Tivoli. *Endpoint Manager*

Version 8.1

Macintosh 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 Macintosh 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 Mac OS. Thousands of Inspectors have been created to expose the inner workings of Macintosh 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 Macintosh-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 Macintosh Operating System. Contact your Tivoli Endpoint Manager marketing representative for information about Inspector Guides for other operating systems, including Windows, Solaris, HPUX, AIX, and a variety of Linux operating systems.

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

Primitive Objects

The relevance language is based upon a comprehensive set of primitive objects. These primitives are the basic building blocks of the more complex objects to follow. The Core Inspector Guide documents the bulk of the primitive object inspectors. Where a specialized platform-specific method exists to create, inspect, or manipulate primitive objects, they will be documented in the respective Inspector guide.

Boolean

Creation Methods

These boolean creation methods are in addition to the other properties that return the boolean type.

Key Phrase	*Form*	Description
boolean <integer> of <array>	*Numbered*	Get, from an array, a boolean keyed by the specified integer. Mac
powerpc	*PlainGlobal*	Returns TRUE if the cpu is a PowerPC, FALSE if it is a 68000 chip. Mac

Integer

Integers are represented internally as 64-bit signed values.

Creation Methods

These integer creation methods are in addition to the other properties that return the integer type.

Key Phrase	*Form*	Description
cpu speed	*PlainGlobal*	Returns the clock speed of the cpu in Hz. Mac
gestalt <string>	*NamedGlobal*	This reads a 32 bit integer from the MacOS. The selector name is the four character OSType that chooses which item is being inspected. The interpretation of the result depends on the selector. It might represent an integer or a version, for instance. Mac

Key Phrase	*Form*	Description
keyboard type	*PlainGlobal*	Returns the keyboard type. Mac
machine type	*PlainGlobal*	Returns the machine type. The value is from a long enumeration of all Mac platforms. See the header file Gestalt.h. Mac
nubus map	*PlainGlobal*	Returns the nubus map. Mac
physical ram	*PlainGlobal*	Returns the amount of physical ram in the computer. Mac

String

String are typically core objects, but some string Inspectors may be client-specific.

- Note: A string literal is written within double quotes. Special characters must be inserted by using the percent sign followed by 2 hex digits. Special characters include those characters with ASCII codes less than the 'space' character (hex 20) or greater than 'tilde' character (hex 7f) as well as the percent character itself (25 hex). For example, to create a string containing a null character and a percent character use "a null is %00, the percent itself is %25". Conversion to upper and lower case is also provided. String works in combination with the string position and substring data types. A string position is a point within a string. It can be compared to an integer, but it also acts as a pointer within a string so that the preceding and following text can be extracted. A substring is a part of a larger string. All operations allowed on a string can be performed on a substring. There are two substrings "be" in the string "To be or not to be". The substrings only differ in their positions within the string.

Creation Methods

These string creation methods are in addition to the other properties that return the string type.

Key Phrase	*Form*	Description
machine name	*PlainGlobal*	Returns the machine name. The value is from a long enumeration of all Mac platforms. See the header file Gestalt.h or Apple's. Mac

NOTE: Many Inspectors return string values from the operating system using a variety of APIs. For the most part, these strings are encoded as single-byte character sets (SBCSs) or multi-byte character sets (MBCSs) depending on the active code page. You can use the code page Inspectors to determine which page is currently active on the client.

Array

These Inspectors return a list of values in a dictionary array.

Creation Methods

Key Phrase	*Form*	Description
array <integer> of <array>	*Numbered*	An array from an array by index. Mac

Properties

Key Phrase	*Form*	Return Type	Description
array <integer> of <array>	*Numbered*	<array>	An array from an array by index. Mac
boolean <integer> of <array>	*Numbered*	<boolean>	Get, from an array, a boolean keyed by the specified integer. Mac
date <integer> of <array>	*Numbered*	<time>	Get, from an array, a date keyed by the specified integer. Mac
dictionary <integer> of <array>	*Numbered*	<dictionary>	Get, from an array, a dictionary keyed by the specified integer. Mac
integer <integer> of <array>	*Numbered*	<integer>	Get, from an array, an integer keyed by the specified integer. Mac
size of <array>	*Plain*	<integer>	Returns the number of unique elements in the given array. Mac
string <integer> of <array>	*Numbered*	<string>	Get, from an array, a string keyed by the specified integer. Mac
value of <array>	*Plain*	<osxvalue>	Values of the array. Mac

Examples

- `strings of values of array 0 of array "com.apple.iTunes" of dictionary of file "com.apple.help.plist" of preferences folder`

- Returns any values of type string from the array, for example iTunes Help or file://localhost/Applications/iTunes.app/Contents/Resources/English.lproj/iTunes.Help/.

- `boolean 1 of array "NSTableView Sort Ordering NSNavOutlineColumnSettings.v1" of preference "com.apple.Console"`
- Returns the first boolean value in the array.

- `size of array "persistent-apps" of preference "com.apple.dock"`
- Returns the number of unique elements in the specified array.

- `string 0 of array "RecentSearchStrings" of preference "com.apple.safari"`
- Returns the most recent search string.

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
cpu speed	*PlainGlobal*	<integer>	Returns the clock speed of the cpu in Hz. Mac

Key Phrase	*Form*	Return Type	Description
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
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
gestalt <string>	*NamedGlobal*	<integer>	This reads a 32 bit integer from the MacOS. The selector name is the four character OSType that chooses which item is being inspected. The interpretation of the result depends on the selector. It might represent an integer or a version, for instance. Mac
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
logical ram	*PlainGlobal*	<integer>	Returns the amount of logical ram in the computer, physical plus virtual. Mac
machine name	*PlainGlobal*	<string>	Returns the machine name. The value is from a long enumeration of all Mac platforms. See the header file Gestalt.h or Apple's web site. Mac
nubus map	*PlainGlobal*	<integer>	Returns the nubus map. Mac

Key Phrase	*Form*	Return Type	Description
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
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
physical ram	*PlainGlobal*	<integer>	Returns the amount of physical ram in the computer. Mac
powerpc	*PlainGlobal*	<boolean>	Returns TRUE if the cpu is a PowerPC, FALSE if it is a 68000 chip. Mac
virtual memory	*PlainGlobal*	<boolean>	Returns TRUE if virtual memory is turned on. Mac

Key Phrase	*Form*	Return Type	Description
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

- `cpu speed > 2000000000`

Returns TRUE if the CPU speed is greater than two gigahertz.

- `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.

- `gestalt "ram" > 33554432`

Returns TRUE if the installed RAM is greater than the amount specified.

- `physical ram > 33554432`

Returns TRUE if there is more than the specified amount of RAM.

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 Object

These Inspectors provide handles for the various objects available in the file system.

Creation Methods

Key Phrase	*Form*	Description
application	*PlainGlobal*	This Mac Inspector recursively finds all applications (bundles) and executables under the /Applications directory, as well as all of those under the Applications directories of all local users of the machine. Mac
application <string>	*NamedGlobal*	Returns the named applications (bundles) and executables as filesystem objects. Mac
application of <folder>	*Plain*	Recursively finds executables in the supplied folder, returning a list of "application" objects, which are basically filesystem_objects. Mac:7.1
find item <string> of <folder>	*Named*	Returns the filesystem objects matching the item filter string of the given folder. Macintosh 'items' are analogous to files and folders on other systems. Mac:8.0
hfs item <string>	*NamedGlobal*	Returns a filesystem object with the specified item name. The Macintosh has two different path types it can output. HFS (Hierarchical File System) was an OS9 format where the path separator was ':' and there were multiple roots (each volume). POSIX has superseded HFS and is single rooted with '/' as a separator. All of the Macintosh Inspectors that are not otherwise specified default to the POSIX version. Mac:8.0

Key Phrase	*Form*	Description
hfs relative item <string> of <folder>	*Named*	Returns a filesystem object with the given wildcard name from the specified folder, using HFS format. 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. Mac:8.0
item <string>	*NamedGlobal*	Returns a filesystem object with the given name, either a file or a folder. Mac:8.0
item <string> of <folder>	*Named*	Returns the named item (file or folder) from the specified folder. Mac
item ending in <string> of <folder>	*Named*	Returns a list of items (files/folders) ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for "*xxxx". Mac
item of <folder>	*Plain*	Returns a list of the items (file or folder) in the specified folder. Mac
posix item <string>	*NamedGlobal*	Returns a filesystem object with the given name, either a file or a folder, using POSIX formatting. Mac:8.0
posix relative item <string> of <folder>	*Named*	Returns a filesystem object with the given relative item name from the specified folder, using POSIX format. Mac:8.0
relative item <string> of <folder>	*Named*	Returns a filesystem object with the given relative item name from the specified folder, using POSIX format. Mac:8.0
sibling item <string> of <filesystem object>	*Named*	The named sibling of a filesystem object (file, folder). Mac

Properties

Key Phrase	*Form*	Return Type	Description
<filesystem object> as file	*Cast*	<file>	Returns a file or nothing (if, for example, the filesystem object was a folder). Mac

Key Phrase	*Form*	Return Type	Description
<filesystem object> as folder	*Cast*	<folder>	Returns a folder or nothing. Mac
<filesystem object> as string	*Cast*	<string>	Casts a filesystem object as a string. Win:8.0, Lin, Sol, HPUX, AIX, Mac, 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
backup time of <filesystem object>	*Plain*	<time>	The date of the last backup of the specified filesystem object (such as a file or folder). This corresponds to what is shown in the "Get Info" box for this object. Mac
bundle version of <filesystem object>	*Plain*	<version>	Returns the version of the filesystem object corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac
creation time of <filesystem object>	*Plain*	<time>	The date and time of creation of the specified file or folder. This corresponds to what is shown in the "Get Info" box. Win, Mac, WM
hfs path of <filesystem object>	*Plain*	<string>	Returns the path to a filesystem object in HFS terms (colons as delimiters). Mac
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
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
posix path of <filesystem object>	*Plain*	<string>	The POSIX file path for the file or folder. Mac

Key Phrase	*Form*	Return Type	Description
sibling file <string> of <filesystem object>	*Named*	<file>	A named property. A file in the same folder as the specified file object. Mac
sibling folder <string> of <filesystem object>	*Named*	<folder>	A named property. A folder in the same folder as the specified file object. Mac
sibling item <string> of <filesystem object>	*Named*	<filesystem object>	The named sibling of a filesystem object (file, folder). Mac
version of <filesystem object>	*Plain*	<version>	This returns the version information from "vers" resource 1 of the file. It is usually present in applications, and may exist in data files as well. It corresponds to what appears in the "Get Info" box for the specified filesystem object. Mac

Operators

Key phrase	Return Type	Description
<filesystem object> = <filesystem object>	*<boolean>*	Compares two Macintosh file system objects. Mac

Examples

- `applications of folder "/Users/fizzle/bin"`

  Returns a list of applications in the specified directory, such as /Users/fizzle/bin/p4, /Users/fizzle/bin/mylocaltool.

- `names of items of applications folder`

  Returns a list of applications, such as DS_Store, .localized, AddressBook.app, AppleScript, Calculator.app, Chess.app, DVD Player.app, etc….

- `application "iTunes.app" as file`

  Fails because that is in fact a folder.

- `creation time of file "System" of System Folder > time "3 jan 1998 00:00+0000"`

  Returns TRUE if the creation time of the system file is newer than the specified date.

- `name of object "iChat.app" of applications folder`

  Returns iChat.app.

- `posix paths of items whose (name of it starts with "i") of applications folder`

  Returns a list of the paths of applications starting with "i", such as /Applications/iCal.app, /Applications/iChat.app or /Applications/iTunes.app.

- `exists sibling file "iTunes.app" of application "iChat.app"`

  Returns TRUE if both applications are in the same folder.

- `version of primary application "ttxt" is greater than "1.3"`

  Returns TRUE if the default application for "ttxt" is more recent than version 1.3.

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
<filesystem object> as file	*Cast*	Returns a file or nothing (if, for example, the filesystem object was a folder). Mac
descendant of <folder>	*Plain*	Returns a list of all the descendant files of the specified folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
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
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 ending in <string> of <folder>	*Named*	Returns a list of files ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for "*xxxx". Mac
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
hfs file <string>	*NamedGlobal*	Returns the hierachical file system (HFS) file specified by <string>. Mac
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
posix file <string>	*NamedGlobal*	Returns the POSIX file specified by <string>. Mac
relative file <string> of <folder>	*Named*	Returns the file with the path specified by <string> relative to the given <folder>. Mac
relative hfs file <string> of <folder>	*Named*	Returns the HFS file with the path specified by <string> relative to the given <folder>. Mac

Key Phrase	*Form*	Description
relative posix file <string> of <folder>	*Named*	Returns the POSIX file with the path specified by <string> relative to the given <folder>. Mac
sibling file <string> of <filesystem object>	*Named*	A named property. A file in the same folder as the specified file object. Mac

Properties

Key Phrase	*Form*	Return Type	Description
alias of <file>	*Plain*	<boolean>	Returns TRUE if the file is an alias for another file. Mac
array of <file>	*Plain*	<array>	This Inspector creates an OSXArray from the specified XML Data file. Mac:8.1
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
data fork of <file>	*Plain*	<datafork>	Returns information about the data fork of the specified file. Mac
dictionary of <file>	*Plain*	<dictionary>	Returns the dictionary object for the specified file. Mac
drive of <file>	*Plain*	<volume>	Returns the drive (as a <volume>) associated with the specified file. This is a Macintosh-only Inspector. Mac

Key Phrase	*Form*	Return Type	Description
filesystem of <file>	*Plain*	<volume>	Returns the volume corresponding to the filesystem of the specified file. Mac
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
length of <file>	*Plain*	<integer>	The total length of the data and resource forks. Mac
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
locked of <file>	*Plain*	<boolean>	Returns TRUE if the file is locked. Mac
resource fork of <file>	*Plain*	<resfork>	Returns information about the resource fork of the file. Mac
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

Key Phrase	*Form*	Return Type	Description
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
version <integer> of <file>	*Numbered*	<version>	Returns the nth version information from the "vers" resource of the given file. Typically n=1, but other information may be stored in "vers" resources greater than 1. Mac
visible of <file>	*Plain*	<boolean>	Returns TRUE if the file is visible. Mac
volume of <file>	*Plain*	<volume>	The volume containing the file. Mac

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.

- `files ending in ".a" of folder "/usr/lib"`

Returns a list of files with the specified ending, such as: /usr/lib/libcpp_kext.a, /usr/lib/libfl.a, /usr/lib/libioabc.a, and others.

- `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.

- `relative file "Safari.app/Contents/MacOS/Safari" of applications folder`

Returns the concatenation of the specified folder and the given path, /Applications/Safari.app/Contents/MacOS/Safari.

- `filesystem of folder "/Users/MyUserName/Library"`

Returns /Users/MyUserName.

- `Length of data fork of file "Microsoft Word" of folder "Microsoft Office X" of Applications folder is greater than 100000`

Returns TRUE if the data fork of the system file is greater than the specified amount.

- `locked of file "this file" of folder "this folder"`

Returns TRUE if the specified file is locked.

- `version 0 of file "fname" = "1.0"`

Checks for the zero version, if one exists.

- `volume of (application "iTunes.app" as folder) is volume of system folder`

Returns TRUE if iTunes is installed on the same volume as the system folder.

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
<filesystem object> as folder	*Cast*	Returns a folder or nothing. Mac
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
apple extras folder of <domain>	*Plain*	Returns the apple extras folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
apple menu items folder of <domain>	*Plain*	Returns the apple menu items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
application support folder of <domain>	*Plain*	Returns the application support folder of the specified OS X domain, typically /Library/Application Support. If the domain is not specified, it defaults to the system domain. Mac
applications folder of <domain>	*Plain*	Returns the applications folder of the specified OS X domain, typically /Applications. If the domain is not specified, it defaults to the system domain. Mac
assistants folder of <domain>	*Plain*	Returns the assistants folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
audio folder of <domain>	*Plain*	Returns the audio folder of the specified OS X domain, typically /Library/Audio. If the domain is not specified, it defaults to the system domain. Mac
cache folder of <domain>	*Plain*	Returns the cache folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
carbon folder of <domain>	*Plain*	Returns the carbon folder of the specified OS X domain, typically /Library/Carbon. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Description
chewable items folder of <domain>	*Plain*	Returns the chewable items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
classic folder of <domain>	*Plain*	Returns the classic folder of the specified OS X domain, typically the /System Folder. If the domain is not specified, it defaults to the system domain. Mac
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
color sync folder of <domain>	*Plain*	Returns the color sync folder of the specified OS X domain, typically /System/Library/ColorSync. If the domain is not specified, it defaults to the system domain. Mac
colorsync profiles folder of <domain>	*Plain*	Returns the colorsync profiles folder of the specified OS X domain, typically /System/Library/ColorSync/Profiles. If the domain is not specified, it defaults to the system domain. Mac
component folder of <domain>	*Plain*	Returns the component folder of the specified OS X domain, typically /System/Library/Components. If the domain is not specified, it defaults to the system domain. Mac
contextual menu items folder of <domain>	*Plain*	Returns the contextual menu items folder of the specified OS X domain, typically /Library/Contextual Menu Items. If the domain is not specified, it defaults to the system domain. Mac
control panels folder of <domain>	*Plain*	Returns the control panels folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
control strip modules folder of <domain>	*Plain*	Returns the control strip modules folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
core services folder of <domain>	*Plain*	Returns the core services folder of the specified OS X domain, typically /System/Library/CoreServices. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Description
current user folder of <domain>	*Plain*	Returns the current user folder of the specified OS X domain, typically found at /Users/username. If the domain is not specified, it defaults to the system domain. Mac
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
desktop folder of <domain>	*Plain*	Returns the desktop folder of the specified OS X domain, typically /Users/Username/Desktop. If the domain is not specified, it defaults to the system domain. Mac
developer docs folder of <domain>	*Plain*	Returns the developer docs folder of the specified OS X domain, typically found at /Developer/Documentation. If the domain is not specified, it defaults to the system domain. Mac
developer folder of <domain>	*Plain*	Returns the developer folder of the specified OS X domain, typically found at /Developer. If the domain is not specified, it defaults to the system domain. If the domain is not specified, it defaults to the system domain. Mac
developer help folder of <domain>	*Plain*	Returns the help folder of the specified OS X domain, typically /Developer/Documentation/Help. If the domain is not specified, it defaults to the system domain. Mac
disabled control panels folder of <domain>	*Plain*	Returns the control panels folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
disabled extensions folder of <domain>	*Plain*	Returns the extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
disabled shutdown items folder of <domain>	*Plain*	Returns the shutdown folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Description
disabled startup items folder of <domain>	*Plain*	Returns the startup items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
disabled system extensions folder of <domain>	*Plain*	Returns the systems extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
documentation folder of <domain>	*Plain*	Returns the documentation folder for the given OS X domain, typically found at /Library/Documentation. If the domain is not specified, it defaults to the system domain. Mac
documents folder of <domain>	*Plain*	Returns the documents folder for the specified OS X domain, typically found at /User/Username/Documents. If the domain is not specified, it defaults to the system domain. Mac
domain library folder of <domain>	*Plain*	Returns the domain library folder of the specified OS X domain, typically found at /Library. If the domain is not specified, it defaults to the system domain. Mac
domain top folder of <domain>	*Plain*	Returns the top folder of the specified OS X domain, typically found at /System. If the domain is not specified, it defaults to the system domain. Mac
extensions folder of <domain>	*Plain*	Returns the extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
favorites folder of <domain>	*Plain*	Returns the favorites folder of the specified OS X domain, typically /Users/username/Library/Favorites. If the domain is not specified, it defaults to the system domain. Mac
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, Mac: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 ending in <string> of <folder>	*Named*	Returns a list of folders ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for "*xxxx". Mac
folder of <folder>	*Plain*	Iterates through the sub-folders of the folder object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
fonts folder of <domain>	*Plain*	Returns the font folder of the specified OS X domain, typically /System/Library/Fonts. If the domain is not specified, it defaults to the system domain. Mac
framework <string> of <domain>	*Named*	Returns a folder of the form "/<domain>/Library/Frameworks/<string>.framework". • Note: This Inspector appends .framework for you, so don't provide it. The framework inspector needs a domain, and without it defaults to the system domain. Mac
framework folder of <domain>	*Plain*	Returns the framework folder of the specified OS X domain, typically /System/Library/Frameworks. If the domain is not specified, it defaults to the system domain. Mac
help folder of <domain>	*Plain*	Returns the help folder of the specified OS X domain, typically /Library/Documentation/Help. If the domain is not specified, it defaults to the system domain. Mac
hfs folder <string>	*NamedGlobal*	Returns the hierachical file system (HFS) folder specified by <string>. Mac

