Tivoli. *Endpoint Manager*

Version 8.1

Windows 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 Windows 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 Windows. Thousands of Inspectors have been created to expose the inner workings of Windows 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 only 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 Windows-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.

To get the most out of this guide, it helps to have some experience with the Windows Registry.

This document describes Inspectors for Windows 2000, Server 2003, XP, Vista and Windows 7. Contact your Tivoli Endpoint Manager marketing representative for information about Inspector Guides for other operating systems, including Solaris, Mac, 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. 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.

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
escape of <string>	*Plain*	Returns a string containing a \\ for every \ character found. Useful for setting registry key values to strings in regset action commands. Win, WM
expand environment string of <string>	*Plain*	Uses the Windows ExpandEnvironmentStrings API to translate a string containing special Windows environment variables. For example, %windir%\my.dll might expand to c:\winnt\my.dll. Win
expand x64 environment string of <string>	*Plain*	Given an environment string such as %ProgramFiles%, this Inspector expands it to 'C:\Program Files (x86)'. Although similar to 'expand environment string of <string>', this Inspector works on 64-bit Windows systems only. On a 32-bit system, it returns 'No Such Object'. Win:8.1

Properties

Key Phrase	*Form*	Return Type	Description
expand x64 environment string of <string>	*Plain*	<string>	Given an environment string such as %ProgramFiles%, this Inspector expands it to 'C:\Program Files (x86)'. Although similar to 'expand environment string of <string>', this Inspector works on 64-bit Windows systems only. On a 32-bit system, it returns 'No Such Object'. Win:8.1

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.

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
ansi code page	*PlainGlobal*	<integer>	Returns an integer value of the Windows API GetACP. Win, WM
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
backoffice bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the backoffice bit of the Windows operating system suite mask is set. Win

Key Phrase	*Form*	Return Type	Description
blade bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the blade bit of the Suite Mask (a part of the Windows OS version) is set. Win
communications bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the communications bit of the Suite Mask (a part of the Windows OS version) is set. Win
computer id	*PlainGlobal*	<integer>	This is a unique integer assigned to the computer by the BES system. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
computer name	*PlainGlobal*	<string>	Returns a string corresponding to the name of the computer as it appears on the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
custom site subscription effective date <string>	*NamedGlobal*	<time>	Returns the date the custom site (specified by <string>) was last subscribed or unsubscribed. It is used internally by BES to manage custom site subscriptions. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
datacenter bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the datacenter bit of the Suite Mask (a part of the Windows OS version) is set. Win
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
embedded nt bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the embedded nt bit of the Suite Mask (a part of the Windows OS version) is set. Win
embedded restricted bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the embedded restricted bit of the Suite Mask (a part of the Windows OS version) is set. Win

Key Phrase	*Form*	Return Type	Description
enterprise bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the enterprise bit of the Suite Mask (a part of the Windows OS version) is set. Win
hostname	*PlainGlobal*	<string>	Returns the standard host name, usually for the computer's network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
hyperthreading capable	*PlainGlobal*	<boolean>	Returns a boolean TRUE if the agent is able to detect that the processor is capable of running with hyperthreading enabled. Win, WM
hyperthreading enabled	*PlainGlobal*	<boolean>	Returns TRUE if the machine is running with hyperthreading enabled, a method in which each physical processor on the machine presents itself as multiple logical processors to the operating system. Win, WM
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
local administrator	*PlainGlobal*	<boolean>	Returns the boolean TRUE if the user belongs to the local administrator group. Also returns TRUE for Win9x and WinME. Win
logical processor count	*PlainGlobal*	<integer>	Returns the number of logical processors available per physical processor. This can be interpreted as the number of hyperthreads that could be enabled on the machine. On a machine with 2 physical processors, each with 2 possible hyperthreads per processor, the 'physical processor count' and the 'logical processor count' would both return 2, while the 'number of processors' would return 4, since there are a total of 4 logical processors available for work. With hyperthreading turned off, the 'number of processors', 'logical processor count' and 'physical processor count' would all be 2. Disabling one of those processors will then give 'number of processors'=1, 'logical processor count'=2, and 'physical processor count'=1. If the number of processors / physical processor count != logical processor count, you can turn on hyperthreading. Win, WM

Key Phrase	*Form*	Return Type	Description
oem code page	*PlainGlobal*	<integer>	Returns an integer value of the Windows API GetOEMCP. Win, WM
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
personal bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the personal bit of the Suite Mask (a part of the Windows OS version) is set. Win
physical processor count	*PlainGlobal*	<integer>	Returns the number of physical processors on the machine. Note that 'number of processors' returns the number of logical processors. To determine the number of logical processors per physical processor, use 'number of processors / physical processor count'. Win, WM
single user ts bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the single user ts bit of the Suite Mask (a part of the Windows OS version) is set. Win

Key Phrase	*Form*	Return Type	Description
small business bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the small business bit of the Suite Mask (a part of the Windows OS version) is set. Win
small business restricted bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the small business restricted bit of the Suite Mask (a part of the Windows OS version) is set. Win
system language	*PlainGlobal*	<string>	Returns the language of the system as a string. It is identified using the GetSystemDefaultLangID() system call. See the language keyword of the application object for a list of possible language value. Win, Lin, Sol, HPUX, AIX, WM, Ubu
terminal bit <operating system suite mask>	*Index<operating system suite mask>Global*	<boolean>	Returns TRUE if the terminal bit of the Suite Mask (a part of the Windows OS version) is set. Win
total processor core count	*PlainGlobal*	<integer>	Returns an integer corresponding to the total number of processor cores. Win, WM
user language	*PlainGlobal*	<string>	Returns the language of the system as a string. It is identified by using the GetUserDefaultLangId() system call. See the language keyword of the application object for a list of possible language value. Win, WM
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

- `ansi code page = 1252`

Returns TRUE on English Windows systems.

- `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.

- `number of processors / physical processor count != logical processor count`

Returns TRUE if hyperthreading has not yet been turned on.

Registry Objects

These are the keywords for dealing with the Windows registry. Particular attention is paid to registered applications and their associated file extensions.

Registry

These are the Inspectors that expose the Windows registry.

Creation Methods

Key Phrase	*Form*	Description
native registry	*PlainGlobal*	On 32 bit versions of windows, this returns the same as x32 registry. On 64 bit versions of windows, this returns the same as x64 registry. Win, WM
registry	*PlainGlobal*	Creates an object for accessing the registry. Returns a <dummy type> on the Macintosh. Win, , WM
x32 registry	*PlainGlobal*	Returns a 32-bit registry object. This Inspector is equivalent to the ordinary registry Inspector. Win, WM
x64 registry	*PlainGlobal*	Returns a 64-bit registry object. This Inspector is for 64-bit computers only; there is no 64-bit registry on a 32-bit computer. • Note that "x64 registry" and "native registry" on 64-bit machines do NOT provide the same view as the 64-bit version of regedit (the "physical" view). If you try to access the physical location of the 32-bit view keys using a 64-bit view, it will be mapped back to the equivalent location in the 64-bit view. Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
application <string> of <registry>	*Named*	<application>	Returns an application object matching name provided. See application. Win, WM
application folder <string> of <registry>	*Named*	<folder>	Returns the folder containing the matching name provided. See application. The application does not have to exist. The folder has to exist. Win, WM

Key Phrase	*Form*	Return Type	Description
application of <registry>	*Plain*	<application>	Iterates through the properly installed applications. See application. Win, WM
current user key <logged on user> of <registry>	*Index<logged on user>*	<registry key>	On 2000/2003/XP/Vista, this Inspector returns RegOpenCurrentUser. Under NT4, it uses HKEY_USERS\<sid>. On these systems, the Inspector may fail if run in a non-privileged context. Under Windows 9x, it returns HKEY_USERS\<username> if it exists. Otherwise it uses HKEY_USERS\Default. CAUTION: This Inspector will produce errors when run on a terminal server where a user has logged off but there is a 'keep alive' setting on the user session. This is because there is technically a logged-on user but that user's Current User registry hive is not loaded. To avoid this issue, use a qualifying 'whose' statement: • current user keys (logged on users whose (active of it)) of registry Win:7.0
device key <string> of <registry>	*Named*	<registry key>	Iterates through all the keys known to the configuration manager for active devices whose "DeviceDesc" matches the name provided. See registry key. Win
device key of <registry>	*Plain*	<registry key>	Iterates through all the keys known to the configuration manager for active devices. See registry key. Win
file extension <string> of <registry>	*Named*	<registry key>	Returns a key associated with the named extension. See registry key. Win, WM
file type <string> of <registry>	*Named*	<registry key>	Returns a key associated the named file type. See registry key. Win, WM
key <string> of <registry>	*Named*	<registry key>	Returns a key associated with the name provided. See registry key. Win, WM

Examples

```
name of application of key ".txt" of key "HKEY_CLASSES_ROOT" of the registry = "NOTEPAD.EXE"
```

True when text files are to be opened with notepad.exe on the current machine.

- `value of file extension "bmp" of registry = "Paint.Picture"`

  Returns TRUE if there is only one value of the key "HKEY_CLASSES_ROOT\.bmp" and it contains the string "Paint.Picture".

- `file extension ".txt" of the registry`

  Returns a key corresponding to the application designated to process files with this extension. The dot is optional in the name provided. Looks for the key under HKEY_CLASSES_ROOT.

- `file type "txtfile" of the registry`

  Returns a key whose existence indicates that there is an application designated to process text files. Looks for the key under HKEY_CLASSES_ROOT.

- `key "HKEY_CLASSES_ROOT\txtfile" of the registry`

  Returns a key whose existence indicates that there is an application designated to process text files. Looks for the key under HKEY_CLASSES_ROOT.

- `key "txtfile" of key "HKEY_CLASSES_ROOT" of the registry`

  Returns a key whose existence indicates that there is an application designated to process text files. Looks for the key under HKEY_CLASSES_ROOT.

Registry Key

The registry key objects represent Windows registry keys whose existence and properties can be inspected. Keys can be identified by name. There are several Inspectors that return keys from parts of the registry that store file associations and active device drivers.

Creation Methods

Key Phrase	*Form*	Description
current user key <logged on user> of <registry>	*Index<logged on user>*	On 2000/2003/XP/Vista, this Inspector returns RegOpenCurrentUser. Under NT4, it uses HKEY_USERS\<sid>. On these systems, the Inspector may fail if run in a non-privileged context. Under Windows 9x, it returns HKEY_USERS\<username> if it exists. Otherwise it uses HKEY_USERS\Default. Win:7.0
device key <string> of <registry>	*Named*	Iterates through all the keys known to the configuration manager for active devices whose "DeviceDesc" matches the name provided. Win
device key of <registry>	*Plain*	Iterates through all the keys known to the configuration manager for active devices. Win

Key Phrase	*Form*	Description
driver key of <active device>	*Plain*	The key identified by adding the value of 'driver key value name of active device' to HKLM\System\CurrentControlSet\Control\Class\. Win
driver key of <registry key>	*Plain*	Uses the value of "Driver" of the key to indirectly return a key corresponding to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\<value of Driver>. Win
file extension <string> of <registry>	*Named*	Creates a key object provided the registry indicates support for the named file extension. Win, WM
file type <string> of <registry>	*Named*	Creates a key object provided the registry indicates support for the named file type. Win, WM
key <string> of <registry key>	*Named*	Creates an object for the named sub-key of the key. Win, WM
key <string> of <registry>	*Named*	Creates an object for the named key. The name may be a full path to a key of the form "HKEY_CLASSES_ROOT\Fixlet.Pool\". Win, WM
key of <registry key>	*Plain*	Iterates through the sub-keys of a key. Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
application <string> of <registry key>	*Named*	<application>	Returns the application associated with the named command. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file type. Win, WM
application folder <string> of <registry key>	*Named*	<folder>	Returns the parent folder associated with the named application. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file type. Win, WM
application folder of <registry key>	*Plain*	<folder>	Returns the parent folder associated with the named application. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file type. Win, WM

Key Phrase	*Form*	Return Type	Description
application of <registry key>	*Plain*	<application>	Returns the application associated with the "open" command. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file extension. Win, WM
default value of <registry key>	*Plain*	<registry key value>	Returns the unnamed value associated with a key as a string. It does not necessarily exist. Win, WM
driver key of <registry key>	*Plain*	<registry key>	Normally used as a property of a device key. Looks up the value of "Driver" of the key provided to indirectly return another key corresponding to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\<value of Driver>. Win
key <string> of <registry key>	*Named*	<registry key>	Returns a key for the named sub-key. Win, WM
key of <registry key>	*Plain*	<registry key>	Iterates through the sub-keys of the key. Win, WM
last write time of <registry key>	*Plain*	<time>	Whenever a registry key is written (created or modified) Windows updates the last write time. This Inspector returns that time to you on all Windows agents. Win:8.0
name of <registry key>	*Plain*	<string>	Returns the name of the key as a string. Win, WM
security descriptor of <registry key>	*Plain*	<security descriptor>	Specifies the security descriptor associated with the specified registry key. Win
value <string> of <registry key>	*Named*	<registry key value>	Returns the named value stored under the key. See registry key value. Win, WM
value of <registry key>	*Plain*	<registry key value>	Iterates through values stored under a key. Win, WM

NOTE:

The terminology of keys, values, default values and values that have names and data is chosen to match the convention's used by the Windows registry editor as well as the API's provided by the Windows operating system for accessing this information.
Top branches of the Windows registry include:

HKEY_CLASSES_ROOT
HKEY_LOCAL_MACHINE

HKEY_CURRENT_USER
HKEY_USERS
HKEY_CURRENT_CONFIG
HKEY_DYN_DATA
HKEY_PERFORMANCE_DATA (NT)

The trailing slashes on registry key names are optional.

File extensions, File types, and associated applications:
The following table represents a small part of the registry. It illustrates the relationship between the notions of file extension, file type, and the shell commands associated with the inspector keywords

Description	HKEY_CLASSES_ROOT\	Default Value
File extension key	.txt	default value = txtfile
File type key	txtfile\shell\	
Named command	txtfile\shell\open\command	default value = c:\windows\NOTEPAD.EXE %1
Named command	txtfile\shell\print\command	default value = c:\windows\NOTEPAD.EXE /p %1

Device Keys of the registry:
The Configuration Manager of the Windows 9x operating system maintains a list of active devices under the HKEY_DYN_DATA\Config Manager\Enum key of the registry. The items in the list contain values named "HardwareKey" which are the names of keys under HKEY_LOCAL_MACHINE\Enum. The value "DeviceDesc" contains a description of the device. The device key inspectors allow you to determine if a particular piece of hardware matching the Device Description is currently active.
Device key "Hardware ABC from Company XYZ" will only return a key if there is an entry under HKEY_DYN_DATA\Config Manager\Enum that points to it.

Examples

- `file extension ".txt" of the registry`

  Returns a key corresponding to the application that opens files with this extension. The dot is optional in the name provided.

- `name of application of file extension "html" of the registry = "iexplore.exe"`

  Verifies that the name of the application assigned to process html documents is Internet Explorer.

- `file type "txtfile" of the registry`

  Returns a key whose existence may indicate that there is an application designated to process files of this type. Looks for the key under HKEY_CLASSES_ROOT.

- `key "HKEY_CLASSES_ROOT\txtfile" of the registry`

  Returns a key whose existence indicates that there is an application designated to process text files.

- `application "print" of key "HKEY_CLASSES_ROOT\.txt" of registry`

  Returns the application designated to print the files with ".txt" extensions.

- `application "bigfix.exe" of the registry as string`
  - Results in a string of the form "BigFix.exe" "1.0.32.0" "BigFix Client Application" "1.0.32.0" "BigFix Inc.".

- `name of application of key ".txt" of key "HKEY_CLASSES_ROOT" of the registry = "NOTEPAD.EXE"`
  - True when text files are to be opened with notepad.exe on the current machine.

- `default value of key ".txt" of key "HKEY_CLASSES_ROOT" of the registry = "txtfile"`
  - True when the file extension is of type txtfile.

- `names of keys of key "HKEY_CLASSES_ROOT\txtfile\shell" of the registry`
  - Iterates through all the sub-keys of the key provided. In this case, returning all the shell commands available to process the given file type.

- `type of value "ProfileFlags" of key "HKEY_CURRENT_CONFIG" of registry = "REG_BINARY"`
  - Returns TRUE when a value named ProfileFlags under the key "HKEY_CURRENT_CONFIG" exists and contains binary data.

- `size of value whose (name of it = "ProfileFlags") of key "HKEY_CURRENT_CONFIG" of registry = 4`
  - Returns TRUE when a value named ProfileFlags exists as a child of the key "HKEY_CURRENT_CONFIG" and the size of it is 4.

- `value "AutoRewind" of key "HKEY_CURRENT_USER\Software\Microsoft\ActiveMovie\Control\Media Player" of registry = 1`
  - Returns TRUE if the Media Player is set to AutoRewind.

Registry Key Value

This Inspector is used to access values stored within a registry key. All values have sizes and types. All of the values of a registry key have names except one, and it is called the 'default value'. The type of the data stored in the value determines what casting operations are allowed. We have implemented several casting Inspectors that you can use to extract values from the registry.

Creation Methods

Key Phrase	*Form*	Description
default value of <registry key>	*Plain*	Every key may have a default or unnamed value. This inspector returns the default value of the key. This value has the same properties as any other registry key value except that it does not have a name property. Win, WM
value <string> of <registry key>	*Named*	Creates an object with the value of the key. The name property of the value will match the name provided. Win, WM
value of <registry key>	*Plain*	Creates an object with all the values of a key. Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
<registry key value> as application	*Cast*	<application>	If the data stored in the value is a string and it is the full pathname of an application that exists on disk, the application object is returned. Win, WM
<registry key value> as file	*Cast*	<file>	If the data stored in the value is a string and it is the full pathname of a file that exists on disk, the file object is returned. Win, WM
<registry key value> as folder	*Cast*	<folder>	If the data stored in the value is a string and it is the full pathname of a folder that exists on disk, the folder object is returned. Win, WM
<registry key value> as integer	*Cast*	<integer>	Returns the value stored in the registry entry provided it can be fully represented as an integer. Win, WM
<registry key value> as string	*Cast*	<string>	Returns a string if the data of the value is of type REG_SZ. Win, WM

Key Phrase	*Form*	Return Type	Description
<registry key value> as system file	*Cast*	<file>	If the data stored in the value is a string and it is a relative pathname from the system folder of a file that exists on disk, the corresponding file object is returned. Win, WM
<registry key value> as time	*Cast*	<time>	If the data stored in the value is a string in MIME compliant date format, this property will return a time object. If the data stored is a binary value and is 16 or more bytes in length, its first 16 bytes are interpreted as a SYSTEMTIME and the corresponding time object is returned. See time. Win, WM
name of <registry key value>	*Plain*	<string>	Returns the name of the value as a string. (see escape of <string> for more information). Win, WM
size of <registry key value>	*Plain*	<integer>	Returns the size of the data as an integer. Win, WM
type of <registry key value>	*Plain*	<registry key value type>	Returns the type of the data of the value. See type of value of key or registry. Win, WM

Operators

Key phrase	Return Type	Description
<registry key value> {cmp} <integer>	<*boolean*>	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, WM
<registry key value> {cmp} <registry key value>	<*boolean*>	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, WM
<registry key value> {cmp} <string>	<*boolean*>	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, WM

NOTE:

Eleven literal types are currently recognized. Future types may be handled as numeric types. The possible numeric values of each type and their string literal values include:

0	REG_NONE
1	REG_SZ
2	REG_EXPAND_SZ
3	REG_BINARY
4	REG_DWORD
5	REG_DWORD_BIG_ENDIAN
6	REG_LINK
7	REG_MULTI_SZ
8	REG_RESOURCE_LIST
9	REG_FULL_RESOURCE_DESCRIPTOR
10	REG_RESOURCE_REQUIREMENTS_LIST

Examples

- `default value of key ".txt" of key "HKEY_CLASSES_ROOT" of the registry = "txtfile"`
  - True when the file extension is of type txtfile.

- `type of value "ProfileFlags" of key "HKEY_CURRENT_CONFIG" of registry = "REG_BINARY"`
  - Returns TRUE when a value named ProfileFlags under the key "HKEY_CURRENT_CONFIG" exists and contains binary data.

- `value "AutoRewind" of key "HKEY_CURRENT_USER\Software\Microsoft \ActiveMovie\Control\Media Player" of registry = 1`
  - Returns TRUE when the specified value of the key equals 1.

- `size of value whose (name of it = "ProfileFlags") of key "HKEY_CURRENT_CONFIG" of registry = 4`
  - Returns TRUE when a value named ProfileFlags exists as a child of the key "HKEY_CURRENT_CONFIG" and the size of it is 4.

Registry Key Value Type

The type identifier of the data associated with a registry key value.

Creation Methods

Key Phrase	*Form*	Description
type of <registry key value>	*Plain*	Creates an integer designating the type of data stored in the registry key value. See the registry MS documentation for these numeric values, which correspond to the enumerated constants discussed in the "<registry key value type> as string" property. Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
<registry key value type> as string	*Cast*	<string>	Returns the type of value as a string. One of REG_SZ, REG_NONE, REG_DWORD, REG_LINK, REG_BINARY, REG_MULTI_SZ, REG_EXPAND_SZ, REG_RESOURCE_LIST, REG_DWORD_LITTLE_ENDIAN, REG_DWORD_BIG_ENDIAN, REG_FULL_RESOURCE_DESCRIPTOR, REG_RESOURCE_REQUIREMENTS_LIST. Win, WM

Operators

Key phrase	Return Type	Description
<registry key value type> {cmp} <integer>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, WM
<registry key value type> {cmp} <registry key value type>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, WM
<registry key value type> {cmp} <string>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, WM

Filesystem Objects

This chapter covers the keywords for extracting information from the file system, like files, drives, pathnames, folders, and more. 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

Properties

Key Phrase	*Form*	Return Type	Description
<filesystem object> as string	*Cast*	<string>	Casts a filesystem object as a string. Win:8.0, Lin, Sol, HPUX, AIX, Mac, Ubu
accessed time of <filesystem object>	*Plain*	<time>	When the filesystem object (file or folder) was last accessed. Some file systems maintain this property. Win, Lin, Sol, HPUX, AIX, WM, Ubu
ancestor of <filesystem object>	*Plain*	<folder>	Returns all ancestor folders (recursive parent folders) of the given filesystem object (file or folder). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
archive of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the Archive bit is turned on for the specified file or folder (filesystem object). This bit is often used by backup software. Win, WM
compressed of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (filesystem object) has been compressed. Win, WM
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
drive of <filesystem object>	*Plain*	<drive>	Returns the drive associated with the specified file or folder (filesystem object). Win, WM
hidden of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (filesystem object) is marked as hidden. Win, WM
location of <filesystem object>	*Plain*	<string>	Returns the name of the directory in which the file or folder (filesystem object) is located. Win, Lin, Sol, HPUX, AIX, WM, Ubu

Key Phrase	*Form*	Return Type	Description
modification time of <filesystem object>	*Plain*	<time>	The date and time of latest modification of the file. This corresponds to what is shown in the "Get Info" box. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
name of <filesystem object>	*Plain*	<string>	This returns the name of the file or folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
normal of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (filesystem object) is 'normal'. Win, WM
offline of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (the filesystem object) is marked as 'offline'. Win, WM
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
readonly of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (the filesystem object) is marked as read-only. Win, WM
system of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (the filesystem object) is marked as a system folder. Win, WM
temporary of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (the filesystem object) is marked as a temporary folder. Win, WM

Examples

- `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.

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
<registry key value> as file	*Cast*	If the value contains a string and the string points to an existing file, a file object is returned. Win, WM
<registry key value> as system file	*Cast*	If the value contains a string and the string points to an file, a file object is returned. Relative paths are interpreted relative to the system folder. Win, WM
active device file	*PlainGlobal*	Under Windows NT, returns a list of file objects corresponding the list returned from the Windows NT EnumDeviceDrivers() function. Win
active device file <string>	*NamedGlobal*	Under Windows NT, returns a file object corresponding to the name provided. Names provided need only match the last component of the file. This inspector uses the Windows NT EnumDeviceDrivers() function. Win
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 of <folder>	*Plain*	Iterates through the files of a folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
file of <service>	*Plain*	This returns the executable file associated with the given <service> under Windows 2000 operating systems. Win
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
image file of <process>	*Plain*	Returns a file object representing the image file used to launch the process. In some cases, this subproperty may not exist. For example, the "System" process does not have an image file. Win:8.0
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
system file <string>	*NamedGlobal*	Creates the file objects corresponding to the named file within the system folder. Win, WM
system ini device file	*PlainGlobal*	Creates a list of file objects corresponding to all the device= lines of the system.ini file. Note that items whose pathnames start with '*' are not placed into this list. Win

Key Phrase	*Form*	Description
system ini device file <string>	*NamedGlobal*	Creates a file object corresponding to a device file loaded as a result of a device= line of the system.ini file. The name provided should match the last component of the full path. Win
windows file <string>	*NamedGlobal*	Returns a file object corresponding to the relative pathname (within the Windows folder) provided. See file. Win, WM
x32 file <string>	*NamedGlobal*	Returns an object corresponding to a 32 bit file with name specified by <string>. Win
x64 file <string>	*NamedGlobal*	Returns an object corresponding to a 64 bit file with pathname specified by <string>. On a 32-bit computer, this is equivalent to a normal file Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. Win

Properties

Key Phrase	*Form*	Return Type	Description
<file> as string	*Cast*	<string>	Creates a string containing the full pathname of the specified file. See <file>. Win, , , , , WM
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
executable file format of <file>	*Plain*	<string>	Returns a four-byte string containing the format specifier for the specified file. Win, WM

Key Phrase	*Form*	Return Type	Description
file version of <file>	*Plain*	<version>	Returns the file version extracted from the file's resource block. See version. Win, WM
first raw version block of <file>	*Plain*	<file version block>	Returns the first version block directly from a PE file. If the first block is sufficient for your purposes, use this version inspector for best speed. Win, WM
key <string> of <file>	*Named*	<string>	Returns a key and its value from the given structured text file. It iterates over lines that start with the key name (as specified by <string>) followed by an = or : character. When searching, white space is ignored. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
line <integer> of <file>	*Numbered*	<file line>	Returns the nth line (specified by <integer>) from the given file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line containing <string> of <file>	*Named*	<file line>	Returns all lines from the given file that contain the specified string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line of <file>	*Plain*	<file line>	Iterates over all the lines of the specified file. NOTE: lines are truncated to 1023 characters. CAUTION: This Inspector maintains a handle to the specified file, so during its operation it may block any other applications that attempt to open the file. Inspectors open files as with both read and write sharing, so apps that open with compatibleaccess will not block. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
line starting with <string> of <file>	*Named*	<file line>	Same as line <string> of <file>, returns the lines of the given file that start with the specified string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
only raw version block of <file>	*Plain*	<file version block>	Returns the only version block directly from a PE file. Win, WM
only version block of <file>	*Plain*	<file version block>	Most files only have 1 version block. This property allows language independent access when there is only one version block present. The result is the same as 'version block 1'. Win, WM

Key Phrase	*Form*	Return Type	Description
pem encoded certificate of <file>	*Plain*	<x509 certificate>	Reads and returns the certificate from a file in the PEM format. This can be used to analyze encryption credentials on decrypting relays or root servers. Win:7.1, WM
product version of <file>	*Plain*	<version>	Returns the product version extracted from the file's resource block. See version. Win, WM
raw file version of <file>	*Plain*	<version>	Returns the file version directly from a PE file. Win, WM
raw product version of <file>	*Plain*	<version>	Returns the product version directly from a PE file. Win, WM
raw version block <integer> of <file>	*Numbered*	<file version block>	Returns the numbered version block directly from a PE file. Win, WM
raw version block <string> of <file>	*Named*	<file version block>	Returns the named version block directly from a PE file. Win, WM
raw version block of <file>	*Plain*	<file version block>	Returns the version block directly from a PE file. Win, WM
raw version of <file>	*Plain*	<version>	Returns the version directly from a PE file. Win, WM
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
security descriptor of <file>	*Plain*	<security descriptor>	Specifies the security descriptor associated with the specified file. Win
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

Key Phrase	*Form*	Return Type	Description
shortcut of <file>	*Plain*	<file shortcut>	Returns the properties and locates the target of a file shortcut: • pathname (string) • start in pathname (string) • argument string (string) • icon pathname (string) • icon index (integer). Win
size of <file>	*Plain*	<integer>	Returns the size in bytes of a file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
variable of <file>	*Plain*	<string>	Returns the names of variables contained in an INF style file, in the format [section].name=value. Win, Lin, Sol, HPUX, AIX, WM, Ubu
version block <integer> of <file>	*Numbered*	<file version block>	You can identify the particular version block you want to access by ordinal number. Win, WM
version block <string> of <file>	*Named*	<file version block>	You can identify the particular version block you are looking up by name. The name you provide should match the id string of the version block. Win, WM
version block of <file>	*Plain*	<file version block>	Iterates through the version blocks of a file. Win, WM
version of <file>	*Plain*	<version>	Synonym for file version of <file>. Win, , WM
xml document of <file>	*Plain*	<xml dom document>	Returns an XML Document Object Model (DOM) for the specified file. Win, WM

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

NOTE:

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

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

Examples

- `wait "{pathname of download file "update.exe"}"`

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

- `Number of find files "siteico*.bmp" of client folder of current site = 3`

Returns TRUE if there are 3 files matching the wildcard pattern siteico*.bmp.

- `names of find files "*.exe" of windows folder`

Returns a list of the names of all the executable programs in the Windows folder.

- `modification time of masthead of current site < time "4 Aug 1997 01:00 pdt"`

TRUE if the masthead is older than the specified date.

- `exists windows file "command.com"`

Verifies the existence of the named file in the Windows folder.

- `byte 0 of file "C:\test.txt"`

Returns the first byte in the specified file.

- `content of file "oeminfo.ini" of system folder as lowercase contains "dell"`

Returns TRUE if the specified file contains the string "dell" anywhere in the file.

- `executable file format of client`

Returns a string like "PE%00%00".

- `file version of application "iexplore.exe" of the registry < "4"`

Test for older version of IE -- returns TRUE is version is less than 4.

- `lines of file "c:\autoexec.bat"`

  Returns all the lines in the specified autoexec.bat file.

- `product version of file "qna.exe" of parent folder of regapp "bigfix.exe" = product version of regapp "bigfix.exe"`

  Verifies the existence a co-executable located in the same folder with the proper version.

- `product version of regapp "bigfix.exe" > version "1.0.21"`

  Returns TRUE if the application has a version of 1.0.22 or higher, and FALSE if the application has a version of 1.0.21 or less.

- `pathname of shortcut of file "BigFix.lnk" of (value "Common Desktop" of key "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders" of registry as folder)`

  Returns the pathname associated with the shortcut.

- `version of file (pathname of shortcut of file "BigFix.lnk" of folder "c:\Documents and Settings\All Users\Desktop")`

  Returns the version number of the application to which the shortcut points.

Application

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

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

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

Creation Methods

Key Phrase	*Form*	Description
<registry key value> as application	*Cast*	If the value is of type string, and the string is a full pathname to an executable that exists on disk, an application object is created. Win, WM
application <string>	*NamedGlobal*	Creates an application object for the name provided. Win, Lin, Sol, HPUX, AIX, WM, Ubu
application <string> of <folder>	*Named*	As with the file object, you can create an application object by naming it relative to its parent folder. Win, Lin, Sol, HPUX, AIX, WM, Ubu
application <string> of <registry key>	*Named*	Creates the application object associated with the named command. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file type. Win, WM
application <string> of <registry>	*Named*	Creates the application object associated with the name provided. The name provided must be the name of a sub-key of the 'App Paths' registry key. See notes. Win, WM
application of <registry key>	*Plain*	Creates the application object associated with the "open" command. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file extension. Win, WM
application of <registry>	*Plain*	Iterates through the 'App Paths' registry key creating objects for the applications that exist. See notes. Win, WM
default web browser	*PlainGlobal*	Returns the application object (typically the web browser) associated with HTML documents. Win
recent application	*PlainGlobal*	Iterates through the list of recently executed applications, creating application objects. This includes the list of all currently running applications. Win, Lin, Sol:8.1, WM, Ubu
recent application <string>	*NamedGlobal*	If named application has been executed recently, this inspector creates an application object. Only specify the last component of the filename. Win, Lin, Sol:8.1, WM, Ubu

Key Phrase	*Form*	Description
regapp	*PlainGlobal*	This Inspector returns the applications available to the Client. • On Windows systems, this Inspector returns all the application objects defined under the 'App Paths' key of the registry. • On a Macintosh, it 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. It returns a <filesystem object>. This Inspector is the same as application, and is included for compatibility with Windows. Win, , WM
regapp <string>	*NamedGlobal*	Returns an application object for the name provided. See application and regapp. Win, WM
running application	*PlainGlobal*	Iterates through the list of running applications. Win, Lin, Sol:8.1, WM, Ubu
running application <string>	*NamedGlobal*	If the named application is currently executing then this inspector creates an application object. Only specify the last component of the file name. Win, Lin, Sol:8.1, WM, Ubu
x32 application <string>	*NamedGlobal*	Returns an object corresponding to the 32-bit application specified by <string>. Win
x64 application <string>	*NamedGlobal*	Returns an object corresponding to the 64-bit application specified by <string>. On a 32-bit computer, this is equivalent to a normal application Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. Win

Examples

- `application "qna.exe" of parent folder of application "bigfix.exe" of the registry`

  Using the application of folder creation method, this example locates the 'sibling' application qna.exe provided it exists in the same folder in which the registered application 'bigfix.exe' is installed.

- `exists application "notepad.exe" of the windows folder`

  Using the application of folder creation method, this example locates the notepad application provided it exists in the Windows folder.

- `name of application "print" of key "HKEY_CLASSES_ROOT\.gif" of registry`
  - Returns the name of the application currently responsible for printing gif files.

- `name of application "print" of key "HKEY_CLASSES_ROOT\giffile" of the registry`
  - The same as above when the default value of the key HKEY_CLASSES_ROOT\.gif contains giffile.

These two examples demonstrate the method used by Windows to maintain file associations in the registry.

- `application of key "HKEY_CLASSES_ROOT\mailto" of the registry`
  - This example returns the application responsible for handling mailto requests in your web browser.

- `names of regapps`
  - Primarily used in QnA to obtain lists of applications installed under the "app path" key of the registry.

- `byte 0 of regapp "bigfix.exe" = 77`
  - TRUE if the first byte in the specified file is ASCII 77.

- `regapp "IEXPLORE.EXE"`
  - Returns the application object associated with the named registry key. Checks to see if the executable exists and if so, returns the application object. Case is ignored.

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

Folder

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

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

Creation Methods

Key Phrase	*Form*	Description
<registry key value> as folder	*Cast*	If the value in the registry is a string, and the string points to an existing folder, a folder object is returned. Win, WM
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

Key Phrase	*Form*	Description
application folder <string> of <registry key>	*Named*	Synonym for pathname of parent folder of regapp <string>. Win, WM
application folder <string> of <registry>	*Named*	Creates a folder object for the name given. Name is used to search through AppPaths of the registry. Application doesn't have to exist. Folder must exist. Win, WM
application folder of <registry key>	*Plain*	Creates a folder object for the name given. If the registry key has a "shell\open\command\" subkey and the unnamed value points to an executable, this will return the parent folder of the executable if the application and folder exist. Win, WM
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
csidl folder <integer>	*NumberedGlobal*	Returns the csidl folder corresponding to the specified integer. The windows SHGetSpecialFolderLocation API is used to look up paths to special folders, which are identified by passing the specified integer as the second argument of the API call. These values and their meaning are described in the windows ShlObj.h include file found in the development sdk. • Note that some of these folders do not exist in the Local System context. Win:7.0, WM
descendant folder of <folder>	*Plain*	Returns the descendant folders, recursively, of the given folder. The folder equivalent of "descendants of <folder>". Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
find folder <string> of <folder>	*Named*	Finds the folder with the given wildcard name inside another folder. A wildcard string uses an asterisk to stand for any number of characters (incuding zero), and a question mark to stand for exactly one character. Thus A??.txt would match All.txt and AXE.txt but not all.txt or a.txt. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
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 <drive>	*Named*	Creates a folder object for the name provided if it exists on the drive provided. Win, WM

Key Phrase	*Form*	Description
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 of <folder>	*Plain*	Iterates through the sub-folders of the folder object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
install folder <integer>	*NumberedGlobal*	Creates a folder object corresponding to the number provided. The placement of some system folders can be found using numbers that have been associated with those folders. • For information on the integer values and their meaning, see the section on Folders on Windows Devices in the Resources chapter at the end of this guide. Win, WM
parent folder of <filesystem object>	*Plain*	The folder containing the specified file or folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
root folder of <drive>	*Plain*	Creates a folder object for the root of the given drive. Win, WM
system wow64 folder	*PlainGlobal*	Returns a filesystem object corresponding to a "Windows On Windows 64" system folder, which does not exist on 32-bit Windows. You can find out more about the WOW64 system folder at the Microsoft site: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/getsystemwow64directory.asp. Win, WM
system x32 folder	*PlainGlobal*	Returns a filesystem object corresponding to a 32-bit system folder. On a 32-bit machine, this is equivalent to the normal system folder. Win, WM
system x64 folder	*PlainGlobal*	Returns a filesystem object corresponding to a 64-bit system folder. This is the same as the system folder, but with file system redirection disabled. For more information about file redirection, see the Microsoft site http://msdn.microsoft.com/library/default.asp?url=/library/en-us/win64/win64/file_system_redirector.asp. Win, WM
windows folder	*PlainGlobal*	Creates a folder object of the Windows folder. This is operating system dependent. Under Win98 this is usually c:\Windows. Win, WM
x32 folder <string>	*NamedGlobal*	Returns a filesystem object corresponding to a 32-bit folder with the specified pathname. Win

Key Phrase	*Form*	Description
x64 folder <string>	*NamedGlobal*	Returns a filesystem object corresponding to a 64-bit folder with the given pathname. On a 32-bit computer, this is equivalent to a normal folder Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. This action is transitive: any resulting filesystem objects will also have redirection disabled. For example, "pathnames of files of x64 folder <path>" will disable redirection when locating the folder, iterating over the files in the folder and calculating pathnames. Win

Properties

Key Phrase	*Form*	Return Type	Description
application <string> of <folder>	*Named*	<application>	Returns an application object for the named file located in the folder. See application. Win, Lin, Sol, HPUX, AIX, WM, Ubu
descendant folder of <folder>	*Plain*	<folder>	Returns the descendant folders, recursively, of the given folder. The folder equivalent of "descendants of <folder>". Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
descendant of <folder>	*Plain*	<file>	Returns a list of all the descendant files of the specified folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
file <string> of <folder>	*Named*	<file>	Returns a file object for the named file located in the folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
file of <folder>	*Plain*	<file>	Iterates through the files of a folder returning file objects. When combined with a whose clause you can select files with specific properties. See file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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

Key Phrase	*Form*	Return Type	Description
find folder <string> of <folder>	*Named*	<folder>	Finds the folder with the given wildcard name inside another folder. A wildcard string uses an asterisk to stand for any number of characters (incuding zero), and a question mark to stand for exactly one character. Thus A??.txt would match All.txt and AXE.txt but not all.txt or a.txt. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
folder <string> of <folder>	*Named*	<folder>	Returns a folder object for the named sub-folder. Trailing slashes should be omitted from the name. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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
security descriptor of <folder>	*Plain*	<security descriptor>	Specifies the security descriptor associated with the specified folder. Win

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

- `pathname of csidl folder 26`

  Returns the path corresponding to CSIDL folder 26 (the application shared data folder, CSIDL_APPDATA).

- `exists folder "c:\program files"`

  Checks for the existence of the program files folder.

- `exists folder "fonts" of the windows folder`

  Returns TRUE if fonts is a subdirectory of the Windows directory.

- `install folder 11`

  Returns a folder object for system folder identified with this number.

- `exists file whose (name of it contains ".pdf") of folder "name"`

  Returns TRUE if some file in the folder has a name including the string ".pdf".

Drive

The drive object is available to inspect these aspects of the file system.

Creation Methods

Key Phrase	*Form*	Description
drive of <filesystem object>	*Plain*	Returns the drive associated with the specified file or folder (filesystem object). Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
file system type of <drive>	*Plain*	<string>	Value as reported by GetVolumeInformation. Win, WM
file_supports_encryption of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
file_supports_object_ids of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
file_supports_reparse_points of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
file_supports_sparse_files of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
file_volume_quotas of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
folder <string> of <drive>	*Named*	<folder>	Returns a folder object corresponding to the name given provided that folder exists on the drive. Win, WM
free space of <drive>	*Plain*	<integer>	Returns the number of unused bytes of storage for the drive. (Only available for fixed disks). Win, WM
fs_case_is_preserved of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
fs_case_sensitive of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
fs_file_compression of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM

Key Phrase	*Form*	Return Type	Description
fs_persistent_acls of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
fs_unicode_stored_on_disk of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
fs_vol_is_compressed of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win, WM
name of <drive>	*Plain*	<string>	Returns the name of the drive. Names look like 'c:' and 'D:'. Win, WM
numeric type of <drive>	*Plain*	<integer>	Returns the type of drive as an integer. Win, WM
root folder of <drive>	*Plain*	<folder>	Returns the folder corresponding to the root of the drive. Win, WM
total space of <drive>	*Plain*	<integer>	Returns the size in bytes of the drive. (Only available for fixed disks). Win, WM
type of <drive>	*Plain*	<string>	Returns the type of drive as a string. Win, WM

NOTE:

The drive object does not exist if the file is located on a file server. The expression
drive of file "command.com" of folder "\\oak\c\windows"
will fail even though the file exists. Drive objects do not exist for shared files and shared folders unless they have been mapped as a drive letter. The name of drives may be upper or lower case. The type of drive can be inspected. The values as string and integer are:

Type of drive	Numeric type
DRIVE_UNKNOWN	0
DRIVE_NO_ROOT_DIR	1
DRIVE_REMOVABLE	2
DRIVE_FIXED	3
DRIVE_REMOTE	4
DRIVE_CDROM	5
DRIVE_RAMDISK	6

Examples

- `free space of drive "c:" < 1000000`
  - Returns TRUE if there is less than one million bytes of space left on drive C.
- `name of drive of regapp "vshield.exe" as lowercase = "e:"`
  - Returns TRUE if the application exists on drive E.
- `numeric type of drive "e:" = 5`
  - Returns TRUE if drive E is a CD-ROM. (See notes).
- `total space of drive "c:" > 2000000000`
  - Returns TRUE when the drive is capable of holding more than 2 billion bytes.
- `type of drive of the system folder = "DRIVE_FIXED"`
  - Returns TRUE if the system folder is on a fixed disk drive.

File Shortcut

Shortcuts to files can be constructed in the file system. The shortcut contains some additional properties that can be inspected.

