this property evaluates to the empty string. Lin:7.0, Ubu
value of <runlevel>	*Plain*	<string>	Returns the current runlevel. It evaluates to a platform-dependent string indicating the current runlevel. For instance, on Linux the value '3' indicates runlevel 3 and 'S' indicates single user mode. Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Ubu

SMBIOS objects

The System Management BIOS (SMBIOS) specification defines data structures and access methods in the BIOS. These Inspectors are designed to supersede the DMI Inspectors.

Smbios

The SMBIOS (System Management BIOS) is a database containing information about the system hardware and firmware. It consists of a series of structures that each contain a logical grouping of basic system information such as processor, baseboard, memory, ports, slots, and more. Each structure, in turn, contains specific named values that can be interrogated. For instance, you easily access information about the client processor chip, including the manufacturer, speed and more by accessing the desired value of the 'processor information' structure. These SMBIOS Inspectors are based on DMTF Standard version 2.6.1. For a complete listing of the SMBIOS structures and the strings used to query them, refer to the SMBIOS Inspectors in the Resources section at the end of this guide.

Creation Methods

Key Phrase	*Form*	Description
smbios	*PlainGlobal*	Returns a basic smbios object, a function formerly served by the Desktop Management Interface, or DMI. This object contains searchable structures related to the computer BIOS. Win:8.0, Lin:8.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
structure <string> of <smbios>	*Named*	<smbios structure>	Returns the smbios structure corresponding to the specified string as defined in the 'Type' entries in the SMBIOS specification. Win:8.0, Lin:8.0, Ubu
structure of <smbios>	*Plain*	<smbios structure>	A property containing information pertaining to an instance of a given smbios structure as defined in the 'Type' entries in the SMBIOS specification. This is implemented as a named iterated property. Win:8.0, Lin:8.0, Ubu

Examples

- ```exists structure "bios_information" of smbios```
- Returns TRUE if the structure named BIOS Information exists.

- ```number of structures of smbios > 10```
- Returns TRUE if the number of smbios structures exceeds ten.

Smbios Structure

The SMBIOS (System Management BIOS) is composed of a set of named structures, such as processor_information, bios_information, and more.

Creation Methods

Key Phrase	Form	Description
structure <string> of <smbios>	*Named*	Returns the smbios structure corresponding to the specified string as defined in the 'Type' entries in the SMBIOS specification. Win:8.0, Lin:8.0, Ubu
structure of <smbios value>	*Plain*	Returns the parent smbios structure(s) containing the specified value. Win:8.0, Lin:8.0, Ubu
structure of <smbios>	*Plain*	Returns the smbios structure(s) as defined in the 'Type' entries in the SMBIOS specification. This is implemented as a named iterated property. Win:8.0, Lin:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
integer <string> of <smbios structure>	*Named*	<integer>	Returns the integer data of the named value in a specified smbios structure. Returns NSO if the data type is not integer. Win:8.0, Lin:8.0, Ubu
integer value <string> of <smbios structure>	*Named*	<smbios value>	Returns the integer value corresponding to the specified value of the smbios structure. This only exists for those values whose data type is integer. Win:8.0, Lin:8.0, Ubu
length of <smbios structure>	*Plain*	<integer>	Returns the length of the smbios structure as defined in the SMBIOS specification. Win:8.0, Lin:8.0, Ubu

Key Phrase	*Form*	Return Type	Description
name of <smbios structure>	*Plain*	<string>	Returns a string containing the name of the specified smbios structure as derived from the 'Type' entry in the SMBIOS specification. To create a valid Relevance keyword, some rules are applied to the name: • Spaces are replaced with underscores • Parenthesized sections are deleted • Leading and trailing whitespace is removed • Uppercase is transformed to lowercase • Keywords beginning with a number have a 'b' prepended. Win:8.0, Lin:8.0, Ubu
string <string> of <smbios structure>	*Named*	<string>	Returns the named value in the specified smbios structure. Win:8.0, Lin:8.0, Ubu
string value <string> of <smbios structure>	*Named*	<smbios value>	Returns the string value corresponding to the specified value of the smbios structure. This only exists for those values whose data type is string. Win:8.0, Lin:8.0, Ubu
type of <smbios structure>	*Plain*	<integer>	Returns the type of the smbios structure (as an integer) defined in the SMBIOS specification. Win:8.0, Lin:8.0, Ubu
value <string> of <smbios structure>	*Named*	<smbios value>	Returns the named value of the specified smbios structure. Win:8.0, Lin:8.0, Ubu
value of <smbios structure>	*Plain*	<smbios value>	Returns the element of the given smbios structure that contains information specific to the particular machine. Values can have integer or string data. There can be multiple value entries with the same name, so values are implemented as iterated and named iterated properties of the smbios structure. Win:8.0, Lin:8.0, Ubu

Examples

■ `names of structures of values "bios_version" of structures of smbios`

▶ Returns the names of the smbios structures that contain the specified value, in this case the named value 'bios version'.

■ `number of structures of smbios > 10`

▶ Returns TRUE if the number of smbios structures exceeds ten.

- `integer "bios_rom_size" of structure "bios_information" of smbios`
  - ▶ Returns the integer value of the bios rom size from the 'bios_information' structure.

- `integer value "bios_characteristics" of structure "bios_information" of smbios`
  - ▶ Returns the integer value corresponding to the 'bios characteristics' of the 'bios information' structure.

- `string value "bios_version" of structure "bios_information" of smbios`
  - ▶ Returns the string value corresponding to the 'bios version' of the 'bios information' structure.

Smbios Value

Information about the SMBIOS (System Management BIOS) is contained in a set of data values stored in collections of individual smbios structures.

Creation Methods

Key Phrase	Form	Description
integer value <string> of <smbios structure>	*Named*	Returns the integer value corresponding to the specified value of the smbios structure. This only exists for those values whose data type is integer. Win:8.0, Lin:8.0, Ubu
string value <string> of <smbios structure>	*Named*	Returns the smbios value corresponding to the specified string. This only exists for those values whose data type is string. Win:8.0, Lin:8.0, Ubu
value <string> of <smbios structure>	*Named*	Returns the named value of the specified smbios structure. Win:8.0, Lin:8.0, Ubu
value of <smbios structure>	*Plain*	Returns the element of the given smbios structure that contains information specific to the particular machine. Values can have integer or string data. There can be multiple value entries with the same name, so values are implemented as iterated and named iterated properties of the smbios structure. Win:8.0, Lin:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
<smbios value> as string	*Cast*	<string>	Casts a smbios value as a string type. Win:8.0, Lin:8.0, Ubu

Key Phrase	*Form*	Return Type	Description
name of <smbios value>	*Plain*	<string>	Returns the name(s) of the specified smbios values, typically as derived from a smbios structure. Win:8.0, Lin:8.0, Ubu
offset of <smbios value>	*Plain*	<integer>	Returns the offset position relative to the start of the given smbios value in its smbios structure. For more information about these offsets, see the SMBIOS specification. Win:8.0, Lin:8.0, Ubu
structure of <smbios value>	*Plain*	<smbios structure>	Returns the parent smbios structure(s) containing the specified value. Win:8.0, Lin:8.0, Ubu
type of <smbios value>	*Plain*	<string>	Returns the data type of the specified smbios value. This can be one of: • Byte • Word • Dword • Qword • String • BinaryString • MultipleString. Win:8.0, Lin:8.0, Ubu

Examples

■   `integer value "bios_characteristics" of structure "bios_information" of smbios`

▶   Returns the integer value corresponding to the 'bios characteristics' of the 'bios information' structure.

■   `string value "bios_version" of structure "bios_information" of smbios`

▶   Returns the string value corresponding to the 'bios version' of the 'bios information' structure.

■   `names of values of structure "bios_information" of smbios`

▶   Returns a list of the names of all the values of the specified smbios structure.

■   `offsets of values of structure "bios_information" of smbios`

▶   Returns a list of the offsets of each value in the specified smbios structure.

■   `names of structures of values "bios_version" of structures of smbios`

▶   Returns the names of the smbios structures that contain the specified value, in this case the named value 'bios version'.

- `types of values of structure "bios_information" of smbios`
- ▶ Returns a list of the types of each value in the 'bios information' SMBIOS structure.

Site Objects

These keywords query the properties of Fixlet sites to which the client is subscribed.

Site

A Site object is provided to access properties of Fixlet sites.

Creation Methods

Key Phrase	Form	Description
current site	*PlainGlobal*	Creates the site object corresponding to the site that provided the current Fixlet. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site	*PlainGlobal*	Iterates through all the sites. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site \<string>	*NamedGlobal*	Creates the site object that corresponds to the name provided. The name is interpreted as a site locator and is therefore a URL. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site of \<fixlet>	*Plain*	Returns the site corresponding to the specified Fixlet message. Win:8.1, Lin:8.1, Sol:8.1, HPUX:8.1, AIX:8.1, Mac:8.1, Ubu

Properties

Key Phrase	Form	Return Type	Description
client folder of \<site>	*Plain*	\<folder>	The folder containing the site content on the client machine. Site content is gathered into this location. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
fixlet of \<site>	*Plain*	\<fixlet>	Iterates through the Fixlet messages of the specified site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
gather schedule authority of \<site>	*Plain*	\<string>	Returns a string corresponding to the authority of the site schedule, for example: Publisher, Custom, Manual or Disabled. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
gather schedule time interval of <site>	Plain	<time interval>	Returns the time interval between automatic gathering of site content. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
group <integer> of <site>	Numbered	<site group>	Returns an object corresponding to the numbered group of the specified site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
last gather time of <site>	Plain	<time>	Returns the time of last successful gathering from the site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
masthead of <site>	Plain	<file>	Each site has a masthead, and the masthead is saved into the site data folder upon successful creation. This property returns a file object that corresponds to the copy in the site data folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
name of <site>	Plain	<string>	The name of the site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
relevant fixlet of <site>	Plain	<fixlet>	Iterates through the Relevant Fixlet messages for the specified site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
relevant offer action of <site>	Plain	<action>	Returns the list of relevant actions that are offers for the specified site. This Inspector could be useful in a client UI dashboard listing the current set of relevant offers. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
setting <string> of <site>	Named	<setting>	Returns the setting whose name matches the string provided from the Fixlet site settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
setting of <site>	Plain	<setting>	Returns one or more settings from the site settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site tag of <site>	Plain	<string>	Returns the last component of the specified site's url, eg. 'actionsite', 'enteprisesecurity', etcetera. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site version list of <site>	Plain	<site version list>	Returns the last gathered site version list (manyversion) of the specified site. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.0, Mac:7.1, WM, Ubu
subscribe time of <site>	Plain	<time>	Returns the time that the current machine began subscribing to the site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
type of <site>	*Plain*	<string>	Returns one of the following 4 literal strings: • Master Action Site • Operator Site • Custom Site • Fixlet Site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
url of <site>	*Plain*	<string>	Returns the Locator found in the masthead. A site locator is used to synchronize with the site. It normally contains the URL of a remote file system folder, or the URL of a cgi-bin program that provides a remote directory listing of the site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
version of <site>	*Plain*	<integer>	Returns the version number of the site content. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

NOTE: The 'as string' property yields a string formatted with the site name.

Examples

- ```
  exists site "actionsite"
  ```
▶ TRUE when the action site exists on the target machine.

- ```
  exists file "siteicon.bmp" of client folder of current site
  ```
▶ TRUE if the specified file exists in the client folder.

- ```
  last gather time of current site > now - 30 * day
  ```
▶ Return TRUE if it has been over 30 days since last gathering, or synchronizing, with the site.

- ```
  last gather time of current site < time "4 Aug 1997 01:00 pdt"
  ```
▶ Returns TRUE if the site was last synchronized before the specified date.

- ```
  modification time of masthead of current site < time "4 Aug 1997 01:00 pdt"
  ```
▶ Returns TRUE if the masthead of the current site is older than the specified date.

Site Group

These Inspectors return information on the automatic groups defined for a given site.

Creation Methods

Key Phrase	Form	Description
group <integer> of <site>	*Numbered*	Returns an object corresponding to the numbered group of the specified site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
id of <site group>	*Plain*	<integer>	Returns the numeric ID of the specified site group. This is the number assigned to an automatic group when it is first defined. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
member of <site group>	*Plain*	<boolean>	Returns TRUE if the current computer is a member of the specified group. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Site Version List

These Inspectors examine the multidimensional version numbers (ManyVersions) that are used by the Database to reconcile reconnected sites after a DSA failback event.

Creation Methods

Key Phrase	Form	Description
site version list of <site>	*Plain*	Returns the last gathered site version list (manyversion) of the specified site. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.0, Mac:7.1, WM, Ubu

Fixlet Objects

These Inspectors return information about individual Fixlets.

Fixlet

These Inspectors can provide important information about the Fixlet messages at any site. These Inspectors only work in the context of property evaluation, not Fixlet evaluation.

Creation Methods

Key Phrase	Form	Description
current analysis	PlainGlobal	This Client Inspector is used to locate the site corresponding to the current analysis in order to look at certain related files. This is helpful for SCM content that resides in Fixlet sites and can be copied to custom sites. The value of 'current analysis' will move with the copy. In the Client context, this Inspector has global scope and returns a Fixlet. • Note: When used in a session context, this Inspector has a scope limited to the BES Fixlet. Win:8.1, Lin:8.1, Sol:8.1, HPUX:8.1, AIX:8.1, Mac:8.1, Ubu
fixlet of <site>	Plain	This Inspector iterates over all the Fixlet messages in the given site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
relevant fixlet of <site>	Plain	Iterates over all the relevant Fixlet messages in the specified site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
header <string> of <fixlet>	Named	<fixlet_header>	Returns the named header (case insensitive) of the specified Fixlet message. Fixlet headers are name:value pairs. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
header of <fixlet>	Plain	<fixlet_header>	Iterates over all the headers of the Fixlet message. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
id of <fixlet>	Plain	<integer>	Returns the numeric ID number of the specified Fixlet message. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
relevance of <fixlet>	*Plain*	<boolean>	Returns a boolean TRUE or False, depending on the Relevance of the specified Fixlet message. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site of <fixlet>	*Plain*	<site>	Returns the site corresponding to the specified Fixlet message. Win:8.1, Lin:8.1, Sol:8.1, HPUX:8.1, AIX:8.1, Mac:8.1, Ubu

Fixlet_header

Fixlet headers are name:value pairs that can provide important information about the Fixlet messages at any site. These Inspectors only work in the context of property evaluation, not Fixlet evaluation.

Creation Methods

Key Phrase	Form	Description
header <string> of <fixlet>	*Named*	Returns the named header (case insensitive) of the specified Fixlet message. Fixlet headers are name:value pairs. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
header of <fixlet>	*Plain*	Iterates over all the headers of the Fixlet message. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
name of <fixlet_header>	*Plain*	<string>	Headers are name:value pairs, separated by a colon. This Inspector returns the name on the left hand side of the pair. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
value of <fixlet_header>	*Plain*	<string>	Headers are name:value pairs, separated by a colon. This Inspector returns the value on the right hand side of the pair. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Examples

■ `number of relevant fixlets whose (value of header "x-fixlet-source-severity" of it as lowercase = "critical") of site "enterprise security".`

▶ Returns the number of critical fixlets in the Enterprise Security site.

Client Objects

These Inspectors retrieve information about the application containing the relevance evaluator.

Client

The client object allows access to properties of the client application hosting the relevance evaluation, typically a BigFix program. In addition, the client maintains a collection of settings with both name and value properties that are inspectable using the client object. These Inspectors share properties of application types, such as version and size.

Type Derivation: This object type is derived from the <application> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
client	*PlainGlobal*	Returns the client object corresponding to the BigFix application evaluating the current relevance expression. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
administrator <string> of <client>	*Named*	<setting>	If the administrator named in the <string> is enabled on the given <client> computer, this property returns a setting with the given name and the value 'allow.' For instance, if the name of the administrator is joe_admin, then the client would return a setting object with the name 'joe_admin' and a value of 'allow'. Casting this as a string would return 'joe_admin=allow'. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
administrator of <client>	*Plain*	<setting>	Returns one or more settings each representing an administrator of the client. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
brand of <client>	*Plain*	<string>	Returns the branding ID of a client computer. BigFix is the norm, but there are other brands that use the technology, including Trend Micro. Win:8.1, Lin:8.1, Sol:8.1, HPUX:8.1, AIX:8.1, Mac:8.1, Ubu

Key Phrase	Form	Return Type	Description
evaluationcycle of <client>	Plain	<evaluation cycle>	Returns an object corresponding to the time it takes to evaluate the content set on the specified BigFix Client. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
registration address of <client>	Plain	<ipv4or6 address>	This Inspector returns the IP address (as an <ipv4or6 address> type) that the specified BigFix client registered with. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
registration cidr address of <client>	Plain	<string>	This Inspector returns the cidr address from the adapter that the specified BigFix client registered with. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
registration mac address of <client>	Plain	<string>	This Inspector returns the MAC address that the specified BigFix client registered with. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
registration subnet address of <client>	Plain	<ipv4or6 address>	This Inspector returns the subnet address (as an <ipv4or6 address> type) from the adapter that the specified BigFix client registered with. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
setting <string> of <client>	Named	<setting>	Returns a client setting whose name matches the string provided from the client settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
setting of <client>	Plain	<setting>	Returns one or more settings from the client settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
upload progress of <client>	Plain	<string>	Returns a status message string indicating No Progress, Errors or a string like the following to indicate the upload progress: • <filename>: x of <filesize> bytes in <number> seconds. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
version of <client>	Plain	<version>	The product version of the BES application (BESClient or QnA). • Note: On the Macintosh only, this Inspector returns a <string>. Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Examples

■   `registration mac address of client`

▶   Returns a MAC address such as 00-1e-c9-4d-ce-5c.

- `version of client as string`
  - ▶ Returns a string like "4.0.3.7".

Setting

A setting is a simple object with name and value properties. It is a property of a client, or a property of a site. Settings of a site have a site scope. Settings of the client have a client scope. See the 'setting' commands in the action guide for more details.

Creation Methods

Key Phrase	Form	Description
administrator <string> of <client>	Named	Creates a setting with the given name on the given <client> computer. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
administrator of <client>	Plain	Returns one or more settings each representing an administrator of the client. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
setting <string> of <client>	Named	Returns the setting whose name matches the string provided from the client settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
setting <string> of <site>	Named	Returns the setting whose name matches the string provided from the site settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
setting of <client>	Plain	Returns one or more settings from the client settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
setting of <site>	Plain	Returns one or more settings from the site settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
<setting> as string	Cast	<string>	Returns a string formatted as <name>=<value> for the setting. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
effective date of <setting>	Plain	<time>	Returns the date when the setting was last modified. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
enabled of <setting>	Plain	<boolean>	Returns TRUE if the specified setting is enabled. Win:7.0, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
name of <setting>	Plain	<string>	Returns the name of the setting. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
value of <setting>	Plain	<string>	Returns the value of the setting. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Examples

- ■ `names of settings of site "actionsite"`
- ▶ Returns the names of all the settings of the site named "actionsite".

Selected Server

These Inspectors return information about the BES Server or BES Relay to which the BigFix agent reports.

Creation Methods

Key Phrase	Form	Description
selected server	PlainGlobal	The BES Server or BES Relay to which the agent reports. Returned as the "selected server" type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
competition size of <selected server>	Plain	<integer>	The number of servers in the competition from which this server was selected. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
competition weight of <selected server>	Plain	<integer>	The total of the weights of the servers in the competition from which this server was selected. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
distance of <selected server>	Plain	<integer range>	The distance, in IP gateway hops, to the server. Among servers with the same priority, closer servers are preferred. Returns an integer range, since the exact distance may not be known. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
full gateway address of <selected server>	*Plain*	<ipv4or6 address>	During relay selection, a traceroute-like list of the hops between the client and its relay (the selected server) is recorded. That list is accessible through this Inspector. Unlike the 'gateway address' Inspector, this Inspector includes hops that don't reply as 0.0.0.0. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
gateway address <integer> of <selected server>	*Numbered*	<ipv4or6 address>	During relay selection, a traceroute-like list of the hops between the client and its relay (the selected server) is recorded. The elements of that list is accessible through this Inspector. • Prior to version 8.0, this inspector returned an <ipv4 address> type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
gateway address of <selected server>	*Plain*	<ipv4or6 address>	During relay selection, a traceroute-like list of the hops between the client and its relay (the selected server) is recorded. That list is accessible through this Inspector. However, this Inspector ignores hops that don't reply. If you need the full list, use the 'full gateway address' Inspector. • Prior to version 8.0, this inspector returned an <ipv4 address> type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
ip address of <selected server>	*Plain*	<ipv4or6 address>	The ipv4or6 address to which reports are sent. • Prior to version 8.0, this inspector returned an <ipv4 address> type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
name of <selected server>	*Plain*	<string>	The DNS name of the server, if known. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
port number of <selected server>	*Plain*	<integer>	The port number to which reports are sent. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
priority of <selected server>	*Plain*	<integer>	The priority assigned to the server by the BES console. Servers with low priorities are preferred to servers with high priority. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
weight of <selected server>	*Plain*	<integer>	The weight assigned to the server by the BES console. Servers with the same priority and approximate distance compete to be chosen; servers with higher weights are more likely to be chosen. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Current Relay

These Inspectors refer to the BES Server or Relay that the client last registered with.

Creation Methods

Key Phrase	Form	Description
current relay	*PlainGlobal*	Returns an object corresponding to the server or relay that the client last registered with. This may be a BES Relay or the BES root server. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
version of <current relay>	*Plain*	<version>	Returns a version object that is the version of the server that the client last registered with. This may be a BES Relay or the BES root server. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Root Server

These Inspectors refer to the root server that the Bes Client is currently connected to.

Creation Methods

Key Phrase	Form	Description
root server	*PlainGlobal*	Returns an object representing the root BES Server to which the client last registered. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
host name of <root server>	*Plain*	<string>	The host (DNS) name of the BES root server that the BES Client last registered with. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
id of <root server>	*Plain*	<integer>	The DSA Server ID of the BES root server that the BES Client last registered with. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Evaluation Cycle

An Evaluation cycle represents a complete run through all the content available on the BigFix Client, measured in milliseconds. These Inspectors return statistics based on the time sampled whenever the client returns to the beginning of its content set. These Inspectors require a Client context.

Creation Methods

Key Phrase	Form	Description
evaluationcycle of <client>	*Plain*	Returns an object corresponding to the time it takes to evaluate the content set on the specified BigFix Client. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
average of <evaluation cycle>	*Plain*	<integer>	Returns the average time, in milliseconds, that it takes to evaluate a given BigFix Client content set. The average is based on the last ten cycles. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
maximum of <evaluation cycle>	*Plain*	<integer>	Returns the maximum time, in milliseconds, that it takes to evaluate a given BigFix Client content set. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Examples

- ```
  average of evaluationcycle of client
  ```
  ▸ Returns the average evaluation cycle time in milliseconds.

Application Usage Summary

To enable these Inspectors, you first need to create the client setting _BESClient_UsageManager_EnableAppUsageSummary and initialize it to 1. You must also configure the set of applications to monitor by creating the client setting _BESClient_UsageManager_EnableAppUsageSummaryApps and initializing it to a list of apps to include (or exclude). The value of this setting should look like +:app1:app2:app3: to add apps to the scope, and -:app1:app2: to exclude apps. The case is ignored. For instance, to only track summary usage on the Word application, use the value +:winword.exe:.

Creation Methods

Key Phrase	*Form*	Description
application usage summary	*PlainGlobal*	Returns an application usage summary containing information including the start time, duration and other statistics on client applications. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
application usage summary <string>	*NamedGlobal*	Returns the usage summary for the application specified in <string>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
first start time of <application usage summary>	*Plain*	<time>	Returns the start time of the specified application since the computer was configured to track it, regardless of reboots. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
instance of <application usage summary>	*Plain*	<application usage summary instance>	Returns a list of all the instances of a specified application usage summary. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
last start time of <application usage summary>	*Plain*	<time>	Returns the last time this specified application was started. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
last time seen of <application usage summary>	*Plain*	<time>	Returns the last time this specified application was seen running. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
name of <application usage summary>	*Plain*	<string>	Returns the names of the applications that are currently enabled for usage summaries. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
running of <application usage summary>	*Plain*	<boolean>	Returns TRUE if the specified application is currently running. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
total duration of <application usage summary>	*Plain*	<time interval>	Returns the total elapsed time that the specified application has been running. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
total run count of <application usage summary>	*Plain*	<integer>	Returns the number of times that the specified application has been run since the client was configured to track it. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Application Usage Summary Instance

These Inspectors return information about the multiple instances of specific applications.

Creation Methods

Key Phrase	Form	Description
instance of <application usage summary>	*Plain*	Returns a list of all the instances of a specified application usage summary. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
first start time of <application usage summary instance>	*Plain*	<time>	Returns the start time of the specified application instance since the computer was configured to track it, regardless of reboots. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
last start time of <application usage summary instance>	*Plain*	<time>	Returns the last time this specified application was started. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
last time seen of <application usage summary instance>	*Plain*	<time>	Returns the last time this specified application was seen running. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	Form	Return Type	Description
name of <application usage summary instance>	Plain	<string>	Returns the name(s) of the application instance(s) currently enabled for usage summaries. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
size of <application usage summary instance>	Plain	<integer>	Returns the size of the specified application instance. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
string version of <application usage summary instance>	Plain	<string>	Returns the version of the specified application instance as a string value. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
total duration of <application usage summary instance>	Plain	<time interval>	Returns the total elapsed time that the specified application instance has been running. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
total run count of <application usage summary instance>	Plain	<integer>	Returns the number of times that the specified application instance has been run since the client was configured to track it. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
version of <application usage summary instance>	Plain	<version>	Returns the version of the specified application instance. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

License Objects

These Inspectors retrieve information about the licensing of particular BigFix products.