Key Phrase	*Form*	Description
home directory of <user>	*Plain*	On a Mac, this Inspector returns the dsAttrTypeStandard:NFSHomeDirectory attribute of the specified user as a folder. • Note: On a Windows system, this inspector has a different interpretation: it returns the directory (as a string) where the user files are stored for the specified user. Mac:7.1
internet plugins folder	*PlainGlobal*	Returns the folder object corresponding to the internet plugins. Mac
internet plugins folder of <domain>	*Plain*	Returns the internet plugins folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
iss download folder	*PlainGlobal*	The Macintosh download folder. This Inspector was previously labeled "download," which now refers to a world object that performs dynamic downloading. Mac:7.2
iss download folder of <domain>	*Plain*	Returns the Macintosh download folder for the specified domain. Mac:7.2
kernel extensions folder of <domain>	*Plain*	Returns the kernel extensions folder of the specified OS X domain, typically /System/Library/Extensions. If the domain is not specified, it defaults to the system domain. Mac
locales folder of <domain>	*Plain*	Returns the locales folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
location manager modules folder of <domain>	*Plain*	Returns the location manager modules folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
location manager preferences folder of <domain>	*Plain*	Returns the location manager preferences folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
locations folder of <domain>	*Plain*	Returns the locations folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Description
macos read me folder of <domain>	*Plain*	Returns the Mac OS read me folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
modem scripts folder of <domain>	*Plain*	Returns the modem scripts folder of the specified OS X domain, typically /System/Library/Modem Scripts. If the domain is not specified, it defaults to the system domain. Mac
parent folder of <filesystem object>	*Plain*	The folder containing the specified file or folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
posix folder <string>	*NamedGlobal*	Returns the POSIX folder specified by <string>. Mac
preferences folder of <domain>	*Plain*	Returns the preferences folder of the specified OS X domain, typically /Users/username/Library/Preferences. If the domain is not specified, it defaults to the system domain. Mac
printer descriptions folder of <domain>	*Plain*	Returns the printer descriptions folder of the specified OS X domain, typically /System/Library/Printers/PPDs. If the domain is not specified, it defaults to the system domain. Mac
printer drivers folder of <domain>	*Plain*	Returns the printer drivers folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
printers folder of <domain>	*Plain*	Returns the printers folder of the specified OS X domain, typically /System/Library/Printers. If the domain is not specified, it defaults to the system domain. Mac
printmonitor documents folder of <domain>	*Plain*	Returns the printmonitor documents folder of the specified OS X domain, typically /Library/Printers/PrintMonitor Documents. If the domain is not specified, it defaults to the system domain. Mac
private framework folder of <domain>	*Plain*	Returns the private framework folder of the specified OS X domain, typically /System/Library/PrivateFrameworks. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Description
quicktime folder of <domain>	*Plain*	Returns the quicktime folder of the specified OS X domain, typically /System/Library/QuickTime. If the domain is not specified, it defaults to the system domain. Mac
receipts folder of <domain>	*Plain*	Returns the receipts folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
relative folder <string> of <folder>	*Named*	Returns the folder with the path specified by <string> relative to the given <folder>. Mac
relative hfs folder <string> of <folder>	*Named*	Returns the HFS folder with the path specified by <string> relative to the given <folder>. Mac
relative posix folder <string> of <folder>	*Named*	Returns the POSIX folder with the path specified by <string> relative to the given <folder>. Mac
scripting additions folder of <domain>	*Plain*	Returns the scripting additions folder of the specified OS X domain, typically /System/Library/Scripting Additions. If the domain is not specified, it defaults to the system domain. Mac
shared folder of <domain>	*Plain*	Returns the shared folder of the specified OS X domain, typically /Users/Shared. If the domain is not specified, it defaults to the system domain. Mac
shared libraries folder of <domain>	*Plain*	Returns the shared libraries folder of the specified OS X domain, typically /System/Library/CFMSupport. If the domain is not specified, it defaults to the system domain. Mac
shutdown items folder of <domain>	*Plain*	Returns the shutdown items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
sibling folder <string> of <filesystem object>	*Named*	A named property. A folder in the same folder as the specified file object. Mac
sound folder of <domain>	*Plain*	Returns the sound folder of the specified OS X domain, typically /System/Library/Sound. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Description
speech folder of <domain>	*Plain*	Returns the speech folder of the specified OS X domain, typically /System/Library/Speech. If the domain is not specified, it defaults to the system domain. Mac
startup items folder of <domain>	*Plain*	Returns the startup items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
system folder of <domain>	*Plain*	Returns the system folder of the specified OS X domain, typically /System. If the domain is not specified, it defaults to the system domain. Mac
temporary items folder of <domain>	*Plain*	Returns the temporary items folder of the specified OS X domain, typically /private/tmp/. If the domain is not specified, it defaults to the system domain. Mac
text encodings folder of <domain>	*Plain*	Returns the text encodings folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
themes folder of <domain>	*Plain*	Returns the themes folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
user temp folder of <domain>	*Plain*	Returns the user temp folder of the specified OS X domain, typically /private/tmp/uid where uid is the user ID number. If the domain is not specified, it defaults to the system domain. Mac
users folder of <domain>	*Plain*	Returns the users folder of the specified OS X domain, typically /Users. If the domain is not specified, it defaults to the system domain. Mac
utilities folder of <domain>	*Plain*	Returns the utilities folder of the specified OS X domain, typically /Applications/Utilities. If the domain is not specified, it defaults to the system domain. Mac
voices folder of <domain>	*Plain*	Returns the voices folder of the specified OS X domain, typically /System/Library/Speech/Voices. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Description
volume settings folder of <domain>	*Plain*	Returns the volume settings folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac

Properties

Key Phrase	*Form*	Return Type	Description
application of <folder>	*Plain*	<filesystem object>	Recursively finds executables in the supplied folder, returning a list of "application" objects, which are basically filesystem_objects. Mac:7.1
bundle of <folder>	*Plain*	<bundle>	Returns a bundle from a folder (if it has one like application folders do). Mac
bundle version of <folder>	*Plain*	<version>	Returns the version of the folder corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac
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
drive of <folder>	*Plain*	<volume>	Returns the drive associated with the specified folder. Mac
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 ending in <string> of <folder>	*Named*	<file>	Returns a list of files ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for "*xxxx". Mac
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

Key Phrase	*Form*	Return Type	Description
filesystem of <folder>	*Plain*	<volume>	Returns the volume corresponding to the filesystem of the specified folder. Mac
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
find item <string> of <folder>	*Named*	<filesystem object>	Returns the filesystem objects matching the item wildcard string of the given folder. Macintosh 'items' are analogous to files and folders on other systems. 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. Mac:8.0
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
folder ending in <string> of <folder>	*Named*	<folder>	Returns a list of folders ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for "*xxxx". Mac
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

Key Phrase	*Form*	Return Type	Description
hfs relative item <string> of <folder>	*Named*	<filesystem object>	Returns a filesystem object with the given wildcard name from the specified folder, using HFS format. 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. Mac:8.0
item <string> of <folder>	*Named*	<filesystem object>	Returns the named item (file or folder) from the specified folder. Mac
item ending in <string> of <folder>	*Named*	<filesystem object>	Returns a list of items (files/folders) ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for "*xxxx". Mac
item of <folder>	*Plain*	<filesystem object>	Returns a list of the items (file or folder) in the specified folder. Mac
posix relative item <string> of <folder>	*Named*	<filesystem object>	Returns a filesystem object with the given relative item name from the specified folder, using POSIX format. Mac:8.0
relative file <string> of <folder>	*Named*	<file>	Returns the file with the path specified by <string> relative to the given <folder>. Mac
relative folder <string> of <folder>	*Named*	<folder>	Returns the folder with the path specified by <string> relative to the given <folder>. Mac
relative hfs file <string> of <folder>	*Named*	<file>	Returns the HFS file with the path specified by <string> relative to the given <folder>. Mac
relative hfs folder <string> of <folder>	*Named*	<folder>	Returns the HFS folder with the path specified by <string> relative to the given <folder>. Mac
relative item <string> of <folder>	*Named*	<filesystem object>	Returns a filesystem object with the given relative item name from the specified folder, using POSIX format. Mac:8.0

Key Phrase	*Form*	Return Type	Description
relative posix file <string> of <folder>	*Named*	<file>	Returns the POSIX file with the path specified by <string> relative to the given <folder>. Mac
relative posix folder <string> of <folder>	*Named*	<folder>	Returns the POSIX folder with the path specified by <string> relative to the given <folder>. Mac
version of <folder>	*Plain*	<version>	Returns a version for a folder representing an application. Mac
volume of <folder>	*Plain*	<volume>	The volume containing the specified folder. Mac

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.

Examples

- `color sync folder of system domain`
  - Returns /System/Library/ColorSync.

- `color sync folder of local domain`
  - Returns /Library/ColorSync.

- `application support folder of user domain`
  - Returns /var/root/Library/Application Support.

- `framework "MyPrivate" of local domain`
  - Returns /Library/Frameworks/MyPrivate.framework.

- `parent folder of application "Terminal.app" is Utilities folder`
  - TRUE if the specified application is in the Utilities folder.

- `applications of folder "/Users/fizzle/bin"`
  - Returns a list of applications in the specified directory, such as /Users/fizzle/bin/p4, /Users/fizzle/bin/mylocaltool.

- `exists (bundle of applications folder)`
  - Typically returns TRUE.

- `files ending in ".a" of folder "/usr/lib"`
  - Returns a list of files with the specified ending, such as: /usr/lib/libcpp_kext.a, /usr/lib/libfl.a, /usr/lib/libioabc.a, and others.

Bundle

On HPUX platforms, a bundle object is derived from a product type and used to inspect software installations as managed by the Software Depot. On Macintosh platforms, a bundle refers to a core foundation folder hierarchy derived from CFBundle.

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

Creation Methods

Key Phrase	*Form*	Description
bundle <string>	*NamedGlobal*	Returns a bundle (CFBundle) by name. Mac
bundle of <folder>	*Plain*	Returns a bundle from a folder (if it has one like application folders do). Mac

Creation Methods

Key Phrase	*Form*	Description
bundle <string>	*NamedGlobal*	Returns a bundle (CFBundle) by name. Mac
bundle of <folder>	*Plain*	Returns a bundle from a folder (if it has one like application folders do). Mac

Properties

Key Phrase	*Form*	Return Type	Description
bundle version of <bundle>	*Plain*	<version>	Returns the version of the bundle corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac
creator of <bundle>	*Plain*	<file signature>	The creator 4-letter code of the bundle (for example, FNDR for Finder). Mac
global dictionary of <bundle>	*Plain*	<dictionary>	The bundle's information dictionary. Mac

Key Phrase	*Form*	Return Type	Description
local dictionary of <bundle>	*Plain*	<dictionary>	The bundle's localized information dictionary. Mac
type of <bundle>	*Plain*	<file type>	Returns the file type of the bundle. Can be used only to compare equality between file types. Mac
version of <bundle>	*Plain*	<version>	Version of the given bundle. Mac

Properties

Key Phrase	*Form*	Return Type	Description
bundle version of <bundle>	*Plain*	<version>	Returns the version of the bundle corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac
creator of <bundle>	*Plain*	<file signature>	The creator 4-letter code of the bundle (for example, FNDR for Finder). Mac
global dictionary of <bundle>	*Plain*	<dictionary>	The bundle's information dictionary. Mac
local dictionary of <bundle>	*Plain*	<dictionary>	The bundle's localized information dictionary. Mac
type of <bundle>	*Plain*	<file type>	Returns the file type of the bundle. Can be used only to compare equality between file types. Mac
version of <bundle>	*Plain*	<version>	Version of the given bundle. Mac

Examples

- `type of bundle of applications folder = type of bundle of domain library folder`

  Returns TRUE if the specified types are the same.

Volume

The following Inspectors refer to the mounted drive volumes.

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
drive <integer>	*NumberedGlobal*	Returns the volume corresponding to the numbered drive. Mac
filesystem	*PlainGlobal*	Returns the volume corresponding to the filesystem. Typically used to return a list of the filesystems (drives, volumes) on the client computer. Drives, volumes and filesystems are treated the same on the Macintosh and all return a <volume> type. Mac
filesystem <integer>	*NumberedGlobal*	Returns the volume corresponding to the numbered filesystem. Mac
filesystem <string>	*NamedGlobal*	Returns the volume corresponding to the named filesystem. Mac
filesystem of <file>	*Plain*	Returns the filesystem (volume or drive) containing the specified file. Drive, volume and filesystem are treated the same on the Mac. Mac
filesystem of <folder>	*Plain*	Returns the filesystem (volume or drive) containing the specified folder. Drive, volume and filesystem are treated the same on the Mac. Mac
volume	*PlainGlobal*	An iterated property. Examines all currently mounted volumes which will include the startup volume, CD-ROM, floppies and other removable media and file sharing volumes. Mac
volume <integer>	*NumberedGlobal*	A numbered property. Volume 1 is the startup volume. Others follow in sequence. Mac

Key Phrase	*Form*	Description
volume <string>	*NamedGlobal*	A named property representing a volume. Mac
volume of <file>	*Plain*	The volume containing the file. Mac
volume of <folder>	*Plain*	The volume containing the specified folder. Mac

Properties

Key Phrase	*Form*	Return Type	Description
allocation block count of <volume>	*Plain*	<integer>	The number of allocation blocks, used or free, on the volume. Mac
directory count of <volume>	*Plain*	<integer>	The number of directories on the volume. Mac
file count of <volume>	*Plain*	<integer>	The number of files on the volume. Mac
flag of <volume>	*Plain*	<integer>	Returns the Mac-specific volume flags.More info on these flags can be found at the Apple developer site, under Carbon > Reference > File Manager. Mac
free percent of <volume>	*Plain*	<integer>	Returns the percentage of room available on the specified volume. Mac
free space of <volume>	*Plain*	<integer>	The number of free bytes on the volume. Mac
init date of <volume>	*Plain*	<time>	Returns the initialization date of a disk volume. Mac
modification time of <volume>	*Plain*	<time>	Creates a time object corresponding to the time the volume was last modified. Mac
name of <volume>	*Plain*	<string>	The name of the volume. Mac
size of <volume>	*Plain*	<integer>	Returns a number corresponding to the total number of bytes (used and unused) on the specified volume. Mac

Key Phrase	*Form*	Return Type	Description
total space of <volume>	*Plain*	<integer>	The total amount of space, used and free, on the volume. Mac
type of <volume>	*Plain*	<string>	This Mac Inspector returns the drive type as a string, such as 'DRIVE_FIXED'. Mac:8.1
used percent of <volume>	*Plain*	<integer>	Returns the used percentage of room on the specified volume (or drive or filesystem). Mac
used space of <volume>	*Plain*	<integer>	Returns a number corresponding to the used bytes on the specified volume. Mac

Operators

Key phrase	Return Type	Description
<volume> = <volume>	*<boolean>*	Compare two volumes. Mac

Examples

- `filesystems`
  - Returns a list of filesystems, eg., /, /Users/MyUserName, etcetera.
- `filesystem of folder "/Users/MyUserName/Library"`
  - Returns /Users/MyUserName.
- `free percentage of volume 1`
  - Returns a number between 0 and 100.
- `init date of volume of system folder`
  - Returns the date the system folder was initialized.
- `size of volume 1`
  - Returns the size of the specified volume in bytes, eg. 159697911808.
- `used percentage of volume 2`
  - Returns a number between 0 and 100.

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

Properties

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

Key Phrase	*Form*	Description
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
bundle version of <bundle>	*Plain*	Returns the version of the bundle corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac
bundle version of <filesystem object>	*Plain*	Returns the version of the filesystem object corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac
bundle version of <folder>	*Plain*	Returns the version of the folder corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac
quickdraw version	*PlainGlobal*	Returns the version of QuickDraw installed. Mac
rom version	*PlainGlobal*	Returns the version of the system ROM. Mac
system version	*PlainGlobal*	Returns the version of MacOS. Mac
version <integer> of <file>	*Numbered*	Returns the nth version information from the "vers" resource of the given file. Typically n=1, but other information may be stored in "vers" resources greater than 1. Mac
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 <bundle>	*Plain*	Version of the given bundle. Mac
version of <client>	*Plain*	The product version of the BES application (BESClient or QnA). Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
version of <component>	*Plain*	The version of the component, as determined by the component maker. Mac
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 <filesystem object>	*Plain*	This returns the version information from "vers" resource 1 of the file. It is usually present in applications, and may exist in data files as well. It corresponds to what appears in the "Get Info" box for the specified filesystem object. Mac
version of <folder>	*Plain*	Returns a version for a folder representing an application. Mac
version of <operating system>	*Plain*	Returns the version of the operating system. Win:8.0, Mac:8.0
version of <scsibus>	*Plain*	Version of the SCSI bus. Mac
version of <usb>	*Plain*	The version of the USB installed in the system. Taken from the Gestalt Manager. Mac

Properties

Key Phrase	*Form*	Return Type	Description
bug revision of <version>	*Plain*	<integer>	If the stage is present the bug revision is the number after the stage. If absent, zero is implied. Mac
build revision of <version>	*Plain*	<integer>	Returns the final component of a version (Major.Minor.RevisionStageBuild). Mac:7.1
major revision of <version>	*Plain*	<integer>	The number before the first period in the version string. Mac
minor revision of <version>	*Plain*	<integer>	The number immediately after the first period in the version string. If absent, zero is implied. Mac

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 bundle of folder "iTunes.app" of applications folder`
  - Returns the version of the specified folder bundle, such as 4.2.
- `version of client as string`
  - Returns a string like "4.0.3.7".
- `build revision of version "7.1.2.70"`
  - Returns 70.
- `major revision of version of file "name" > 4`
  - Returns TRUE if the major revision number is greater than the specified number.

Domain

Mac OS X defines several file system domains to control access to system resources on multi-user systems. These include the User, Local, Network, Classic and System domains. The domain for a given resource or folder determines its accessibility to the user. For example, while a user-installed font is only available to that user, an administrator-installed font is available to all network users. The following Inspectors allow folder access to be parceled out according to domain.

- NOTE: The "user domain" refers to the root user, not the currently logged in user.