Creation Methods

Key Phrase	*Form*	Description
shortcut of <file>	*Plain*	Creates a shortcut object for the file. If the file is not a shortcut, this property does not exist. Win

Properties

Key Phrase	*Form*	Return Type	Description
argument string of <file shortcut>	*Plain*	<string>	Returns the arguments that are passed to the application to which the shortcut points when the user attempts to open the shortcut. Win
icon index of <file shortcut>	*Plain*	<integer>	The index number of the icon in the file containing the icon associated with the shortcut. Win
icon pathname of <file shortcut>	*Plain*	<string>	The full pathname of the file containing the icon associated with the shortcut. Win

Key Phrase	*Form*	Return Type	Description
pathname of <file shortcut>	*Plain*	<string>	Returns the full pathname of the object to which the shortcut points. Win
start in pathname of <file shortcut>	*Plain*	<string>	Returns the full pathname the system sets the current directory when the user launches the shortcut. Win

Examples

- `pathname of shortcut of file "BigFix.lnk" of (value "Common Desktop" of key "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders" of registry as folder)`
  - Returns the pathname associated with shortcut.

- `pathname of parent folder of system folder = pathname of windows folder`
  - Checks that the system folder is located inside the Windows folder.

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.

Examples

- `exists Section "General" of file "oeminfo.ini" of system folder`
  - Returns TRUE if a section named "General" appears in the named "ini" file.

- `key "Manufacturer" of section "General" of file "oeminfo.ini" of system folder`
  - Returns the name of the computer manufacturer, such as "Dell Computer Corporation".

File Content

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

Creation Methods

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

Properties

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

Operators

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

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

Examples

- `content of file "oeminfo.ini" of system folder contains "Manufacturer=Dell"`
  - Returns TRUE if the exact character sequence "Dell" is located in the file.

- `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
file version of <file>	*Plain*	Creates a version object associated with the FILEVERSION property of the file. Win, WM
product version of <file>	*Plain*	Creates a version object associated with the PRODUCTVERSION property of the file. Win, WM

Key Phrase	*Form*	Description
raw file version of <file>	*Plain*	Same as file version, but allows a workaround for anomalous behavior on Windows systems with the Windows language pack installed (the MUI). Win, WM
raw product version of <file>	*Plain*	Same as product version, but allows a workaround for anomalous behavior on Windows systems with the Windows language pack installed (the MUI). Win, WM
raw version of <file>	*Plain*	Same as version, but allows a workaround for anomalous behavior on Windows systems with the Windows language pack installed (the MUI). Win, WM
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 <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 <file>	*Plain*	Shorthand for file version of <file>. Win, , WM
version of <operating system>	*Plain*	Returns the version of the operating system. Win:8.0, Mac:8.0
version of <service>	*Plain*	This Inspector takes the specified service property and retrieves its version (file version). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu

NOTE: Using the numeric version data is better than identifying an application based on version block strings. If you know the numeric version information and that the developer has identified each release of his application uniquely, then this is the way to proceed. It requires far less overhead than the other method. Furthermore, if you know that the numeric version data is monotonically increasing then you can compare their values using the special comparison operators.

Examples

- `file version of file "Winsock.dll" of windows folder = "4.0.0.1111"`

  Returns TRUE if the dll has the specified version number.

- `product version of regapp "bigfix.exe" > version "1.0.21"`

  TRUE if the application has a version of 1.0.22 or higher. FALSE if the application has a version of 1.0.21 or less.

- `product version of file "qna.exe" of parent folder of regapp "bigfix.exe" = product version of regapp "bigfix.exe"`

  Verifies the existence a co-executable located in the same folder with the proper version.

- `product version of regapp "bigfix.exe" > version "1.0.21"`

  Returns TRUE if the application has a version of 1.0.22 or higher, and FALSE if the application has a version of 1.0.21 or less.

- `version of file "Winsock.dll" of windows folder = "4.0.0.1111"`

  The plain version inspector is a shorthand for file version.

File Version Block

You can inspect the version blocks of a file. There may be several language-specific version blocks. Version blocks contain version and name information in a human readable form for the specified language. This is the information that Windows displays in the file properties dialog. This technique uses string values and has a limited array of comparators. For better speed, utility and compactness see the version object.

Creation Methods

Key Phrase	*Form*	Description
first raw version block of <file>	*Plain*	Returns the first version block directly from a PE file. If the first block is sufficient for your purposes, use this version inspector for best speed. Win, WM
only raw version block of <file>	*Plain*	Returns the only version block directly from a PE file. Win, WM
only version block of <file>	*Plain*	Most applications only have 1 version block. This inspector allows language independent access when there is only one version block present. Win, WM
raw version block <integer> of <file>	*Numbered*	Returns the numbered version block directly from a PE file. Win, WM

Key Phrase	*Form*	Description
raw version block <string> of <file>	*Named*	Returns the named version block directly from a PE file. Win, WM
raw version block of <file>	*Plain*	Returns the version block directly from a PE file. Win, WM
version block <integer> of <file>	*Numbered*	You can identify the particular version block you are looking up by ordinal number. 'Version block 1' is equivalent to 'Only Version block'. Win, WM
version block <string> of <file>	*Named*	You can identify the particular version block you are looking up by name. The name you provide should match the id string of the version block. Win, WM
version block of <file>	*Plain*	Iterates through the version blocks of a file. Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
codepage of <file version block>	*Plain*	<string>	A string representation of the codepage portion of the id of this version block. See notes for known codepage strings. For example, 'Unicode'. Win, WM
id of <file version block>	*Plain*	<string>	A string representation containing both the language and codepage of this version block. The format is 8 hex digits, 4 of the codepage concatenated with 4 of the language. For example, '040904b0'. See notes for known values. Win, WM
language of <file version block>	*Plain*	<string>	A string representation of the language portion of the id of this version block. For example, 'English (United States)'. See notes for known values. Win, WM
value <string> of <file version block>	*Named*	<string>	Returns a string corresponding to the name provided. Values have names such as 'CompanyName', 'FileDescription', 'FileVersion'. Win, WM

NOTE:

The value, ID, language and codepage properties of the file version block have the following typical values:

value:

Each application can define its own set of values. Standard values include:

- CompanyName
- FileDescription *
- FileVersion *
- InternalName
- LegalCopyright *
- OriginalFilename
- ProductName
- ProductVersion
- Comments
- LegalTrademarks
- PrivateBuild
- SpecialBuild

* As displayed on the version property sheet of the properties of a file.

id:

The version block id is an eight character string. The left 4 characters of the string identify the language while the right 4 characters of the string identify the codepage of a version block. When looking up a version block by its name, you specify the id as a string. The id's of version blocks are case insensitive.

language:

The language inspector returns the full language name. Language names are found using the left 4 hex characters of the id. Thus if the id of the version block is '040904b0', then the language returned would be 'English (United States)'.

Here are some sample language identifiers (left 4 hex chars):

0000	Language Neutral	0800	Language Neutral	1801	Arabic (Morocco)
0400	Process Default Language	0801	Arabic (Iraq)	1809	English (Ireland)
0401	Arabic (Saudi Arabia)	0804	Chinese (PRC)	180A	Spanish (Panama)
0402	Bulgarian	0807	German (Swiss)	1C01	Arabic (Tunisia)
0403	Catalan	0809	English (British)	1C09	English (South Africa)
0404	Chinese (Taiwan)	080A	Spanish (Mexican)	1C0A	Spanish (Dominican Republic)
0405	Czech	080C	French (Belgian)	2001	Arabic (Oman)
0406	Danish	0810	Italian (Swiss)	2009	English (Jamaica)
0407	German (Standard)	0812	Korean (Johab)	200A	Spanish (Venezuela)
0408	Greek	0813	Dutch (Belgian)	2401	Arabic (Yemen)
0409	English (United States)	0814	Norwegian (Nynorsk)	2409	English (Caribbean)
040A	Spanish (traditional Sort)	0816	Portuguese (Standard)	240A	Spanish (Colombia)
040B	Finnish	081A	Serbian (Latin)	2801	Arabic (Syria)
040C	French (Standard)	081D	Swedish (Finland)	2809	English (Belize)
040E	Hungarian	0C01	Arabic (Egypt)	280A	Spanish (Peru)
040F	Icelandic	0C04	Chinese (Hong Kong)	2C01	Arabic (Jordan)
0410	Italian (Standard)	0C07	German (Austrian)	2C09	English (Trinidad)
0411	Japanese	0C09	English (Australian)	2C0A	Spanish (Argentina)
0412	Korean	0C0A	Spanish (Modern Sort)	3001	Arabic (Lebanon)
0413	Dutch (Standard)	0C0C	French (Canadian)	300A	Spanish (Ecuador)
0414	Norwegian (Bokmal)	0C1A	Serbian (Cyrillic)	3401	Arabic (Kuwait)
0415	Polish	1001	Arabic (Lybia)	340A	Spanish (Chile)
0416	Portuguese (Brazilian)	1004	Chinese (Singapore)	3801	Arabic (U.A.E)
0418	Romanian	1007	German (Luxembourg)	380A	Spanish (Uruguay)
0419	Russian	1009	English (Canadian)	3C01	Arabic (Bahrain)
041A	Croation	100A	Spanish (Guatemala)	3C0A	Spanish (Paraguay)
041B	Slovak	100C	French (Swiss)	4001	Arabic (Qatar)
041D	Swedish	1401	Arabic (Algeria)	400A	Spanish (Bolivia)
041F	Turkish	1407	German (Liechtenstein)	440A	Spanish (El Salvador)
0423	Belarusian	1409	English (New Zealand)	480A	Spanish (Honduras)
0424	Slovene	140A	Spanish (Costa Rica)	4C0A	Spanish (Nicaragua)
042D	Basque	140C	French (Luxembourg)	500A	Spanish (Puerto Rico)

codepage:
The right 4 characters of the id correspond to the codepage as in these examples:

0000	7-bit ASCII	04B0	Unicode	04E5	Windows, Greek
03A4	Windows, Japan	0400	Windows, Latin-2	04E7	Windows, Hebrew
03B5	Windows, Korean	04E3	Windows, Cyrillic	2710	Macintosh, Roman
03B6	Windows, Taiwan	04E4	Windows, Multilingual	2711	Macintosh, Japanese

The string 'Unknown' is returned for an unidentified language or codepage.

Examples

- `value "CompanyName" of version block 1 of regapp "bigfix.exe" = "BigFix Inc."`

  Returns TRUE if the "CompanyName" value of the given file's version block equals the specified string.

- `exists version block "040904B0" of regapp "bigfix.exe"`

  Returns TRUE if the designated version block exists. The case of the name of the version block is ignored.

- `codepage of only version block of regapp "bigfix.exe" is "Unicode"`

  Returns TRUE if the version block codepage for the specified file is unicode.

- `id of only version block of regapp "bigfix.exe" is "040904b0"`

  Returns TRUE if the given file's version block id is the specified string.

- `language of version block 1 of regapp "bigfix.exe" = "English (United States)"`

  Returns TRUE if the given file's version block language is as specified.

- `value "FileVersion" of version block 1 of regapp "bigfix.exe" as version`

  When casting a string value to a version, the parser skips through the string until it identifies something that can be interpreted as a version. This is convenient for extracting version numbers from strings containing added text.

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

Examples

- `line 2 of (file "printmon.inf" of system folder)`

  Returns the second line of the specified file.

- `lines containing "Signature]" of file "mmdriver.inf" of system folder`

  Returns a list corresponding to the top-level sections involving signatures in the specified .inf file.

- `lines of file "mmdriver.inf" of system folder`

  Returns a list of the lines of the specified file.

- `lines starting with "[" of file "mmdriver.inf" of system folder`

  Returns a list corresponding to all the top-level sections (lines that start with "[") in the specified .inf file.

- `next line of line containing "[mciavi]" of file "mmdriver.inf" of system folder`

  Returns the line after the one containing "[mciavi]", which is a string such as "1:MSVFW32.DLL".

- `previous line of previous line of previous line of line containing "[mciavi]" of file "mmdriver.inf" of system folder`

  Returns the line 3 lines previous to the one containing the specified phrase in the given file.

- `line number of line containing "[mciavi]" of file "mmdriver.inf" of system folder`

  Returns the line number of the specified line in the given file.

- `next line of line containing "[mciavi]" of file "mmdriver.inf" of system folder`

  Returns the line after the one containing "[mciavi]", which is a string such as "1:MSVFW32.DLL".

- `previous line of previous line of previous line of line containing "[mciavi]" of file "mmdriver.inf" of system folder`

  Returns the line 3 lines previous to the one containing the specified phrase in the given file.

Xml Dom Document

These are the Inspectors for the XML Document Object Module (DOM) for specified XML files. The console uses MSXML 6.0 if it is available. Otherwise it falls back to 4.0. The console requires at least 4.0 since 3.0 does not provide XML schema validation.

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

Creation Methods

Key Phrase	*Form*	Description
xml document of <file>	*Plain*	Returns the XML Document Object Module (DOM) for the specified file. Win, WM

Download Storage Folder

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

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

Creation Methods

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

Properties

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

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
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

Key Phrase	*Form*	Return Type	Description
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

Examples

- `date of bios = "09/16/97"`
  - Returns TRUE if the BIOS date matches the value provided.
- `following text of last "/" of date of bios`
  - Returns the year of the bios as a string. For "09/07/99" it would return "99".
- `preceding text of first "/" of date of bios`
  - Returns the month of the bios date as a string.
- `preceding text of first "/" of following text of first "/" of date of bios`
  - Returns the day of the bios date as a string.
- `version of bios as lowercase contains "phoenix"`
  - Returns TRUE if the version string contains "phoenix", "PHOENIX" or "Phoenix".

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
boot time of <operating system>	*Plain*	<time>	Returns the time of the last restart. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
build number high of <operating system>	*Plain*	<integer>	Numeric representation of the most significant 16 bits of the build number. Win, WM
build number low of <operating system>	*Plain*	<integer>	Numeric representation of the least significant 16 bits of the build number. Win, WM
csd version of <operating system>	*Plain*	<string>	Returns the Corrective Service Disk version of the operating system. The szCSDVersion as returned by the GetVersionEx system call. The format varies depending on the installed service packs. For WinNT it contains a string such as "Service Pack 3", for Win95 it can contain a string such as "B". Win, WM
ia64 of <operating system>	*Plain*	<boolean>	Returns TRUE iff the BES Client is running on Itanium. Win:7.0, WM
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

Key Phrase	*Form*	Return Type	Description
major version of <operating system>	*Plain*	<integer>	Returns integer which is the dwMajorVersion returned by the GetVersionEx system call. Note that while the WinNT major version tracks the release (3 for 3.51, 4 for 4.0, and 5.0 for Windows 2000, 5.1 for Windows XP), the major version for Win95 and Win98 is always 4. Win, WM
metric <integer> of <operating system>	*Numbered*	<integer>	This inspector uses the windows GetSystemMetrics API. The integer constants and their meaning are defined by Microsoft. For example, the integer 87 indicates that the operating system is a media center edition. The integer constants are defined in WinUser.h. Win
minor version of <operating system>	*Plain*	<integer>	Numeric representation of the minor version of the operating system. Win, WM
name of <operating system>	*Plain*	<string>	Returns the name of the operating system as a string. Names might include Win98, WinNT, and more. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
performance counter frequency of <operating system>	*Plain*	<hertz>	The rate at which the performance counter is being incremented (per second). Win, WM
performance counter of <operating system>	*Plain*	<integer>	Retrieves a 64-bit performance counter value. Win, WM
platform id of <operating system>	*Plain*	<integer>	Returns the dwPlatformId as returned by the GetVersionEx system call. Possible values are 1 (Win95/95) and 2 (WinNT). Win, WM
product info numeric of <operating system>	*Plain*	<integer>	This Windows-specific inspector returns the integer from the Windows GetProductInfo API. The inspector only provides meaningful results for Windows Vista and newer versions of the OS. The major/minor version of the OS must be 6.0 or greater for the result to be meaningful. For more information, refer to the Microsoft article at http://msdn2.microsoft.com/en-us/library/ms724358(VS.85).aspx. Win:7.0, WM

Key Phrase	*Form*	Return Type	Description
product info string of <operating system>	*Plain*	<string>	On Windows versions 6.0 and newer (Vista minimum), this inspector returns a string derived from the GetProductInfo API. It will be one of the following values: • Unlicensed • Business • Cluster Server • Server Datacenter • Server Datacenter Core • Enterprise • Server Enterprise • Server Enterprise Core • Server Enterprise Itanium • Home Basic • Home Server • Server for Small Business • Small Business Server • Small Business Server Premium • Server Standard • Server Standard Core • Starter • Storage Server Enterprise • Storage Server Standard • Storage Server Workgroup • Ultimate • Web Server • Unknown. Win:7.0, WM
product type of <operating system>	*Plain*	<operating system product type>	Returns the product type of the operating system, which includes Workstations, Domain Controllers and Servers. Win, WM
release of <operating system>	*Plain*	<string>	Information about the release of the operating system, formatted as a <version> on the Macintosh, but a <string> on UNIX and Windows. Win, Lin, Sol, HPUX, AIX, , WM, Ubu
service pack major version of <operating system>	*Plain*	<integer>	Returns the major version number of the current service pack of the specified OS. Win, WM

Key Phrase	*Form*	Return Type	Description
service pack minor version of <operating system>	*Plain*	<integer>	Returns the minor version number of the current service pack of the specified OS. Win, WM
suite mask of <operating system>	*Plain*	<operating system suite mask>	Returns the bit-mapped suite mask for the operating system, which contains further fine-grain information about the version. Win, WM
unix of <operating system>	*Plain*	<boolean>	Returns TRUE if the local computer is a UNIX system. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
uptime of <operating system>	*Plain*	<time interval>	Returns a time interval that represents the elapsed time since the operating system was last booted. • Note: Depending on the notebook, this interval may not include time spent in hibernation. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
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
x64 of <operating system>	*Plain*	<boolean>	Returns TRUE if the current operating system is 64-bits. Win, WM

Examples

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

- `build number high of operating system = 1027`
  - Returns TRUE if the high word of the build number = 0403 hex.

- `build number low of operating system = 1212`
  - Returns TRUE if the low word of the build number = 04BC hex.

- `csd version of the operating system = "B"`
  - Returns TRUE on a Win95 System with Corrective Service Disk version = "B".

- `major version of operating system = 4`
  - Returns TRUE if the major version (before the dot) is 4, such as 4.1, 4.2, and others.

- `Metric 87 of operating system`
  - Returns TRUE if the OS is a Media Center Edition.

- `minor version of operating system = 0`
  - Returns TRUE if the minor part of a version number (after the dot) is 0, such as 4.0, 5.0, and others.

- `name of operating system = "WinXP"`
  - Returns TRUE on a WinXP System.

- `platform id of operating system = 1`
  - Returns TRUE on a Win95 System.

- `product info numeric of operating system`
  - Returns an integer value such as 3, but only if the major/minor version of the OS is 6.0 or greater. On previous versions, this inspector returns 0 (zero).

- `product info string of operating system`
  - Returns a string such as 'Small Business Server' on a Windows OS greater than version 6.0. Otherwise, it returns 'Unknown'.

- `release of operating system = "OSR2.1"`
  - Returns TRUE if the Win95 computer is running under operating system release 2.1.

Processor

The processor object is used to identify the number and properties of processors in the system. You can identify the manufacturer of the CPU as well as the speed and other features. Many operating systems provide for multiple processors. You can inspect any one of them by their ordinal number.

- For more information on Windows processors, see the Resource section at the end of this guide.

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
brand id of <processor>	*Plain*	<integer>	This inspector returns the integer known as the brand id, returned from the assembly language cpuid extended instruction. Win, WM
brand string of <processor>	*Plain*	<string>	Returns the vendor-defined brand names for newer processors. Win, WM
extended family of <processor>	*Plain*	<integer>	Integer representing the extended family of CPU. See the notes for the meaning of these numbers. Win, WM
extended model of <processor>	*Plain*	<integer>	Integer representing the extended model of CPU. See the notes for the meaning of these numbers. Win, WM
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, and others. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
family of <processor>	*Plain*	<integer>	Returns an integer representing the family of the CPU. See the notes for the meaning of these numbers. • Note: As of BES 6.0, this Inspector returns a string on Solaris and AIX computers. Win, Lin, , , WM, Ubu
feature mask of <processor>	*Plain*	<integer>	Returns the feature flags from the CPUID instruction. The feature mask contains bits that identify extra features the processor may provide such as MMX support or if the Processor ID is enabled on the processor. Win, WM
model of <processor>	*Plain*	<integer>	Returns the model number of the CPU. • Note: On Solaris, HPUX and AIX computers, this Inspector returns a <string> as of BES 6.0. Win, Lin, WM, Ubu
speed of <processor>	*Plain*	<hertz>	Returns the speed of the processor in Hertz. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
stepping of <processor>	*Plain*	<integer>	Returns the stepping number of the processor. This item can be helpful in identifying very specific processor features or limitations. Win, Lin, WM, Ubu

Key Phrase	*Form*	Return Type	Description
type of <processor>	*Plain*	<integer>	Numeric type of the CPU. Values include: • 0 - standard • 1 - overdrive • 2 - dual CPU capable • 3 - reserved • Note: this Inspector returns a <string> type as of BES version 6.0 on UNIX machines and version 5.1 on the Macintosh. Win, WM
vendor name of <processor>	*Plain*	<string>	The manufacturer of the CPU. Names include: • GenuineIntel • AuthenticAMD • CyrixInstead • CentaurHauls • AmbiguousCPU. Win, Lin, WM, Ubu

Examples

- `number of processors > 1`

  Returns TRUE if the computer is a multi-processor system.

- `family name of main processor = "Pentium III"`

  Returns TRUE for a computer with a Pentium III cpu.

- `bit 18 of feature mask of main processor`

  Returns TRUE if the processor ID feature is enabled on this processor.

- `model of processor = 15`

  Returns TRUE if the processor model is 15. For more information, see the Intel and AMD documentation on the Model number returned by the CPUID instruction.

- `speed of main processor < 2000 * MHz`

  Returns TRUE is the cpu is slower than 2Ghz.

- `types of processors`

  Returns a list of processor types for each CPU on the local machine.

- `vendor name of main processor = "GenuineIntel"`

  Returns TRUE for an Intel processor chip.

- `number of processors whose (vendor name of it = "AuthenticAMD" or vendor name of it = "CyrixInstead" or vendor name of it = "CentaurHauls") = 1`

  Returns TRUE for a single processor system with the given vendors.

Ram

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
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.

Active Device

On Windows NT systems (including XP and 2K), these Inspectors returns a set of objects corresponding to the active devices on the machine, modems, graphics cards, printers, and more.

Creation Methods

Key Phrase	*Form*	Description
active device	*PlainGlobal*	Creates a list of all active devices found using the Configuration Manager SetupDiGetClassDevs NT API. The locations of active devices are relative to the Windows Directory. Win

Properties

Key Phrase	*Form*	Return Type	Description
class of <active device>	*Plain*	<string>	Returns the name of the class of the active device provided. Win
description of <active device>	*Plain*	<string>	Returns the description of the active device provided. Win
driver key of <active device>	*Plain*	<registry key>	The key identified by adding the value of 'driver key value name of active device' to HKLM\System\CurrentControlSet\Control\Class\. Win
driver key value name of <active device>	*Plain*	<string>	Returns the driver key value name of the active device provided. Win
friendly name of <active device>	*Plain*	<string>	Returns the friendly name of the active device. Win
hardware id of <active device>	*Plain*	<string>	Returns the hardware id of the active device provided. Win
location information of <active device>	*Plain*	<string>	Returns a string containing information about the bus location of the device. Win
manufacturer of <active device>	*Plain*	<string>	Returns the manufactures string of the active device. Win
problem id of <active device>	*Plain*	<integer>	Configuration manager defined number describing device installation or use problems. Returned by call to CM_Get_DevInst_Status. Win
service key value name of <active device>	*Plain*	<string>	Returns the service key value name. Win
status of <active device>	*Plain*	<integer>	Configuration manager defined status bits conveying device driver status. Win

Examples

- `number of active devices`
  - Returns the number of active devices as determined by the Configuration Manager.
- `exists active device file "system32\ntoskrnl.exe"`
  - Returns TRUE if ntoskrnl.exe exists in the System32 folder of the Windows folder.
- `exists active device file "C:\WINNT\System32\ntoskrnl.exe"`
  - Returns FALSE because this Inspector path is relative to the Windows directory.
- `exists active device whose (class of it = "Display")`
  - Returns TRUE if there is an active device named "Display".
- `(description of it) of active devices whose ( class of it ="Display")`
  - Provides a list of the descriptions of the active display devices.
- `exists service key value name whose (it = "PGPdisk" ) of active devices`
  - Returns TRUE if the designated service key value name exists on this system.

Local Mssql Database

These Inspectors retrieve the properties of the MS SQL databases on the local machine.

Creation Methods

Key Phrase	*Form*	Description
local mssql database	*PlainGlobal*	Creates the local MSSQL database objects. Win
local mssql database <string>	*NamedGlobal*	Creates the local Microsoft SQL (MSSQL) database object identified by the name provided. Win

Properties

Key Phrase	*Form*	Return Type	Description
audit level of <local mssql database>	*Plain*	<integer>	Returns the integer audit level of the MSSQL database. Win
has blank sa password of <local mssql database>	*Plain*	<boolean>	Returns TRUE if the MSSQL database sa account has a blank password. Win

Key Phrase	*Form*	Return Type	Description
instance name of <local mssql database>	*Plain*	<string>	Returns the name of the MSSQL database. Win
login mode of <local mssql database>	*Plain*	<integer>	Returns the login mode of the MSSQL database. Win
running of <local mssql database>	*Plain*	<boolean>	Returns a boolean indicating if the MSSQL database is running. Win

Service

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

Creation Methods

Key Phrase	*Form*	Description
all running service	*PlainGlobal*	Returns a list of all running Windows 32 services, such as ATAPI and BESRelay. Win:8.1
all service	*PlainGlobal*	Returns a list of all Windows 32 services, including the BESRelay. Win:8.1
driver running service	*PlainGlobal*	Returns a list of all running Windows 32 system drivers. Win:8.1
driver service	*PlainGlobal*	Returns a list of all Windows 32 system drivers. Win:8.1
main gather service	*PlainGlobal*	Returns a service object for the main gathering service, typically located on the main server. • Note: On a Macintosh, returns <nothing>. Included for compatibility. Win, Lin, Sol, HPUX, AIX, , Ubu
relay service	*PlainGlobal*	Returns a service object for the relay component of BES. • Note: On a Macintosh, this returns <nothing>. Win, Lin, Sol, HPUX, AIX, , Ubu
running service	*PlainGlobal*	Creates objects corresponding to all the running services. Win

Key Phrase	*Form*	Description
running service <string>	*NamedGlobal*	Creates the running service object for the specified name. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
service	*PlainGlobal*	Creates objects for all the services. Win
service <string>	*NamedGlobal*	Returns the service object matching the name provided, regardless of its running state. • On a Macintosh, returns a <dummy> type. Win, Lin, Sol, HPUX, AIX, , Ubu
win32 running service	*PlainGlobal*	A synonym for 'running service'. This can be any running service from atapi to the BESRelay. Win:8.1
win32 service	*PlainGlobal*	A synonym for 'service'. This can be anything from atapi to the BESRelay. Win:8.1

Properties

Key Phrase	*Form*	Return Type	Description
<service> as string	*Cast*	<string>	Returns a string containing the Service name, Display name, and State of the service. Win
can interact with desktop of <service>	*Plain*	<boolean>	Indicates the system is configured to allow the service to interact with the desktop. Win
checkpoint of <service>	*Plain*	<integer>	Service specific value indicating its checkpoint state. Win
display name of <service>	*Plain*	<string>	Returns the display name of the service. Win
driver type of <service>	*Plain*	<boolean>	Returns true if the specified service is a driver type. Win:8.1
file of <service>	*Plain*	<file>	Returns a file object corresponding to the specified <service>. Win
image path of <service>	*Plain*	<string>	Returns the full path to the service executable. Win

Key Phrase	*Form*	Return Type	Description
login account of <service>	*Plain*	<string>	Returns the login account under which the service is configured to run. Win
security descriptor of <service>	*Plain*	<security descriptor>	This Windows-specific Inspector returns a security descriptor for the specified service. Win:7.0
service name of <service>	*Plain*	<string>	Returns the name of the service. Win
service specific exit code of <service>	*Plain*	<integer>	Service specific exit code. Win
start type of <service>	*Plain*	<string>	Returns a string that represents the service startup configuration. It describes when the driver is loaded, which can be one of: • boot: started by OS loader (usually these are needed to launch the OS). • system: started during OS initialization (used by PnP drivers that do device detection after the loader is finished). • auto: started by the Service Control Manager (SCM). • demand: started on demand, either by PnP manager when a device is enumerated or by the SCM in response to user demand. • disabled: can't be started (used to temporarily disable driver services). Win
state of <service>	*Plain*	<string>	Returns one of Continuing, Pausing, Paused, Running, Starting, Stopping, Stopped, Unknown. Win, Lin, Sol, HPUX, AIX, Ubu
version of <service>	*Plain*	<version>	This Inspector takes the specified service property and retrieves its version (file version). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Ubu
win32 exit code of <service>	*Plain*	<integer>	Service specific Win32 exit code. Win
win32 type of <service>	*Plain*	<boolean>	Returns true if the specified service is a Win32 type. Win:8.1

Examples

- `running services`

  Returns a list of all the currently running services.

- `expand environment string of ( image path of service "AudioSrv")`

  Returns a pathname for the Windows Audio service, such as C:\WINDOWS\System32\svchost.exe -k netsvcs.

- `(DISPLAY name of it, security descriptor of it ) of service "TapiSrv"`

  Returns a value such as 'Telephony, D:(A;OICI;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;OICI;CCLCSWRPLO;;;BU)'.

- `version of service "BESClient"`

  Returns a version number, such as '8.0.584.0'.

Process

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

Creation Methods

Key Phrase	*Form*	Description
process	*PlainGlobal*	Returns all process objects currently running. Win:8.0, Lin, Sol, HPUX, AIX, Mac, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
base priority of <process>	*Plain*	<priority class>	Refers to the base priority of the process, which determines the order in which threads are scheduled for the processor. • The base priority is set by the process code (not the OS), however the OS sets and changes the dynamic priorities of threads in the process within the range of the base. • Use the Task Manager to change the base priority of processes. Win:8.0

Key Phrase	*Form*	Return Type	Description
creation time of <process>	*Plain*	<time>	Returns the creation time of the specified process as a time type. Win:8.0
gdi object count of <process>	*Plain*	<integer>	Returns the count of the GDI (Graphics Device Interface) objects associated with the specified process as an integer. Win:8.0
handle count of <process>	*Plain*	<integer>	Refers to the number of object handles in the object table of the specified process. Win:8.0
id of <process>	*Plain*	<integer>	Returns the integer ID of the specified process. Win:8.0, Lin, Sol, HPUX, AIX, Mac, Ubu
image file of <process>	*Plain*	<file>	Returns a file object representing the image file used to launch the process. In some cases, this subproperty may not exist. For example, the "System" process does not have an image file. Win:8.0
io other count of <process>	*Plain*	<integer>	Refers to the number of I/O process operations that are neither reads nor writes, including file, network, and device I/O, but excluding I/O directed to the console input object handles. An example might be a control function. Win:8.0
io other size of <process>	*Plain*	<integer>	Refers to the number of bytes transferred in I/O operations generated by a process including file, network, and device I/Os but excluding reads, writes and bytes directed to the console input object. Win:8.0
io read count of <process>	*Plain*	<integer>	Refers to the number of read I/O operations generated by a process, including file, network, and device I/Os, but excluding I/O reads directed to the console input object handles. Win:8.0
io read size of <process>	*Plain*	<integer>	Refers to the number of bytes read in I/O operations generated by a process, including file, network, and device I/Os, but excluding read bytes directed to the console input object handles. Win:8.0

Key Phrase	*Form*	Return Type	Description
io write count of <process>	*Plain*	<integer>	Refers to the number of write I/O operations generated by a process, including file, network, and device I/Os, but excluding I/O writes directed to the console input object handles. Win:8.0
io write size of <process>	*Plain*	<integer>	Refers to the number of bytes written in I/O operations generated by a process, including file, network, and device I/Os, but excluding write bytes directed to the console input object handles. Win:8.0
kernel time of <process>	*Plain*	<time interval>	Returns the time in kernel mode, as a time interval. Win:8.0
name of <process>	*Plain*	<string>	Returns the name (as a string) of the specified process. Win:8.0, Lin, Sol, HPUX, AIX, Mac, Ubu
page fault count of <process>	*Plain*	<integer>	Returns the number of page faults generated by the specified process. Win:8.0
page file usage of <process>	*Plain*	<integer>	Returns the amount of page file space used by the specified process. This value is equal to the VMSize in TaskMgr. Win:8.0
peak page file usage of <process>	*Plain*	<integer>	Returns the maximum page file space used throughout the lifetime of the specified process. Win:8.0
peak working set size of <process>	*Plain*	<integer>	Returns the size of the peak working set of the specified process. Win:8.0
quota nonpaged pool usage of <process>	*Plain*	<integer>	Returns the quota amount of nonpaged pool usage for the specified process. Win:8.0
quota paged pool usage of <process>	*Plain*	<integer>	Returns the quota amount of paged pool usage for the specified process. Win:8.0
quota peak nonpaged pool usage of <process>	*Plain*	<integer>	Returns the peak quota amount of nonpaged pool usage for the specified process. Win:8.0

Key Phrase	*Form*	Return Type	Description
quota peak paged pool usage of <process>	*Plain*	<integer>	Returns the peak quota amount of paged pool usage for the specified process. Win:8.0
session id of <process>	*Plain*	<integer>	Returns the ID of the session under which the process is running. This requires Win2000 or later and corresponds to the 'Session ID' column in Task Manager. Win:8.0
user object count of <process>	*Plain*	<integer>	Returns the count of USER objects associated with the specified process as an integer. Win:8.0
user of <process>	*Plain*	<security identifier>	Returns a security identifier associated with the user of specified process. This Inspector requires Terminal Services or Win2000+, and may not be available if the inspector application does not have sufficient permissions. This object corresponds to the 'User Name' column in Task Manager. Win:8.0
user time of <process>	*Plain*	<time interval>	Returns the time in user mode as a time interval. Win:8.0
working set size of <process>	*Plain*	<integer>	Returns the amount of memory (as an integer number of bytes) required by a process for efficient execution under an operating system that uses paged memory. Thrashing may occur in systems without sufficient memory. This value allow you to monitor the changing memory requirements of the specified process. Win:8.0
wow64 of <process>	*Plain*	<boolean>	Returns a boolean TRUE if the specified process is a 32-bit process running in WOW64 compatibility mode. Always returns FALSE on pre-WinXP SP2. Win:8.0

Examples

- `names of processes whose (pid of it < 20)`

  Returns a list of all process with an ID less than 20.

Language

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

Creation Methods

Key Phrase	*Form*	Description
system locale	*PlainGlobal*	Determines which bitmap fonts, and OEM, ANSI, and MAC code pages are defaults for the system. This only affects applications that are not fully Unicode. Win, Lin, Sol, HPUX, AIX, WM, Ubu
system ui language	*PlainGlobal*	Determines the default language of menus and dialogs, messages, INF files, and help files. Win, Lin, Sol, HPUX, AIX, WM, Ubu
user locale	*PlainGlobal*	Determines which settings are used for formatting dates, times, currency, and numbers as a default for each user. Also determines the sort order for sorting text. Win, WM
user ui language	*PlainGlobal*	Non-MUI: Same as system UI Language. • MUI: Determines the language of menus and dialogs, messages, and help files. Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
<language> as string	*Cast*	<string>	Returns the language of the system locale. Win, Lin, Sol, HPUX, AIX, WM, Ubu
primary language of <language>	*Plain*	<primary language>	Extracts the primary language identifier from a language. Win, Lin, Sol, HPUX, AIX, WM, Ubu

Examples

- `system locale as string`

  Returns English (United States) if the system locale is US English.

Primary Language

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

Creation Methods

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

Properties

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

Examples

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

Operating System Product Type

These Inspectors return the product type of the operating system, which includes Workstations, Domain Controllers and Servers.

Creation Methods

Key Phrase	*Form*	Description
nt domain controller product type	*PlainGlobal*	Returns an object corresponding to OS product type of nt domain controller. Win
nt server product type	*PlainGlobal*	Returns an object corresponding to OS product type of nt server. Win
nt workstation product type	*PlainGlobal*	Returns an object corresponding to OS product type of nt workstation. Win
operating system product type <integer>	*NumberedGlobal*	Returns an object corresponding to the numbered OS product type. Win, WM
product type of <operating system>	*Plain*	Returns the product type of the operating system, which includes Workstations, Domain Controllers and Servers. Win, WM

Operators

Key phrase	Return Type	Description
<operating system product type> = <operating system product type>	*<boolean>*	Compare two operating system product types for equality. Win, WM

Operating System Suite Mask

These Inspectors provide detailed information about the operating system version.

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

Creation Methods

Key Phrase	*Form*	Description
suite mask of <operating system>	*Plain*	Returns the bit-mapped suite mask for the operating system, which contains further fine-grain information about the version. Win, WM

Examples

- `suite mask of operating system`

  Returns the suite mask for the operating system.

Event Log

These Inspectors return information about the specified Windows Event logs, including the System, Security and the Application log.

Creation Methods

Key Phrase	*Form*	Description
application event log	*PlainGlobal*	Returns an object corresponding to an application event log, one of the event logs created by most Windows systems. Win
event log <string>	*NamedGlobal*	Returns the event log object with the specified name. Win
security event log	*PlainGlobal*	Returns an event log object for the security event log. Win
system event log	*PlainGlobal*	Returns a system event log, which records OS or component events, such as the failure of a bootup service. Win

Properties

Key Phrase	*Form*	Return Type	Description
oldest record number of <event log>	*Plain*	<integer>	Returns an integer corresponding to the oldest record number on the Client computer's event log. Win
record <integer> of <event log>	*Numbered*	<event log record>	Returns the nth record corresponding to the specified event log. Win
record count of <event log>	*Plain*	<integer>	Returns the record count for the specified event log. Win
record of <event log>	*Plain*	<event log record>	Returns the record corresponding to the specified event log, for instance the application or system event log. Win

Examples

- `exists application event log`
  - Returns TRUE if the application event log exists on this computer.

- `exists event log "Application"`
  - Returns TRUE if the application event log exists on this computer.

- `oldest record number of application event log`
  - Returns the number of the oldest record in the application event log. This is not the same as the record count.

- `exists record (oldest record number of it) of application event log`
  - Returns TRUE if there is an oldest record in the application event log.

- `record count of application event log`
  - Returns the current record count of the application event log.

Event Log Record

These Inspectors return individual records from the Windows Event logs, which record information about operating system events.

Creation Methods

Key Phrase	*Form*	Description
record <integer> of <event log>	*Numbered*	Returns the nth record corresponding to the specified event log. Win
record of <event log>	*Plain*	Retrieves the record from the event log. Win

Properties

Key Phrase	*Form*	Return Type	Description
category of <event log record>	*Plain*	<integer>	Returns the category of the specified event log, which is sometimes used to further describe the related action. Win
computer of <event log record>	*Plain*	<string>	Returns the name of the computer that has entered a record in the specified log. Win
description of <event log record>	*Plain*	<string>	Returns a human-readable description of the specified event log record. Win
event id of <event log record>	*Plain*	<integer>	Returns an integer corresponding to the ID of the specified record in the Client computer's event log. Win
event type of <event log record>	*Plain*	<event log event type>	Returns the type of the specified event log record, such as error, warning, information, and more. Win
length of <event log record>	*Plain*	<integer>	Returns the length of the specified record. This is not the same as the length of the description. Win
record number of <event log record>	*Plain*	<integer>	Returns the integer value of the record number corresponding to the specified event log record. Win

Key Phrase	*Form*	Return Type	Description
source of <event log record>	*Plain*	<string>	Returns the source name (from the application, service, or component that logged the event) of the specified event log record. Win
time generated of <event log record>	*Plain*	<time>	Returns the time that the specified event log record was generated. Win
time written of <event log record>	*Plain*	<time>	Returns the time that the specified event record was written to the log. Win
user sid of <event log record>	*Plain*	<security identifier>	Returns the user security ID for the specified record in the event log. Win

Examples

- `exists record (oldest record number of it) of application event log`

  Returns TRUE if there is an oldest record in the application event log.

- `exists records of application event log`

  Returns TRUE if the application event log contains any records.

- `category of record (oldest record number of it) of application event log`

  Returns the category of the oldest record of the application event log.

- `computer of record (oldest record number of it) of application event log`

  Returns the name of the computer that logged the last entry in the application event log.

- `description of record (oldest record number of it) of application event log`

  Returns a description of the oldest record in the application event log. Typically includes a description of the programs and what happened for each entry in the event log.

- `event id of record (oldest record number of it) of application event log`

  Returns the id number of the oldest record in the application event log.

- `length of record (oldest record number of it) of application event log`

  Returns the length of the specified record in the application event log.

- `record number of record (oldest record number of it) of application event log`

  Returns the integer record number corresponding to the oldest record in the application event log.

- `source of record (oldest record number of it) of application event log`

  Returns name of the source of the oldest record in the application event log.

- `time generated of record (oldest record number of it) of application event log`

  Returns the time (in day, month, year, time, zone format) that the oldest record in the application event log was generated.

- `time written of record (oldest record number of it) of application event log`

  Returns the time (in day, month, year, time, zone format) that the oldest record in the application event log was written.

- `user sid of record (oldest record number of it) of application event log`

  Returns the user security ID for the oldest record in the application event log, for instance NT AUTHORITY\SYSTEM.

Event Log Event Type

These Inspectors return information about the types of Windows Event log entries, which record various operating system events including errors, warnings and general information.

Creation Methods

Key Phrase	*Form*	Description
audit failure event log event type	*PlainGlobal*	Returns an object corresponding to an audit failure -- an event related to the failed execution of an action. Win
audit success event log event type	*PlainGlobal*	Returns an object corresponding to an audit success in an event log. Win
error event log event type	*PlainGlobal*	Returns an object corresponding to an error event in the log, such as the failure of a service to start. Win
event log event type <integer>	*NumberedGlobal*	Returns an event type object corresponding to the specified number. The enumerated types include: • 1: error event • 2: warning event • 4: information event • 8: audit success event • 16: audit failure event. Win

Key Phrase	*Form*	Description
event type of <event log record>	*Plain*	Returns the event type of the specified record from the event log. Win
information event log event type	*PlainGlobal*	Returns an object corresponding to an information event, which is generally related to a successful action. Win
warning event log event type	*PlainGlobal*	Returns an object corresponding to a warning in the event log. Warnings can be used to prevent future system problems. Win

Operators

Key phrase	Return Type	Description
<event log event type> = <event log event type>	*<boolean>*	Compare two event log event types for equality. Win

Examples

- `audit failure event log event type= event type of record (oldest record number of it) of application event log`

  Returns TRUE if the oldest record of the application event log contains an audit failure.