License

These Inspectors are available to inspect the properties of the deployment license.

Creation Methods

Key Phrase	Form	Description
bes license	PlainGlobal	Synonym for 'client license'. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
client license	PlainGlobal	Creates the global object containing client licensing information. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
allow unmentioned site of <license>	*Plain*	<boolean>	If this property is TRUE, then the deployment is allowed to use sites that aren't mentioned in the license of any BES products. If FALSE, those sites will not be usable. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
common name of <license>	*Plain*	<string>	Returns the name of the person (such as John Smith) who requested the action site license. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
email address of <license>	*Plain*	<string>	Returns the email address of the person (such as John_Smith@bigcorp.com) who requested the action site license. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
encryption certificate of <license>	*Plain*	<x509 certificate>	Provides the encryption certificate that is currently active and which will be used by clients to encrypt reports. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
evaluation of <license>	*Plain*	<boolean>	Returns TRUE if client is running an evaluation license. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
expiration date of <license>	*Plain*	<time>	Returns date when license will expire. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
expiration state of <license>	*Plain*	<string>	Returns a string, one of "Unrestricted", "Grace" or "Restricted". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
fips mode of <license>	*Plain*	<boolean>	Returns TRUE if the BES action masthead specifies that applications (the client, console, or web reports, depending on the context) in the deployment should operate in FIPS 140-2 compliant mode. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
gather url of <license>	*Plain*	<string>	Returns the gather URL for the deployment's main Action site as specified in the deployment masthead. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
maximum seat count of <license>	*Plain*	<integer>	Returns maximum seat count allowed by the license. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
organization of <license>	Plain	<string>	Returns the organization of the person (such as Bigcorp, Inc.) who requested the action site license. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
product of <license>	Plain	<bes product>	Returns BES product objects obtained from the product fields of the specified license. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
registrar number of <license>	Plain	<integer>	A unique number assigned to the issuer of the Action Site certificate. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
seat count state of <license>	Plain	<string>	Returns one of "Unrestricted", "Grace" or "Restricted". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
seat of <license>	Plain	<integer>	The license number assigned to the client. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site number of <license>	Plain	<integer>	A unique number assigned to the Action Site certificate. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
start date of <license>	Plain	<time>	The starting date specified for the BigFix license. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
type of <license>	Plain	<string>	Returns the string that was assigned to the license when it was authorized by BigFix. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Examples

- ```
  maximum seat count of bes license
  ```
- Returns the current number of BigFix Clients allowed by this license.

BES Product

A BigFix license can include more than one product. Each product has an expiration date and a maximum seat count for any type of computer, or by non-windows server, windows server, or workstation. Each product also has a name and a list of site urls. For example, a patch management product might include site urls pointing to the individual patch sites.

Creation Methods

Key Phrase	*Form*	Description
product of <license>	*Plain*	Returns BES product objects obtained from the product fields of the specified license. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
computer count of <bes product>	*Plain*	<integer>	Returns the number of computers allowed under the license terms of the specified BES product. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
expiration date of <bes product>	*Plain*	<date>	Returns the expiration date for the specified bes product. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
name of <bes product>	*Plain*	<string>	Returns the name of the specified licensed BES product. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
non windows server count of <bes product>	*Plain*	<integer>	Returns the number of non-Windows servers included in the license for the specified BES Product. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
site url of <bes product>	*Plain*	<string>	Returns a list of the URLs associated with the specified BES product. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
windows server count of <bes product>	*Plain*	<integer>	Returns the number of Windows Servers licensed for the specified product. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
workstation count of <bes product>	*Plain*	<integer>	Returns the number of workstations licensed for the specified product. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Environment Objects

The environment objects are provided to access environment variables. Note that you are inspecting the environment of the application executing the relevance clause (typically the BigFix Agent/Client), which may or may not match the environment of other applications on the computer.

Environment

Environment variables define a particular set of paths and variables for a computer or an application. These Inspectors let you examine this set.

Creation Methods

Key Phrase	*Form*	Description
environment	*PlainGlobal*	Creates the one and only environment object. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
environment of <process>	*Plain*	Returns the environment of the specified process. Lin:8.0, Sol:8.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
variable <string> of <environment>	*Named*	<environment variable>	Returns an environment variable that matches the given name. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
variable of <environment>	*Plain*	<environment variable>	Iterates through all the environment variables defined. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

Examples

- `exists environment`
▸ TRUE if the computer has an environment object.

Environment Variable

Every variable defined by the environment has both a name and a value. Both names and values are treated as strings.

Creation Methods

Key Phrase	Form	Description
variable <string> of <environment>	*Named*	Creates the variable of the environment matching the name provided. The capitalization of the name is ignored. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
variable of <environment>	*Plain*	Iterates through all the environment variables defined. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

Properties

Key Phrase	Form	Return Type	Description
<environment variable> as string	*Cast*	<string>	Casting the variable as a string yields a string containing the variable name and the value of the variable separated by ' = '. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
name of <environment variable>	*Plain*	<string>	Returns the name of the variable. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
value of <environment variable>	*Plain*	<string>	Returns the value of the variable. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

Examples

- ■ `exists variable "PATH" of environment`
- ▶ TRUE if a path variable has been defined in this environment.

- ■ `number of variables of environment`
- ▶ Returns the total number of variables in this environment.

Authorization Objects

These inspectors retrieve security and access settings.

Client_cryptography

These Inspectors expose cryptographic properties exclusive to the client.

Creation Methods

Key Phrase	Form	Description
client cryptography	*PlainGlobal*	This Inspector is similar to the core cryptography object except that it returns properties exclusive to the client (whereas <cryptography> is also available in the Console/Web Reports contexts). Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
desired encrypt report of <client_cryptography>	*Plain*	<boolean>	Returns TRUE if the client is configured to attempt to encrypt reports. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
encrypt report failure message of <client_cryptography>	*Plain*	<string>	If the client is not successfully encrypting reports, this Inspector returns the failure message. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
encrypt report of <client_cryptography>	*Plain*	<boolean>	Returns TRUE if the client is successfully encrypting reports. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

X509 Certificate

X.509 is a public key infrastructure standard, specifying formats for public key certificates and revocations. These Inspectors interpret the certificate from a file in the PEM format. They can be used to analyze encryption credentials on decrypting relays or root servers.

Creation Methods

Key Phrase	Form	Description
encryption certificate of <license>	*Plain*	Provides the encryption certificate that is currently active and which will be used by clients to encrypt reports. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

User Objects

These Inspectors return information about local and current user accounts, including names, logins, passwords and more.

User

These Inspectors allow you to list properties of all users, whether they are logged in or not.

Type Derivation: This object type is derived from the <security account> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
current user	*PlainGlobal*	Creates an object corresponding to the current users, whether they are logged in or not. Lin, Sol, HPUX, AIX, Ubu
user	*PlainGlobal*	Creates objects for all users, logged in or not. Win:8.1, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
user <string>	*NamedGlobal*	Returns an object representing the user (logged in or not) specified by <string>. Win:8.1, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Properties

Key Phrase	Form	Return Type	Description
name of <user>	*Plain*	<string>	Returns the name of all the specified user, whether logged in or not. Win:8.1, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tty of <user>	*Plain*	<string>	Returns the tty of the user. Lin, Sol, HPUX, AIX, Ubu

Examples

- ■ `names of users`
- ▶ Returns a list of all the users.

Action Objects

These are the keywords associated with properties that can be inspected while BigFix Actions are being executed.

Action

These are the keywords associated with properties available for inspection during the execution of BigFix Actions.

Creation Methods

Key Phrase	Form	Description
action	*PlainGlobal*	Creates an action object corresponding to the BigFix Action currently being parsed. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
action <integer>	*NumberedGlobal*	Creates an action object matching the <integer> id. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
active action	*PlainGlobal*	Creates an action object corresponding to the currently executing action. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
relevant offer action of <site>	Plain	Returns the list of relevant actions that are offers for the specified site. This Inspector could be useful in a client UI dashboard listing the current set of relevant offers. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
active of <action>	*Plain*	<boolean>	Returns TRUE if the action is currently running (active). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
active start time of <action>	*Plain*	<time>	Returns the time the action started. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
complete time of <action>	*Plain*	<time>	Returns the time the action completed. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
constrained of <action>	*Plain*	<boolean>	Returns TRUE if action is unable to run yet. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Return Type	Description
exit code of \<action\>	*Plain*	\<integer\>	Returns an integer corresponding to the exit code of the specified action. This value will not exist if the action has not yet produced an exit code. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
group leader of \<action\>	*Plain*	\<boolean\>	Returns TRUE if the action is a group action and the action component is the group leader. When you deploy a mult-action from the BES Console, it constructs a group action with a group leader to control the overall behavior of the action. This inspector is used internally to manage the progress of the group action. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
id of \<action\>	*Plain*	\<integer\>	Returns the numeric ID associated with the specified Action. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
last change time of \<action\>	*Plain*	\<time\>	Returns the time when the action state last changed. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
offer accepted of \<action\>	*Plain*	\<boolean\>	Returns TRUE when users indicated they want to run the action by accepting the offer presented by the BES Client UI. When an offer has been accepted, the Client evaluates its constraints and runs as soon as conditions allow. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
offer of \<action\>	*Plain*	\<boolean\>	Returns TRUE when the Action is presented as an offer (as indicated by the header "x-offer: 1"). Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
origin fixlet id of \<action\>	*Plain*	\<integer\>	Returns the Fixlet id that contained the action. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
parameter \<string\> of \<action\>	*Named*	\<string\>	Returns the value of parameter \<string\> for the active Action. Parameters only live as long as the action is active. Among the inspectable parameters is the 'action issue date' that is added to each Action by the BigFix Console at issue time. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pending login of \<action\>	*Plain*	\<boolean\>	Returns TRUE if the specified action included an 'action requires login' command, and a login has not yet occurred since the action has run. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
pending of <action>	*Plain*	<boolean>	Returns TRUE if action is available to run. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pending restart of <action>	*Plain*	<boolean>	Returns TRUE if the specified action included an 'action requires restart' command and a restart has not occurred since the action has run. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pending time of <action>	*Plain*	<time>	Returns the time the action became pending. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
status of <action>	*Plain*	<string>	Returns one of the following strings: • Running = when the action is currently active. • Executed = no longer relevant and action has completed. • Not Relevant = action was not relevant. • Waiting = action is relevant, but waiting to run. • Not Executed = action is relevant, unconstrained, but has not yet started. • Failed = action is relevant, unconstrained, has completed, but is still relevant. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
waiting for download of <action>	*Plain*	<boolean>	Returns TRUE if client is waiting for mirroring server to have downloads required by the action. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Examples

■ `parameter "action issue date" of action`

▶ This Inspector returns the date the action was issued, a parameter added to each action by the BigFix Console.

Networking Objects

This chapter includes the various networking Inspectors.

Network

These are the keywords used to query the local network configuration.

Creation Methods

Key Phrase	Form	Description
network	*PlainGlobal*	Creates an object containing properties of the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	Form	Return Type	Description
adapter of <network>	*Plain*	<network adapter>	Returns the one or more network adapter objects of the network. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
any adapter of <network>	*Plain*	<network adapter>	This Inspector returns the same as 'adapter of <network>', but it includes loopback and tunnels. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
find adapter <string> of <network>	*Named*	<network adapter>	This Inspector lets you find a network adapter from the "Friendly Name". Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
interface <integer> of <network>	*Numbered*	<network interface>	Returns the Nth interface of the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
interface of <network>	*Plain*	<network interface>	Returns all the interfaces of the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
ip interface <integer> of <network>	*Numbered*	<network ip interface>	Returns the Nth ip interface of the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
ip interface of <network>	*Plain*	<network ip interface>	Returns all the ip interfaces of the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
ipv4 interface of <network>	*Plain*	<network adapter interface>	Returns an IPv4 network adapter interface from the specified network. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	*Form*	Return Type	Description
ipv4or6 interface of <network>	*Plain*	<network adapter interface>	Returns all the ipv4or6 network adapter interfaces from the specified network. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv6 interface of <network>	*Plain*	<network adapter interface>	Returns all the ipv6 interfaces of the specified network. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Examples

- ■ `address of find adapter "Local Area Connection" of network`
- ▶ Returns an IP address corresponding to the Local Area Connection of the network.

Network Interface

The network interface object describes a generic network interface, and has information about the name and family of that interface. On the Mac these are commonly of type AF_INET, AF_LINK and AF_INET6.

Creation Methods

Key Phrase	*Form*	Description
interface <integer> of <network>	*Numbered*	Creates an object with the specified network interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
interface of <network>	*Plain*	Creates an object with all the interfaces of the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
family of <network interface>	*Plain*	<integer>	Returns an family designator of the address family (i.e., 2=AFI_NET). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Examples

- ■ `names of interfaces of network`
- ▶ Returns a list of the network interface names, for example, lo0, gif0, stf0, en0.

Network Ip Interface

In general, the network ip interface object holds locally determined properties of logical network devices configured on the computer. On the Mac, these correspond to interfaces of type AF_INET. The properties that are available depend on the socket support installed on the computer. For Windows computers with winsock 2 support installed, for instance, the information is obtained by an ioctl call and includes Interface address, Interface broadcast address, Interface network mask, Broadcast support flag, Multicast support flag, Loopback interface flag and Point to point interface flag.

Type Derivation: This object type is derived from the <network interface> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	*Form*	Description
ip interface <integer> of <network>	*Numbered*	Creates an object with the specified ip interface of the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
ip interface of <network>	*Plain*	Creates an object or an object list (using the plural keyword) with all the ip interfaces of the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
address of <network ip interface>	*Plain*	<ipv4 address>	Returns the ip address of the ip interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
alias of <network ip interface>	*Plain*	<boolean>	Returns TRUE if the network ip interface has an alias defined for it (a virtual device, rather than a physical device). Lin, Sol, HPUX, AIX, Mac:8.0, Ubu
broadcast address of <network ip interface>	*Plain*	<ipv4 address>	Returns the broadcast address of the specified interface as an IPv4 type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
broadcast support of <network ip interface>	*Plain*	<boolean>	Indicates that broadcast messages are supported by the ip interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
cidr address of <network ip interface>	*Plain*	<string>	Returns the Classless Inter-Domain Routing address for the specified network ip interface as a string type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	Form	Return Type	Description
cidr string of <network ip interface>	Plain	<string>	Returns the Classless Inter-Domain Routing value for the specified network ip interface as a string type. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
loopback of <network ip interface>	Plain	<boolean>	Indicates that the particular network ip interface is a loopback interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
mac address of <network ip interface>	Plain	<string>	Returns the MAC address (AKA hardware address) of the network ip interface object. The mac address is formatted as a string of lower case hex digits separated by '-'. Lin, Sol, HPUX, AIX, Mac:8.0, Ubu
multicast support of <network ip interface>	Plain	<boolean>	Indicates that multicast messages are supported by the ip interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
name of <network ip interface>	Plain	<string>	Returns the name of the network ip interface object. Typical names are lan0, lo0. Virtual interfaces are usually of the form lan0:2. Lin, Sol, HPUX, AIX, Mac:8.0, Ubu
point to point of <network ip interface>	Plain	<boolean>	Indicates that the interface is a point-to-point interface. Usually TRUE for dialup connections. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
subnet address of <network ip interface>	Plain	<ipv4 address>	Returns the subnet address (IPv4) to which the specified interface belongs. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
subnet mask of <network ip interface>	Plain	<ipv4 address>	Returns the subnet mask (IPv4) of the specified network ip interface. • As of version 8.0, this Inspector type is derived from an <ipv4or6 address> type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
up of <network ip interface>	Plain	<boolean>	Returns TRUE if the specified network IP interface is currently up. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Examples

■ `names of ip interfaces of network`

▶ Returns a list of the names of the network IP interfaces, for example, lo0, en0.

- `addresses of ip interfaces of network`
▶ Returns a list of the IP addresses of the network IP interfaces, for example, 127.0.0.1, 192.168.1.100, etcetera.

- `address of ip interface whose (loopback of it = false) of network = "192.168.127.127"`
▶ Returns TRUE if the given IP address doesn't have loopback.

- `mac address whose (it = "00-61-b1-d1-7d-29") of ip interfaces of network`
▶ Returns the mac address of the specified network ip interface object.

Network Adapter

One or more network adapters may be inspected using this property of the network object. Each network adapter has a number of interesting properties such as the MAC address.

Creation Methods

Key Phrase	Form	Description
adapter of <network adapter interface>	Plain	Returns the adapters associated with the specified network adapter interface. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
adapter of <network>	Plain	Returns one or more adapters of the network. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
any adapter of <network>	Plain	This Inspector returns the same as 'adapter of <network>', but it includes loopback and tunnels. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
find adapter <string> of <network>	Named	This Inspector lets you find a network adapter from the "Friendly Name". Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
address of <network adapter>	Plain	<ipv4 address>	Returns the ip address of the network adapter (returns the first address if it is a list). Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
cidr address of <network adapter>	Plain	<string>	Returns the CIDR address of the specified network adapter as a string type, for example, 192.168.0.0/16 (IPv4) or 2001:db8::/32 (IPv6). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	Form	Return Type	Description
cidr string of <network adapter>	*Plain*	<string>	Returns the Classless Inter-Domain Routing value for the specified network adapter as a string value. Win:7.1, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
friendly name of <network adapter>	*Plain*	<string>	Returns a user-friendly name for the adapter, for example "Local Area Connection 1". Win:7.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, WM, Ubu
ipv4 interface of <network adapter>	*Plain*	<network adapter interface>	Returns the IPv4 interface of the specified network adapter as a <network adapter ip interface> type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv4or6 interface of <network adapter>	*Plain*	<network adapter interface>	Returns the ipv4or network adapter interface from the specified network adapter. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv6 interface of <network adapter>	*Plain*	<network adapter interface>	Returns the IPv6 interfaces of the specified network adapter as a network adapter interface type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
loopback of <network adapter>	*Plain*	<boolean>	Returns TRUE if the specified network adapter is a loopback interface. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, Ubu
mac address of <network adapter>	*Plain*	<string>	Returns the mac address of the network adapter. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
multicast support of <network adapter>	*Plain*	<boolean>	Returns TRUE if multicast messages are supported by the specified network adapter. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, Ubu
name of <network adapter>	*Plain*	<string>	Returns the name of the network adapter. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
subnet address of <network adapter>	*Plain*	<ipv4 address>	Returns the subnet address (IPv4) of the specified network adapter. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
subnet mask of <network adapter>	*Plain*	<ipv4 address>	Returns the subnet mask (IPv4) of the specified network adapter. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
up of <network adapter>	*Plain*	<boolean>	Returns TRUE if the specified network adaoter is currently working. Interfaces like wifi may be turned it off to save power, but this Inspector will still tell you if it is active. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, Ubu

Network Adapter Interface

A network adapter interfaces a computer to a network. These Inspectors expose the adapter so that you can determine its properties, such as its address, subnet mask, mac address and whether or not it supports broadcast, multicast or point-to-point.

Creation Methods

Key Phrase	Form	Description
ipv4 interface of <network adapter>	*Plain*	Returns the IPv4 interface of the specified network adapter as a <network adapter ip interface> type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv4 interface of <network>	*Plain*	Returns an IPv4 network adapter interface from the specified network. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv4or6 interface of <network adapter>	*Plain*	Returns the ipv4or network adapter interface from the specified network adapter. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv4or6 interface of <network>	*Plain*	Returns all the ipv4or6 network adapter interfaces from the specified network. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv6 interface of <network adapter>	*Plain*	Returns the IPv6 interfaces of the specified network adapter as a network adapter interface type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv6 interface of <network>	*Plain*	Returns all the ipv6 interfaces of the specified network. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
adapter of <network adapter interface>	*Plain*	<network adapter>	Returns the adapters associated with the specified network adapter interface. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
address of <network adapter interface>	*Plain*	<ipv4or6 address>	Returns the IP address of the specified network adapter interface as an ipv4or6 address type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
broadcast address of <network adapter interface>	*Plain*	<ipv4or6 address>	Creates an object with the broadcast address (ipv4or6) of the specified network adapter interface. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	Form	Return Type	Description
broadcast support of <network adapter interface>	*Plain*	<boolean>	Returns TRUE if the given network adapter interface has broadcast support. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
cidr address of <network adapter interface>	*Plain*	<string>	Returns the CIDR address of the specified interface as a string type, for example, 192.168.0.0/16 (IPv4) or 2001:db8::/32 (IPv6). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
cidr string of <network adapter interface>	*Plain*	<string>	A cidr string (see CIDR_notation at Wikipedia) is a string representation of a cidr address. It looks like an ip address followed by a slash and then the number of leading non-zero bits of the routing prefix. For example, 192.168.0.0/16 for IPv4, and 2001:db8::/32 for IPv6. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
loopback of <network adapter interface>	*Plain*	<boolean>	Returns TRUE if the specified interface supports loopbacks. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
mac address of <network adapter interface>	*Plain*	<string>	Returns the MAC address of the specified network adapter interface as a string type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
multicast support of <network adapter interface>	*Plain*	<boolean>	Returns TRUE if the specified interface supports multicasting. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
point to point of <network adapter interface>	*Plain*	<boolean>	A network adapter interface can be a point-to-point interface, such as you might use for a VPN connection or a SLIP connection. This Inspector returns TRUE if the specified network adapter interface is configured to run point-to-point. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
subnet address of <network adapter interface>	*Plain*	<ipv4or6 address>	Returns the subnet address of the specified interface as an ipv4or6 address type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
subnet mask of <network adapter interface>	*Plain*	<ipv4or6 address>	Returns the subnet mask of the specified interface as an ipv4or6 address type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
up of <network adapter interface>	*Plain*	<boolean>	Returns TRUE if the specified interface is currently up and working. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Ipv4 Address

This is an Internet Protocol address, version 4. IP addresses are composed of four single-byte integers separated by periods, such as "192.5.0.7".

- NOTE: Prior to version 8.0 of BigFix, this was not a derived type.

Type Derivation: This object type is derived from the <ipv4or6 address> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
address of <network adapter>	*Plain*	Returns the ip address of the network adapter. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
address of <network ip interface>	*Plain*	Creates an object with the ip address of the interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
broadcast address of <network ip interface>	*Plain*	Returns the broadcast address of the specified interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
subnet address of <network adapter>	*Plain*	Returns the subnet address (IPv4) of the specified network adapter. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
subnet address of <network ip interface>	*Plain*	Creates an object with the subnet address of the network interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
subnet mask of <network adapter>	*Plain*	Returns the subnet mask of the network adapter. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
subnet mask of <network ip interface>	*Plain*	Returns the subnet mask (IPv4) of the specified network ip interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Operators

Key phrase	Return Type	Description
<ipv4 address> {cmp} <ipv4 address>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: - {cmp} is one of: =, !=, <, <=, >, >= . Win, Lin, Sol, HPUX, AIX, Mac, WM

Key phrase	Return Type	Description
<ipv4 address> {cmp} <string>	<boolean>	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, Lin, Sol, HPUX, AIX, Mac, WM

Examples

■ `exists ip interface whose (address of it = "127.0.0.1" and loopback of it) of network`

▶ Returns TRUE if the specified ip interface (with loopback) exists on this computer.

■ `addresses of ip interfaces of network`

▶ Returns a list of IP addresses configured on the machine.

Ipv4or6 Address

These Inspectors allow you to represent IPv4 and IPv6 addresses as a common type. From these inclusive Inspectors, you can derive the corresponding v4 and v6 IP addresses.

Creation Methods

Key Phrase	Form	Description
address of <network adapter interface>	Plain	Returns the ipv4or6 address of the specified network adapter interface. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
broadcast address of <network adapter interface>	Plain	Creates an object with the broadcast address (ipv4or6) of the specified network adapter interface. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
full gateway address of <selected server>	Plain	During relay selection, a traceroute-like list of the hops between the client and its relay (the selected server) is recorded. That list is accessible through this Inspector. Unlike the 'gateway address' Inspector, this Inspector includes hops that don't reply as 0.0.0.0. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
gateway address <integer> of <selected server>	Numbered	During relay selection, a traceroute-like list of the hops between the client and its relay (the selected server) is recorded. The elements of that list is accessible through this Inspector. • Prior to version 8.0, this inspector returned an <ipv4 address> type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	Form	Description
gateway address of <selected server>	*Plain*	During relay selection, a traceroute-like list of the hops between the client and its relay (the selected server) is recorded. That list is accessible through this Inspector. However, this Inspector ignores hops that don't reply. If you need the full list, use the 'full gateway address' Inspector. • Prior to version 8.0, this inspector returned an <ipv4 address> type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
ip address of <selected server>	*Plain*	The ipv4or6 address to which reports are sent. • Prior to version 8.0, this inspector created an <ipv4 address> type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
registration address of <client>	*Plain*	This Inspector returns the IP address (as an <ipv4or6 address> type) that the specified BigFix client registered with. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
registration subnet address of <client>	*Plain*	This Inspector returns the subnet address (as an <ipv4or6 address> type) from the adapter that the specified BigFix client registered with. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
subnet address of <network adapter interface>	*Plain*	Returns the subnet address of the specified interface as an ipv4or6 address type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
subnet mask of <network adapter interface>	*Plain*	Returns the subnet mask of the specified interface as an ipv4or6 address type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Installed System Software

These Inspectors help you manage system objects and software packages.