Creation Methods

Key Phrase	*Form*	Description
classic domain	*PlainGlobal*	Returns a classic domain object. Mac
local domain	*PlainGlobal*	Returns a local domain object. Mac
network domain	*PlainGlobal*	Returns a network domain object. Mac
on appropriate disk domain	*PlainGlobal*	Returns one of the Macintosh domains. In most cases, this is the equivalent of kOnAppropriateDisk. On Mac OS X, this constant is used instead of the constant kOnSytemDisk to indicate any disk. For more information, see the Apple documentation on Carbon domain constants. Mac

Key Phrase	*Form*	Description
on system disk domain	*PlainGlobal*	Returns the OnSystemDisk domain. Mac
system domain	*PlainGlobal*	Returns a system domain object. Mac
user domain	*PlainGlobal*	Returns a user domain object. • NOTE: The "user domain" refers to the root user, not the currently logged in user. Mac

Properties

Key Phrase	*Form*	Return Type	Description
apple extras folder of <domain>	*Plain*	<folder>	Returns the apple extras folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
apple menu items folder of <domain>	*Plain*	<folder>	Returns the apple menu items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
application support folder of <domain>	*Plain*	<folder>	Returns the application support folder of the specified OS X domain, typically /Library/Application Support. If the domain is not specified, it defaults to the system domain. Mac
applications folder of <domain>	*Plain*	<folder>	Returns the applications folder of the specified OS X domain, typically /Applications. If the domain is not specified, it defaults to the system domain. Mac
assistants folder of <domain>	*Plain*	<folder>	Returns the assistants folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
audio folder of <domain>	*Plain*	<folder>	Returns the audio folder of the specified OS X domain, typically /Library/Audio. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Return Type	Description
cache folder of <domain>	*Plain*	<folder>	Returns the cache folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
carbon folder of <domain>	*Plain*	<folder>	Returns the carbon folder of the specified OS X domain, typically /Library/Carbon. If the domain is not specified, it defaults to the system domain. Mac
chewable items folder of <domain>	*Plain*	<folder>	Returns the chewable items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
classic folder of <domain>	*Plain*	<folder>	Returns the classic folder of the specified OS X domain, typically the /System Folder. If the domain is not specified, it defaults to the system domain. Mac
color sync folder of <domain>	*Plain*	<folder>	Returns the color sync folder of the specified OS X domain, typically /System/Library/ColorSync. If the domain is not specified, it defaults to the system domain. Mac
colorsync profiles folder of <domain>	*Plain*	<folder>	Returns the colorsync profiles folder of the specified OS X domain, typically /System/Library/ColorSync/Profiles. If the domain is not specified, it defaults to the system domain. Mac
component folder of <domain>	*Plain*	<folder>	Returns the component folder of the specified OS X domain, typically /System/Library/Components. If the domain is not specified, it defaults to the system domain. Mac
contextual menu items folder of <domain>	*Plain*	<folder>	Returns the contextual menu items folder of the specified OS X domain, typically /Library/Contextual Menu Items. If the domain is not specified, it defaults to the system domain. Mac
control panels folder of <domain>	*Plain*	<folder>	Returns the control panels folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Return Type	Description
control strip modules folder of <domain>	*Plain*	<folder>	Returns the control strip modules folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
core services folder of <domain>	*Plain*	<folder>	Returns the core services folder of the specified OS X domain, typically /System/Library/CoreServices. If the domain is not specified, it defaults to the system domain. Mac
current user folder of <domain>	*Plain*	<folder>	Returns the current user folder of the specified OS X domain, typically found at /Users/username. If the domain is not specified, it defaults to the system domain. Mac
desktop folder of <domain>	*Plain*	<folder>	Returns the desktop folder of the specified OS X domain, typically /Users/Username/Desktop. If the domain is not specified, it defaults to the system domain. Mac
developer docs folder of <domain>	*Plain*	<folder>	Returns the developer docs folder of the specified OS X domain, typically found at /Developer/Documentation. If the domain is not specified, it defaults to the system domain. Mac
developer folder of <domain>	*Plain*	<folder>	Returns the developer folder of the specified OS X domain, typically found at /Developer. If the domain is not specified, it defaults to the system domain. If the domain is not specified, it defaults to the system domain. Mac
developer help folder of <domain>	*Plain*	<folder>	Returns the help folder of the specified OS X domain, typically /Developer/Documentation/Help. If the domain is not specified, it defaults to the system domain. Mac
disabled control panels folder of <domain>	*Plain*	<folder>	Returns the control panels folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Return Type	Description
disabled extensions folder of <domain>	*Plain*	<folder>	Returns the extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
disabled shutdown items folder of <domain>	*Plain*	<folder>	Returns the shutdown folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
disabled startup items folder of <domain>	*Plain*	<folder>	Returns the startup items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
disabled system extensions folder of <domain>	*Plain*	<folder>	Returns the systems extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
documentation folder of <domain>	*Plain*	<folder>	Returns the documentation folder for the given OS X domain, typically found at /Library/Documentation. If the domain is not specified, it defaults to the system domain. Mac
documents folder of <domain>	*Plain*	<folder>	Returns the documents folder for the specified OS X domain, typically found at /User/Username/Documents. If the domain is not specified, it defaults to the system domain. Mac
domain library folder of <domain>	*Plain*	<folder>	Returns the domain library folder of the specified OS X domain, typically found at /Library. If the domain is not specified, it defaults to the system domain. Mac
domain top folder of <domain>	*Plain*	<folder>	Returns the top folder of the specified OS X domain, typically found at /System. If the domain is not specified, it defaults to the system domain. Mac
extensions folder of <domain>	*Plain*	<folder>	Returns the extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Return Type	Description
favorites folder of <domain>	*Plain*	<folder>	Returns the favorites folder of the specified OS X domain, typically /Users/username/Library/Favorites. If the domain is not specified, it defaults to the system domain. Mac
fonts folder of <domain>	*Plain*	<folder>	Returns the font folder of the specified OS X domain, typically /System/Library/Fonts. If the domain is not specified, it defaults to the system domain. Mac
framework <string> of <domain>	*Named*	<folder>	Returns a folder of the form "/System/Library/Frameworks/<string>.framework". • Note: This Inspector appends .framework for you, so don't provide it. The framework inspector needs a domain, and without it defaults to the system domain. Mac
framework folder of <domain>	*Plain*	<folder>	Returns the framework folder of the specified OS X domain, typically /System/Library/Frameworks. If the domain is not specified, it defaults to the system domain. Mac
help folder of <domain>	*Plain*	<folder>	Returns the help folder of the specified OS X domain, typically /Library/Documentation/Help. If the domain is not specified, it defaults to the system domain. Mac
internet plugins folder of <domain>	*Plain*	<folder>	Returns the internet plugins folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
iss download folder of <domain>	*Plain*	<folder>	Returns the Macintosh download folder for the specified domain. Mac:7.2
kernel extensions folder of <domain>	*Plain*	<folder>	Returns the kernel extensions folder of the specified OS X domain, typically /System/Library/Extensions. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Return Type	Description
locales folder of <domain>	*Plain*	<folder>	Returns the locales folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
location manager modules folder of <domain>	*Plain*	<folder>	Returns the location manager modules folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
location manager preferences folder of <domain>	*Plain*	<folder>	Returns the location manager preferences folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
locations folder of <domain>	*Plain*	<folder>	Returns the locations folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
macos read me folder of <domain>	*Plain*	<folder>	Returns the Mac OS read me folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
modem scripts folder of <domain>	*Plain*	<folder>	Returns the modem scripts folder of the specified OS X domain, typically /System/Library/Modem Scripts. If the domain is not specified, it defaults to the system domain. Mac
preferences folder of <domain>	*Plain*	<folder>	Returns the preferences folder of the specified OS X domain, typically /Users/username/Library/Preferences. If the domain is not specified, it defaults to the system domain. Mac
printer descriptions folder of <domain>	*Plain*	<folder>	Returns the printer descriptions folder of the specified OS X domain, typically /System/Library/Printers/PPDs. If the domain is not specified, it defaults to the system domain. Mac
printer drivers folder of <domain>	*Plain*	<folder>	Returns the printer drivers folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Return Type	Description
printers folder of <domain>	*Plain*	<folder>	Returns the printers folder of the specified OS X domain, typically /System/Library/Printers. If the domain is not specified, it defaults to the system domain. Mac
printmonitor documents folder of <domain>	*Plain*	<folder>	Returns the printmonitor documents folder of the specified OS X domain, typically /Library/Printers/PrintMonitor Documents. If the domain is not specified, it defaults to the system domain. Mac
private framework folder of <domain>	*Plain*	<folder>	Returns the private framework folder of the specified OS X domain, typically /System/Library/PrivateFrameworks. If the domain is not specified, it defaults to the system domain. Mac
quicktime folder of <domain>	*Plain*	<folder>	Returns the quicktime folder of the specified OS X domain, typically /System/Library/QuickTime. If the domain is not specified, it defaults to the system domain. Mac
receipts folder of <domain>	*Plain*	<folder>	Returns the receipts folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
scripting additions folder of <domain>	*Plain*	<folder>	Returns the scripting additions folder of the specified OS X domain, typically /System/Library/Scripting Additions. If the domain is not specified, it defaults to the system domain. Mac
shared folder of <domain>	*Plain*	<folder>	Returns the shared folder of the specified OS X domain, typically /Users/Shared. If the domain is not specified, it defaults to the system domain. Mac
shared libraries folder of <domain>	*Plain*	<folder>	Returns the shared libraries folder of the specified OS X domain, typically /System/Library/CFMSupport. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Return Type	Description
shutdown items folder of <domain>	*Plain*	<folder>	Returns the shutdown items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
sound folder of <domain>	*Plain*	<folder>	Returns the sound folder of the specified OS X domain, typically /System/Library/Sound. If the domain is not specified, it defaults to the system domain. Mac
speech folder of <domain>	*Plain*	<folder>	Returns the speech folder of the specified OS X domain, typically /System/Library/Speech. If the domain is not specified, it defaults to the system domain. Mac
startup items folder of <domain>	*Plain*	<folder>	Returns the startup items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
system folder of <domain>	*Plain*	<folder>	Returns the system folder of the specified OS X domain, typically /System. If the domain is not specified, it defaults to the system domain. Mac
temporary items folder of <domain>	*Plain*	<folder>	Returns the temporary items folder of the specified OS X domain, typically /private/tmp/. If the domain is not specified, it defaults to the system domain. Mac
text encodings folder of <domain>	*Plain*	<folder>	Returns the text encodings folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
themes folder of <domain>	*Plain*	<folder>	Returns the themes folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac
user temp folder of <domain>	*Plain*	<folder>	Returns the user temp folder of the specified OS X domain, typically /private/tmp/uid where uid is the user ID number. If the domain is not specified, it defaults to the system domain. Mac

Key Phrase	*Form*	Return Type	Description
users folder of <domain>	*Plain*	<folder>	Returns the users folder of the specified OS X domain, typically /Users. If the domain is not specified, it defaults to the system domain. Mac
utilities folder of <domain>	*Plain*	<folder>	Returns the utilities folder of the specified OS X domain, typically /Applications/Utilities. If the domain is not specified, it defaults to the system domain. Mac
voices folder of <domain>	*Plain*	<folder>	Returns the voices folder of the specified OS X domain, typically /System/Library/Speech/Voices. If the domain is not specified, it defaults to the system domain. Mac
volume settings folder of <domain>	*Plain*	<folder>	Returns the volume settings folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac

File Type

These Inspectors provide access to the four character file type associated with some types of files.

Creation Methods

Key Phrase	*Form*	Description
file type <string>	*NamedGlobal*	The phrase 'file type' can be used to create a file type object from a string. Mac
type of <bundle>	*Plain*	Returns the file type of the bundle. Can be used only to compare equality between file types. Mac
type of <file>	*Plain*	Refers to the file type of the specified file. Mac

Operators

Key phrase	Return Type	Description
<file type> = <file type>	<*boolean*>	Compare two file types. Mac

Examples

- `creator of "that file" of "that folder" is file type "ttxt"`
- Returns TRUE if "that file" is a TeachText file.

- `type of "that file" is type of "this file"`
- Compares the types of two files.

File Signature

These Inspectors provide access to each of the four character file signatures associated with some types of files.

Creation Methods

Key Phrase	*Form*	Description
creator of <bundle>	*Plain*	The creator 4-letter code of the bundle (for example, FNDR for Finder). Mac
creator of <file>	*Plain*	This refers to the four-character identifier used in all MacOS files. The creator tells what application should open the specified file. Mac

Operators

Key phrase	Return Type	Description
<file signature> = <file signature>	<*boolean*>	Compares two Macintosh file signatures. Mac

Examples

- `creator of bundle of applications folder = creator of bundle of preferences folder`
- Returns TRUE if the creator of the application and preference folders are the same.

Component

Components are handled by the MacOS Component Manager. These Inspectors provide access to the various software components available through the Component Manager. The information contained in a component is placed there by its maker. How it should be interpreted is also determined by the maker.

Creation Methods

Key Phrase	*Form*	Description
component	*PlainGlobal*	Is an iterated property. The MacOS supports software "components", for example QuickTime codecs. This iterator can examine the components that are available. Mac

Properties

Key Phrase	*Form*	Return Type	Description
info of <component>	*Plain*	<string>	Returns information about the component, as determined by the creator of the component. Mac
maker of <component>	*Plain*	<string>	Returns the name of the maker of the specified component. Mac
name of <component>	*Plain*	<string>	The name of the component, as determined by its maker. Mac
subtype of <component>	*Plain*	<string>	The subtype of the component, as determined by the component maker. Mac
type of <component>	*Plain*	<string>	The type of the component, as determined by the component maker. Mac
version of <component>	*Plain*	<version>	The version of the component, as determined by the component maker. Mac

Datafork

These Inspectors refer to the data fork of a filesystem object.

Creation Methods

Key Phrase	*Form*	Description
data fork of <file>	*Plain*	Returns information about the data fork of the specified file. Mac

Properties

Key Phrase	*Form*	Return Type	Description
length of <datafork>	*Plain*	<integer>	The logical length of the data fork of the file. Mac
size of <datafork>	*Plain*	<integer>	Returns the size of the specified datafork. Mac

Resfork

These Inspectors refer to the resource fork of a filesystem object.

Creation Methods

Key Phrase	*Form*	Description
resource fork of <file>	*Plain*	Returns information about the resource fork of the file. Mac

Properties

Key Phrase	*Form*	Return Type	Description
length of <resfork>	*Plain*	<integer>	The logical length of the resource fork of the file. Mac
size of <resfork>	*Plain*	<integer>	Returns the size of the resource fork. Mac

Dictionary

These Inspectors provide access to an XML dictionary as used in .plist (property list) files.

Creation Methods

Key Phrase	*Form*	Description
dictionary <integer> of <array>	*Numbered*	Get, from an array, a dictionary keyed by the specified integer. Mac
dictionary <string> of <dictionary>	*Named*	Get, from a dictionary, a dictionary keyed by the specified string. Mac
dictionary <string> of <preference>	*Named*	Get, from a preference, a dictionary keyed by the specified string. Mac
dictionary of <file>	*Plain*	Returns the dictionary object for the specified file, if it exists. Mac
dictionary of <osxvalue>	*Plain*	Cast the osxvalue (essentially untyped) to a dictionary. Mac
dictionary of <registrynode>	*Plain*	Returns a dictionary from a node in the IORegistry. Mac
dictionary of <registryroot>	*Plain*	Returns a dictionary from the root of the registry. Mac
global dictionary of <bundle>	*Plain*	The bundle's information dictionary. Mac
local dictionary of <bundle>	*Plain*	The bundle’s localized information dictionary. Mac

Properties

Key Phrase	*Form*	Return Type	Description
array <string> of <dictionary>	*Named*	<array>	Get, from a dictionary, an array keyed by the specified string. Mac
boolean <string> of <dictionary>	*Named*	<boolean>	Get, from a dictionary, a boolean keyed by the string. Mac

Key Phrase	*Form*	Return Type	Description
cstring <string> of <dictionary>	*Named*	<string>	Get, from a dictionary, the string keyed by the specified string. Works for objects in a dictionary of type String or Data. If the type is Data, any null terminator will be stripped. Mac
data <string> of <dictionary>	*Named*	<string>	Returns the specified dictionary data as a string. Mac
date <string> of <dictionary>	*Named*	<time>	Get, from a dictionary, a date keyed by the specified string. Mac
dictionary <string> of <dictionary>	*Named*	<dictionary>	Get, from a dictionary, a dictionary keyed by the specified string. Mac
entry of <dictionary>	*Plain*	<dictionaryentry>	A key-value pair of a dictionary. Mac
integer <string> of <dictionary>	*Named*	<integer>	Get, from a dictionary, an integer keyed by the specified string. Mac
key of <dictionary>	*Plain*	<string>	The keys of a dictionary. Mac
size of <dictionary>	*Plain*	<integer>	The size of the given dictionary. Mac
string <string> of <dictionary>	*Named*	<string>	Get, from a dictionary, the string keyed by the specified string. Works for objects in a dictionary of type String or Data. If the type is Data, any null terminator will be returned as part of the string. Use cstring if you expect a Data object containing a null terminated string. Mac

Examples

- `exists dictionary 0 of array "LogFileDicts" of preference "com.apple.Console"`

  Returns TRUE if the specified dictionary exists.

- `exists dictionary "Timer" of dictionary "SUCheckSchedulerTag" of dictionary "com.apple.SoftwareUpdate" of dictionary "AbsoluteSchedule" of dictionary of file "com.apple.scheduler.plist" of preferences folder`

  Returns TRUE if the specified dictionary exists.

- `exists dictionary "Timer" of dictionary "SUCheckSchedulerTag" of dictionary "com.apple.SoftwareUpdate" of dictionary "AbsoluteSchedule" of preference "com.apple.scheduler"`
  - Returns TRUE if the specified dictionary exists.

- `exists dictionary of file "com.apple.scheduler.plist" of preferences folder`
  - Returns TRUE if the specified dictionary exists.

- `keys of entries of dictionary of usb plane of iokit registry`
  - Returns the specified keys, for example: Device Speed, iSerialNumber, bNumConfigurations, IOUserClientClass, AAPL, current-available, bDeviceClass, USB Product Name, IOCFPlugInTypes.

- `keys of entries of dictionary of iokit registry`
  - Returns the specified keys, for example: IOKitBuildVersion, IONDRVFrameBufferGeneration, IOConsoleUsers, IOKitDiagnostics, IORegistryPlanes, IOMaximumMappedIOByteCount, IOCatalogue.

- `string "CFBundleVersion" of global dictionary of bundle of folder "iTunes.app" of applications folder`
  - Returns the specified bundle version.

- `keys of entries of local dictionary of bundle of folder "iTunes.app" of applications folder`
  - Returns information in the default system langauge for bundles that include localized resources, such as Movie File, iTunes Remote Library URL, MP2 Audio File, WAVE Audio File, MP3 Audio File, CFBundleHelpBookName, Equalizer Settings, Tunes Database File, and more...

- `boolean "trash-full" of dictionary of file "com.apple.dock.plist" of preferences folder`
  - Returns TRUE if the given flag is set.

- `date "date" of dictionary "Timer" of dictionary "SUCheckSchedulerTag" of dictionary "com.apple.SoftwareUpdate" of dictionary "AbsoluteSchedule" of dictionary of file "com.apple.scheduler.plist" of preferences folder`
  - Returns the date of the specified dictionary.

- `integer "mod-count" of dictionary of file "com.apple.dock.plist" of preferences folder`
  - Returns the mod-count of the specified dictionary as an integer.

- `keys of global dictionary of bundle of folder "iTunes.app" of applications folder`
  - Returns a list of the specified keys, for example: CFBundlePackageType, CFBundleVersion, CFBundleHelpBookName, CFBundleName, CSResourcesFileMapped, LSMulitpleInstancesProhibted, CFBundleDocumentTypes, etc….

- `sizes of dictionaries of file "com.apple.help.plist" of preferences folder`
  - Returns a list of the number of elements in the specified dictionary.

- `string "FXSearchFieldTarget" of dictionary of file "com.apple.finder.plist" of preferences folder`
  - Returns a string, such as Spcf.

Country

These Inspectors allow comparing countries (as seen in version numbers).

Creation Methods

Key Phrase	*Form*	Description
country <string>	*NamedGlobal*	Maps a country script string (for example verAfrikaans) to its region code. Mac

Operators

Key phrase	Return Type	Description
<country> = <country>	*<boolean>*	Compares two countries, as found in a Macintosh version. Mac

Examples

- `exists (country "verChina")`
  - See the Apple developer documentation for more information.

Osxvalue

These Inspectors provide access to the elements of a dictionary.

Creation Methods

Key Phrase	*Form*	Description
value of <array>	*Plain*	Values of the array. Mac
value of <dictionaryentry>	*Plain*	Values of the dictionary entry. Mac

Properties

Key Phrase	*Form*	Return Type	Description
array of <osxvalue>	*Plain*	<array>	Casts the osxvalue (which is essentially untyped) to an array. Mac
boolean of <osxvalue>	*Plain*	<boolean>	Casts the osxvalue (which is essentially untyped) to a boolean. Mac
cstring of <osxvalue>	*Plain*	<string>	Strips off any trailing null from an osxvalue, returning a proper string from the dictionary. • NOTE: In Version 6.0.9.54, the plural inspector is misspelled as csrings. Mac
data of <osxvalue>	*Plain*	<string>	Casts the specified osxvalue to a string. Mac
date of <osxvalue>	*Plain*	<time>	Casts the osxvalue (essentially untyped) to a date. Mac
dictionary of <osxvalue>	*Plain*	<dictionary>	Casts the osxvalue (essentially untyped) to a dictionary. Mac
integer of <osxvalue>	*Plain*	<integer>	Casts the osxvalue (essentially untyped) to an integer. Mac
string of <osxvalue>	*Plain*	<string>	The string of the osxvalue, cast to a string if necessary. Mac
type of <osxvalue>	*Plain*	<string>	The type of the osxvalue (for example boolean, string, integer, data, date, array, dictionary, or unknown type). Mac

Examples

- `strings of values of array "RecentSearchStrings" of preference "com.apple.safari"`

  Returns a list of the most recent search strings, in temporal order.