- `error event log event type= event type of record (oldest record number of it) of application event log`

  Returns TRUE if the oldest record of the application event log contains an error.

WMI Objects

These are the Inspectors for Windows Management Instrumentation (WMI).

Wmi

A wmi object provides access to the WMI (Windows Management Instrumentation) query facility. This object provides access to a large amount of configuration and client-specific data.

CAUTION: Because these Inspectors are written on top of the IWbemLocator::ConnectServer APIs you may experience problems unique to this interface. On a small number of systems, these APIs may hang the client. BES version 7.2 corrects this behavior. If you have an earlier version of BES, you can set _BESClient_Inspector_DisableWMI to 1 to disable these Inspectors. A Fixlet or Task that uses a disabled inspector will report false; retrieved properties that request a disabled inspector value will report an error.

For the latest information on issues surrounding the WMI inspectors, search the Tivoli Endpoint Manager support knowledge base.

Creation Methods

Key Phrase	*Form*	Description
full wmi <string>	*NamedGlobal*	Returns a wmi object which can retrieve all values, including system values. Win
rsop computer wmi	*PlainGlobal*	Provides access to the Resultant Set of Policy (RSoP) WMI classes via the RsopLoggingModeProvider. This is used to examine the state of the GPO (Group Policy object) security policies. Win:7.0
rsop user wmi <security identifier>	*Index<security identifier>Global*	Each user has its own RSoP (Resultant Set of Policy) namespace based on the user's Security Identifier (SID). This Inspector returns the namespace specified by the <security identifier>. Win:7.0
wmi	*PlainGlobal*	Returns the wmi object corresponding to the "root\cimv2" namespace. Win
wmi <string>	*NamedGlobal*	Returns the wmi object corresponding to the namespace string provided. Win

Properties

Key Phrase	*Form*	Return Type	Description
select <string> of <wmi>	*Named*	<wmi select>	Returns the wmi select whose name matches the string provided. Win
select object <string> of <wmi>	*Named*	<wmi object>	Returns the desired property (specified by <string>) from the given wmi object. Win

Wmi Select

A value returned as a result of a WMI select query. You can find more information at the MSDN Library (http://msdn.microsoft.com/library/) under WMI Classes. WMI Inspectors can provide you with useful information about your Client computers. For instance, to get the asset tag from a dell, use:

- string value of select "SerialNumber from Win32_systemenclosure" of wmi.

CAUTION: Because these Inspectors are written on top of the IWbemLocator::ConnectServer APIs you may experience certain problems unique to this interface. On a small number of systems, these APIs may actually hang the client. BES version 7.2 corrects this behavior. If you have an earlier version of BES, you can set _BESClient_Inspector_DisableWMI to 1 to disable these Inspectors. A Fixlet or Task that uses a disabled inspector will report false; retrieved properties that request a disabled inspector value will report an error. For the latest information on issues surrounding the WMI inspectors, search the Tivoli Endpoint Manager support knowledge base.

Creation Methods

Key Phrase	*Form*	Description
property <string> of <wmi object>	*Named*	Returns the Inspector properties of the form <string> of <wmi object>. Win
property of <wmi object>	*Plain*	Returns the Inspector properties of the specified wmi object. Win
select <string> of <wmi>	*Named*	Returns the wmi select whose name matches the string provided. Win

Properties

Key Phrase	*Form*	Return Type	Description
<wmi select> as string	*Cast*	<string>	Returns a string formatted as <name>=<value> for the wmi select. Win
boolean value <integer> of <wmi select>	*Numbered*	<boolean>	Returns the WMI value cast to boolean. Win
boolean value of <wmi select>	*Plain*	<boolean>	Returns the boolean value of a <wmi select> object (exists only for boolean objects). Win
integer value <integer> of <wmi select>	*Numbered*	<integer>	Returns the WMI value cast to an integer. Win

Key Phrase	*Form*	Return Type	Description
integer value of <wmi select>	*Plain*	<integer>	Returns the integer value of a <wmi select> object whose value is of type integer. Win
name of <wmi select>	*Plain*	<string>	Returns the name of the wmi select. Win
string value <integer> of <wmi select>	*Numbered*	<string>	Returns the numbered string of a WMI array value. Win
string value of <wmi select>	*Plain*	<string>	Returns the string value of the wmi select. Win
time value <integer> of <wmi select>	*Numbered*	<time>	Returns the numbered time of a WMI array value. Win
time value of <wmi select>	*Plain*	<time>	Returns the time value of the wmi select whose value is of type time. Win
type of <wmi select>	*Plain*	<integer>	Returns the type of data stored in the wmi select value. Possible values are: • 8 (VT_BSTR) • 11 (VT_BOOL) • 22 (VT_INT) • (See MSDN documentation for the meaning of additional values). Win

NOTE:

Here are a few other examples of using the wmi Inspectors. Each of the examples below hands back dozens of wmi objects:

Q: selects "* from Win32_ComputerSystem" of wmi
Q: selects "* from win32_keyboard" of wmi
Q: selects "* from win32_CDROMDrive" of wmi
Q: selects "* from win32_DiskDrive" of wmi
Q: selects "* from win32_BIOS" of wmi
Q: selects "* from win32_CacheMemory" of wmi
Q: selects "* from win32_DMIChannel" of wmi
Q: selects "* from win32_FloppyController" of wmi
Q: selects "* from win32_IDEController" of wmi
Q: selects "* from win32_IRQResource" of wmi
Q: selects "* from win32_MemoryDevice" of wmi
Q: selects "* from win32_MotherboardDevice" of wmi
Q: selects "* from win32_ParallelPort" of wmi
Q: selects "* from Win32_PNPDevice" of wmi
Q: selects "* from win32_Processor" of wmi
Q: selects "* from win32_SerialPort" of wmi

Q: selects "* from win32_SoundDevice" of wmi
Q: selects "* from win32_NetworkAdapter" of wmi
Q: selects "* from win32_NetworkAdapterSetting" of wmi
Q: selects "* from win32_Battery" of wmi
Q: selects "* from win32_PrinterPrinterDriver" of wmi
Q: selects "* from win32_PrinterSetting" of wmi

Wmi Object

These Inspectors allow you to analyze the properties of WMI objects.

Creation Methods

Key Phrase	*Form*	Description
select object <string> of <wmi>	*Named*	Returns the desired property (specified by <string>) from the given wmi object. Win

Properties

Key Phrase	*Form*	Return Type	Description
<wmi object> as string	*Cast*	<string>	Casts the given wmi object as a string type. Win
property <string> of <wmi object>	*Named*	<wmi select>	Returns the Inspector properties of the form <string> of <wmi object>. Win
property of <wmi object>	*Plain*	<wmi select>	Returns the Inspector properties of the specified wmi object. Win

Examples

- `Property "SerialNumber" of select object "* from win32_operatingsystem" of wmi`

  Returns serial number of the selected wmi object, in a form like SerialNumber=76487-OEM-0012903-00925.

- `properties "select" of type "wmi"`

  Returns Inspector properties of the form "select of <wmi>".

SMBIOS objects

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

Smbios

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

Creation Methods

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

Properties

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

Examples

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

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

Smbios Structure

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

Creation Methods

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

Properties

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

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

Examples

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

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

```
number of structures of smbios > 10
```

Returns TRUE if the number of smbios structures exceeds ten.

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

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

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

Smbios Value

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

Creation Methods

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

Properties

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

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

Examples

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

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

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

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

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

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

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

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

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

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

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

Site Objects

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

Site

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

Creation Methods

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

Properties

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

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

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

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

Examples

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

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

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

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

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

Site Group

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

Creation Methods

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

Properties

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

Site Version List

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

Creation Methods

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

Fixlet Objects

These Inspectors return information about individual Fixlets.

Fixlet

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

Creation Methods

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

Properties

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

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

Fixlet_header

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

Creation Methods

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

Properties

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

Examples

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

Client Objects

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

Client

The client object allows access to properties of the client application hosting the relevance evaluation, typically a BigFix/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager Client. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
registration address of <client>	*Plain*	<ipv4or6 address>	This Inspector returns the IP address (as an <ipv4or6 address> type) that the specified BigFix/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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

Examples

- `registration mac address of client`
- Returns a MAC address such as 00-1e-c9-4d-ce-5c.

Setting

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

Creation Methods

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

Properties

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

Creation Methods

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

Properties

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

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

Current Relay

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

Creation Methods

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

Properties

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

Root Server

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

Creation Methods

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

Properties

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

Evaluation Cycle

An Evaluation cycle represents a complete run through all the content available on the BigFix/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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

Examples

- `last start time of application usage summary "winword.exe"`
- Returns the date and time Word was last started.

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

Key Phrase	*Form*	Return Type	Description
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
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/Tivoli Endpoint Manager 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

Key Phrase	*Form*	Description
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

Key Phrase	*Form*	Return Type	Description
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
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/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager. 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/Tivoli Endpoint Manager Clients allowed by this license.

BES Product

A BigFix/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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
x64 variable <string> of <environment>	*Named*	<environment variable>	Returns the same as the the named variable Inspector, but from outside the WOW64 environment. Win:8.1
x64 variable of <environment>	*Plain*	<environment variable>	Returns the same as the the variable Inspector, but from outside the WOW64 environment. Win:8.1

Examples

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

- `value of variable "path" of environment contains "\extras\"`
  - TRUE if there is an environment variable named "path" and its value contains "\extras\".

- `number of variables of environment`

  Returns the total number of variables in the environment space.

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
x64 variable <string> of <environment>	*Named*	Returns the same as the the named variable Inspector, but from outside the WOW64 environment. Win:8.1
x64 variable of <environment>	*Plain*	Returns the same as the the variable Inspector, but from outside the WOW64 environment. Win:8.1

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.

Access Control List

An Access Control List, or ACL, is a list of security protections that applies to an object. An object can be a file, process, event, or anything else having a security descriptor. An entry in an access control list (ACL) is an access control entry (ACE). These Inspectors work by exposing the GetEffectiveRightsFromAcl method, as explained at the MSDN site.

- NOTE: Requires Windows XP, Windows 2000 Professional, or Windows NT Workstation 3.1 and later.

CAUTION: These Windows Inspectors utilize the GetEffectiveRightsFromAcl API, which may introduce extra network load, particularly when looking up permissions of users and groups that do not exist or have complex relationships.

Properties

Key Phrase	*Form*	Return Type	Description
effective access mode for <security account> of <access control list>	*Index<security account>*	<integer>	Returns an integer corresponding to the access mode for the trustee specified by the security account of the given access control list. Win:8.0
effective access mode for <string> of <access control list>	*Named*	<integer>	Returns an integer corresponding to the access mode for the trustee specified by <string> of the given access control list. Win
effective access system security permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has access system security permissions on the given access control list. Win:8.0
effective access system security permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has access system security permissions on the given access control list. Win

Key Phrase	*Form*	Return Type	Description
effective append permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has append permissions on the given access control list. Win:8.0
effective append permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has append permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective change notification permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has change notification permissions on the given access control list. Win:8.0
effective change notification permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has change notification permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective create file permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has file creation permissions on the given access control list. Win:8.0
effective create file permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has file creation permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective create folder permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has folder creation permissions on the given access control list. Win:8.0
effective create folder permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has folder creation permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win

Key Phrase	*Form*	Return Type	Description
effective create link permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has link creation permissions on the given access control list. Win:8.0
effective create link permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has link creation permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective create subkey permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has subkey creation permissions on the given access control list. Win:8.0
effective create subkey permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has subkey creation permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective delete child permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has child deletion permissions on the given access control list. Win:8.0
effective delete child permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has child deletion permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective delete permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has delete permissions on the given access control list. Win:8.0
effective delete permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has delete permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win

Key Phrase	*Form*	Return Type	Description
effective enumerate subkeys permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the specified security account provides the right to list the subkeys of a registry key. Win:8.0
effective enumerate subkeys permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has subkey enumeration permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective execute permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has execution permissions on the given access control list. Win:8.0
effective execute permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has execution permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective generic all permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has all generic permissions on the given access control list. Win:8.0
effective generic all permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has all generic permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective generic execute permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has generic execution permissions on the given access control list. Win:8.0
effective generic execute permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by the security account has generic execution permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win

Key Phrase	*Form*	Return Type	Description
effective generic read permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has generic read permissions on the given access control list. Win:8.0
effective generic read permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by the security account has generic read permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective generic write permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has generic write permissions on the given access control list. Win:8.0
effective generic write permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by the security account has generic write permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective list permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has list permissions on the given access control list. Win:8.0
effective list permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has list permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective maximum allowed permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has maximum allowed permissions on the given access control list. Win:8.0
effective maximum allowed permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by the security account has maximum allowed permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win

Key Phrase	*Form*	Return Type	Description
effective query value permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has query value permissions on the given access control list. Win:8.0
effective query value permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has query value permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective read attributes permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has read attribute permissions on the given access control list. Win:8.0
effective read attributes permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has read attribute permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective read control permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has read control permissions on the given access control list. Win:8.0
effective read control permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has read control permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective read extended attributes permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has extended read attribute permissions on the given access control list. Win:8.0
effective read extended attributes permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has extended read attribute permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win

Key Phrase	*Form*	Return Type	Description
effective read permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has read permissions on the given access control list. Win:8.0
effective read permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has read permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective set value permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has value setting permissions on the given access control list. Win:8.0
effective set value permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has value setting permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective synchronize permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has synchronization permissions on the given access control list. Win:8.0
effective synchronize permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has synchronization permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective traverse permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has traverse permissions on the given access control list. Win:8.0
effective traverse permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has traverse permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win

Key Phrase	*Form*	Return Type	Description
effective write attributes permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has attribute writing permissions on the given access control list. Win:8.0
effective write attributes permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has attribute writing permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective write dac permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has dac writing permissions on the given access control list. Win:8.0
effective write dac permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has dac writing permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective write extended attributes permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has extended attribute writing permissions on the given access control list. Win:8.0
effective write extended attributes permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has extended attribute writing permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
effective write owner permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has write owner permissions on the given access control list. Win:8.0
effective write owner permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has write owner permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win

Key Phrase	*Form*	Return Type	Description
effective write permission for <security account> of <access control list>	*Index<security account>*	<boolean>	Returns TRUE if the trustee specified by the security account has write permissions on the given access control list. Win:8.0
effective write permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has write permissions on the given access control list. CAUTION: This Inspector may result in increased network load. Win
entry of <access control list>	*Plain*	<access control entry>	Iterates the ACEs of an ACL. Win

NOTE:

The ACCESS_MASK is returned from the effective access mode as a double word defining standard, specific, and generic rights. These rights are used in access control entries (ACEs) and are the primary means of determining access to an object.

Bits	Meaning
0 through 15	Specific rights. Contains the access mask specific to the object type associated with the mask.
16 through 23	Contains the object's standard access rights.
24	The Access system security bit is used to indicate access to a system access control list (SACL). If this flag is set in the access mask of an audit access ACE (successful or unsuccessful access), the SACL access will be audited.
25	Maximum allowed.
26 through 27	Reserved.
28	Generic all.
29	Generic execute.
30	Generic write.
31	Generic read.

The standard rights bits from 16 to 23 contain the object's standard access rights and can be a combination of the following predefined flags:

Bit	Flag	Meaning
16	DELETE	Delete access.
17	READ_CONTROL	Read access to the owner, group, and discretionary access control list (DACL) of the security descriptor.
18	WRITE_DAC	Write access to the DACL.
19	WRITE_OWNER	Write access to owner.
20	SYNCHRONIZE	Synchronize access.

Examples

- `effective access mode for "Administrators" of dacls of security descriptors of system folder as hexadecimal`

  Returns a hex value corresponding to the access mode of the system folder for users logged in as Administrators.

- `effective append permission for "Power Users" of dacls of security descriptors of windows folder`

  Returns TRUE if Power Users have append permissions on the system folder.

- `effective create folder permissions for "Administrators" of dacls of security descriptors of folders of folder "c:\"`

  Returns a list of TRUE/FALSE values corresponding to the ability of the Administrator to create new folders in each of the existing folders of the c: drive.

- `effective synchronize permission for "Administrators" of dacls of security descriptors of system folder`

  Returns TRUE if the Administrator has permission to syncrhonize with the system folder.

Access Control Entry

An Access Control Entity, or ACE, is an entry in an access control list (ACL). An ACE contains a set of access rights and a security identifier (SID) that identifies a trustee for whom the rights are allowed, denied, or audited.

Creation Methods

Key Phrase	*Form*	Description
entry of <access control list>	*Plain*	Iterates the ACEs of an ACL. Win

Properties

Key Phrase	*Form*	Return Type	Description
access mode of <access control entry>	*Plain*	<integer>	For a discretionary ACL (DACL), this flag indicates whether the ACL allows (1) or denies (3) the specified access rights. Win
append permission of <access control entry>	*Plain*	<boolean>	For a file ACE, returns TRUE if the ACE grants or denies append permissions. Win
change notification permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies change notification permissions. Win
create file permission of <access control entry>	*Plain*	<boolean>	For a folder ACE, returns TRUE if the ACE grants or denies create file permissions. Win
create folder permission of <access control entry>	*Plain*	<boolean>	For a folder ACE, returns TRUE if the ACE grants or denies create folder permissions. Win
create link permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies create key link permissions. Win
create subkey permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies creation of subkey permissions. Win
delete child permission of <access control entry>	*Plain*	<boolean>	For a folder ACE, returns TRUE if the ACE grants or denies child deletion permissions. Win

Key Phrase	*Form*	Return Type	Description
delete permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or generic delete permissions. Win
enumerate subkeys permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or enumerate subkey permissions. Win
execute permission of <access control entry>	*Plain*	<boolean>	For a file ACE, returns TRUE if the ACE grants or denies execute permissions. Win
generic all permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies all generic permissions. Win
generic execute permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies generic execute permissions. Win
generic read permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies generic read permissions. Win
generic write permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies generic write permissions. Win
inheritance of <access control entry>	*Plain*	<integer>	A set of bit flags that determines whether other containers or objects can inherit the ACE from the primary object to which the ACL is attached. The actual values of the constants are: • NO_INHERITANCE = 0 • SUB_OBJECTS_ONLY_INHERIT = 1 • SUB_CONTAINERS_ONLY_INHERIT = 2 • SUB_CONTAINERS_AND_OBJECTS_INHERIT = 3 • OJECT_INHERIT_ACE = 1 • CONTAINER_INHERIT_ACE = 2 • NO_PROPAGATE_INHERIT_ACE = 4 • INHERIT_ONLY_ACE = 8. Win
list permission of <access control entry>	*Plain*	<boolean>	For a folder ACE, returns TRUE if the ACE grants or denies list permissions. Win

Key Phrase	*Form*	Return Type	Description
maximum allowed permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies maximum allowed permissions. Win
query value permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies query value permissions. Win
read attributes permission of <access control entry>	*Plain*	<boolean>	For a file or folder ACE, returns TRUE if the ACE grants or denies read attributes permissions. Win
read control permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies reading access control permissions. Win
read extended attributes permission of <access control entry>	*Plain*	<boolean>	For a file or folder ACE, returns TRUE if the ACE grants or denies read extended attributes permissions. Win
read permission of <access control entry>	*Plain*	<boolean>	For a file ACE, returns TRUE if the ACE grants or denies read permissions. Win
set value permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies set value permissions. Win
synchronize permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies synchronize permissions. Win
traverse permission of <access control entry>	*Plain*	<boolean>	For the specified folder ACE, returns TRUE if it grants or denies traverse • folder permission. Win
trustee of <access control entry>	*Plain*	<security identifier>	Returns the trustee to whom the specified ACE applies. Win
trustee type of <access control entry>	*Plain*	<integer>	Returns the type of trustee to whom the specified ACE applies. Win
write attributes permission of <access control entry>	*Plain*	<boolean>	For a file or folder ACE, returns TRUE if the ACE grants or denies write attribute permissions. Win

Key Phrase	*Form*	Return Type	Description
write dac permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies write DAC permissions. Win
write extended attributes permission of <access control entry>	*Plain*	<boolean>	For a file or folder ACE, returns TRUE if the ACE grants or denies write extended attribute permissions. Win
write owner permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies write owner permissions. Win
write permission of <access control entry>	*Plain*	<boolean>	For a file ACE, returns TRUE if the ACE grants or denies write permissions. Win

System Access Control List

These Inspectors retrieve information from the access control list that controls the generation of audit messages for attempts to access a securable object. The ability to get or set an object's SACL is controlled by a privilege typically held only by system administrators.

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

Creation Methods

Key Phrase	*Form*	Description
sacl of <security descriptor>	*Plain*	Returns the system access control list (SACL), an ACL that controls the generation of audit messages for attempts to access a securable object. Win:7.1

Properties

Key Phrase	*Form*	Return Type	Description
<system access control list> as string	*Cast*	<string>	Converts the specified system access control list (SACL) into a string value in the Microsoft Security Descriptor String Format. Win:7.1

Discretionary Access Control List

These Inspectors retrieve information from the access control list that is monitored by the owner of the object and specifies what kinds of access particular users or groups can have to the specified object.

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

Creation Methods

Key Phrase	*Form*	Description
dacl of <security descriptor>	*Plain*	Returns the discretionary access control list (DACL) that identifies the users and groups who are allowed or denied access to the specified security descriptor. Win:7.1

Properties

Key Phrase	*Form*	Return Type	Description
<discretionary access control list> as string	*Cast*	<string>	Converts the discretionary system access control list (DACL) into a string value in the Microsoft Security Descriptor String Format. Win:7.1

Security Account

The security account type serves as a base type for the "user" and "local group" types and for properties common to users and groups.

Creation Methods

Key Phrase	*Form*	Description
account with privilege <string>	*NamedGlobal*	Returns a security account constant corresponding to an account with the privilege specified in the string. Win:8.0
account with privileges	*PlainGlobal*	Returns a security account constant corresponding to an 'account with privileges'. Win:8.0
anonymous logon group	*PlainGlobal*	This refers to users who have logged in anonymously. Win:8.0
authenticated users group	*PlainGlobal*	This refers to a group including users whose identities were authenticated when they logged on. Membership is controlled by the operating system. Win:8.0
batch group	*PlainGlobal*	This refers to a group including all users who have logged on through a batch queue facility such as the task scheduler. Membership is controlled by the operating system. Win:8.0
builtin administrators group	*PlainGlobal*	This refers to a built-in group . After the initial installation of the OS, the first member of the group is the Administrator account. When a computer then joins a domain, the Domain Admins group is added to the Administrators group. When a server becomes a domain controller, the Enterprise Admins group is also added to the Administrators group. The Administrators group has built-in capabilities that give its members full control over the system. The group is the default owner of any object that is created by any other member of the group. Win:8.0
builtin backup operators group	*PlainGlobal*	This refers to a built-in group which, by default, has no members. Backup Operators can back up and restore all files on a computer, regardless of file permissions. Backup Operators can log on to a computer and shut it down. Win:8.0

Key Phrase	*Form*	Description
builtin guests group	*PlainGlobal*	This refers to a built-in group which, by default, only contains the Guest account. This group allows otherwise unauthorized users to log on with limited privileges to a computer's built-in Guest account. Win:8.0
builtin network configuration operators group	*PlainGlobal*	This refers to XP machines, where some admin privileges include managing the configuration of networking features. Win:8.0
builtin power users group	*PlainGlobal*	This refers to a built-in group which, by default, has no members. This group does not exist for domain controllers. Power Users can create other local users and groups as well as modify and delete accounts. They can also remove users from the other groups. Power Users also can install, manage and delete applications, local printers and file shares. Win:8.0
builtin remote desktop users group	*PlainGlobal*	This refers to the XP only. Members of this group are granted the right to log in remotely. Win:8.0
builtin replicator group	*PlainGlobal*	This refers to Windows NT domains. This group is called Replicators and is used by the directory replication service. In 2K/XP the group is present but is not used. Win:8.0
builtin users group	*PlainGlobal*	This refers to a built-in group. After the initial installation of the OS, the first member is the Authenticated Users group. When a computer subsequently joins a domain, the Domain Users group is added to the Users group. These users can perform tasks such as running applications, using printers, shutting down or locking the computer. Users can install applications for their use only, provided the installation program supports per-user installation. Win:8.0
creator group group	*PlainGlobal*	This refers to a placeholder in an inheritable access control entry (ACE). When the ACE is inherited, the system replaces the SID with the one from the primary group of the object's current owner. The primary group is used only by the POSIX subsystem. Win:8.0

Key Phrase	*Form*	Description
creator owner group	*PlainGlobal*	This refers to a placeholder in an inheritable access control entry (ACE). When the ACE is inherited, the system replaces this SID with the one from the object's current owner. Win:8.0
dialup group	*PlainGlobal*	This refers to a group implicitly including all users who logged in to the system through a dial-up connection. Membership is controlled by the operating system. Win:8.0
everyone group	*PlainGlobal*	This refers to a group including all users, even anonymous users and guests. Membership is controlled by the operating system. Win:8.0
interactive group	*PlainGlobal*	This refers to a group including all users who have logged on interactively. Membership is controlled by the operating system. Win:8.0
local service group	*PlainGlobal*	Returns a security account constant corresponding to a 'local service group'. Win:8.0
network group	*PlainGlobal*	This refers to a group implicitly including all users who are logged on through a network connection. Membership is controlled by the operating system. Win:8.0
network service group	*PlainGlobal*	Returns a security account constant corresponding to a 'network service group'. Win:8.0
remote interactive logon group	*PlainGlobal*	Refers to the group of users who log on using an RDP connection. Win:8.0
security account <string>	*NamedGlobal*	This is a named Inspector that uses the LookupAccountName API function to return an object representing a user or group. Win:7.2
service group	*PlainGlobal*	This refers to a group including all security principals that have logged on as a service. Membership is controlled by the operating system. Win:8.0
system group	*PlainGlobal*	Returns a security account constant corresponding to a 'system group'. Win:8.0

Key Phrase	*Form*	Description
terminal server user group	*PlainGlobal*	Refers to a group including all users who have logged on to a Terminal Services server. Membership is controlled by the operating system. Win:8.0
well known account <integer>	*NumberedGlobal*	Returns a security account constant corresponding to a numbered 'well known account'. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
privilege of <security account>	*Plain*	<string>	Returns a string describing the privileges assigned to the specified security account. For more information, see the MSDN article on LsaEnumerateAccountRights. For a description of the possible constants that can be returned, see the articles on Account Rights Constants and Privilege Constants. Win:7.1
sid of <security account>	*Plain*	<security identifier>	Returns the Security ID (SID) associated with the specified security account. Win:7.1

Examples

- `privileges of security account "Network Service"`

  Returns a list of privileges for the specified security account, such as SeAuditPrivilege, SeChangeNotifyPrivilege, and others.

Security Descriptor

A structure and associated data that contains the security information for a securable object. A security descriptor identifies the object's owner and primary group. It can also contain a DACL that controls access to the object, and a SACL that controls the logging of attempts to access the object.

Creation Methods

Key Phrase	*Form*	Description
security descriptor of <file>	*Plain*	Specifies the security descriptor associated with the specified file. Win

Key Phrase	*Form*	Description
security descriptor of <folder>	*Plain*	Specifies the security descriptor associated with the specified folder. Win
security descriptor of <network share>	*Plain*	Specifies the security descriptor associated with the specified network share. Win
security descriptor of <registry key>	*Plain*	Specifies the security descriptor associated with the specified registry key. Win
security descriptor of <scheduled task>	*Plain*	Returns the security descriptor for the specified scheduled task. • 2.0 interface only. Win:8.0
security descriptor of <service>	*Plain*	This Windows-specific Inspector returns a security descriptor for the specified service. Win:7.0
security descriptor of <task folder>	*Plain*	Returns the security descriptor for the specified task folder. Win:8.0
security descriptor of <task registration info>	*Plain*	Returns the security descriptor of the scheduled task referred to by the specified task registration information object. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
<security descriptor> as string	*Cast*	<string>	Returns the security descriptor in string format. Win
control of <security descriptor>	*Plain*	<integer>	Returns the integer property obtained by using the Microsoft Windows GetSecurityDescriptorControl API. This integer contains bits that indicate DACL behaviors as well as default behaviors. See the MSDN documentation of SECURITY_DESCRIPTOR_CONTROL for more information. Win

Key Phrase	*Form*	Return Type	Description
dacl of <security descriptor>	*Plain*	<discretionary access control list>	Returns the discretionary access control list (DACL) that identifies the users and groups who are allowed or denied access to the specified security descriptor. Win:7.1
group of <security descriptor>	*Plain*	<security identifier>	Returns the security identifier of the group of the specified security descriptor. Win
owner of <security descriptor>	*Plain*	<security identifier>	Returns the security identifier of the owner of the specified security descriptor. Win
sacl of <security descriptor>	*Plain*	<system access control list>	Returns the system access control list (SACL), an ACL that controls the generation of audit messages for attempts to access a securable object. Win:7.1

Examples

- `(DISPLAY name of it, security descriptor of it ) of service "TapiSrv"`

  Returns a value such as 'Telephony, D:(A;OICI;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;OICI;CCLCSWRPLO;;;BU)'.

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
group of <security descriptor>	*Plain*	Returns the SID of the group of the specified security descriptor. Win
owner of <security descriptor>	*Plain*	Returns the security identifier of the owner of the specified security descriptor. Win
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

Key Phrase	*Form*	Description
sid of <logged on user>	*Plain*	Returns the Security ID (SID) of the user associated with the session's primary access token. With Windows 2003/XP/Vista, this is determined by WTSQueryUserToken. With NT4/2000 it is determined by the apparent shell process running in the given session. This Inspector may fail if run in a non-privileged context. The SID does not exist under Windows 9x. Win:7.0
sid of <security account>	*Plain*	Returns the Security ID (SID) associated with the specified security account. Win:7.1
trustee of <access control entry>	*Plain*	Returns the trustee to whom the specified ACE applies. Win
user of <process>	*Plain*	Returns a security identifier associated with the user of specified process. This Inspector requires Terminal Services or Win2000+, and may not be available if the inspector application does not have sufficient permissions. This object corresponds to the 'User Name' column in Task Manager. Win:8.0
user sid of <event log record>	*Plain*	Returns the user security ID for the specified record in the event log. Win

Properties

Key Phrase	*Form*	Return Type	Description
<security identifier> as string	*Cast*	<string>	Returns the security identifier in string format. Win, Mac:8.1
account name of <security identifier>	*Plain*	<string>	Retrieves the name of the account for this SID and the name of the first domain on which this SID is found. Win
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

Key Phrase	*Form*	Return Type	Description
domain name of <security identifier>	*Plain*	<string>	Returns the domain name of the first domain on which the specified SID is found. Win

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

Examples

- `component string of owner of security descriptor of windows folder`
- Returns a string of the form: S-1-5-32-544.

Security Database

These Inspectors retrieve information from the security accounts manager (SAM) database or, in the case of domain controllers, the Active Directory. The Security database and its properties expose the NetUserModalsGet API, levels 0 and 3. For more information, see the NetUserModalsGet Function at the MSDN site: http://msdn.microsoft.com.

Creation Methods

Key Phrase	*Form*	Description
security database	*PlainGlobal*	Returns the security accounts manager (SAM) database or, in the case of domain controllers, the Active Directory. Win:7.1

Properties

Key Phrase	*Form*	Return Type	Description
account lockout duration of <security database>	*Plain*	<time interval>	Returns the time interval corresponding to how long a locked account remains locked before it is automatically unlocked. For more information, see the MSDN article on NetUserModalsGet. Win:7.1

Key Phrase	*Form*	Return Type	Description
account lockout observation window of <security database>	*Plain*	<time interval>	Returns a time interval corresponding to the maximum time that can elapse between any two failed logon attempts before lockout occurs. For more information, see the MSDN article on NetUserModalsGet. Win:7.1
account lockout threshold of <security database>	*Plain*	<integer>	Returns an integer corresponding to the number of invalid password authentications that can occur before an account is marked 'locked out.' For more information, see the MSDN article on NetUserModalsGet. Win:7.1
force logoff interval of <security database>	*Plain*	<time interval>	Returns the time interval between the end of the valid logon time and the time when the user must log off the network. A value of zero indicates that the user must log off immediately as soon as the valid logon time expires. For more information, see the MSDN article on NetUserModalsGet. Win:7.1
maximum password age of <security database>	*Plain*	<time interval>	Returns a time interval corresponding to the maximum password age found in the specified security database. Win:7.1
minimum password age of <security database>	*Plain*	<time interval>	Returns a time interval corresponding to the minimum password age found in the specified security database. Win:7.1
minimum password length of <security database>	*Plain*	<integer>	Returns an integer corresponding to the minimum password length found in the specified security database. Win:7.1
password history length of <security database>	*Plain*	<integer>	Returns the integer length of the password history maintained by the security database. A new password cannot match any of the previous passwords in the specified history. For more information, see the MSDN article on NetUserModalsGet. Win:7.1

Audit Policy

The audit policy inspectors return the policies put in place for recording information about security-related operations on the client computer. For example, you can set a policy to monitor the modification of files. This will trigger an audit entry showing whenever a file is modified, the associated user account, and the date and time of the action. You can audit both successful and failed attempts at actions. Often, the failed attempts are more interesting, as they may indicate attempts to unsuccessfully subvert a policy. For instance, a successful login is not as interesting as a repeated failure might be.

Creation Methods

Key Phrase	*Form*	Description
audit policy	*PlainGlobal*	Windows Vista (and later versions of Windows) allows a finer granularity with audit policies by using subcategories. Setting audit policy at the category level overrides the new subcategory feature. A new registry key introduced in Vista is used to manage subcategories without requiring a change to Group Policy. This registry can be set to prevent the application of category-level audit policy from both Group Policy and the Local Security Policy admin tool. Win:7.2

Properties

Key Phrase	*Form*	Return Type	Description
account logon category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the Account Logon category of the audit policy. Win:7.2
account management category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the Account Management category of the audit policy. Win:7.2
category of <audit policy>	*Plain*	<audit policy category>	Returns the categories of the specified audit policy. Win:7.2
detailed tracking category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the Detailed Tracking category of the specified audit policy. Win:7.2
ds access category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the DS Access category of the audit policy. Win:7.2

Key Phrase	*Form*	Return Type	Description
logon logoff category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the Logon/Logoff category of the audit policy. Win:7.2
object access category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the Object Access category of the audit policy. Win:7.2
policy change category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the Policy Change category of the audit policy. Win:7.2
privilege use category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the Privilege Use category of the audit policy. Win:7.2
system category of <audit policy>	*Plain*	<audit policy category>	Returns an object corresponding to the System category of the audit policy. Win:7.2

Examples

- `(name of it, (audit success of it, audit failure of it) of system policy of it) of subcategories of categories of audit policy`

  This example lists the names along with the success and failure status of all the subcategories of the audit policy.

- `(name of it, audit success of system policies of it) of subcategories of account logon category of audit policy`

  Returns the names and the system policy audit success status of the account logon subcategories.

- `names of subcategories of account management category of audit policy`

  Returns a list of the subcategory names of the of the account management categories.

- `names of categories of audit policy`

  Returns the names of the audit policy categories, including System, Logon/Logoff, Object Access, Privilege Use, Detailed Tracking, Policy Change, Account Management, DS Access and Account Logon.

- `names of subcategories of ds access category of audit policy`

  Returns the names of the specified subcategories. Produces the same result as 'names of subcategories of category whose (name of it is "DS Access") of audit policy'.

Audit Policy Category

Windows audit policies, as of Vista and later, are divided into categories. Currently there are 9 categories, including System, Logon/Logoff, Object Access, Privilege Use, Detailed Tracking, Policy Change, Account Management, DS Access and Account Logon.

Creation Methods

Key Phrase	*Form*	Description
account logon category of <audit policy>	*Plain*	Returns an object corresponding to the Account Logon category of the audit policy. Win:7.2
account management category of <audit policy>	*Plain*	Returns an object corresponding to the Account Management category of the audit policy. Win:7.2
category of <audit policy>	*Plain*	Returns the categories of the specified audit policy. Win:7.2
detailed tracking category of <audit policy>	*Plain*	Returns an object corresponding to the Detailed Tracking category of the specified audit policy. Win:7.2
ds access category of <audit policy>	*Plain*	Returns an object corresponding to the DS Access category of the audit policy. Win:7.2
logon logoff category of <audit policy>	*Plain*	Returns an object corresponding to the Logon/Logoff category of the audit policy. Win:7.2
object access category of <audit policy>	*Plain*	Returns an object corresponding to the Object Access category of the audit policy. Win:7.2
policy change category of <audit policy>	*Plain*	Returns an object corresponding to the Policy Change category of the audit policy. Win:7.2
privilege use category of <audit policy>	*Plain*	Returns an object corresponding to the Privilege Use category of the audit policy. Win:7.2
system category of <audit policy>	*Plain*	Returns an object corresponding to the System category of the audit policy. Win:7.2

Properties

Key Phrase	*Form*	Return Type	Description
name of <audit policy category>	*Plain*	<string>	Returns the name of the specified audit policy category. Win:7.2
subcategory of <audit policy category>	*Plain*	<audit policy subcategory>	Returns the subcategory for the specified audit policy category. Win:7.2

Examples

- `(name of it, audit success of system policies of it) of subcategories of account logon category of audit policy`

  Returns the names and the system policy audit success status of the account logon subcategories.

- `names of subcategories of account management category of audit policy`

  Returns a list of the subcategory names of the of the account management categories.

- `names of categories of audit policy`

  Returns the names of the audit policy categories, including System, Logon/Logoff, Object Access, Privilege Use, Detailed Tracking, Policy Change, Account Management, DS Access and Account Logon.

- `names of subcategories of ds access category of audit policy`

  Returns the names of the specified subcategories. Produces the same result as 'names of subcategories of category whose (name of it is "DS Access") of audit policy'.

Audit Policy Subcategory

Windows audit policy categories, as of Vista and later, are divided into about 50 subcategories. This level of granularity is designed to narrow in on specific security-related operations on the client computer, helping to filter out the normal noise of an active environment.

Creation Methods

Key Phrase	*Form*	Description
subcategory of <audit policy category>	*Plain*	Returns a list of the subcategories for the specified audit policy category. Win:7.2

Properties

Key Phrase	*Form*	Return Type	Description
effective policy <security account> of <audit policy subcategory>	*Index<security account>*	<audit policy information>	Returns the effective audit policy information for the specified subcategory for the given security account. The effective audit policy is determined by combining the system audit policy with per-user policy. Win:7.2
name of <audit policy subcategory>	*Plain*	<string>	Returns the name of the specified audit policy subcategory. Win:7.2
per user policy <security account> of <audit policy subcategory>	*Index<security account>*	<audit policy information>	Returns the per-user audit policy information for the given audit-policy subcategory and the specified security account. Win:7.2
system policy of <audit policy subcategory>	*Plain*	<audit policy information>	Returns the audit policy information (audit success or audit failure) corresponding to the specified audit policy subcategory. Win:7.2

Examples

- `(name of it, audit failure of effective policy (security account "Network Service") of it) of subcategories of categories of audit policy`

  Returns a list of the names and audit failure states of the all the audit policy subcategories for the specified security account.

- `names of subcategories of category whose (name of it is "System") of audit policy`

  Returns the names of the subcategories of the 'System' audit policy category. These include Security State Change, Security System Extension, System Integrity, Ipsec Driver and Other System Events.

- `audit success of system policy of subcategory whose (name of it is "Security Group Management") of account management category of audit policy`

  Returns the boolean audit success status of the specified system policy.

Audit Policy Information

These Inspectors return the two attributes of the audit policy for a given subcategory: whether or not succesful operations will be audited ("audit success"), and whether or not unsuccessful operations will be audited ("audit failure").

Creation Methods

Key Phrase	*Form*	Description
effective policy <security account> of <audit policy subcategory>	*Index<security account>*	Returns the effective audit policy for the specified subcategory for the given security account. The effective audit policy is determined by combining the system audit policy with per-user policy. Win:7.2
per user policy <security account> of <audit policy subcategory>	*Index<security account>*	Returns the per-user audit policy information for the given audit-policy subcategory and the specified security account. Win:7.2
system policy of <audit policy subcategory>	*Plain*	Returns the audit policy information (audit success or audit failure) corresponding to the specified audit policy subcategory. Win:7.2

Properties

Key Phrase	*Form*	Return Type	Description
audit failure of <audit policy information>	*Plain*	<boolean>	Returns the boolean audit failure status of the specified audit policy information. Win:7.2
audit success of <audit policy information>	*Plain*	<boolean>	Returns the boolean audit success status of the specified audit policy information. Win:7.2

Examples

- `(name of it, audit failure of effective policy (security account "Network Service") of it) of subcategories of categories of audit policy`

  Returns a list of the names and audit failure states of the all the audit policy subcategories for the specified security account.

- `audit success of system policy of subcategory whose (name of it is "Security Group Management") of account management category of audit policy`

  Returns the boolean audit success status of the specified system policy.

- `(name of it, audit failure of system policy of it) of subcategories of category whose (name of it is "System") of audit policy`

  Returns the names and audit failure status of each of the subcategories of the System audit policy category.

- `audit success of system policies of subcategory whose (name of it is "User Account Management") of account management category of audit policy`

  Returns the TRUE/FALSE status of the specified audit policy subcategory.

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
pem encoded certificate of <file>	*Plain*	Reads and returns the certificate from a file in the PEM format. This can be used to analyze encryption credentials on decrypting relays or root servers. Win:7.1, WM

Local Group

These Inspectors return information on local groups as defined on the local BES Client computer using the windows NetLocalGroupEnum API, one of Windows Network Management Functions. Local groups have names, comments, members and security IDs.

Creation Methods

Key Phrase	*Form*	Description
local group	*PlainGlobal*	Returns local groups defined on the local computer using the windows NetLocalGroupEnum API. Several local groups are defined simply by a default operating system install, and have names such as Administrators, Backup Operators, Guests, Network Configuration Operators, Power users, Users, and more. Some software applications also define local groups to help manage protections. Win
local group <string>	*NamedGlobal*	Returns a local group corresponding to the given name, such as Adminstrator, Guests, and others. Win

Properties

Key Phrase	*Form*	Return Type	Description
comment of <local group>	*Plain*	<string>	Returns a string containing a comment associated with the specfied local group (Administrator, Guest, Users). Win
member of <local group>	*Plain*	<local group member>	Returns a list of the members of the specified local group. Win
name of <local group>	*Plain*	<string>	Returns the name of the local group. Win

Examples

- `sids of members of local group "Administrators"`
- Returns a list of the member security IDs of the local administrators group.