Rpmdatabase

These are the Linux RPM Package Manager Inspectors. The RPM database contains entries for all the installed programs on this computer.

CAUTION: The RedHat RPM database Inspectors can contend with other RPM applications (including the RPM command itself) for access to the RPM database. The database APIs use a locking scheme and any application that holds the lock at the time it crashes can block subsequent applications from accessing the database until the lock is cleared. To disable the RPM Inspectors, set _BESClient_Inspector_DisableRPM to 1. A Fixlet or Task that uses a disabled inspector will report false; retrieved properties that request a disabled inspector value will report an error. For the latest information on issues surrounding the RPM inspectors, search the BigFix support knowledge base.

Creation Methods

Key Phrase	Form	Description
rpm	*PlainGlobal*	Creates an object representing the RPM database of the computer. Lin
rpm <string>	*NamedGlobal*	Creates an object representing a named RPM database of the computer. Lin

Properties

Key Phrase	Form	Return Type	Description
capability <string> of <rpmdatabase>	*Named*	<capability>	Returns a capability object matching this string if some package in the RPM database provides that capability. Lin
installed <string> of <rpmdatabase>	*Named*	<boolean>	Returns TRUE if some package in the RPM database provides the capability represented by this string. Lin
package <string> of <rpmdatabase>	*Named*	<package>	Returns the named package from the given RPM database. Lin

Key Phrase	*Form*	Return Type	Description
package conflicting with <capability> of <rpmdatabase>	*Index<capability>*	<package>	Returns a package in the RPM database that conflicts with this capability. If any packages are returned, then the rpm command will not allow any package providing this capability to be installed. Lin
package installing <capability> of <rpmdatabase>	*Index<capability>*	<package>	Finds the package that installed the specified capability. Lin
package of <rpmdatabase>	*Plain*	<package>	Returns all the packages in the given RPM database. Lin
package providing <capability> of <rpmdatabase>	*Index<capability>*	<package>	Returns a package in the RPM database that provides this capability. If no packages are returned, then the rpm command will not allow any packages requiring this capability to be installed. Lin
package requiring <capability> of <rpmdatabase>	*Index<capability>*	<package>	Returns a package in the RPM database that requires this capability. If any packages are returned, then the rpm command will not allow any installed package that provides this capability to be uninstalled. Lin

Examples

- ```
  package conflicting with (capability "apache2") of rpm
  ```
  ▶ Returns a string like "apache-1.3.23-88", indicating which package(s) conflict with the specified capability.

- ```
  package installing (capability "/opt/BESClient/BESLib/libEngine.so") of rpm
  ```
  ▶ Returns BESAgent-4.1.4.2-1, the package that installed the specified capability.

- ```
  package providing (capability "ssh") of rpm
  ```
  ▶ Returns a string like "openssh-3.0.2p1-162", indicating which package(s) provide the specified capability.

- ```
  package requiring (capability "perl") of rpm
  ```
  ▶ Returns a string like "apache-1.3.23-88", which is an application package that requires Perl.

Package

This filesystem object can inspect the propeties of an RPM (RPM Package Manager) package. Each package contains information about the program, including name and version.

Creation Methods

Key Phrase	Form	Description
package <string> of <rpmdatabase>	*Named*	Creates a named package from the RPM database. Lin
package conflicting with <capability> of <rpmdatabase>	*Index<capability>*	Returns a package in the RPM database that conflicts with this capability. If any packages are returned, then the rpm command will not allow any package providing this capability to be installed. Lin
package installing <capability> of <rpmdatabase>	*Index<capability>*	Finds the package that installed the specified capability. Lin
package of <rpmdatabase>	*Plain*	Returns all the packages in the RPM database. Lin
package providing <capability> of <rpmdatabase>	*Index<capability>*	Returns a package in the RPM database that provides this capability. If no packages are returned, then the rpm command will not allow any packages requiring this capability to be installed. Lin
package requiring <capability> of <rpmdatabase>	*Index<capability>*	Returns a package in the RPM database that requires this capability. If any packages are returned, then the rpm command will not allow any installed package that provides this capability to be uninstalled. Lin

Properties

Key Phrase	Form	Return Type	Description
<package> as string	*Cast*	<string>	Creates a string containing the package's name, version and release. Lin
architecture of <package>	*Plain*	<string>	The architecture represents the CPU type that the RPM Package was designed to be used on. Typical values are i386, i686, or x86_64, but packages can be created with archtectures like 'noarch' or have no architecture specified at all. Lin:7.0

Key Phrase	*Form*	Return Type	Description
conflict of <package>	*Plain*	<capability>	Returns capability objects that conflict with this package in the RPM database. Lin
installed file of <package>	*Plain*	<capability>	The list of actual files that the package leaves installed. Lin
name of <package>	*Plain*	<string>	Returns the name of the given RPM package. Lin
obsolete of <package>	*Plain*	<capability>	Returns a capability object that this package obsoletes. Lin
provide of <package>	*Plain*	<capability>	Returns capability objects for each capability that this package provides. Lin
require of <package>	*Plain*	<capability>	Returns capability objects for each capability that this package requires. Lin
rpm version record of <package>	*Plain*	<rpm package version record>	Returns the RPM version records of the specified package. Lin:8.0
unique name of <package>	*Plain*	<string>	Returns the unique values of a given list of <package> types, removing duplicates and sorting by value. Lin:8.0
version of <package>	*Plain*	<version>	Returns the version of the given RPM package. Lin

Examples

- ```version of package "glibc" of rpm```
  ▶ Returns the version of the glibc package of the rpm database.