- `strings of values of entries of dictionary of file "com.apple.safari" of preferences folder`

- Returns a list of strings corresponding to the dictionary values of the specified file, for example: 12/19/2002, /Users/bigfix/Library/Safari/Icons, 125.12, }%00%00%00, 2004-02-20 19:36:50 -0800, 262 485 424 261 0 0 1024 746, etc….

Preference

These Inspectors provide access to application preference files.

Creation Methods

Key Phrase	*Form*	Description
preference <string>	*NamedGlobal*	The named set of preferences. Mac

Properties

Key Phrase	*Form*	Return Type	Description
array <string> of <preference>	*Named*	<array>	Get, from a preference file, an array keyed by the specified string. Mac
boolean <string> of <preference>	*Named*	<boolean>	Get, from a preference file, a boolean keyed by the specified string. Mac
date <string> of <preference>	*Named*	<time>	Get, from a preference, a date keyed by the specified string. Mac
dictionary <string> of <preference>	*Named*	<dictionary>	Get, from a preference, a dictionary keyed by the specified string. Mac
integer <string> of <preference>	*Named*	<integer>	Get, from a preference, an integer keyed by the specified string. Mac
string <string> of <preference>	*Named*	<string>	Get, from a preference, the string keyed by the specified string. Mac

Examples

- `boolean "autohide" of preference "com.apple.dock"`
- Returns TRUE if the autohide preference is set.

- `date "date" of dictionary "Timer" of dictionary "SUCheckSchedulerTag" of dictionary "com.apple.SoftwareUpdate" of dictionary "AbsoluteSchedule" of preference "com.apple.scheduler"`
  - Returns the date of the specified dictionary.

- `integer "mod-count" of preference "com.apple.dock"`
  - Returns the mod-count of the specified preference as an integer.

- `string "FXSearchFieldTarget" of preference "com.apple.finder"`
  - Returns a string, such as Spcf.

Stage

These are for inspecting the stage portion of version numbers, which is the penultimate section of a version string: Major.Minor.ReleaseStageBuild.

Properties

Key Phrase	*Form*	Return Type	Description
<stage> as string	*Cast*	<string>	A Macintosh version is of the form: Major.Minor.ReleaseStageBuild. The stage directly precedes the final (Build) number. Most versioning schemes use a period, but the Mac allows for d, a, b and f standing for Development , Alpha, Beta and Final. The period has a higher value than Final. The stage is used in version comparisons and is ranked higher than the build number. For instance, 7.1.2a43 is less than 7.1.2f42 because a is lower than f, even though the final number is bigger. Mac:7.1

Examples

- `stage of version "7.1.2b70"`
  - Returns b.

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
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

Key Phrase	*Form*	Return Type	Description
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

Directory Services

These keywords give you access to Active Directory objects.

Active Directory Local Computer

These are the Active Directory Inspectors for the local computer. Caching (using XML files to represent the last data update) limits how often the client refreshes the AD data, reducing network traffic. These inspectors require network access to work. If no network connection is available and the information is not cached, then the inspectors will return NoSuchObject. AD user requests are limited to users that are either cached or have a local profile on the specified machine

CAUTION: These Windows Inspectors utilize the ADsGetObject API to collect the active directory properties of the machine. In situations where the Active Directory Inspector succeeds, it caches the results and will attempt to update the value every 12 hours. If it cannot communicate with the active directory infrastructure, it will try up to 5 times separated by 1 minute intervals. If will then wait one hour between subsequent attempts. As a consequence, these Inspectors may take more time than expected. Use the _BESClient_ActiveDirectoryPathOverride setting to modify this behavior.

Creation Methods

Key Phrase	*Form*	Description
local computer of <active directory server>	*Plain*	Represents your computer within the Active Directory. Win, Mac

Properties

Key Phrase	*Form*	Return Type	Description
distinguished name error message of <active directory local computer>	*Plain*	<string>	Active Directory error if unable to get the distinguished name (this is for debugging purposes). Win, Mac

Key Phrase	*Form*	Return Type	Description
distinguished name of <active directory local computer>	*Plain*	<string>	Returns the computer's fully qualified active directory name in the distinguished name format, for instance, 'CN=ALBATROSS, CN=Computers, DC=bigfix, DC=com'. Win, Mac
group <string> of <active directory local computer>	*Named*	<active directory group>	Returns the Active Directory group corresponding to the specified group of the given AD local computer. Win:8.1, Mac:8.1
group of <active directory local computer>	*Plain*	<active directory group>	Returns a list of the active directory groups for the specified local user. Win:8.1, Mac:8.1
groups error message of <active directory local computer>	*Plain*	<string>	Returns the error message (if any) received when trying to get the groups for the specified active directory local computer. Win:8.1, Mac:8.1
sample time of <active directory local computer>	*Plain*	<time>	Returns the time that the specified item was sampled from Active Directory. Win:8.0, Mac:8.0

Examples

- `distinguished name of local computer of active directory`
- Returns CN=mymachinename,CN=Computers,DC=bigfix,DC=com.

- `sample time of local computer of active directory`
- Returns a time corresponding to the specified sample time.

Active Directory Server

These are the Active Directory Server Inspectors. These are the base types that allows access to the AD objects such as local machine and local user.

CAUTION: These Windows Inspectors utilize the ADsGetObject API to collect the active directory properties of the machine. In situations where the Active Directory Inspector succeeds, it caches the results and will attempt to update the value every 12 hours. If it cannot communicate with the active directory infrastructure, it will try up to 5 times separated by 1 minute intervals. If will then wait one hour between subsequent attempts. As a consequence, these Inspectors may take more time than expected. Use the _BESClient_ActiveDirectoryPathOverride setting to modify this behavior.

Creation Methods

Key Phrase	*Form*	Description
active directory	*PlainGlobal*	Returns an object containing the properties of the Active Directory to which your machine is attached. Win, Mac

Properties

Key Phrase	*Form*	Return Type	Description
local computer of <active directory server>	*Plain*	<active directory local computer>	Represents your computer within the Active Directory. Win, Mac
local user <string> of <active directory server>	*Named*	<active directory local user>	Returns the named local user of the specified active directory server. Win:8.1, Mac:8.1
local user of <active directory server>	*Plain*	<active directory local user>	Returns the local users associated with the specified active directory server. Win:8.1, Mac:8.1
logged on user <string> of <active directory server>	*Named*	<active directory local user>	Returns the Active Directory local user object which allows inspection of AD properties for the specified currently logged in user. Win:8.1, Mac:8.1
logged on user of <active directory server>	*Plain*	<active directory local user>	Returns the Active Directory local user object which allows inspection of AD properties for the currently logged in users of the specified AD server. Win:8.1, Mac:8.1

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 number of <operating system>	*Plain*	<string>	Returns the integer build number (as a string) of the operating system. Mac
build of <operating system>	*Plain*	<string>	Returns a string corresponding to the build number of the OS. Lin, Sol, HPUX, AIX, Mac, 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
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

Key Phrase	*Form*	Return Type	Description
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
version of <operating system>	*Plain*	<version>	Returns the version of the operating system. Win:8.0, Mac:8.0
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.

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

Properties

Key Phrase	*Form*	Return Type	Description
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

Key Phrase	*Form*	Return Type	Description
speed of <processor>	*Plain*	<hertz>	Returns the speed of the processor in Hertz. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
type of <processor>	*Plain*	<string>	Numeric type of the CPU. Values include: • 0 - standard • 1 - overdrive • 2 - dual CPU capable • 3 - reserved • Note: this Inspector returns an <integer> type as on Windows platforms. Sol, AIX, Mac

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.

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

Properties

Key Phrase	*Form*	Return Type	Description
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

Examples

- `size of ram / (1024 * 1024)`

Returns the size of RAM in megabytes.

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

Properties

Key Phrase	*Form*	Return Type	Description
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
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.

Computer

These Inspectors provide access to the name of the computer.

Creation Methods

Key Phrase	*Form*	Description
computer	*PlainGlobal*	Refers to the computer itself. Mac

Properties

Key Phrase	*Form*	Return Type	Description
name of <computer>	*Plain*	<string>	The name of the computer. Mac

Examples

- `name of computer`
- Returns the name of the computer.

Registryroot

These are the Inspectors for the planes of the IOKit Registry

Creation Methods

Key Phrase	*Form*	Description
iokit registry	*PlainGlobal*	Returns the root of the IOKit registry. Mac

Properties

Key Phrase	*Form*	Return Type	Description
audio plane of <registryroot>	*Plain*	<registrynode>	The audio plane of IOKit's registry. Mac
devicetree plane of <registryroot>	*Plain*	<registrynode>	Returns the device tree plane of IOKit's registry. The device tree contains extensive information about devices in the system. Mac

Key Phrase	*Form*	Return Type	Description
dictionary of <registryroot>	*Plain*	<dictionary>	Returns a dictionary from the root of the IOKit registry. Mac
firewire plane of <registryroot>	*Plain*	<registrynode>	The firewire plane of IOKit's registry. Mac
node <string> of <registryroot>	*Named*	<registrynode>	The named node of the root of the IOKit's registry. Mac
power plane of <registryroot>	*Plain*	<registrynode>	The power plane of IOKit's registry. Mac
service plane of <registryroot>	*Plain*	<registrynode>	The service plane of IOKit's registry. Mac
usb plane of <registryroot>	*Plain*	<registrynode>	The usb plane of IOKit's registry. Mac

Examples

- `name of audio plane of iokit registry`
- Returns the name of the audio portion of the IOKit registry.

- `name of devicetree plane of iokit registry`
- Returns the name of the device tree.

Registrynode

These Inspectors provide access to the nodes of the IOKit Registry.

Creation Methods

Key Phrase	*Form*	Description
audio plane of <registryroot>	*Plain*	The audio plane of IOKit's registry. Mac
devicetree plane of <registryroot>	*Plain*	Returns the device tree plane of IOKit's registry. Mac
firewire plane of <registryroot>	*Plain*	The firewire plane of IOKit's registry. Mac
node <string> of <registrynode>	*Named*	The named node of the given node of the IOKit registry. Mac

Key Phrase	*Form*	Description
node <string> of <registryroot>	*Named*	The named node of the root of the IOKit's registry. Mac
node of <registrynode>	*Plain*	The nodes of the given node of the IOKit registry. Mac
power plane of <registryroot>	*Plain*	The power plane of IOKit's registry. Mac
service plane of <registryroot>	*Plain*	The service plane of IOKit's registry. Mac
usb plane of <registryroot>	*Plain*	The usb plane of IOKit's registry. Mac

Properties

Key Phrase	*Form*	Return Type	Description
classname of <registrynode>	*Plain*	<string>	The class name of the IOKit registry node. Mac
dictionary of <registrynode>	*Plain*	<dictionary>	Returns a dictionary from a node in the IORegistry. Mac
name of <registrynode>	*Plain*	<string>	Name of the given IOKit registry node. Mac
node <string> of <registrynode>	*Named*	<registrynode>	The named node of the given node of the IOKit registry. Mac
node of <registrynode>	*Plain*	<registrynode>	The nodes of the given node of the IOKit registry. Mac
path of <registrynode>	*Plain*	<string>	Path of the node in the IOKit registry. Mac

Examples

- `name of devicetree plane of iokit registry`
  - See: man ioreg or the Apple Developer documentation.
- `name of firewire plane of iokit registry`
  - See: man ioreg or the Apple Developer documentation.

- `exists node "IOPowerConnection" of power plane of iokit registry`
  - Returns TRUE if the specified node exists.

- `names of nodes whose (name of it contains "DMAEngine") of audio plane of iokit registry`
  - Returns a list of nodes with names containing the specified string, such as Apple02DBDMAAudioDMAEngine.

- `exists (node "IORootParent" of iokit registry)`
  - Returns TRUE if the specified node exists.

- `names of nodes of power plane of iokit registry`
  - Returns a list of names, such as IOPowerConnection.

- `name of power plane of iokit registry`
  - See: man ioreg or the Apple Developer documentation.

- `name of service plane of iokit registry`
  - See: man ioreg or the Apple Developer documentation.

- `name of usb plane of iokit registry`
  - See man ioreg or the Apple Developer documentation.

- `classname of service plane of iokit registry`
  - Returns a classname, such as IOPlatformExpertDevice or IOUSBRootHubDevice.

- `classname of usb plane of iokit registry`
  - Returns a classname for the specified registry.

- `name of power plane of iokit registry`
  - Returns a string, such as IORootParent.

- `path of usb plane of iokit registry`
  - Typically returns the string "IOUSB:/".

Scsibus

These Inspectors refer to the Small Computer System Interface bus components.

Creation Methods

Key Phrase	*Form*	Description
scsibus	*PlainGlobal*	An iterated property. When used without a number and not iterated it means SCSI bus 0. Mac

Key Phrase	*Form*	Description
scsibus <integer>	*NumberedGlobal*	Returns a SCSI bus with the given number. Mac

Properties

Key Phrase	*Form*	Return Type	Description
scsidevice <integer> of <scsibus>	*Numbered*	<scsidevice>	Returns a SCSI device with the given number. Mac
scsidevice of <scsibus>	*Plain*	<scsidevice>	Returns a SCSI device associated with the given bus. Mac
version of <scsibus>	*Plain*	<version>	Version of the SCSI bus. Mac
wide16 scsi of <scsibus>	*Plain*	<boolean>	Whether wide16 is available for the given SCSI bus. Mac
wide32 scsi of <scsibus>	*Plain*	<boolean>	Whether wide32 is available for the given SCSI bus. Mac

Scsidevice

These Inspectors refer to the Small Computer System Interface devices connected to the Client computer.

Creation Methods

Key Phrase	*Form*	Description
scsidevice	*PlainGlobal*	An iterated property. It is derived from calls to the MacOS. Mac
scsidevice <integer>	*NumberedGlobal*	Returns a SCSI device with the given number. Mac
scsidevice <integer> of <scsibus>	*Numbered*	Returns a SCSI device with the given number. Mac

Key Phrase	*Form*	Description
scsidevice of <scsibus>	*Plain*	Returns a SCSI device associated with the given bus. Mac

Properties

Key Phrase	*Form*	Return Type	Description
product of <scsidevice>	*Plain*	<string>	The product string for the given SCSI device. Mac
revision of <scsidevice>	*Plain*	<string>	The revision of the SCSI device. Mac
type of <scsidevice>	*Plain*	<string>	Returns a SCSI device type, such as: DISK, TAPE, PRINTER, CPU, WORM, CDROM, SCAN, DISK, or UNKNOWN. Mac
vendor of <scsidevice>	*Plain*	<string>	Vendor string for given SCSI device. Mac

Usb

These are the Universal Serial Bus Inspectors.

Creation Methods

Key Phrase	*Form*	Description
usb	*PlainGlobal*	The Universal Serial Bus. Mac

Properties

Key Phrase	*Form*	Return Type	Description
isochronous of <usb>	*Plain*	<boolean>	The isochronous attribute of the USB installed. Taken from the Gestalt Manager. Mac
version of <usb>	*Plain*	<version>	The version of the USB installed in the system. Taken from the Gestalt Manager. Mac

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
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

Key Phrase	*Form*	Return Type	Description
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.

- `masthead path of site "BESSupport"`

  Typically returns a string such as: Macintosh HD:private:var:root:Library:Preferences:BigFix Enterprise:BES Support:__Local:Masthead.

- `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
process owner of <client>	*Plain*	<client process owner>	The name of the owner of the BigFix client. Mac
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

Key Phrase	*Form*	Return Type	Description
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
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

Key Phrase	*Form*	Return Type	Description
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
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

Key Phrase	*Form*	Return Type	Description
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

Client Process Owner

These Inspectors tell what user context the client is running in.

Creation Methods

Key Phrase	*Form*	Description
process owner of <client>	*Plain*	Owner of the agent process. Mac

Properties

Key Phrase	*Form*	Return Type	Description
<client process owner> as string	*Cast*	<string>	Owner of the agent process. Mac
long name of <client process owner>	*Plain*	<string>	The long name of the client process owner. Mac
name of <client process owner>	*Plain*	<string>	Owner of the agent process. Mac
short name of <client process owner>	*Plain*	<string>	The short name of the client process owner. Mac

Examples

- `process owner of client as string`
  - Typically returns "root".

- `long name of process owner of client`
  - OS X usernames typically have a long and a short form. For a properly installed Agent, long name of process owner of client should return the same as short name, namely 'root'.

- `name of process owner of client`
  - Should usually be root.

- `short name of process owner of client`
  - OS X usernames typically have a long and a short form. For a properly installed Agent, short name of process owner of client should return the same as long name, namely 'root'.

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
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

Key Phrase	*Form*	Return Type	Description
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

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.

Security Identifier

A Security Identifier, or SID, is a data structure that identifies user, group, and computer accounts. Every account on a network is issued a unique SID when the account is first created. Internal processes in Windows refer to an account's SID rather than the account's user or group name.

Creation Methods

Key Phrase	*Form*	Description
sid of <active directory group>	*Plain*	Returns the security identifier object corresponding to the specified Active Directory groups for the local machine. Win:8.1, Mac:8.1

Properties

Key Phrase	*Form*	Return Type	Description
<security identifier> as string	*Cast*	<string>	Returns the security identifier in string format. Win, Mac:8.1
component string of <security identifier>	*Plain*	<string>	This Windows-specific inspector returns a string formatted using the ConvertSidToStringSid windows API, discussed at: http://msdn2.microsoft.com/en-us/library/aa376399(VS.85).aspx. Win:7.0, Mac:8.1

Operators

Key phrase	Return Type	Description
<security identifier> = <security identifier>	<*boolean*>	Tests two <security identifier> (SID) values for equality using EqualSid. Win:7.0, Mac:8.1

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
local user	*PlainGlobal*	Itererates of all cached Active Directory local users. This Inspector only works in the client context when caching is enabled. Win:8.1, Mac:8.1
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
user of <logged on user>	*Plain*	Returns a user object from a 'logged on' user. This is for Active Directory expressions to bridge the gaps between user types. This retains the domain information of the logged on user within the user object where other user types might not. Win:8.1, Mac:8.1

Properties

Key Phrase	*Form*	Return Type	Description
active directory user of <user>	*Plain*	<active directory local user>	Returns an <active directory local user> object from the specified logged-on user object. This bridges the gaps between user types when using Active Directory Inspectors. It retains the domain information of the logged-on user within the user object where other user types might not. Win:8.1, Mac:8.1

Key Phrase	*Form*	Return Type	Description
attribute <string> of <user>	*Named*	<user attribute>	Returns the specified named user attribute for the given user. These attributes are gathered from the LocalHost node of Apple's OpenDirectory system. Mac:7.1
attribute of <user>	*Plain*	<user attribute>	Returns a list of attributes for the given user, logged in or not. These attributes are gathered from the LocalHost node of Apple's OpenDirectory system. Mac:7.1
home directory of <user>	*Plain*	<folder>	On a Mac, this Inspector returns the dsAttrTypeStandard:NFSHomeDirectory attribute of the specified user as a folder. • Note: On a Windows system, this inspector has a different interpretation: it returns the directory (as a string) where the user files are stored for the specified user. Mac:7.1
id of <user>	*Plain*	<string>	Returns the dsAttrTypeStandard:UniqueID attribute of the specified user. Mac:7.1
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
primary group id of <user>	*Plain*	<string>	On a Mac, this Inspector returns the dsAttrTypeStandard:PrimaryGroupID attribute for the specified user. • Note: On a Windows computer, this Inspector returns the gid as an integer, not a string. Mac:7.1

Examples

- `attributes of user "fizzle"`

  Returns a list of attributes, such as: dsAttrTypeStandard:RecordName: fizzle, dsAttrTypeStandard:UniqueID: 501, dsAttrTypeStandard:NFSHomeDirectory: /Users/fizzle...

- `names of users`

  Returns a list of all the users.

Logged On User

These Windows and Macintosh Inspectors return information about the currently logged-on user. With the advent of Terminal Services and Fast User Switching, these Inspectors are designed to iterate over all logged on users.