- `comment of local group "Administrators"`
- Returns the string "Administrators have complete and unrestricted access to the computer/domain".

- `names of local groups`
- Returns a list of the local groups, such as Administrators, Guests, Users, and others.

Local Group Member

These Inspectors return information (such as security IDs) on members of local groups as defined on the local BES Client computer using the windows NetLocalGroupEnum API, one of Windows Network Management Functions.

Creation Methods

Key Phrase	*Form*	Description
member of <local group>	*Plain*	Returns an object corresponding to a member of the specified local group. Win

Properties

Key Phrase	*Form*	Return Type	Description
<local group member> as string	*Cast*	<string>	Casts a local group member as a string. Win

Examples

- `members of local group "Administrators"`
- Returns a list of the members of the local administration group.

User Objects

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

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
domain user	*PlainGlobal*	Returns all of the users that are members of the domain for which the machine is a user. Win:8.1
domain user <string>	*NamedGlobal*	Returns the user object corresponding to the specified name. Win:8.1
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
local user <string>	*NamedGlobal*	Returns the Active Directory local user object for the specified local user. This allows inspection of AD properties for that user, whether cached or currently logged in. Win:8.1

Key Phrase	*Form*	Description
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
account disabled flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user's account is disabled. Win:8.1
account expiration of <user>	*Plain*	<time>	Returns the time when the specified user's account is set to expire. Win:8.1
accounts operator flag of <user>	*Plain*	<boolean>	Returns TRUE if this user has the accounts operator privilege. Win:8.1
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
admin privilege of <user>	*Plain*	<boolean>	Returns TRUE if the specified user has a privilege level of 'admin'. Win:8.1
allowed workstations string of <user>	*Plain*	<string>	Returns a list of workstations the specified user is allowed to login to. If this string is empty, no restrictions apply. Win:8.1

Key Phrase	*Form*	Return Type	Description
application parameter string of <user>	*Plain*	<string>	Returns a string used by Microsoft products to store user configuration information. Win:8.1
bad password count of <user>	*Plain*	<integer>	Returns the number of attempts to logon to the specified user account with a bad password. Win:8.1
code page of <user>	*Plain*	<integer>	Returns the code page corresponding to the specified user's preferred language. Win:8.1
comment of <user>	*Plain*	<string>	Returns the comments associated with this user's account. Win:8.1
communications operator flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user has communications operator privileges. Win:8.1
country code of <user>	*Plain*	<integer>	Returns the country code of the user's preferred language. Win:8.1
full name of <user>	*Plain*	<string>	Returns the full name of the specified user. Win:8.1
guest privilege of <user>	*Plain*	<boolean>	Returns TRUE if the specified user has a privilege level of 'guest'. Win:8.1
home directory drive of <user>	*Plain*	<string>	Returns the name of the drive assigned to the specified user's home directory. Win:8.1
home directory of <user>	*Plain*	<string>	On a Windows system, this Inspector returns the directory (as a string) where the user files are stored for the specified user. • Note: On a Mac, this Inspector has a different interpretation: it returns the dsAttrTypeStandard:NFSHomeDirectory attribute (as a folder) of the specified user. Win:8.1
home directory required flag of <user>	*Plain*	<boolean>	Returns TRUE if a home directory is required for the specified user. Win:8.1

Key Phrase	*Form*	Return Type	Description
interdomain trust account flag of <user>	*Plain*	<boolean>	Returns TRUE if the given user's account stipulates that a domain should trust other domains. Win:8.1
last logoff of <user>	*Plain*	<time>	Returns the time when the specified user last logged off. Win:8.1
last logon of <user>	*Plain*	<time>	Returns the time when the specified user last logged on. Win:8.1
locked out flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user is currently locked out. Win:8.1
logged on user of <user>	*Plain*	<logged on user>	Converts a user into a 'logged on' user type -- if and only if the specified user is currently logged in. Win:8.1
logon count of <user>	*Plain*	<integer>	Returns the number of times which the specified user's account has successfully logged on to the local machine. Win:8.1
logon script of <user>	*Plain*	<string>	Returns the pathname (as a string) of the specified user's logon script. Win:8.1
logon server of <user>	*Plain*	<string>	Returns the name of the server (as a string) to which logon requests are sent for the specified user's account. Win:8.1
maximum storage of <user>	*Plain*	<integer>	Returns the specified user's disk quota. Will return FALSE if the user has no disk quota. Win:8.1
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
no password required flag of <user>	*Plain*	<boolean>	Returns TRUE if no password is required for the specified user. Win:8.1
normal account flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user's account has a default account type corresponding to a typical user. Win:8.1

Key Phrase	*Form*	Return Type	Description
password age of <user>	*Plain*	<time interval>	Returns a time interval since the specified user's password was last changed. Win:8.1
password change disabled flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user is not allowed to change his password. Win:8.1
password expiration disabled flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user's password on this account will never expire. Win:8.1
password expired of <user>	*Plain*	<boolean>	Returns TRUE if the specified user's password has expired. Win:8.1
primary group id of <user>	*Plain*	<integer>	On a Windows computer, this Inspector returns the group id (as an integer) corresponding to the specified user. • Note: on a Mac, this Inspector returns a string corresponding to the dsAttrTypeStandard:PrimaryGroupID attribute for the specified user. Win:8.1
print operator flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user has print operator privileges. Win:8.1
profile folder of <user>	*Plain*	<string>	Returns the pathname of the folder which contains the specified user's profile. Win:8.1
script flag of <user>	*Plain*	<boolean>	Returns TRUE if the logon script for the specified user has been executed. Win:8.1
server operator flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user has server operator privileges. Win:8.1
server trust account flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user has server trust account privileges (for a backup domain controller). Win:8.1
temporary duplicate account flag of <user>	*Plain*	<boolean>	Returns TRUE if this is a temporary duplicate account of the specified user. Win:8.1

Key Phrase	*Form*	Return Type	Description
user comment of <user>	*Plain*	<string>	Returns the user comments of the specified user. Win:8.1
user id of <user>	*Plain*	<integer>	Returns the specified user's RID number. Win:8.1
user privilege of <user>	*Plain*	<boolean>	Returns TRUE if the specified user has a privilege level of 'user'. Win:8.1
workstation trust account flag of <user>	*Plain*	<boolean>	Returns TRUE if the specified user's account is for a workstation or server. Win:8.1

Examples

- `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

Key Phrase	*Form*	Description
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
logged on user of <user>	*Plain*	Converts a user into a 'logged on' user type -- if and only if the specified user is currently logged in. Win:8.1

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
activity history of <logged on user>	*Plain*	<activity history>	Returns the activity history of the specified logged-on user. This. Win:8.0
name of <logged on user>	*Plain*	<string>	If Terminal Services is available and enabled under NT4/2000/2003/XP/Vista, this Inspector returns the result of WTSQuerySessionInformation with WTSUserName. With Terminal Services disabled, it examines the ACLs on the security descriptor of the "winsta0" window station. Under Windows 9x, returns the "Current User" of "SYSTEM\CurrentControlSet\Control" if it exists. Otherwise returns No Such Object. Win:7.0, 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
sid of <logged on user>	*Plain*	<security identifier>	Returns the Security ID (SID) of the user associated with the session's primary access token. With Windows 2003/XP/Vista, this is determined by WTSQueryUserToken. With NT4/2000 it is determined by the apparent shell process running in the given session. This Inspector may fail if run in a non-privileged context. The SID does not exist under Windows 9x. Win:7.0

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

Activity History

Inspectors of this type keep track of the activity of a single logged-on user. You may iterate over all logged-on users and get the history for each user separately. User information is purged at log off and power off/client off, even if the user immediately logs back in. These Inspectors retrieve information within a tracking window (defaulting to 14 days) or, if the window is still open, the start of that window. Information is in the form of a list of (interval, state) tuples. The first element of the list is the current state of the system. The event lists are fetched from the client each time 'activity history' is referenced, so you should avoid referencing these Inspectors more than once in a relevance statement.

- Note: Activity tracking only works while the Client UI is running. These Inspectors only work with Windows 2000 or better.

Creation Methods

Key Phrase	*Form*	Description
activity history of <logged on user>	*Plain*	Returns the activity history of the specified logged-on user. This. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
user interval of <activity history>	*Plain*	<system power interval>	Returns a power interval (containing a time range and a power state, such as standby or idle) corresponding to the specified user activity history. Win:8.0

Action Objects

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

Action

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

Creation Methods

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

Properties

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

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

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

Examples

- `parameter "action issue date" of action`

  This Inspector returns the date the action was issued, a parameter added to each action by the BigFix/Tivoli Endpoint Manager Console.

Networking Objects

This chapter includes the various networking Inspectors.

Network

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
adapter of <network>	*Plain*	<network adapter>	Returns the one or more network adapter objects of the network. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
any adapter of <network>	*Plain*	<network adapter>	This Inspector returns the same as 'adapter of <network>', but it includes loopback and tunnels. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
connection of <network>	*Plain*	<connection>	Returns a connection to the specified network. This Inspector requires Windows XP or better. Win
dns server of <network>	*Plain*	<network address list>	Returns a list of DNS servers used by the local computer. Win, WM
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

Key Phrase	*Form*	Return Type	Description
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
winsock2 supported of <network>	*Plain*	<boolean>	Indicates that winsock2 is supported by the network. If this returns FALSE, many of the other properties of the interface are not available for inspection. Win, WM

Examples

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

Network Interface

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

Creation Methods

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

Properties

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

Examples

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

Network Ip Interface

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

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
address of <network ip interface>	*Plain*	<ipv4 address>	Returns the ip address of the ip interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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

Key Phrase	*Form*	Return Type	Description
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
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
point to point of <network ip interface>	*Plain*	<boolean>	Indicates that the interface is a point-to-point interface. Usually TRUE for dialup connections. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
subnet address of <network ip interface>	*Plain*	<ipv4 address>	Returns the subnet address (IPv4) to which the specified interface belongs. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
subnet mask of <network ip interface>	*Plain*	<ipv4 address>	Returns the subnet mask (IPv4) of the specified network ip interface. • As of version 8.0, this Inspector type is derived from an <ipv4or6 address> type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
up of <network ip interface>	*Plain*	<boolean>	Returns TRUE if the specified network IP interface is currently up. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Examples

- `names of ip interfaces of network`

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

- `addresses of ip interfaces of network`

  Returns a list of the IP addresses of the network IP interfaces, for example: 127.0.0.1, 192.168.1.100, and more.

- `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.

Network Share

The network share Inspector does not work on Windows 95/98/Me. The password and permission properties are relevant only for shares using share-level security. User-level security is given by the security descriptor. The use limit property will throw NoSuchObject if use is unlimited.

Creation Methods

Key Phrase	*Form*	Description
network share	*PlainGlobal*	Creates a network shared object. Win
network share <string>	*NamedGlobal*	Creates a named network shared object. Win

Properties

Key Phrase	*Form*	Return Type	Description
attribute permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to modify the attributes of the shared resource (such as the date and time when a file was last modified). Win
comment of <network share>	*Plain*	<string>	Returns a string specifying an optional comment about the shared resource. Win
create permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to create an instance of a shared resource (such as a file). Win
delete permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to delete the resource. Win
execute permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to execute the resource. Win
name of <network share>	*Plain*	<string>	Returns a string specifying the name of the specified shared resource. Win

Key Phrase	*Form*	Return Type	Description
password of <network share>	*Plain*	<string>	A string that specifies the share's password (when the server is running with share-level security). Note that the Windows Server 2003 family, Windows XP, Windows 2000, and Windows NT do not support share-level security. Win
path of <network share>	*Plain*	<string>	A string containing the local path for the shared resource. Win
permission permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to modify the permissions (read, write, create, execute, and delete) for the specified network share. Win
read permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to read data from a resource and, by default, to execute the specified network share. Win
security descriptor of <network share>	*Plain*	<security descriptor>	Specifies the security descriptor associated with the specified network share. Win
type of <network share>	*Plain*	<integer>	Specifies an integer value that indicates the type of share. (See the Microsoft document on SHARE_INFO_502). Win
use count of <network share>	*Plain*	<integer>	Specifies an integer value that indicates the number of current connections to the specified network share. Win
use limit of <network share>	*Plain*	<integer>	Specifies an integer value indicating the maximum number of concurrent connections that the shared resource can accommodate. Win
write permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to write data to the specified network share. Win

Network Address List

A network adapter may be configured to respond to a list of network addresses. This object type provides access to such a list.

Creation Methods

Key Phrase	*Form*	Description
address list of <network adapter>	*Plain*	Returns the address list of the network adapter. Win, WM
dns server of <network adapter>	*Plain*	Returns a list of DNS servers used by the specified adapter. Win, WM
dns server of <network>	*Plain*	Returns a list of DNS servers used by the local computer. Win, WM
gateway list of <network adapter>	*Plain*	Returns the gateway network address list of the network adapter. Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
address of <network address list>	*Plain*	<ipv4 address>	Returns the IP address of the address list. Win, WM
cidr address of <network address list>	*Plain*	<string>	Returns the CIDR address of the address list as a string type, for example: 192.168.0.0/16 (IPv4) or 2001:db8::/32 (IPv6). Win:8.0
cidr string of <network address list>	*Plain*	<string>	Returns the Classless Inter-Domain Routing value for the specified network address list as a string value. Win:7.1, WM
subnet address of <network address list>	*Plain*	<ipv4 address>	Returns the subnet address (IPv4) of the specified network address list. Win, WM
subnet mask of <network address list>	*Plain*	<ipv4 address>	Returns the subnet mask (IPv4) of the specified network address list. Win, WM

Network Adapter

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

Creation Methods

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

Properties

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

Key Phrase	*Form*	Return Type	Description
dhcp server of <network adapter>	*Plain*	<ipv4 address>	Returns the ip address of the dhcp server of the network adapter (returns the first address if it is a list). Win, WM
dns server of <network adapter>	*Plain*	<network address list>	Returns a list of DNS servers used by the specified adapter. Win, WM
dns suffix of <network adapter>	*Plain*	<string>	Returns the Domain Name System (DNS) suffix associated with the specified adapter. Win:7.0, WM
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
gateway list of <network adapter>	*Plain*	<network address list>	Returns the gateway network address list of the network adapter. Win, WM
gateway of <network adapter>	*Plain*	<ipv4 address>	Returns the ip address of the gateway of the network adapter. Win, WM
internet connection firewall of <network adapter>	*Plain*	<internet connection firewall>	Creates a Windows XP firewall object. Win
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 dns server of <network adapter>	*Plain*	<ipv4or6 address>	Returns the ipv4or6 address of the DNS server of the specified network adapter. Win:8.0
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 address of <network adapter>	*Plain*	<ipv6 address>	Returns the local IP address (as IPv6) of the adapter. Only for XP/Server 2003 and later. Win:7.0, WM
ipv6 dns server of <network adapter>	*Plain*	<ipv6 address>	Returns the DNS server address (as IPv6) of the adapter. Only for XP/Server 2003 and later. Win:7.0, WM

Key Phrase	*Form*	Return Type	Description
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
lease expires of <network adapter>	*Plain*	<time>	Returns the time that the dhcp lease will expire of the network adapter. Win, WM
lease obtained of <network adapter>	*Plain*	<time>	Returns the time that the dhcp lease was obtained of the network adapter. Win, WM
link speed of <network adapter>	*Plain*	<integer>	This is a property of a network adapter. It returns the maximum speed of the NIC card in bits per second. Win, WM
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
maximum transmission unit of <network adapter>	*Plain*	<integer>	The maximum transmission unit (MTU) size, in bytes, of the specified adapter. Win:7.0, WM
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
primary wins server of <network adapter>	*Plain*	<ipv4 address>	Returns the IPv4 address of the primary wins server of the specified network adapter. Win, WM
secondary wins server of <network adapter>	*Plain*	<ipv4 address>	Returns the IPv4 address of the secondary wins server of the specified network adapter. • As of version 8.0 of BES, this Inspector type is derived from an <ipv4or6 address> type. Win, WM

Key Phrase	*Form*	Return Type	Description
status of <network adapter>	*Plain*	<integer>	The operational status for the interface as defined in RFC 2863. It can be one of the values from the IF_OPER_STATUS enumeration type defined in the Iftypes.h header file. On Windows Vista and later, the header files were reorganized and this enumeration is defined in the Ifdef.h header file. Win:7.0, WM
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
tunnel of <network adapter>	*Plain*	<boolean>	Returns TRUE if the specidfied adapter is a tunnel. Win:8.0
type of <network adapter>	*Plain*	<integer>	Returns the interface type of the specified adapter as defined by the Internet Assigned Names Authority (IANA). Possible values for the interface type are listed in the Ipifcons.h header file. Win:7.0, WM
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
wakeonlan enabled of <network adapter>	*Plain*	<boolean>	Returns TRUE if the specified network adapter is configured to react to Wake-On-Lan requests. Wake-On-Lan is a mechanism used to trigger a boot of a machine in standby mode by sending a special packet. • Note: Wake-On-Lan is only supported for Windows 2000 and XP machines. Win, WM
wins enabled of <network adapter>	*Plain*	<boolean>	Returns TRUE if WINS is enabled on the network adapter. Win, WM

Network Adapter Interface

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

Creation Methods

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

Properties

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

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

Connection

This object is used to query your connections. These are all properties of the Internet Connection Firewall, as returned in the NETCON_PROPERTIES structure.

Creation Methods

Key Phrase	*Form*	Description
connection of <network>	*Plain*	Returns a connection to the specified network. This Inspector requires Windows XP or better. Win

Properties

Key Phrase	*Form*	Return Type	Description
device name of <connection>	*Plain*	<string>	Returns the name of the device associated with the specified connection. Win
firewall of <connection>	*Plain*	<internet connection firewall>	Returns the internet connection firewall object corresponding to the specified connection. Win
guid of <connection>	*Plain*	<string>	Returns the globally-unique identifier (GUID) for the specified connection. Win
media type of <connection>	*Plain*	<media type>	Returns the media type of for the specified connection. Win
name of <connection>	*Plain*	<string>	Returns the name of the specified connection. Win
status of <connection>	*Plain*	<connection status>	Returns the status of the specified connection. Win

Connection Status

This object returns information about the status of your connections.

Creation Methods

Key Phrase	*Form*	Description
connection status <integer>	*NumberedGlobal*	Returns the connection status based on its integer value. This Inspector is included to take advantage of new (or undocumented) additions to the status values. Win
connection status authenticating	*PlainGlobal*	Returns the value NCS_AUTHENTICATING: The connection is waiting for authentication to occur. Win
connection status authentication failed	*PlainGlobal*	Returns the value NCS_AUTHENTICATION_FAILED: Authentication has failed on this connection. Win
connection status authentication succeeded	*PlainGlobal*	Returns the value NCS_AUTHENTICATION_SUCCEEDED: Authentication has succeeded on this connection. Win
connection status connected	*PlainGlobal*	Returns the value NCS_CONNECTED: The connection is in a connected state. Win
connection status connecting	*PlainGlobal*	Returns the value NCS_CONNECTING: The connection is in the process of connecting. Win
connection status disconnected	*PlainGlobal*	Returns the value NCS_DISCONNECTED: The connection is disconnected. Win
connection status disconnecting	*PlainGlobal*	Returns the value NCS_DISCONNECTING: The connection is in the process of disconnecting. Win
connection status hardware disabled	*PlainGlobal*	Returns the value NCS_HARDWARE_DISABLED: The hardware for the connection is present, but is not enabled. Win
connection status hardware malfunction	*PlainGlobal*	Returns the value NCS_HARDWARE_MALFUNCTION: A malfunction has occurred in the hardware for the connection. Win

Key Phrase	*Form*	Description
connection status media disconnected	*PlainGlobal*	Returns the value NCS_MEDIA_DISCONNECTED: The media, for example the network cable, is disconnected. Win
connection status no hardware present	*PlainGlobal*	Returns the value NCS_NO_HARDWARE_PRESENT: The hardware for the connection, for example network interface card (NIC), is not present. Win
status of <connection>	*Plain*	Returns the current status of the connection. You can compare this status to the 'connection status' global objects, which act as constants. Some of the possible values include: • NCS_DISCONNECTED: The connection is disconnected. • NCS_CONNECTING: The connection is in the process of connecting. • NCS_CONNECTED: The connection is in a connected state. • NCS_DISCONNECTING: The connection is in the process of disconnecting. • NCS_HARDWARE_NOT_PRESENT: The hardware for the connection, for example network interface card (NIC), is not present. • NCS_HARDWARE_DISABLED: The hardware for the connection is present, but is not enabled. • NCS_HARDWARE_MALFUNCTION: A malfunction has occurred in the hardware for the connection. • NCS_MEDIA_DISCONNECTED: The media, for example the network cable, is disconnected. • NCS_AUTHENTICATING: The connection is waiting for authentication to occur. • NCS_AUTHENTICATION_SUCCEEDED: Authentication has succeeded on this connection. • NCS_AUTHENTICATION_FAILED: Authentication has failed on this connection. • NCS_INVALID_ADDRESS: The address is invalid. • NCS_CREDENTIALS_REQUIRED: Security credentials are required. Win

Operators

Key phrase	Return Type	Description
<connection status> = <connection status>	*<boolean>*	Compare the statuses of two connections. Win

Internet Protocol

Returns the firewall internet protocol corresponding to the Microsoft enumerated types, either tcp or udp. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
internet protocol <integer>	*NumberedGlobal*	Returns the firewall internet protocol specified, either tcp or udp. These correspond to the Microsoft Windows Firewall enumerated types: • NET_FW_IP_PROTOCOL_TCP • NET_FW_IP_PROTOCOL_UDP. Win
protocol of <firewall open port>	*Plain*	Returns the Protocol property of the firewall open port. Win
protocol of <firewall rule>	*Plain*	Retrieves the Protocol property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
tcp	*PlainGlobal*	Returns an internet protocol corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_IP_PROTOCOL_TCP. Win
udp	*PlainGlobal*	Returns an internet protocol corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_IP_PROTOCOL_UDP. Win

Operators

Key phrase	Return Type	Description
<internet protocol> = <internet protocol>	<*boolean*>	Compares two firewall internet protocols for equality only. Win

Examples

- `exists globally open port whose (port of it = 52311 and protocol of it = udp and enabled of it) of current profile of local policy of firewall`

  Returns TRUE if the BES Client can receive pings.

Ip Version

The ip version inspectors distinguish between ipv6 and ipv4. The integer representations of these are chosen to meet existing standards.

Creation Methods

Key Phrase	*Form*	Description
ip version of <firewall authorized application>	*Plain*	Returns the the IP version for the specified firewall authorized application. Win
ip version of <firewall open port>	*Plain*	Returns the the IP version for the specified firewall open port. Win
ip version of <firewall service>	*Plain*	Returns the the IP version for the specified firewall service. Win

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/Tivoli Endpoint Manager, 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 address list>	*Plain*	Returns the ip address of the network adapter list. Win, WM
address of <network ip interface>	*Plain*	Creates an object with the ip address of the interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
broadcast address of <network ip interface>	*Plain*	Returns the broadcast address of the specified interface. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
dhcp server of <network adapter>	*Plain*	Returns the ip address of the dhcp server of the network adapter. Win, WM
gateway of <network adapter>	*Plain*	Returns the ip address of the gateway of the network adapter. Win, WM
primary wins server of <network adapter>	*Plain*	Returns the ip address of the primary wins server of the network adapter. Win, WM
secondary wins server of <network adapter>	*Plain*	Returns the IPv4 address of the secondary wins server of the specified network adapter. Win, WM
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 address list>	*Plain*	Returns the subnet address of the network address list. Win, WM
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 address list>	*Plain*	Returns the subnet mask (IPv4) of the network address list. Win, WM
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
target ip address of <port mapping>	*Plain*	Returns the target IPv4 address of the specified port mapping. Win

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.

Ipv6 Address

These Inspectors deal with the Internet Protocol addressing scheme, version 6.

- NOTE: Prior to version 8.0 of BigFix/Tivoli Endpoint Manager, 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
ipv6 address of <network adapter>	*Plain*	Returns the local IPv6 address of the adapter. Only for XP/Server 2003 and later. Win:7.0, WM
ipv6 dns server of <network adapter>	*Plain*	Returns the DNS server IPv6 address of the adapter. Only for XP/Server 2003 and later. Win:7.0, WM

Ipv4or6 Address

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

Creation Methods

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

Key Phrase	*Form*	Description
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
ipv4or6 dns server of <network adapter>	*Plain*	Returns the ipv4or6 address of the DNS server of the specified network adapter. Win:8.0
registration address of <client>	*Plain*	This Inspector returns the IP address (as an <ipv4or6 address> type) that the specified BigFix/Tivoli Endpoint Manager 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/Tivoli Endpoint Manager 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
target ipv4or6 address of <port mapping>	*Plain*	Returns the target IP address (ipv4or6) of the specified port mapping. Win:8.0

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
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

Firewall Objects

These Inspectors retrieve the various firewall settings for the BES Client computer.

Firewall

The Firewall Inspectors allow you to view the settings of the Windows Firewall on Windows Clients. By inspecting the firewall properties, you can determine which applications have access to unsolicited traffic and how the firewall is configured for various subsets of your network. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the MSDN site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
firewall	*PlainGlobal*	Returns the global firewall object for this computer. Win

Properties

Key Phrase	*Form*	Return Type	Description
current profile type of <firewall>	*Plain*	<firewall profile type>	Returns the current profile type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_PROFILE_TYPE. Win
local policy modify state of <firewall>	*Plain*	<firewall local policy modify state>	Determines if adding or setting a rule or group of rules will take effect in the specified firewall profile. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
local policy of <firewall>	*Plain*	<firewall policy>	Returns the local policy of the specified firewall. Win
rule group currently enabled <string> of <firewall>	*Named*	<boolean>	Determines whether a specified group of firewall rules is enabled or disabled for the current profile, considering the firewall's state, BlockAllInboundTraffic state and group policy overrides state. The string is used to group rules together. It can be the group name or an indirect string to the group name in the form of "@yourresourcedll.dll,-23255." Rules belonging to this group will be queried. Win:7.0

Key Phrase	*Form*	Return Type	Description
rule of <firewall>	*Plain*	<firewall rule>	Retrieves the collection of rules for the specified firewall. For more information, see the MSDN Library article on INetFwPolicy2. Win:7.0
service restriction of <firewall>	*Plain*	<firewall service restriction>	Retrieves the access interface to manipulate the Windows Service Hardening store. On Windows operating systems earlier than Vista, this Inspector always returns no-such-object. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0

Firewall Authorized Application

These Inspectors apply to applications that are authorized to exchange traffic through the Windows Firewall. These properties correspond to the INetFwAuthorizedApplication interface in the Windows Firewall API. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
authorized application of <firewall profile>	*Plain*	Provides access to the properties of any application that has been authorized to have firewall openings. There are several methods attached to this Inspector, including: • Name • ProcessImageFileName • IpVersion • Scope • RemoteAddress • Enabled. Win

Properties

Key Phrase	*Form*	Return Type	Description
enabled of <firewall authorized application>	*Plain*	<boolean>	Returns the contents of the Enabled property for the specified application. Returns TRUE if the settings for this application are currently enabled. Win

Key Phrase	*Form*	Return Type	Description
ip version of <firewall authorized application>	*Plain*	<ip version>	Returns the contents of the IpVersion property for the specified application. Win
name of <firewall authorized application>	*Plain*	<string>	Returns the contents of the Friendly Name property for the specified application. Win
process image file name of <firewall authorized application>	*Plain*	<string>	Returns the contents of the ProcessImageFileName property for the specified application. Win
remote addresses of <firewall authorized application>	*Plain*	<string>	Returns the contents of the RemoteAddresses property for the specified application. This property accesses a set of remote addresses that an application can use to listen for traffic. Win
scope of <firewall authorized application>	*Plain*	<firewall scope>	Returns the contents of the Scope property for the specified application. This property controls the network scope that a port can listen to. Win

Firewall Profile

These Inspectors provide access to the firewall profile. These properties correspond to the INetFwProfile interface in the Windows Firewall API. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
current profile of <firewall policy>	*Plain*	Returns the profile currently in effect for the specified firewall policy. Win
domain profile of <firewall policy>	*Plain*	Returns the domain profile of the specified firewall policy. The domain profile settings are used when a computer is connected to a network that contains the organization's domain controllers. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win

Key Phrase	*Form*	Description
private profile of <firewall policy>	*Plain*	Retrieves the private profile type from the specified firewall policy. This profile type is used for home and other private network types. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
public profile of <firewall policy>	*Plain*	Retrieves the public profile type from the specified firewall policy. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
standard profile of <firewall policy>	*Plain*	Returns the standard profile of the specified firewall policy. The standard profile settings are used when a computer is connected to a network that does not contain the organization's domain controllers. Win

Properties

Key Phrase	*Form*	Return Type	Description
authorized application of <firewall profile>	*Plain*	<firewall authorized application>	Access to the AuthorizedApplications collection for this profile. This Inspector can be iterated through all the authorized applications in each firewall profile. Win
exceptions allowed of <firewall profile>	*Plain*	<boolean>	Returns the property that indicates whether exceptions should be allowed by the firewall. Win
excluded interface of <firewall profile>	*Plain*	<string>	Gets the value of the ExcludedInterfaces property from the specified firewall profile. This property contains the list of interfaces excluded from a the profile's firewall rules. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
firewall enabled of <firewall profile>	*Plain*	<boolean>	Gets the value of the FirewallEnabled setting. Win
globally open port of <firewall profile>	*Plain*	<firewall open port>	Provides access to the GloballyOpenPorts collection for this profile. This property can be iterated. Win
icmp settings of <firewall profile>	*Plain*	<firewall icmp settings>	Gets the object governing settings for ICMP packets. Win

Key Phrase	*Form*	Return Type	Description
inbound connections allowed of <firewall profile>	*Plain*	<boolean>	Determines whether the default action for inbound traffic for the specified firewall profile is NET_FW_ACTION_ALLOW. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
notifications disabled of <firewall profile>	*Plain*	<boolean>	Gets the value of the NotificationsDisabled setting, TRUE or FALSE. Win
outbound connections allowed of <firewall profile>	*Plain*	<boolean>	Determines whether the default action for outbound traffic for the specified firewall profile is NET_FW_ACTION_ALLOW. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
remote admin settings of <firewall profile>	*Plain*	<firewall remote admin settings>	Gets the object containing the remote administration settings. Win
rule group enabled <string> of <firewall profile>	*Named*	<boolean>	Determines whether a specified group of firewall rules are enabled or disabled. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
service of <firewall profile>	*Plain*	<firewall service>	Gets the collection containing the services for this profile. This Inspector can be iterated over all services. Win
type of <firewall profile>	*Plain*	<firewall profile type>	Returns the type of the specified firewall profile: domain, standard or current. Win
unicast responses to multicast broadcast disabled of <firewall profile>	*Plain*	<boolean>	Gets the value of the UnicastResponsesToMulticastBroadcastDisabled setting. Win

Examples

- `firewall enabled of current profile of local policy of firewall`

Verify that the firewall is enabled.

Firewall Profile Type

These Inspectors provide access to the firewall profile type. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
current firewall profile type	*PlainGlobal*	Retrieves the type of firewall profile that is currently in effect. Win
current profile type of <firewall>	*Plain*	Returns the current profile type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_PROFILE_TYPE. Win
domain firewall profile type	*PlainGlobal*	Returns the current profile type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_PROFILE_DOMAIN. Win
firewall profile type <integer>	*NumberedGlobal*	Returns the firewall profile type corresponding to the given integer: • 0: Domain • 1: Standard • 2: Current. Win
private firewall profile type	*PlainGlobal*	Retrieves the enumerated variable corresponding to private profile type. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
public firewall profile type	*PlainGlobal*	Retrieves the enumerated variable corresponding to public profile type. This profile type is used for public internet access points. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
standard firewall profile type	*PlainGlobal*	Returns the Standard firewall profile type. Win
type of <firewall profile>	*Plain*	Returns the type of the specified firewall profile, corresponding to the Microsoft Windows Firewall enumerated types: • NET_FW_PROFILE_DOMAIN • NET_FW_PROFILE_STANDARD • NET_FW_PROFILE_CURRENT. Win

Operators

Key phrase	Return Type	Description
<firewall profile type> = <firewall profile type>	<*boolean*>	Compares two firewall policies for equality only. Win

Examples

- `current profile type of firewall = domain firewall profile type`
- Returns TRUE if the current profile type is domain.

- `current profile type of firewall = standard firewall profile type`
- Returns TRUE if the current profile type is standard.

Firewall Policy

These Inspectors provide access to the local, current, domain and standard firewall policies. These properties correspond to the INetFwPolicy interface in the Windows Firewall API. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
local policy of <firewall>	*Plain*	Returns the local policy of the specified firewall. Win

Properties

Key Phrase	*Form*	Return Type	Description
current profile of <firewall policy>	*Plain*	<firewall profile>	Returns the profile currently in effect for the specified firewall policy. Win
domain profile of <firewall policy>	*Plain*	<firewall profile>	Returns the domain profile of the specified firewall policy. The domain profile settings are used when a computer is connected to a network that contains the organization's domain controllers. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win

Key Phrase	*Form*	Return Type	Description
private profile of <firewall policy>	*Plain*	<firewall profile>	Retrieves the private profile type from the specified firewall policy. This profile type is used for home and other private network types. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
public profile of <firewall policy>	*Plain*	<firewall profile>	Retrieves the public profile type from the specified firewall policy. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
standard profile of <firewall policy>	*Plain*	<firewall profile>	Returns the standard profile of the specified firewall policy. The standard profile settings are used when a computer is connected to a network that does not contain the organization's domain controllers. This Inspector corresponds to the Microsoft Windows Firewall enumerated type NET_FW_PROFILE_STANDARD. Win

Firewall Scope

These Inspectors provide access to the firewall scope. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
all firewall scope	*PlainGlobal*	Returns the scope of computers that allow ALL traffic through the firewall, corresponding to the Microsoft enumerated type NET_FW_SCOPE_ALL. Win
custom firewall scope	*PlainGlobal*	Returns the custom firewall scope, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SCOPE_CUSTOM. Win

Key Phrase	*Form*	Description
firewall scope <integer>	*NumberedGlobal*	Returns the scope of addresses from which a port can listen, corresponding to the Microsoft enumerated types: • NET_FW_SCOPE_ALL • NET_FW_SCOPE_LOCAL_SUBNET • NET_FW_SCOPE_CUSTOM. Win
local subnet firewall scope	*PlainGlobal*	Returns the local subnet firewall scope, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SCOPE_LOCAL_SUBNET. Win
scope of <firewall authorized application>	*Plain*	Retrieves the contents of the Scope property of the authorized application. Win
scope of <firewall open port>	*Plain*	Retrieves the contents of the Scope property of the open port. Win
scope of <firewall service>	*Plain*	Retrieves the contents of the Scope property of the firewall service. Win

Operators

Key phrase	Return Type	Description
<firewall scope> = <firewall scope>	*<boolean>*	Compares two firewall scopes for equality only. Win

Firewall Open Port

These Inspectors provide access to the properties of a port that has been opened in the Windows Firewall. These properties correspond to the INetFwOpenPort interface in the Windows Firewall API. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
globally open port of <firewall profile>	*Plain*	Provides access to the GloballyOpenPorts collection for this profile. Win
globally open port of <firewall service>	*Plain*	Returns the collection of globally open ports associated with the firewall service. Win

Properties

Key Phrase	*Form*	Return Type	Description
built in of <firewall open port>	*Plain*	<boolean>	Returns the contents of the BuiltIn property of the firewall open port. Win
enabled of <firewall open port>	*Plain*	<boolean>	Returns the contents of the Enabled property of the firewall open port. Win
ip version of <firewall open port>	*Plain*	<ip version>	Returns the IpVersion property of the firewall open port. Win
name of <firewall open port>	*Plain*	<string>	Returns the Name property of the firewall open port. Win
port of <firewall open port>	*Plain*	<integer>	Returns the Port property of the firewall open port. Win
protocol of <firewall open port>	*Plain*	<internet protocol>	Returns the Protocol property of the firewall open port. Win
remote addresses of <firewall open port>	*Plain*	<string>	Returns the RemoteAddresses property of the firewall open port. Win

Key Phrase	*Form*	Return Type	Description
scope of <firewall open port>	*Plain*	<firewall scope>	Returns the Scope property of the firewall open port. Win

Firewall Service

These Inspectors provide access to the properties of a service that may be authorized to listen through the firewall. These properties correspond to the INetFwService interface in the Windows Firewall API. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
service of <firewall profile>	*Plain*	Gets the collection containing the services for this profile. Win

Properties

Key Phrase	*Form*	Return Type	Description
customized of <firewall service>	*Plain*	<boolean>	Returns a flag that indicates whether at least one of the ports associated with the service has been customized. Either TRUE or FALSE. Win
enabled of <firewall service>	*Plain*	<boolean>	Returns the enabled flag for the specified firewall service. Win
globally open port of <firewall service>	*Plain*	<firewall open port>	Returns the collection of globally open ports associated with the firewall service. Win
ip version of <firewall service>	*Plain*	<ip version>	Returns the the IP version for the specified firewall service. Win
name of <firewall service>	*Plain*	<string>	Returns the friendly name of the firewall service. Win
remote addresses of <firewall service>	*Plain*	<string>	Returns the contents of the RemoteAddresses property for the specified firewall service. Win

Key Phrase	*Form*	Return Type	Description
scope of <firewall service>	*Plain*	<firewall scope>	Retrieves the contents of the Scope property of the firewall service. Win
type of <firewall service>	*Plain*	<firewall service type>	Returns the type of the specified firewall service (file and print, upnp, remote desktop or none). Win

Examples

- `exists globally open port whose (port of it = 52311 and protocol of it = udp and enabled of it) of current profile of local policy of firewall`

  Returns TRUE if the BES Client can receive pings.

Firewall Service Type

These Inspectors provide access to the firewall service type. These properties correspond to the INetFwService.Type interface in the Windows Firewall API. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
file_and_print firewall service type	*PlainGlobal*	Returns the global service type for file and print sharing, corresponding to the Microsoft enumerated type: NET_FW_SERVICE_FILE_AND_PRINT. Win
firewall service type <integer>	*NumberedGlobal*	Returns the firewall service type specified by <integer>, corresponding to the Microsoft Windows Firewall enumerated types: • NET_FW_SERVICE_FILE_AND_PRINT • NET_FW_SERVICE_UPNP • NET_FW_SERVICE_REMOTE_DESKTOP • NET_FW_SERVICE_NONE. Win
none firewall service type	*PlainGlobal*	Returns the no firewall service type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SERVICE_NONE. Win

Key Phrase	*Form*	Description
remote desktop firewall service type	*PlainGlobal*	Returns the remote desktop firewall service type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SERVICE_REMOTE_DESKTOP. Win
type of <firewall service>	*Plain*	Returns the type of the specified firewall service, corresponding to the Microsoft Windows Firewall enumerated types: • NET_FW_SERVICE_FILE_AND_PRINT • NET_FW_SERVICE_UPNP • NET_FW_SERVICE_REMOTE_DESKTOP • NET_FW_SERVICE_NONE. Win
upnp firewall service type	*PlainGlobal*	Returns the UPnP (Universal Plug and Play) firewall service type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SERVICE_UPNP. • Note: UPnP is not the same as PnP. UPnP is used for network connectivity via TCP/IP to various devices (scanners, printers, and more.). Win

Operators

Key phrase	Return Type	Description
<firewall service type> = <firewall service type>	*<boolean>*	Compares two firewall service types for equality only. Win

Examples

- `names of services of current profile of local policy of firewall`

Returns the names of the services of the current profile of the firewall's local policy, such as File and Printer Sharing, UPnP Framework & Remote Desktop.

Firewall Icmp Settings

These Inspectors provide access to the settings controlling Internet Control Message Protocol (ICMP) packets. These properties correspond to the INetFwIcmpSettings interface in the Windows Firewall API. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
icmp settings of <firewall profile>	*Plain*	The Microsoft Windows Firewall INetFwIcmpSettings interface provides access to the settings controlling Internet Control Message Protocol (ICMP) packets. Win

Properties

Key Phrase	*Form*	Return Type	Description
allow inbound echo request of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowInboundEchoRequest property. Type common to IPv4 and IPv6. Win
allow inbound mask request of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowInboundMaskRequest property. Type common to IPv4 only. Win
allow inbound router request of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowInboundRouterRequest property. Type common to IPv4 only. Win
allow inbound timestamp request of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowInboundTimestampRequest property. Type common to IPv4 only. Win
allow outbound destination unreachable of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundDestinationUnreachable property. Type common to IPv4 and IPv6. Win
allow outbound packet too big of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundPacketTooBig property. Type common to IPv6 only. Win

Key Phrase	*Form*	Return Type	Description
allow outbound parameter problem of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundParameterProblem property. Type common to IPv4 and IPv6. Win
allow outbound source quench of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundSourceQuench property. Type common to IPv4 only. Win
allow outbound time exceeded of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundTimeExceeded property. Type common to IPv4 and IPv6. Win
allow redirect of <firewall icmp settings>	*Plain*	<boolean>	Accesses the AllowRedirect property. Type common to IPv4 and IPv6. Win

Firewall Remote Admin Settings

These Inspectors provide access to the settings that control remote administration. These properties correspond to the INetFwRemoteAdminSettings interface in the Windows Firewall API. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
remote admin settings of <firewall profile>	*Plain*	Gets the object containing the remote administration settings. These settings include the following properties: • Enabled • IpVersion • RemoteAddresses • Scope. Win

Firewall Action

The firewall action Inspectors provide wrappers around the Windows Vista Firewall API. These are Vista-only Inspectors.

Creation Methods

Key Phrase	*Form*	Description
action of <firewall rule>	*Plain*	Retrieves the Action property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
allow firewall action	*PlainGlobal*	Returns a value corresponding to the NET_FW_ACTION enumerated type, which specifies the action for a rule or default setting. This is a Vista Inspector. For more information, see the MSDN Library entry for NET_FW_ACTION. Win:7.0
block firewall action	*PlainGlobal*	Returns a value corresponding to the NET_FW_ACTION enumerated type, which specifies the action for a rule or default setting. This is a Vista Inspector. For more information, see the MSDN Library entry for NET_FW_ACTION. Win:7.0
firewall action <integer>	*NumberedGlobal*	Returns the Nth enumerated type from the Firewall Action interface. This provides an alternative way to access the firewall action variables and requires Vista. For more information, see the MSDN Library entry for NET_FW_ACTION. Win:7.0

Operators

Key phrase	Return Type	Description
<firewall action> = <firewall action>	*<boolean>*	Compares two enumerated types corresponding to the firewall action. This is a Vista Inspector. For more information see the MSDN Library reference to NET_FW_ACTION. Win:7.0

Firewall Local Policy Modify State

These Inspectors provide a wrapper for the Windows Firewall Policy Modification State. They are Vista-only. On a non-Vista machine, they return no results. For more information see the MSDN Library reference to NET_FW_MODIFY_STATE at http://msdn.microsoft.com/library/.