- ```package conflicting with (capability "nonfreessh") of rpm```
  ▶ Returns a string like "openssh-3.0.2p1-162", indicating which package(s) conflict with the specified capability.

- ```package installing (capability "/opt/BESClient/BESLib/libEngine.so") of rpm```
  ▶ Returns BESAgent-4.1.4.2-1, the package that installed the specified capability.

- ■ `package providing (capability "sh") of rpm`
- ▶ Returns a string like "bash-2.05-321", indicating which package(s) provide the specified capability.

- ■ `package requiring (capability "iptables") of rpm`
- ▶ Returns a string like "personal-firewall-1.2-68", which is an application package that requires iptables.

- ■ `package "apache" of rpm as string`
- ▶ Returns a string with information about the package, such as "apache-1.3.23-88".

- ■ `architecture of package "BESAgent" of rpm`
- ▶ On a SuSE Linux Enterprise Server 9.0, 64 bit, this will return x86_64, while on a Fedora Core 3, 32 bit, it will return i386.

- ■ `conflict of package "openssh" of rpm`
- ▶ Returns a string, such as "nonfreessh". The openssh program is specifying a conflict with nonfreessh.

- ■ `obsolete of package "mod_php4" of rpm`
- ▶ Returns a string such as "mod_php", which a capability that is obsoleted by the current package.

- ■ `require of package "apache" of rpm`
- ▶ Returns a string such as "/bin/sh" to indicate a requirement of the package.

Capability

A package can have capabilities that it requires, provides, obsoletes, or conflicts with. For instance, apache may require sh, and provide httpd. If you want to install apache, you will need to have sh in the RPM database. If you want to install a package that uses httpd, you will need to install apache.

Creation Methods

Key Phrase	*Form*	Description
capability <string>	*NamedGlobal*	Converts a string to a capability object for rpm dependency version comparisons. Strings with version information take the form capability relation version, where relation is one of {<,>,>=,<=,=}. If a relation and version are not specified, then comparisons will assume a capability at any version number. Lin
capability <string> of <rpmdatabase>	*Named*	Returns a capability object matching this string if some package in the RPM database provides that capability. <string> refers to built-in properties of the installed rpm version itself, not normal package capabilities. Lin

Key Phrase	Form	Description
conflict of <package>	*Plain*	Returns capability objects that conflict with this package in the RPM database. Lin
installed file of <package>	*Plain*	The list of actual files that the package leaves installed. Lin
obsolete of <package>	*Plain*	Returns a capability object that this package obsoletes. Lin
provide of <package>	*Plain*	Returns capability objects for each capability that this package provides. Lin
require of <package>	*Plain*	Returns capability objects for each capability that this package requires. Lin

Properties

Key Phrase	Form	Return Type	Description
<capability> as string	*Cast*	<string>	Creates a string containing the capability's name, relation and version. Lin
name of <capability>	*Plain*	<string>	Returns the name of this capability object. Lin
relation of <capability>	*Plain*	<string>	If this capability object has version information, the relation is returned. Lin
version of <capability>	*Plain*	<string>	If this capability object has version information, the version is returned. Lin

Operators

Key phrase	Return Type	Description
<capability> contains <capability>	<boolean>	Returns a boolean TRUE if the first capability satisfies the dependency specified by the second capability. A capability without version information is interpreted as a capability over all versions. If a package requires a capability not contained in the RPM database, then the rpm command will not allow that package to be installed. Lin

Examples

- ■ `capability "anything"`
- ▶ Returns "anything". You can create any capability of a world object.

- ■ `capability "apache" of rpm`
- ▶ If installed, returns "apache". You can only create capability of rpm objects if the capability is actually in the RPM database.

- ■ `conflict of package "apache" of rpm`
- ▶ Returns a string, such as "apache2". The apache package specifies that it conflicts with apache2, so rpm should not install apache if apache2 is on the system.

- ■ `installed files of package "BESAgent" of rpm`
- ▶ Returns a list of the installed files.

- ■ `obsolete of package "openssh" of rpm`
- ▶ Returns a string such as "ssh", which is a capability that is obsoleted by the current package.

- ■ `provide of package "apache" of rpm`
- ▶ Returns a string such as "httpd", which indicates a functional capability of the apache software.

- ■ `require of package "libxslt" of rpm`
- ▶ Returns a capability, such as "libxml2 >= 2.4.7" or else, it returns no such object.

- ■ `capability "libxml2 >= 2.4.7"`
- ▶ Casts the capability as a string, returning "libxml2 >= 2.4.7".

- ■ `relation of capability "sh <= 4.1"`
- ▶ Returns "<=", the "less than or equal" relation.

- ■ `version of capability "sh <= 4.1"`
- ▶ Returns "4.1", the version of the specified capability.

- `capability "x" contains capability "x = 1.0"`
  - ▶ Returns TRUE. Having "x" at any version satisfies a requirement to have "x" at version 1.0.

- `capability "x = 1.0" contains capability "x"`
  - ▶ Returns TRUE. Having "x" at version 1.0 satisfies a requirement to have "x" at any version.

- `capability "x = 1.0" contains capability "x > 1.0"`
  - ▶ Returns FALSE. Having x = version 1.0 will not satisfy a requirement to have x > version 1.0.

- `capability "x = 1.0" contains capability "x < 2.0"`
  - ▶ Returns TRUE. Having x = version 1.0 will satisfy a requirement to have x < version 2.0.

- `capability "x" contains capability "y"`
  - ▶ Returns FALSE. A capability can only be contained/satisfied by a capability with the same name.

Rpm Package Release

These Inspectors disclose properties of the RPM package version record Inspector. They are collected from the 'Release' component of the Epoch, Version and Release fields associated with an RPM package. The behavior of this inspector differs from a string, because they are sorted according to RPM rules defined by the rpmvercmp() function of the RPM library. Don't create an <rpm package release> inspector with embedded '-' or whitespace characters. These characters are disallowed in RPM, and will cause the statement to fail.

• NOTE: Requires the presence of the RPM library and librpminfo (an open source library created by BigFix containing modified RPM library code) on the client machine. librpminfo is installed on the client as part of the normal BES client RPM installation procedure.

Creation Methods

RPM is the software package manager for Linux systems. There are package versions and releases that indicate which distro is targeted.

Key Phrase	Form	Description
maximum of <rpm package release>	*Plain*	Returns the maximum value from a list of <rpm package release> types. Lin:8.0
minimum of <rpm package release>	*Plain*	Returns the minimum value from a list of <rpm package release> types. Lin:8.0
release of <rpm package version record>	*Plain*	Returns the RPM Package release for the specified version record. Lin:8.0

Key Phrase	Form	Description
release of <short rpm package version record>	*Plain*	Returns the RPM Package release for the specified short version record. Lin:8.0
rpm package release <rpm package release>	*Index<rpm package release>Global*	Iterates through the RPM package release objects. Lin:8.0
rpm package release <string>	*NamedGlobal*	Returns the named RPM package release. Lin:8.0

Properties

Key Phrase	Form	Return Type	Description
<rpm package release> as string	*Cast*	<string>	Casts an RPM Package release object as a string type. Lin:8.0
extrema of <rpm package release>	*Plain*	<(rpm package release, rpm package release)>	Returns the minimum and maximum extreme values of the given list of RPM package releases. Lin:8.0
maximum of <rpm package release>	*Plain*	<rpm package release>	Returns the maximum value from a list of <rpm package release> types. Lin:8.0
minimum of <rpm package release>	*Plain*	<rpm package release>	Returns the minimum value from a list of <rpm package release> types. Lin:8.0
unique value of <rpm package release>	*Plain*	<rpm package release with multiplicity>	Returns the unique values of a given list of <rpm package release> types, removing duplicates and sorting by value. Lin:8.0

Operators

Key phrase	Return Type	Description
<rpm package release> {cmp} <rpm package release>	<boolean>	Compares two RPM Package releases, where {cmp} is one of: <, <=, =. Lin:8.0

Key phrase	Return Type	Description
<rpm package release> {cmp} <string>	<boolean>	Compares an RPM Package release with a string value, where {cmp} is one of: <, <=, =. Lin:8.0
<string> {cmp} <rpm package release>	<boolean>	Compares a string to an RPM Package release, where {cmp} is one of: <, <=, =. Lin:8.0

Rpm Package Release with Multiplicity

These Inspectors deal with arrays of RPM package releases, allowing you to extract unique releases and count them.

Type Derivation: This object type is derived from the <rpm package release> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
unique value of <rpm package release>	*Plain*	Returns the unique values of a given list of <rpm package release> types, removing duplicates and sorting by value. Lin:8.0

Properties

Key Phrase	Form	Return Type	Description
multiplicity of <rpm package release with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <rpm package release> types. Lin:8.0

Rpm Package Version

These Inspectors disclose properties of the RPM package version record Inspector. They are collected from the 'Version' component of the Epoch, Version and Release fields associated with an RPM package. The behavior of this inspector differs from a string, because they are sorted according to RPM rules defined by the rpmvercmp() function of the RPM library. Don't create an <rpm package release> inspector with embedded '-' or whitespace characters. These characters are disallowed in RPM, and will cause the statement to fail.

• NOTE: Requires the presence of the RPM library and librpminfo (an open source library created by BigFix containing modified RPM library code) on the client machine. librpminfo is installed on the client as part of the normal BES client RPM installation procedure.

Creation Methods

Key Phrase	*Form*	Description
maximum of <rpm package version>	*Plain*	Returns the maximum value from a list of <rpm package version> types. Lin:8.0
minimum of <rpm package version>	*Plain*	Returns the minimum value from a list of <rpm package version> types. Lin:8.0
rpm package version <rpm package version>	*Index<rpm package version>Global*	Iterates through the RPM package version objects. Lin:8.0
rpm package version <string>	*NamedGlobal*	Returns the named RPM package version. Lin:8.0
version of <rpm package version record>	*Plain*	Returns the version of the specified RPM package version record. Lin:8.0
version of <short rpm package version record>	*Plain*	Returns the version of the specified short (no epoch) RPM package version record. Lin:8.0

Properties

Key Phrase	*Form*	Return Type	Description
<rpm package version> as string	*Cast*	<string>	Casts an RPM package version as a string type. Lin:8.0

Key Phrase	Form	Return Type	Description
extrema of <rpm package version>	*Plain*	<(rpm package version, rpm package version)>	Returns the minimum and maximum extreme values of the given list of RPM package versions. Lin:8.0
maximum of <rpm package version>	*Plain*	<rpm package version>	Returns the maximum value from a list of <rpm package version> types. Lin:8.0
minimum of <rpm package version>	*Plain*	<rpm package version>	Returns the minimum value from a list of <rpm package version> types. Lin:8.0
unique value of <rpm package version>	*Plain*	<rpm package version with multiplicity>	Returns the unique values of a given list of <rpm package version> types, removing duplicates and sorting by value. Lin:8.0

Operators

Key phrase	Return Type	Description
<rpm package version> {cmp} <rpm package version>	*<boolean>*	Compares two RPM Package versions, where {cmp} is one of: <, <=, =. Lin:8.0
<rpm package version> {cmp} <string>	*<boolean>*	Compares an RPM Package version to a string, where {cmp} is one of: <, <=, =. Lin:8.0
<string> {cmp} <rpm package version>	*<boolean>*	Compares a string to an RPM Package version, where {cmp} is one of: <, <=, =. Lin:8.0

Rpm Package Version with Multiplicity

These Inspectors deal with arrays of RPM package versions, allowing you to extract unique versions and count them.

Type Derivation: This object type is derived from the <rpm package version> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
unique value of <rpm package version>	*Plain*	Returns the unique values of a given list of <rpm package version> types, removing duplicates and sorting by value. Lin:8.0

Properties

Key Phrase	Form	Return Type	Description
multiplicity of <rpm package version with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <rpm package version> types. Lin:8.0

Rpm Package Version Record

These Inspectors provide version-related information, such as Epoch, Version and Release, for an RPM package. Inspectors of this type are sorted according to an RPM-specific ordering mechanism defined by the rpmVersionCompare() function of the RPM library.

• NOTE: Requires the presence of the RPM library and librpminfo (an open source library created by BigFix containing modified RPM library code) on the client machine. librpminfo is installed on the client as part of the normal BES client RPM installation procedure.

Creation Methods

Key Phrase	Form	Description
<short rpm package version record> as rpm package version record	*Cast*	Casts a short RPM Package version record as a long version. Lin:8.0
long form of <short rpm package version record>	*Plain*	Returns an ROM Package Version record from the short version of the same. Lin:8.0

Key Phrase	Form	Description
maximum of <rpm package version record>	*Plain*	Returns the maximum value from a list of <rpm package version record> types. Lin:8.0
minimum of <rpm package version record>	*Plain*	Returns the minimum value from a list of <rpm package version record> types. Lin:8.0
no epoch of <rpm package version record>	*Plain*	Strips the epoch from the specified RPM package version record. Lin:8.0
rpm package version record <rpm package version record>	*Index<rpm package version record>Global*	Iterates through the RPM package version record objects. Lin:8.0
rpm package version record <short rpm package version record>	*Index<short rpm package version record>Global*	Iterates through the RPM package version record objects (short version). Lin:8.0
rpm package version record <string>	*NamedGlobal*	Returns the named RPM package version record. Lin:8.0
rpm version record of <package>	*Plain*	Returns the RPM version records of the specified package. Lin:8.0

Properties

Key Phrase	Form	Return Type	Description
<rpm package version record> as short rpm package version record	*Cast*	<short rpm package version record>	Casts an RPM package version record as a short version by stripping the epoch off. Lin:8.0
<rpm package version record> as string	*Cast*	<string>	Casts an RPM package version record as a string type. Lin:8.0
epoch of <rpm package version record>	*Plain*	<integer>	Returns the integer Epoch (which grants RPM more flexibility in versioning) of the specified RPM Package version record. Lin:8.0

Key Phrase	Form	Return Type	Description
extrema of <rpm package version record>	Plain	<(rpm package version record, rpm package version record)>	Returns the minimum and maximum extreme values of the given list of RPM package version records. Lin:8.0
maximum of <rpm package version record>	Plain	<rpm package version record>	Returns the maximum value from a list of <rpm package version record> types. Lin:8.0
minimum of <rpm package version record>	Plain	<rpm package version record>	Returns the minimum value from a list of <rpm package version record> types. Lin:8.0
no epoch of <rpm package version record>	Plain	<rpm package version record>	Strips the epoch from the specified RPM package version record. Lin:8.0
release of <rpm package version record>	Plain	<rpm package release>	Returns the RPM Package release for the specified version record. Lin:8.0
short form of <rpm package version record>	Plain	<short rpm package version record>	This Inspector strips of the Epoch information from the specified RPM package version record object. Lin:8.0
unique value of <rpm package version record>	Plain	<rpm package version record with multiplicity>	Returns the unique values of a given list of <rpm package version record> types, removing duplicates and sorting by value. Lin:8.0
version of <rpm package version record>	Plain	<rpm package version>	Returns the version of the specified RPM package version record. Lin:8.0

Operators

Key phrase	Return Type	Description
<rpm package version record> {cmp} <rpm package version record>	<boolean>	Compares two RPM Package version records, where {cmp} is one of: <, <=, =. Lin:8.0

Key phrase	Return Type	Description
<rpm package version record> {cmp} <short rpm package version record>	<boolean>	Compares an RPM Package version record with a short RPM package version record, where {cmp} is one of: <, <=, =. Lin:8.0
<rpm package version record> {cmp} <string>	<boolean>	Compares an RPM package version record against a string, where {cmp} is one of: <, <=, =. Lin:8.0
<short rpm package version record> {cmp} <rpm package version record>	<boolean>	Compares an RPM Package version record with a short version, where {cmp} is one of: <, <=, =. Lin:8.0
<string> {cmp} <rpm package version record>	<boolean>	Compares a string to an RPM Package version record, where {cmp} is one of: <, <=, =. Lin:8.0

Rpm Package Version Record with Multiplicity

These Inspectors deal with arrays of RPM package version records, allowing you to extract unique records and count them.

Type Derivation: This object type is derived from the <rpm package version record> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	Form	Description
unique value of <rpm package version record>	Plain	Returns the unique values of a given list of <rpm package version record> types, removing duplicates and sorting by value. Lin:8.0

Properties

Key Phrase	Form	Return Type	Description
multiplicity of <rpm package version record with multiplicity>	Plain	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <rpm package version record> types. Lin:8.0

Short Rpm Package Version Record

This is the same as an RPM package version record, except that the epoch (if one exists) is not shown when converted to a string. However, even though it isn't displayed, the epoch is still retained. If you need to remove the epoch, use one of the 'no epoch' Inspectors.

Creation Methods

Key Phrase	Form	Description
<rpm package version record> as short rpm package version record	*Cast*	Casts an RPM package version record as a short version by stripping the epoch off. Lin:8.0
maximum of <short rpm package version record>	*Plain*	Returns the maximum value from a list of <short rpm package version record> types. Lin:8.0
minimum of <short rpm package version record>	*Plain*	Returns the minimum value from a list of <short rpm package version record> types. Lin:8.0
no epoch of <short rpm package version record>	*Plain*	Returns a short RPM package version record with the epoch stripped out. Lin:8.0
short form of <rpm package version record>	*Plain*	This Inspector strips of the Epoch information from the specified RPM package version record object. Lin:8.0
short rpm package version record <rpm package version record>	*Index<rpm package version record>Global*	Iterates throught the specified RPM package version record objects, stripping the Epoch information out. Lin:8.0
short rpm package version record <short rpm package version record>	*Index<short rpm package version record>Global*	Iterates throught the specified short RPM package version record objects. Lin:8.0

Properties

Key Phrase	Form	Return Type	Description
<short rpm package version record> as rpm package version record	*Cast*	<rpm package version record>	Casts a short RPM Package version record as a long version. Lin:8.0

Key Phrase	Form	Return Type	Description
<short rpm package version record> as string	*Cast*	<string>	Casts a short RPM Package version record as a string type. Lin:8.0
epoch of <short rpm package version record>	*Plain*	<integer>	Returns the integer Epoch (which grants RPM more flexibility in versioning) of the specified short version of the RPM Package version record. Lin:8.0
extrema of <short rpm package version record>	*Plain*	<(short rpm package version record, short rpm package version record)>	Returns the minimum and maximum extreme values of the given list of short RPM package version records. Lin:8.0
long form of <short rpm package version record>	*Plain*	<rpm package version record>	Returns an ROM Package Version record from the short version of the same. Lin:8.0
maximum of <short rpm package version record>	*Plain*	<short rpm package version record>	Returns the maximum value from a list of <short rpm package version record> types. Lin:8.0
minimum of <short rpm package version record>	*Plain*	<short rpm package version record>	Returns the minimum value from a list of <short rpm package version record> types. Lin:8.0
no epoch of <short rpm package version record>	*Plain*	<short rpm package version record>	Returns a short RPM package version record with the epoch stripped out. Lin:8.0
release of <short rpm package version record>	*Plain*	<rpm package release>	Returns the RPM Package release for the specified short version record. Lin:8.0
unique value of <short rpm package version record>	*Plain*	<short rpm package version record with multiplicity>	Returns the unique values of a given list of <short rpm package version record> types, removing duplicates and sorting by value. Lin:8.0

Key Phrase	*Form*	Return Type	Description
version of <short rpm package version record>	*Plain*	<rpm package version>	Returns the version of the specified short RPM package version record. Lin:8.0

Operators

Key phrase	Return Type	Description
<short rpm package version record> {cmp} <short rpm package version record>	*<boolean>*	Compares two short RPM Package version records, where {cmp} is one of: <, <=, =. Lin:8.0

Short Rpm Package Version Record with Multiplicity

These Inspectors deal with arrays of short RPM package version records, allowing you to extract unique records and count them.

Type Derivation: This object type is derived from the <short rpm package version record> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	*Form*	Description
unique value of <short rpm package version record>	*Plain*	Returns the unique values of a given list of <short rpm package version record> types, removing duplicates and sorting by value. Lin:8.0

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <short rpm package version record with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <short rpm package version record> types. Lin:8.0

Power Inspectors

These Inspectors return information about the energy usage patterns of BigFix Clients and their attached monitors.

Power Level

These Inspectors provide exposure to the underlying batter and power information used by low-power modes. On Windows, this uses the GetSystemPowerStatus system call, and on OSX, it uses the IOPSCopyPowerSourcesList functionality.

Creation Methods

Key Phrase	Form	Description
power level	PlainGlobal	Returns a power level representing the underlying state of the battery or charging system. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	Form	Return Type	Description
<power level> as string	Cast	<string>	Converts a power level into a human-readable string. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
full of <power level>	Plain	<boolean>	Returns TRUE if the battery is fully charged. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
low of <power level>	Plain	<boolean>	Returns TRUE if the battery is at a low charge level. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
normal of <power level>	Plain	<boolean>	Returns TRUE if the battery is at a normal charge level. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
plugged of <power level>	Plain	<boolean>	Returns TRUE if the computer is currently plugged in to AC power. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ups of <power level>	Plain	<boolean>	Returns TRUE if the computer is currently being powered by a UPS. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Examples

- `power level`
▶ Returns the current power level, which might be something like "full battery power plugged in".

- `full of power level`
▶ Returns TRUE if the battery is currently topped off.

- `plugged of power level`
▶ Returns TRUE if the computer is currently plugged in.

Key Phrases (Inspectors)

This section of the guide provides an alphabetical list of the Inspector keywords. It details the *context* object type (From an object), and the *resulting* object type (Creates an object). This list includes all Inspectors that are relevant to the context of the current guide, including the core and regex Inspectors. You can retrieve any Inspector defined in this guide by clicking on its link in the right column.

Key Phrase	Plural	Creates a	From a	*Form*	Ref
abbr <string> of <html>	abbrs	<html>	<html>	*Named*	core
abbr <string> of <string>	abbrs	<html>	<string>	*Named*	core
abbr of <html>	abbrs	<html>	<html>	*Plain*	core
abbr of <string>	abbrs	<html>	<string>	*Plain*	core
absolute value of <hertz>	absolute values	<hertz>	<hertz>	*Plain*	core
absolute value of <integer>	absolute values	<integer>	<integer>	*Plain*	core
absolute value of <time interval>	absolute values	<time interval>	<time interval>	*Plain*	core
accessed time of <filesystem object>	accessed times	<time>	<filesystem object>	*Plain*	lin
accessed time of <symlink>	accessed times	<time>	<symlink>	*Plain*	lin
acronym <string> of <html>	acronyms	<html>	<html>	*Named*	core
acronym <string> of <string>	acronyms	<html>	<string>	*Named*	core
acronym of <html>	acronyms	<html>	<html>	*Plain*	core
acronym of <string>	acronyms	<html>	<string>	*Plain*	core
action	actions	<action>	<world>	*PlainGlobal*	lin
action <integer>	actions	<action>	<world>	*NumberedGlobal*	lin
action lock state	action lock states	<action lock state>	<world>	*PlainGlobal*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
active action	active actions	<action>	<world>	*PlainGlobal*	lin
active of <action>	actives	<boolean>	<action>	*Plain*	lin
active start time of <action>	active start times	<time>	<action>	*Plain*	lin
adapter of <network adapter interface>	adapters	<network adapter>	<network adapter interface>	*Plain*	lin
adapter of <network>	adapters	<network adapter>	<network>	*Plain*	lin
address <string> of <html>	addresss	<html>	<html>	*Named*	core
address <string> of <string>	addresss	<html>	<string>	*Named*	core
address of <html>	addresss	<html>	<html>	*Plain*	core
address of <network adapter interface>	addresses	<ipv4or6 address>	<network adapter interface>	*Plain*	lin
address of <network adapter>	addresses	<ipv4 address>	<network adapter>	*Plain*	lin
address of <network ip interface>	addresses	<ipv4 address>	<network ip interface>	*Plain*	lin
address of <string>	addresss	<html>	<string>	*Plain*	core
administrator <string> of <client>	administrators	<setting>	<client>	*Named*	lin
administrator of <client>	administrators	<setting>	<client>	*Plain*	lin
alias of <network ip interface>	aliases	<boolean>	<network ip interface>	*Plain*	lin
allow unmentioned site of <license>	allow unmentioned sites	<boolean>	<license>	*Plain*	lin
ancestor of <filesystem object>	ancestors	<folder>	<filesystem object>	*Plain*	lin
ancestor of <symlink>	ancestors	<folder>	<symlink>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
anchor <string> of <html>	anchors	<html>	<html>	*Named*	core
anchor <string> of <string>	anchors	<html>	<string>	*Named*	core
anchor of <html>	anchors	<html>	<html>	*Plain*	core
anchor of <string>	anchors	<html>	<string>	*Plain*	core
any adapter of <network>	any adapters	<network adapter>	<network>	*Plain*	lin
any ip version	any ip versions	<ip version>	<world>	*PlainGlobal*	core
apparent registration server time	apparent registration server times	<time>	<world>	*PlainGlobal*	lin
application <string>	applications	<application>	<world>	*NamedGlobal*	lin
application <string> of <folder>	applications	<application>	<folder>	*Named*	lin
application usage summary	application usage summaries	<application usage summary>	<world>	*PlainGlobal*	lin
application usage summary <string>	application usage summaries	<application usage summary>	<world>	*NamedGlobal*	lin
april	aprils	<month>	<world>	*PlainGlobal*	core