• Windows Note: If Terminal Services are available (NT/2000/2003/XP/Vista) and enabled, these Inspectors iterate over the active and disconnected sessions as returned by WTSEnumerateSessions. Disconnected sessions are those where a user logs on, but is currently inactive. On Vista, the non-interactive session 0 (used for services isolation) is not included. If Terminal Services aren't available, the ACLs on the security descriptor of the "winsta0" window station are examined for user logons. On Windows 9x systems, these Inspectors return the user session associated with the registry value "Current User" of "SYSTEM\CurrentControlSet\Control" if it exists. Otherwise, if a shell process process such as Explorer.exe is running, they return a single session associated with an unnamed user (which occurs when the user cancels the 9x login dialog).

Creation Methods

Key Phrase	*Form*	Description
current user	*PlainGlobal*	Returns the active, console (local) user, if logged on. Otherwise does not exist. Win:7.0, Mac:7.1, WM
logged on user	*PlainGlobal*	Returns zero or more users logged on to this computer. This Inspector iterates through all logged-on users, using Fast User Switching, Terminal Services, ACLs, and on Win 9x, the registry. Win:7.0, Mac:7.1, WM

Properties

Key Phrase	*Form*	Return Type	Description
active of <logged on user>	*Plain*	<boolean>	Returns TRUE if the specified user session is active (either as a current Fast User or an active terminal services connection). Win:7.0, Mac:7.1, WM
remote of <logged on user>	*Plain*	<boolean>	Returns TRUE if the user session is a remote terminal services connection. Win:7.0, Mac:7.1, WM
session id of <logged on user>	*Plain*	<string>	Returns the session id, which uniquely identifies a logged on user session. A logged-on user is a subclass of a user, and adding the session id uniquely identifies the session. Mac:7.1

Key Phrase	*Form*	Return Type	Description
user of <logged on user>	*Plain*	<user>	Returns a user object from a 'logged on' user. This is for Active Directory expressions to bridge the gaps between user types. This retains the domain information of the logged on user within the user object where other user types might not. Win:8.1, Mac:8.1

User Attribute

These Macintosh Inspectors provide information, such as user ID and home directory, of the specified user.

Creation Methods

Key Phrase	*Form*	Description
attribute <string> of <user>	*Named*	Returns the specified named user attribute for the given user. These attributes are gathered from the LocalHost node of Apple's OpenDirectory system. Mac:7.1
attribute of <user>	*Plain*	Returns a list of attributes for the given user, logged in or not. These attributes are gathered from the LocalHost node of Apple's OpenDirectory system. Mac:7.1

Properties

Key Phrase	*Form*	Return Type	Description
<user attribute> as string	*Cast*	<string>	Returns a list of user attributes. These can be inspected for value and key, but this Inspector concatenates them so the cast yields a string of the form "key: value". These attributes are gathered from the LocalHost node of Apple's OpenDirectory system (much like ActiveDirectory). For more information, see the Apple developer site. Mac:7.1
key of <user attribute>	*Plain*	<string>	Returns the key names of the specified user attribute, as specified by the LocalHost node of Apple's OpenDirectory system. Mac:7.1

Key Phrase	*Form*	Return Type	Description
value of <user attribute>	*Plain*	<string>	Returns the value of the specified user attribute, as specified by the LocalHost node of Apple's OpenDirectory system. Mac:7.1

Examples

- `attributes of user "fizzle"`

  Returns a list of attributes, such as: dsAttrTypeStandard:RecordName: fizzle, dsAttrTypeStandard:UniqueID: 501, dsAttrTypeStandard:NFSHomeDirectory: /Users/fizzle...

- `attribute "dsAttrTypeStandard:UniqueID" of user "fizzle"`

  Returns 501.

- `keys of attributes of user "fizzle"`

  Returns the key names of the specified attribute, such as: dsAttrTypeStandard:RecordName, dsAttrTypeStandard:UniqueID, dsAttrTypeStandard:NFSHomeDirectory.

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

Key Phrase	*Form*	Description
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
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

Key Phrase	*Form*	Return Type	Description
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
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

Key Phrase	*Form*	Return Type	Description
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 <integer> of <network>	*Numbered*	<network adapter>	Returns the nth adapter of the specified network. Mac:7.1

Key Phrase	*Form*	Return Type	Description
adapter <string> of <network>	*Named*	<network adapter>	Returns the named adapter of the specified network. Mac:7.1
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
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
link interface <integer> of <network>	*Numbered*	<network link interface>	Returns the Nth network link interface of the given network. Mac
link interface of <network>	*Plain*	<network link interface>	Returns the network link interface of the specified network. Mac

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 adapter>	*Plain*	Creates a network interface object from the specified network adapter. Mac:7.1
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 name of <network interface>	*Plain*	<string>	Returns the name of the interface family. There are about 35 of these, but the most common is AF_INET, or normal IP interface. AF_LINK is used for wifi devices and AF_INET6 is for IP v6 addresses. There are specific inspectors for AF_INET (ip interface) and AF_LINK (link interface). The family name of inspector returns this type as a string corresponding to the #define value in the header file. Mac
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
name of <network interface>	*Plain*	<string>	Returns the name of the network interface object. Mac

Key Phrase	*Form*	Return Type	Description
up of <network interface>	*Plain*	<boolean>	Returns TRUE if the specified network interface 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. Mac

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 adapter>	*Plain*	Creates an object or an object list (using the plural keyword) with all the ip interfaces of the specified network adapter. Mac:7.1
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
primary internet connection	*PlainGlobal*	This contains information about the current internet connection. Mac

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
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

Key Phrase	*Form*	Return Type	Description
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 Link Interface

The network link interface objects correspond to interfaces of type AF_LINK.

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
link interface <integer> of <network>	*Numbered*	Returns the Nth network link interface of the given network. Mac

Key Phrase	*Form*	Description
link interface of <network adapter>	*Plain*	Returns the network link interface(s) of the specified network adapter. Mac:7.1
link interface of <network>	*Plain*	Returns the network link interface(s) of the specified network. Mac

Properties

Key Phrase	*Form*	Return Type	Description
mac address of <network link interface>	*Plain*	<string>	Returns the MAC address of the specified network link interface. Mac

Examples

- `names of link interfaces of network`
  - Returns a list of the names of the network link interfaces, for example, lo0, gif0, stf0, en0, en1, fw0, etcetera.

- `mac addresses of link interfaces of network`
  - Returns a list of the MAC addresses of the network link interfaces, for example, 00-14-c8-3a-82-11, 00-16-bc-72-2c-57, 00-14-e4-26-fe-4c-14-37, etcetera.

- `mac address of link interface (whose name of it is "fw0") of network`
  - Returns the MAC address of the specified network link interface.

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 <integer> of <network>	*Numbered*	Returns the nth adapter of the specified network. Mac:7.1
adapter <string> of <network>	*Named*	Returns the named adapter of the specified network. Mac:7.1

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
broadcast address of <network adapter>	*Plain*	<ipv4 address>	Returns the broadcast address of the specified network adapter as an <ipv4 address>. Mac:7.1
broadcast support of <network adapter>	*Plain*	<boolean>	Returns TRUE if the specified network adapter supports broadcast. Mac:7.1
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
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
interface of <network adapter>	*Plain*	<network interface>	Returns a network interface object from the specified network adapter. Mac:7.1

Key Phrase	*Form*	Return Type	Description
ip interface of <network adapter>	*Plain*	<network ip interface>	Returns an object or an object list (using the plural keyword) with all the ip interfaces of the specified network adapter. Mac:7.1
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
link interface of <network adapter>	*Plain*	<network link interface>	Returns the network link interface(s) of the specified network adapter. Mac:7.1
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
point to point of <network adapter>	*Plain*	<boolean>	Returns TRUE if the specified network adapter is a point-to-point interface. Usually TRUE for dialup connections. Mac:7.1
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

Key Phrase	*Form*	Return Type	Description
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
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

Key Phrase	*Form*	Return Type	Description
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 adapter>	*Plain*	Creates an object with the ipv4 broadcast address of the specified network adapter. Mac:7.1
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

Key Phrase	*Form*	Description
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
<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

Key Phrase	*Form*	Description
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
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

Active Directory Group

These Inspectors constitute the base types that allow access to the AD group information, such as distinguished name.

Creation Methods

Key Phrase	*Form*	Description
group <string> of <active directory local computer>	*Named*	Returns the Active Directory group corresponding to the specified group of the given AD local computer. Win:8.1, Mac:8.1
group <string> of <active directory local user>	*Named*	Returns the Active Directory group corresponding to the specified group of the given AD local user. Win:8.1, Mac:8.1
group of <active directory local computer>	*Plain*	Returns a list of the active directory groups for the specified local user. Win:8.1, Mac:8.1
group of <active directory local user>	*Plain*	Returns a list of the active directory groups for the specified local computer. Win:8.1, Mac:8.1

Properties

Key Phrase	*Form*	Return Type	Description
distinguished name error message of <active directory group>	*Plain*	<string>	Returns the error message (if any) received when trying to get the distinguished name for the specified active directory group. Win:8.1, Mac:8.1
distinguished name of <active directory group>	*Plain*	<string>	Returns the distinguished name (as a string) of the specified active directory group. Win:8.1, Mac:8.1
name of <active directory group>	*Plain*	<string>	Returns the name (as strings) of the specified Active Directory group for the local machine. Win:8.1, Mac:8.1
sample time of <active directory group>	*Plain*	<time>	Returns the last sample time for the specified active directory group. Win:8.1, Mac:8.1
sid of <active directory group>	*Plain*	<security identifier>	Returns the security identifier object corresponding to the specified Active Directory groups for the local machine. Win:8.1, Mac:8.1

Active Directory Local User

These are the Inspectors for the AD local users. These include the distinguished name and groups.

Creation Methods

Key Phrase	*Form*	Description
active directory user of <user>	*Plain*	Returns an <active directory local user> object from the specified logged-on user object. This bridges the gaps between user types when using Active Directory Inspectors. It retains the domain information of the logged-on user within the user object where other user types might not. Win:8.1, Mac:8.1
local user <string> of <active directory server>	*Named*	Returns the named local user of the specified active directory server. Win:8.1, Mac:8.1
local user of <active directory server>	*Plain*	Returns the local users associated with the specified active directory server. Win:8.1, Mac:8.1
logged on user <string> of <active directory server>	*Named*	Returns the Active Directory local user object which allows inspection of AD properties for the specified currently logged in user. Win:8.1, Mac:8.1
logged on user of <active directory server>	*Plain*	Returns the Active Directory local user object which allows inspection of AD properties for the currently logged in users of the specified AD server. Win:8.1, Mac:8.1

Properties

Key Phrase	*Form*	Return Type	Description
distinguished name error message of <active directory local user>	*Plain*	<string>	Returns the error message (if any) received when trying to get the distinguished name for the specified active directory local user. Win:8.1, Mac:8.1
distinguished name of <active directory local user>	*Plain*	<string>	Returns the distinguished name (as a string) of the specified active directory local user. Win:8.1, Mac:8.1
group <string> of <active directory local user>	*Named*	<active directory group>	Returns the Active Directory group corresponding to the specified group of the given AD local user. Win:8.1, Mac:8.1

Key Phrase	*Form*	Return Type	Description
group of <active directory local user>	*Plain*	<active directory group>	Returns a list of the active directory groups for the specified local computer. Win:8.1, Mac:8.1
groups error message of <active directory local user>	*Plain*	<string>	Returns the error message (if any) received when trying to get the groups for the specified active directory local user. Win:8.1, Mac:8.1
name of <active directory local user>	*Plain*	<string>	Returns the name (as a string) of the specified Active Directory local user for the local machine. Win:8.1, Mac:8.1
sample time of <active directory local user>	*Plain*	<time>	Returns the last sample time for the specified active directory local user. Win:8.1, Mac:8.1

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

Key Phrase	*Form*	Return Type	Description
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.

Power State

These Inspectors return the state of a device, encapsulating the enumerated types used by the Client. There are two categories, system and monitor. System (computer) states include active, idle, logged off, standby, off and invalid. Monitor (display) states include on, off and invalid.

Creation Methods

Key Phrase	*Form*	Description
active state	*PlainGlobal*	Returns a power state constant signifying that the client system is active. Win:8.0, Mac:8.1

Key Phrase	*Form*	Description
idle state	*PlainGlobal*	Returns a power state constant signifying that the client system is idle. Win:8.0, Mac:8.1
invalid state	*PlainGlobal*	Returns a power state constant signifying that the client system is invalid. Win:8.0, Mac:8.1
logged off state	*PlainGlobal*	Returns a power state constant signifying that the client system is logged off. Win:8.0, Mac:8.1
monitor invalid state	*PlainGlobal*	Returns a power state constant signifying that the attached computer monitor state is invalid. Win:8.0, Mac:8.1
monitor off state	*PlainGlobal*	Returns a power state constant signifying that the attached computer monitor is off. Win:8.0, Mac:8.1
monitor on state	*PlainGlobal*	Returns a power state constant signifying that the attached computer monitor is on. Win:8.0, Mac:8.1
monitor standby state	*PlainGlobal*	Returns a power state constant signifying that the attached computer monitor is in a standby state. Win:8.0, Mac:8.1
off state	*PlainGlobal*	Returns a power state constant signifying that the client system is off. Win:8.0, Mac:8.1
standby state	*PlainGlobal*	Creates the power state corresponding to standby. Win:8.0, Mac:8.1
state of <monitor power interval>	*Plain*	Returns the state of the specified 'monitor power' interval. This state is one of the following: • on • off • invalid. Win:8.0, Mac:8.1
state of <system power interval>	*Plain*	Returns the power state associated with the specified system power interval. Win:8.0, Mac:8.1

Properties

Key Phrase	*Form*	Return Type	Description
<power state> as string	*Cast*	<string>	Casts a power state as a string type. Win:8.0, Mac:8.1

Operators

Key phrase	Return Type	Description
<power state> = <power state>	*<boolean>*	Returns TRUE if the two provided power states are equal. Win:8.0, Mac:8.1

Power History

These Inspectors retrieve information about your client computers and their displays within a tracking window (defaulting to 14 days). The information is in the form of a list of <interval, state> tuples for the system (computer) and <interval, state, monitor count> for monitors (attached displays). The first element of the list is the current state of the system. These Inspectors allow you to track computer usage for power management applications. The event lists are fetched from the client whenever 'power history' is referenced, and referencing 'system intervals of <power history>' simply iterates over the built list of intervals. Avoid referencing 'power history' multiple times in relevance as it rebuilds the list each time (increasing overhead) and may introduce inconsistency if the window slides between references.

Creation Methods

Key Phrase	*Form*	Description
power history	*PlainGlobal*	Returns the power history of the client computer. This points to historical information (the default is 14 days) about the power usage of the client computer and its attached monitor. Win:8.0, Mac:8.1

Properties

Key Phrase	*Form*	Return Type	Description
current monitor interval of <power history>	*Plain*	<monitor power interval>	Returns a monitor power interval reflecting how long the monitor has been turned on for the specified power history. Win:8.0, Mac:8.1

Key Phrase	*Form*	Return Type	Description
current system interval of <power history>	*Plain*	<system power interval>	Returns the most recent interval of the system intervals list. Win:8.0, Mac:8.1
last monitor interval in <power state> of <power history>	*Index<power state>*	<monitor power interval>	Returns the last time interval for the display monitor as logged in the power history. You must specify the power state you are interested in (either monitor on or monitor off). Win:8.1, Mac:8.1
last monitor interval in monitor off state of <power history>	*Plain*	<monitor power interval>	Returns the last time interval during which the display monitor was OFF from the specified power history. Win:8.1, Mac:8.1
last monitor interval in monitor on state of <power history>	*Plain*	<monitor power interval>	Returns the last time interval during which the display monitor was ON from the specified power history. Win:8.1, Mac:8.1
last system interval in <power state> of <power history>	*Index<power state>*	<system power interval>	Returns the power interval corresponding to the last time the computer system was in the specified power state (active, idle, logged off, off, standby) as logged in the power history. Win:8.1, Mac:8.1
last system interval in active state of <power history>	*Plain*	<system power interval>	Returns the interval corresponding to the last active state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1
last system interval in idle state of <power history>	*Plain*	<system power interval>	Returns the interval corresponding to the last idle state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1
last system interval in logged off state of <power history>	*Plain*	<system power interval>	Returns the interval corresponding to the last logged off state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1
last system interval in off state of <power history>	*Plain*	<system power interval>	Returns the interval corresponding to the last off state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1
last system interval in standby state of <power history>	*Plain*	<system power interval>	Returns the interval corresponding to the last standby state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1

Key Phrase	*Form*	Return Type	Description
monitor interval of <power history>	*Plain*	<monitor power interval>	Returns the interval corresponding to the 'monitor power' entry in the specified power history. Win:8.0, Mac:8.1
system interval of <power history>	*Plain*	<system power interval>	Returns a list of computer states and their start and end times (for example the computer was idle from Wed, 15 Sep 2010 12:30:00 -0700 to Wed, 15 Sep 2010 18:22:00 -0700, which is an interval of 5:52:00). Win:8.0, Mac:8.1

Examples

- `(state of it, start of range of it, end of range of it, length of range of it) of system intervals of power history`

  Returns a list of computer states as well as their start times, end times and elapsed times.

System Power Interval

These Inspectors return an interval or a list of intervals that contain information about the client computer. Each system interval is composed of a time range and a power state, which can include on, off, standby or hibernate.

Creation Methods

Key Phrase	*Form*	Description
current system interval of <power history>	*Plain*	Returns the most recent interval of the system intervals list. Win:8.0, Mac:8.1
last system interval in <power state> of <power history>	*Index<power state>*	Returns the power interval corresponding to the last time the computer system was in the specified power state (active, idle, logged off, off, standby) as logged in the power history. Win:8.1, Mac:8.1
last system interval in active state of <power history>	*Plain*	Returns the interval corresponding to the last active state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1
last system interval in idle state of <power history>	*Plain*	Returns the interval corresponding to the last idle state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1

Key Phrase	*Form*	Description
last system interval in logged off state of <power history>	*Plain*	Returns the interval corresponding to the last logged off state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1
last system interval in off state of <power history>	*Plain*	Returns the interval corresponding to the last off state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1
last system interval in standby state of <power history>	*Plain*	Returns the interval corresponding to the last standby state of the computer system, as logged in the given power history. Win:8.1, Mac:8.1
system interval of <power history>	*Plain*	Returns a list of computer states and their start and end times (for example the computer was idle from Wed, 15 Sep 2010 12:30:00 -0700 to Wed, 15 Sep 2010 18:22:00 -0700, which is an interval of 5:52:00). Win:8.0, Mac:8.1

Properties

Key Phrase	*Form*	Return Type	Description
range of <system power interval>	*Plain*	<time range>	Returns the time range associated with the specified 'system power' interval. This will provide you with the start and stop time for any particular state of the BES Client system. Win:8.0, Mac:8.1
state of <system power interval>	*Plain*	<power state>	Returns the power state associated with the specified system power interval. Win:8.0, Mac:8.1

Monitor Power Interval

These Inspectors return information about the monitors (displays) attached to a given computer. Each computer can have multiple monitors, and they can be either on or off. This information can be used in a power-usage study. Monitor intervals are retrieved as tuples in the form of <interval, state, monitor count>, where the interval contains the start and end time, the state (on or off) and the number of monitors that are attached to the computer.

Creation Methods

Key Phrase	*Form*	Description
current monitor interval of <power history>	*Plain*	Creates a monitor power interval reflecting how long the monitor has been turned on for the specified power history. Win:8.0, Mac:8.1
last monitor interval in <power state> of <power history>	*Index<power state>*	Returns the last time interval for the display monitor as logged in the power history. You must specify the power state you are interested in (either monitor on or monitor off). Win:8.1, Mac:8.1
last monitor interval in monitor off state of <power history>	*Plain*	Returns the last time interval during which the display monitor was OFF from the specified power history. Win:8.1, Mac:8.1
last monitor interval in monitor on state of <power history>	*Plain*	Returns the last time interval during which the display monitor was ON from the specified power history. Win:8.1, Mac:8.1
monitor interval of <power history>	*Plain*	Returns the interval corresponding to the 'monitor power' entry in the specified power history. Win:8.0, Mac:8.1

Properties

Key Phrase	*Form*	Return Type	Description
count of <monitor power interval>	*Plain*	<integer>	Returns the number of 'monitor power' intervals currently logged. Win:8.0, Mac:8.1
range of <monitor power interval>	*Plain*	<time range>	Returns the time range associated with the specified 'monitor power' interval. This will provide you with the start and stop time for any particular monitor (display) state. Win:8.0, Mac:8.1

Key Phrase	*Form*	Return Type	Description
state of <monitor power interval>	*Plain*	<power state>	Returns the state of the specified 'monitor power' interval. This state is one of the following: • on • off • invalid. Win:8.0, Mac:8.1

Miscellaneous

These Inspectors are used to fail gracefully or to provide a placeholder for Inspectors that may not exist on all operating systems.