Creation Methods

Key Phrase	*Form*	Description
firewall local policy modify state <integer>	*NumberedGlobal*	Returns a value corresponding to the NET_FW_MODIFY_STATE enumerated type, which specifies the effect of modifications to the current policy. The <integer> specifies the zero-based enumeration, and provides an alternative access method. For more information, see the MSDN Library entry for NET_FW_MODIFY_STATE. Win:7.0
gp override firewall local policy modify state	*PlainGlobal*	Returns a value corresponding to the NET_FW_MODIFY_STATE enumerated type, which specifies the effect of modifications to the current policy. For more information, see the MSDN Library entry for NET_FW_MODIFY_STATE. Win:7.0
inbound blocked firewall local policy modify state	*PlainGlobal*	Returns a value corresponding to the NET_FW_MODIFY_STATE enumerated type, which specifies the effect of modifications to the current policy. For more information, see the MSDN Library entry for NET_FW_MODIFY_STATE. Win:7.0
local policy modify state of <firewall>	*Plain*	Determines if adding or setting a rule or group of rules will take effect in the specified firewall profile. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
ok firewall local policy modify state	*PlainGlobal*	Returns a value corresponding to the NET_FW_MODIFY_STATE enumerated type, which specifies the effect of modifications to the current policy. For more information, see the MSDN Library entry for NET_FW_MODIFY_STATE. Win:7.0

Operators

Key phrase	Return Type	Description
<firewall local policy modify state> = <firewall local policy modify state>	<*boolean*>	Compares two enumerated types corresponding to the firewall policy modification state. This is a Vista Inspector. For more information see the MSDN Library reference to NET_FW_MODIFY_STATE. Win:7.0

Firewall Rule

These Inspectors provide a wrapper for the Windows Firewall Rule. They are Vista-only. On a non-Vista machine, they return no results. For more information, see the MSDN library for INetFwRule at http://msdn.microsoft.com/library/.

Creation Methods

Key Phrase	*Form*	Description
rule of <firewall service restriction>	*Plain*	Retrieves the collection of Windows Service Hardening network rules for the specified firewall service restriction. For more information, see the MSDN Library section on INetFwServiceRestriction. Win:7.0
rule of <firewall>	*Plain*	Retrieves the collection of rules for the specified firewall. For more information, see the MSDN Library article on INetFwPolicy2. Win:7.0

Properties

Key Phrase	*Form*	Return Type	Description
action of <firewall rule>	*Plain*	<firewall action>	Retrieves the Action property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
application name of <firewall rule>	*Plain*	<string>	Retrieves the application name property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0

Key Phrase	*Form*	Return Type	Description
currently active of <firewall rule>	*Plain*	<boolean>	Currently active is true if and only if the specified firewall rule is enabled (the Enabled property is true) AND the currently active profile type (as defined by the CurrentProfileTypes property of INetFwPolicy2) is one of the profiles for which the rule applies (as defined by the Profiles property). For more information see the MSDN Library articles for INetFwRule, including the CurrentProfileTypes and Profiles properties. Win:7.0
description of <firewall rule>	*Plain*	<string>	Retrieves the Description property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
edge traversal allowed of <firewall rule>	*Plain*	<boolean>	Retrieves the EdgeTraversal property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
enabled of <firewall rule>	*Plain*	<boolean>	Retrieves the Enabled property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
grouping of <firewall rule>	*Plain*	<string>	Retrieves the Grouping property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
icmp types_and_codes string of <firewall rule>	*Plain*	<string>	Retrieves the IcmpTypesAndCodes property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0

Key Phrase	*Form*	Return Type	Description
inbound of <firewall rule>	*Plain*	<boolean>	Retrieves the inbound direction property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule and IDispatch. Win:7.0
interface of <firewall rule>	*Plain*	<string>	Retrieves the Interfaces property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
interface types string of <firewall rule>	*Plain*	<string>	Retrieves the Interface types property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
local addresses string of <firewall rule>	*Plain*	<string>	Retrieves the Interfaces property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
local ports string of <firewall rule>	*Plain*	<string>	Retrieves the LocalPorts property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
name of <firewall rule>	*Plain*	<string>	Retrieves the Name property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
outbound of <firewall rule>	*Plain*	<boolean>	Retrieves the outbound direction property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule and IDispatch. Win:7.0

Key Phrase	*Form*	Return Type	Description
profile <firewall profile type> of <firewall rule>	*Index<firewall profile type>*	<boolean>	Retrieves the given profile property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
protocol of <firewall rule>	*Plain*	<internet protocol>	Retrieves the Protocol property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
remote addresses string of <firewall rule>	*Plain*	<string>	Retrieves the RemoteAddresses property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
remote ports string of <firewall rule>	*Plain*	<string>	Retrieves the RemotePorts property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
service name of <firewall rule>	*Plain*	<string>	Retrieves the ServiceName property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0

Firewall Service Restriction

These Inspectors provide a wrapper for the Windows Firewall Service Restriction. They are Vista-only. On a non-Vista machine, they return no results. For more information see the MSDN Library reference to INetFwServiceRestriction at http://msdn.microsoft.com/library/.

Creation Methods

Key Phrase	*Form*	Description
service restriction of <firewall>	*Plain*	Retrieves the access interface to manipulate the Windows Service Hardening store. On Windows operating systems earlier than Vista, this Inspector always returns no-such-object. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0

Properties

Key Phrase	*Form*	Return Type	Description
rule of <firewall service restriction>	*Plain*	<firewall rule>	Retrieves the collection of Windows Service Hardening network rules for the specified firewall service restriction. For more information, see the MSDN Library section on INetFwServiceRestriction. Win:7.0
service restricted <(string, string)> of <firewall service restriction>	*Index<(string, string)>*	<boolean>	Returns the service restriction state of a given service. There are two strings passed to define the restriction: the service name and the application name. For more information, see the MSDN Library section on INetFwServiceRestriction. Win:7.0

Internet Connection Firewall

These Inspectors provide access to the settings of the Internet Connection Firewall introduced in Windows XP. The Internet Connection Firewall helps to protect a computer that is directly connected to the Internet, or a home network, from network attacks.

Creation Methods

Key Phrase	*Form*	Description
firewall of <connection>	*Plain*	Returns the internet connection firewall object corresponding to the specified connection. Win
internet connection firewall of <network adapter>	*Plain*	Creates a Windows XP firewall object. Win

Properties

Key Phrase	*Form*	Return Type	Description
enabled of <internet connection firewall>	*Plain*	<boolean>	Returns TRUE if the local computer has the Windows XP built-in firewall enabled. Win
port mapping of <internet connection firewall>	*Plain*	<port mapping>	Creates a port mapping object for the built-in firewall. Win

Port Mapping

Refers to a port mapping object for the built-in firewall.

Creation Methods

Key Phrase	*Form*	Description
port mapping of <internet connection firewall>	*Plain*	Creates a port mapping object for the built-in firewall. Win

Properties

Key Phrase	*Form*	Return Type	Description
enabled of <port mapping>	*Plain*	<boolean>	A boolean indicating whether or not the port mapping is enabled. Win
external port of <port mapping>	*Plain*	<integer>	Returns the external port number of the specified ICF port mapping. Win
internal port of <port mapping>	*Plain*	<integer>	Returns the internal port number of the specified ICF port mapping. Win
name of <port mapping>	*Plain*	<string>	Returns the name of the specified port mapping. Win
options of <port mapping>	*Plain*	<integer>	See port mapping at MSDN. Win
protocol of <port mapping>	*Plain*	<string>	Returns a string like "tcp" or "udp", corresponding to the protocol of the specified port mapping. Win
target ip address of <port mapping>	*Plain*	<ipv4 address>	Returns the target IPv4 address of the specified port mapping. Win
target ipv4or6 address of <port mapping>	*Plain*	<ipv4or6 address>	Returns the target IP address (ipv4or6) of the specified port mapping. Win:8.0
target name of <port mapping>	*Plain*	<string>	Returns the target name of the specified port mapping. Win

Media Type

This object allows you to inspect the media type of your Internet Connection Firewall type connection, as returned in the NETCON_PROPERTIES structure.

Creation Methods

Key Phrase	*Form*	Description
media type <integer>	*NumberedGlobal*	Returns the media type based on its integer value. This Inspector is included to take advantage of new (or undocumented) additions to the media types. Win
media type bridge	*PlainGlobal*	Returns the value NCM_BRIDGE: Bridged connection. Win
media type direct	*PlainGlobal*	Returns the value NCM_DIRECT: Direct serial connection through a serial port. Win
media type isdn	*PlainGlobal*	Returns the value NCM_ISDN: Connection is through an integrated services digital network (ISDN) line. Win
media type lan	*PlainGlobal*	Returns the value NCM_LAN: Connection is to a local area network (LAN). Win
media type of <connection>	*Plain*	Returns the media type of the specified connection: • NCM_NONE: No media is present. • NCM_DIRECT: Direct serial connection through a serial port. • NCM_ISDN: Connection is through an integrated services digital network (ISDN) line. • NCM_LAN: Connection is to a local area network (LAN). • NCM_PHONE: Dial-up connection over a conventional phone line. • NCM_TUNNEL: Virtual private network (VPN) connection. • NCM_PPPOE: Point-to-Point protocol (PPP) over Ethernet. • NCM_BRIDGE: Bridged connection. • NCM_SHAREDACCESSHOST_LAN: Shared connection to a LAN. • NCM_SHAREDACCESSHOST_RAS: Shared connection to a remote or wide area network (WAN). Win

Key Phrase	*Form*	Description
media type phone	*PlainGlobal*	Returns the value NCM_PHONE: Dial-up connection over a conventional phone line. Win
media type pppoe	*PlainGlobal*	Returns the value NCM_PPPOE: Point-to-Point protocol (PPP) over Ethernet. Win
media type shared access host lan	*PlainGlobal*	Returns the value NCM_SHAREDACCESSHOST_LAN: Shared connection to a LAN. Win
media type shared access host ras	*PlainGlobal*	Returns the value NCM_SHAREDACCESSHOST_RAS: Shared connection to a remote or wide area network (WAN). Win
media type tunnel	*PlainGlobal*	Returns the value NCM_TUNNEL: Virtual private network (VPN) connection. Win

Operators

Key phrase	Return Type	Description
<media type> = <media type>	*<boolean>*	Compares two media types. Win

Task Objects

These Inspectors help you write Relevance expressions to deal with the Windows Task Manager and its various triggers.

Task Definition

These Inspectors give access to the components of a task, such as the settings, triggers, actions and registration information.

Creation Methods

Key Phrase	*Form*	Description
definition of <scheduled task>	*Plain*	Returns the task definition for the specified scheduled task for both the 1.0 and 2.0 interface. These definitions include the following properties: • Actions: a list of the actions performed by the task • Data: information associated with the task • Principal: refers to the security credentials for the task. • RegistrationInfo: data such as the author & date of registration • Settings: info on how the scheduler performs the task • Triggers: a list of the possible triggers that can start the task • XMLText: the XML-formatted definition of the task. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
action of <task definition>	*Plain*	<task action>	Returns the action(s) performed by the task specified by the given definition. Win:8.0
data of <task definition>	*Plain*	<string>	Returns the data associated with the task specified by the given definition. Win:8.0
principal of <task definition>	*Plain*	<task principal>	Returns the principle for the task that provides the security credentials for the task associated with the specified task definition. Win:8.0

Key Phrase	*Form*	Return Type	Description
registration info of <task definition>	*Plain*	<task registration info>	Returns data such as the author & date of registration associated with the specified task definition. Win:8.0
setting of <task definition>	*Plain*	<task settings>	Returns the settings associated with the given task. These settings include dealing with starts, restarts and stops, how to manage hidden or multiple tasks, and more. Win:8.0
trigger of <task definition>	*Plain*	<task trigger>	Returns the triggers associated with the specified task definition. Win:8.0
xml of <task definition>	*Plain*	<string>	Returns the XML-formatted definition of the task associated with the specified task definition. Win:8.0

Task Settings

These Inspectors examine the settings used by the Task Scheduler service to perform the task. These settings include dealing with starts, restarts and stops, how to manage hidden or multiple tasks, and more.

Creation Methods

Key Phrase	*Form*	Description
setting of <task definition>	*Plain*	Returns the settings associated with the given task. These settings include dealing with starts, restarts and stops, how to manage hidden or multiple tasks, and more. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
allow demand start of <task settings>	*Plain*	<boolean>	Returns TRUE if the task can be started by using the run command or the context menu. Win:8.0

Key Phrase	*Form*	Return Type	Description
allow hard terminate of <task settings>	*Plain*	<boolean>	Returns TRUE if the task can be terminated by using TerminateProcess. Win:8.0
at compatibility of <task settings>	*Plain*	<boolean>	Returns TRUE if the task scheduler is compatible with AT. Tasks compatible with AT are only allowed to have a single time trigger. Win:8.0
delete expired task after of <task settings>	*Plain*	<time interval>	Returns a time interval that the Task Scheduler will wait after expiration before deleting the task. Win:8.0
disallow start when on battery of <task settings>	*Plain*	<boolean>	Returns TRUE if the task will not be started when the computer is running on battery power. Win:8.0
enabled of <task settings>	*Plain*	<boolean>	Returns TRUE if the task is enabled. The task associated with the specified settings can be performed only when this setting is TRUE. Win:8.0
execution time limit of <task settings>	*Plain*	<time interval>	Returns a time interval corresponding to the amount of time allowed to complete the task. Win:8.0
hidden of <task settings>	*Plain*	<boolean>	Returns TRUE if the task will not be visible in the UI. Administrators can override this setting with a master switch that forces all tasks to be visible. Win:8.0
idle setting of <task settings>	*Plain*	<task idle settings>	Returns the Idle Settings of the specified task. These can include: • Idle duration • Restart On Idle • Stop On Idle End • Wait Timeout. Win:8.0
ignore new instance of <task settings>	*Plain*	<boolean>	Returns TRUE if the MultipleInstances task setting is TASK_INSTANCES_IGNORE_NEW. This instructs the scheduler not to start a new instance of the task if the task is already running. Win:8.0

Key Phrase	*Form*	Return Type	Description
network setting of <task settings>	*Plain*	<task network settings>	Returns a task network settings object containing a network profile ID and name. If this setting exists and the 'run only when network available' property is TRUE, the task will wait to execute until the specified network profile is available. Win:8.0
parallel instance of <task settings>	*Plain*	<boolean>	Returns TRUE if the MultipleInstances task setting is TASK_INSTANCES_PARALLEL. This instructs the scheduler to go ahead and start a new instance of the task while the task is already running. Win:8.0
priority of <task settings>	*Plain*	<integer>	Returns the numeric priority level of the associated task. Win:8.0
queue instance of <task settings>	*Plain*	<boolean>	Returns TRUE if the MultipleInstances task setting is TASK_INSTANCES_QUEUE. This instructs the scheduler to start a new instance of the task only after all existing instances are completed. Win:8.0
restart count of <task settings>	*Plain*	<integer>	Returns the number of times that the Task Scheduler will attempt to restart the associated task. Win:8.0
restart interval of <task settings>	*Plain*	<time interval>	Returns a time interval quantifying how much time the Task Scheduler will allocate to restarting the associated task. Win:8.0
run only when idle of <task settings>	*Plain*	<boolean>	Returns TRUE if the Task Scheduler is set to run the associated task only from an idle computer. Win:8.0
run only when network available of <task settings>	*Plain*	<boolean>	Returns TRUE if the Task Scheduler is set to run the task only when a network is available. If the 'network setting' exists, the task will be launched only when that network becomes available. Win:8.0
start when available of <task settings>	*Plain*	<boolean>	Returns TRUE if the Task Scheduler is allowed to start the task any time after its scheduled time has lapsed. Win:8.0

Key Phrase	*Form*	Return Type	Description
stop existing instance of <task settings>	*Plain*	<boolean>	Returns TRUE if the MultipleInstances task setting is TASK_INSTANCES_STOP_EXISTING. This instructs the scheduler to stop an existing instance of the task before starting a new instance. Win:8.0
stop when going on battery of <task settings>	*Plain*	<boolean>	Returns TRUE if the task should be stopped whenever the computer starts to run on battery power. Win:8.0
v1 compatibility of <task settings>	*Plain*	<boolean>	Returns TRUE if the specified task setting is compatible with version 1.0 of the scheduler. Win:8.0
v2 compatibility of <task settings>	*Plain*	<boolean>	Returns TRUE if the specified task setting is compatible with version 2.0 of the scheduler. Win:8.0
wake to run of <task settings>	*Plain*	<boolean>	Returns TRUE if the Task Scheduler should wake the computer whenever it's time to start the task. Win:8.0
xml of <task settings>	*Plain*	<string>	Returns a string containing an XML-formatted version of the specified task settings. Win:8.0

Priority Class

Threads are scheduled based on priority, which ranges from zero, the lowest, to 31, the highest. The zero-page thread, a system thread in charge of zeroing free pages when all other threads are finished, is the only thread allowed to have a priority of zero.

Creation Methods

Key Phrase	*Form*	Description
above normal priority	*PlainGlobal*	Returns a priority class object with 'above normal' priority. Threads with this status are between normal and high priority. Win:8.0

Key Phrase	*Form*	Description
base priority of <process>	*Plain*	Returns the base priority at which threads in the specified process execute. Corresponds to the 'Base Pri' column in the Task Manager. The return value is one of the following priority class values: • normal priority • high priority • idle priority • realtime priority • above normal priority • below normal priority. Win:8.0
below normal priority	*PlainGlobal*	Returns a priority class object with 'below normal' priority. This applies to threads that can wait for other threads to complete before running. Threads with this status are in between idle and normal priority. Win:8.0
high priority	*PlainGlobal*	Returns a priority class object with 'high' priority. These threads will steal processor time from other threads and should be used very carefully. Typically, they are used strictly for responding to time-critical events. Win:8.0
idle priority	*PlainGlobal*	Returns a priority class object with 'idle' priority. Processes that monitor the Client, such as screen savers, typically use this priority to keep them from interrupting higher priority threads. Win:8.0
normal priority	*PlainGlobal*	Returns a priority class object with 'normal' priority. This is the default priority class of a process. Win:8.0
realtime priority	*PlainGlobal*	Returns a priority class object with 'realtime' priority. These threads should be sparingly used, since they may interrupt user input and some disk operations. They are typically used for short, time-sensitive communication with low level hardware. Win:8.0

Operators

Key phrase	Return Type	Description
<priority class> = <priority class>	*<boolean>*	Returns TRUE if the two priority classes are the same. Win:8.0

Running Task

This group of Inspectors is built on top of the Windows Task Scheduler (see the MSDN reference). They only work with the 2.0 interface (Win 7, Vista and Server 2008). The scheduler allows tasks to be run according to various criteria. These Inspectors provide information such as the name and action(s) of each running task.

- These Inspectors are for the 2.0 interface only.

Creation Methods

Key Phrase	*Form*	Description
running task	*PlainGlobal*	Retrieves a list of all the currently running tasks, including hidden tasks. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
current action of <running task>	*Plain*	<string>	Returns the currently executing action of the specified running task. Win:8.0
disabled state of <running task>	*Plain*	<boolean>	Returns TRUE if the specified running task is disabled. Win:8.0
engine pid of <running task>	*Plain*	<integer>	Returns the process ID of the specified running task. Win:8.0
instance guid of <running task>	*Plain*	<string>	Returns the globally unique identifier corresponding to the specified running task. Win:8.0
name of <running task>	*Plain*	<string>	Returns the name of the specified running task. Win:8.0
path of <running task>	*Plain*	<string>	Returns the path associated with the specified running task. Win:8.0
queued state of <running task>	*Plain*	<boolean>	Returns TRUE if the specified running task is queued up. Win:8.0
ready state of <running task>	*Plain*	<boolean>	Returns TRUE if the specified running task is ready. Win:8.0

Key Phrase	*Form*	Return Type	Description
running state of <running task>	*Plain*	<boolean>	Returns TRUE if the specified running task is running. Win:8.0
unknown state of <running task>	*Plain*	<boolean>	Returns TRUE if the specified running task is unknown. Win:8.0

Examples

- `running tasks`
  - Returns a list of all currently running tasks, as determined by the Task Scheduler.

Task Named Value Pair

These Inspectors return name-value pairs associated with various aspects of a scheduled task.

Creation Methods

Key Phrase	*Form*	Description
header field of <email task action>	*Plain*	Returns the header information (as named value pairs) from the email triggered by a scheduled task. Win:8.0
value query of <event task trigger>	*Plain*	Returns a list of named XPath queries as name-value pairs. Each query in the list is applied to the last matching event XML returned from the subscription query specified in the Subscription property. The name of the query can be used as a variable in the message of a ShowMessage action. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
name of <task named value pair>	*Plain*	<string>	Returns the string name(s) associated with the specified name-value pair(s). Win:8.0
value of <task named value pair>	*Plain*	<string>	Returns the string value(s) associated with the specified name-value pair(s). Win:8.0

Scheduled Task

These Inspectors are built on top of the Windows Task Scheduler (see the MSDN reference). There are two versions: the 1.0 interface (Win 2000, XP & Server 2003) and the 2.0 interface (Win 7, Vista and Server 2008) which is favored when available. The Task Scheduler Inspector set reflects the 2.0 interface layout which in turn maps back to the 1.0 interface. Features of the 1.0 interface are available in the 2.0 interface, but not vice-versa. If the 2.0 interface isn't available, you may encounter undefined objects.

- Each of these Inspectors works with both 1.0 and 2.0 unless explicitly noted.

Creation Methods

Key Phrase	*Form*	Description
descendant of <task folder>	*Plain*	This Inspector creates a list of scheduled tasks from the specified folder, as well as all subfolders. • 2.0 interface only. Win:8.0
first interface scheduled task	*PlainGlobal*	Returns a list of all the scheduled tasks. • 1.0 interface only. Win:8.0
scheduled task	*PlainGlobal*	This Inspector is typically used to return a list of all scheduled tasks. It attempts to use the 2.0 interface first (which includes hidden tasks), then drops down to 1.0 if necessary. Win:8.0
scheduled task <string>	*NamedGlobal*	Returns a scheduled task with the specified name. It attempts to use the 2.0 interface if available, otherwise it uses the 1.0 interface. This method only looks in the root folder for 2.0 tasks, whch include hidden tasks. Win:8.0
scheduled task <string> of <task folder>	*Named*	Returns a scheduled task with the specified name in the root folder for 2.0 tasks, whch include hidden tasks. • 2.0 interface only. Win:8.0
scheduled task of <task folder>	*Plain*	This Inspector creates a list of scheduled tasks from the specified folder. • 2.0 interface only. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
definition of <scheduled task>	*Plain*	<task definition>	Returns the task definition for the specified scheduled task for both the 1.0 and 2.0 interface. Win:8.0
disabled state of <scheduled task>	*Plain*	<boolean>	Returns a boolean indicating the whether the specified task is disabled or not. Win:8.0
enabled of <scheduled task>	*Plain*	<boolean>	Returns a boolean TRUE if the specified scheduled task is enabled. Win:8.0
last run time of <scheduled task>	*Plain*	<time>	Returns the time corresponding to when the specified scheduled task was last run. Win:8.0
last task result of <scheduled task>	*Plain*	<integer>	Returns an integer corresponding to the last result of the specified scheduled task. Win:8.0
missed run count of <scheduled task>	*Plain*	<integer>	Returns an integer corresponding to the missed run count of the specified scheduled task. Win:8.0
name of <scheduled task>	*Plain*	<string>	Returns the name of the specified scheduled task. Win:8.0
next run time of <scheduled task>	*Plain*	<time>	Returns the next time that the specified task is scheduled to run. Win:8.0
path of <scheduled task>	*Plain*	<string>	Returns the path of the specified scheduled task. • 2.0 interface only. Win:8.0
queued state of <scheduled task>	*Plain*	<boolean>	Returns a boolean indicating the whether the specified task is queued or not. Win:8.0
ready state of <scheduled task>	*Plain*	<boolean>	Returns a boolean indicating the whether the specified task is ready or not. Win:8.0
running state of <scheduled task>	*Plain*	<boolean>	Returns a boolean indicating the whether the specified task is running or not. Win:8.0

Key Phrase	*Form*	Return Type	Description
security descriptor of <scheduled task>	*Plain*	<security descriptor>	Returns the security descriptor for the specified scheduled task. • 2.0 interface only. Win:8.0
trigger string of <scheduled task>	*Plain*	<string>	Returns the triggering string for the specified scheduled task. • 1.0 interface only. Win:8.0
unknown state of <scheduled task>	*Plain*	<boolean>	Returns TRUE if the state of the scheduled task is unknown. • 2.0 interface only. Win:8.0
xml of <scheduled task>	*Plain*	<string>	Returns a string containing the XML content of the specified scheduled task. • 2.0 interface only. Win:8.0

Examples

- `scheduled tasks of task folder "\"`

  Returns the 2.0 interface list of scheduled tasks from the specified folder.

Task Folder

These objects provide the methods that are used to retrieve tasks from the folder, as well as from its subfolders. Task folders are specified by name, path and include scheduled tasks.

Creation Methods

Key Phrase	*Form*	Description
task folder <string>	*NamedGlobal*	Gets the task folder with the given string name. Win:8.0
task folder of <task folder>	*Plain*	Returns the subfolders of the given task folder. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
descendant of <task folder>	*Plain*	<scheduled task>	Returns the descendant tasks, which include the tasks in the specified folder plus the tasks in all of its subfolders. Win:8.0
name of <task folder>	*Plain*	<string>	Returns the name(s) used to identify the folder that contains a task. Win:8.0
path of <task folder>	*Plain*	<string>	Returns the path(s) to the location of the task folder(s). Win:8.0
scheduled task <string> of <task folder>	*Named*	<scheduled task>	Returns the named scheduled task in the specified task folder. Win:8.0
scheduled task of <task folder>	*Plain*	<scheduled task>	Returns a list of the scheduled tasks of the specified task folder. Win:8.0
security descriptor of <task folder>	*Plain*	<security descriptor>	Returns the security descriptor for the specified task folder. Win:8.0
task folder of <task folder>	*Plain*	<task folder>	Returns the subfolders of the given task folder. Win:8.0

Task Principal

These Inspectors provide information about the scheduled task principal, which incapsulates the security credentials. The principal object includes a display name, a logon type, a run level and a set of IDs.

Creation Methods

Key Phrase	*Form*	Description
principal of <task definition>	*Plain*	Returns the principle for the specified task definition. A task principal is an object containing the associated security credentials. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
display name of <task principal>	*Plain*	<string>	Returns the name of the principal (as a string) that is displayed in the Task Scheduler UI. Win:8.0
group id of <task principal>	*Plain*	<string>	Returns the identifier of the user group (as a string) that is required to run the tasks associated with the principal. Win:8.0
group logon of <task principal>	*Plain*	<boolean>	Returns TRUE if the task logon type is set to group activation. In this case, the user ID speicifies the group. Win:8.0
highest runlevel of <task principal>	*Plain*	<boolean>	Returns TRUE if the specfied task principle will be run with the highest privileges. Win:8.0
id of <task principal>	*Plain*	<string>	Returns the identifier of the specified task principal as a string type. Win:8.0
interactive token logon of <task principal>	*Plain*	<boolean>	Returns TRUE if the task logon type is set to interactive token, meaning the task will only run in an existing interactive session. Win:8.0
interactive token password logon of <task principal>	*Plain*	<boolean>	Returns TRUE if the task logon type is set to interactive token or password. If the user is logged on, the interactive token is used. Otherwise the password is used. The password must have been specified when the task was registered. Win:8.0
lua runlevel of <task principal>	*Plain*	<boolean>	Returns TRUE if the specfied task principle will be run with the least privileged user account (LUA). Win:8.0
none logon of <task principal>	*Plain*	<boolean>	Returns TRUE if the logon method for the task principal is not specified. Used for non-NT credentials. Win:8.0
password logon of <task principal>	*Plain*	<boolean>	Returns TRUE if a password is used for logging on the user. The password must be supplied when the task is registered. Win:8.0

Key Phrase	*Form*	Return Type	Description
s4u logon of <task principal>	*Plain*	<boolean>	Returns TRUE if an existing interactive token is used to run the task. This requires the user to employ a service for user (S4U) logon. With S4U logons, no password is stored and neither network nor encrypted files can be accessed. Win:8.0
service account logon of <task principal>	*Plain*	<boolean>	Returns TRUE if a service account is used as a login. This implies that the task is being initiated by a Local System, Local Service, or Network Service account in a security context. Win:8.0
user id of <task principal>	*Plain*	<string>	Returns the user identifier (as a string) required to run the tasks associated with the principal. Win:8.0

Task Registration Info

These objects provide the administrative information used to describe a scheduled task. This information includes details such as a description, the name of the author, the date the task was registered, and the security descriptor.

Creation Methods

Key Phrase	*Form*	Description
registration info of <task definition>	*Plain*	Returns the task registration info object for the specified task definition. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
author of <task registration info>	*Plain*	<string>	Returns the author of the scheduled task included in the specified task registration information object. Win:8.0
date of <task registration info>	*Plain*	<time>	Returns the registration date (as a time type) of the scheduled task included in the specified task registration information object. Win:8.0

Key Phrase	*Form*	Return Type	Description
description of <task registration info>	*Plain*	<string>	Returns a string containing the description of the scheduled task included in the specified task registration information object. Win:8.0
documentation of <task registration info>	*Plain*	<string>	Returns a string containing any additional documentation pertaining to the scheduled task included in the specified task registration information object. Win:8.0
security descriptor of <task registration info>	*Plain*	<security descriptor>	Returns the security descriptor of the scheduled task referred to by the specified task registration information object. Win:8.0
source of <task registration info>	*Plain*	<string>	Returns a string containing the source (component, service application or user) of the scheduled task referred to by the specified task registration information object. Win:8.0
uri of <task registration info>	*Plain*	<string>	Returns a string containing the URI of the scheduled task referred to by the specified task registration information object. Win:8.0
version of <task registration info>	*Plain*	<string>	Returns a string containing the version number of the scheduled task included in the specified task registration information object. Win:8.0
xml of <task registration info>	*Plain*	<string>	Returns a string containing the XML-formatted version of the specified task registration information object. Win:8.0

Task Repetition Pattern

These Inspectors define how often a scheduled task should be run and determine how long the repetition pattern will be repeated after the task is initiated.

Creation Methods

Key Phrase	*Form*	Description
repetition of <task trigger>	*Plain*	Returns an object describing how often a scheduled task should be run and how long the repetition pattern should be repeated after the task is triggered. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
duration of <task repetition pattern>	*Plain*	<time interval>	Returns a time interval specifying how long the given task pattern is repeated. Win:8.0
interval of <task repetition pattern>	*Plain*	<time interval>	Returns a time interval corresponding to the time between each restart of the task. Win:8.0
stop at duration end of <task repetition pattern>	*Plain*	<boolean>	Returns TRUE if the task repetition pattern has reached the end of its run. Win:8.0

Task Action

These Inspectors grant access to the set of scheduled task actions. Actions have an ID and a type. The action types include starting COM handles, executing programs, sending emails and displaying messages.

Creation Methods

Key Phrase	*Form*	Description
action of <task definition>	*Plain*	Returns the task action(s) associated with the task definition. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
<task action> as com handler task action	*Cast*	<com handler task action>	Casts the specified task action as a com handler task action. Win:8.0
<task action> as email task action	*Cast*	<email task action>	Casts the specified task action as an email action. Win:8.0
<task action> as exec task action	*Cast*	<exec task action>	Casts the specified task action as an executable action. Win:8.0
<task action> as show message task action	*Cast*	<show message task action>	Casts the specified task action as a message display action. Win:8.0
id of <task action>	*Plain*	<string>	Returns the user-defined identifier for the specified action. Win:8.0
type of <task action>	*Plain*	<task action type>	Returns the type of the specified task action. This is one of the following types: • com handler: start a custom COM handler. • email: send an email. • exec: perform a command-line operation such as running a script, launching an executable, or, if the name of. Win:8.0

Task Action Type

These Inspectors return the Action Type(s) for a Windows Task. Action Types include Executables, Email messages, Handlers and Messages. For more information, see Task Actions at the MSDN site.

Creation Methods

Key Phrase	*Form*	Description
com handler task action type	*PlainGlobal*	Creates a 'task action type' corresponding to a task that activates a COM handler. Win:8.0
email task action type	*PlainGlobal*	Returns a 'task action type' corresponding to a task that sends an email. Win:8.0

Key Phrase	*Form*	Description
exec task action type	*PlainGlobal*	Returns a 'task action type' corresponding to a task that runs a program. Win:8.0
show message task action type	*PlainGlobal*	Returns a 'task action type' corresponding to a task that creates a message that will pop up on the desktop. Win:8.0
task action type <integer>	*NumberedGlobal*	Returns the task action type according to its enumerated value: • 0: TASK_ACTION_EXEC • 5: TASK_ACTION_COM_HANDLER • 6: TASK_ACTION_SEND_EMAIL • 7: TASK_ACTION_SHOW_MESSAGE. Win:8.0
type of <task action>	*Plain*	Returns the type of the specified task action. This is one of the following types: • com handler: fire a custom COM handler. • email: send an email. • exec: perform a command-line operation such as running a script, launching an executable, or, if the name of a document is provided, launching it with its associated application. • show message: display a message on the desktop. Win:8.0

Operators

Key phrase	Return Type	Description
<task action type> = <task action type>	*<boolean>*	Returns TRUE if the two task action types are equal. Win:8.0

Examples

```
(it as Com Handler Task Action) of Actions whose (Type of it = Com Handler Task Action Type) of Definition of Scheduled Task "Test"
```

Returns a list of any 'com handler' actions in the scheduled task named 'Test'.

Email Task Action

These Inspectors deal with Windows Email Actions, which are specified by tasks triggered by various events like computer state changes or scheduled times. The Email Action causes an email to be sent. For more information, see Email Action at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task action> as email task action	*Cast*	Casts the specified task action as an email action. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
attachment of <email task action>	*Plain*	<string>	Returns the array of attachments (as strings) to be sent with the email referenced by the specified email task action. Win:8.0
bcc of <email task action>	*Plain*	<string>	Returns the email address(es) that you want to Bcc in the email referenced by the specified email task action. Win:8.0
body of <email task action>	*Plain*	<string>	Returns the body of the email message referenced by the specified email task action. Win:8.0
cc of <email task action>	*Plain*	<string>	Returns the email address(es) that you want to Bcc in the email referenced by the specified email task action. Win:8.0
from of <email task action>	*Plain*	<string>	Returns the 'from' email address for the email referenced by the specified email task action. Win:8.0
header field of <email task action>	*Plain*	<task named value pair>	Returns the header information (as named value pairs) from the email triggered by a scheduled task. Win:8.0
replyto of <email task action>	*Plain*	<string>	Returns the email reply address(es) for the email referenced by the specified email task action. Win:8.0

Key Phrase	*Form*	Return Type	Description
server of <email task action>	*Plain*	<string>	Returns the name of the server used to send the email referenced by the specified email task action. Win:8.0
subject of <email task action>	*Plain*	<string>	Returns the subject line for the email referenced by the specified email task action. Win:8.0
to of <email task action>	*Plain*	<string>	Returns the address(es) for the 'to' line of the email referenced by the specified email task action. Win:8.0

Exec Task Action

These Inspectors deal with Windows Exec Actions, which are specified by tasks triggered by various events like computer state changes or scheduled times. The Exec Action causes a program to run. For more information, see Exec Action at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task action> as exec task action	*Cast*	Casts the specified task action as an executable action. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
argument string of <exec task action>	*Plain*	<string>	Returns the arguments associated with the command-line operation referenced by the exec task action. Win:8.0
path of <exec task action>	*Plain*	<string>	Returns the path to the executable file specified in the task action. Win:8.0
working directory of <exec task action>	*Plain*	<string>	Returns the the directory containing either the executable file or files used by the executable specified in the task action. Win:8.0

Com Handler Task Action

These Inspectors deal with Windows COM Handler Actions, which are specified by tasks triggered by various events like computer state changes or scheduled times. The COM Handler Action causes a handler to be fired. For more information, see COM Handler Action at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task action> as com handler task action	*Cast*	Casts the specified task action as a com handler task action. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
class id of <com handler task action>	*Plain*	<string>	Returns a string containing the class ID of the specified 'com handler task action', corresponding to a task that activates a COM handler. Win:8.0
data of <com handler task action>	*Plain*	<string>	Returns a string containing the data of the specified 'com handler task action', corresponding to a task that activates a COM handler. Win:8.0

Show Message Task Action

These Inspectors deal with Windows Show Message Actions, which are specified by tasks triggered by various events like computer state changes or scheduled times. The Show Message Action causes a a message box to be displayed. For more information, see Show Message Action at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task action> as show message task action	*Cast*	Casts the specified task action as a message display action. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
message body of <show message task action>	*Plain*	<string>	Returns the message text (as a string) associated with the specified task action. This is the text displayed in the message box. Win:8.0
title of <show message task action>	*Plain*	<string>	Returns the title text (as a string) associated with the specified task action. This is the title displayed at the top of the message box. Win:8.0

Task Idle Settings

These objects specify how the Task Scheduler performs tasks when the computer is in an idle condition. These settings concern the duration, restart, stop and wait conditions for the specified idle settings. For more information about idle conditions, see 'Task Idle Conditions' at MSDN.

Creation Methods

Key Phrase	*Form*	Description
idle setting of <task settings>	*Plain*	Returns the Idle Settings of the specified task. These can include: • Idle duration • Restart On Idle • Stop On Idle End • Wait Timeout. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
idle duration of <task idle settings>	*Plain*	<time interval>	Returns a value that indicates the amount of time that the computer must be in an idle state before the task is run. Win:8.0
restart on idle of <task idle settings>	*Plain*	<boolean>	Returns a Boolean value that indicates whether the task is restarted when the computer cycles into an idle condition more than once. Win:8.0
stop on idle end of <task idle settings>	*Plain*	<boolean>	Returns a Boolean value that indicates that the Task Scheduler will terminate the task if the idle condition ends before the task is completed. Win:8.0
wait timeout of <task idle settings>	*Plain*	<time interval>	Returns a time interval that represents the amount of time that the Task Scheduler will wait for an idle condition to occur. Win:8.0

Task Network Settings

These Inspectors provide the settings used by the Task Scheduler to obtain a network profile. A network settings object has an ID and a name.

Creation Methods

Key Phrase	*Form*	Description
network setting of <task settings>	*Plain*	Returns a task network settings object containing a network profile ID and name. If this setting exists and the 'run only when network available' property is TRUE, the task will wait to execute until the specified network profile is available. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
id of <task network settings>	*Plain*	<string>	Returns a GUID that identifies a network profile. Win:8.0
name of <task network settings>	*Plain*	<string>	Returns the name of a network profile. The name is used for display purposes. Win:8.0

Task Trigger

Task triggers for the Task Scheduler have properties that identify and modify the action of each trigger.

Creation Methods

Key Phrase	*Form*	Description
trigger of <task definition>	*Plain*	Returns the trigger associated with the specfied task definition. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
<task trigger> as boot task trigger	*Cast*	<boot task trigger>	Casts the given task trigger as a boot trigger type. Win:8.0

Key Phrase	*Form*	Return Type	Description
<task trigger> as daily task trigger	*Cast*	<daily task trigger>	Casts the given task trigger as a daily trigger type. Win:8.0
<task trigger> as event task trigger	*Cast*	<event task trigger>	Casts the given task trigger as an event trigger type. Win:8.0
<task trigger> as idle task trigger	*Cast*	<idle task trigger>	Casts the given task trigger as an idle trigger type. Win:8.0
<task trigger> as logon task trigger	*Cast*	<logon task trigger>	Casts the given task trigger as a logon trigger type. Win:8.0
<task trigger> as monthly task trigger	*Cast*	<monthly task trigger>	Casts the given task trigger as a monthly trigger type. Win:8.0
<task trigger> as monthlydow task trigger	*Cast*	<monthlydow task trigger>	Casts the given task trigger as a monthly DOW trigger type. Win:8.0
<task trigger> as registration task trigger	*Cast*	<registration task trigger>	Casts the given task trigger as a registration trigger type. Win:8.0
<task trigger> as session state change task trigger	*Cast*	<session state change task trigger>	Casts the given task trigger as a session trigger type. Win:8.0
<task trigger> as time task trigger	*Cast*	<time task trigger>	Casts the given task trigger as a time trigger type. Win:8.0
<task trigger> as weekly task trigger	*Cast*	<weekly task trigger>	Casts the given task trigger as a weekly trigger type. Win:8.0
enabled of <task trigger>	*Plain*	<boolean>	Returns TRUE if the specified trigger is enabled. Win:8.0
end boundary of <task trigger>	*Plain*	<time>	Returns a time corresponding to when the specified trigger is deactivated, after which it will no longer run. Win:8.0
execution time limit of <task trigger>	*Plain*	<time interval>	Returns a time interval corresponding to the maximum time allotted to the task. Win:8.0
id of <task trigger>	*Plain*	<string>	Returns the identifier for the trigger (as a string). Win:8.0

Key Phrase	*Form*	Return Type	Description
repetition of <task trigger>	*Plain*	<task repetition pattern>	Returns a task repetition pattern corresponding to how often the task will run and how long the repetition pattern will be repeated after the task has been started. Win:8.0
start boundary of <task trigger>	*Plain*	<time>	Returns the date and time when the trigger is activated, after which the task is launched. Win:8.0
type of <task trigger>	*Plain*	<task trigger type>	Returns the task trigger type corresponding to the specified task trigger. Win:8.0

Task Trigger Type

These objects represent the possible types of triggers used by the Task Scheduler to fire off a task. Tasks can be triggered by dates, user actions and system events.