april <integer>	aprils	<day of year>	<world>	*NumberedGlobal*	core
april <integer> of <integer>	aprils	<date>	<integer>	*Numbered*	core
april of <integer>	aprils	<month and year>	<integer>	*Plain*	core
architecture of <operating system>	architectures	<string>	<operating system>	*Plain*	lin
architecture of <package>	architectures	<string>	<package>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
august	augusts	<month>	<world>	*PlainGlobal*	core
august <integer>	augusts	<day of year>	<world>	*NumberedGlobal*	core
august <integer> of <integer>	augusts	<date>	<integer>	*Numbered*	core
august of <integer>	augusts	<month and year>	<integer>	*Plain*	core
available amount of <ram>	available amounts	<integer>	<ram>	*Plain*	lin
average of <evaluation cycle>	averages	<integer>	<evaluation cycle>	*Plain*	lin
b <string> of <html>	bs	<html>	<html>	*Named*	core
b <string> of <string>	bs	<html>	<string>	*Named*	core
b of <html>	bs	<html>	<html>	*Plain*	core
b of <string>	bs	<html>	<string>	*Plain*	core
base <string> of <html>	bases	<html>	<html>	*Named*	core
base <string> of <string>	bases	<html>	<string>	*Named*	core
base of <html>	bases	<html>	<html>	*Plain*	core
base of <string>	bases	<html>	<string>	*Plain*	core
bes license	bes licenses	<license>	<world>	*PlainGlobal*	lin
big <string> of <html>	bigs	<html>	<html>	*Named*	core
big <string> of <string>	bigs	<html>	<string>	*Named*	core
big of <html>	bigs	<html>	<html>	*Plain*	core
big of <string>	bigs	<html>	<string>	*Plain*	core
binary operator <string>	binary operators	<binary operator>	<world>	*NamedGlobal*	core
binary operator returning <type>	binary operators returning	<binary operator>	<world>	*Index<type>Global*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
bit <integer>	bits	<bit set>	<world>	*NumberedGlobal*	core
bit <integer> of <bit set>	bits	<boolean>	<bit set>	*Numbered*	core
bit <integer> of <integer>	bits	<boolean>	<integer>	*Numbered*	core
bit set <string>	bit sets	<bit set>	<world>	*NamedGlobal*	core
blockquote <string> of <html>	blockquotes	<html>	<html>	*Named*	core
blockquote <string> of <string>	blockquotes	<html>	<string>	*Named*	core
blockquote of <html>	blockquotes	<html>	<html>	*Plain*	core
blockquote of <string>	blockquotes	<html>	<string>	*Plain*	core
body <string> of <html>	bodys	<html>	<html>	*Named*	core
body <string> of <string>	bodys	<html>	<string>	*Named*	core
body of <html>	bodys	<html>	<html>	*Plain*	core
body of <string>	bodys	<html>	<string>	*Plain*	core
bogomips of <processor>	bogomipses	<integer>	<processor>	*Plain*	lin
boolean <string>	booleans	<boolean>	<world>	*NamedGlobal*	core
boot time of <operating system>	boot times	<time>	<operating system>	*Plain*	lin
br	brs	<html>	<world>	*PlainGlobal*	core
br <string>	brs	<html>	<world>	*NamedGlobal*	core
brand of <client>	brands	<string>	<client>	*Plain*	lin
broadcast address of <network adapter interface>	broadcast addresses	<ipv4or6 address>	<network adapter interface>	*Plain*	lin
broadcast address of <network ip interface>	broadcast addresses	<ipv4 address>	<network ip interface>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
broadcast support of <network adapter interface>	broadcast supports	<boolean>	<network adapter interface>	Plain	lin
broadcast support of <network ip interface>	broadcast supports	<boolean>	<network ip interface>	Plain	lin
buffered amount of <ram>	buffered amounts	<integer>	<ram>	Plain	lin
build of <operating system>	builds	<string>	<operating system>	Plain	lin
byte <integer> of <file>	bytes	<integer>	<file>	Numbered	lin
cached amount of <ram>	cached amounts	<integer>	<ram>	Plain	lin
capability <string>	capabilities	<capability>	<world>	NamedGlobal	lin
capability <string> of <rpmdatabase>	capabilities	<capability>	<rpmdatabase>	Named	lin
caption <string> of <html>	captions	<html>	<html>	Named	core
caption <string> of <string>	captions	<html>	<string>	Named	core
caption of <html>	captions	<html>	<html>	Plain	core
caption of <string>	captions	<html>	<string>	Plain	core
case insensitive regex <string>	case insensitive regexes	<regular expression>	<world>	NamedGlobal	regx
case insensitive regular expression <string>	case insensitive regular expressions	<regular expression>	<world>	NamedGlobal	regx
cast <string>	casts	<cast>	<world>	NamedGlobal	core
cast from of <type>	casts from	<cast>	<type>	Plain	core
cast returning <type>	casts returning	<cast>	<world>	Index<type>Global	core

Key Phrase	Plural	Creates a	From a	Form	Ref
change time of <symlink>	change times	<time>	<symlink>	*Plain*	[lin]
character <integer>	characters	<string>	<world>	*NumberedGlobal*	core
character <integer> of <string>	characters	<substring>	<string>	*Numbered*	core
character of <string>	characters	<substring>	<string>	*Plain*	core
cidr address of <network adapter interface>	cidr addresses	<string>	<network adapter interface>	*Plain*	[lin]
cidr address of <network adapter>	cidr addresses	<string>	<network adapter>	*Plain*	[lin]
cidr address of <network ip interface>	cidr addresses	<string>	<network ip interface>	*Plain*	[lin]
cidr string of <network adapter interface>	cidr strings	<string>	<network adapter interface>	*Plain*	[lin]
cidr string of <network adapter>	cidr strings	<string>	<network adapter>	*Plain*	[lin]
cidr string of <network ip interface>	cidr strings	<string>	<network ip interface>	*Plain*	[lin]
cite <string> of <html>	cites	<html>	<html>	*Named*	core
cite <string> of <string>	cites	<html>	<string>	*Named*	core
cite of <html>	cites	<html>	<html>	*Plain*	core
cite of <string>	cites	<html>	<string>	*Plain*	core
client	clients	<client>	<world>	*PlainGlobal*	[lin]
client cryptography	client cryptographies	<client_cryptography>	<world>	*PlainGlobal*	[lin]
client folder of <site>	client folders	<folder>	<site>	*Plain*	[lin]
client license	client licenses	<license>	<world>	*PlainGlobal*	[lin]

Key Phrase	Plural	Creates a	From a	*Form*	Ref
code <string> of <html>	codes	<html>	<html>	*Named*	core
code <string> of <string>	codes	<html>	<string>	*Named*	core
code of <html>	codes	<html>	<html>	*Plain*	core
code of <string>	codes	<html>	<string>	*Plain*	core
codename of <operating system>	codenames	<string>	<operating system>	*Plain*	lin
col <string> of <html>	cols	<html>	<html>	*Named*	core
col <string> of <string>	cols	<html>	<string>	*Named*	core
col of <html>	cols	<html>	<html>	*Plain*	core
col of <string>	cols	<html>	<string>	*Plain*	core
colgroup <string> of <html>	colgroups	<html>	<html>	*Named*	core
colgroup <string> of <string>	colgroups	<html>	<string>	*Named*	core
colgroup of <html>	colgroups	<html>	<html>	*Plain*	core
colgroup of <string>	colgroups	<html>	<string>	*Plain*	core
coma bug of <processor>	coma bugs	<boolean>	<processor>	*Plain*	lin
command line argument <integer> of <process>	command line arguments	<string>	<process>	*Numbered*	lin
command line argument of <process>	command line arguments	<string>	<process>	*Plain*	lin
common name of <license>	common names	<string>	<license>	*Plain*	lin
competition size of <selected server>	competition sizes	<integer>	<selected server>	*Plain*	lin
competition weight of <selected server>	competition weights	<integer>	<selected server>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
complete time of <action>	complete times	<time>	<action>	*Plain*	lin
component <integer> of <site version list>	components	<integer>	<site version list>	*Numbered*	core
computer count of <bes product>	computer counts	<integer>	<bes product>	*Plain*	lin
computer id	computer ids	<integer>	<world>	*PlainGlobal*	lin
computer name	computer names	<string>	<world>	*PlainGlobal*	lin
concatenation <html> of <html>	concatenations	<html>	<html>	*Index<html>*	core
concatenation <html> of <string>	concatenations	<html>	<string>	*Index<html>*	core
concatenation <string> of <html>	concatenations	<html>	<html>	*Named*	core
concatenation <string> of <string>	concatenations	<string>	<string>	*Named*	core
concatenation of <html>	concatenations	<html>	<html>	*Plain*	core
concatenation of <string>	concatenations	<string>	<string>	*Plain*	core
conflict of <package>	conflicts	<capability>	<package>	*Plain*	lin
conjunction of <boolean>	conjunctions	<boolean>	<boolean>	*Plain*	core
constrained of <action>	constraineds	<boolean>	<action>	*Plain*	lin
content of <file>	contents	<file content>	<file>	*Plain*	lin
controller of <action lock state>	controllers	<string>	<action lock state>	*Plain*	lin
cpuid level of <processor>	cpuid levels	<integer>	<processor>	*Plain*	lin
cryptography	cryptographies	<cryptography>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
current analysis	current analyses	<fixlet>	<world>	*PlainGlobal*	lin
current date	current dates	<date>	<world>	*PlainGlobal*	core
current day_of_month	current days_of_mo nth	<day of month>	<world>	*PlainGlobal*	core
current day_of_week	current days_of_wee k	<day of week>	<world>	*PlainGlobal*	core
current day_of_year	current days_of_yea r	<day of year>	<world>	*PlainGlobal*	core
current month	current months	<month>	<world>	*PlainGlobal*	core
current month_and_year	current months_and _years	<month and year>	<world>	*PlainGlobal*	core
current relay	current relays	<current relay>	<world>	*PlainGlobal*	lin
current site	current sites	<site>	<world>	*PlainGlobal*	lin
current time_of_day	current times_of_day	<time of day with time zone>	<world>	*PlainGlobal*	core
current time_of_day <time zone>	current times_of_day	<time of day with time zone>	<world>	*Index<time zone>Global*	core
current user	current users	<user>	<world>	*PlainGlobal*	lin
current year	current years	<year>	<world>	*PlainGlobal*	core
custom site subscription effective date <string>	custom site subscription effective dates	<time>	<world>	*NamedGlobal*	lin
date <string>	dates	<date>	<world>	*NamedGlobal*	core
date <time zone> of <time>	dates	<date>	<time>	*Index<time zone>*	core
date of <bios>	dates	<string>	<bios>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
day	days	<time interval>	<world>	*PlainGlobal*	core
day of <day of year>	days	<day of month>	<day of year>	*Plain*	core
day_of_month <integer>	days_of_month	<day of month>	<world>	*NumberedGlobal*	core
day_of_month <string>	days_of_month	<day of month>	<world>	*NamedGlobal*	core
day_of_month of <date>	days_of_month	<day of month>	<date>	*Plain*	core
day_of_week <string>	days_of_week	<day of week>	<world>	*NamedGlobal*	core
day_of_week of <date>	days_of_week	<day of week>	<date>	*Plain*	core
day_of_year of <date>	days_of_year	<day of year>	<date>	*Plain*	core
dd <string> of <html>	dds	<html>	<html>	*Named*	core
dd <string> of <string>	dds	<html>	<string>	*Named*	core
dd of <html>	dds	<html>	<html>	*Plain*	core
dd of <string>	dds	<html>	<string>	*Plain*	core
december	decembers	<month>	<world>	*PlainGlobal*	core
december <integer>	decembers	<day of year>	<world>	*NumberedGlobal*	core
december <integer> of <integer>	decembers	<date>	<integer>	*Numbered*	core
december of <integer>	decembers	<month and year>	<integer>	*Plain*	core
default web browser	default web browsers	<file>	<world>	*PlainGlobal*	lin
definition list <string> of <html>	definition lists	<html>	<html>	*Named*	core
definition list <string> of <string>	definition lists	<html>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
definition list of <html>	definition lists	<html>	<html>	*Plain*	core
definition list of <string>	definition lists	<html>	<string>	*Plain*	core
del <string> of <html>	dels	<html>	<html>	*Named*	core
del <string> of <string>	dels	<html>	<string>	*Named*	core
del of <html>	dels	<html>	<html>	*Plain*	core
del of <string>	dels	<html>	<string>	*Plain*	core
dependency known of <property>	dependencies known	<boolean>	<property>	*Plain*	core
descendant folder of <folder>	descendant folders	<folder>	<folder>	*Plain*	lin
descendant of <folder>	descendants	<file>	<folder>	*Plain*	lin
desired encrypt report of <client_cryptography>	desired encrypt reports	<boolean>	<client_cryptography>	*Plain*	lin
desired fips mode of <cryptography>	desired fips modes	<boolean>	<cryptography>	*Plain*	core
device file <filesystem object>	device files	<device file>	<world>	*Index<filesystem object>Global*	lin
device file <string>	device files	<device file>	<world>	*NamedGlobal*	lin
device file <string> of <folder>	device files	<device file>	<folder>	*Named*	lin
device file <symlink>	device files	<device file>	<world>	*Index<symlink>Global*	lin
device file of <folder>	device files	<device file>	<folder>	*Plain*	lin
device type of <device file>	device types	<string>	<device file>	*Plain*	lin
dfn <string> of <html>	dfns	<html>	<html>	*Named*	core
dfn <string> of <string>	dfns	<html>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
dfn of <html>	dfns	<html>	<html>	*Plain*	core
dfn of <string>	dfns	<html>	<string>	*Plain*	core
direct object type of <property>	direct object types	<type>	<property>	*Plain*	core
disjunction of <boolean>	disjunctions	<boolean>	<boolean>	*Plain*	core
distance of <selected server>	distances	<integer range>	<selected server>	*Plain*	lin
div <string> of <html>	divs	<html>	<html>	*Named*	core
div <string> of <string>	divs	<html>	<string>	*Named*	core
div of <html>	divs	<html>	<html>	*Plain*	core
div of <string>	divs	<html>	<string>	*Plain*	core
divided by zero of <floating point>	divided by zeroes	<boolean>	<floating point>	*Plain*	core
dmi	dmis	<dmi>	<world>	*PlainGlobal*	lin
dns name	dns names	<string>	<world>	*PlainGlobal*	lin
domain name	domain names	<string>	<world>	*PlainGlobal*	lin
domainname	domainnames	<string>	<world>	*PlainGlobal*	lin
download file <string>	download files	<file>	<world>	*NamedGlobal*	lin
download path <string>	download paths	<string>	<world>	*NamedGlobal*	lin
download storage folder	download storage folders	<download storage folder>	<world>	*PlainGlobal*	lin
drive	drives	<filesystem>	<world>	*PlainGlobal*	lin
drive <string>	drives	<filesystem>	<world>	*NamedGlobal*	lin
drive of <device file>	drives	<filesystem>	<device file>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
drive of <fifo file>	drives	<filesystem>	<fifo file>	*Plain*	lin
drive of <file>	drives	<filesystem>	<file>	*Plain*	lin
drive of <folder>	drives	<filesystem>	<folder>	*Plain*	lin
drive of <socket file>	drives	<filesystem>	<socket file>	*Plain*	lin
drive of <symlink>	drives	<filesystem>	<symlink>	*Plain*	lin
dt <string> of <html>	dts	<html>	<html>	*Named*	core
dt <string> of <string>	dts	<html>	<string>	*Named*	core
dt of <html>	dts	<html>	<html>	*Plain*	core
dt of <string>	dts	<html>	<string>	*Plain*	core
effective date of <action lock state>	effective dates	<time>	<action lock state>	*Plain*	lin
effective date of <setting>	effective dates	<time>	<setting>	*Plain*	lin
effective time of <runlevel>	effective times	<time>	<runlevel>	*Plain*	lin
element of <integer set>	elements	<integer>	<integer set>	*Plain*	core
element of <string set>	elements	<string>	<string set>	*Plain*	core
em <string> of <html>	ems	<html>	<html>	*Named*	core
em <string> of <string>	ems	<html>	<string>	*Named*	core
em of <html>	ems	<html>	<html>	*Plain*	core
em of <string>	ems	<html>	<string>	*Plain*	core
email address of <license>	email addresses	<string>	<license>	*Plain*	lin
enabled of <setting>	enableds	<boolean>	<setting>	*Plain*	lin
encrypt report failure message of <client_cryptography>	encrypt report failure messages	<string>	<client_cryptography>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
encrypt report of <client_cryptography>	encrypt reports	<boolean>	<client_cryptography>	*Plain*	lin
encryption certificate of <license>	encryption certificates	<x509 certificate>	<license>	*Plain*	lin
end of <substring>	ends	<string position>	<substring>	*Plain*	core
end of <time range>	ends	<time>	<time range>	*Plain*	core
environment	environments	<environment>	<world>	*PlainGlobal*	lin
environment of <process>	environments	<environment>	<process>	*Plain*	lin
epoch of <rpm package version record>	epochs	<integer>	<rpm package version record>	*Plain*	lin
epoch of <short rpm package version record>	epochs	<integer>	<short rpm package version record>	*Plain*	lin
error <string>	errors	<undefined>	<world>	*NamedGlobal*	core
evaluation of <license>	evaluations	<boolean>	<license>	*Plain*	lin
evaluationcycle of <client>	evaluationcycles	<evaluation cycle>	<client>	*Plain*	lin
execute of <mode_mask>	executes	<boolean>	<mode_mask>	*Plain*	lin
exit code of <action>	exit codes	<integer>	<action>	*Plain*	lin
expiration date of <action lock state>	expiration dates	<time>	<action lock state>	*Plain*	lin
expiration date of <bes product>	expiration dates	<date>	<bes product>	*Plain*	lin
expiration date of <license>	expiration dates	<time>	<license>	*Plain*	lin
expiration state of <license>	expiration states	<string>	<license>	*Plain*	lin
extrema of <date>	extremas	<(date, date)>	<date>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
extrema of <day of month>	extremas	<(day of month, day of month)>	<day of month>	*Plain*	core
extrema of <day of year>	extremas	<(day of year, day of year)>	<day of year>	*Plain*	core
extrema of <floating point>	extremas	<(floating point, floating point)>	<floating point>	*Plain*	core
extrema of <hertz>	extremas	<(hertz, hertz)>	<hertz>	*Plain*	core
extrema of <integer>	extremas	<(integer, integer)>	<integer>	*Plain*	core
extrema of <ipv4 address>	extremas	<(ipv4 address, ipv4 address)>	<ipv4 address>	*Plain*	core
extrema of <ipv4or6 address>	extremas	<(ipv4or6 address, ipv4or6 address)>	<ipv4or6 address>	*Plain*	core
extrema of <ipv6 address>	extremas	<(ipv6 address, ipv6 address)>	<ipv6 address>	*Plain*	core
extrema of <month and year>	extremas	<(month and year, month and year)>	<month and year>	*Plain*	core
extrema of <month>	extremas	<(month, month)>	<month>	*Plain*	core
extrema of <number of months>	extremas	<(number of months, number of months)>	<number of months>	*Plain*	core
extrema of <rpm package release>	extremas	<(rpm package release, rpm package release)>	<rpm package release>	*Plain*	lin
extrema of <rpm package version record>	extremas	<(rpm package version record, rpm package version record)>	<rpm package version record>	*Plain*	lin
extrema of <rpm package version>	extremas	<(rpm package version, rpm package version)>	<rpm package version>	*Plain*	lin
extrema of <short rpm package version record>	extremas	<(short rpm package version record, short rpm package version record)>	<short rpm package version record>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
extrema of <site version list>	extremas	<(site version list, site version list)>	<site version list>	*Plain*	core
extrema of <time interval>	extremas	<(time interval, time interval)>	<time interval>	*Plain*	core
extrema of <time of day>	extremas	<(time of day, time of day)>	<time of day>	*Plain*	core
extrema of <time>	extremas	<(time, time)>	<time>	*Plain*	core
extrema of <version>	extremas	<(version, version)>	<version>	*Plain*	core
extrema of <year>	extremas	<(year, year)>	<year>	*Plain*	core
f00f bug of <processor>	f00f bugs	<boolean>	<processor>	*Plain*	lin
false	falses	<boolean>	<world>	*PlainGlobal*	core
family name of <processor>	family names	<string>	<processor>	*Plain*	lin
family of <network interface>	families	<integer>	<network interface>	*Plain*	lin
family of <processor>	families	<integer>	<processor>	*Plain*	lin
fdiv bug of <processor>	fdiv bugs	<boolean>	<processor>	*Plain*	lin
february	februarys	<month>	<world>	*PlainGlobal*	core
february <integer>	februarys	<day of year>	<world>	*NumberedGlobal*	core
february <integer> of <integer>	februarys	<date>	<integer>	*Numbered*	core
february of <integer>	februarys	<month and year>	<integer>	*Plain*	core
fifo file <filesystem object>	fifo files	<fifo file>	<world>	*Index<filesystem object>Global*	lin
fifo file <string>	fifo files	<fifo file>	<world>	*NamedGlobal*	lin
fifo file <string> of <folder>	fifo files	<fifo file>	<folder>	*Named*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
fifo file <symlink>	fifo files	<fifo file>	<world>	*Index<symlink>Global*	lin
fifo file of <folder>	fifo files	<fifo file>	<folder>	*Plain*	lin
file <string>	files	<file>	<world>	*NamedGlobal*	lin
file <string> of <folder>	files	<file>	<folder>	*Named*	lin
file <symlink>	files	<file>	<world>	*Index<symlink>Global*	lin
file count of <filesystem>	file counts	<integer>	<filesystem>	*Plain*	lin
file of <folder>	files	<file>	<folder>	*Plain*	lin
filesystem	filesystems	<filesystem>	<world>	*PlainGlobal*	lin
filesystem <string>	filesystems	<filesystem>	<world>	*NamedGlobal*	lin
filesystem of <device file>	filesystems	<filesystem>	<device file>	*Plain*	lin
filesystem of <fifo file>	filesystems	<filesystem>	<fifo file>	*Plain*	lin
filesystem of <file>	filesystems	<filesystem>	<file>	*Plain*	lin
filesystem of <folder>	filesystems	<filesystem>	<folder>	*Plain*	lin
filesystem of <socket file>	filesystems	<filesystem>	<socket file>	*Plain*	lin
filesystem of <symlink>	filesystems	<filesystem>	<symlink>	*Plain*	lin
final part <time interval> of <time range>	final parts	<time range>	<time range>	*Index<time interval>*	core
find adapter <string> of <network>	find adapters	<network adapter>	<network>	*Named*	lin
find file <string> of <folder>	find files	<file>	<folder>	*Named*	lin
find folder <string> of <folder>	find folders	<folder>	<folder>	*Named*	lin
finite of <floating point>	finites	<boolean>	<floating point>	*Plain*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
fips mode failure message of <cryptography>	fips mode failure messages	<string>	<cryptography>	Plain	core
fips mode of <cryptography>	fips modes	<boolean>	<cryptography>	Plain	core
fips mode of <license>	fips modes	<boolean>	<license>	Plain	lin
first <day of week> of <month and year>	firsts	<date>	<month and year>	Index<day of week>	core
first <integer> of <string>	firsts	<substring>	<string>	Numbered	core
first <string> of <string>	firsts	<substring>	<string>	Named	core
first friday of <month and year>	first fridays	<date>	<month and year>	Plain	core
first match <regular expression> of <string>	first matches	<regular expression match>	<string>	Index<regular expression>	regx
first monday of <month and year>	first mondays	<date>	<month and year>	Plain	core
first saturday of <month and year>	first saturdays	<date>	<month and year>	Plain	core
first start time of <application usage summary instance>	first start times	<time>	<application usage summary instance>	Plain	lin
first start time of <application usage summary>	first start times	<time>	<application usage summary>	Plain	lin
first sunday of <month and year>	first sundays	<date>	<month and year>	Plain	core
first thursday of <month and year>	first thursdays	<date>	<month and year>	Plain	core
first tuesday of <month and year>	first tuesdays	<date>	<month and year>	Plain	core
first wednesday of <month and year>	first wednesdays	<date>	<month and year>	Plain	core
fixlet of <site>	fixlets	<fixlet>	<site>	Plain	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
flag list of <processor>	flag lists	<string>	<processor>	*Plain*	lin
floating point <floating point>	floating points	<floating point>	<world>	*Index<floating point>Global*	core
floating point <string>	floating points	<floating point>	<world>	*NamedGlobal*	core
folder <string>	folders	<folder>	<world>	*NamedGlobal*	lin
folder <string> of <folder>	folders	<folder>	<folder>	*Named*	lin
folder <symlink>	folders	<folder>	<world>	*Index<symlink>Global*	lin
folder of <folder>	folders	<folder>	<folder>	*Plain*	lin
following text of <string position>	following texts	<substring>	<string position>	*Plain*	core
following text of <substring>	following texts	<substring>	<substring>	*Plain*	core
fpu exception of <processor>	fpu exceptions	<boolean>	<processor>	*Plain*	lin
fpu of <processor>	fpus	<boolean>	<processor>	*Plain*	lin
free amount of <ram>	free amounts	<integer>	<ram>	*Plain*	lin
free amount of <swap>	free amounts	<integer>	<swap>	*Plain*	lin
free file count of <filesystem>	free file counts	<integer>	<filesystem>	*Plain*	lin
free percent of <filesystem>	free percents	<integer>	<filesystem>	*Plain*	lin
free space of <filesystem>	free spaces	<integer>	<filesystem>	*Plain*	lin
friday	fridays	<day of week>	<world>	*PlainGlobal*	core
friendly name of <network adapter>	friendly names	<string>	<network adapter>	*Plain*	lin
full gateway address of <selected server>	full gateway addresses	<ipv4or6 address>	<selected server>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
full of <power level>	fulls	<boolean>	<power level>	*Plain*	lin
gateway address <integer> of <selected server>	gateway addresses	<ipv4or6 address>	<selected server>	*Numbered*	lin
gateway address of <selected server>	gateway addresses	<ipv4or6 address>	<selected server>	*Plain*	lin
gather schedule authority of <site>	gather schedule authoritys	<string>	<site>	*Plain*	lin
gather schedule time interval of <site>	gather schedule time intervals	<time interval>	<site>	*Plain*	lin
gather url of <license>	gather urls	<string>	<license>	*Plain*	lin
ghz	ghzs	<hertz>	<world>	*PlainGlobal*	core
gid of <filesystem object>	gids	<integer>	<filesystem object>	*Plain*	lin
gid of <symlink>	gids	<integer>	<symlink>	*Plain*	lin
greatest hz	greatest hzs	<hertz>	<world>	*PlainGlobal*	core
greatest integer	greatest integers	<integer>	<world>	*PlainGlobal*	core
greatest time interval	greatest time intervals	<time interval>	<world>	*PlainGlobal*	core
group <integer> of <site>	groups	<site group>	<site>	*Numbered*	lin
group execute of <filesystem object>	group executes	<boolean>	<filesystem object>	*Plain*	lin
group leader of <action>	group leaders	<boolean>	<action>	*Plain*	lin
group mask of <filesystem object>	group masks	<integer>	<filesystem object>	*Plain*	lin
group mask of <mode>	group masks	<mode_mask>	<mode>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
group name of <filesystem object>	group names	<string>	<filesystem object>	*Plain*	lin
group name of <symlink>	group names	<string>	<symlink>	*Plain*	lin
group read of <filesystem object>	group reads	<boolean>	<filesystem object>	*Plain*	lin
group write of <filesystem object>	group writes	<boolean>	<filesystem object>	*Plain*	lin
h1 <string> of <html>	h1s	<html>	<html>	*Named*	core
h1 <string> of <string>	h1s	<html>	<string>	*Named*	core
h1 of <html>	h1s	<html>	<html>	*Plain*	core
h1 of <string>	h1s	<html>	<string>	*Plain*	core
h2 <string> of <html>	h2s	<html>	<html>	*Named*	core
h2 <string> of <string>	h2s	<html>	<string>	*Named*	core
h2 of <html>	h2s	<html>	<html>	*Plain*	core
h2 of <string>	h2s	<html>	<string>	*Plain*	core
h3 <string> of <html>	h3s	<html>	<html>	*Named*	core
h3 <string> of <string>	h3s	<html>	<string>	*Named*	core
h3 of <html>	h3s	<html>	<html>	*Plain*	core
h3 of <string>	h3s	<html>	<string>	*Plain*	core
h4 <string> of <html>	h4s	<html>	<html>	*Named*	core
h4 <string> of <string>	h4s	<html>	<string>	*Named*	core
h4 of <html>	h4s	<html>	<html>	*Plain*	core
h4 of <string>	h4s	<html>	<string>	*Plain*	core
h5 <string> of <html>	h5s	<html>	<html>	*Named*	core
h5 <string> of <string>	h5s	<html>	<string>	*Named*	core
h5 of <html>	h5s	<html>	<html>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
h5 of <string>	h5s	<html>	<string>	*Plain*	core
h6 <string> of <html>	h6s	<html>	<html>	*Named*	core
h6 <string> of <string>	h6s	<html>	<string>	*Named*	core
h6 of <html>	h6s	<html>	<html>	*Plain*	core
h6 of <string>	h6s	<html>	<string>	*Plain*	core
head <string> of <html>	heads	<html>	<html>	*Named*	core
head <string> of <string>	heads	<html>	<string>	*Named*	core
head of <html>	heads	<html>	<html>	*Plain*	core
head of <string>	heads	<html>	<string>	*Plain*	core
header <string> of <fixlet>	headers	<fixlet_header>	<fixlet>	*Named*	lin
header of <fixlet>	headers	<fixlet_header>	<fixlet>	*Plain*	lin
hexadecet <integer> of <ipv4or6 address>	hexadecets	<integer>	<ipv4or6 address>	*Numbered*	core
hexadecet <integer> of <ipv6 address>	hexadecets	<integer>	<ipv6 address>	*Numbered*	core
hexadecimal integer <string>	hexadecimal integers	<integer>	<world>	*NamedGlobal*	core
hexadecimal string <string>	hexadecimal strings	<string>	<world>	*NamedGlobal*	core
hlt bug of <processor>	hlt bugs	<boolean>	<processor>	*Plain*	lin
host name	host names	<string>	<world>	*PlainGlobal*	lin
host name of <root server>	host names	<string>	<root server>	*Plain*	lin
hostname	hostnames	<string>	<world>	*PlainGlobal*	lin
hour	hours	<time interval>	<world>	*PlainGlobal*	core
hour_of_day of <time of day with time zone>	hours_of_day	<integer>	<time of day with time zone>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
hour_of_day of <time of day>	hours_of_day	<integer>	<time of day>	*Plain*	core
hr	hrs	<html>	<world>	*PlainGlobal*	core
hr <string>	hrs	<html>	<world>	*NamedGlobal*	core
html <string>	htmls	<html>	<world>	*NamedGlobal*	core
html <string> of <html>	htmls	<html>	<html>	*Named*	core
html <string> of <string>	htmls	<html>	<string>	*Named*	core
html concatenation <string> of <html>	html concatenations	<html>	<html>	*Named*	core
html concatenation of <html>	html concatenations	<html>	<html>	*Plain*	core