Dummy

These Inspectors are place holders for compatability with Windows clients

Properties

Key Phrase	*Form*	Return Type	Description
state of <dummy>	*Plain*	<string>	A dummy inspector to provide compatibility with other operating systems. Mac

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
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*	mac
action <integer>	actions	<action>	<world>	*NumberedGlobal*	mac
action lock state	action lock states	<action lock state>	<world>	*PlainGlobal*	mac
active action	active actions	<action>	<world>	*PlainGlobal*	mac
active directory user of <user>	active directory users	<active directory local user>	<user>	*Plain*	mac
active of <action>	actives	<boolean>	<action>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
active of <logged on user>	actives	<boolean>	<logged on user>	*Plain*	mac
active start time of <action>	active start times	<time>	<action>	*Plain*	mac
active state	active states	<power state>	<world>	*PlainGlobal*	mac
adapter <integer> of <network>	adapters	<network adapter>	<network>	*Numbered*	mac
adapter <string> of <network>	adapters	<network adapter>	<network>	*Named*	mac
adapter of <network adapter interface>	adapters	<network adapter>	<network adapter interface>	*Plain*	mac
adapter of <network>	adapters	<network adapter>	<network>	*Plain*	mac
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*	mac
address of <network adapter>	addresses	<ipv4 address>	<network adapter>	*Plain*	mac
address of <network ip interface>	addresses	<ipv4 address>	<network ip interface>	*Plain*	mac
address of <string>	addresss	<html>	<string>	*Plain*	core
administrator <string> of <client>	administrators	<setting>	<client>	*Named*	mac
administrator of <client>	administrators	<setting>	<client>	*Plain*	mac
alias of <file>	aliases	<boolean>	<file>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
alias of <network ip interface>	aliases	<boolean>	<network ip interface>	*Plain*	mac
allocation block count of <volume>	allocation block counts	<integer>	<volume>	*Plain*	mac
allow unmentioned site of <license>	allow unmentioned sites	<boolean>	<license>	*Plain*	mac
ancestor of <filesystem object>	ancestors	<folder>	<filesystem object>	*Plain*	mac
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*	mac
any ip version	any ip versions	<ip version>	<world>	*PlainGlobal*	core
apparent registration server time	apparent registration server times	<time>	<world>	*PlainGlobal*	mac
apple extras folder of <domain>	apple extras folders	<folder>	<domain>	*Plain*	mac
apple menu items folder of <domain>	apple menu items folders	<folder>	<domain>	*Plain*	mac
application	applications	<filesystem object>	<world>	*PlainGlobal*	mac
application <string>	applications	<filesystem object>	<world>	*NamedGlobal*	mac
application of <folder>	applications	<filesystem object>	<folder>	*Plain*	mac
application support folder of <domain>	application support folders	<folder>	<domain>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
application usage summary	application usage summaries	<application usage summary>	<world>	*PlainGlobal*	mac
application usage summary <string>	application usage summaries	<application usage summary>	<world>	*NamedGlobal*	mac
applications folder of <domain>	applications folders	<folder>	<domain>	*Plain*	mac
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*	mac
array <integer> of <array>	arrays	<array>	<array>	*Numbered*	mac
array <string> of <dictionary>	arrays	<array>	<dictionary>	*Named*	mac
array <string> of <preference>	arrays	<array>	<preference>	*Named*	mac
array of <file>	arrays	<array>	<file>	*Plain*	mac
array of <osxvalue>	arrays	<array>	<osxvalue>	*Plain*	mac
assistants folder of <domain>	assistants folders	<folder>	<domain>	*Plain*	mac
attribute <string> of <user>	attributes	<user attribute>	<user>	*Named*	mac
attribute of <user>	attributes	<user attribute>	<user>	*Plain*	mac
audio folder of <domain>	audio folders	<folder>	<domain>	*Plain*	mac
audio plane of <registryroot>	audio planes	<registrynode>	<registryroot>	*Plain*	mac