Creation Methods

Key Phrase	*Form*	Description
boot task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a boot trigger, which starts a task when the system is booted up. Win:8.0
daily task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a daily trigger, which starts a task on a daily schedule. Win:8.0
event task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to an event trigger, which starts a task when a system event occurs. Win:8.0
idle task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a idle trigger, which starts a task when the system is enters the idle state. Win:8.0
logon task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a logon trigger, which starts a task when a user logs on. When the scheduler starts, all the logged-on users are noted, and any logon triggers that match the users will launch their associated task. Win:8.0

Key Phrase	*Form*	Description
monthly task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a Monthly trigger, which starts a task when the associated day of the week occurs, such as the third of each month from July through December. Win:8.0
monthlydow task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a MonthlyDOW trigger, which starts a task when the associated day of the week occurs, such as the second Monday from January through June. Win:8.0
registration task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a registration trigger, which starts a task whenever it gets registered or updated. Win:8.0
session state change task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a session state change trigger, which starts a task on console connects or disconnects, remote connects or disconnects, or workstation lock or unlock notifications. Win:8.0
task trigger type <integer>	*NumberedGlobal*	Creates a task trigger type corresponding to the given integer. Win:8.0
time task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a time trigger, which starts a task on a specific date and time. Win:8.0
type of <task trigger>	*Plain*	Returns the type of the specified task trigger. Win:8.0
weekly task trigger type	*PlainGlobal*	Creates a task trigger type corresponding to a weekly trigger, which starts a task on a weekly schedule, such as 9:00 AM each Wednesday. Win:8.0

Operators

Key phrase	Return Type	Description
<task trigger type> = <task trigger type>	*<boolean>*	Returns TRUE if the two task trigger types are equal. Win:8.0

Time Task Trigger

Time triggers are used by the Task Scheduler to launch a task at a specific date and time.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as time task trigger	*Cast*	Casts the given task trigger as a time trigger type. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
random delay of <time task trigger>	*Plain*	<time interval>	Returns a delay time that is randomly added to the start time of the specified time task trigger. Win:8.0

Weekly Task Trigger

Weekly triggers are used by the Task Scheduler to launch a task at a specific time of day on a weekly schedule, such as 9:00 PM every other Friday.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as weekly task trigger	*Cast*	Casts the given task trigger as a weekly trigger type. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
days run of <weekly task trigger>	*Plain*	<day of week>	Returns a day of week corresponding to the days that a task will run given the specified weekly task trigger. Win:8.0
random delay of <weekly task trigger>	*Plain*	<time interval>	Returns a delay time that is randomly added to the start time of the specified weekly task trigger. Win:8.0
weeks interval of <weekly task trigger>	*Plain*	<time interval>	Returns the interval between the weeks in the schedule provided by the specified weekly task trigger. Win:8.0

Daily Task Trigger

These Inspectors deal with tasks that are triggered on a daily basis. For example, a task might start at 9:00 AM every day or every other day. More information about this subset of task triggers can be found by searching for Daily Trigger at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as daily task trigger	*Cast*	Casts the given task trigger as a daily trigger type. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
days interval of <daily task trigger>	*Plain*	<time interval>	Returns the interval between the days in the schedule referred to by the specified daily task trigger. Win:8.0
random delay of <daily task trigger>	*Plain*	<time interval>	Returns a random time interval to be added to the start time of the trigger referred to by the specified daily task trigger. Win:8.0

Monthly Task Trigger

These Inspectors deal with tasks that are triggered on a monthly basis. For example, a task might start at 9:00 AM on specific days of specific months. More information about this subset of task triggers can be found by searching for Monthly Trigger at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as monthly task trigger	*Cast*	Casts a given task trigger as a monthly task trigger. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
days run of <monthly task trigger>	*Plain*	<day of month>	Returns the days of the month during which the task will run. Win:8.0
months run of <monthly task trigger>	*Plain*	<month>	Returns the months of the year during which the task will run. Win:8.0
random delay of <monthly task trigger>	*Plain*	<time interval>	Returns the upper limit of a random time delay that will be added to the trigger start time. The format is P<days>DT<hours>H<minutes>M<seconds>S. For example, P3DT6H represents a 3 day, 6 hour maximum, and the random time will be less than or equal to that value. Win:8.0
run on last day in month of <monthly task trigger>	*Plain*	<boolean>	Returns TRUE if the specified monthly trigger is set to run on the last day of the month. Win:8.0

Monthlydow Task Trigger

These Inspectors deal with tasks that are triggered on a repeating day-of-the-week basis. For example, the task might start at 9:00 AM on specific days of the week, weeks of the month, or months of the year. More information about this subset of task triggers can be found by searching for MonthlyDOW Trigger at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as monthlydow task trigger	*Cast*	Casts the specified task trigger as a monthly day of week trigger. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
days run of <monthlydow task trigger>	*Plain*	<day of week>	Returns the days of the month during which the monthly day-of-week task will run. Win:8.0
months run of <monthlydow task trigger>	*Plain*	<month>	Returns the months of the year during which the monthly day-of-week task will run. Win:8.0
random delay of <monthlydow task trigger>	*Plain*	<time interval>	Returns the upper limit of a random time delay that will be added to the monthly day-of-week trigger start time. The format is P<days>DT<hours>H<minutes>M<seconds>S. For example, P3DT6H represents a 3 day, 6 hour maximum, and the random time will be less than or equal to that value. Win:8.0
run on fifth week in month of <monthlydow task trigger>	*Plain*	<boolean>	Returns TRUE if the monthly day-of-week trigger will run the task in the fifth week of the month. Win:8.0
run on first week in month of <monthlydow task trigger>	*Plain*	<boolean>	Returns TRUE if the monthly day-of-week trigger will run the task in the first week of the month. Win:8.0
run on fourth week in month of <monthlydow task trigger>	*Plain*	<boolean>	Returns TRUE if the monthly day-of-week trigger will run the task in the fourth week of the month. Win:8.0

Key Phrase	*Form*	Return Type	Description
run on last week in month of <monthlydow task trigger>	*Plain*	<boolean>	Returns TRUE if the monthly day-of-week trigger will run the task in the last week of the month. Win:8.0
run on second week in month of <monthlydow task trigger>	*Plain*	<boolean>	Returns TRUE if the monthly day-of-week trigger will run the task in the second week of the month. Win:8.0
run on third week in month of <monthlydow task trigger>	*Plain*	<boolean>	Returns TRUE if the monthly day-of-week trigger will run the task in the third week of the month. Win:8.0

Session State Change Task Trigger

These Inspectors return information about Windows actions that are triggered by changes in Terminal Server session states, such as switching users or logging in remotely.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as session state change task trigger	*Cast*	Casts the given task trigger as a session trigger type. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
console connect of <session state change task trigger>	*Plain*	<boolean>	Returns a boolean TRUE if the terminal server console has experienced a connection state change, for instance when users are switched on the client computer. Win:8.0
console disconnect of <session state change task trigger>	*Plain*	<boolean>	Returns a boolean TRUE if the terminal server console has experienced a disconnect state change, for instance when users are switched on the client computer. Win:8.0

Key Phrase	*Form*	Return Type	Description
delay of <session state change task trigger>	*Plain*	<time interval>	Returns a time interval corresponding to the delay between the detection of the specified session state change and the time the task is started. Win:8.0
remote connect of <session state change task trigger>	*Plain*	<boolean>	Returns a boolean TRUE if the terminal server console has experienced a remote connection change, for instance when a user connects to a session using a remote desktop connection. Win:8.0
remote disconnect of <session state change task trigger>	*Plain*	<boolean>	Returns a boolean TRUE if the terminal server console has experienced a remote disconnect change, for instance when a user disconnects from a session using a remote desktop connection. Win:8.0
session lock of <session state change task trigger>	*Plain*	<boolean>	Returns a boolean TRUE if the terminal server console has experienced a session lock change, for instance when a computer is locked. Win:8.0
session unlock of <session state change task trigger>	*Plain*	<boolean>	Returns a boolean TRUE if the terminal server console has experienced a session unlock change, for instance when a computer is unlocked. Win:8.0
user id of <session state change task trigger>	*Plain*	<string>	Returns the user identifier (as a string) required to run the tasks associated with the specified session state-change trigger. Win:8.0

Registration Task Trigger

These Inspectors deal with tasks that are triggered whenever the task is registered or updated. More information about this subset of task triggers can be found by searching for Registration triggers at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as registration task trigger	*Cast*	Casts the given task trigger as a registration trigger type. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
delay of <registration task trigger>	*Plain*	<time interval>	This Inspector returns the amount of time between when the task is registered and when the task is started. . Win:8.0

Idle Task Trigger

These Inspectors deal with tasks that are triggered when the system enters an idle state. More information about this subset of task triggers can be found by searching for Idle Trigger at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as idle task trigger	*Cast*	Casts the given task trigger as an idle trigger type. Win:8.0

Event Task Trigger

These Inspectors deal with tasks that are triggered by a specific event, such as a system start, logon or idle. More information about event task triggers can be found by searching for Task Triggers at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as event task trigger	*Cast*	Casts the given task trigger as an event trigger type. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
delay of <event task trigger>	*Plain*	<time interval>	Returns a value indicating the amount of time lapsed between the event trigger and the start of the task. Win:8.0
subscription of <event task trigger>	*Plain*	<string>	Returns the XPath query string identifying the trigger event. Win:8.0
value query of <event task trigger>	*Plain*	<task named value pair>	Returns a list of named XPath queries as name-value pairs. Each query in the list is applied to the last matching event XML returned from the subscription query specified in the Subscription property. The name of the query can be used as a variable in the message of a ShowMessage action. Win:8.0

Logon Task Trigger

These Inspectors deal with tasks that are triggered when a user logs on to the computer. More information about this subset of task triggers can be found by searching for Logon Trigger at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as logon task trigger	*Cast*	Casts the given task trigger as a logon trigger type. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
delay of <logon task trigger>	*Plain*	<time interval>	A task that is scheduled to run when the user is logged on can be delayed by 30 seconds or as long as a day. This Inspector returns the delay for the specified logon task trigger as a time interval. Win:8.0

Key Phrase	*Form*	Return Type	Description
user id of <logon task trigger>	*Plain*	<string>	Returns the user identifier (as a string) required to run the tasks associated with the specified logon trigger. Win:8.0

Boot Task Trigger

These Inspectors deal with tasks that are triggered when the system is booted up. More information about this subset of task triggers can be found by searching for Boot Trigger at the MSDN site.

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

Creation Methods

Key Phrase	*Form*	Description
<task trigger> as boot task trigger	*Cast*	Casts the given task trigger as a boot trigger type. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
delay of <boot task trigger>	*Plain*	<time interval>	Returns a delay time interval to add to the start time of the trigger referred to by the specified boot task trigger. Win:8.0

Power Inspectors

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

Power Level

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

Creation Methods

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

Properties

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

Examples

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

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

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

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
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

Key Phrase	*Form*	Description
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
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

Key Phrase	*Form*	Return Type	Description
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
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 (and more. 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
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 (and more. 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

Key Phrase	*Form*	Description
user interval of <activity history>	*Plain*	Returns a power interval (containing a time range and a power state, such as standby or idle) corresponding to the specified user activity history. Win:8.0

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

Key Phrase	*Form*	Description
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
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

Microsoft IIS Metabase Objects

The Microsoft IIS Metabase is a repository for most IIS configuration values. The following inspectors retrieve information about the Microsoft IIS Metabase.

Metabase

The IIS metabase is similar in structure to the Windows Registry, providing hierarchal storage of IIS configuration properties for Web sites, virtual directories, FTP, and more.

Creation Methods

Key Phrase	*Form*	Description
metabase	*PlainGlobal*	Returns the IIS metabase object. Win

Properties

Key Phrase	*Form*	Return Type	Description
key <string> of <metabase>	*Named*	<metabase key>	Returns the named key of the metabase. Win
key of <metabase>	*Plain*	<metabase key>	Returns the root key of the IIS metabase. Win

Metabase Identifier

A metabase identifier is a unique numerical identifier which IIS recognizes internally.

Creation Methods

Key Phrase	*Form*	Description
identifier of <metabase value>	*Plain*	Returns the metabase identifier of the specified value. Win

Properties

Key Phrase	*Form*	Return Type	Description

Key Phrase	*Form*	Return Type	Description
<metabase identifier> as integer	*Cast*	<integer>	The integer value of the specified identifier. Use the Microsoft MetaEdit utility to find the integer value of an identifier. Win

Operators

Key phrase	Return Type	Description
<metabase identifier> = <metabase identifier>	*<boolean>*	Compares two metabase identifiers for equality. Win

Metabase Key

Like a registry key, a metabase key contains named properties about the IIS metabase.

Creation Methods

Key Phrase	*Form*	Description
key <string> of <metabase key>	*Named*	Returns the named subkey of the specified metabase key. Win
key <string> of <metabase>	*Named*	Returns the named key of the metabase. Win
key of <metabase key>	*Plain*	Iterates the subkeys of the specified metabase key. Win
key of <metabase>	*Plain*	Returns the root key of the IIS metabase. Win

Properties

Key Phrase	*Form*	Return Type	Description
key <string> of <metabase key>	*Named*	<metabase key>	Returns the named subkey of the specified metabase key. Win
key of <metabase key>	*Plain*	<metabase key>	Iterates the subkeys of the specified metabase key. Win

Key Phrase	*Form*	Return Type	Description
name of <metabase key>	*Plain*	<string>	Returns the name of the specified metabase key. Win
value of <metabase key>	*Plain*	<metabase value>	Returns the value of the specified metabase key. Win

Examples

- `names of keys of metabase`

Depending on the metabase configuration, returns the names of the metabase keys, such as:

- LM
- Schema.

- `names of keys of key "/LM" of metabase`

Depending on the metabase configuration, returns the key names in the metabase "/LM" key, such as IISADMIN, W3SVC or MimeMap.

Metabase Type

The type identifier of the data associated with a metabase key value.

Creation Methods

Key Phrase	*Form*	Description
type of <metabase value>	*Plain*	Returns the type of the specified metabase value. Win

Properties

Key Phrase	*Form*	Return Type	Description
<metabase type> as integer	*Cast*	<integer>	Meanings same as registry types: • 1 = DWORD • 2 = STRING • 3 = BINARY • 4 = EXPANDSZ • 5 = MULTISZ. Win

Key Phrase	*Form*	Return Type	Description
<metabase type> as string	*Cast*	<string>	Returns metabase types as strings: • "DWord" • "String" • "Binary" • "ExpandSz" • "MultiSz". Win

Operators

Key phrase	Return Type	Description
<metabase type> = <metabase type>	*<boolean>*	Compares two metabase types for equality. Win

Metabase User Type

The user type is a DWORD that specifies how the property value is used. User types enable IIS to classify properties by application.

Creation Methods

Key Phrase	*Form*	Description
user type of <metabase value>	*Plain*	Returns the user type of the specified metabase value. Win

Properties

Key Phrase	*Form*	Return Type	Description
<metabase user type> as integer	*Cast*	<integer>	Returns the metabase user type as an integer: • 2 = Server • 2 = File • 100 = WAM • 200 = ASP App. Win

Key Phrase	*Form*	Return Type	Description
<metabase user type> as string	*Cast*	<string>	Returns the metabase user type as a string: • "Server" • "File" • "WAM" • "ASP App" • "(Other)". Win

Operators

Key phrase	Return Type	Description
<metabase user type> = <metabase user type>	*<boolean>*	Compares two metabase user types for equality. Win

Metabase Value

This Inspector is used to access values stored in an IIS metabase key. The type of the data stored in the value determines what casting operations are allowed. There are several casting Inspectors that you can use to extract values from the registry.

Creation Methods

Key Phrase	*Form*	Description
value of <metabase key>	*Plain*	Returns the value of the specified metabase key. Win

Properties

Key Phrase	*Form*	Return Type	Description
<metabase value> as integer	*Cast*	<integer>	Returns the integer value of the metabase value. Win
<metabase value> as string	*Cast*	<string>	Returns the string value of the metabase value. Win
identifier of <metabase value>	*Plain*	<metabase identifier>	Returns the metabase identifier of the specified value. Win

Key Phrase	*Form*	Return Type	Description
inherit attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the inherit attribute set. Win
insert path attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the insert path attribute set. Win
reference attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the reference attribute set. Win
secure attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the secure attribute set. Win
type of <metabase value>	*Plain*	<metabase type>	Returns the type of the specified metabase value. Win
user type of <metabase value>	*Plain*	<metabase user type>	Returns the user type of the specified metabase value. Win
volatile attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the volatile attribute set. Win

Examples

- `inherit attributes of values of key "/Schema" of metabase`

  Returns a boolean True or False depending on the inherit attributes of each sub-key in the specified key of the metabase.

- `volatile attributes of values of key "/LM" of metabase`

  Returns a boolean True or False depending on the volatile attributes of each sub-key in the specified key of the metabase.