html of <html>	htmls	<html>	<html>	*Plain*	core
html of <string>	htmls	<html>	<string>	*Plain*	core
html tag <(string, html)>	html tags	<html>	<world>	*Index<(string, html)>Global*	core
html tag <(string, html attribute list)>	html tags	<html>	<world>	*Index<(string, html attribute list)>Global*	core
html tag <(string, html attribute list, html)>	html tags	<html>	<world>	*Index<(string, html attribute list, html)>Global*	core
html tag <(string, html attribute list, string)>	html tags	<html>	<world>	*Index<(string, html attribute list, string)>Global*	core
html tag <(string, string)>	html tags	<html>	<world>	*Index<(string, string)>Global*	core
html tag <string> of <html>	html tags	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
html tag <string> of <string>	html tags	<html>	<string>	*Named*	core
hz	hzs	<hertz>	<world>	*PlainGlobal*	core
id of <action>	ids	<integer>	<action>	*Plain*	lin
id of <fixlet>	ids	<integer>	<fixlet>	*Plain*	lin
id of <process>	ids	<integer>	<process>	*Plain*	lin
id of <root server>	ids	<integer>	<root server>	*Plain*	lin
id of <site group>	ids	<integer>	<site group>	*Plain*	lin
index of <processor>	indexes	<integer>	<processor>	*Plain*	lin
index type of <property>	index types	<type>	<property>	*Plain*	core
inexact of <floating point>	inexacts	<boolean>	<floating point>	*Plain*	core
infinite of <floating point>	infinites	<boolean>	<floating point>	*Plain*	core
initial part <time interval> of <time range>	initial parts	<time range>	<time range>	*Index<time interval>*	core
ins <string> of <html>	inss	<html>	<html>	*Named*	core
ins <string> of <string>	inss	<html>	<string>	*Named*	core
ins of <html>	inss	<html>	<html>	*Plain*	core
ins of <string>	inss	<html>	<string>	*Plain*	core
installed <string> of <rpmdatabase>	installeds	<boolean>	<rpmdatabase>	*Named*	lin
installed file of <package>	installed files	<capability>	<package>	*Plain*	lin
instance of <application usage summary>	instances	<application usage summary instance>	<application usage summary>	*Plain*	lin
integer <integer>	integers	<integer>	<world>	*NumberedGlobal*	core
integer <string>	integers	<integer>	<world>	*NamedGlobal*	core

Tivoli Endpoint Manager

Key Phrase	Plural	Creates a	From a	Form	Ref
integer <string> of <smbios structure>	integers	<integer>	<smbios structure>	*Named*	lin
integer ceiling of <floating point>	integer ceilings	<integer>	<floating point>	*Plain*	core
integer floor of <floating point>	integer floors	<integer>	<floating point>	*Plain*	core
integer in <(integer, integer)>	integers in	<integer>	<world>	*Index<(integer, integer)>Global*	core
integer in <(integer, integer, integer)>	integers in	<integer>	<world>	*Index<(integer, integer, integer)>Global*	core
integer to <integer>	integers to	<integer>	<world>	*NumberedGlobal*	core
integer value <string> of <smbios structure>	integer values	<smbios value>	<smbios structure>	*Named*	lin
interface <integer> of <network>	interfaces	<network interface>	<network>	*Numbered*	lin
interface of <network>	interfaces	<network interface>	<network>	*Plain*	lin
intersection of <integer set>	intersections	<integer set>	<integer set>	*Plain*	core
intersection of <string set>	intersections	<string set>	<string set>	*Plain*	core
invalid before of <x509 certificate>	invalid befores	<time>	<x509 certificate>	*Plain*	core
invalid of <floating point>	invalids	<boolean>	<floating point>	*Plain*	core
ip address of <selected server>	ip addresses	<ipv4or6 address>	<selected server>	*Plain*	lin
ip interface <integer> of <network>	ip interfaces	<network ip interface>	<network>	*Numbered*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
ip interface of <network>	ip interfaces	<network ip interface>	<network>	*Plain*	<u>lin</u>
ip version <integer>	ip versions	<ip version>	<world>	*NumberedGlobal*	core
ip version of <ipv4or6 address>	ip versions	<ip version>	<ipv4or6 address>	*Plain*	core
ipv4	ipv4s	<ip version>	<world>	*PlainGlobal*	core
ipv4 address <string>	ipv4 addresses	<ipv4 address>	<world>	*NamedGlobal*	core
ipv4 interface of <network adapter>	ipv4 interfaces	<network adapter interface>	<network adapter>	*Plain*	<u>lin</u>
ipv4 interface of <network>	ipv4 interfaces	<network adapter interface>	<network>	*Plain*	<u>lin</u>
ipv4 part of <ipv4or6 address>	ipv4 parts	<ipv4 address>	<ipv4or6 address>	*Plain*	core
ipv4 part of <ipv6 address>	ipv4 parts	<ipv4 address>	<ipv6 address>	*Plain*	core
ipv4or6 address <string>	ipv4or6 addresses	<ipv4or6 address>	<world>	*NamedGlobal*	core
ipv4or6 interface of <network adapter>	ipv4or6 interfaces	<network adapter interface>	<network adapter>	*Plain*	<u>lin</u>
ipv4or6 interface of <network>	ipv4or6 interfaces	<network adapter interface>	<network>	*Plain*	<u>lin</u>
ipv6	ipv6s	<ip version>	<world>	*PlainGlobal*	core
ipv6 address <string>	ipv6 addresses	<ipv6 address>	<world>	*NamedGlobal*	core
ipv6 interface of <network adapter>	ipv6 interfaces	<network adapter interface>	<network adapter>	*Plain*	<u>lin</u>
ipv6 interface of <network>	ipv6 interfaces	<network adapter interface>	<network>	*Plain*	<u>lin</u>
italic <string> of <html>	italics	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
italic <string> of <string>	italics	<html>	<string>	*Named*	core
italic of <html>	italics	<html>	<html>	*Plain*	core
italic of <string>	italics	<html>	<string>	*Plain*	core
january	januarys	<month>	<world>	*PlainGlobal*	core
january <integer>	januarys	<day of year>	<world>	*NumberedGlobal*	core
january <integer> of <integer>	januarys	<date>	<integer>	*Numbered*	core
january of <integer>	januarys	<month and year>	<integer>	*Plain*	core
july	julys	<month>	<world>	*PlainGlobal*	core
july <integer>	julys	<day of year>	<world>	*NumberedGlobal*	core
july <integer> of <integer>	julys	<date>	<integer>	*Numbered*	core
july of <integer>	julys	<month and year>	<integer>	*Plain*	core
june	junes	<month>	<world>	*PlainGlobal*	core
june <integer>	junes	<day of year>	<world>	*NumberedGlobal*	core
june <integer> of <integer>	junes	<date>	<integer>	*Numbered*	core
june of <integer>	junes	<month and year>	<integer>	*Plain*	core
kbd <string> of <html>	kbds	<html>	<html>	*Named*	core
kbd <string> of <string>	kbds	<html>	<string>	*Named*	core
kbd of <html>	kbds	<html>	<html>	*Plain*	core
kbd of <string>	kbds	<html>	<string>	*Plain*	core
key <string> of <file section>	keys	<string>	<file section>	*Named*	lin
key <string> of <file>	keys	<string>	<file>	*Named*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
khz	khzs	<hertz>	<world>	*PlainGlobal*	core
last <integer> of <string>	lasts	<substring>	<string>	*Numbered*	core
last <string> of <string>	lasts	<substring>	<string>	*Named*	core
last change time of <action>	last change times	<time>	<action>	*Plain*	lin
last gather time of <site>	last gather times	<time>	<site>	*Plain*	lin
last relay select time	last relay select times	<time>	<world>	*PlainGlobal*	lin
last start time of <application usage summary instance>	last start times	<time>	<application usage summary instance>	*Plain*	lin
last start time of <application usage summary>	last start times	<time>	<application usage summary>	*Plain*	lin
last time seen of <application usage summary instance>	last times seen	<time>	<application usage summary instance>	*Plain*	lin
last time seen of <application usage summary>	last times seen	<time>	<application usage summary>	*Plain*	lin
leap of <year>	leaps	<boolean>	<year>	*Plain*	core
least hz	least hzs	<hertz>	<world>	*PlainGlobal*	core
least integer	least integers	<integer>	<world>	*PlainGlobal*	core
least significant one bit of <bit set>	least significant one bits	<integer>	<bit set>	*Plain*	core
least time interval	least time intervals	<time interval>	<world>	*PlainGlobal*	core
left operand type of <binary operator>	left operand types	<type>	<binary operator>	*Plain*	core
left shift <integer> of <bit set>	left shifts	<bit set>	<bit set>	*Numbered*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
length of <month and year>	lengths	<time interval>	<month and year>	*Plain*	core
length of <rope>	lengths	<integer>	<rope>	*Plain*	core
length of <smbios structure>	lengths	<integer>	<smbios structure>	*Plain*	lin
length of <string>	lengths	<integer>	<string>	*Plain*	core
length of <time range>	lengths	<time interval>	<time range>	*Plain*	core
length of <year>	lengths	<time interval>	<year>	*Plain*	core
less significance <integer> of <floating point>	less significances	<floating point>	<floating point>	*Numbered*	core
li <string> of <html>	lis	<html>	<html>	*Named*	core
li <string> of <string>	lis	<html>	<string>	*Named*	core
li of <html>	lis	<html>	<html>	*Plain*	core
li of <string>	lis	<html>	<string>	*Plain*	core
line <integer> of <file>	lines	<file line>	<file>	*Numbered*	lin
line containing <string> of <file>	lines containing	<file line>	<file>	*Named*	lin
line number of <file line>	line numbers	<integer>	<file line>	*Plain*	lin
line of <file>	lines	<file line>	<file>	*Plain*	lin
line starting with <string> of <file>	lines starting with	<file line>	<file>	*Named*	lin
link <string> of <html>	links	<html>	<html>	*Named*	core
link <string> of <string>	links	<html>	<string>	*Named*	core
link count of <filesystem object>	link counts	<integer>	<filesystem object>	*Plain*	lin
link count of <symlink>	link counts	<integer>	<symlink>	*Plain*	lin
link of <html>	links	<html>	<html>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
link of <string>	links	<html>	<string>	*Plain*	core
local time <string>	local times	<time>	<world>	*NamedGlobal*	core
local time zone	local time zones	<time zone>	<world>	*PlainGlobal*	core
location of <filesystem object>	locations	<string>	<filesystem object>	*Plain*	lin
location of <symlink>	locations	<string>	<symlink>	*Plain*	lin
lock string of <action lock state>	lock strings	<string>	<action lock state>	*Plain*	lin
locked of <action lock state>	lockeds	<boolean>	<action lock state>	*Plain*	lin
long form of <short rpm package version record>	long forms	<rpm package version record>	<short rpm package version record>	*Plain*	lin
loopback of <network adapter interface>	loopbacks	<boolean>	<network adapter interface>	*Plain*	lin
loopback of <network adapter>	loopbacks	<boolean>	<network adapter>	*Plain*	lin
loopback of <network ip interface>	loopbacks	<boolean>	<network ip interface>	*Plain*	lin
low of <power level>	lows	<boolean>	<power level>	*Plain*	lin
mac address of <network adapter interface>	mac addresses	<string>	<network adapter interface>	*Plain*	lin
mac address of <network adapter>	mac addresses	<string>	<network adapter>	*Plain*	lin
mac address of <network ip interface>	mac addresses	<string>	<network ip interface>	*Plain*	lin
mac of <operating system>	macs	<boolean>	<operating system>	*Plain*	lin
main gather service	main gather services	<service>	<world>	*PlainGlobal*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
main processor	main processors	<processor>	<world>	*PlainGlobal*	lin
major of <device file>	majors	<integer>	<device file>	*Plain*	lin
march	marchs	<month>	<world>	*PlainGlobal*	core
march <integer>	marchs	<day of year>	<world>	*NumberedGlobal*	core
march <integer> of <integer>	marchs	<date>	<integer>	*Numbered*	core
march of <integer>	marchs	<month and year>	<integer>	*Plain*	core
masthead of <site>	mastheads	<file>	<site>	*Plain*	lin
match <regular expression> of <string>	matches	<regular expression match>	<string>	*Index<regular expression>*	regx
maximum of <date>	maxima	<date>	<date>	*Plain*	core
maximum of <day of month>	maxima	<day of month>	<day of month>	*Plain*	core
maximum of <day of year>	maxima	<day of year>	<day of year>	*Plain*	core
maximum of <evaluation cycle>	maximums	<integer>	<evaluation cycle>	*Plain*	lin
maximum of <floating point>	maxima	<floating point>	<floating point>	*Plain*	core
maximum of <hertz>	maxima	<hertz>	<hertz>	*Plain*	core
maximum of <integer>	maxima	<integer>	<integer>	*Plain*	core
maximum of <ipv4 address>	maxima	<ipv4 address>	<ipv4 address>	*Plain*	core
maximum of <ipv4or6 address>	maxima	<ipv4or6 address>	<ipv4or6 address>	*Plain*	core
maximum of <ipv6 address>	maxima	<ipv6 address>	<ipv6 address>	*Plain*	core
maximum of <month and year>	maxima	<month and year>	<month and year>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
maximum of <month>	maxima	<month>	<month>	*Plain*	core
maximum of <number of months>	maxima	<number of months>	<number of months>	*Plain*	core
maximum of <rpm package release>	maxima	<rpm package release>	<rpm package release>	*Plain*	lin
maximum of <rpm package version record>	maxima	<rpm package version record>	<rpm package version record>	*Plain*	lin
maximum of <rpm package version>	maxima	<rpm package version>	<rpm package version>	*Plain*	lin
maximum of <short rpm package version record>	maxima	<short rpm package version record>	<short rpm package version record>	*Plain*	lin
maximum of <site version list>	maxima	<site version list>	<site version list>	*Plain*	core
maximum of <time interval>	maxima	<time interval>	<time interval>	*Plain*	core
maximum of <time of day>	maxima	<time of day>	<time of day>	*Plain*	core
maximum of <time>	maxima	<time>	<time>	*Plain*	core
maximum of <version>	maxima	<version>	<version>	*Plain*	core
maximum of <year>	maxima	<year>	<year>	*Plain*	core
maximum seat count of <license>	maximum seat counts	<integer>	<license>	*Plain*	lin
may	mays	<month>	<world>	*PlainGlobal*	core
may <integer>	mays	<day of year>	<world>	*NumberedGlobal*	core
may <integer> of <integer>	mays	<date>	<integer>	*Numbered*	core
may of <integer>	mays	<month and year>	<integer>	*Plain*	core
member of <site group>	members	<boolean>	<site group>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
meta <string> of <html>	metas	<html>	<html>	Named	core
meta <string> of <string>	metas	<html>	<string>	Named	core
meta of <html>	metas	<html>	<html>	Plain	core
meta of <string>	metas	<html>	<string>	Plain	core
mhz	mhzs	<hertz>	<world>	PlainGlobal	core
microsecond	microseconds	<time interval>	<world>	PlainGlobal	core
midnight	midnights	<time of day>	<world>	PlainGlobal	core
millisecond	milliseconds	<time interval>	<world>	PlainGlobal	core
minimum of <date>	minima	<date>	<date>	Plain	core
minimum of <day of month>	minima	<day of month>	<day of month>	Plain	core
minimum of <day of year>	minima	<day of year>	<day of year>	Plain	core
minimum of <floating point>	minima	<floating point>	<floating point>	Plain	core
minimum of <hertz>	minima	<hertz>	<hertz>	Plain	core
minimum of <integer>	minima	<integer>	<integer>	Plain	core
minimum of <ipv4 address>	minima	<ipv4 address>	<ipv4 address>	Plain	core
minimum of <ipv4or6 address>	minima	<ipv4or6 address>	<ipv4or6 address>	Plain	core
minimum of <ipv6 address>	minima	<ipv6 address>	<ipv6 address>	Plain	core
minimum of <month and year>	minima	<month and year>	<month and year>	Plain	core
minimum of <month>	minima	<month>	<month>	Plain	core
minimum of <number of months>	minima	<number of months>	<number of months>	Plain	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
minimum of <rpm package release>	minima	<rpm package release>	<rpm package release>	*Plain*	lin
minimum of <rpm package version record>	minima	<rpm package version record>	<rpm package version record>	*Plain*	lin
minimum of <rpm package version>	minima	<rpm package version>	<rpm package version>	*Plain*	lin
minimum of <short rpm package version record>	minima	<short rpm package version record>	<short rpm package version record>	*Plain*	lin
minimum of <site version list>	minima	<site version list>	<site version list>	*Plain*	core
minimum of <time interval>	minima	<time interval>	<time interval>	*Plain*	core
minimum of <time of day>	minima	<time of day>	<time of day>	*Plain*	core
minimum of <time>	minima	<time>	<time>	*Plain*	core
minimum of <version>	minima	<version>	<version>	*Plain*	core
minimum of <year>	minima	<year>	<year>	*Plain*	core
minor of <device file>	minors	<integer>	<device file>	*Plain*	lin
minute	minutes	<time interval>	<world>	*PlainGlobal*	core
minute_of_hour of <time of day with time zone>	minutes_of_hour	<integer>	<time of day with time zone>	*Plain*	core
minute_of_hour of <time of day>	minutes_of_hour	<integer>	<time of day>	*Plain*	core
mode of <filesystem object>	modes	<mode>	<filesystem object>	*Plain*	lin
model name of <processor>	model names	<string>	<processor>	*Plain*	lin
model of <processor>	models	<integer>	<processor>	*Plain*	lin
modification time of <filesystem object>	modification times	<time>	<filesystem object>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
modification time of <symlink>	modification times	<time>	<symlink>	*Plain*	lin
module <string>	modules	<module>	<world>	*NamedGlobal*	core
monday	mondays	<day of week>	<world>	*PlainGlobal*	core
month	months	<number of months>	<world>	*PlainGlobal*	core
month <integer>	months	<month>	<world>	*NumberedGlobal*	core
month <string>	months	<month>	<world>	*NamedGlobal*	core
month of <date>	months	<month>	<date>	*Plain*	core
month of <day of year>	months	<month>	<day of year>	*Plain*	core
month of <month and year>	months	<month>	<month and year>	*Plain*	core
month_and_year of <date>	months_and _years	<month and year>	<date>	*Plain*	core
more significance <integer> of <floating point>	more significances	<floating point>	<floating point>	*Numbered*	core
most significant one bit of <bit set>	most significant one bits	<integer>	<bit set>	*Plain*	core
multicast support of <network adapter interface>	multicast supports	<boolean>	<network adapter interface>	*Plain*	lin
multicast support of <network adapter>	multicast supports	<boolean>	<network adapter>	*Plain*	lin
multicast support of <network ip interface>	multicast supports	<boolean>	<network ip interface>	*Plain*	lin
multiplicity of <date with multiplicity>	multiplicities	<integer>	<date with multiplicity>	*Plain*	core
multiplicity of <day of month with multiplicity>	multiplicities	<integer>	<day of month with multiplicity>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
multiplicity of <day of week with multiplicity>	multiplicities	<integer>	<day of week with multiplicity>	*Plain*	core
multiplicity of <day of year with multiplicity>	multiplicities	<integer>	<day of year with multiplicity>	*Plain*	core
multiplicity of <floating point with multiplicity>	multiplicities	<integer>	<floating point with multiplicity>	*Plain*	core
multiplicity of <hertz with multiplicity>	multiplicities	<integer>	<hertz with multiplicity>	*Plain*	core
multiplicity of <integer with multiplicity>	multiplicities	<integer>	<integer with multiplicity>	*Plain*	core
multiplicity of <ipv4 address with multiplicity>	multiplicities	<integer>	<ipv4 address with multiplicity>	*Plain*	core
multiplicity of <ipv4or6 address with multiplicity>	multiplicities	<integer>	<ipv4or6 address with multiplicity>	*Plain*	core
multiplicity of <ipv6 address with multiplicity>	multiplicities	<integer>	<ipv6 address with multiplicity>	*Plain*	core
multiplicity of <month and year with multiplicity>	multiplicities	<integer>	<month and year with multiplicity>	*Plain*	core
multiplicity of <month with multiplicity>	multiplicities	<integer>	<month with multiplicity>	*Plain*	core
multiplicity of <number of months with multiplicity>	multiplicities	<integer>	<number of months with multiplicity>	*Plain*	core
multiplicity of <rpm package release with multiplicity>	multiplicities	<integer>	<rpm package release with multiplicity>	*Plain*	lin
multiplicity of <rpm package version record with multiplicity>	multiplicities	<integer>	<rpm package version record with multiplicity>	*Plain*	lin
multiplicity of <rpm package version with multiplicity>	multiplicities	<integer>	<rpm package version with multiplicity>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
multiplicity of <short rpm package version record with multiplicity>	multiplicities	<integer>	<short rpm package version record with multiplicity>	*Plain*	lin
multiplicity of <site version list with multiplicity>	multiplicities	<integer>	<site version list with multiplicity>	*Plain*	core
multiplicity of <string with multiplicity>	multiplicities	<integer>	<string with multiplicity>	*Plain*	core
multiplicity of <time interval with multiplicity>	multiplicities	<integer>	<time interval with multiplicity>	*Plain*	core
multiplicity of <time of day with multiplicity>	multiplicities	<integer>	<time of day with multiplicity>	*Plain*	core
multiplicity of <time of day with time zone with multiplicity>	multiplicities	<integer>	<time of day with time zone with multiplicity>	*Plain*	core
multiplicity of <time range with multiplicity>	multiplicities	<integer>	<time range with multiplicity>	*Plain*	core
multiplicity of <time with multiplicity>	multiplicities	<integer>	<time with multiplicity>	*Plain*	core
multiplicity of <time zone with multiplicity>	multiplicities	<integer>	<time zone with multiplicity>	*Plain*	core
multiplicity of <version with multiplicity>	multiplicities	<integer>	<version with multiplicity>	*Plain*	core
multiplicity of <year with multiplicity>	multiplicities	<integer>	<year with multiplicity>	*Plain*	core
multivalued of <property>	multivalueds	<boolean>	<property>	*Plain*	core
name of <application usage summary instance>	names	<string>	<application usage summary instance>	*Plain*	lin
name of <application usage summary>	names	<string>	<application usage summary>	*Plain*	lin
name of <bes product>	names	<string>	<bes product>	*Plain*	lin
name of <binary operator>	names	<string>	<binary operator>	*Plain*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
name of <capability>	names	<string>	<capability>	*Plain*	lin
name of <cast>	names	<string>	<cast>	*Plain*	core
name of <environment variable>	names	<string>	<environment variable>	*Plain*	lin
name of <filesystem object>	names	<string>	<filesystem object>	*Plain*	lin
name of <filesystem>	names	<string>	<filesystem>	*Plain*	lin
name of <fixlet_header>	names	<string>	<fixlet_header>	*Plain*	lin
name of <network adapter>	names	<string>	<network adapter>	*Plain*	lin
name of <network ip interface>	names	<string>	<network ip interface>	*Plain*	lin
name of <operating system>	names	<string>	<operating system>	*Plain*	lin
name of <package>	names	<string>	<package>	*Plain*	lin
name of <process>	names	<string>	<process>	*Plain*	lin
name of <selected server>	names	<string>	<selected server>	*Plain*	lin
name of <setting>	names	<string>	<setting>	*Plain*	lin
name of <site>	names	<string>	<site>	*Plain*	lin
name of <smbios structure>	names	<string>	<smbios structure>	*Plain*	lin
name of <smbios value>	names	<string>	<smbios value>	*Plain*	lin
name of <symlink>	names	<string>	<symlink>	*Plain*	lin
name of <type>	names	<string>	<type>	*Plain*	core
name of <unary operator>	names	<string>	<unary operator>	*Plain*	core
name of <user>	names	<string>	<user>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
nan of <floating point>	nans	<boolean>	<floating point>	*Plain*	core
network	networks	<network>	<world>	*PlainGlobal*	lin
next line of <file line>	next lines	<file line>	<file line>	*Plain*	lin
no epoch of <rpm package version record>	no epochs	<rpm package version record>	<rpm package version record>	*Plain*	lin
no epoch of <short rpm package version record>	no epochs	<short rpm package version record>	<short rpm package version record>	*Plain*	lin
non windows server count of <bes product>	non windows server counts	<integer>	<bes product>	*Plain*	lin
noon	noons	<time of day>	<world>	*PlainGlobal*	core
normal of <floating point>	normals	<boolean>	<floating point>	*Plain*	core
normal of <power level>	normals	<boolean>	<power level>	*Plain*	lin
november	novembers	<month>	<world>	*PlainGlobal*	core
november <integer>	novembers	<day of year>	<world>	*NumberedGlobal*	core
november <integer> of <integer>	novembers	<date>	<integer>	*Numbered*	core
november of <integer>	novembers	<month and year>	<integer>	*Plain*	core
now	nows	<time>	<world>	*PlainGlobal*	core
numeric value of <string>	numeric values	<integer>	<string>	*Plain*	core
obsolete of <package>	obsoletes	<capability>	<package>	*Plain*	lin
october	octobers	<month>	<world>	*PlainGlobal*	core
october <integer>	octobers	<day of year>	<world>	*NumberedGlobal*	core
october <integer> of <integer>	octobers	<date>	<integer>	*Numbered*	core
october of <integer>	octobers	<month and year>	<integer>	*Plain*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
offer accepted of <action>	offer accepteds	<boolean>	<action>	*Plain*	lin
offer of <action>	offers	<boolean>	<action>	*Plain*	lin
offset of <smbios value>	offsets	<integer>	<smbios value>	*Plain*	lin
ol <string> of <html>	ols	<html>	<html>	*Named*	core
ol <string> of <string>	ols	<html>	<string>	*Named*	core
ol of <html>	ols	<html>	<html>	*Plain*	core
ol of <string>	ols	<html>	<string>	*Plain*	core
one bit of <bit set>	one bits	<integer>	<bit set>	*Plain*	core
operand type of <cast>	operand types	<type>	<cast>	*Plain*	core
operand type of <unary operator>	operand types	<type>	<unary operator>	*Plain*	core
operating system	operating systems	<operating system>	<world>	*PlainGlobal*	lin
ordered list <string> of <html>	ordered lists	<html>	<html>	*Named*	core
ordered list <string> of <string>	ordered lists	<html>	<string>	*Named*	core
ordered list of <html>	ordered lists	<html>	<html>	*Plain*	core
ordered list of <string>	ordered lists	<html>	<string>	*Plain*	core
organization of <license>	organizations	<string>	<license>	*Plain*	lin
origin fixlet id of <action>	origin fixlet ids	<integer>	<action>	*Plain*	lin
other execute of <filesystem object>	other executes	<boolean>	<filesystem object>	*Plain*	lin
other mask of <filesystem object>	other masks	<integer>	<filesystem object>	*Plain*	lin
other mask of <mode>	other masks	<mode_mask>	<mode>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
other read of <filesystem object>	other reads	<boolean>	<filesystem object>	*Plain*	[lin](#)
other write of <filesystem object>	other writes	<boolean>	<filesystem object>	*Plain*	[lin](#)
overflow of <floating point>	overflows	<boolean>	<floating point>	*Plain*	core
p <string> of <html>	ps	<html>	<html>	*Named*	core
p <string> of <string>	ps	<html>	<string>	*Named*	core
p of <html>	ps	<html>	<html>	*Plain*	core
p of <string>	ps	<html>	<string>	*Plain*	core
package <string> of <rpmdatabase>	packages	<package>	<rpmdatabase>	*Named*	[lin](#)
package conflicting with <capability> of <rpmdatabase>	packages conflicting with	<package>	<rpmdatabase>	*Index<capability>*	[lin](#)
package installing <capability> of <rpmdatabase>	packages installing	<package>	<rpmdatabase>	*Index<capability>*	[lin](#)
package of <rpmdatabase>	packages	<package>	<rpmdatabase>	*Plain*	[lin](#)
package providing <capability> of <rpmdatabase>	packages providing	<package>	<rpmdatabase>	*Index<capability>*	[lin](#)
package requiring <capability> of <rpmdatabase>	packages requiring	<package>	<rpmdatabase>	*Index<capability>*	[lin](#)
parameter <string>	parameters	<string>	<world>	*NamedGlobal*	[lin](#)
parameter <string> of <action>	parameters	<string>	<action>	*Named*	[lin](#)
parent folder of <filesystem object>	parent folders	<folder>	<filesystem object>	*Plain*	[lin](#)
parent folder of <symlink>	parent folders	<folder>	<symlink>	*Plain*	[lin](#)

Key Phrase	Plural	Creates a	From a	*Form*	Ref
parent of <type>	parents	<type>	<type>	*Plain*	core
parenthesized part <integer> of <regular expression match>	parenthesized parts	<substring>	<regular expression match>	*Numbered*	regx
parenthesized part of <regular expression match>	parenthesized parts	<substring>	<regular expression match>	*Plain*	regx
pathname of <filesystem object>	pathnames	<string>	<filesystem object>	*Plain*	lin
pathname of <symlink>	pathnames	<string>	<symlink>	*Plain*	lin
pending login	pending logins	<boolean>	<world>	*PlainGlobal*	lin
pending login of <action>	pending logins	<boolean>	<action>	*Plain*	lin
pending of <action>	pendings	<boolean>	<action>	*Plain*	lin
pending restart	pending restarts	<boolean>	<world>	*PlainGlobal*	lin
pending restart <string>	pending restarts	<boolean>	<world>	*NamedGlobal*	lin
pending restart name	pending restart names	<string>	<world>	*PlainGlobal*	lin
pending restart of <action>	pending restarts	<boolean>	<action>	*Plain*	lin
pending time of <action>	pending times	<time>	<action>	*Plain*	lin
pid of <process>	pids	<integer>	<process>	*Plain*	lin
platform id of <language>	platform ids	<string>	<language>	*Plain*	lin
plugged of <power level>	pluggeds	<boolean>	<power level>	*Plain*	lin
plural name of <property>	plural names	<string>	<property>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
point to point of <network adapter interface>	point to points	<boolean>	<network adapter interface>	*Plain*	lin
point to point of <network ip interface>	point to points	<boolean>	<network ip interface>	*Plain*	lin
port number of <selected server>	port numbers	<integer>	<selected server>	*Plain*	lin
position <integer> of <string>	positions	<string position>	<string>	*Numbered*	core
position of <string>	positions	<string position>	<string>	*Plain*	core