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
average of <evaluation cycle>	averages	<integer>	<evaluation cycle>	*Plain*	mac
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
backup time of <filesystem object>	backup times	<time>	<filesystem object>	*Plain*	mac
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*	mac
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
boolean <integer> of <array>	booleans	<boolean>	<array>	*Numbered*	mac
boolean <string>	booleans	<boolean>	<world>	*NamedGlobal*	core
boolean <string> of <dictionary>	booleans	<boolean>	<dictionary>	*Named*	mac
boolean <string> of <preference>	booleans	<boolean>	<preference>	*Named*	mac
boolean of <osxvalue>	booleans	<boolean>	<osxvalue>	*Plain*	mac
boot time of <operating system>	boot times	<time>	<operating system>	*Plain*	mac
br	brs	<html>	<world>	*PlainGlobal*	core
br <string>	brs	<html>	<world>	*NamedGlobal*	core
brand of <client>	brands	<string>	<client>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
broadcast address of <network adapter interface>	broadcast addresses	<ipv4or6 address>	<network adapter interface>	*Plain*	mac
broadcast address of <network adapter>	broadcast addresses	<ipv4 address>	<network adapter>	*Plain*	mac
broadcast address of <network ip interface>	broadcast addresses	<ipv4 address>	<network ip interface>	*Plain*	mac
broadcast support of <network adapter interface>	broadcast supports	<boolean>	<network adapter interface>	*Plain*	mac
broadcast support of <network adapter>	broadcast supports	<boolean>	<network adapter>	*Plain*	mac
broadcast support of <network ip interface>	broadcast supports	<boolean>	<network ip interface>	*Plain*	mac
bug revision of <version>	bug revisions	<integer>	<version>	*Plain*	mac
build number of <operating system>	build numbers	<string>	<operating system>	*Plain*	mac
build of <operating system>	builds	<string>	<operating system>	*Plain*	mac
build revision of <version>	build revisions	<integer>	<version>	*Plain*	mac
bundle <string>	bundles	<bundle>	<world>	*NamedGlobal*	mac
bundle of <folder>	bundles	<bundle>	<folder>	*Plain*	mac
bundle version of <bundle>	bundle versions	<version>	<bundle>	*Plain*	mac
bundle version of <filesystem object>	bundle versions	<version>	<filesystem object>	*Plain*	mac
bundle version of <folder>	bundle versions	<version>	<folder>	*Plain*	mac
byte <integer> of <file>	bytes	<integer>	<file>	*Numbered*	mac
cache folder of <domain>	cache folders	<folder>	<domain>	*Plain*	mac
caption <string> of <html>	captions	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
carbon folder of <domain>	carbon folders	<folder>	<domain>	*Plain*	mac
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
character <integer>	characters	<string>	<world>	*NumberedGlobal*	core
character <integer> of <string>	characters	<substring>	<string>	*Numbered*	core
character of <string>	characters	<substring>	<string>	*Plain*	core
chewable items folder of <domain>	chewable items folders	<folder>	<domain>	*Plain*	mac
cidr address of <network adapter interface>	cidr addresses	<string>	<network adapter interface>	*Plain*	mac
cidr address of <network adapter>	cidr addresses	<string>	<network adapter>	*Plain*	mac
cidr address of <network ip interface>	cidr addresses	<string>	<network ip interface>	*Plain*	mac
cidr string of <network adapter interface>	cidr strings	<string>	<network adapter interface>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
cidr string of <network adapter>	cidr strings	<string>	<network adapter>	*Plain*	mac
cidr string of <network ip interface>	cidr strings	<string>	<network ip interface>	*Plain*	mac
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
classic domain	classic domains	<domain>	<world>	*PlainGlobal*	mac
classic folder of <domain>	classic folders	<folder>	<domain>	*Plain*	mac
classname of <registrynode>	classnames	<string>	<registrynode>	*Plain*	mac
client	clients	<client>	<world>	*PlainGlobal*	mac
client cryptography	client cryptographies	<client_cryptography>	<world>	*PlainGlobal*	mac
client folder of <site>	client folders	<folder>	<site>	*Plain*	mac
client license	client licenses	<license>	<world>	*PlainGlobal*	mac
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
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
color sync folder of <domain>	color sync folders	<folder>	<domain>	*Plain*	mac
colorsync profiles folder of <domain>	colorsync profiles folders	<folder>	<domain>	*Plain*	mac
common name of <license>	common names	<string>	<license>	*Plain*	mac
competition size of <selected server>	competition sizes	<integer>	<selected server>	*Plain*	mac
competition weight of <selected server>	competition weights	<integer>	<selected server>	*Plain*	mac
complete time of <action>	complete times	<time>	<action>	*Plain*	mac
component	components	<component>	<world>	*PlainGlobal*	mac
component <integer> of <site version list>	components	<integer>	<site version list>	*Numbered*	core
component folder of <domain>	component folders	<folder>	<domain>	*Plain*	mac
component string of <security identifier>	component strings	<string>	<security identifier>	*Plain*	mac
computer	computers	<computer>	<world>	*PlainGlobal*	mac
computer count of <bes product>	computer counts	<integer>	<bes product>	*Plain*	mac
computer id	computer ids	<integer>	<world>	*PlainGlobal*	mac
computer name	computer names	<string>	<world>	*PlainGlobal*	mac
concatenation <html> of <html>	concatenations	<html>	<html>	*Index<html>*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
conjunction of <boolean>	conjunctions	<boolean>	<boolean>	*Plain*	core
constrained of <action>	constraineds	<boolean>	<action>	*Plain*	mac
content of <file>	contents	<file content>	<file>	*Plain*	mac
contextual menu items folder of <domain>	contextual menu items folders	<folder>	<domain>	*Plain*	mac
control panels folder of <domain>	control panels folders	<folder>	<domain>	*Plain*	mac
control strip modules folder of <domain>	control strip modules folders	<folder>	<domain>	*Plain*	mac
controller of <action lock state>	controllers	<string>	<action lock state>	*Plain*	mac
core services folder of <domain>	core services folders	<folder>	<domain>	*Plain*	mac
count of <monitor power interval>	counts	<integer>	<monitor power interval>	*Plain*	mac
country <string>	countries	<country>	<world>	*NamedGlobal*	mac
cpu speed	cpu speeds	<integer>	<world>	*PlainGlobal*	mac
creation time of <filesystem object>	creation times	<time>	<filesystem object>	*Plain*	mac
creator of <bundle>	creators	<file signature>	<bundle>	*Plain*	mac
cryptography	cryptographies	<cryptography>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
cstring <string> of <dictionary>	cstrings	<string>	<dictionary>	*Named*	mac
cstring of <osxvalue>	cstrings	<string>	<osxvalue>	*Plain*	mac
current analysis	current analyses	<fixlet>	<world>	*PlainGlobal*	mac
current date	current dates	<date>	<world>	*PlainGlobal*	core
current day_of_month	current days_of_month	<day of month>	<world>	*PlainGlobal*	core
current day_of_week	current days_of_week	<day of week>	<world>	*PlainGlobal*	core
current day_of_year	current days_of_year	<day of year>	<world>	*PlainGlobal*	core
current monitor interval of <power history>	current monitor intervals	<monitor power interval>	<power history>	*Plain*	mac
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*	mac
current site	current sites	<site>	<world>	*PlainGlobal*	mac
current system interval of <power history>	current system intervals	<system power interval>	<power history>	*Plain*	mac
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	<logged on user>	<world>	*PlainGlobal*	mac
current user folder of <domain>	current user folders	<folder>	<domain>	*Plain*	mac
current year	current years	<year>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
custom site subscription effective date <string>	custom site subscription effective dates	<time>	<world>	*NamedGlobal*	mac
data <string> of <dictionary>	datas	<string>	<dictionary>	*Named*	mac
data fork of <file>	data forks	<datafork>	<file>	*Plain*	mac
data of <osxvalue>	datas	<string>	<osxvalue>	*Plain*	mac
date <integer> of <array>	dates	<time>	<array>	*Numbered*	mac
date <string>	dates	<date>	<world>	*NamedGlobal*	core
date <string> of <dictionary>	dates	<time>	<dictionary>	*Named*	mac
date <string> of <preference>	dates	<time>	<preference>	*Named*	mac
date <time zone> of <time>	dates	<date>	<time>	*Index<time zone>*	core
date of <bios>	dates	<string>	<bios>	*Plain*	mac
date of <osxvalue>	dates	<time>	<osxvalue>	*Plain*	mac
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
definition list <string> of <html>	definition lists	<html>	<html>	*Named*	core
definition list <string> of <string>	definition lists	<html>	<string>	*Named*	core
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*	mac
descendant of <folder>	descendants	<file>	<folder>	*Plain*	mac
desired encrypt report of <client_cryptography>	desired encrypt reports	<boolean>	<client_cryptography>	*Plain*	mac
desired fips mode of <cryptography>	desired fips modes	<boolean>	<cryptography>	*Plain*	core
desktop folder of <domain>	desktop folders	<folder>	<domain>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
developer docs folder of <domain>	developer docs folders	<folder>	<domain>	*Plain*	mac
developer folder of <domain>	developer folders	<folder>	<domain>	*Plain*	mac
developer help folder of <domain>	developer help folders	<folder>	<domain>	*Plain*	mac
devicetree plane of <registryroot>	devicetree planes	<registrynode>	<registryroot>	*Plain*	mac
dfn <string> of <html>	dfns	<html>	<html>	*Named*	core
dfn <string> of <string>	dfns	<html>	<string>	*Named*	core
dfn of <html>	dfns	<html>	<html>	*Plain*	core
dfn of <string>	dfns	<html>	<string>	*Plain*	core
dictionary <integer> of <array>	dictionaries	<dictionary>	<array>	*Numbered*	mac
dictionary <string> of <dictionary>	dictionaries	<dictionary>	<dictionary>	*Named*	mac
dictionary <string> of <preference>	dictionaries	<dictionary>	<preference>	*Named*	mac
dictionary of <file>	dictionaries	<dictionary>	<file>	*Plain*	mac
dictionary of <osxvalue>	dictionaries	<dictionary>	<osxvalue>	*Plain*	mac
dictionary of <registrynode>	dictionaries	<dictionary>	<registrynode>	*Plain*	mac
dictionary of <registryroot>	dictionaries	<dictionary>	<registryroot>	*Plain*	mac
direct object type of <property>	direct object types	<type>	<property>	*Plain*	core
directory count of <volume>	directory counts	<integer>	<volume>	*Plain*	mac
disabled control panels folder of <domain>	disabled control panels folders	<folder>	<domain>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
disabled extensions folder of <domain>	disabled extensions folders	<folder>	<domain>	*Plain*	mac
disabled shutdown items folder of <domain>	disabled shutdown items folders	<folder>	<domain>	*Plain*	mac
disabled startup items folder of <domain>	disabled startup items folders	<folder>	<domain>	*Plain*	mac
disabled system extensions folder of <domain>	disabled system extensions folders	<folder>	<domain>	*Plain*	mac
disjunction of <boolean>	disjunctions	<boolean>	<boolean>	*Plain*	core
distance of <selected server>	distances	<integer range>	<selected server>	*Plain*	mac
distinguished name error message of <active directory group>	distinguished name error messages	<string>	<active directory group>	*Plain*	mac
distinguished name error message of <active directory local computer>	distinguished name error messages	<string>	<active directory local computer>	*Plain*	mac
distinguished name error message of <active directory local user>	distinguished name error messages	<string>	<active directory local user>	*Plain*	mac
distinguished name of <active directory group>	distinguished names	<string>	<active directory group>	*Plain*	mac
distinguished name of <active directory local computer>	distinguished names	<string>	<active directory local computer>	*Plain*	mac
distinguished name of <active directory local user>	distinguished names	<string>	<active directory local user>	*Plain*	mac
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
dns name	dns names	<string>	<world>	*PlainGlobal*	mac
documentation folder of <domain>	documentation folders	<folder>	<domain>	*Plain*	mac
documents folder of <domain>	documents folders	<folder>	<domain>	*Plain*	mac
domain library folder of <domain>	domain library folders	<folder>	<domain>	*Plain*	mac
domain top folder of <domain>	domain top folders	<folder>	<domain>	*Plain*	mac
download file <string>	download files	<file>	<world>	*NamedGlobal*	mac
download path <string>	download paths	<string>	<world>	*NamedGlobal*	mac
drive	drives	<volume>	<world>	*PlainGlobal*	mac
drive <integer>	drives	<volume>	<world>	*NumberedGlobal*	mac
drive <string>	drives	<volume>	<world>	*NamedGlobal*	mac
drive of <file>	drives	<volume>	<file>	*Plain*	mac
drive of <folder>	drives	<volume>	<folder>	*Plain*	mac
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*	mac
effective date of <setting>	effective dates	<time>	<setting>	*Plain*	mac
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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*	mac
enabled of <setting>	enableds	<boolean>	<setting>	*Plain*	mac
encrypt report failure message of <client_cryptography>	encrypt report failure messages	<string>	<client_cryptography>	*Plain*	mac
encrypt report of <client_cryptography>	encrypt reports	<boolean>	<client_cryptography>	*Plain*	mac
encryption certificate of <license>	encryption certificates	<x509 certificate>	<license>	*Plain*	mac
end of <substring>	ends	<string position>	<substring>	*Plain*	core
end of <time range>	ends	<time>	<time range>	*Plain*	core
entry of <dictionary>	entries	<dictionaryentry>	<dictionary>	*Plain*	mac
environment	environments	<environment>	<world>	*PlainGlobal*	mac
error <string>	errors	<undefined>	<world>	*NamedGlobal*	core
evaluation of <license>	evaluations	<boolean>	<license>	*Plain*	mac
evaluationcycle of <client>	evaluationcycles	<evaluation cycle>	<client>	*Plain*	mac
exit code of <action>	exit codes	<integer>	<action>	*Plain*	mac
expiration date of <action lock state>	expiration dates	<time>	<action lock state>	*Plain*	mac
expiration date of <bes product>	expiration dates	<date>	<bes product>	*Plain*	mac
expiration date of <license>	expiration dates	<time>	<license>	*Plain*	mac
expiration state of <license>	expiration states	<string>	<license>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
extensions folder of <domain>	extensions folders	<folder>	<domain>	*Plain*	mac
extrema of <date>	extremas	<(date, date)>	<date>	*Plain*	core
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 <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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
false	falses	<boolean>	<world>	*PlainGlobal*	core
family name of <network interface>	family names	<string>	<network interface>	*Plain*	mac
family name of <processor>	family names	<string>	<processor>	*Plain*	mac
family of <network interface>	families	<integer>	<network interface>	*Plain*	mac
favorites folder of <domain>	favorites folders	<folder>	<domain>	*Plain*	mac
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
file <string>	files	<file>	<world>	*NamedGlobal*	mac
file <string> of <folder>	files	<file>	<folder>	*Named*	mac
file count of <volume>	file counts	<integer>	<volume>	*Plain*	mac
file ending in <string> of <folder>	files ending in	<file>	<folder>	*Named*	mac
file of <folder>	files	<file>	<folder>	*Plain*	mac
file signature <string>	file signatures	<file signature>	<world>	*NamedGlobal*	mac
file type <string>	file types	<file type>	<world>	*NamedGlobal*	mac
filesystem	filesystems	<volume>	<world>	*PlainGlobal*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
filesystem <integer>	filesystems	<volume>	<world>	*NumberedGlobal*	mac
filesystem <string>	filesystems	<volume>	<world>	*NamedGlobal*	mac
filesystem of <file>	filesystems	<volume>	<file>	*Plain*	mac
filesystem of <folder>	filesystems	<volume>	<folder>	*Plain*	mac
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*	mac
find file <string> of <folder>	find files	<file>	<folder>	*Named*	mac
find folder <string> of <folder>	find folders	<folder>	<folder>	*Named*	mac
find item <string> of <folder>	find items	<filesystem object>	<folder>	*Named*	mac
finite of <floating point>	finites	<boolean>	<floating point>	*Plain*	core
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*	mac
firewire plane of <registryroot>	firewire planes	<registrynode>	<registryroot>	*Plain*	mac
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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*	mac
first start time of <application usage summary>	first start times	<time>	<application usage summary>	*Plain*	mac
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*	mac
flag of <volume>	flags	<integer>	<volume>	*Plain*	mac
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*	mac
folder <string> of <folder>	folders	<folder>	<folder>	*Named*	mac
folder ending in <string> of <folder>	folders ending in	<folder>	<folder>	*Named*	mac
folder of <folder>	folders	<folder>	<folder>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
following text of <string position>	following texts	<substring>	<string position>	*Plain*	core
following text of <substring>	following texts	<substring>	<substring>	*Plain*	core
fonts folder of <domain>	fonts folders	<folder>	<domain>	*Plain*	mac
framework <string> of <domain>	frameworks	<folder>	<domain>	*Named*	mac
framework folder of <domain>	framework folders	<folder>	<domain>	*Plain*	mac
free percent of <volume>	free percents	<integer>	<volume>	*Plain*	mac
free space of <volume>	free spaces	<integer>	<volume>	*Plain*	mac
friday	fridays	<day of week>	<world>	*PlainGlobal*	core
friendly name of <network adapter>	friendly names	<string>	<network adapter>	*Plain*	mac
full gateway address of <selected server>	full gateway addresses	<ipv4or6 address>	<selected server>	*Plain*	mac
full of <power level>	fulls	<boolean>	<power level>	*Plain*	mac
gateway address <integer> of <selected server>	gateway addresses	<ipv4or6 address>	<selected server>	*Numbered*	mac
gateway address of <selected server>	gateway addresses	<ipv4or6 address>	<selected server>	*Plain*	mac
gather schedule authority of <site>	gather schedule authoritys	<string>	<site>	*Plain*	mac
gather schedule time interval of <site>	gather schedule time intervals	<time interval>	<site>	*Plain*	mac
gather url of <license>	gather urls	<string>	<license>	*Plain*	mac
gestalt <string>	gestalts	<integer>	<world>	*NamedGlobal*	mac
ghz	ghzs	<hertz>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
global dictionary of <bundle>	global dictionaries	<dictionary>	<bundle>	*Plain*	mac
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*	mac
group <string> of <active directory local computer>	groups	<active directory group>	<active directory local computer>	*Named*	mac
group <string> of <active directory local user>	groups	<active directory group>	<active directory local user>	*Named*	mac
group leader of <action>	group leaders	<boolean>	<action>	*Plain*	mac
group of <active directory local computer>	groups	<active directory group>	<active directory local computer>	*Plain*	mac
group of <active directory local user>	groups	<active directory group>	<active directory local user>	*Plain*	mac
groups error message of <active directory local computer>	groups error messages	<string>	<active directory local computer>	*Plain*	mac
groups error message of <active directory local user>	groups error messages	<string>	<active directory local user>	*Plain*	mac
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
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*	mac
header of <fixlet>	headers	<fixlet_header>	<fixlet>	*Plain*	mac
help folder of <domain>	help folders	<folder>	<domain>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
hfs file <string>	hfs files	<file>	<world>	*NamedGlobal*	mac
hfs folder <string>	hfs folders	<folder>	<world>	*NamedGlobal*	mac
hfs item <string>	hfs items	<filesystem object>	<world>	*NamedGlobal*	mac
hfs path of <filesystem object>	hfs paths	<string>	<filesystem object>	*Plain*	mac
hfs relative item <string> of <folder>	hfs relative items	<filesystem object>	<folder>	*Named*	mac
home directory of <user>	home directories	<folder>	<user>	*Plain*	mac
host name of <root server>	host names	<string>	<root server>	*Plain*	mac
hostname	hostnames	<string>	<world>	*PlainGlobal*	mac
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
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
html tag <string> of <string>	html tags	<html>	<string>	*Named*	core
hz	hzs	<hertz>	<world>	*PlainGlobal*	core
id of <action>	ids	<integer>	<action>	*Plain*	mac
id of <fixlet>	ids	<integer>	<fixlet>	*Plain*	mac
id of <process>	ids	<integer>	<process>	*Plain*	mac
id of <root server>	ids	<integer>	<root server>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
id of <site group>	ids	<integer>	<site group>	*Plain*	mac
id of <user>	ids	<string>	<user>	*Plain*	mac
idle state	idle states	<power state>	<world>	*PlainGlobal*	mac
index type of <property>	index types	<type>	<property>	*Plain*	core
infinite of <floating point>	infinites	<boolean>	<floating point>	*Plain*	core
info of <component>	infos	<string>	<component>	*Plain*	mac
init date of <volume>	init dates	<time>	<volume>	*Plain*	mac
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
instance of <application usage summary>	instances	<application usage summary instance>	<application usage summary>	*Plain*	mac
integer <integer>	integers	<integer>	<world>	*NumberedGlobal*	core
integer <integer> of <array>	integers	<integer>	<array>	*Numbered*	mac
integer <string>	integers	<integer>	<world>	*NamedGlobal*	core
integer <string> of <dictionary>	integers	<integer>	<dictionary>	*Named*	mac
integer <string> of <preference>	integers	<integer>	<preference>	*Named*	mac
integer ceiling of <floating point>	integer ceilings	<integer>	<floating point>	*Plain*	core
integer floor of <floating point>	integer floors	<integer>	<floating point>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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 of <osxvalue>	integers	<integer>	<osxvalue>	*Plain*	mac
integer to <integer>	integers to	<integer>	<world>	*NumberedGlobal*	core
interface <integer> of <network>	interfaces	<network interface>	<network>	*Numbered*	mac
interface of <network adapter>	interfaces	<network interface>	<network adapter>	*Plain*	mac
interface of <network>	interfaces	<network interface>	<network>	*Plain*	mac
internet plugins folder	internet plugins folders	<folder>	<world>	*PlainGlobal*	mac
internet plugins folder of <domain>	internet plugins folders	<folder>	<domain>	*Plain*	mac
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 state	invalid states	<power state>	<world>	*PlainGlobal*	mac
iokit registry	iokit registries	<registryroot>	<world>	*PlainGlobal*	mac
ip address of <selected server>	ip addresses	<ipv4or6 address>	<selected server>	*Plain*	mac
ip interface <integer> of <network>	ip interfaces	<network ip interface>	<network>	*Numbered*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
ip interface of <network adapter>	ip interfaces	<network ip interface>	<network adapter>	*Plain*	mac
ip interface of <network>	ip interfaces	<network ip interface>	<network>	*Plain*	mac
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*	mac
ipv4 interface of <network>	ipv4 interfaces	<network adapter interface>	<network>	*Plain*	mac
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*	mac
ipv4or6 interface of <network>	ipv4or6 interfaces	<network adapter interface>	<network>	*Plain*	mac
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*	mac
ipv6 interface of <network>	ipv6 interfaces	<network adapter interface>	<network>	*Plain*	mac
isochronous of <usb>	isochronouses	<boolean>	<usb>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
iss download folder	download folders	<folder>	<world>	*PlainGlobal*	mac
iss download folder of <domain>	iss download folders	<folder>	<domain>	*Plain*	mac
italic <string> of <html>	italics	<html>	<html>	*Named*	core
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
item <string>	items	<filesystem object>	<world>	*NamedGlobal*	mac
item <string> of <folder>	items	<filesystem object>	<folder>	*Named*	mac
item ending in <string> of <folder>	items ending in	<filesystem object>	<folder>	*Named*	mac
item of <folder>	items	<filesystem object>	<folder>	*Plain*	mac
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
kernel extensions folder of <domain>	kernel extensions folders	<folder>	<domain>	*Plain*	mac
key <string> of <file section>	keys	<string>	<file section>	*Named*	mac
key <string> of <file>	keys	<string>	<file>	*Named*	mac
key of <dictionary>	keys	<string>	<dictionary>	*Plain*	mac
key of <dictionaryentry>	keys	<string>	<dictionaryentry>	*Plain*	mac
key of <user attribute>	keys	<string>	<user attribute>	*Plain*	mac
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*	mac
last gather time of <site>	last gather times	<time>	<site>	*Plain*	mac
last monitor interval in <power state> of <power history>	last monitor intervals in	<monitor power interval>	<power history>	*Index<power state>*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
last monitor interval in monitor off state of <power history>	last monitor intervals in monitor off state	<monitor power interval>	<power history>	*Plain*	mac
last monitor interval in monitor on state of <power history>	last monitor intervals in monitor on state	<monitor power interval>	<power history>	*Plain*	mac
last relay select time	last relay select times	<time>	<world>	*PlainGlobal*	mac
last start time of <application usage summary instance>	last start times	<time>	<application usage summary instance>	*Plain*	mac
last start time of <application usage summary>	last start times	<time>	<application usage summary>	*Plain*	mac
last system interval in <power state> of <power history>	last system intervals in	<system power interval>	<power history>	*Index<power state>*	mac
last system interval in active state of <power history>	last system intervals in active state	<system power interval>	<power history>	*Plain*	mac
last system interval in idle state of <power history>	last system intervals in idle state	<system power interval>	<power history>	*Plain*	mac
last system interval in logged off state of <power history>	last system intervals in logged off state	<system power interval>	<power history>	*Plain*	mac
last system interval in off state of <power history>	last system intervals in off state	<system power interval>	<power history>	*Plain*	mac
last system interval in standby state of <power history>	last system intervals in standby state	<system power interval>	<power history>	*Plain*	mac
last time seen of <application usage summary instance>	last times seen	<time>	<application usage summary instance>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
last time seen of <application usage summary>	last times seen	<time>	<application usage summary>	*Plain*	mac
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
length of <datafork>	lengths	<integer>	<datafork>	*Plain*	mac
length of <file>	lengths	<integer>	<file>	*Plain*	mac
length of <month and year>	lengths	<time interval>	<month and year>	*Plain*	core
length of <resfork>	lengths	<integer>	<resfork>	*Plain*	mac
length of <rope>	lengths	<integer>	<rope>	*Plain*	core
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
li of <string>	lis	<html>	<string>	*Plain*	core
line <integer> of <file>	lines	<file line>	<file>	*Numbered*	mac
line containing <string> of <file>	lines containing	<file line>	<file>	*Named*	mac
line number of <file line>	line numbers	<integer>	<file line>	*Plain*	mac
line of <file>	lines	<file line>	<file>	*Plain*	mac
line starting with <string> of <file>	lines starting with	<file line>	<file>	*Named*	mac
link <string> of <html>	links	<html>	<html>	*Named*	core
link <string> of <string>	links	<html>	<string>	*Named*	core
link interface <integer> of <network>	link interfaces	<network link interface>	<network>	*Numbered*	mac
link interface of <network adapter>	link interfaces	<network link interface>	<network adapter>	*Plain*	mac
link interface of <network>	link interfaces	<network link interface>	<network>	*Plain*	mac
link of <html>	links	<html>	<html>	*Plain*	core
link of <string>	links	<html>	<string>	*Plain*	core
local computer of <active directory server>	local computers	<active directory local computer>	<active directory server>	*Plain*	mac
local dictionary of <bundle>	local dictionaries	<dictionary>	<bundle>	*Plain*	mac
local domain	local domains	<domain>	<world>	*PlainGlobal*	mac
local time <string>	local times	<time>	<world>	*NamedGlobal*	core
local time zone	local time zones	<time zone>	<world>	*PlainGlobal*	core
local user	local users	<user>	<world>	*PlainGlobal*	mac
local user <string> of <active directory server>	local users	<active directory local user>	<active directory server>	*Named*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
local user of <active directory server>	local users	<active directory local user>	<active directory server>	*Plain*	mac
locales folder of <domain>	locales folders	<folder>	<domain>	*Plain*	mac
location manager modules folder of <domain>	location manager modules folders	<folder>	<domain>	*Plain*	mac
location manager preferences folder of <domain>	location manager preferences folders	<folder>	<domain>	*Plain*	mac
locations folder of <domain>	locations folders	<folder>	<domain>	*Plain*	mac
lock string of <action lock state>	lock strings	<string>	<action lock state>	*Plain*	mac
locked of <action lock state>	lockeds	<boolean>	<action lock state>	*Plain*	mac
locked of <file>	lockeds	<boolean>	<file>	*Plain*	mac
logged off state	logged off states	<power state>	<world>	*PlainGlobal*	mac
logged on user	logged on users	<logged on user>	<world>	*PlainGlobal*	mac
logged on user <string> of <active directory server>	logged on users	<active directory local user>	<active directory server>	*Named*	mac
logged on user of <active directory server>	logged on users	<active directory local user>	<active directory server>	*Plain*	mac
logical ram	logical rams	<integer>	<world>	*PlainGlobal*	mac
long name of <client process owner>	long names	<string>	<client process owner>	*Plain*	mac
loopback of <network adapter interface>	loopbacks	<boolean>	<network adapter interface>	*Plain*	mac
loopback of <network adapter>	loopbacks	<boolean>	<network adapter>	*Plain*	mac
loopback of <network ip interface>	loopbacks	<boolean>	<network ip interface>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
low of <power level>	lows	<boolean>	<power level>	*Plain*	mac
mac address of <network adapter interface>	mac addresses	<string>	<network adapter interface>	*Plain*	mac
mac address of <network adapter>	mac addresses	<string>	<network adapter>	*Plain*	mac
mac address of <network ip interface>	mac addresses	<string>	<network ip interface>	*Plain*	mac
mac address of <network link interface>	mac addresses	<string>	<network link interface>	*Plain*	mac
mac of <operating system>	macs	<boolean>	<operating system>	*Plain*	mac
machine name	machine names	<string>	<world>	*PlainGlobal*	mac
macos read me folder of <domain>	macos read me folders	<folder>	<domain>	*Plain*	mac
main processor	main processors	<processor>	<world>	*PlainGlobal*	mac
major revision of <version>	major revisions	<integer>	<version>	*Plain*	mac
maker of <component>	makers	<string>	<component>	*Plain*	mac
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*	mac
match <regular expression> of <string>	matches	<regular expression match>	<string>	*Index<regular expression>*	regx
maximum of <date>	maxima	<date>	<date>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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*	mac
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
maximum of <month>	maxima	<month>	<month>	*Plain*	core
maximum of <number of months>	maxima	<number of months>	<number of months>	*Plain*	core
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
maximum seat count of <license>	maximum seat counts	<integer>	<license>	*Plain*	mac
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
mean of <floating point>	means	<floating point>	<floating point>	*Plain*	core
mean of <integer>	means	<floating point>	<integer>	*Plain*	core
member of <site group>	members	<boolean>	<site group>	*Plain*	mac
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
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 revision of <version>	minor revisions	<integer>	<version>	*Plain*	mac
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
modem scripts folder of <domain>	modem scripts folders	<folder>	<domain>	*Plain*	mac
modification time of <filesystem object>	modification times	<time>	<filesystem object>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
modification time of <volume>	modification times	<time>	<volume>	*Plain*	mac
monday	mondays	<day of week>	<world>	*PlainGlobal*	core
monitor interval of <power history>	monitor intervals	<monitor power interval>	<power history>	*Plain*	mac
monitor invalid state	monitor invalid states	<power state>	<world>	*PlainGlobal*	mac
monitor off state	monitor off states	<power state>	<world>	*PlainGlobal*	mac
monitor on state	monitor on states	<power state>	<world>	*PlainGlobal*	mac
monitor standby state	monitor standby states	<power state>	<world>	*PlainGlobal*	mac
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*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
multicast support of <network adapter>	multicast supports	<boolean>	<network adapter>	*Plain*	mac
multicast support of <network ip interface>	multicast supports	<boolean>	<network ip interface>	*Plain*	mac
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
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 <site version list with multiplicity>	multiplicities	<integer>	<site version list with multiplicity>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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 <active directory group>	names	<string>	<active directory group>	*Plain*	mac
name of <active directory local user>	names	<string>	<active directory local user>	*Plain*	mac
name of <application usage summary instance>	names	<string>	<application usage summary instance>	*Plain*	mac
name of <application usage summary>	names	<string>	<application usage summary>	*Plain*	mac
name of <bes product>	names	<string>	<bes product>	*Plain*	mac
name of <binary operator>	names	<string>	<binary operator>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
name of <cast>	names	<string>	<cast>	*Plain*	core
name of <client process owner>	names	<string>	<client process owner>	*Plain*	mac
name of <component>	names	<string>	<component>	*Plain*	mac
name of <computer>	names	<string>	<computer>	*Plain*	mac
name of <environment variable>	names	<string>	<environment variable>	*Plain*	mac
name of <filesystem object>	names	<string>	<filesystem object>	*Plain*	mac
name of <fixlet_header>	names	<string>	<fixlet_header>	*Plain*	mac
name of <network adapter>	names	<string>	<network adapter>	*Plain*	mac
name of <network interface>	names	<string>	<network interface>	*Plain*	mac
name of <network ip interface>	names	<string>	<network ip interface>	*Plain*	mac
name of <operating system>	names	<string>	<operating system>	*Plain*	mac
name of <process>	names	<string>	<process>	*Plain*	mac
name of <registrynode>	names	<string>	<registrynode>	*Plain*	mac
name of <selected server>	names	<string>	<selected server>	*Plain*	mac
name of <setting>	names	<string>	<setting>	*Plain*	mac
name of <site>	names	<string>	<site>	*Plain*	mac
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*	mac
name of <volume>	names	<string>	<volume>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
nan of <floating point>	nans	<boolean>	<floating point>	*Plain*	core
network	networks	<network>	<world>	*PlainGlobal*	mac
network domain	network domains	<domain>	<world>	*PlainGlobal*	mac
next line of <file line>	next lines	<file line>	<file line>	*Plain*	mac
node <string> of <registrynode>	nodes	<registrynode>	<registrynode>	*Named*	mac
node <string> of <registryroot>	nodes	<registrynode>	<registryroot>	*Named*	mac
node of <registrynode>	nodes	<registrynode>	<registrynode>	*Plain*	mac
non windows server count of <bes product>	non windows server counts	<integer>	<bes product>	*Plain*	mac
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*	mac
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
nubus map	nubus maps	<integer>	<world>	*PlainGlobal*	mac
numeric value of <string>	numeric values	<integer>	<string>	*Plain*	core
october	octobers	<month>	<world>	*PlainGlobal*	core
october <integer>	octobers	<day of year>	<world>	*NumberedGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
october <integer> of <integer>	octobers	<date>	<integer>	*Numbered*	core
october of <integer>	octobers	<month and year>	<integer>	*Plain*	core
off state	off states	<power state>	<world>	*PlainGlobal*	mac
offer accepted of <action>	offer accepteds	<boolean>	<action>	*Plain*	mac
offer of <action>	offers	<boolean>	<action>	*Plain*	mac
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
on appropriate disk domain	on appropriate disk domains	<domain>	<world>	*PlainGlobal*	mac
on system disk domain	on system disk domains	<domain>	<world>	*PlainGlobal*	mac
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*	mac
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*	mac
origin fixlet id of <action>	origin fixlet ids	<integer>	<action>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
parameter <string>	parameters	<string>	<world>	*NamedGlobal*	mac
parameter <string> of <action>	parameters	<string>	<action>	*Named*	mac
parent folder of <filesystem object>	parent folders	<folder>	<filesystem object>	*Plain*	mac
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
path of <registrynode>	paths	<string>	<registrynode>	*Plain*	mac
pathname of <filesystem object>	pathnames	<string>	<filesystem object>	*Plain*	mac
pending login	pending logins	<boolean>	<world>	*PlainGlobal*	mac
pending login of <action>	pending logins	<boolean>	<action>	*Plain*	mac
pending of <action>	pendings	<boolean>	<action>	*Plain*	mac
pending restart	pending restarts	<boolean>	<world>	*PlainGlobal*	mac
pending restart <string>	pending restarts	<boolean>	<world>	*NamedGlobal*	mac
pending restart name	pending restart names	<string>	<world>	*PlainGlobal*	mac
pending restart of <action>	pending restarts	<boolean>	<action>	*Plain*	mac
pending time of <action>	pending times	<time>	<action>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
physical ram	physical rams	<integer>	<world>	*PlainGlobal*	mac
pid of <process>	pids	<integer>	<process>	*Plain*	mac
plugged of <power level>	pluggeds	<boolean>	<power level>	*Plain*	mac
plural name of <property>	plural names	<string>	<property>	*Plain*	core
point to point of <network adapter interface>	point to points	<boolean>	<network adapter interface>	*Plain*	mac
point to point of <network adapter>	point to points	<boolean>	<network adapter>	*Plain*	mac
point to point of <network ip interface>	point to points	<boolean>	<network ip interface>	*Plain*	mac
port number of <selected server>	port numbers	<integer>	<selected server>	*Plain*	mac
position <integer> of <string>	positions	<string position>	<string>	*Numbered*	core
position of <string>	positions	<string position>	<string>	*Plain*	core
posix file <string>	posix files	<file>	<world>	*NamedGlobal*	mac
posix folder <string>	posix folders	<folder>	<world>	*NamedGlobal*	mac
posix item <string>	posix items	<filesystem object>	<world>	*NamedGlobal*	mac
posix path of <filesystem object>	posix paths	<string>	<filesystem object>	*Plain*	mac
posix relative item <string> of <folder>	posix relative items	<filesystem object>	<folder>	*Named*	mac
power history	power histories	<power history>	<world>	*PlainGlobal*	mac
power level	power levels	<power level>	<world>	*PlainGlobal*	mac
power plane of <registryroot>	power planes	<registrynode>	<registryroot>	*Plain*	mac
powerpc	powerpcs	<boolean>	<world>	*PlainGlobal*	mac
pre <string> of <html>	pres	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
preference <string>	preferences	<preference>	<world>	*NamedGlobal*	mac
preferences folder of <domain>	preferences folders	<folder>	<domain>	*Plain*	mac
previous line of <file line>	previous lines	<file line>	<file line>	*Plain*	mac
primary group id of <user>	primary group ids	<string>	<user>	*Plain*	mac
primary internet connection	primary internet connections	<network ip interface>	<world>	*PlainGlobal*	mac
printer descriptions folder of <domain>	printer descriptions folders	<folder>	<domain>	*Plain*	mac
printer drivers folder of <domain>	printer drivers folders	<folder>	<domain>	*Plain*	mac
printers folder of <domain>	printers folders	<folder>	<domain>	*Plain*	mac
printmonitor documents folder of <domain>	printmonitor documents folders	<folder>	<domain>	*Plain*	mac
priority of <selected server>	priorities	<integer>	<selected server>	*Plain*	mac
private framework folder of <domain>	private framework folders	<folder>	<domain>	*Plain*	mac
process	processes	<process>	<world>	*PlainGlobal*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
process <integer>	processes	<process>	<world>	*NumberedGlobal*	mac
process id of <process>	process ids	<integer>	<process>	*Plain*	mac
process owner of <client>	process owners	<client process owner>	<client>	*Plain*	mac
processor	processors	<processor>	<world>	*PlainGlobal*	mac
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*	mac
product of <scsidevice>	products	<string>	<scsidevice>	*Plain*	mac
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
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
quickdraw version	quickdraw versions	<version>	<world>	*PlainGlobal*	mac
quicktime folder of <domain>	quicktime folders	<folder>	<domain>	*Plain*	mac
ram	rams	<ram>	<world>	*PlainGlobal*	mac
range after <time> of <time range>	ranges after	<time range>	<time range>	*Index<time>*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
range before <time> of <time range>	ranges before	<time range>	<time range>	*Index<time>*	core
range of <monitor power interval>	ranges	<time range>	<monitor power interval>	*Plain*	mac
range of <system power interval>	ranges	<time range>	<system power interval>	*Plain*	mac
receipts folder of <domain>	receipts folders	<folder>	<domain>	*Plain*	mac
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*	mac
registration address of <client>	registration addresses	<ipv4or6 address>	<client>	*Plain*	mac
registration cidr address of <client>	registration cidr addresses	<string>	<client>	*Plain*	mac
registration mac address of <client>	registration mac addresses	<string>	<client>	*Plain*	mac
registration subnet address of <client>	registration subnet addresses	<ipv4or6 address>	<client>	*Plain*	mac
regular expression <string>	regular expressions	<regular expression>	<world>	*NamedGlobal*	regx
relative file <string> of <folder>	relative files	<file>	<folder>	*Named*	mac
relative folder <string> of <folder>	relative folders	<folder>	<folder>	*Named*	mac
relative hfs file <string> of <folder>	relative hfs files	<file>	<folder>	*Named*	mac
relative hfs folder <string> of <folder>	relative hfs folders	<folder>	<folder>	*Named*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
relative item <string> of <folder>	relative items	<filesystem object>	<folder>	*Named*	mac
relative posix file <string> of <folder>	relative posix files	<file>	<folder>	*Named*	mac
relative posix folder <string> of <folder>	relative posix folders	<folder>	<folder>	*Named*	mac
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
relevance of <fixlet>	relevances	<boolean>	<fixlet>	*Plain*	mac
relevant fixlet of <site>	relevant fixlets	<fixlet>	<site>	*Plain*	mac
relevant offer action of <site>	relevant offer actions	<action>	<site>	*Plain*	mac
remote of <logged on user>	remotes	<boolean>	<logged on user>	*Plain*	mac
resource fork of <file>	resource forks	<resfork>	<file>	*Plain*	mac
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
revision of <scsidevice>	revisions	<string>	<scsidevice>	*Plain*	mac
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
rom version	rom versions	<version>	<world>	*PlainGlobal*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
root server	root servers	<root server>	<world>	*PlainGlobal*	mac
rope <string>	ropes	<rope>	<world>	*NamedGlobal*	core
running of <application usage summary>	runnings	<boolean>	<application usage summary>	*Plain*	mac
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
sample time of <active directory group>	sample times	<time>	<active directory group>	*Plain*	mac
sample time of <active directory local computer>	sample times	<time>	<active directory local computer>	*Plain*	mac
sample time of <active directory local user>	sample times	<time>	<active directory local user>	*Plain*	mac
saturday	saturdays	<day of week>	<world>	*PlainGlobal*	core
scripting additions folder of <domain>	scripting additions folders	<folder>	<domain>	*Plain*	mac
scsibus	scsibuses	<scsibus>	<world>	*PlainGlobal*	mac
scsibus <integer>	scsibuses	<scsibus>	<world>	*NumberedGlobal*	mac
scsidevice	scsidevices	<scsidevice>	<world>	*PlainGlobal*	mac
scsidevice <integer>	scsidevices	<scsidevice>	<world>	*NumberedGlobal*	mac
scsidevice <integer> of <scsibus>	scsidevices	<scsidevice>	<scsibus>	*Numbered*	mac
scsidevice of <scsibus>	scsidevices	<scsidevice>	<scsibus>	*Plain*	mac
seat count state of <license>	seat count states	<string>	<license>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
seat of <license>	seats	<integer>	<license>	*Plain*	mac
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*	mac
selected server	selected servers	<selected server>	<world>	*PlainGlobal*	mac
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 plane of <registryroot>	service planes	<registrynode>	<registryroot>	*Plain*	mac
session id of <logged on user>	session ids	<string>	<logged on user>	*Plain*	mac
set of <integer>	sets	<integer set>	<integer>	*Plain*	core
set of <string>	sets	<string set>	<string>	*Plain*	core
setting <string> of <client>	settings	<setting>	<client>	*Named*	mac
setting <string> of <site>	settings	<setting>	<site>	*Named*	mac
setting of <client>	settings	<setting>	<client>	*Plain*	mac
setting of <site>	settings	<setting>	<site>	*Plain*	mac
sha1 of <file>	sha1s	<string>	<file>	*Plain*	mac
sha1 of <x509 certificate>	sha1s	<string>	<x509 certificate>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
shared folder of <domain>	shared folders	<folder>	<domain>	*Plain*	mac
shared libraries folder of <domain>	shared libraries folders	<folder>	<domain>	*Plain*	mac
short name of <client process owner>	short names	<string>	<client process owner>	*Plain*	mac
shutdown items folder of <domain>	shutdown items folders	<folder>	<domain>	*Plain*	mac
sibling file <string> of <filesystem object>	sibling files	<file>	<filesystem object>	*Named*	mac
sibling folder <string> of <filesystem object>	sibling folders	<folder>	<filesystem object>	*Named*	mac
sibling item <string> of <filesystem object>	sibling items	<filesystem object>	<filesystem object>	*Named*	mac
sid of <active directory group>	sids	<security identifier>	<active directory group>	*Plain*	mac
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
site	sites	<site>	<world>	*PlainGlobal*	mac
site <string>	sites	<site>	<world>	*NamedGlobal*	mac
site number of <license>	site numbers	<integer>	<license>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
site of <fixlet>	sites	<site>	<fixlet>	*Plain*	mac
site tag of <site>	site tags	<string>	<site>	*Plain*	mac
site url of <bes product>	site urls	<string>	<bes product>	*Plain*	mac
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*	mac
size of <application usage summary instance>	sizes	<integer>	<application usage summary instance>	*Plain*	mac
size of <array>	sizes	<integer>	<array>	*Plain*	mac
size of <datafork>	sizes	<integer>	<datafork>	*Plain*	mac
size of <dictionary>	sizes	<integer>	<dictionary>	*Plain*	mac
size of <file>	sizes	<integer>	<file>	*Plain*	mac
size of <integer set>	sizes	<integer>	<integer set>	*Plain*	core
size of <ram>	sizes	<integer>	<ram>	*Plain*	mac
size of <resfork>	sizes	<integer>	<resfork>	*Plain*	mac
size of <string set>	sizes	<integer>	<string set>	*Plain*	core
size of <type>	sizes	<integer>	<type>	*Plain*	core
size of <volume>	sizes	<integer>	<volume>	*Plain*	mac
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
sound folder of <domain>	sound folders	<folder>	<domain>	*Plain*	mac
span <string> of <html>	spans	<html>	<html>	*Named*	core
span <string> of <string>	spans	<html>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
span of <html>	spans	<html>	<html>	*Plain*	core
span of <string>	spans	<html>	<string>	*Plain*	core
speech folder of <domain>	speech folders	<folder>	<domain>	*Plain*	mac
speed of <processor>	speeds	<hertz>	<processor>	*Plain*	mac
standard deviation of <floating point>	standard deviations	<floating point>	<floating point>	*Plain*	core
standard deviation of <integer>	standard deviations	<floating point>	<integer>	*Plain*	core
standby state	standby states	<power state>	<world>	*PlainGlobal*	mac
start date of <license>	start dates	<time>	<license>	*Plain*	mac
start of <substring>	starts	<string position>	<substring>	*Plain*	core
start of <time range>	starts	<time>	<time range>	*Plain*	core
startup items folder of <domain>	startup items folders	<folder>	<domain>	*Plain*	mac
state of <dummy>	states	<string>	<dummy>	*Plain*	mac
state of <monitor power interval>	states	<power state>	<monitor power interval>	*Plain*	mac
state of <system power interval>	states	<power state>	<system power interval>	*Plain*	mac
status of <action>	statuss	<string>	<action>	*Plain*	mac
string <integer> of <array>	strings	<string>	<array>	*Numbered*	mac
string <string>	strings	<string>	<world>	*NamedGlobal*	core
string <string> of <dictionary>	strings	<string>	<dictionary>	*Named*	mac
string <string> of <preference>	strings	<string>	<preference>	*Named*	mac
string of <osxvalue>	strings	<string>	<osxvalue>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
string version of <application usage summary instance>	string versions	<string>	<application usage summary instance>	*Plain*	mac
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
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*	mac
subnet address of <network adapter>	subnet addresses	<ipv4 address>	<network adapter>	*Plain*	mac
subnet address of <network ip interface>	subnet addresses	<ipv4 address>	<network ip interface>	*Plain*	mac
subnet mask of <network adapter interface>	subnet masks	<ipv4or6 address>	<network adapter interface>	*Plain*	mac
subnet mask of <network adapter>	subnet masks	<ipv4 address>	<network adapter>	*Plain*	mac
subnet mask of <network ip interface>	subnet masks	<ipv4 address>	<network ip interface>	*Plain*	mac
subscribe time of <site>	subscribe times	<time>	<site>	*Plain*	mac
substring <(integer, integer)> of <string>	substrings	<substring>	<string>	*Index<(integer, integer)>*	core
substring <string> of <string>	substrings	<substring>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
subtype of <component>	subtypes	<string>	<component>	*Plain*	mac
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
symbol of <binary operator>	symbols	<string>	<binary operator>	*Plain*	core
symbol of <unary operator>	symbols	<string>	<unary operator>	*Plain*	core
system domain	system domains	<domain>	<world>	*PlainGlobal*	mac
system folder of <domain>	system folders	<folder>	<domain>	*Plain*	mac
system interval of <power history>	system intervals	<system power interval>	<power history>	*Plain*	mac
system version	system versions	<version>	<world>	*PlainGlobal*	mac
table <string> of <html>	tables	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
temporary items folder of <domain>	temporary items folders	<folder>	<domain>	*Plain*	mac
text encodings folder of <domain>	text encodings folders	<folder>	<domain>	*Plain*	mac
tfoot <string> of <html>	tfoots	<html>	<html>	*Named*	core
tfoot <string> of <string>	tfoots	<html>	<string>	*Named*	core
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
thead of <html>	theads	<html>	<html>	*Plain*	core
thead of <string>	theads	<html>	<string>	*Plain*	core
themes folder of <domain>	themes folders	<folder>	<domain>	*Plain*	mac
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 duration of <application usage summary instance>	total durations	<time interval>	<application usage summary instance>	*Plain*	mac
total duration of <application usage summary>	total durations	<time interval>	<application usage summary>	*Plain*	mac
total run count of <application usage summary instance>	total run counts	<integer>	<application usage summary instance>	*Plain*	mac
total run count of <application usage summary>	total run counts	<integer>	<application usage summary>	*Plain*	mac
total space of <volume>	total spaces	<integer>	<volume>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
tuesday	tuesdays	<day of week>	<world>	*PlainGlobal*	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 <bundle>	types	<file type>	<bundle>	*Plain*	mac
type of <component>	types	<string>	<component>	*Plain*	mac
type of <license>	types	<string>	<license>	*Plain*	mac
type of <osxvalue>	types	<string>	<osxvalue>	*Plain*	mac
type of <processor>	types	<string>	<processor>	*Plain*	mac
type of <scsidevice>	types	<string>	<scsidevice>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
type of <site>	types	<string>	<site>	*Plain*	mac
type of <volume>	types	<string>	<volume>	*Plain*	mac
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
unary operator <string>	unary operators	<unary operator>	<world>	*NamedGlobal*	core
unary operator returning <type>	unary operators returning	<unary operator>	<world>	*Index<type>Global*	core
union of <integer set>	unions	<integer set>	<integer set>	*Plain*	core
union of <string set>	unions	<string set>	<string set>	*Plain*	core
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unique value of <ipv6 address>	unique values	<ipv6 address with multiplicity>	<ipv6 address>	*Plain*	core
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 <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
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unix of <operating system>	unixes	<boolean>	<operating system>	*Plain*	mac
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*	mac
up of <network adapter>	ups	<boolean>	<network adapter>	*Plain*	mac
up of <network interface>	ups	<boolean>	<network interface>	*Plain*	mac
up of <network ip interface>	ups	<boolean>	<network ip interface>	*Plain*	mac
upload progress of <client>	upload progresses	<string>	<client>	*Plain*	mac
ups of <power level>	upss	<boolean>	<power level>	*Plain*	mac
uptime of <operating system>	uptimes	<time interval>	<operating system>	*Plain*	mac
url of <site>	urls	<string>	<site>	*Plain*	mac
usb	usbs	<usb>	<world>	*PlainGlobal*	mac
usb plane of <registryroot>	usb planes	<registrynode>	<registryroot>	*Plain*	mac
used percent of <volume>	used percents	<integer>	<volume>	*Plain*	mac
used space of <volume>	used spaces	<integer>	<volume>	*Plain*	mac
user	users	<user>	<world>	*PlainGlobal*	mac
user <string>	users	<user>	<world>	*NamedGlobal*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
user domain	user domains	<domain>	<world>	*PlainGlobal*	mac
user of <logged on user>	users	<user>	<logged on user>	*Plain*	mac
user temp folder of <domain>	user temp folders	<folder>	<domain>	*Plain*	mac
users folder of <domain>	users folders	<folder>	<domain>	*Plain*	mac
usual name of <property>	usual names	<string>	<property>	*Plain*	core
utilities folder of <domain>	utilities folders	<folder>	<domain>	*Plain*	mac
value of <array>	values	<osxvalue>	<array>	*Plain*	mac
value of <dictionaryentry>	values	<osxvalue>	<dictionaryentry>	*Plain*	mac
value of <environment variable>	values	<string>	<environment variable>	*Plain*	mac
value of <fixlet_header>	values	<string>	<fixlet_header>	*Plain*	mac
value of <setting>	values	<string>	<setting>	*Plain*	mac
value of <user attribute>	values	<string>	<user attribute>	*Plain*	mac
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*	mac
variable of <environment>	variables	<environment variable>	<environment>	*Plain*	mac
vendor of <scsidevice>	vendors	<string>	<scsidevice>	*Plain*	mac
version <integer> of <file>	versions	<version>	<file>	*Numbered*	mac
version <string>	versions	<version>	<world>	*NamedGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
version of <application usage summary instance>	versions	<version>	<application usage summary instance>	*Plain*	mac
version of <bios>	versions	<string>	<bios>	*Plain*	mac
version of <bundle>	versions	<version>	<bundle>	*Plain*	mac
version of <client>	versions	<version>	<client>	*Plain*	mac
version of <component>	versions	<version>	<component>	*Plain*	mac
version of <current relay>	versions	<version>	<current relay>	*Plain*	mac
version of <filesystem object>	versions	<version>	<filesystem object>	*Plain*	mac
version of <folder>	versions	<version>	<folder>	*Plain*	mac
version of <operating system>	versions	<version>	<operating system>	*Plain*	mac
version of <scsibus>	versions	<version>	<scsibus>	*Plain*	mac
version of <site>	versions	<integer>	<site>	*Plain*	mac
version of <usb>	versions	<version>	<usb>	*Plain*	mac
virtual memory	virtual memories	<boolean>	<world>	*PlainGlobal*	mac
visible of <file>	visibles	<boolean>	<file>	*Plain*	mac
voices folder of <domain>	voices folders	<folder>	<domain>	*Plain*	mac
volume	volumes	<volume>	<world>	*PlainGlobal*	mac
volume <integer>	volumes	<volume>	<world>	*NumberedGlobal*	mac
volume <string>	volumes	<volume>	<world>	*NamedGlobal*	mac
volume of <file>	volumes	<volume>	<file>	*Plain*	mac
volume of <folder>	volumes	<volume>	<folder>	*Plain*	mac
volume settings folder of <domain>	volume settings folders	<folder>	<domain>	*Plain*	mac