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
above normal priority	above normal priorities	<priority class>	<world>	*PlainGlobal*	win
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
access mode of <access control entry>	access modes	<integer>	<access control entry>	*Plain*	win
accessed time of <filesystem object>	accessed times	<time>	<filesystem object>	*Plain*	win
account disabled flag of <user>	account disabled flags	<boolean>	<user>	*Plain*	win
account expiration of <user>	account expirations	<time>	<user>	*Plain*	win
account lockout duration of <security database>	account lockout durations	<time interval>	<security database>	*Plain*	win
account lockout observation window of <security database>	account lockout observation windows	<time interval>	<security database>	*Plain*	win
account lockout threshold of <security	account lockout	<integer>	<security	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
database>	thresholds		database>		
account logon category of <audit policy>	account logon categories	<audit policy category>	<audit policy>	*Plain*	win
account management category of <audit policy>	account management categories	<audit policy category>	<audit policy>	*Plain*	win
account name of <security identifier>	account names	<string>	<security identifier>	*Plain*	win
account with privilege <string>	accounts with privilege	<security account>	<world>	*NamedGlobal*	win
account with privileges	accounts with privileges	<security account>	<world>	*PlainGlobal*	win
accounts operator flag of <user>	accounts operator flags	<boolean>	<user>	*Plain*	win
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*	win
action <integer>	actions	<action>	<world>	*NumberedGlobal*	win
action lock state	action lock states	<action lock state>	<world>	*PlainGlobal*	win
action of <firewall rule>	actions	<firewall action>	<firewall rule>	*Plain*	win
action of <task definition>	actions	<task action>	<task definition>	*Plain*	win
active action	active actions	<action>	<world>	*PlainGlobal*	win
active device	active devices	<active device>	<world>	*PlainGlobal*	win
active device file	active device	<file>	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
	files				
active device file <string>	active device files	<file>	<world>	*NamedGlobal*	win
active directory user of <user>	active directory users	<active directory local user>	<user>	*Plain*	win
active of <action>	actives	<boolean>	<action>	*Plain*	win
active of <logged on user>	actives	<boolean>	<logged on user>	*Plain*	win
active start time of <action>	active start times	<time>	<action>	*Plain*	win
active state	active states	<power state>	<world>	*PlainGlobal*	win
activity history of <logged on user>	activity histories	<activity history>	<logged on user>	*Plain*	win
adapter of <network adapter interface>	adapters	<network adapter>	<network adapter interface>	*Plain*	win
adapter of <network>	adapters	<network adapter>	<network>	*Plain*	win
address <string> of <html>	addresss	<html>	<html>	*Named*	core
address <string> of <string>	addresss	<html>	<string>	*Named*	core
address list of <network adapter>	address lists	<network address list>	<network adapter>	*Plain*	win
address of <html>	addresss	<html>	<html>	*Plain*	core
address of <network adapter interface>	addresses	<ipv4or6 address>	<network adapter interface>	*Plain*	win
address of <network adapter>	addresses	<ipv4 address>	<network adapter>	*Plain*	win
address of <network address list>	addresses	<ipv4 address>	<network address list>	*Plain*	win
address of <network ip interface>	addresses	<ipv4 address>	<network ip interface>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
address of <string>	addresss	<html>	<string>	*Plain*	core
admin privilege of <user>	admin privileges	<boolean>	<user>	*Plain*	win
administrator <string> of <client>	administrators	<setting>	<client>	*Named*	win
administrator of <client>	administrators	<setting>	<client>	*Plain*	win
all firewall scope	all firewall scopes	<firewall scope>	<world>	*PlainGlobal*	win
all running service	all running services	<service>	<world>	*PlainGlobal*	win
all service	all services	<service>	<world>	*PlainGlobal*	win
allow demand start of <task settings>	allow demand starts	<boolean>	<task settings>	*Plain*	win
allow firewall action	allow firewall actions	<firewall action>	<world>	*PlainGlobal*	win
allow hard terminate of <task settings>	allow hard terminates	<boolean>	<task settings>	*Plain*	win
allow inbound echo request of <firewall icmp settings>	allow inbound echo requests	<boolean>	<firewall icmp settings>	*Plain*	win
allow inbound mask request of <firewall icmp settings>	allow inbound mask requests	<boolean>	<firewall icmp settings>	*Plain*	win
allow inbound router request of <firewall icmp settings>	allow inbound router requests	<boolean>	<firewall icmp settings>	*Plain*	win
allow inbound timestamp request of <firewall icmp settings>	allow inbound timestamp requests	<boolean>	<firewall icmp settings>	*Plain*	win
allow outbound destination unreachable of <firewall icmp settings>	allow outbound destination unreachables	<boolean>	<firewall icmp settings>	*Plain*	win
allow outbound packet too big of <firewall icmp	allow outbound packet too bigs	<boolean>	<firewall icmp settings>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
settings>					
allow outbound parameter problem of <firewall icmp settings>	allow outbound parameter problems	<boolean>	<firewall icmp settings>	*Plain*	win
allow outbound source quench of <firewall icmp settings>	allow outbound source quenches	<boolean>	<firewall icmp settings>	*Plain*	win
allow outbound time exceeded of <firewall icmp settings>	allow outbound time exceededs	<boolean>	<firewall icmp settings>	*Plain*	win
allow redirect of <firewall icmp settings>	allow redirects	<boolean>	<firewall icmp settings>	*Plain*	win
allow unmentioned site of <license>	allow unmentioned sites	<boolean>	<license>	*Plain*	win
allowed workstations string of <user>	allowed workstations strings	<string>	<user>	*Plain*	win
ancestor of <filesystem object>	ancestors	<folder>	<filesystem object>	*Plain*	win
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
anonymous logon group	anonymous logon groups	<security account>	<world>	*PlainGlobal*	win
ansi code page	ansi code pages	<integer>	<world>	*PlainGlobal*	win
any adapter of <network>	any adapters	<network adapter>	<network>	*Plain*	win
any ip version	any ip versions	<ip version>	<world>	*PlainGlobal*	core
apparent registration server time	apparent registration	<time>	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
	server times				
append permission of <access control entry>	append permissions	<boolean>	<access control entry>	*Plain*	win
application <string>	applications	<application>	<world>	*NamedGlobal*	win
application <string> of <folder>	applications	<application>	<folder>	*Named*	win
application <string> of <registry key>	applications	<application>	<registry key>	*Named*	win
application <string> of <registry>	applications	<application>	<registry>	*Named*	win
application event log	application event logs	<event log>	<world>	*PlainGlobal*	win
application folder <string> of <registry key>	application folders	<folder>	<registry key>	*Named*	win
application folder <string> of <registry>	application folders	<folder>	<registry>	*Named*	win
application folder of <registry key>	application folders	<folder>	<registry key>	*Plain*	win
application name of <firewall rule>	application names	<string>	<firewall rule>	*Plain*	win
application of <registry key>	applications	<application>	<registry key>	*Plain*	win
application of <registry>	applications	<application>	<registry>	*Plain*	win
application parameter string of <user>	application parameter strings	<string>	<user>	*Plain*	win
application usage summary	application usage summaries	<application usage summary>	<world>	*PlainGlobal*	win
application usage summary <string>	application usage summaries	<application usage summary>	<world>	*NamedGlobal*	win
april	aprils	<month>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
archive of <filesystem object>	archives	<boolean>	<filesystem object>	*Plain*	win
argument string of <exec task action>	argument strings	<string>	<exec task action>	*Plain*	win
argument string of <file shortcut>	argument strings	<string>	<file shortcut>	*Plain*	win
at compatibility of <task settings>	at compatibilities	<boolean>	<task settings>	*Plain*	win
attachment of <email task action>	attachments	<string>	<email task action>	*Plain*	win
attribute <integer> of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Numbered*	core
attribute <string> of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Named*	core
attribute of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Plain*	core
attribute permission of <network share>	attribute permissions	<boolean>	<network share>	*Plain*	win
audit failure event log event type	audit failure event log event types	<event log event type>	<world>	*PlainGlobal*	win
audit failure of <audit policy information>	audit failures	<boolean>	<audit policy information>	*Plain*	win
audit level of <local mssql database>	audit levels	<integer>	<local mssql database>	*Plain*	win
audit policy	audit policies	<audit policy>	<world>	*PlainGlobal*	win
audit success event log event type	audit success event log event types	<event log event type>	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
audit success of <audit policy information>	audit successes	<boolean>	<audit policy information>	*Plain*	win
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
authenticated users group	authenticated users groups	<security account>	<world>	*PlainGlobal*	win
author of <task registration info>	authors	<string>	<task registration info>	*Plain*	win
authorized application of <firewall profile>	authorized applications	<firewall authorized application>	<firewall profile>	*Plain*	win
average of <evaluation cycle>	averages	<integer>	<evaluation cycle>	*Plain*	win
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
backoffice bit <operating system suite mask>	backoffice bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
bad password count of <user>	bad password counts	<integer>	<user>	*Plain*	win
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
base of <string>	bases	<html>	<string>	*Plain*	core
base priority of <process>	base priorities	<priority class>	<process>	*Plain*	win
batch group	batch groups	<security account>	<world>	*PlainGlobal*	win
bcc of <email task action>	bccs	<string>	<email task action>	*Plain*	win
below normal priority	below normal priorities	<priority class>	<world>	*PlainGlobal*	win
bes license	bes licenses	<license>	<world>	*PlainGlobal*	win
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
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
blade bit <operating system suite mask>	blade bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
block firewall action	block firewall actions	<firewall action>	<world>	*PlainGlobal*	win
blockquote <string> of <html>	blockquotes	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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 <email task action>	bodies	<string>	<email task action>	*Plain*	win
body of <html>	bodys	<html>	<html>	*Plain*	core
body of <string>	bodys	<html>	<string>	*Plain*	core
boolean <string>	booleans	<boolean>	<world>	*NamedGlobal*	core
boolean value <integer> of <wmi select>	boolean values	<boolean>	<wmi select>	*Numbered*	win
boolean value of <wmi select>	boolean values	<boolean>	<wmi select>	*Plain*	win
boot task trigger type	boot task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
boot time of <operating system>	boot times	<time>	<operating system>	*Plain*	win
br	brs	<html>	<world>	*PlainGlobal*	core
br <string>	brs	<html>	<world>	*NamedGlobal*	core
brand id of <processor>	brand ids	<integer>	<processor>	*Plain*	win
brand of <client>	brands	<string>	<client>	*Plain*	win
brand string of <processor>	brand strings	<string>	<processor>	*Plain*	win
broadcast address of <network adapter interface>	broadcast addresses	<ipv4or6 address>	<network adapter interface>	*Plain*	win
broadcast address of	broadcast	<ipv4 address>	<network ip	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<network ip interface>	addresses		interface>		
broadcast support of <network adapter interface>	broadcast supports	<boolean>	<network adapter interface>	*Plain*	win
broadcast support of <network ip interface>	broadcast supports	<boolean>	<network ip interface>	*Plain*	win
build number high of <operating system>	build number highs	<integer>	<operating system>	*Plain*	win
build number low of <operating system>	build number lows	<integer>	<operating system>	*Plain*	win
built in of <firewall open port>	built ins	<boolean>	<firewall open port>	*Plain*	win
builtin administrators group	builtin administrators groups	<security account>	<world>	*PlainGlobal*	win
builtin backup operators group	builtin backup operators groups	<security account>	<world>	*PlainGlobal*	win
builtin guests group	builtin guests groups	<security account>	<world>	*PlainGlobal*	win
builtin network configuration operators group	builtin network configuration operators groups	<security account>	<world>	*PlainGlobal*	win
builtin power users group	builtin power users groups	<security account>	<world>	*PlainGlobal*	win
builtin remote desktop users group	builtin remote desktop users groups	<security account>	<world>	*PlainGlobal*	win
builtin replicator group	builtin replicator groups	<security account>	<world>	*PlainGlobal*	win
builtin users group	builtin users groups	<security account>	<world>	*PlainGlobal*	win
byte <integer> of <file>	bytes	<integer>	<file>	*Numbered*	win
can interact with	can interact with	<boolean>	<service>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
desktop of <service>	desktops				
caption <string> of <html>	captions	<html>	<html>	*Named*	core
caption <string> of <string>	captions	<html>	<string>	*Named*	core
caption of <html>	captions	<html>	<html>	*Plain*	core
caption of <string>	captions	<html>	<string>	*Plain*	core
case insensitive regex <string>	case insensitive regexes	<regular expression>	<world>	*NamedGlobal*	regx
case insensitive regular expression <string>	case insensitive regular expressions	<regular expression>	<world>	*NamedGlobal*	regx
cast <string>	casts	<cast>	<world>	*NamedGlobal*	core
cast from of <type>	casts from	<cast>	<type>	*Plain*	core
cast returning <type>	casts returning	<cast>	<world>	*Index<type>Global*	core
category of <audit policy>	categories	<audit policy category>	<audit policy>	*Plain*	win
category of <event log record>	categories	<integer>	<event log record>	*Plain*	win
cc of <email task action>	ccs	<string>	<email task action>	*Plain*	win
change notification permission of <access control entry>	change notification permissions	<boolean>	<access control entry>	*Plain*	win
character <integer>	characters	<string>	<world>	*NumberedGlobal*	core
character <integer> of <string>	characters	<substring>	<string>	*Numbered*	core
character of <string>	characters	<substring>	<string>	*Plain*	core
checkpoint of <service>	checkpoints	<integer>	<service>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
child node <integer> of <xml dom node>	child nodes	<xml dom node>	<xml dom node>	*Numbered*	core
child node of <xml dom node>	child nodes	<xml dom node>	<xml dom node>	*Plain*	core
cidr address of <network adapter interface>	cidr addresses	<string>	<network adapter interface>	*Plain*	win
cidr address of <network adapter>	cidr addresses	<string>	<network adapter>	*Plain*	win
cidr address of <network address list>	cidr addresses	<string>	<network address list>	*Plain*	win
cidr address of <network ip interface>	cidr addresses	<string>	<network ip interface>	*Plain*	win
cidr string of <network adapter interface>	cidr strings	<string>	<network adapter interface>	*Plain*	win
cidr string of <network adapter>	cidr strings	<string>	<network adapter>	*Plain*	win
cidr string of <network address list>	cidr strings	<string>	<network address list>	*Plain*	win
cidr string of <network ip interface>	cidr strings	<string>	<network ip interface>	*Plain*	win
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
class id of <com handler task action>	class ids	<string>	<com handler task action>	*Plain*	win
class of <active device>	classes	<string>	<active device>	*Plain*	win
client	clients	<client>	<world>	*PlainGlobal*	win
client cryptography	client cryptographies	<client_cryptography>	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
client folder of <site>	client folders	<folder>	<site>	*Plain*	win
client license	client licenses	<license>	<world>	*PlainGlobal*	win
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
code page of <user>	code pages	<integer>	<user>	*Plain*	win
codepage of <file version block>	codepages	<string>	<file version block>	*Plain*	win
col <string> of <html>	cols	<html>	<html>	*Named*	core
col <string> of <string>	cols	<html>	<string>	*Named*	core
col of <html>	cols	<html>	<html>	*Plain*	core
col of <string>	cols	<html>	<string>	*Plain*	core
colgroup <string> of <html>	colgroups	<html>	<html>	*Named*	core
colgroup <string> of <string>	colgroups	<html>	<string>	*Named*	core
colgroup of <html>	colgroups	<html>	<html>	*Plain*	core
colgroup of <string>	colgroups	<html>	<string>	*Plain*	core
com handler task action type	com handler task action types	<task action type>	<world>	*PlainGlobal*	win
comment of <local group>	comments	<string>	<local group>	*Plain*	win
comment of <network share>	comments	<string>	<network share>	*Plain*	win
comment of <user>	comments	<string>	<user>	*Plain*	win
common name of	common names	<string>	<license>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<license>					
communications bit <operating system suite mask>	communications bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
communications operator flag of <user>	communications operator flags	<boolean>	<user>	*Plain*	win
competition size of <selected server>	competition sizes	<integer>	<selected server>	*Plain*	win
competition weight of <selected server>	competition weights	<integer>	<selected server>	*Plain*	win
complete time of <action>	complete times	<time>	<action>	*Plain*	win
component <integer> of <distinguished name>	components	<distinguished name component>	<distinguished name>	*Numbered*	core
component <integer> of <site version list>	components	<integer>	<site version list>	*Numbered*	core
component of <distinguished name>	components	<distinguished name component>	<distinguished name>	*Plain*	core
component string of <security identifier>	component strings	<string>	<security identifier>	*Plain*	win
compressed of <filesystem object>	compresseds	<boolean>	<filesystem object>	*Plain*	win
computer count of <bes product>	computer counts	<integer>	<bes product>	*Plain*	win
computer id	computer ids	<integer>	<world>	*PlainGlobal*	win
computer name	computer names	<string>	<world>	*PlainGlobal*	win
computer of <event log record>	computers	<string>	<event log record>	*Plain*	win
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
connection of <network>	connections	<connection>	<network>	*Plain*	win
connection status <integer>	connection statuses	<connection status>	<world>	*NumberedGlobal*	win
connection status authenticating	connection statuses authenticating	<connection status>	<world>	*PlainGlobal*	win
connection status authentication failed	connection statuses authentication failed	<connection status>	<world>	*PlainGlobal*	win
connection status authentication succeeded	connection statuses authentication succeeded	<connection status>	<world>	*PlainGlobal*	win
connection status connected	connection statuses connected	<connection status>	<world>	*PlainGlobal*	win
connection status connecting	connection statuses connecting	<connection status>	<world>	*PlainGlobal*	win
connection status disconnected	connection statuses disconnected	<connection status>	<world>	*PlainGlobal*	win
connection status	connection	<connection	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
disconnecting	statuses disconnecting	status>			
connection status hardware disabled	connection statuses hardware disabled	<connection status>	<world>	*PlainGlobal*	win
connection status hardware malfunction	connection statuses hardware malfunction	<connection status>	<world>	*PlainGlobal*	win
connection status media disconnected	connection statuses media disconnected	<connection status>	<world>	*PlainGlobal*	win
connection status no hardware present	connection statuses no hardware present	<connection status>	<world>	*PlainGlobal*	win
console connect of <session state change task trigger>	console connects	<boolean>	<session state change task trigger>	*Plain*	win
console disconnect of <session state change task trigger>	console disconnects	<boolean>	<session state change task trigger>	*Plain*	win
constrained of <action>	constraineds	<boolean>	<action>	*Plain*	win
content of <file>	contents	<file content>	<file>	*Plain*	win
control of <security descriptor>	controls	<integer>	<security descriptor>	*Plain*	win
controller of <action lock state>	controllers	<string>	<action lock state>	*Plain*	win
count of <monitor power interval>	counts	<integer>	<monitor power interval>	*Plain*	win
country code of <user>	country codes	<integer>	<user>	*Plain*	win
create file permission of <access control entry>	create file permissions	<boolean>	<access control entry>	*Plain*	win
create folder permission of <access control	create folder permissions	<boolean>	<access control entry>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
entry>					
create link permission of <access control entry>	create link permissions	<boolean>	<access control entry>	*Plain*	win
create permission of <network share>	create permissions	<boolean>	<network share>	*Plain*	win
create subkey permission of <access control entry>	create subkey permissions	<boolean>	<access control entry>	*Plain*	win
creation time of <filesystem object>	creation times	<time>	<filesystem object>	*Plain*	win
creation time of <process>	creation times	<time>	<process>	*Plain*	win
creator group group	creator group groups	<security account>	<world>	*PlainGlobal*	win
creator owner group	creator owner groups	<security account>	<world>	*PlainGlobal*	win
cryptography	cryptographies	<cryptography>	<world>	*PlainGlobal*	core
csd version of <operating system>	csd versions	<string>	<operating system>	*Plain*	win
csidl folder <integer>	csidl folders	<folder>	<world>	*NumberedGlobal*	win
current action of <running task>	current actions	<string>	<running task>	*Plain*	win
current analysis	current analyses	<fixlet>	<world>	*PlainGlobal*	win
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 firewall profile	current firewall	<firewall profile	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
type	profile types	type>			
current monitor interval of <power history>	current monitor intervals	<monitor power interval>	<power history>	*Plain*	win
current month	current months	<month>	<world>	*PlainGlobal*	core
current month_and_year	current months_and_years	<month and year>	<world>	*PlainGlobal*	core
current profile of <firewall policy>	current profiles	<firewall profile>	<firewall policy>	*Plain*	win
current profile type of <firewall>	current profile types	<firewall profile type>	<firewall>	*Plain*	win
current relay	current relays	<current relay>	<world>	*PlainGlobal*	win
current site	current sites	<site>	<world>	*PlainGlobal*	win
current system interval of <power history>	current system intervals	<system power interval>	<power history>	*Plain*	win
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*	win
current user key <logged on user> of <registry>	current user keys	<registry key>	<registry>	*Index<logged on user>*	win
current year	current years	<year>	<world>	*PlainGlobal*	core
currently active of <firewall rule>	currently actives	<boolean>	<firewall rule>	*Plain*	win
custom firewall scope	custom firewall scopes	<firewall scope>	<world>	*PlainGlobal*	win
custom site subscription effective date <string>	custom site subscription effective dates	<time>	<world>	*NamedGlobal*	win
customized of <firewall	customizeds	<boolean>	<firewall service>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
service>					
dacl of <security descriptor>	dacls	<discretionary access control list>	<security descriptor>	*Plain*	win
daily task trigger type	daily task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
data of <com handler task action>	datas	<string>	<com handler task action>	*Plain*	win
data of <task definition>	datas	<string>	<task definition>	*Plain*	win
datacenter bit <operating system suite mask>	datacenter bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
date <string>	dates	<date>	<world>	*NamedGlobal*	core
date <time zone> of <time>	dates	<date>	<time>	*Index<time zone>*	core
date of <bios>	dates	<string>	<bios>	*Plain*	win
date of <task registration info>	dates	<time>	<task registration info>	*Plain*	win
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
days interval of <daily task trigger>	days intervals	<time interval>	<daily task trigger>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
days run of <monthly task trigger>	days runs	<day of month>	<monthly task trigger>	*Plain*	win
days run of <monthlydow task trigger>	days runs	<day of week>	<monthlydow task trigger>	*Plain*	win
days run of <weekly task trigger>	days runs	<day of week>	<weekly task trigger>	*Plain*	win
dd <string> of <html>	dds	<html>	<html>	*Named*	core
dd <string> of <string>	dds	<html>	<string>	*Named*	core
dd of <html>	dds	<html>	<html>	*Plain*	core
dd of <string>	dds	<html>	<string>	*Plain*	core
december	decembers	<month>	<world>	*PlainGlobal*	core
december <integer>	decembers	<day of year>	<world>	*NumberedGlobal*	core
december <integer> of <integer>	decembers	<date>	<integer>	*Numbered*	core
december of <integer>	decembers	<month and year>	<integer>	*Plain*	core
default value of <registry key>	default values	<registry key value>	<registry key>	*Plain*	win
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
definition of <scheduled task>	definitions	<task definition>	<scheduled task>	*Plain*	win
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
del of <string>	dels	<html>	<string>	*Plain*	core
delay of <boot task trigger>	delays	<time interval>	<boot task trigger>	*Plain*	win
delay of <event task trigger>	delays	<time interval>	<event task trigger>	*Plain*	win
delay of <logon task trigger>	delays	<time interval>	<logon task trigger>	*Plain*	win
delay of <registration task trigger>	delays	<time interval>	<registration task trigger>	*Plain*	win
delay of <session state change task trigger>	delays	<time interval>	<session state change task trigger>	*Plain*	win
delete child permission of <access control entry>	delete child permissions	<boolean>	<access control entry>	*Plain*	win
delete expired task after of <task settings>	delete expired task afters	<time interval>	<task settings>	*Plain*	win
delete permission of <access control entry>	delete permissions	<boolean>	<access control entry>	*Plain*	win
delete permission of <network share>	delete permissions	<boolean>	<network share>	*Plain*	win
dependency known of <property>	dependencies known	<boolean>	<property>	*Plain*	core
descendant folder of <folder>	descendant folders	<folder>	<folder>	*Plain*	win
descendant of <folder>	descendants	<file>	<folder>	*Plain*	win
descendant of <task folder>	descendants	<scheduled task>	<task folder>	*Plain*	win
description of <active device>	descriptions	<string>	<active device>	*Plain*	win
description of <event log record>	descriptions	<string>	<event log record>	*Plain*	win
description of <firewall rule>	descriptions	<string>	<firewall rule>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
description of <network adapter>	descriptions	<string>	<network adapter>	*Plain*	win
description of <task registration info>	descriptions	<string>	<task registration info>	*Plain*	win
desired encrypt report of <client_cryptography>	desired encrypt reports	<boolean>	<client_cryptography>	*Plain*	win
desired fips mode of <cryptography>	desired fips modes	<boolean>	<cryptography>	*Plain*	core
detailed tracking category of <audit policy>	detailed tracking categories	<audit policy category>	<audit policy>	*Plain*	win
device key <string> of <registry>	device keys	<registry key>	<registry>	*Named*	win
device key of <registry>	device keys	<registry key>	<registry>	*Plain*	win
device name of <connection>	device names	<string>	<connection>	*Plain*	win
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
dhcp enabled of <network adapter>	dhcp enableds	<boolean>	<network adapter>	*Plain*	win
dhcp server of <network adapter>	dhcp servers	<ipv4 address>	<network adapter>	*Plain*	win
dialup group	dialup groups	<security account>	<world>	*PlainGlobal*	win
direct object type of <property>	direct object types	<type>	<property>	*Plain*	core
disabled state of <running task>	disabled states	<boolean>	<running task>	*Plain*	win
disabled state of <scheduled task>	disabled states	<boolean>	<scheduled task>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
disallow start when on battery of <task settings>	disallow start when on batteries	<boolean>	<task settings>	*Plain*	win
disjunction of <boolean>	disjunctions	<boolean>	<boolean>	*Plain*	core
display name of <service>	display names	<string>	<service>	*Plain*	win
display name of <task principal>	display names	<string>	<task principal>	*Plain*	win
distance of <selected server>	distances	<integer range>	<selected server>	*Plain*	win
distinguished name <string>	distinguished names	<distinguished name>	<world>	*NamedGlobal*	core
distinguished name error message of <active directory group>	distinguished name error messages	<string>	<active directory group>	*Plain*	win
distinguished name error message of <active directory local computer>	distinguished name error messages	<string>	<active directory local computer>	*Plain*	win
distinguished name error message of <active directory local user>	distinguished name error messages	<string>	<active directory local user>	*Plain*	win
distinguished name of <active directory group>	distinguished names	<string>	<active directory group>	*Plain*	win
distinguished name of <active directory local computer>	distinguished names	<string>	<active directory local computer>	*Plain*	win
distinguished name of <active directory local user>	distinguished names	<string>	<active directory local user>	*Plain*	win
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
divided by zero of <floating point>	divided by zeroes	<boolean>	<floating point>	*Plain*	core
dmi	dmis	<dmi>	<world>	*PlainGlobal*	win
dns name	dns names	<string>	<world>	*PlainGlobal*	win
dns server of <network adapter>	dns servers	<network address list>	<network adapter>	*Plain*	win
dns server of <network>	dns servers	<network address list>	<network>	*Plain*	win
dns suffix of <network adapter>	dns suffixes	<string>	<network adapter>	*Plain*	win
documentation of <task registration info>	documentations	<string>	<task registration info>	*Plain*	win
domain firewall profile type	domain firewall profile types	<firewall profile type>	<world>	*PlainGlobal*	win
domain name of <security identifier>	domain names	<string>	<security identifier>	*Plain*	win
domain profile of <firewall policy>	domain profiles	<firewall profile>	<firewall policy>	*Plain*	win
domain user	domain users	<user>	<world>	*PlainGlobal*	win
domain user <string>	domain users	<user>	<world>	*NamedGlobal*	win
download file <string>	download files	<file>	<world>	*NamedGlobal*	win
download path <string>	download paths	<string>	<world>	*NamedGlobal*	win
download storage folder	download storage folders	<download storage folder>	<world>	*PlainGlobal*	win
drive	drives	<drive>	<world>	*PlainGlobal*	win
drive <string>	drives	<drive>	<world>	*NamedGlobal*	win
drive of <filesystem object>	drives	<drive>	<filesystem object>	*Plain*	win
driver key of <active device>	driver keys	<registry key>	<active device>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
driver key of <registry key>	driver keys	<registry key>	<registry key>	*Plain*	win
driver key value name of <active device>	driver key value names	<string>	<active device>	*Plain*	win
driver running service	driver running services	<service>	<world>	*PlainGlobal*	win
driver service	driver services	<service>	<world>	*PlainGlobal*	win
driver type of <service>	driver types	<boolean>	<service>	*Plain*	win
ds access category of <audit policy>	ds access categories	<audit policy category>	<audit policy>	*Plain*	win
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
duration of <task repetition pattern>	durations	<time interval>	<task repetition pattern>	*Plain*	win
edge traversal allowed of <firewall rule>	edge traversal alloweds	<boolean>	<firewall rule>	*Plain*	win
effective access mode for <security account> of <access control list>	effective access modes for	<integer>	<access control list>	*Index<security account>*	win
effective access mode for <string> of <access control list>	effective access modes for	<integer>	<access control list>	*Named*	win
effective access system security permission for <security account> of <access control list>	effective access system security permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective access system security permission for <string> of <access control list>	effective access system security permissions for	<boolean>	<access control list>	*Named*	win
effective append permission for <security	effective append permissions for	<boolean>	<access control list>	*Index<security account>*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
account> of <access control list>					
effective append permission for <string> of <access control list>	effective append permissions for	<boolean>	<access control list>	*Named*	win
effective change notification permission for <security account> of <access control list>	effective change notification permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective change notification permission for <string> of <access control list>	effective change notification permissions for	<boolean>	<access control list>	*Named*	win
effective create file permission for <security account> of <access control list>	effective create file permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective create file permission for <string> of <access control list>	effective create file permissions for	<boolean>	<access control list>	*Named*	win
effective create folder permission for <security account> of <access control list>	effective create folder permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective create folder permission for <string> of <access control list>	effective create folder permissions for	<boolean>	<access control list>	*Named*	win
effective create link permission for <security account> of <access control list>	effective create link permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective create link permission for <string> of <access control list>	effective create link permissions for	<boolean>	<access control list>	*Named*	win
effective create subkey permission for <security account> of <access control list>	effective create subkey permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective create subkey permission for <string>	effective create subkey	<boolean>	<access control	*Named*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
of <access control list>	permissions for		list>		
effective date of <action lock state>	effective dates	<time>	<action lock state>	*Plain*	win
effective date of <setting>	effective dates	<time>	<setting>	*Plain*	win
effective delete child permission for <security account> of <access control list>	effective delete child permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective delete child permission for <string> of <access control list>	effective delete child permissions for	<boolean>	<access control list>	*Named*	win
effective delete permission for <security account> of <access control list>	effective delete permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective delete permission for <string> of <access control list>	effective delete permissions for	<boolean>	<access control list>	*Named*	win
effective enumerate subkeys permission for <security account> of <access control list>	effective enumerate subkeys permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective enumerate subkeys permission for <string> of <access control list>	effective enumerate subkeys permissions for	<boolean>	<access control list>	*Named*	win
effective execute permission for <security account> of <access control list>	effective execute permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective execute permission for <string> of <access control list>	effective execute permissions for	<boolean>	<access control list>	*Named*	win
effective generic all permission for <security account> of <access control list>	effective generic all permissions for	<boolean>	<access control list>	*Index<security account>*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
effective generic all permission for <string> of <access control list>	effective generic all permissions for	<boolean>	<access control list>	*Named*	win
effective generic execute permission for <security account> of <access control list>	effective generic execute permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective generic execute permission for <string> of <access control list>	effective generic execute permissions for	<boolean>	<access control list>	*Named*	win
effective generic read permission for <security account> of <access control list>	effective generic read permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective generic read permission for <string> of <access control list>	effective generic read permissions for	<boolean>	<access control list>	*Named*	win
effective generic write permission for <security account> of <access control list>	effective generic write permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective generic write permission for <string> of <access control list>	effective generic write permissions for	<boolean>	<access control list>	*Named*	win
effective list permission for <security account> of <access control list>	effective list permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective list permission for <string> of <access control list>	effective list permissions for	<boolean>	<access control list>	*Named*	win
effective maximum allowed permission for <security account> of <access control list>	effective maximum allowed permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective maximum allowed permission for <string> of <access control list>	effective maximum allowed permissions for	<boolean>	<access control list>	*Named*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
effective policy <security account> of <audit policy subcategory>	effective policies	<audit policy information>	<audit policy subcategory>	*Index<security account>*	win
effective query value permission for <security account> of <access control list>	effective query value permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective query value permission for <string> of <access control list>	effective query value permissions for	<boolean>	<access control list>	*Named*	win
effective read attributes permission for <security account> of <access control list>	effective read attributes permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective read attributes permission for <string> of <access control list>	effective read attributes permissions for	<boolean>	<access control list>	*Named*	win
effective read control permission for <security account> of <access control list>	effective read control permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective read control permission for <string> of <access control list>	effective read control permissions for	<boolean>	<access control list>	*Named*	win
effective read extended attributes permission for <security account> of <access control list>	effective read extended attributes permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective read extended attributes permission for <string> of <access control list>	effective read extended attributes permissions for	<boolean>	<access control list>	*Named*	win
effective read permission for <security account> of <access control list>	effective read permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective read permission for <string> of <access control list>	effective read permissions for	<boolean>	<access control list>	*Named*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
effective set value permission for <security account> of <access control list>	effective set value permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective set value permission for <string> of <access control list>	effective set value permissions for	<boolean>	<access control list>	*Named*	win
effective synchronize permission for <security account> of <access control list>	effective synchronize permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective synchronize permission for <string> of <access control list>	effective synchronize permissions for	<boolean>	<access control list>	*Named*	win
effective traverse permission for <security account> of <access control list>	effective traverse permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective traverse permission for <string> of <access control list>	effective traverse permissions for	<boolean>	<access control list>	*Named*	win
effective write attributes permission for <security account> of <access control list>	effective write attributes permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective write attributes permission for <string> of <access control list>	effective write attributes permissions for	<boolean>	<access control list>	*Named*	win
effective write dac permission for <security account> of <access control list>	effective write dac permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective write dac permission for <string> of <access control list>	effective write dac permissions for	<boolean>	<access control list>	*Named*	win
effective write extended attributes permission for <security account> of <access control list>	effective write extended attributes permissions for	<boolean>	<access control list>	*Index<security account>*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
effective write extended attributes permission for <string> of <access control list>	effective write extended attributes permissions for	<boolean>	<access control list>	*Named*	win
effective write owner permission for <security account> of <access control list>	effective write owner permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective write owner permission for <string> of <access control list>	effective write owner permissions for	<boolean>	<access control list>	*Named*	win
effective write permission for <security account> of <access control list>	effective write permissions for	<boolean>	<access control list>	*Index<security account>*	win
effective write permission for <string> of <access control list>	effective write permissions for	<boolean>	<access control list>	*Named*	win
element of <integer set>	elements	<integer>	<integer set>	*Plain*	core
element of <string set>	elements	<string>	<string set>	*Plain*	core
em <string> of <html>	ems	<html>	<html>	*Named*	core
em <string> of <string>	ems	<html>	<string>	*Named*	core
em of <html>	ems	<html>	<html>	*Plain*	core
em of <string>	ems	<html>	<string>	*Plain*	core
email address of <license>	email addresses	<string>	<license>	*Plain*	win
email task action type	email task action types	<task action type>	<world>	*PlainGlobal*	win
embedded nt bit <operating system suite mask>	embedded nt bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
embedded restricted bit <operating system suite mask>	embedded restricted bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
enabled of <firewall authorized application>	enableds	<boolean>	<firewall authorized application>	*Plain*	win
enabled of <firewall open port>	enableds	<boolean>	<firewall open port>	*Plain*	win
enabled of <firewall rule>	enableds	<boolean>	<firewall rule>	*Plain*	win
enabled of <firewall service>	enableds	<boolean>	<firewall service>	*Plain*	win
enabled of <internet connection firewall>	enableds	<boolean>	<internet connection firewall>	*Plain*	win
enabled of <port mapping>	enableds	<boolean>	<port mapping>	*Plain*	win
enabled of <scheduled task>	enableds	<boolean>	<scheduled task>	*Plain*	win
enabled of <setting>	enableds	<boolean>	<setting>	*Plain*	win
enabled of <task settings>	enableds	<boolean>	<task settings>	*Plain*	win
enabled of <task trigger>	enableds	<boolean>	<task trigger>	*Plain*	win
encrypt report failure message of <client_cryptography>	encrypt report failure messages	<string>	<client_cryptography>	*Plain*	win
encrypt report of <client_cryptography>	encrypt reports	<boolean>	<client_cryptography>	*Plain*	win
encryption certificate of <license>	encryption certificates	<x509 certificate>	<license>	*Plain*	win
end boundary of <task trigger>	end boundaries	<time>	<task trigger>	*Plain*	win
end of <substring>	ends	<string position>	<substring>	*Plain*	core
end of <time range>	ends	<time>	<time range>	*Plain*	core
engine pid of <running task>	engine pids	<integer>	<running task>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
enterprise bit <operating system suite mask>	enterprise bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
entry of <access control list>	entries	<access control entry>	<access control list>	*Plain*	win
enumerate subkeys permission of <access control entry>	enumerate subkeys permissions	<boolean>	<access control entry>	*Plain*	win
environment	environments	<environment>	<world>	*PlainGlobal*	win
error <string>	errors	<undefined>	<world>	*NamedGlobal*	core
error event log event type	error event log event types	<event log event type>	<world>	*PlainGlobal*	win
escape of <string>	escapes	<string>	<string>	*Plain*	win
evaluation of <license>	evaluations	<boolean>	<license>	*Plain*	win
evaluationcycle of <client>	evaluationcycles	<evaluation cycle>	<client>	*Plain*	win
event id of <event log record>	event ids	<integer>	<event log record>	*Plain*	win
event log <string>	event logs	<event log>	<world>	*NamedGlobal*	win
event log event type <integer>	event log event types	<event log event type>	<world>	*NumberedGlobal*	win
event task trigger type	event task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
event type of <event log record>	event types	<event log event type>	<event log record>	*Plain*	win
everyone group	everyone groups	<security account>	<world>	*PlainGlobal*	win
exceptions allowed of <firewall profile>	exceptions alloweds	<boolean>	<firewall profile>	*Plain*	win
excluded interface of <firewall profile>	excluded interfaces	<string>	<firewall profile>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
exec task action type	exec task action types	<task action type>	<world>	*PlainGlobal*	win
executable file format of <file>	executable file formats	<string>	<file>	*Plain*	win
execute permission of <access control entry>	execute permissions	<boolean>	<access control entry>	*Plain*	win
execute permission of <network share>	execute permissions	<boolean>	<network share>	*Plain*	win
execution time limit of <task settings>	execution time limits	<time interval>	<task settings>	*Plain*	win
execution time limit of <task trigger>	execution time limits	<time interval>	<task trigger>	*Plain*	win
exit code of <action>	exit codes	<integer>	<action>	*Plain*	win
expand environment string of <string>	expand environment strings	<string>	<string>	*Plain*	win
expand x64 environment string of <string>	expand x64 environment strings	<string>	<string>	*Plain*	win
expiration date of <action lock state>	expiration dates	<time>	<action lock state>	*Plain*	win
expiration date of <bes product>	expiration dates	<date>	<bes product>	*Plain*	win
expiration date of <license>	expiration dates	<time>	<license>	*Plain*	win
expiration state of <license>	expiration states	<string>	<license>	*Plain*	win
extended family of <processor>	extended families	<integer>	<processor>	*Plain*	win
extended model of <processor>	extended models	<integer>	<processor>	*Plain*	win
external port of <port mapping>	external ports	<integer>	<port mapping>	*Plain*	win
extrema of <date>	extremas	<(date, date)>	<date>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
extrema of <day of month>	extremas	<(day of month, day of month)>	<day of month>	*Plain*	core
extrema of <day of year>	extremas	<(day of year, day of year)>	<day of year>	*Plain*	core
extrema of <floating point>	extremas	<(floating point, floating point)>	<floating point>	*Plain*	core
extrema of <hertz>	extremas	<(hertz, hertz)>	<hertz>	*Plain*	core
extrema of <integer>	extremas	<(integer, integer)>	<integer>	*Plain*	core
extrema of <ipv4 address>	extremas	<(ipv4 address, ipv4 address)>	<ipv4 address>	*Plain*	core
extrema of <ipv4or6 address>	extremas	<(ipv4or6 address, ipv4or6 address)>	<ipv4or6 address>	*Plain*	core
extrema of <ipv6 address>	extremas	<(ipv6 address, ipv6 address)>	<ipv6 address>	*Plain*	core
extrema of <month and year>	extremas	<(month and year, month and year)>	<month and year>	*Plain*	core
extrema of <month>	extremas	<(month, month)>	<month>	*Plain*	core
extrema of <number of months>	extremas	<(number of months, number of months)>	<number of months>	*Plain*	core
extrema of <site version list>	extremas	<(site version list, site version list)>	<site version list>	*Plain*	core
extrema of <time interval>	extremas	<(time interval, time interval)>	<time interval>	*Plain*	core
extrema of <time of day>	extremas	<(time of day, time of day)>	<time of day>	*Plain*	core
extrema of <time>	extremas	<(time, time)>	<time>	*Plain*	core
extrema of <version>	extremas	<(version, version)>	<version>	*Plain*	core
extrema of <year>	extremas	<(year, year)>	<year>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
false	falses	<boolean>	<world>	*PlainGlobal*	core
family name of <processor>	family names	<string>	<processor>	*Plain*	win
family of <network interface>	families	<integer>	<network interface>	*Plain*	win
family of <processor>	families	<integer>	<processor>	*Plain*	win
feature mask of <processor>	feature masks	<integer>	<processor>	*Plain*	win
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*	win
file <string> of <folder>	files	<file>	<folder>	*Named*	win
file extension <string> of <registry>	file extensions	<registry key>	<registry>	*Named*	win
file of <folder>	files	<file>	<folder>	*Plain*	win
file of <service>	files	<file>	<service>	*Plain*	win
file system type of <drive>	file system types	<string>	<drive>	*Plain*	win
file type <string> of <registry>	file types	<registry key>	<registry>	*Named*	win
file version of <file>	file versions	<version>	<file>	*Plain*	win
file_and_print firewall service type	file_and_print firewall service types	<firewall service type>	<world>	*PlainGlobal*	win
file_supports_encryption of <drive>	file_supports_encryptions	<boolean>	<drive>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
file_supports_object_ids of <drive>	file_supports_object_idss	<boolean>	<drive>	*Plain*	win
file_supports_reparse_points of <drive>	file_supports_reparse_pointss	<boolean>	<drive>	*Plain*	win
file_supports_sparse_files of <drive>	file_supports_sparse_filess	<boolean>	<drive>	*Plain*	win
file_volume_quotas of <drive>	file_volume_quotass	<boolean>	<drive>	*Plain*	win
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*	win
find file <string> of <folder>	find files	<file>	<folder>	*Named*	win
find folder <string> of <folder>	find folders	<folder>	<folder>	*Named*	win
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*	win
firewall	firewalls	<firewall>	<world>	*PlainGlobal*	win
firewall action <integer>	firewall actions	<firewall action>	<world>	*NumberedGlobal*	win
firewall enabled of <firewall profile>	firewalls enabled	<boolean>	<firewall profile>	*Plain*	win
firewall local policy modify state <integer>	firewall local policy modify states	<firewall local policy modify state>	<world>	*NumberedGlobal*	win
firewall of <connection>	firewalls	<internet connection firewall>	<connection>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
firewall profile type <integer>	firewall profile types	<firewall profile type>	<world>	*NumberedGlobal*	win
firewall scope <integer>	firewall scopes	<firewall scope>	<world>	*NumberedGlobal*	win
firewall service type <integer>	firewall service types	<firewall service type>	<world>	*NumberedGlobal*	win
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 child of <xml dom node>	first children	<xml dom node>	<xml dom node>	*Plain*	core
first friday of <month and year>	first fridays	<date>	<month and year>	*Plain*	core
first interface scheduled task	first interface scheduled tasks	<scheduled task>	<world>	*PlainGlobal*	win
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 raw version block of <file>	first raw version blocks	<file version block>	<file>	*Plain*	win
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*	win
first start time of <application usage summary>	first start times	<time>	<application usage summary>	*Plain*	win
first sunday of <month and year>	first sundays	<date>	<month and year>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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*	win
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*	win
folder <string> of <drive>	folders	<folder>	<drive>	*Named*	win
folder <string> of <folder>	folders	<folder>	<folder>	*Named*	win
folder of <folder>	folders	<folder>	<folder>	*Plain*	win
following text of <string position>	following texts	<substring>	<string position>	*Plain*	core
following text of <substring>	following texts	<substring>	<substring>	*Plain*	core
force logoff interval of <security database>	force logoff intervals	<time interval>	<security database>	*Plain*	win
format <string>	formats	<format>	<world>	*NamedGlobal*	core
free space of <drive>	free spaces	<integer>	<drive>	*Plain*	win
friday	fridays	<day of week>	<world>	*PlainGlobal*	core
friendly name of <active device>	friendly names	<string>	<active device>	*Plain*	win
friendly name of <network adapter>	friendly names	<string>	<network adapter>	*Plain*	win
from of <email task action>	froms	<string>	<email task action>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
fs_case_is_preserved of <drive>	fs_case_is_preserveds	<boolean>	<drive>	*Plain*	win
fs_case_sensitive of <drive>	fs_case_sensitives	<boolean>	<drive>	*Plain*	win
fs_file_compression of <drive>	fs_file_compressions	<boolean>	<drive>	*Plain*	win
fs_persistent_acls of <drive>	fs_persistent_acls	<boolean>	<drive>	*Plain*	win
fs_unicode_stored_on_disk of <drive>	fs_unicode_stored_on_disks	<boolean>	<drive>	*Plain*	win
fs_vol_is_compressed of <drive>	fs_vol_is_compresseds	<boolean>	<drive>	*Plain*	win
full gateway address of <selected server>	full gateway addresses	<ipv4or6 address>	<selected server>	*Plain*	win
full name of <user>	full names	<string>	<user>	*Plain*	win
full of <power level>	fulls	<boolean>	<power level>	*Plain*	win
full wmi <string>	full wmis	<wmi>	<world>	*NamedGlobal*	win
gateway address <integer> of <selected server>	gateway addresses	<ipv4or6 address>	<selected server>	*Numbered*	win
gateway address of <selected server>	gateway addresses	<ipv4or6 address>	<selected server>	*Plain*	win
gateway list of <network adapter>	gateway lists	<network address list>	<network adapter>	*Plain*	win
gateway of <network adapter>	gateways	<ipv4 address>	<network adapter>	*Plain*	win
gather schedule authority of <site>	gather schedule authoritys	<string>	<site>	*Plain*	win
gather schedule time interval of <site>	gather schedule time intervals	<time interval>	<site>	*Plain*	win
gather url of <license>	gather urls	<string>	<license>	*Plain*	win
gdi object count of	gdi object	<integer>	<process>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<process>	counts				
generic all permission of <access control entry>	generic all permissions	<boolean>	<access control entry>	*Plain*	win
generic execute permission of <access control entry>	generic execute permissions	<boolean>	<access control entry>	*Plain*	win
generic read permission of <access control entry>	generic read permissions	<boolean>	<access control entry>	*Plain*	win
generic write permission of <access control entry>	generic write permissions	<boolean>	<access control entry>	*Plain*	win
ghz	ghzs	<hertz>	<world>	*PlainGlobal*	core
globally open port of <firewall profile>	globally open ports	<firewall open port>	<firewall profile>	*Plain*	win
globally open port of <firewall service>	globally open ports	<firewall open port>	<firewall service>	*Plain*	win
gp override firewall local policy modify state	gp override firewall local policy modify states	<firewall local policy modify state>	<world>	*PlainGlobal*	win
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*	win
group <string> of <active directory local computer>	groups	<active directory group>	<active directory local computer>	*Named*	win
group <string> of <active directory local user>	groups	<active directory group>	<active directory local user>	*Named*	win
group id of <task	group ids	<string>	<task principal>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
principal>					
group leader of <action>	group leaders	<boolean>	<action>	*Plain*	win
group logon of <task principal>	group logons	<boolean>	<task principal>	*Plain*	win
group of <active directory local computer>	groups	<active directory group>	<active directory local computer>	*Plain*	win
group of <active directory local user>	groups	<active directory group>	<active directory local user>	*Plain*	win
group of <security descriptor>	groups	<security identifier>	<security descriptor>	*Plain*	win
grouping of <firewall rule>	groupings	<string>	<firewall rule>	*Plain*	win
groups error message of <active directory local computer>	groups error messages	<string>	<active directory local computer>	*Plain*	win
groups error message of <active directory local user>	groups error messages	<string>	<active directory local user>	*Plain*	win
guest privilege of <user>	guest privileges	<boolean>	<user>	*Plain*	win
guid of <connection>	guids	<string>	<connection>	*Plain*	win
h1 <string> of <html>	h1s	<html>	<html>	*Named*	core
h1 <string> of <string>	h1s	<html>	<string>	*Named*	core
h1 of <html>	h1s	<html>	<html>	*Plain*	core
h1 of <string>	h1s	<html>	<string>	*Plain*	core
h2 <string> of <html>	h2s	<html>	<html>	*Named*	core
h2 <string> of <string>	h2s	<html>	<string>	*Named*	core
h2 of <html>	h2s	<html>	<html>	*Plain*	core
h2 of <string>	h2s	<html>	<string>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
handle count of <process>	handle counts	<integer>	<process>	*Plain*	win
hardware id of <active device>	hardware ids	<string>	<active device>	*Plain*	win
has blank sa password of <local mssql database>	has blank sa passwords	<boolean>	<local mssql database>	*Plain*	win
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
head of <string>	heads	<html>	<string>	*Plain*	core
header <string> of <fixlet>	headers	<fixlet_header>	<fixlet>	*Named*	win
header field of <email task action>	header fields	<task named value pair>	<email task action>	*Plain*	win
header of <fixlet>	headers	<fixlet_header>	<fixlet>	*Plain*	win
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
hidden of <filesystem object>	hiddens	<boolean>	<filesystem object>	*Plain*	win
hidden of <task settings>	hiddens	<boolean>	<task settings>	*Plain*	win
high priority	high priorities	<priority class>	<world>	*PlainGlobal*	win
highest runlevel of <task principal>	highest runlevels	<boolean>	<task principal>	*Plain*	win
home directory drive of <user>	home directory drives	<string>	<user>	*Plain*	win
home directory of <user>	home directories	<string>	<user>	*Plain*	win
home directory required flag of <user>	home directory required flags	<boolean>	<user>	*Plain*	win
host name of <root server>	host names	<string>	<root server>	*Plain*	win
hostname	hostnames	<string>	<world>	*PlainGlobal*	win
hour	hours	<time interval>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
html <string> of <string>	htmls	<html>	<string>	*Named*	core
html concatenation <string> of <html>	html concatenations	<html>	<html>	*Named*	core
html concatenation of <html>	html concatenations	<html>	<html>	*Plain*	core
html of <html>	htmls	<html>	<html>	*Plain*	core
html of <string>	htmls	<html>	<string>	*Plain*	core
html tag <(string, html)>	html tags	<html>	<world>	*Index<(string, html)>Global*	core
html tag <(string, html attribute list)>	html tags	<html>	<world>	*Index<(string, html attribute list)>Global*	core
html tag <(string, html attribute list, html)>	html tags	<html>	<world>	*Index<(string, html attribute list, html)>Global*	core
html tag <(string, html attribute list, string)>	html tags	<html>	<world>	*Index<(string, html attribute list, string)>Global*	core
html tag <(string, string)>	html tags	<html>	<world>	*Index<(string, string)>Global*	core
html tag <string> of <html>	html tags	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
html tag <string> of <string>	html tags	<html>	<string>	*Named*	core
hyperthreading capable	hyperthreading capables	<boolean>	<world>	*PlainGlobal*	win
hyperthreading enabled	hyperthreading enableds	<boolean>	<world>	*PlainGlobal*	win
hz	hzs	<hertz>	<world>	*PlainGlobal*	core
ia64 of <operating system>	ia64s	<boolean>	<operating system>	*Plain*	win
icmp settings of <firewall profile>	icmp settingses	<firewall icmp settings>	<firewall profile>	*Plain*	win
icmp types_and_codes string of <firewall rule>	icmp types_and_codes strings	<string>	<firewall rule>	*Plain*	win
icon index of <file shortcut>	icon indexes	<integer>	<file shortcut>	*Plain*	win
icon pathname of <file shortcut>	icon pathnames	<string>	<file shortcut>	*Plain*	win
id of <action>	ids	<integer>	<action>	*Plain*	win
id of <file version block>	ids	<string>	<file version block>	*Plain*	win
id of <fixlet>	ids	<integer>	<fixlet>	*Plain*	win
id of <process>	ids	<integer>	<process>	*Plain*	win
id of <root server>	ids	<integer>	<root server>	*Plain*	win
id of <site group>	ids	<integer>	<site group>	*Plain*	win
id of <task action>	ids	<string>	<task action>	*Plain*	win
id of <task network settings>	ids	<string>	<task network settings>	*Plain*	win
id of <task principal>	ids	<string>	<task principal>	*Plain*	win
id of <task trigger>	ids	<string>	<task trigger>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
identifier of <metabase value>	identifiers	<metabase identifier>	<metabase value>	*Plain*	win
idle duration of <task idle settings>	idle durations	<time interval>	<task idle settings>	*Plain*	win
idle priority	idle priorities	<priority class>	<world>	*PlainGlobal*	win
idle setting of <task settings>	idle settings	<task idle settings>	<task settings>	*Plain*	win
idle state	idle states	<power state>	<world>	*PlainGlobal*	win
idle task trigger type	idle task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
ignore new instance of <task settings>	ignore new instances	<boolean>	<task settings>	*Plain*	win
image file of <process>	image files	<file>	<process>	*Plain*	win
image path of <service>	image paths	<string>	<service>	*Plain*	win