power level	power levels	<power level>	<world>	*PlainGlobal*	lin
pre <string> of <html>	pres	<html>	<html>	*Named*	core
pre <string> of <string>	pres	<html>	<string>	*Named*	core
pre of <html>	pres	<html>	<html>	*Plain*	core
pre of <string>	pres	<html>	<string>	*Plain*	core
preceding text of <string position>	preceding texts	<substring>	<string position>	*Plain*	core
preceding text of <substring>	preceding texts	<substring>	<substring>	*Plain*	core
previous line of <file line>	previous lines	<file line>	<file line>	*Plain*	lin
previous value of <runlevel>	previous values	<string>	<runlevel>	*Plain*	lin
primary language of <language>	primary languages	<primary language>	<language>	*Plain*	lin
priority of <selected server>	priorities	<integer>	<selected server>	*Plain*	lin
process	processes	<process>	<world>	*PlainGlobal*	lin
process <integer>	processes	<process>	<world>	*NumberedGlobal*	lin
process <string>	processes	<process>	<world>	*NamedGlobal*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
process id of <process>	process ids	<integer>	<process>	*Plain*	lin
processor	processors	<processor>	<world>	*PlainGlobal*	lin
processor <integer>	processors	<processor>	<world>	*NumberedGlobal*	lin
product of <floating point>	products	<floating point>	<floating point>	*Plain*	core
product of <integer>	products	<integer>	<integer>	*Plain*	core
product of <license>	products	<bes product>	<license>	*Plain*	lin
property <string>	properties	<property>	<world>	*NamedGlobal*	core
property <string> of <type>	properties	<property>	<type>	*Named*	core
property of <type>	properties	<property>	<type>	*Plain*	core
property returning <type>	properties returning	<property>	<world>	*Index<type>Global*	core
property returning <type> of <type>	properties returning	<property>	<type>	*Index<type>*	core
provide of <package>	provides	<capability>	<package>	*Plain*	lin
q <string> of <html>	qs	<html>	<html>	*Named*	core
q <string> of <string>	qs	<html>	<string>	*Named*	core
q of <html>	qs	<html>	<html>	*Plain*	core
q of <string>	qs	<html>	<string>	*Plain*	core
ram	rams	<ram>	<world>	*PlainGlobal*	lin
random access memory	random access memories	<ram>	<world>	*PlainGlobal*	lin
range after <time> of <time range>	ranges after	<time range>	<time range>	*Index<time>*	core
range before <time> of <time range>	ranges before	<time range>	<time range>	*Index<time>*	core

Tivoli Endpoint Manager

Key Phrase	Plural	Creates a	From a	*Form*	Ref
read of <mode_mask>	reads	<boolean>	<mode_mask>	*Plain*	lin
recent application	recent applications	<application>	<world>	*PlainGlobal*	lin
recent application <string>	recent applications	<application>	<world>	*NamedGlobal*	lin
regex <string>	regexes	<regular expression>	<world>	*NamedGlobal*	regx
regex escape of <string>	regex escapes	<string>	<string>	*Plain*	regx
registrar number of <license>	registrar numbers	<integer>	<license>	*Plain*	lin
registration address of <client>	registration addresses	<ipv4or6 address>	<client>	*Plain*	lin
registration cidr address of <client>	registration cidr addresses	<string>	<client>	*Plain*	lin
registration mac address of <client>	registration mac addresses	<string>	<client>	*Plain*	lin
registration subnet address of <client>	registration subnet addresses	<ipv4or6 address>	<client>	*Plain*	lin
regular expression <string>	regular expressions	<regular expression>	<world>	*NamedGlobal*	regx
relation of <capability>	relations	<string>	<capability>	*Plain*	lin
relative significance place <integer> of <floating point>	relative significance places	<floating point>	<floating point>	*Numbered*	core
relative significance place of <floating point>	relative significance places	<floating point>	<floating point>	*Plain*	core
relay service	relay services	<service>	<world>	*PlainGlobal*	lin
release of <operating system>	releases	<string>	<operating system>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
release of <rpm package version record>	releases	<rpm package release>	<rpm package version record>	*Plain*	lin
release of <short rpm package version record>	releases	<rpm package release>	<short rpm package version record>	*Plain*	lin
relevance of <fixlet>	relevances	<boolean>	<fixlet>	*Plain*	lin
relevant fixlet of <site>	relevant fixlets	<fixlet>	<site>	*Plain*	lin
relevant offer action of <site>	relevant offer actions	<action>	<site>	*Plain*	lin
require of <package>	requires	<capability>	<package>	*Plain*	lin
result type of <binary operator>	result types	<type>	<binary operator>	*Plain*	core
result type of <cast>	result types	<type>	<cast>	*Plain*	core
result type of <property>	result types	<type>	<property>	*Plain*	core
result type of <unary operator>	result types	<type>	<unary operator>	*Plain*	core
right operand type of <binary operator>	right operand types	<type>	<binary operator>	*Plain*	core
right shift <integer> of <bit set>	right shifts	<bit set>	<bit set>	*Numbered*	core
root folder	root folders	<folder>	<world>	*PlainGlobal*	lin
root server	root servers	<root server>	<world>	*PlainGlobal*	lin
rope <string>	ropes	<rope>	<world>	*NamedGlobal*	core
rpm	rpms	<rpmdatabase>	<world>	*PlainGlobal*	lin
rpm <string>	rpms	<rpmdatabase>	<world>	*NamedGlobal*	lin
rpm package release <rpm package release>	rpm package releases	<rpm package release>	<world>	*Index<rpm package release>Global*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
rpm package release <string>	rpm package releases	<rpm package release>	<world>	*NamedGlobal*	lin
rpm package version <rpm package version>	rpm package versions	<rpm package version>	<world>	*Index<rpm package version>Global*	lin
rpm package version <string>	rpm package versions	<rpm package version>	<world>	*NamedGlobal*	lin
rpm package version record <rpm package version record>	rpm package version records	<rpm package version record>	<world>	*Index<rpm package version record>Global*	lin
rpm package version record <short rpm package version record>	rpm package version records	<rpm package version record>	<world>	*Index<short rpm package version record>Global*	lin
rpm package version record <string>	rpm package version records	<rpm package version record>	<world>	*NamedGlobal*	lin
rpm version record of <package>	rpm version records	<rpm package version record>	<package>	*Plain*	lin
runlevel	runlevels	<runlevel>	<world>	*PlainGlobal*	lin
running application	running applications	<application>	<world>	*PlainGlobal*	lin
running application <string>	running applications	<application>	<world>	*NamedGlobal*	lin
running of <application usage summary>	runnings	<boolean>	<application usage summary>	*Plain*	lin
running service <string>	running services	<service>	<world>	*NamedGlobal*	lin
samp <string> of <html>	samps	<html>	<html>	*Named*	core
samp <string> of <string>	samps	<html>	<string>	*Named*	core
samp of <html>	samps	<html>	<html>	*Plain*	core
samp of <string>	samps	<html>	<string>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
saturday	saturdays	<day of week>	<world>	*PlainGlobal*	core
seat count state of <license>	seat count states	<string>	<license>	*Plain*	lin
seat of <license>	seats	<integer>	<license>	*Plain*	lin
second	seconds	<time interval>	<world>	*PlainGlobal*	core
second_of_minute of <time of day with time zone>	seconds_of_minute	<integer>	<time of day with time zone>	*Plain*	core
second_of_minute of <time of day>	seconds_of_minute	<integer>	<time of day>	*Plain*	core
section <string> of <file>	sections	<file section>	<file>	*Named*	lin
selected server	selected servers	<selected server>	<world>	*PlainGlobal*	lin
sep bug of <processor>	sep bugs	<boolean>	<processor>	*Plain*	lin
september	septembers	<month>	<world>	*PlainGlobal*	core
september <integer>	septembers	<day of year>	<world>	*NumberedGlobal*	core
september <integer> of <integer>	septembers	<date>	<integer>	*Numbered*	core
september of <integer>	septembers	<month and year>	<integer>	*Plain*	core
service <string>	services	<service>	<world>	*NamedGlobal*	lin
set of <integer>	sets	<integer set>	<integer>	*Plain*	core
set of <string>	sets	<string set>	<string>	*Plain*	core
setgid of <filesystem object>	setgids	<boolean>	<filesystem object>	*Plain*	lin
setgid of <mode>	setgids	<boolean>	<mode>	*Plain*	lin
setting <string> of <client>	settings	<setting>	<client>	*Named*	lin
setting <string> of <site>	settings	<setting>	<site>	*Named*	lin

Tivoli Endpoint Manager

Key Phrase	Plural	Creates a	From a	*Form*	Ref
setting of <client>	settings	<setting>	<client>	*Plain*	lin
setting of <site>	settings	<setting>	<site>	*Plain*	lin
setuid of <filesystem object>	setuids	<boolean>	<filesystem object>	*Plain*	lin
setuid of <mode>	setuids	<boolean>	<mode>	*Plain*	lin
sha1 of <file>	sha1s	<string>	<file>	*Plain*	lin
sha1 of <x509 certificate>	sha1s	<string>	<x509 certificate>	*Plain*	core
shared amount of <ram>	shared amounts	<integer>	<ram>	*Plain*	lin
short form of <rpm package version record>	short forms	<short rpm package version record>	<rpm package version record>	*Plain*	lin
short rpm package version record <rpm package version record>	short rpm package version records	<short rpm package version record>	<world>	*Index<rpm package version record>Global*	lin
short rpm package version record <short rpm package version record>	short rpm package version records	<short rpm package version record>	<world>	*Index<short rpm package version record>Global*	lin
significance place <integer> of <floating point>	significance places	<floating point>	<floating point>	*Numbered*	core
significance place of <floating point>	significance places	<floating point>	<floating point>	*Plain*	core
significance threshold of <floating point>	significance thresholds	<floating point>	<floating point>	*Plain*	core
significant digits <integer> of <hertz>	significant digitss	<hertz>	<hertz>	*Numbered*	core
significant digits <integer> of <integer>	significant digitss	<integer>	<integer>	*Numbered*	core
singular name of <property>	singular names	<string>	<property>	*Plain*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
site	sites	<site>	<world>	*PlainGlobal*	lin
site <string>	sites	<site>	<world>	*NamedGlobal*	lin
site number of <license>	site numbers	<integer>	<license>	*Plain*	lin
site of <fixlet>	sites	<site>	<fixlet>	*Plain*	lin
site tag of <site>	site tags	<string>	<site>	*Plain*	lin
site url of <bes product>	site urls	<string>	<bes product>	*Plain*	lin
site version list <string>	site version lists	<site version list>	<world>	*NamedGlobal*	core
site version list of <site>	site version lists	<site version list>	<site>	*Plain*	lin
size of <application usage summary instance>	sizes	<integer>	<application usage summary instance>	*Plain*	lin
size of <file>	sizes	<integer>	<file>	*Plain*	lin
size of <filesystem>	sizes	<integer>	<filesystem>	*Plain*	lin
size of <integer set>	sizes	<integer>	<integer set>	*Plain*	core
size of <ram>	sizes	<integer>	<ram>	*Plain*	lin
size of <string set>	sizes	<integer>	<string set>	*Plain*	core
size of <swap>	sizes	<integer>	<swap>	*Plain*	lin
size of <type>	sizes	<integer>	<type>	*Plain*	core
small <string> of <html>	smalls	<html>	<html>	*Named*	core
small <string> of <string>	smalls	<html>	<string>	*Named*	core
small of <html>	smalls	<html>	<html>	*Plain*	core
small of <string>	smalls	<html>	<string>	*Plain*	core
smbios	smbioses	<smbios>	<world>	*PlainGlobal*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
socket file <filesystem object>	socket files	<socket file>	<world>	*Index<filesystem object>Global*	lin
socket file <string>	socket files	<socket file>	<world>	*NamedGlobal*	lin
socket file <string> of <folder>	socket files	<socket file>	<folder>	*Named*	lin
socket file <symlink>	socket files	<socket file>	<world>	*Index<symlink>Global*	lin
socket file of <folder>	socket files	<socket file>	<folder>	*Plain*	lin
span <string> of <html>	spans	<html>	<html>	*Named*	core
span <string> of <string>	spans	<html>	<string>	*Named*	core
span of <html>	spans	<html>	<html>	*Plain*	core
span of <string>	spans	<html>	<string>	*Plain*	core
speed of <processor>	speeds	<hertz>	<processor>	*Plain*	lin
start date of <license>	start dates	<time>	<license>	*Plain*	lin
start of <substring>	starts	<string position>	<substring>	*Plain*	core
start of <time range>	starts	<time>	<time range>	*Plain*	core
state of <service>	states	<string>	<service>	*Plain*	lin
status of <action>	statuss	<string>	<action>	*Plain*	lin
stepping of <processor>	steppings	<integer>	<processor>	*Plain*	lin
sticky of <mode>	stickies	<boolean>	<mode>	*Plain*	lin
string <string>	strings	<string>	<world>	*NamedGlobal*	core
string <string> of <smbios structure>	strings	<string>	<smbios structure>	*Named*	lin
string value <string> of <smbios structure>	string values	<smbios value>	<smbios structure>	*Named*	lin
string version of <application usage summary instance>	string versions	<string>	<application usage summary instance>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
strong <string> of <html>	strongs	<html>	<html>	*Named*	core
strong <string> of <string>	strongs	<html>	<string>	*Named*	core
strong of <html>	strongs	<html>	<html>	*Plain*	core
strong of <string>	strongs	<html>	<string>	*Plain*	core
structure <string> of <smbios>	structures	<smbios structure>	<smbios>	*Named*	lin
structure of <smbios value>	structures	<smbios structure>	<smbios value>	*Plain*	lin
structure of <smbios>	structures	<smbios structure>	<smbios>	*Plain*	lin
strverscmp version <string>	strverscmp versions	<strverscmp version>	<world>	*NamedGlobal*	lin
sub <string> of <html>	subs	<html>	<html>	*Named*	core
sub <string> of <string>	subs	<html>	<string>	*Named*	core
sub of <html>	subs	<html>	<html>	*Plain*	core
sub of <string>	subs	<html>	<string>	*Plain*	core
subnet address of <network adapter interface>	subnet addresses	<ipv4or6 address>	<network adapter interface>	*Plain*	lin
subnet address of <network adapter>	subnet addresses	<ipv4 address>	<network adapter>	*Plain*	lin
subnet address of <network ip interface>	subnet addresses	<ipv4 address>	<network ip interface>	*Plain*	lin
subnet mask of <network adapter interface>	subnet masks	<ipv4or6 address>	<network adapter interface>	*Plain*	lin
subnet mask of <network adapter>	subnet masks	<ipv4 address>	<network adapter>	*Plain*	lin
subnet mask of <network ip interface>	subnet masks	<ipv4 address>	<network ip interface>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
subscribe time of <site>	subscribe times	<time>	<site>	*Plain*	lin
substring <(integer, integer)> of <string>	substrings	<substring>	<string>	*Index<(integer, integer)>*	core
substring <string> of <string>	substrings	<substring>	<string>	*Named*	core
substring after <string> of <string>	substrings after	<substring>	<string>	*Named*	core
substring before <string> of <string>	substrings before	<substring>	<string>	*Named*	core
substring between <string> of <string>	substrings between	<substring>	<string>	*Named*	core
substring separated by <string> of <string>	substrings separated by	<substring>	<string>	*Named*	core
sum of <floating point>	sums	<floating point>	<floating point>	*Plain*	core
sum of <integer>	sums	<integer>	<integer>	*Plain*	core
sum of <time interval>	sums	<time interval>	<time interval>	*Plain*	core
sunday	sundays	<day of week>	<world>	*PlainGlobal*	core
sup <string> of <html>	sups	<html>	<html>	*Named*	core
sup <string> of <string>	sups	<html>	<string>	*Named*	core
sup of <html>	sups	<html>	<html>	*Plain*	core
sup of <string>	sups	<html>	<string>	*Plain*	core
swap	swaps	<swap>	<world>	*PlainGlobal*	lin
symbol of <binary operator>	symbols	<string>	<binary operator>	*Plain*	core
symbol of <unary operator>	symbols	<string>	<unary operator>	*Plain*	core
symlink <filesystem object>	symlinks	<symlink>	<world>	*Index<filesystem object>Global*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
symlink <string>	symlinks	<symlink>	<world>	*NamedGlobal*	lin
symlink <string> of <folder>	symlinks	<symlink>	<folder>	*Named*	lin
symlink <symlink>	symlinks	<symlink>	<world>	*Index<symlink>Global*	lin
symlink of <folder>	symlinks	<symlink>	<folder>	*Plain*	lin
system language	system languages	<string>	<world>	*PlainGlobal*	lin
system locale	system locales	<language>	<world>	*PlainGlobal*	lin
system ui language	system ui languages	<language>	<world>	*PlainGlobal*	lin
table <string> of <html>	tables	<html>	<html>	*Named*	core
table <string> of <string>	tables	<html>	<string>	*Named*	core
table of <html>	tables	<html>	<html>	*Plain*	core
table of <string>	tables	<html>	<string>	*Plain*	core
tbody <string> of <html>	tbodys	<html>	<html>	*Named*	core
tbody <string> of <string>	tbodys	<html>	<string>	*Named*	core
tbody of <html>	tbodys	<html>	<html>	*Plain*	core
tbody of <string>	tbodys	<html>	<string>	*Plain*	core
td <string> of <html>	tds	<html>	<html>	*Named*	core
td <string> of <string>	tds	<html>	<string>	*Named*	core
td of <html>	tds	<html>	<html>	*Plain*	core
td of <string>	tds	<html>	<string>	*Plain*	core
tfoot <string> of <html>	tfoots	<html>	<html>	*Named*	core
tfoot <string> of <string>	tfoots	<html>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
tfoot of <html>	tfoots	<html>	<html>	*Plain*	core
tfoot of <string>	tfoots	<html>	<string>	*Plain*	core
th <string> of <html>	ths	<html>	<html>	*Named*	core
th <string> of <string>	ths	<html>	<string>	*Named*	core
th of <html>	ths	<html>	<html>	*Plain*	core
th of <string>	ths	<html>	<string>	*Plain*	core
thead <string> of <html>	theads	<html>	<html>	*Named*	core
thead <string> of <string>	theads	<html>	<string>	*Named*	core
thead of <html>	theads	<html>	<html>	*Plain*	core
thead of <string>	theads	<html>	<string>	*Plain*	core
thursday	thursdays	<day of week>	<world>	*PlainGlobal*	core
time <string>	times	<time>	<world>	*NamedGlobal*	core
time <time zone> of <time>	times	<time of day with time zone>	<time>	*Index<time zone>*	core
time interval <string>	time intervals	<time interval>	<world>	*NamedGlobal*	core
time of <time of day with time zone>	times	<time of day>	<time of day with time zone>	*Plain*	core
time zone <string>	time zones	<time zone>	<world>	*NamedGlobal*	core
time_of_day <string>	times_of_day	<time of day>	<world>	*NamedGlobal*	core
title <string> of <html>	titles	<html>	<html>	*Named*	core
title <string> of <string>	titles	<html>	<string>	*Named*	core
title of <html>	titles	<html>	<html>	*Plain*	core
title of <string>	titles	<html>	<string>	*Plain*	core
total amount of <ram>	total amounts	<integer>	<ram>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
total amount of <swap>	total amounts	<integer>	<swap>	*Plain*	lin
total duration of <application usage summary instance>	total durations	<time interval>	<application usage summary instance>	*Plain*	lin
total duration of <application usage summary>	total durations	<time interval>	<application usage summary>	*Plain*	lin
total run count of <application usage summary instance>	total run counts	<integer>	<application usage summary instance>	*Plain*	lin
total run count of <application usage summary>	total run counts	<integer>	<application usage summary>	*Plain*	lin
total size of <download storage folder>	total sizes	<integer>	<download storage folder>	*Plain*	lin
total space of <filesystem>	total spaces	<integer>	<filesystem>	*Plain*	lin
tr <string> of <html>	trs	<html>	<html>	*Named*	core
tr <string> of <string>	trs	<html>	<string>	*Named*	core
tr of <html>	trs	<html>	<html>	*Plain*	core
tr of <string>	trs	<html>	<string>	*Plain*	core
true	trues	<boolean>	<world>	*PlainGlobal*	core
tt <string> of <html>	tts	<html>	<html>	*Named*	core
tt <string> of <string>	tts	<html>	<string>	*Named*	core
tt of <html>	tts	<html>	<html>	*Plain*	core
tt of <string>	tts	<html>	<string>	*Plain*	core
tty of <user>	ttys	<string>	<user>	*Plain*	lin
tuesday	tuesdays	<day of week>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
tuple string item <integer> of <string>	tuple string items	<string>	<string>	*Numbered*	core
tuple string item of <string>	tuple string items	<string>	<string>	*Plain*	core
two digit hour of <time of day with time zone>	two digit hours	<string>	<time of day with time zone>	*Plain*	core
two digit hour of <time of day>	two digit hours	<string>	<time of day>	*Plain*	core
two digit minute of <time of day with time zone>	two digit minutes	<string>	<time of day with time zone>	*Plain*	core
two digit minute of <time of day>	two digit minutes	<string>	<time of day>	*Plain*	core
two digit second of <time of day with time zone>	two digit seconds	<string>	<time of day with time zone>	*Plain*	core
two digit second of <time of day>	two digit seconds	<string>	<time of day>	*Plain*	core
type of <filesystem>	types	<string>	<filesystem>	*Plain*	lin
type of <license>	types	<string>	<license>	*Plain*	lin
type of <site>	types	<string>	<site>	*Plain*	lin
type of <smbios structure>	types	<integer>	<smbios structure>	*Plain*	lin
type of <smbios value>	types	<string>	<smbios value>	*Plain*	lin
uid of <filesystem object>	uids	<integer>	<filesystem object>	*Plain*	lin
uid of <symlink>	uids	<integer>	<symlink>	*Plain*	lin
ul <string> of <html>	uls	<html>	<html>	*Named*	core
ul <string> of <string>	uls	<html>	<string>	*Named*	core
ul of <html>	uls	<html>	<html>	*Plain*	core
ul of <string>	uls	<html>	<string>	*Plain*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
unary operator <string>	unary operators	<unary operator>	<world>	*NamedGlobal*	core
unary operator returning <type>	unary operators returning	<unary operator>	<world>	*Index<type>Global*	core
unavailable amount of <ram>	unavailable amounts	<integer>	<ram>	*Plain*	lin
underflow of <floating point>	underflows	<boolean>	<floating point>	*Plain*	core
union of <integer set>	unions	<integer set>	<integer set>	*Plain*	core
union of <string set>	unions	<string set>	<string set>	*Plain*	core
unique name of <package>	unique names	<string>	<package>	*Plain*	lin
unique value of <date>	unique values	<date with multiplicity>	<date>	*Plain*	core
unique value of <day of month>	unique values	<day of month with multiplicity>	<day of month>	*Plain*	core
unique value of <day of week>	unique values	<day of week with multiplicity>	<day of week>	*Plain*	core
unique value of <day of year>	unique values	<day of year with multiplicity>	<day of year>	*Plain*	core
unique value of <floating point>	unique values	<floating point with multiplicity>	<floating point>	*Plain*	core
unique value of <hertz>	unique values	<hertz with multiplicity>	<hertz>	*Plain*	core
unique value of <integer>	unique values	<integer with multiplicity>	<integer>	*Plain*	core
unique value of <ipv4 address>	unique values	<ipv4 address with multiplicity>	<ipv4 address>	*Plain*	core
unique value of <ipv4or6 address>	unique values	<ipv4or6 address with multiplicity>	<ipv4or6 address>	*Plain*	core
unique value of <ipv6 address>	unique values	<ipv6 address with multiplicity>	<ipv6 address>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unique value of <month and year>	unique values	<month and year with multiplicity>	<month and year>	*Plain*	core
unique value of <month>	unique values	<month with multiplicity>	<month>	*Plain*	core
unique value of <number of months>	unique values	<number of months with multiplicity>	<number of months>	*Plain*	core
unique value of <rpm package release>	unique values	<rpm package release with multiplicity>	<rpm package release>	*Plain*	lin
unique value of <rpm package version record>	unique values	<rpm package version record with multiplicity>	<rpm package version record>	*Plain*	lin
unique value of <rpm package version>	unique values	<rpm package version with multiplicity>	<rpm package version>	*Plain*	lin
unique value of <short rpm package version record>	unique values	<short rpm package version record with multiplicity>	<short rpm package version record>	*Plain*	lin
unique value of <site version list>	unique values	<site version list with multiplicity>	<site version list>	*Plain*	core
unique value of <string>	unique values	<string with multiplicity>	<string>	*Plain*	core
unique value of <time interval>	unique values	<time interval with multiplicity>	<time interval>	*Plain*	core
unique value of <time of day with time zone>	unique values	<time of day with time zone with multiplicity>	<time of day with time zone>	*Plain*	core
unique value of <time of day>	unique values	<time of day with multiplicity>	<time of day>	*Plain*	core
unique value of <time range>	unique values	<time range with multiplicity>	<time range>	*Plain*	core
unique value of <time zone>	unique values	<time zone with multiplicity>	<time zone>	*Plain*	core
unique value of <time>	unique values	<time with multiplicity>	<time>	*Plain*	core

Key Phrase	Plural	Creates a	From a	Form	Ref
unique value of <version>	unique values	<version with multiplicity>	<version>	*Plain*	core
unique value of <year>	unique values	<year with multiplicity>	<year>	*Plain*	core
universal time <string>	universal times	<time>	<world>	*NamedGlobal*	core
universal time zone	universal time zones	<time zone>	<world>	*PlainGlobal*	core
unix of <operating system>	unixes	<boolean>	<operating system>	*Plain*	lin
unordered list <string> of <html>	unordered lists	<html>	<html>	*Named*	core
unordered list <string> of <string>	unordered lists	<html>	<string>	*Named*	core
unordered list of <html>	unordered lists	<html>	<html>	*Plain*	core
unordered list of <string>	unordered lists	<html>	<string>	*Plain*	core
up of <network adapter interface>	ups	<boolean>	<network adapter interface>	*Plain*	lin
up of <network adapter>	ups	<boolean>	<network adapter>	*Plain*	lin
up of <network ip interface>	ups	<boolean>	<network ip interface>	*Plain*	lin
upload progress of <client>	upload progresses	<string>	<client>	*Plain*	lin
ups of <power level>	upss	<boolean>	<power level>	*Plain*	lin
uptime of <operating system>	uptimes	<time interval>	<operating system>	*Plain*	lin
url of <site>	urls	<string>	<site>	*Plain*	lin
used amount of <ram>	used amounts	<integer>	<ram>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	*Form*	Ref
used amount of <swap>	used amounts	<integer>	<swap>	*Plain*	lin
used file count of <filesystem>	used file counts	<integer>	<filesystem>	*Plain*	lin
used percent of <filesystem>	used percents	<integer>	<filesystem>	*Plain*	lin
used space of <filesystem>	used spaces	<integer>	<filesystem>	*Plain*	lin
user	users	<user>	<world>	*PlainGlobal*	lin
user <string>	users	<user>	<world>	*NamedGlobal*	lin
user execute of <filesystem object>	user executes	<boolean>	<filesystem object>	*Plain*	lin
user mask of <filesystem object>	user masks	<integer>	<filesystem object>	*Plain*	lin
user mask of <mode>	user masks	<mode_mask>	<mode>	*Plain*	lin
user name of <filesystem object>	user names	<string>	<filesystem object>	*Plain*	lin
user name of <symlink>	user names	<string>	<symlink>	*Plain*	lin
user read of <filesystem object>	user reads	<boolean>	<filesystem object>	*Plain*	lin
user write of <filesystem object>	user writes	<boolean>	<filesystem object>	*Plain*	lin
usual name of <property>	usual names	<string>	<property>	*Plain*	core
value <string> of <smbios structure>	values	<smbios value>	<smbios structure>	*Named*	lin
value accessible of <symlink>	values accessible	<boolean>	<symlink>	*Plain*	lin
value of <environment variable>	values	<string>	<environment variable>	*Plain*	lin
value of <fixlet_header>	values	<string>	<fixlet_header>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
value of \<runlevel>	values	\<string>	\<runlevel>	*Plain*	lin
value of \<setting>	values	\<string>	\<setting>	*Plain*	lin
value of \<smbios structure>	values	\<smbios value>	\<smbios structure>	*Plain*	lin
value of \<symlink>	values	\<string>	\<symlink>	*Plain*	lin
var \<string> of \<html>	vars	\<html>	\<html>	*Named*	core
var \<string> of \<string>	vars	\<html>	\<string>	*Named*	core
var of \<html>	vars	\<html>	\<html>	*Plain*	core
var of \<string>	vars	\<html>	\<string>	*Plain*	core
variable \<string> of \<environment>	variables	\<environment variable>	\<environment>	*Named*	lin
variable of \<environment>	variables	\<environment variable>	\<environment>	*Plain*	lin
variable of \<file>	variables	\<string>	\<file>	*Plain*	lin
vendor name of \<processor>	vendor names	\<string>	\<processor>	*Plain*	lin
version \<string>	versions	\<version>	\<world>	*NamedGlobal*	core
version of \<application usage summary instance>	versions	\<version>	\<application usage summary instance>	*Plain*	lin
version of \<bios>	versions	\<string>	\<bios>	*Plain*	lin
version of \<capability>	versions	\<string>	\<capability>	*Plain*	lin
version of \<client>	versions	\<version>	\<client>	*Plain*	lin
version of \<current relay>	versions	\<version>	\<current relay>	*Plain*	lin
version of \<package>	versions	\<version>	\<package>	*Plain*	lin
version of \<rpm package version record>	versions	\<rpm package version>	\<rpm package version record>	*Plain*	lin

Key Phrase	Plural	Creates a	From a	Form	Ref
version of <service>	versions	<version>	<service>	Plain	lin
version of <short rpm package version record>	versions	<rpm package version>	<short rpm package version record>	Plain	lin