Key Phrase	Plural	Creates a	From a	*Form*	Ref
waiting for download of <action>	waiting for downloads	<boolean>	<action>	*Plain*	mac
wake on lan subnet cidr string	wake on lan subnet cidr strings	<string>	<world>	*PlainGlobal*	mac
wednesday	wednesdays	<day of week>	<world>	*PlainGlobal*	core
week	weeks	<time interval>	<world>	*PlainGlobal*	core
weight of <selected server>	weights	<integer>	<selected server>	*Plain*	mac
wide16 scsi of <scsibus>	wide16 scsis	<boolean>	<scsibus>	*Plain*	mac
wide32 scsi of <scsibus>	wide32 scsis	<boolean>	<scsibus>	*Plain*	mac
windows of <operating system>	windowses	<boolean>	<operating system>	*Plain*	mac
windows server count of <bes product>	windows server counts	<integer>	<bes product>	*Plain*	mac
workstation count of <bes product>	workstation counts	<integer>	<bes product>	*Plain*	mac
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
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>
<cast> as string	<string>	<cast>
<client process owner> as string	<string>	<client process owner>
<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>

Key Phrase	Creates a	From a
<filesystem object> as file	<file>	<filesystem object>
<filesystem object> as folder	<folder>	<filesystem object>
<filesystem object> as string	<string>	<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>
<ipv4 address> as string	<string>	<ipv4 address>

Key Phrase	Creates a	From a
<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>
<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>
<power level> as string	<string>	<power level>
<power state> as string	<string>	<power state>
<property> as string	<string>	<property>
<rope> as string	<string>	<rope>
<security identifier> as string	<string>	<security identifier>
<setting> as string	<string>	<setting>
<site version list> as string	<string>	<site version list>
<stage> as string	<string>	<stage>
<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>

Key Phrase	Creates a	From a
<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>
<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 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>

Key Phrase	Creates a	From a
<string> as zoned time_of_day	<time of day with time zone>	<string>
<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>
<user attribute> as string	<string>	<user attribute>
<version> as string	<string>	<version>
<version> as version	<version>	<version>
<year> as integer	<integer>	<year>
<year> as string	<string>	<year>

Part Three

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.

Part Four

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V