version of <site>	versions	<integer>	<site>	Plain	lin
version string <string> of <module>	version strings	<string>	<module>	Named	core
waiting for download of <action>	waiting for downloads	<boolean>	<action>	Plain	lin
wake on lan subnet cidr string	wake on lan subnet cidr strings	<string>	<world>	PlainGlobal	lin
wednesday	wednesdays	<day of week>	<world>	PlainGlobal	core
week	weeks	<time interval>	<world>	PlainGlobal	core
weight of <selected server>	weights	<integer>	<selected server>	Plain	lin
windows of <operating system>	windowses	<boolean>	<operating system>	Plain	lin
windows server count of <bes product>	windows server counts	<integer>	<bes product>	Plain	lin
workstation count of <bes product>	workstation counts	<integer>	<bes product>	Plain	lin
wp of <processor>	wps	<boolean>	<processor>	Plain	lin
write of <mode_mask>	writes	<boolean>	<mode_mask>	Plain	lin
year	years	<number of months>	<world>	PlainGlobal	core
year <integer>	years	<year>	<world>	NumberedGlobal	core
year <string>	years	<year>	<world>	NamedGlobal	core
year of <date>	years	<year>	<date>	Plain	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
year of <month and year>	years	<year>	<month and year>	*Plain*	core
zone of <time of day with time zone>	zones	<time zone>	<time of day with time zone>	*Plain*	core
zoned time_of_day <string>	zoned times_of_day	<time of day with time zone>	<world>	*NamedGlobal*	core

Casting Operators

Casting operators help you to convert one object type into another. This section contains those casting operators pertinent to this guide, as well as the core and regex inspectors, which are available in all contexts.

Key Phrase	Creates a	From a
<action lock state> as string	<string>	<action lock state>
<binary operator> as string	<string>	<binary operator>
<bios> as string	<string>	<bios>
<bit set> as integer	<integer>	<bit set>
<bit set> as string	<string>	<bit set>
<boolean> as boolean	<boolean>	<boolean>
<boolean> as string	<string>	<boolean>
<capability> as string	<string>	<capability>
<cast> as string	<string>	<cast>
<date> as string	<string>	<date>
<day of month> as integer	<integer>	<day of month>
<day of month> as string	<string>	<day of month>
<day of month> as two digits	<string>	<day of month>
<day of week> as string	<string>	<day of week>
<day of week> as three letters	<string>	<day of week>
<day of year> as string	<string>	<day of year>
<environment variable> as string	<string>	<environment variable>
<file content> as lowercase	<file content>	<file content>
<file content> as uppercase	<file content>	<file content>
<filesystem object> as device file	<device file>	<filesystem object>

Key Phrase	Creates a	From a
<filesystem object> as fifo file	<fifo file>	<filesystem object>
<filesystem object> as socket file	<socket file>	<filesystem object>
<filesystem object> as string	<string>	<filesystem object>
<filesystem object> as symlink	<symlink>	<filesystem object>
<floating point> as floating point	<floating point>	<floating point>
<floating point> as integer	<integer>	<floating point>
<floating point> as scientific notation	<string>	<floating point>
<floating point> as standard notation	<string>	<floating point>
<floating point> as string	<string>	<floating point>
<hertz> as string	<string>	<hertz>
<html> as html	<html>	<html>
<html> as string	<string>	<html>
<integer> as bit set	<bit set>	<integer>
<integer> as bits	<bit set>	<integer>
<integer> as day_of_month	<day of month>	<integer>
<integer> as floating point	<floating point>	<integer>
<integer> as hexadecimal	<string>	<integer>
<integer> as integer	<integer>	<integer>
<integer> as month	<month>	<integer>
<integer> as string	<string>	<integer>
<integer> as year	<year>	<integer>
<ip version> as string	<string>	<ip version>
<ipv4 address> as ipv4or6 address	<ipv4or6 address>	<ipv4 address>
<ipv4 address> as ipv6 address	<ipv6 address>	<ipv4 address>

Key Phrase	Creates a	From a
<ipv4 address> as string	<string>	<ipv4 address>
<ipv4or6 address> as compressed string	<string>	<ipv4or6 address>
<ipv4or6 address> as compressed string with ipv4	<string>	<ipv4or6 address>
<ipv4or6 address> as compressed string with ipv4 with zone index	<string>	<ipv4or6 address>
<ipv4or6 address> as compressed string with zone index	<string>	<ipv4or6 address>
<ipv4or6 address> as string	<string>	<ipv4or6 address>
<ipv4or6 address> as string with ipv4	<string>	<ipv4or6 address>
<ipv4or6 address> as string with ipv4 with zone index	<string>	<ipv4or6 address>
<ipv4or6 address> as string with leading zeros	<string>	<ipv4or6 address>
<ipv4or6 address> as string with leading zeros with zone index	<string>	<ipv4or6 address>
<ipv4or6 address> as string with zone index	<string>	<ipv4or6 address>
<ipv6 address> as compressed string	<string>	<ipv6 address>
<ipv6 address> as compressed string with ipv4	<string>	<ipv6 address>
<ipv6 address> as compressed string with ipv4 with zone index	<string>	<ipv6 address>
<ipv6 address> as compressed string with zone index	<string>	<ipv6 address>
<ipv6 address> as ipv4or6 address	<ipv4or6 address>	<ipv6 address>
<ipv6 address> as string	<string>	<ipv6 address>
<ipv6 address> as string with ipv4	<string>	<ipv6 address>
<ipv6 address> as string with ipv4 with zone index	<string>	<ipv6 address>

Key Phrase	Creates a	From a
<ipv6 address> as string with leading zeros	<string>	<ipv6 address>
<ipv6 address> as string with leading zeros with zone index	<string>	<ipv6 address>
<ipv6 address> as string with zone index	<string>	<ipv6 address>
<language> as string	<string>	<language>
<mode_mask> as integer	<integer>	<mode_mask>
<mode_mask> as string	<string>	<mode_mask>
<mode> as octal string	<string>	<mode>
<mode> as string	<string>	<mode>
<month and year> as string	<string>	<month and year>
<month> as integer	<integer>	<month>
<month> as string	<string>	<month>
<month> as three letters	<string>	<month>
<month> as two digits	<string>	<month>
<number of months> as string	<string>	<number of months>
<operating system> as string	<string>	<operating system>
<package> as string	<string>	<package>
<power level> as string	<string>	<power level>
<primary language> as string	<string>	<primary language>
<property> as string	<string>	<property>
<rope> as string	<string>	<rope>
<rpm package release> as string	<string>	<rpm package release>

Key Phrase	Creates a	From a
<rpm package version record> as short rpm package version record	<short rpm package version record>	<rpm package version record>
<rpm package version record> as string	<string>	<rpm package version record>
<rpm package version> as string	<string>	<rpm package version>
<runlevel> as string	<string>	<runlevel>
<setting> as string	<string>	<setting>
<short rpm package version record> as rpm package version record	<rpm package version record>	<short rpm package version record>
<short rpm package version record> as string	<string>	<short rpm package version record>
<site version list> as string	<string>	<site version list>
<smbios value> as string	<string>	<smbios value>
<string> as boolean	<boolean>	<string>
<string> as date	<date>	<string>
<string> as day_of_month	<day of month>	<string>
<string> as day_of_week	<day of week>	<string>
<string> as floating point	<floating point>	<string>
<string> as hexadecimal	<string>	<string>
<string> as html	<html>	<string>
<string> as integer	<integer>	<string>
<string> as ipv4or6 address	<ipv4or6 address>	<string>
<string> as left trimmed string	<string>	<string>
<string> as local time	<time>	<string>

Key Phrase	Creates a	From a
<string> as local zoned time_of_day	<time of day with time zone>	<string>
<string> as lowercase	<string>	<string>
<string> as month	<month>	<string>
<string> as right trimmed string	<string>	<string>
<string> as site version list	<site version list>	<string>
<string> as string	<string>	<string>
<string> as strverscmp version	<strverscmp version>	<string>
<string> as time	<time>	<string>
<string> as time interval	<time interval>	<string>
<string> as time zone	<time zone>	<string>
<string> as time_of_day	<time of day>	<string>
<string> as trimmed string	<string>	<string>
<string> as universal time	<time>	<string>
<string> as universal zoned time_of_day	<time of day with time zone>	<string>
<string> as uppercase	<string>	<string>
<string> as version	<version>	<string>
<string> as year	<year>	<string>
<string> as zoned time_of_day	<time of day with time zone>	<string>
<symlink> as device file	<device file>	<symlink>
<symlink> as file	<file>	<symlink>
<symlink> as folder	<folder>	<symlink>
<symlink> as socket file	<socket file>	<symlink>

Key Phrase	Creates a	From a
<symlink> as string	<string>	<symlink>
<symlink> as symlink	<symlink>	<symlink>
<time interval> as string	<string>	<time interval>
<time of day with time zone> as string	<string>	<time of day with time zone>
<time of day> as string	<string>	<time of day>
<time range> as string	<string>	<time range>
<time zone> as string	<string>	<time zone>
<time> as local string	<string>	<time>
<time> as string	<string>	<time>
<time> as universal string	<string>	<time>
<type> as string	<string>	<type>
<unary operator> as string	<string>	<unary operator>
<undefined> as string	<string>	<undefined>
<version> as string	<string>	<version>
<version> as version	<version>	<version>
<year> as integer	<integer>	<year>
<year> as string	<string>	<year>

Resources

DMI Inspectors

The desktop management task force (DMTF) defined a set of standards and API's for accessing asset data from the BIOS of the client computer. A BIOS that supports these interfaces is said to be DMI or SMBIOS compliant. These Inspectors were originally provided by BigFix on both Windows and Linux platforms.

Due to potential problems with 16-bit applications, the DMI Inspectors were deprecated in version 6. If you are running an earlier version of BES, you can turn off these Inspectors by setting the BigFix property named _BESClient_Inspector_DisableDMI to 1. A Fixlet or Task that uses a disabled inspector will report false; retrieved properties that request a disabled inspector value will report an error. For a complete list of these deprecated Inspectors, see the **DMI Inspector Guide**. For the latest information on issues surrounding the DMI inspectors, search the BigFix support knowledge base.

Instead of using DMI, use the SMBIOS Inspectors for equivalent functionality. See the next section for a complete list of SMBIOS Inspectors.

SMBIOS Inspectors

The SMBIOS Inspectors were created to replace the deprecated DMI Inspectors, but they are implemented differently. Instead of a unique Inspector for each DMI object, the SMBIOS Inspectors use various strings to retrieve named objects. This allows you to access the same information as the DMI Inspectors without introducing many new keywords to the language. It is simple to convert a DMI Inspector to an equivalent SMBIOS Inspector. For instance, to find out the version number of a client BIOS, you would previously have used a DMI Inspector of the form:

- `bios_version of bios_information of dmi`

The same information can now be retrieved by passing strings to the SMBIOS Inspectors:

- `string value "bios_version" of structure "bios_information" of smbios`

The following table lists the inspectable SMBIOS structures, their constituent objects and the data type of each object. This set is based on version 2.61 of the SMBIOS standard. For more details, see the SMBIOS documentation at: http://www.dmtf.org/.

bios_information	
String	vendor
String	bios_version
Word	bios_starting_address_segment
String	bios_release_date
Byte	bios_rom_size
QWord	bios_characteristics

Byte	system_bios_major_release
Byte	system_bios_minor_release
Byte	embedded_controller_firmware_major_release
Byte	embedded_controller_firmware_minor_release

system_information

String	manufacturer
String	product_name
String	version
String	serial_number
BinaryString	uuid
Byte	wake_up_type
String	sku_number
String	family

base_board_information

String	manufacturer
String	product
String	version
String	serial_number
String	asset_tag
Byte	feature_flags
String	location_in_chassis
Word	chassis_handle
Byte	board_type
Byte	number_of_contained_object_handles

system_enclosure_or_chassis

String	manufacturer
Byte	type
String	version
String	serial_number
String	asset_tag_number
Byte	bootup_state
Byte	power_supply_state
Byte	thermal_state
Byte	security_status
DWord	oem_defined
Byte	height
Byte	number_of_power_cords
Byte	contained_element_count
Byte	contained_element_record_length

processor_information

String	socket_designation
Byte	processor_type
Byte	processor_family

String	processor_manufacturer
QWord	processor_id
String	processor_version
Byte	voltage
Word	external_clock
Word	max_speed
Word	current_speed
Byte	status
Byte	processor_upgrade
Word	l1_cache_handle
Word	l2_cache_handle
Word	l3_cache_handle
String	serial_number
String	asset_tag
String	part_number
Byte	core_count
Byte	core_enabled
Byte	thread_count
Word	processor_characteristics
Word	processor_family_2

memory_controller_information

Byte	error_detecting_method
Byte	error_correcting_capability
Byte	supported_interleave
Byte	current_interleave
Byte	maximum_memory_module_size
Word	supported_speeds
Word	supported_memory_types
Byte	memory_module_voltage
Byte	number_of_associated_memory_slots

memory_module_information

String	socket_designation
Byte	bank_connections
Byte	current_speed
Word	current_memory_type
Byte	installed_size
Byte	enabled_size
Byte	error_status

cache_information

String	socket_designation
Word	cache_configuration
Word	maximum_cache_size

Word	installed_size
Word	supported_sram_type
Word	current_sram_type
Byte	cache_speed
Byte	error_correction_type
Byte	system_cache_type
Byte	associativity

port_connector_information

String	internal_reference_designator
Byte	internal_connector_type
String	external_reference_designator
Byte	external_connector_type
Byte	port_type

system_slots

String	slot_designation
Byte	slot_type
Byte	slot_data_bus_width
Byte	current_usage
Byte	slot_length
Word	slot_id
Byte	slot_characteristics_1
Byte	slot_characteristics_2
Word	segment_group_number
Byte	bus_number
Byte	device_function_number

on_board_devices_information

Byte	device_type
String	description_string

oem_strings

Byte	count
MultipleString	string

system_configuration_options

Byte	count
MultipleString	string

bios_language_information

Byte	installable_languages
Byte	flags
BinaryString	reserved
String	current_language

group_associations

String	group_name
Byte	item_type
Word	item_handle

physical_memory_array	
Byte	location
Byte	use
Byte	memory_error_correction
DWord	maximum_capacity
Word	memory_error_information_handle
Word	number_of_memory_devices

memory_device	
Word	memory_array_handle
Word	memory_error_information_handle
Word	total_width
Word	data_width
Word	size
Byte	form_factor
Byte	device_set
String	device_locator
String	bank_locator
Byte	memory_type
Word	type_detail
Word	speed
String	manufacturer
String	serial_number
String	asset_tag
String	part_number
Byte	attributes

b32_bit_memory_error_information	
Byte	error_type
Byte	error_granularity
Byte	error_operation
DWord	vendor_syndrome
DWord	memory_array_error_address
DWord	device_error_address
DWord	error_resolution

memory_array_mapped_address	
DWord	starting_address
DWord	ending_address
Word	memory_array_handle
Byte	partition_width

memory_device_mapped_address	
DWord	starting_address
DWord	ending_address
Word	memory_device_handle

Word	memory_array_mapped_address_handle
Byte	partition_row_position
Byte	interleave_position
Byte	interleaved_data_depth

built_in_pointing_device

Byte	type
Byte	interface
Byte	number_of_buttons

portable_battery

String	location
String	manufacturer
String	manufacture_date
String	serial_number
String	device_name
Byte	device_chemistry
Word	design_capacity
Word	design_voltage
String	sbds_version_number
Byte	maximum_error_in_battery_data
Word	sbds_serial_number
Word	sbds_manufacture_date
String	sbds_device_chemistry
Byte	design_capacity_multiplier
DWord	oem_specific

system_reset

Byte	capabilities
Word	reset_count
Word	reset_limit
Word	timer_interval
Word	timeout

hardware_security

Byte	hardware_security_settings

system_power_controls

Byte	next_scheduled_power_on_month
Byte	next_scheduled_power_on_day_of_month
Byte	next_scheduled_power_on_hour
Byte	next_scheduled_power_on_minute
Byte	next_scheduled_power_on_second

voltage_probe

String	description
Byte	location_and_status
Word	maximum_value
Word	minimum_value

Word	resolution
Word	tolerance
Word	accuracy
DWord	oem_defined
Word	nominal_value
cooling_device	
Word	temperature_probe_handle
Byte	device_type_and_status
Byte	cooling_unit_group
DWord	oem_defined
Word	nominal_speed
temperature_probe	
String	description
Byte	location_and_status
Word	maximum_value
Word	minimum_value
Word	resolution
Word	tolerance
Word	accuracy
DWord	oem_defined
Word	nominal_value
electrical_current_probe	
String	description
Byte	location_and_status
Word	maximum_value
Word	minimum_value
Word	resolution
Word	tolerance
Word	accuracy
DWord	oem_defined
Word	nominal_value
out_of_band_remote_access	
String	manufacturer_name
Byte	connections
system_boot_information	
BinaryString	reserved
b64_bit_memory_error_information	
Byte	error_type
Byte	error_granularity
Byte	error_operation
DWord	vendor_syndrome
QWord	memory_array_error_address

QWord	device_error_address
DWord	error_resolution

management_device

String	description
Byte	type
DWord	address
Byte	address_type

management_device_component

String	description
Word	management_device_handle
Word	component_handle
Word	threshold_handle

management_device_threshold_data

Word	lower_threshold_non_critical
Word	upper_threshold_non_critical
Word	lower_threshold_critical
Word	upper_threshold_critical
Word	lower_threshold_non_recoverable
Word	upper_threshold_non_recoverable

memory_channel

Byte	channel_type
Byte	maximum_channel_load
Byte	memory_device_count
Byte	memory_device_load
Word	memory_device_handle

ipmi_device_information

Byte	interface_type
Byte	ipmi_specification_revision
Byte	i2c_slave_address
Byte	nv_storage_device_address
QWord	base_address

system_power_supply

Byte	power_unit_group
String	location
String	device_name
String	manufacturer
String	serial_number
String	asset_tag_number
String	model_part_number
String	revision_level
Word	max_power_capacity
Word	power_supply_characteristics
Word	input_voltage_probe_handle

Word	cooling_device_handle
Word	input_current_probe_handle

additional_information

Byte	number_of_additional_information_entries

onboard_devices_extended_information

Byte	reference_designation
Byte	device_type
Byte	device_type_instance
Word	segment_group_number
Byte	bus_number
Byte	device_function_number

Part Four

Notices

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this

document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such

provisions are inconsistent with local law: 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. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

2Z4A/101

11400 Burnet Road

Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their

published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

TRADEMARKS:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also

be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

A

B

C

D

E

expiration state of <license> · 72, 128
extrema of <rpm package release> · 101, 129
extrema of <rpm package version record> · 107, 129
extrema of <rpm package version> · 104, 129
extrema of <short rpm package version record> · 110, 129

F

f00f bug of <processor> · 41, 130
family name of <processor> · 41, 130
family of <network interface> · 83, 130
family of <processor> · 41, 130
fdiv bug of <processor> · 41, 130
fifo file · 11, 22, 31, 36, 127, 130, 131, 180
fifo file <filesystem object> · 36, 130
fifo file <string> · 22, 36, 130
fifo file <string> of <folder> · 22, 36, 130
fifo file <symlink> · 36, 131
fifo file of <folder> · 22, 36, 131
file · 1, 2, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
    16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27,
    28, 29, 30, 31, 32, 33, 34, 36, 45, 57, 58,
    77, 119, 122, 124, 125, 126, 127, 131, 133,
    141, 143, 145, 153, 157, 162, 163, 164,
    175, 176, 179, 184
file <string> · 15, 16, 22, 131
file <string> of <folder> · 16, 22, 131
file <symlink> · 16, 131
file content · 15, 16, 24, 25, 122, 179
file count of <filesystem> · 10, 131
file line · 17, 29, 30, 143, 153, 157
file of <folder> · 16, 22, 131
file section · 17, 23, 25, 141, 162
filesystem · 8, 9, 10, 11, 12, 13, 14, 15, 16,
    17, 20, 22, 27, 30, 31, 32, 34, 35, 36, 37,
    95, 114, 115, 125, 126, 127, 130, 131, 133,
    134, 135, 143, 144, 148, 152, 154, 155,
    156, 162, 163, 164, 165, 167, 170, 171,
    175, 179, 180
filesystem <string> · 9, 131
filesystem object · 8, 9, 10, 11, 12, 13, 14, 15,
    20, 27, 30, 31, 34, 35, 36, 37, 95, 114, 115,
    125, 130, 134, 135, 143, 144, 148, 152,
    154, 155, 156, 162, 163, 165, 167, 171,
    175, 179, 180
Filesystem Objects · 8
filesystem of <device file> · 9, 35, 131
filesystem of <fifo file> · 10, 36, 131
filesystem of <file> · 10, 17, 131

filesystem of <folder> · 10, 22, 131
filesystem of <socket file> · 10, 37, 131
filesystem of <symlink> · 10, 32, 131
find adapter <string> of <network> · 82, 86, 131
find file <string> of <folder> · 16, 22, 131
find folder <string> of <folder> · 20, 22, 131
fips mode of <license> · 72, 132
first start time of <application usage summary instance> · 70, 132
first start time of <application usage summary> · 69, 132
fixlet · 56, 57, 60, 61, 123, 132, 136, 138, 152,
    154, 160, 164, 175
Fixlet Objects · 60
fixlet of <site> · 56, 60, 132
fixlet_header · 60, 136, 152, 175
flag list of <processor> · 41, 133
folder · 6, 9, 10, 12, 13, 15, 16, 18, 19, 20, 21,
    22, 23, 25, 30, 31, 32, 33, 34, 35, 36, 37,
    56, 57, 58, 115, 116, 120, 125, 127, 130,
    131, 133, 155, 160, 165, 168, 184
folder <string> · 21, 22, 133
folder <string> of <folder> · 21, 22, 133
folder <symlink> · 21, 133
folder of <folder> · 21, 23, 133
fpu exception of <processor> · 41, 133
fpu of <processor> · 42, 133
free amount of <ram> · 44, 133
free amount of <swap> · 47, 133
free file count of <filesystem> · 10, 133
free percent of <filesystem> · 10, 133
free space of <filesystem> · 10, 133
friendly name of <network adapter> · 87, 133
full gateway address of <selected server> · 66, 91, 133
full of <power level> · 112, 134

G

gateway address <integer> of <selected server> · 66, 91, 134
gateway address of <selected server> · 66, 92, 134
gather schedule authority of <site> · 56, 134
gather schedule time interval of <site> · 57, 134
gather url of <license> · 72, 134
gid of <filesystem object> · 12, 134
gid of <symlink> · 32, 134
group <integer> of <site> · 57, 59, 134

group execute of <filesystem object> · 12, 134
group leader of <action> · 80, 134
group mask of <filesystem object> · 12, 134
group mask of <mode> · 28, 134
group name of <filesystem object> · 12, 135
group name of <symlink> · 32, 135
group read of <filesystem object> · 12, 135
group write of <filesystem object> · 12, 135

H

header <string> of <fixlet> · 60, 61, 136
header of <fixlet> · 60, 61, 136
hlt bug of <processor> · 42, 136
host name · 6, 7, 67, 136
host name of <root server> · 67, 136
hostname · 6, 136

I

id of <action> · 80, 138
id of <fixlet> · 60, 138
id of <process> · 46, 138
id of <root server> · 67, 138
id of <site group> · 59, 138
index of <processor> · 42, 138
installed <string> of <rpmdatabase> · 93, 138
installed file of <package> · 96, 98, 138
Installed System Software · 93
instance of <application usage summary> · 69, 70, 138
integer <string> of <smbios structure> · 52, 139
integer value <string> of <smbios structure> · 52, 54, 139
interface <integer> of <network> · 82, 83, 139
interface of <network> · 82, 83, 139
ip address of <selected server> · 66, 92, 139
ip interface <integer> of <network> · 82, 84, 139
ip interface of <network> · 82, 84, 140
ipv4 interface of <network adapter> · 87, 88, 140
ipv4 interface of <network> · 82, 88, 140
ipv4or6 interface of <network adapter> · 87, 88, 140
ipv4or6 interface of <network> · 83, 88, 140
ipv6 interface of <network adapter> · 87, 88, 140
ipv6 interface of <network> · 83, 88, 140

K

key <string> of <file section> · 24, 141
key <string> of <file> · 17, 141
Key Phrases (Inspectors) · 114
keywords · 1, 2, 4, 8, 38, 56, 79, 82, 114, 186

L

language · 1, 4, 5, 7, 41, 48, 49, 156, 157, 168, 182, 186, 189, 196
last change time of <action> · 80, 142
last gather time of <site> · 57, 142
last relay select time · 6, 142
last start time of <application usage summary instance> · 70, 142
last start time of <application usage summary> · 69, 142
last time seen of <application usage summary instance> · 70, 142
last time seen of <application usage summary> · 69, 142
length of <smbios structure> · 52, 143
license · 71, 72, 73, 74, 115, 117, 120, 121, 127, 128, 132, 134, 146, 154, 158, 159, 162, 164, 165, 171, 195
License Objects · 71
line <integer> of <file> · 17, 29, 143
line containing <string> of <file> · 17, 29, 143
line number of <file line> · 30, 143
line of <file> · 17, 29, 143
line starting with <string> of <file> · 17, 29, 143
link count of <filesystem object> · 12, 143
link count of <symlink> · 32, 143
location of <filesystem object> · 12, 144
location of <symlink> · 32, 144
lock string of <action lock state> · 144
locked of <action lock state> · 144
long form of <short rpm package version record> · 105, 110, 144
loopback of <network adapter interface> · 89, 144
loopback of <network adapter> · 87, 144
loopback of <network ip interface> · 85, 144
low of <power level> · 112, 144

M

mac address of <network adapter interface> · 89, 144

O

obsolete of <package> · 96, 98, 153
offer accepted of <action> · 80, 154
offer of <action> · 80, 154
offset of <smbios value> · 55, 154
operating system · 1, 2, 3, 6, 15, 19, 38, 39, 40, 49, 116, 118, 119, 121, 144, 152, 154, 159, 174, 177, 182
organization of <license> · 73, 154
origin fixlet id of <action> · 80, 154
other execute of <filesystem object> · 13, 154
other mask of <filesystem object> · 13, 154
other mask of <mode> · 28, 154
other read of <filesystem object> · 13, 155
other write of <filesystem object> · 13, 155

P

package · 25, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 116, 122, 129, 138, 152, 153, 155, 158, 160, 161, 172, 176, 182
package <string> of <rpmdatabase> · 93, 95, 155
package conflicting with <capability> of <rpmdatabase> · 94, 95, 155
package installing <capability> of <rpmdatabase> · 94, 95, 155
package of <rpmdatabase> · 94, 95, 155
package providing <capability> of <rpmdatabase> · 94, 95, 155
package requiring <capability> of <rpmdatabase> · 94, 95, 155
parameter <string> · 6, 80, 155
parameter <string> of <action> · 6, 80, 155
parent folder of <filesystem object> · 13, 21, 155
parent folder of <symlink> · 21, 32, 155
pathname of <filesystem object> · 13, 156
pathname of <symlink> · 33, 156
pending login · 6, 80, 156
pending login of <action> · 80, 156
pending of <action> · 81, 156
pending restart · 6, 7, 81, 156
pending restart <string> · 6, 156
pending restart name · 7, 156
pending restart of <action> · 81, 156
pending time of <action> · 81, 156
pid of <process> · 46, 156
platform id of <language> · 48, 156
plugged of <power level> · 112, 156

point to point of <network adapter interface> · 89, 157
point to point of <network ip interface> · 85, 157
port number of <selected server> · 66, 157
Power Inspectors · 112
power level · 112, 113, 134, 144, 153, 156, 157, 174, 182
previous line of <file line> · 30, 157
previous value of <runlevel> · 50, 157
primary language · 48, 49, 157, 182
primary language of <language> · 48, 49, 157
priority of <selected server> · 66, 157
process · 6, 23, 46, 47, 75, 121, 128, 138, 152, 156, 157, 158
process <integer> · 46, 157
process <string> · 46, 157
process id of <process> · 47, 158
processor · 40, 41, 42, 43, 51, 52, 118, 121, 122, 130, 133, 136, 138, 145, 148, 158, 162, 165, 176, 177, 187, 188
processor <integer> · 40, 158
product of <license> · 73, 74, 158
provide of <package> · 96, 98, 158

R

ram · 43, 44, 117, 119, 133, 158, 163, 164, 169, 172, 174
random access memory · 43, 44, 158
read of <mode_mask> · 29, 159
recent application · 19, 159
recent application <string> · 19, 159
registrar number of <license> · 73, 159
registration address of <client> · 63, 92, 159
registration cidr address of <client> · 63, 159
registration mac address of <client> · 63, 159
registration server · 5
registration subnet address of <client> · 63, 92, 159
relation of <capability> · 98, 159
relay service · 45, 159
release of <operating system> · 39, 159
release of <rpm package version record> · 100, 107, 160
release of <short rpm package version record> · 101, 110, 160
Relevance Language · 2
relevance of <fixlet> · 61, 160
relevant fixlet of <site> · 57, 60, 160
relevant offer action of <site> · 57, 79, 160
require of <package> · 96, 98, 160

T

U

V

W

Y