inbound blocked firewall local policy modify state	inbound blocked firewall local policy modify states	<firewall local policy modify state>	<world>	*PlainGlobal*	win
inbound connections allowed of <firewall profile>	inbound connections alloweds	<boolean>	<firewall profile>	*Plain*	win
inbound of <firewall rule>	inbounds	<boolean>	<firewall rule>	*Plain*	win
index type of <property>	index types	<type>	<property>	*Plain*	core
inexact of <floating point>	inexacts	<boolean>	<floating point>	*Plain*	core
infinite of <floating point>	infinites	<boolean>	<floating point>	*Plain*	core
information event log event type	information event log event types	<event log event type>	<world>	*PlainGlobal*	win
inherit attribute of <metabase value>	inherit attributes	<boolean>	<metabase value>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
inheritance of <access control entry>	inheritances	<integer>	<access control entry>	*Plain*	win
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
insert path attribute of <metabase value>	insert path attributes	<boolean>	<metabase value>	*Plain*	win
install folder <integer>	install folders	<folder>	<world>	*NumberedGlobal*	win
instance guid of <running task>	instance guids	<string>	<running task>	*Plain*	win
instance name of <local mssql database>	instance names	<string>	<local mssql database>	*Plain*	win
instance of <application usage summary>	instances	<application usage summary instance>	<application usage summary>	*Plain*	win
integer <integer>	integers	<integer>	<world>	*NumberedGlobal*	core
integer <string>	integers	<integer>	<world>	*NamedGlobal*	core
integer <string> of <smbios structure>	integers	<integer>	<smbios structure>	*Named*	win
integer ceiling of <floating point>	integer ceilings	<integer>	<floating point>	*Plain*	core
integer floor of <floating point>	integer floors	<integer>	<floating point>	*Plain*	core
integer in <(integer, integer)>	integers in	<integer>	<world>	*Index<(integer, integer)>Global*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
integer in <(integer, integer, integer)>	integers in	<integer>	<world>	*Index<(integer, integer, integer)>Global*	core
integer to <integer>	integers to	<integer>	<world>	*NumberedGlobal*	core
integer value <integer> of <wmi select>	integer values	<integer>	<wmi select>	*Numbered*	win
integer value <string> of <smbios structure>	integer values	<smbios value>	<smbios structure>	*Named*	win
integer value of <wmi select>	integer values	<integer>	<wmi select>	*Plain*	win
interactive group	interactive groups	<security account>	<world>	*PlainGlobal*	win
interactive token logon of <task principal>	interactive token logons	<boolean>	<task principal>	*Plain*	win
interactive token password logon of <task principal>	interactive token password logons	<boolean>	<task principal>	*Plain*	win
interdomain trust account flag of <user>	interdomain trust account flags	<boolean>	<user>	*Plain*	win
interface <integer> of <network>	interfaces	<network interface>	<network>	*Numbered*	win
interface of <firewall rule>	interfaces	<string>	<firewall rule>	*Plain*	win
interface of <network>	interfaces	<network interface>	<network>	*Plain*	win
interface types string of <firewall rule>	interface types strings	<string>	<firewall rule>	*Plain*	win
internal port of <port mapping>	internal ports	<integer>	<port mapping>	*Plain*	win
internet connection firewall of <network adapter>	internet connection firewalls	<internet connection firewall>	<network adapter>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
internet protocol <integer>	internet protocols	<internet protocol>	<world>	*NumberedGlobal*	win
intersection of <integer set>	intersections	<integer set>	<integer set>	*Plain*	core
intersection of <string set>	intersections	<string set>	<string set>	*Plain*	core
interval of <task repetition pattern>	intervals	<time interval>	<task repetition pattern>	*Plain*	win
invalid before of <x509 certificate>	invalid befores	<time>	<x509 certificate>	*Plain*	core
invalid of <floating point>	invalids	<boolean>	<floating point>	*Plain*	core
invalid state	invalid states	<power state>	<world>	*PlainGlobal*	win
io other count of <process>	io other counts	<integer>	<process>	*Plain*	win
io other size of <process>	io other sizes	<integer>	<process>	*Plain*	win
io read count of <process>	io read counts	<integer>	<process>	*Plain*	win
io read size of <process>	io read sizes	<integer>	<process>	*Plain*	win
io write count of <process>	io write counts	<integer>	<process>	*Plain*	win
io write size of <process>	io write sizes	<integer>	<process>	*Plain*	win
ip address of <selected server>	ip addresses	<ipv4or6 address>	<selected server>	*Plain*	win
ip interface <integer> of <network>	ip interfaces	<network ip interface>	<network>	*Numbered*	win
ip interface of <network>	ip interfaces	<network ip interface>	<network>	*Plain*	win
ip version <integer>	ip versions	<ip version>	<world>	*NumberedGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
ip version of <firewall authorized application>	ip versions	<ip version>	<firewall authorized application>	*Plain*	win
ip version of <firewall open port>	ip versions	<ip version>	<firewall open port>	*Plain*	win
ip version of <firewall service>	ip versions	<ip version>	<firewall service>	*Plain*	win
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*	win
ipv4 interface of <network>	ipv4 interfaces	<network adapter interface>	<network>	*Plain*	win
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 dns server of <network adapter>	ipv4or6 dns servers	<ipv4or6 address>	<network adapter>	*Plain*	win
ipv4or6 interface of <network adapter>	ipv4or6 interfaces	<network adapter interface>	<network adapter>	*Plain*	win
ipv4or6 interface of <network>	ipv4or6 interfaces	<network adapter interface>	<network>	*Plain*	win
ipv6	ipv6s	<ip version>	<world>	*PlainGlobal*	core
ipv6 address <string>	ipv6 addresses	<ipv6 address>	<world>	*NamedGlobal*	core
ipv6 address of <network adapter>	ipv6 addresses	<ipv6 address>	<network adapter>	*Plain*	win
ipv6 dns server of	ipv6 dns servers	<ipv6 address>	<network	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<network adapter>			adapter>		
ipv6 interface of <network adapter>	ipv6 interfaces	<network adapter interface>	<network adapter>	*Plain*	win
ipv6 interface of <network>	ipv6 interfaces	<network adapter interface>	<network>	*Plain*	win
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
january	januarys	<month>	<world>	*PlainGlobal*	core
january <integer>	januarys	<day of year>	<world>	*NumberedGlobal*	core
january <integer> of <integer>	januarys	<date>	<integer>	*Numbered*	core
january of <integer>	januarys	<month and year>	<integer>	*Plain*	core
july	julys	<month>	<world>	*PlainGlobal*	core
july <integer>	julys	<day of year>	<world>	*NumberedGlobal*	core
july <integer> of <integer>	julys	<date>	<integer>	*Numbered*	core
july of <integer>	julys	<month and year>	<integer>	*Plain*	core
june	junes	<month>	<world>	*PlainGlobal*	core
june <integer>	junes	<day of year>	<world>	*NumberedGlobal*	core
june <integer> of <integer>	junes	<date>	<integer>	*Numbered*	core
june of <integer>	junes	<month and year>	<integer>	*Plain*	core
kbd <string> of <html>	kbds	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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 time of <process>	kernel times	<time interval>	<process>	*Plain*	win
key <string> of <file section>	keys	<string>	<file section>	*Named*	win
key <string> of <file>	keys	<string>	<file>	*Named*	win
key <string> of <metabase key>	keys	<metabase key>	<metabase key>	*Named*	win
key <string> of <metabase>	keys	<metabase key>	<metabase>	*Named*	win
key <string> of <registry key>	keys	<registry key>	<registry key>	*Named*	win
key <string> of <registry>	keys	<registry key>	<registry>	*Named*	win
key of <metabase key>	keys	<metabase key>	<metabase key>	*Plain*	win
key of <metabase>	keys	<metabase key>	<metabase>	*Plain*	win
key of <registry key>	keys	<registry key>	<registry key>	*Plain*	win
khz	khzs	<hertz>	<world>	*PlainGlobal*	core
language of <file version block>	languages	<string>	<file version block>	*Plain*	win
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*	win
last child of <xml dom node>	last children	<xml dom node>	<xml dom node>	*Plain*	core
last gather time of	last gather times	<time>	<site>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<site>					
last logoff of <user>	last logoffs	<time>	<user>	*Plain*	win
last logon of <user>	last logons	<time>	<user>	*Plain*	win
last monitor interval in <power state> of <power history>	last monitor intervals in	<monitor power interval>	<power history>	*Index<power state>*	win
last monitor interval in monitor off state of <power history>	last monitor intervals in monitor off state	<monitor power interval>	<power history>	*Plain*	win
last monitor interval in monitor on state of <power history>	last monitor intervals in monitor on state	<monitor power interval>	<power history>	*Plain*	win
last relay select time	last relay select times	<time>	<world>	*PlainGlobal*	win
last run time of <scheduled task>	last run times	<time>	<scheduled task>	*Plain*	win
last start time of <application usage summary instance>	last start times	<time>	<application usage summary instance>	*Plain*	win
last start time of <application usage summary>	last start times	<time>	<application usage summary>	*Plain*	win
last system interval in <power state> of <power history>	last system intervals in	<system power interval>	<power history>	*Index<power state>*	win
last system interval in active state of <power history>	last system intervals in active state	<system power interval>	<power history>	*Plain*	win
last system interval in idle state of <power history>	last system intervals in idle state	<system power interval>	<power history>	*Plain*	win
last system interval in logged off state of <power history>	last system intervals in logged off state	<system power interval>	<power history>	*Plain*	win
last system interval in off state of <power	last system intervals in off	<system power interval>	<power history>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
history>	state				
last system interval in standby state of <power history>	last system intervals in standby state	<system power interval>	<power history>	*Plain*	win
last task result of <scheduled task>	last task results	<integer>	<scheduled task>	*Plain*	win
last time seen of <application usage summary instance>	last times seen	<time>	<application usage summary instance>	*Plain*	win
last time seen of <application usage summary>	last times seen	<time>	<application usage summary>	*Plain*	win
last write time of <registry key>	last write times	<time>	<registry key>	*Plain*	win
leap of <year>	leaps	<boolean>	<year>	*Plain*	core
lease expires of <network adapter>	leases expire	<time>	<network adapter>	*Plain*	win
lease obtained of <network adapter>	leases obtained	<time>	<network adapter>	*Plain*	win
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 <event log record>	lengths	<integer>	<event log record>	*Plain*	win
length of <month and year>	lengths	<time interval>	<month and year>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
length of <rope>	lengths	<integer>	<rope>	*Plain*	core
length of <smbios structure>	lengths	<integer>	<smbios structure>	*Plain*	win
length of <string>	lengths	<integer>	<string>	*Plain*	core
length of <time range>	lengths	<time interval>	<time range>	*Plain*	core
length of <year>	lengths	<time interval>	<year>	*Plain*	core
less significance <integer> of <floating point>	less significances	<floating point>	<floating point>	*Numbered*	core
li <string> of <html>	lis	<html>	<html>	*Named*	core
li <string> of <string>	lis	<html>	<string>	*Named*	core
li of <html>	lis	<html>	<html>	*Plain*	core
li of <string>	lis	<html>	<string>	*Plain*	core
line <integer> of <file>	lines	<file line>	<file>	*Numbered*	win
line containing <string> of <file>	lines containing	<file line>	<file>	*Named*	win
line number of <file line>	line numbers	<integer>	<file line>	*Plain*	win
line of <file>	lines	<file line>	<file>	*Plain*	win
line starting with <string> of <file>	lines starting with	<file line>	<file>	*Named*	win
link <string> of <html>	links	<html>	<html>	*Named*	core
link <string> of <string>	links	<html>	<string>	*Named*	core
link of <html>	links	<html>	<html>	*Plain*	core
link of <string>	links	<html>	<string>	*Plain*	core
link speed of <network adapter>	link speeds	<integer>	<network adapter>	*Plain*	win
list permission of <access control entry>	list permissions	<boolean>	<access control entry>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
local addresses string of <firewall rule>	local addresses strings	<string>	<firewall rule>	*Plain*	win
local administrator	local administrators	<boolean>	<world>	*PlainGlobal*	win
local computer of <active directory server>	local computers	<active directory local computer>	<active directory server>	*Plain*	win
local group	local groups	<local group>	<world>	*PlainGlobal*	win
local group <string>	local groups	<local group>	<world>	*NamedGlobal*	win
local mssql database	local mssql databases	<local mssql database>	<world>	*PlainGlobal*	win
local mssql database <string>	local mssql databases	<local mssql database>	<world>	*NamedGlobal*	win
local policy modify state of <firewall>	local policy modify states	<firewall local policy modify state>	<firewall>	*Plain*	win
local policy of <firewall>	local policies	<firewall policy>	<firewall>	*Plain*	win
local ports string of <firewall rule>	local ports strings	<string>	<firewall rule>	*Plain*	win
local service group	local service groups	<security account>	<world>	*PlainGlobal*	win
local subnet firewall scope	local subnet firewall scopes	<firewall scope>	<world>	*PlainGlobal*	win
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*	win
local user <string>	local users	<user>	<world>	*NamedGlobal*	win
local user <string> of <active directory server>	local users	<active directory local user>	<active directory server>	*Named*	win
local user of <active directory server>	local users	<active directory local user>	<active directory server>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
location information of <active device>	location informations	<string>	<active device>	*Plain*	win
location of <filesystem object>	locations	<string>	<filesystem object>	*Plain*	win
lock string of <action lock state>	lock strings	<string>	<action lock state>	*Plain*	win
locked of <action lock state>	lockeds	<boolean>	<action lock state>	*Plain*	win
locked out flag of <user>	locked out flags	<boolean>	<user>	*Plain*	win
logged off state	logged off states	<power state>	<world>	*PlainGlobal*	win
logged on user	logged on users	<logged on user>	<world>	*PlainGlobal*	win
logged on user <string> of <active directory server>	logged on users	<active directory local user>	<active directory server>	*Named*	win
logged on user of <active directory server>	logged on users	<active directory local user>	<active directory server>	*Plain*	win
logged on user of <user>	logged on users	<logged on user>	<user>	*Plain*	win
logical processor count	logical processor counts	<integer>	<world>	*PlainGlobal*	win
login account of <service>	login accounts	<string>	<service>	*Plain*	win
login mode of <local mssql database>	login modes	<integer>	<local mssql database>	*Plain*	win
logon count of <user>	logon counts	<integer>	<user>	*Plain*	win
logon logoff category of <audit policy>	logon logoff categories	<audit policy category>	<audit policy>	*Plain*	win
logon script of <user>	logon scripts	<string>	<user>	*Plain*	win
logon server of <user>	logon servers	<string>	<user>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
logon task trigger type	logon task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
loopback of <network adapter interface>	loopbacks	<boolean>	<network adapter interface>	*Plain*	win
loopback of <network adapter>	loopbacks	<boolean>	<network adapter>	*Plain*	win
loopback of <network ip interface>	loopbacks	<boolean>	<network ip interface>	*Plain*	win
low of <power level>	lows	<boolean>	<power level>	*Plain*	win
lua runlevel of <task principal>	lua runlevels	<boolean>	<task principal>	*Plain*	win
mac address of <network adapter interface>	mac addresses	<string>	<network adapter interface>	*Plain*	win
mac address of <network adapter>	mac addresses	<string>	<network adapter>	*Plain*	win
mac of <operating system>	macs	<boolean>	<operating system>	*Plain*	win
main gather service	main gather services	<service>	<world>	*PlainGlobal*	win
main processor	main processors	<processor>	<world>	*PlainGlobal*	win
major version of <operating system>	major versions	<integer>	<operating system>	*Plain*	win
manufacturer of <active device>	manufacturers	<string>	<active device>	*Plain*	win
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*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
match <regular expression> of <string>	matches	<regular expression match>	<string>	*Index<regular expression>*	regx
maximum allowed permission of <access control entry>	maximum allowed permissions	<boolean>	<access control entry>	*Plain*	win
maximum of <date>	maxima	<date>	<date>	*Plain*	core
maximum of <day of month>	maxima	<day of month>	<day of month>	*Plain*	core
maximum of <day of year>	maxima	<day of year>	<day of year>	*Plain*	core
maximum of <evaluation cycle>	maximums	<integer>	<evaluation cycle>	*Plain*	win
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
maximum of <time of day>	maxima	<time of day>	<time of day>	*Plain*	core
maximum of <time>	maxima	<time>	<time>	*Plain*	core
maximum of <version>	maxima	<version>	<version>	*Plain*	core
maximum of <year>	maxima	<year>	<year>	*Plain*	core
maximum password age of <security database>	maximum password ages	<time interval>	<security database>	*Plain*	win
maximum seat count of <license>	maximum seat counts	<integer>	<license>	*Plain*	win
maximum storage of <user>	maximum storages	<integer>	<user>	*Plain*	win
maximum transmission unit of <network adapter>	maximum transmission units	<integer>	<network adapter>	*Plain*	win
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
media type <integer>	media types	<media type>	<world>	*NumberedGlobal*	win
media type bridge	media types bridge	<media type>	<world>	*PlainGlobal*	win
media type direct	media types direct	<media type>	<world>	*PlainGlobal*	win
media type isdn	media types isdn	<media type>	<world>	*PlainGlobal*	win
media type lan	media types	<media type>	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
	lans				
media type of <connection>	media types	<media type>	<connection>	*Plain*	win
media type phone	media types phone	<media type>	<world>	*PlainGlobal*	win
media type pppoe	media types pppoe	<media type>	<world>	*PlainGlobal*	win
media type shared access host lan	media types shared access host lan	<media type>	<world>	*PlainGlobal*	win
media type shared access host ras	media types shared access host ras	<media type>	<world>	*PlainGlobal*	win
media type tunnel	media types tunnel	<media type>	<world>	*PlainGlobal*	win
member of <local group>	members	<local group member>	<local group>	*Plain*	win
member of <site group>	members	<boolean>	<site group>	*Plain*	win
message body of <show message task action>	message bodies	<string>	<show message task action>	*Plain*	win
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
metabase	metabases	<metabase>	<world>	*PlainGlobal*	win
metric <integer> of <operating system>	metrics	<integer>	<operating system>	*Numbered*	win
mhz	mhzs	<hertz>	<world>	*PlainGlobal*	core
microsecond	microseconds	<time interval>	<world>	*PlainGlobal*	core
midnight	midnights	<time of day>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
millisecond	milliseconds	<time interval>	<world>	*PlainGlobal*	core
minimum of <date>	minima	<date>	<date>	*Plain*	core
minimum of <day of month>	minima	<day of month>	<day of month>	*Plain*	core
minimum of <day of year>	minima	<day of year>	<day of year>	*Plain*	core
minimum of <floating point>	minima	<floating point>	<floating point>	*Plain*	core
minimum of <hertz>	minima	<hertz>	<hertz>	*Plain*	core
minimum of <integer>	minima	<integer>	<integer>	*Plain*	core
minimum of <ipv4 address>	minima	<ipv4 address>	<ipv4 address>	*Plain*	core
minimum of <ipv4or6 address>	minima	<ipv4or6 address>	<ipv4or6 address>	*Plain*	core
minimum of <ipv6 address>	minima	<ipv6 address>	<ipv6 address>	*Plain*	core
minimum of <month and year>	minima	<month and year>	<month and year>	*Plain*	core
minimum of <month>	minima	<month>	<month>	*Plain*	core
minimum of <number of months>	minima	<number of months>	<number of months>	*Plain*	core
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
minimum password age of <security database>	minimum password ages	<time interval>	<security database>	*Plain*	win
minimum password length of <security database>	minimum password lengths	<integer>	<security database>	*Plain*	win
minor version of <operating system>	minor versions	<integer>	<operating system>	*Plain*	win
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
missed run count of <scheduled task>	missed run counts	<integer>	<scheduled task>	*Plain*	win
model of <processor>	models	<integer>	<processor>	*Plain*	win
modification time of <filesystem object>	modification times	<time>	<filesystem object>	*Plain*	win
module <string>	modules	<module>	<world>	*NamedGlobal*	core
monday	mondays	<day of week>	<world>	*PlainGlobal*	core
monitor interval of <power history>	monitor intervals	<monitor power interval>	<power history>	*Plain*	win
monitor invalid state	monitor invalid states	<power state>	<world>	*PlainGlobal*	win
monitor off state	monitor off states	<power state>	<world>	*PlainGlobal*	win
monitor on state	monitor on states	<power state>	<world>	*PlainGlobal*	win
monitor standby state	monitor standby states	<power state>	<world>	*PlainGlobal*	win
month	months	<number of months>	<world>	*PlainGlobal*	core
month <integer>	months	<month>	<world>	*NumberedGlo*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
				bal	
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
monthly task trigger type	monthly task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
monthlydow task trigger type	monthlydow task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
months run of <monthly task trigger>	months runs	<month>	<monthly task trigger>	*Plain*	win
months run of <monthlydow task trigger>	months runs	<month>	<monthlydow task trigger>	*Plain*	win
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*	win
multicast support of <network adapter>	multicast supports	<boolean>	<network adapter>	*Plain*	win
multicast support of <network ip interface>	multicast supports	<boolean>	<network ip interface>	*Plain*	win
multiplicity of <date with multiplicity>	multiplicities	<integer>	<date with multiplicity>	*Plain*	core
multiplicity of <day of month with multiplicity>	multiplicities	<integer>	<day of month with multiplicity>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
multiplicity of <day of week with multiplicity>	multiplicities	<integer>	<day of week with multiplicity>	*Plain*	core
multiplicity of <day of year with multiplicity>	multiplicities	<integer>	<day of year with multiplicity>	*Plain*	core
multiplicity of <floating point with multiplicity>	multiplicities	<integer>	<floating point with multiplicity>	*Plain*	core
multiplicity of <hertz with multiplicity>	multiplicities	<integer>	<hertz with multiplicity>	*Plain*	core
multiplicity of <integer with multiplicity>	multiplicities	<integer>	<integer with multiplicity>	*Plain*	core
multiplicity of <ipv4 address with multiplicity>	multiplicities	<integer>	<ipv4 address with multiplicity>	*Plain*	core
multiplicity of <ipv4or6 address with multiplicity>	multiplicities	<integer>	<ipv4or6 address with multiplicity>	*Plain*	core
multiplicity of <ipv6 address with multiplicity>	multiplicities	<integer>	<ipv6 address with multiplicity>	*Plain*	core
multiplicity of <month and year with multiplicity>	multiplicities	<integer>	<month and year with multiplicity>	*Plain*	core
multiplicity of <month with multiplicity>	multiplicities	<integer>	<month with multiplicity>	*Plain*	core
multiplicity of <number of months with multiplicity>	multiplicities	<integer>	<number of months with multiplicity>	*Plain*	core
multiplicity of <site version list with multiplicity>	multiplicities	<integer>	<site version list with multiplicity>	*Plain*	core
multiplicity of <string with multiplicity>	multiplicities	<integer>	<string with multiplicity>	*Plain*	core
multiplicity of <time interval with multiplicity>	multiplicities	<integer>	<time interval with multiplicity>	*Plain*	core
multiplicity of <time of day with multiplicity>	multiplicities	<integer>	<time of day with multiplicity>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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*	win
name of <active directory local user>	names	<string>	<active directory local user>	*Plain*	win
name of <application usage summary instance>	names	<string>	<application usage summary instance>	*Plain*	win
name of <application usage summary>	names	<string>	<application usage summary>	*Plain*	win
name of <audit policy category>	names	<string>	<audit policy category>	*Plain*	win
name of <audit policy subcategory>	names	<string>	<audit policy subcategory>	*Plain*	win
name of <bes product>	names	<string>	<bes product>	*Plain*	win
name of <binary operator>	names	<string>	<binary operator>	*Plain*	core
name of <cast>	names	<string>	<cast>	*Plain*	core
name of <connection>	names	<string>	<connection>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
name of <drive>	names	<string>	<drive>	*Plain*	win
name of <environment variable>	names	<string>	<environment variable>	*Plain*	win
name of <filesystem object>	names	<string>	<filesystem object>	*Plain*	win
name of <firewall authorized application>	names	<string>	<firewall authorized application>	*Plain*	win
name of <firewall open port>	names	<string>	<firewall open port>	*Plain*	win
name of <firewall rule>	names	<string>	<firewall rule>	*Plain*	win
name of <firewall service>	names	<string>	<firewall service>	*Plain*	win
name of <fixlet_header>	names	<string>	<fixlet_header>	*Plain*	win
name of <local group>	names	<string>	<local group>	*Plain*	win
name of <logged on user>	names	<string>	<logged on user>	*Plain*	win
name of <metabase key>	names	<string>	<metabase key>	*Plain*	win
name of <network adapter>	names	<string>	<network adapter>	*Plain*	win
name of <network share>	names	<string>	<network share>	*Plain*	win
name of <operating system>	names	<string>	<operating system>	*Plain*	win
name of <port mapping>	names	<string>	<port mapping>	*Plain*	win
name of <process>	names	<string>	<process>	*Plain*	win
name of <registry key value>	names	<string>	<registry key value>	*Plain*	win
name of <registry key>	names	<string>	<registry key>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
name of <running task>	names	<string>	<running task>	*Plain*	win
name of <scheduled task>	names	<string>	<scheduled task>	*Plain*	win
name of <selected server>	names	<string>	<selected server>	*Plain*	win
name of <setting>	names	<string>	<setting>	*Plain*	win
name of <site>	names	<string>	<site>	*Plain*	win
name of <smbios structure>	names	<string>	<smbios structure>	*Plain*	win
name of <smbios value>	names	<string>	<smbios value>	*Plain*	win
name of <task folder>	names	<string>	<task folder>	*Plain*	win
name of <task named value pair>	names	<string>	<task named value pair>	*Plain*	win
name of <task network settings>	names	<string>	<task network settings>	*Plain*	win
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*	win
name of <wmi select>	names	<string>	<wmi select>	*Plain*	win
nan of <floating point>	nans	<boolean>	<floating point>	*Plain*	core
native registry	native registries	<registry>	<world>	*PlainGlobal*	win
network	networks	<network>	<world>	*PlainGlobal*	win
network group	network groups	<security account>	<world>	*PlainGlobal*	win
network service group	network service groups	<security account>	<world>	*PlainGlobal*	win
network setting of <task settings>	network settings	<task network settings>	<task settings>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
network share	network shares	<network share>	<world>	*PlainGlobal*	win
network share <string>	network shares	<network share>	<world>	*NamedGlobal*	win
next line of <file line>	next lines	<file line>	<file line>	*Plain*	win
next run time of <scheduled task>	next run times	<time>	<scheduled task>	*Plain*	win
next sibling of <xml dom node>	next siblings	<xml dom node>	<xml dom node>	*Plain*	core
no password required flag of <user>	no password required flags	<boolean>	<user>	*Plain*	win
node name of <xml dom node>	node names	<string>	<xml dom node>	*Plain*	core
node type of <xml dom node>	node types	<integer>	<xml dom node>	*Plain*	core
node value of <xml dom node>	node values	<string>	<xml dom node>	*Plain*	core
non windows server count of <bes product>	non windows server counts	<integer>	<bes product>	*Plain*	win
none firewall service type	none firewall service types	<firewall service type>	<world>	*PlainGlobal*	win
none logon of <task principal>	none logons	<boolean>	<task principal>	*Plain*	win
noon	noons	<time of day>	<world>	*PlainGlobal*	core
normal account flag of <user>	normal account flags	<boolean>	<user>	*Plain*	win
normal of <filesystem object>	normals	<boolean>	<filesystem object>	*Plain*	win
normal of <floating point>	normals	<boolean>	<floating point>	*Plain*	core
normal of <power level>	normals	<boolean>	<power level>	*Plain*	win
normal priority	normal priorities	<priority class>	<world>	*PlainGlobal*	win
notifications disabled of	notifications	<boolean>	<firewall profile>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<firewall profile>	disableds				
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
nt domain controller product type	nt domain controller product types	<operating system product type>	<world>	*PlainGlobal*	win
nt server product type	nt server product types	<operating system product type>	<world>	*PlainGlobal*	win
nt workstation product type	nt workstation product types	<operating system product type>	<world>	*PlainGlobal*	win
numeric type of <drive>	numeric types	<integer>	<drive>	*Plain*	win
numeric value of <string>	numeric values	<integer>	<string>	*Plain*	core
object access category of <audit policy>	object access categories	<audit policy category>	<audit policy>	*Plain*	win
october	octobers	<month>	<world>	*PlainGlobal*	core
october <integer>	octobers	<day of year>	<world>	*NumberedGlobal*	core
october <integer> of <integer>	octobers	<date>	<integer>	*Numbered*	core
october of <integer>	octobers	<month and year>	<integer>	*Plain*	core
oem code page	oem code pages	<integer>	<world>	*PlainGlobal*	win
off state	off states	<power state>	<world>	*PlainGlobal*	win
offer accepted of <action>	offer accepteds	<boolean>	<action>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
offer of <action>	offers	<boolean>	<action>	*Plain*	win
offline of <filesystem object>	offlines	<boolean>	<filesystem object>	*Plain*	win
offset of <smbios value>	offsets	<integer>	<smbios value>	*Plain*	win
ok firewall local policy modify state	ok firewall local policy modify states	<firewall local policy modify state>	<world>	*PlainGlobal*	win
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
oldest record number of <event log>	oldest record numbers	<integer>	<event log>	*Plain*	win
one bit of <bit set>	one bits	<integer>	<bit set>	*Plain*	core
only raw version block of <file>	only raw version blocks	<file version block>	<file>	*Plain*	win
only version block of <file>	only version blocks	<file version block>	<file>	*Plain*	win
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*	win
operating system product type <integer>	operating system product types	<operating system product type>	<world>	*NumberedGlobal*	win
options of <port mapping>	optionss	<integer>	<port mapping>	*Plain*	win
ordered list <string> of <html>	ordered lists	<html>	<html>	*Named*	core
ordered list <string> of	ordered lists	<html>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<string>					
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*	win
origin fixlet id of <action>	origin fixlet ids	<integer>	<action>	*Plain*	win
outbound connections allowed of <firewall profile>	outbound connections alloweds	<boolean>	<firewall profile>	*Plain*	win
outbound of <firewall rule>	outbounds	<boolean>	<firewall rule>	*Plain*	win
overflow of <floating point>	overflows	<boolean>	<floating point>	*Plain*	core
owner document of <xml dom node>	owner documents	<xml dom document>	<xml dom node>	*Plain*	core
owner of <security descriptor>	owners	<security identifier>	<security descriptor>	*Plain*	win
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
pad of <version>	pads	<version>	<version>	*Plain*	core
page fault count of <process>	page fault counts	<integer>	<process>	*Plain*	win
page file usage of <process>	page file usages	<integer>	<process>	*Plain*	win
parallel instance of <task settings>	parallel instances	<boolean>	<task settings>	*Plain*	win
parameter <string>	parameters	<string>	<world>	*NamedGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
parameter <string> of <action>	parameters	<string>	<action>	*Named*	win
parent folder of <filesystem object>	parent folders	<folder>	<filesystem object>	*Plain*	win
parent node of <xml dom node>	parent nodes	<xml dom node>	<xml dom node>	*Plain*	core
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
password age of <user>	password ages	<time interval>	<user>	*Plain*	win
password change disabled flag of <user>	password change disabled flags	<boolean>	<user>	*Plain*	win
password expiration disabled flag of <user>	password expiration disabled flags	<boolean>	<user>	*Plain*	win
password expired of <user>	passwords expired	<boolean>	<user>	*Plain*	win
password history length of <security database>	password history lengths	<integer>	<security database>	*Plain*	win
password logon of <task principal>	password logons	<boolean>	<task principal>	*Plain*	win
password of <network share>	passwords	<string>	<network share>	*Plain*	win
path of <exec task action>	paths	<string>	<exec task action>	*Plain*	win
path of <network share>	paths	<string>	<network share>	*Plain*	win
path of <running task>	paths	<string>	<running task>	*Plain*	win
path of <scheduled task>	paths	<string>	<scheduled task>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
path of <task folder>	paths	<string>	<task folder>	*Plain*	win
pathname of <file shortcut>	pathnames	<string>	<file shortcut>	*Plain*	win
pathname of <filesystem object>	pathnames	<string>	<filesystem object>	*Plain*	win
peak page file usage of <process>	peak page file usages	<integer>	<process>	*Plain*	win
peak working set size of <process>	peak working set sizes	<integer>	<process>	*Plain*	win
pem encoded certificate of <file>	pem encoded certificates	<x509 certificate>	<file>	*Plain*	win
pending login	pending logins	<boolean>	<world>	*PlainGlobal*	win
pending login of <action>	pending logins	<boolean>	<action>	*Plain*	win
pending of <action>	pendings	<boolean>	<action>	*Plain*	win
pending restart	pending restarts	<boolean>	<world>	*PlainGlobal*	win
pending restart <string>	pending restarts	<boolean>	<world>	*NamedGlobal*	win
pending restart name	pending restart names	<string>	<world>	*PlainGlobal*	win
pending restart of <action>	pending restarts	<boolean>	<action>	*Plain*	win
pending time of <action>	pending times	<time>	<action>	*Plain*	win
per user policy <security account> of <audit policy subcategory>	per user policies	<audit policy information>	<audit policy subcategory>	*Index<security account>*	win
performance counter frequency of <operating system>	performance counter frequencies	<hertz>	<operating system>	*Plain*	win
performance counter of <operating system>	performance counters	<integer>	<operating system>	*Plain*	win
permission permission	permission	<boolean>	<network share>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
of <network share>	permissions				
personal bit <operating system suite mask>	personal bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
physical processor count	physical processor counts	<integer>	<world>	*PlainGlobal*	win
platform id of <operating system>	platform ids	<integer>	<operating system>	*Plain*	win
plugged of <power level>	pluggeds	<boolean>	<power level>	*Plain*	win
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*	win
point to point of <network ip interface>	point to points	<boolean>	<network ip interface>	*Plain*	win
policy change category of <audit policy>	policy change categories	<audit policy category>	<audit policy>	*Plain*	win
port mapping of <internet connection firewall>	port mappings	<port mapping>	<internet connection firewall>	*Plain*	win
port number of <selected server>	port numbers	<integer>	<selected server>	*Plain*	win
port of <firewall open port>	ports	<integer>	<firewall open port>	*Plain*	win
position <integer> of <string>	positions	<string position>	<string>	*Numbered*	core
position of <string>	positions	<string position>	<string>	*Plain*	core
power history	power histories	<power history>	<world>	*PlainGlobal*	win
power level	power levels	<power level>	<world>	*PlainGlobal*	win
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
previous line of <file line>	previous lines	<file line>	<file line>	*Plain*	win
previous sibling of <xml dom node>	previous siblings	<xml dom node>	<xml dom node>	*Plain*	core
primary group id of <user>	primary group ids	<integer>	<user>	*Plain*	win
primary language of <language>	primary languages	<primary language>	<language>	*Plain*	win
primary wins server of <network adapter>	primary wins servers	<ipv4 address>	<network adapter>	*Plain*	win
principal of <task definition>	principals	<task principal>	<task definition>	*Plain*	win
print operator flag of <user>	print operator flags	<boolean>	<user>	*Plain*	win
priority of <selected server>	priorities	<integer>	<selected server>	*Plain*	win
priority of <task settings>	priorities	<integer>	<task settings>	*Plain*	win
private firewall profile type	private firewall profile types	<firewall profile type>	<world>	*PlainGlobal*	win
private profile of <firewall policy>	private profiles	<firewall profile>	<firewall policy>	*Plain*	win
privilege of <security account>	privileges	<string>	<security account>	*Plain*	win
privilege use category of	privilege use	<audit policy	<audit policy>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<audit policy>	categories	category>			
problem id of <active device>	problem ids	<integer>	<active device>	*Plain*	win
process	processes	<process>	<world>	*PlainGlobal*	win
process image file name of <firewall authorized application>	process image file names	<string>	<firewall authorized application>	*Plain*	win
processor	processors	<processor>	<world>	*PlainGlobal*	win
processor <integer>	processors	<processor>	<world>	*NumberedGlobal*	win
product info numeric of <operating system>	product info numerics	<integer>	<operating system>	*Plain*	win
product info string of <operating system>	product info strings	<string>	<operating system>	*Plain*	win
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*	win
product type of <operating system>	product types	<operating system product type>	<operating system>	*Plain*	win
product version of <file>	product versions	<version>	<file>	*Plain*	win
profile <firewall profile type> of <firewall rule>	profiles	<boolean>	<firewall rule>	*Index<firewall profile type>*	win
profile folder of <user>	profile folders	<string>	<user>	*Plain*	win
property <string>	properties	<property>	<world>	*NamedGlobal*	core
property <string> of <type>	properties	<property>	<type>	*Named*	core
property <string> of <wmi object>	properties	<wmi select>	<wmi object>	*Named*	win
property of <type>	properties	<property>	<type>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
property of <wmi object>	properties	<wmi select>	<wmi object>	*Plain*	win
property returning <type>	properties returning	<property>	<world>	*Index<type>Global*	core
property returning <type> of <type>	properties returning	<property>	<type>	*Index<type>*	core
protocol of <firewall open port>	protocols	<internet protocol>	<firewall open port>	*Plain*	win
protocol of <firewall rule>	protocols	<internet protocol>	<firewall rule>	*Plain*	win
protocol of <port mapping>	protocols	<string>	<port mapping>	*Plain*	win
public firewall profile type	public firewall profile types	<firewall profile type>	<world>	*PlainGlobal*	win
public profile of <firewall policy>	public profiles	<firewall profile>	<firewall policy>	*Plain*	win
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
query value permission of <access control entry>	query value permissions	<boolean>	<access control entry>	*Plain*	win
queue instance of <task settings>	queue instances	<boolean>	<task settings>	*Plain*	win
queued state of <running task>	queued states	<boolean>	<running task>	*Plain*	win
queued state of <scheduled task>	queued states	<boolean>	<scheduled task>	*Plain*	win
quota nonpaged pool usage of <process>	quota nonpaged pool usages	<integer>	<process>	*Plain*	win
quota paged pool usage	quota paged	<integer>	<process>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
of <process>	pool usages				
quota peak nonpaged pool usage of <process>	quota peak nonpaged pool usages	<integer>	<process>	*Plain*	win
quota peak paged pool usage of <process>	quota peak paged pool usages	<integer>	<process>	*Plain*	win
ram	rams	<ram>	<world>	*PlainGlobal*	win
random access memory	random access memories	<ram>	<world>	*PlainGlobal*	win
random delay of <daily task trigger>	random delays	<time interval>	<daily task trigger>	*Plain*	win
random delay of <monthly task trigger>	random delays	<time interval>	<monthly task trigger>	*Plain*	win
random delay of <monthlydow task trigger>	random delays	<time interval>	<monthlydow task trigger>	*Plain*	win
random delay of <time task trigger>	random delays	<time interval>	<time task trigger>	*Plain*	win
random delay of <weekly task trigger>	random delays	<time interval>	<weekly task trigger>	*Plain*	win
range after <time> of <time range>	ranges after	<time range>	<time range>	*Index<time>*	core
range before <time> of <time range>	ranges before	<time range>	<time range>	*Index<time>*	core
range of <monitor power interval>	ranges	<time range>	<monitor power interval>	*Plain*	win
range of <system power interval>	ranges	<time range>	<system power interval>	*Plain*	win
raw file version of <file>	raw file versions	<version>	<file>	*Plain*	win
raw product version of <file>	raw product versions	<version>	<file>	*Plain*	win
raw version block <integer> of <file>	raw version blocks	<file version block>	<file>	*Numbered*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
raw version block <string> of <file>	raw version blocks	<file version block>	<file>	*Named*	win
raw version block of <file>	raw version blocks	<file version block>	<file>	*Plain*	win
raw version of <file>	raw versions	<version>	<file>	*Plain*	win
read attributes permission of <access control entry>	read attributes permissions	<boolean>	<access control entry>	*Plain*	win
read control permission of <access control entry>	read control permissions	<boolean>	<access control entry>	*Plain*	win
read extended attributes permission of <access control entry>	read extended attributes permissions	<boolean>	<access control entry>	*Plain*	win
read permission of <access control entry>	read permissions	<boolean>	<access control entry>	*Plain*	win
read permission of <network share>	read permissions	<boolean>	<network share>	*Plain*	win
readonly of <filesystem object>	readonlys	<boolean>	<filesystem object>	*Plain*	win
ready state of <running task>	ready states	<boolean>	<running task>	*Plain*	win
ready state of <scheduled task>	ready states	<boolean>	<scheduled task>	*Plain*	win
realtime priority	realtime priorities	<priority class>	<world>	*PlainGlobal*	win
recent application	recent applications	<application>	<world>	*PlainGlobal*	win
recent application <string>	recent applications	<application>	<world>	*NamedGlobal*	win
record <integer> of <event log>	records	<event log record>	<event log>	*Numbered*	win
record count of <event log>	record counts	<integer>	<event log>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
record number of <event log record>	record numbers	<integer>	<event log record>	*Plain*	win
record of <event log>	records	<event log record>	<event log>	*Plain*	win
reference attribute of <metabase value>	reference attributes	<boolean>	<metabase value>	*Plain*	win
regapp	regapps	<application>	<world>	*PlainGlobal*	win
regapp <string>	regapps	<application>	<world>	*NamedGlobal*	win
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*	win
registration address of <client>	registration addresses	<ipv4or6 address>	<client>	*Plain*	win
registration cidr address of <client>	registration cidr addresses	<string>	<client>	*Plain*	win
registration info of <task definition>	registration infos	<task registration info>	<task definition>	*Plain*	win
registration mac address of <client>	registration mac addresses	<string>	<client>	*Plain*	win
registration subnet address of <client>	registration subnet addresses	<ipv4or6 address>	<client>	*Plain*	win
registration task trigger type	registration task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
registry	registries	<registry>	<world>	*PlainGlobal*	win
regular expression <string>	regular expressions	<regular expression>	<world>	*NamedGlobal*	regx
relative significance place <integer> of <floating point>	relative significance places	<floating point>	<floating point>	*Numbered*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
relative significance place of <floating point>	relative significance places	<floating point>	<floating point>	*Plain*	core
relay service	relay services	<service>	<world>	*PlainGlobal*	win
release of <operating system>	releases	<string>	<operating system>	*Plain*	win
relevance of <fixlet>	relevances	<boolean>	<fixlet>	*Plain*	win
relevant fixlet of <site>	relevant fixlets	<fixlet>	<site>	*Plain*	win
relevant offer action of <site>	relevant offer actions	<action>	<site>	*Plain*	win
remote addresses of <firewall authorized application>	remote addresseses	<string>	<firewall authorized application>	*Plain*	win
remote addresses of <firewall open port>	remote addresseses	<string>	<firewall open port>	*Plain*	win
remote addresses of <firewall service>	remote addresseses	<string>	<firewall service>	*Plain*	win
remote addresses string of <firewall rule>	remote addresses strings	<string>	<firewall rule>	*Plain*	win
remote admin settings of <firewall profile>	remote admin settingses	<firewall remote admin settings>	<firewall profile>	*Plain*	win
remote connect of <session state change task trigger>	remote connects	<boolean>	<session state change task trigger>	*Plain*	win
remote desktop firewall service type	remote desktop firewall service types	<firewall service type>	<world>	*PlainGlobal*	win
remote disconnect of <session state change task trigger>	remote disconnects	<boolean>	<session state change task trigger>	*Plain*	win
remote interactive logon group	remote interactive logon groups	<security account>	<world>	*PlainGlobal*	win
remote of <logged on	remotes	<boolean>	<logged on user>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
user>					
remote ports string of <firewall rule>	remote ports strings	<string>	<firewall rule>	*Plain*	win
repetition of <task trigger>	repetitions	<task repetition pattern>	<task trigger>	*Plain*	win
replyto of <email task action>	replytos	<string>	<email task action>	*Plain*	win
restart count of <task settings>	restart counts	<integer>	<task settings>	*Plain*	win
restart interval of <task settings>	restart intervals	<time interval>	<task settings>	*Plain*	win
restart on idle of <task idle settings>	restart on idles	<boolean>	<task idle settings>	*Plain*	win
result type of <binary operator>	result types	<type>	<binary operator>	*Plain*	core
result type of <cast>	result types	<type>	<cast>	*Plain*	core
result type of <property>	result types	<type>	<property>	*Plain*	core
result type of <unary operator>	result types	<type>	<unary operator>	*Plain*	core
right operand type of <binary operator>	right operand types	<type>	<binary operator>	*Plain*	core
right shift <integer> of <bit set>	right shifts	<bit set>	<bit set>	*Numbered*	core
root folder of <drive>	root folders	<folder>	<drive>	*Plain*	win
root server	root servers	<root server>	<world>	*PlainGlobal*	win
rope <string>	ropes	<rope>	<world>	*NamedGlobal*	core
rsop computer wmi	rsop computer wmis	<wmi>	<world>	*PlainGlobal*	win
rsop user wmi <security identifier>	rsop user wmis	<wmi>	<world>	*Index<security identifier>Global*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
rule group currently enabled <string> of <firewall>	rule group currently enableds	<boolean>	<firewall>	*Named*	win
rule group enabled <string> of <firewall profile>	rule group enableds	<boolean>	<firewall profile>	*Named*	win
rule of <firewall service restriction>	rules	<firewall rule>	<firewall service restriction>	*Plain*	win
rule of <firewall>	rules	<firewall rule>	<firewall>	*Plain*	win
run on fifth week in month of <monthlydow task trigger>	run on fifth week in months	<boolean>	<monthlydow task trigger>	*Plain*	win
run on first week in month of <monthlydow task trigger>	run on first week in months	<boolean>	<monthlydow task trigger>	*Plain*	win
run on fourth week in month of <monthlydow task trigger>	run on fourth week in months	<boolean>	<monthlydow task trigger>	*Plain*	win
run on last day in month of <monthly task trigger>	run on last day in months	<boolean>	<monthly task trigger>	*Plain*	win
run on last week in month of <monthlydow task trigger>	run on last week in months	<boolean>	<monthlydow task trigger>	*Plain*	win
run on second week in month of <monthlydow task trigger>	run on second week in months	<boolean>	<monthlydow task trigger>	*Plain*	win
run on third week in month of <monthlydow task trigger>	run on third week in months	<boolean>	<monthlydow task trigger>	*Plain*	win
run only when idle of <task settings>	run only when idles	<boolean>	<task settings>	*Plain*	win
run only when network available of <task settings>	run only when network availables	<boolean>	<task settings>	*Plain*	win
running application	running applications	<application>	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
running application <string>	running applications	<application>	<world>	*NamedGlobal*	win
running of <application usage summary>	runnings	<boolean>	<application usage summary>	*Plain*	win
running of <local mssql database>	runnings	<boolean>	<local mssql database>	*Plain*	win
running service	running services	<service>	<world>	*PlainGlobal*	win
running service <string>	running services	<service>	<world>	*NamedGlobal*	win
running state of <running task>	running states	<boolean>	<running task>	*Plain*	win
running state of <scheduled task>	running states	<boolean>	<scheduled task>	*Plain*	win
running task	running tasks	<running task>	<world>	*PlainGlobal*	win
s4u logon of <task principal>	s4u logons	<boolean>	<task principal>	*Plain*	win
sacl of <security descriptor>	sacls	<system access control list>	<security descriptor>	*Plain*	win
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*	win
sample time of <active directory local computer>	sample times	<time>	<active directory local computer>	*Plain*	win
sample time of <active directory local user>	sample times	<time>	<active directory local user>	*Plain*	win
saturday	saturdays	<day of week>	<world>	*PlainGlobal*	core
scheduled task	scheduled tasks	<scheduled task>	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
scheduled task <string>	scheduled tasks	<scheduled task>	<world>	*NamedGlobal*	win
scheduled task <string> of <task folder>	scheduled tasks	<scheduled task>	<task folder>	*Named*	win
scheduled task of <task folder>	scheduled tasks	<scheduled task>	<task folder>	*Plain*	win
scope of <firewall authorized application>	scopes	<firewall scope>	<firewall authorized application>	*Plain*	win
scope of <firewall open port>	scopes	<firewall scope>	<firewall open port>	*Plain*	win
scope of <firewall service>	scopes	<firewall scope>	<firewall service>	*Plain*	win
script flag of <user>	script flags	<boolean>	<user>	*Plain*	win
seat count state of <license>	seat count states	<string>	<license>	*Plain*	win
seat of <license>	seats	<integer>	<license>	*Plain*	win
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
secondary wins server of <network adapter>	secondary wins servers	<ipv4 address>	<network adapter>	*Plain*	win
section <string> of <file>	sections	<file section>	<file>	*Named*	win
secure attribute of <metabase value>	secure attributes	<boolean>	<metabase value>	*Plain*	win
security account <string>	security accounts	<security account>	<world>	*NamedGlobal*	win
security database	security databases	<security database>	<world>	*PlainGlobal*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
security descriptor of <file>	security descriptors	<security descriptor>	<file>	*Plain*	win
security descriptor of <folder>	security descriptors	<security descriptor>	<folder>	*Plain*	win
security descriptor of <network share>	security descriptors	<security descriptor>	<network share>	*Plain*	win
security descriptor of <registry key>	security descriptors	<security descriptor>	<registry key>	*Plain*	win
security descriptor of <scheduled task>	security descriptors	<security descriptor>	<scheduled task>	*Plain*	win
security descriptor of <service>	security descriptors	<security descriptor>	<service>	*Plain*	win
security descriptor of <task folder>	security descriptors	<security descriptor>	<task folder>	*Plain*	win
security descriptor of <task registration info>	security descriptors	<security descriptor>	<task registration info>	*Plain*	win
security event log	security event logs	<event log>	<world>	*PlainGlobal*	win
select <string> of <wmi>	selects	<wmi select>	<wmi>	*Named*	win
select <string> of <xml dom node>	selects	<xml dom node>	<xml dom node>	*Named*	core
select object <string> of <wmi>	select objects	<wmi object>	<wmi>	*Named*	win
selected server	selected servers	<selected server>	<world>	*PlainGlobal*	win
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
server of <email task action>	servers	<string>	<email task action>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
server operator flag of <user>	server operator flags	<boolean>	<user>	*Plain*	win
server trust account flag of <user>	server trust account flags	<boolean>	<user>	*Plain*	win
service	services	<service>	<world>	*PlainGlobal*	win
service <string>	services	<service>	<world>	*NamedGlobal*	win
service account logon of <task principal>	service account logons	<boolean>	<task principal>	*Plain*	win
service group	service groups	<security account>	<world>	*PlainGlobal*	win
service key value name of <active device>	service key value names	<string>	<active device>	*Plain*	win
service name of <firewall rule>	service names	<string>	<firewall rule>	*Plain*	win
service name of <service>	service names	<string>	<service>	*Plain*	win
service of <firewall profile>	services	<firewall service>	<firewall profile>	*Plain*	win
service pack major version of <operating system>	service pack major versions	<integer>	<operating system>	*Plain*	win
service pack minor version of <operating system>	service pack minor versions	<integer>	<operating system>	*Plain*	win
service restricted <(string, string)> of <firewall service restriction>	service restricteds	<boolean>	<firewall service restriction>	*Index<(string, string)>*	win
service restriction of <firewall>	service restrictions	<firewall service restriction>	<firewall>	*Plain*	win
service specific exit code of <service>	service specific exit codes	<integer>	<service>	*Plain*	win
session id of <process>	session ids	<integer>	<process>	*Plain*	win
session lock of <session	session locks	<boolean>	<session state	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
state change task trigger>			change task trigger>		
session state change task trigger type	session state change task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
session unlock of <session state change task trigger>	session unlocks	<boolean>	<session state change task trigger>	*Plain*	win
set of <integer>	sets	<integer set>	<integer>	*Plain*	core
set of <string>	sets	<string set>	<string>	*Plain*	core
set value permission of <access control entry>	set value permissions	<boolean>	<access control entry>	*Plain*	win
setting <string> of <client>	settings	<setting>	<client>	*Named*	win
setting <string> of <site>	settings	<setting>	<site>	*Named*	win
setting of <client>	settings	<setting>	<client>	*Plain*	win
setting of <site>	settings	<setting>	<site>	*Plain*	win
setting of <task definition>	settings	<task settings>	<task definition>	*Plain*	win
sha1 of <file>	sha1s	<string>	<file>	*Plain*	win
sha1 of <x509 certificate>	sha1s	<string>	<x509 certificate>	*Plain*	core
shortcut of <file>	shortcuts	<file shortcut>	<file>	*Plain*	win
show message task action type	show message task action types	<task action type>	<world>	*PlainGlobal*	win
sid of <active directory group>	sids	<security identifier>	<active directory group>	*Plain*	win
sid of <logged on user>	sids	<security identifier>	<logged on user>	*Plain*	win
sid of <security	sids	<security	<security	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
account>		identifier>	account>		
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
single user ts bit <operating system suite mask>	single user ts bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
singular name of <property>	singular names	<string>	<property>	*Plain*	core
site	sites	<site>	<world>	*PlainGlobal*	win
site <string>	sites	<site>	<world>	*NamedGlobal*	win
site number of <license>	site numbers	<integer>	<license>	*Plain*	win
site of <fixlet>	sites	<site>	<fixlet>	*Plain*	win
site tag of <site>	site tags	<string>	<site>	*Plain*	win
site url of <bes product>	site urls	<string>	<bes product>	*Plain*	win
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*	win
size of <application usage summary instance>	sizes	<integer>	<application usage summary instance>	*Plain*	win
size of <file>	sizes	<integer>	<file>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
size of <integer set>	sizes	<integer>	<integer set>	*Plain*	core
size of <ram>	sizes	<integer>	<ram>	*Plain*	win
size of <registry key value>	sizes	<integer>	<registry key value>	*Plain*	win
size of <string set>	sizes	<integer>	<string set>	*Plain*	core
size of <type>	sizes	<integer>	<type>	*Plain*	core
small <string> of <html>	smalls	<html>	<html>	*Named*	core
small <string> of <string>	smalls	<html>	<string>	*Named*	core
small business bit <operating system suite mask>	small business bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
small business restricted bit <operating system suite mask>	small business restricted bits	<boolean>	<world>	*Index<operating system suite mask>Global*	win
small of <html>	smalls	<html>	<html>	*Plain*	core
small of <string>	smalls	<html>	<string>	*Plain*	core
smbios	smbioses	<smbios>	<world>	*PlainGlobal*	win
source of <event log record>	sources	<string>	<event log record>	*Plain*	win
source of <task registration info>	sources	<string>	<task registration info>	*Plain*	win
span <string> of <html>	spans	<html>	<html>	*Named*	core
span <string> of <string>	spans	<html>	<string>	*Named*	core
span of <html>	spans	<html>	<html>	*Plain*	core
span of <string>	spans	<html>	<string>	*Plain*	core
speed of <processor>	speeds	<hertz>	<processor>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
standard deviation of <floating point>	standard deviations	<floating point>	<floating point>	*Plain*	core
standard deviation of <integer>	standard deviations	<floating point>	<integer>	*Plain*	core
standard firewall profile type	standard firewall profile types	<firewall profile type>	<world>	*PlainGlobal*	win
standard profile of <firewall policy>	standard profiles	<firewall profile>	<firewall policy>	*Plain*	win
standby state	standby states	<power state>	<world>	*PlainGlobal*	win
start boundary of <task trigger>	start boundaries	<time>	<task trigger>	*Plain*	win
start date of <license>	start dates	<time>	<license>	*Plain*	win
start in pathname of <file shortcut>	start in pathnames	<string>	<file shortcut>	*Plain*	win
start of <substring>	starts	<string position>	<substring>	*Plain*	core
start of <time range>	starts	<time>	<time range>	*Plain*	core
start type of <service>	start types	<string>	<service>	*Plain*	win
start when available of <task settings>	start when availables	<boolean>	<task settings>	*Plain*	win
state of <monitor power interval>	states	<power state>	<monitor power interval>	*Plain*	win
state of <service>	states	<string>	<service>	*Plain*	win
state of <system power interval>	states	<power state>	<system power interval>	*Plain*	win
status of <action>	statuss	<string>	<action>	*Plain*	win
status of <active device>	statuss	<integer>	<active device>	*Plain*	win
status of <connection>	statuses	<connection status>	<connection>	*Plain*	win
status of <network adapter>	statuses	<integer>	<network adapter>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
stepping of <processor>	steppings	<integer>	<processor>	*Plain*	win
stop at duration end of <task repetition pattern>	stop at duration ends	<boolean>	<task repetition pattern>	*Plain*	win
stop existing instance of <task settings>	stop existing instances	<boolean>	<task settings>	*Plain*	win
stop on idle end of <task idle settings>	stop on idle ends	<boolean>	<task idle settings>	*Plain*	win
stop when going on battery of <task settings>	stop when going on batteries	<boolean>	<task settings>	*Plain*	win
string <string>	strings	<string>	<world>	*NamedGlobal*	core
string <string> of <smbios structure>	strings	<string>	<smbios structure>	*Named*	win
string value <integer> of <wmi select>	string values	<string>	<wmi select>	*Numbered*	win
string value <string> of <smbios structure>	string values	<smbios value>	<smbios structure>	*Named*	win
string value of <wmi select>	string values	<string>	<wmi select>	*Plain*	win
string version of <application usage summary instance>	string versions	<string>	<application usage summary instance>	*Plain*	win
strong <string> of <html>	strongs	<html>	<html>	*Named*	core
strong <string> of <string>	strongs	<html>	<string>	*Named*	core
strong of <html>	strongs	<html>	<html>	*Plain*	core
strong of <string>	strongs	<html>	<string>	*Plain*	core
structure <string> of <smbios>	structures	<smbios structure>	<smbios>	*Named*	win
structure of <smbios value>	structures	<smbios structure>	<smbios value>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
structure of <smbios>	structures	<smbios structure>	<smbios>	*Plain*	win
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
subcategory of <audit policy category>	subcategories	<audit policy subcategory>	<audit policy category>	*Plain*	win
subject of <email task action>	subjects	<string>	<email task action>	*Plain*	win
subnet address of <network adapter interface>	subnet addresses	<ipv4or6 address>	<network adapter interface>	*Plain*	win
subnet address of <network adapter>	subnet addresses	<ipv4 address>	<network adapter>	*Plain*	win
subnet address of <network address list>	subnet addresses	<ipv4 address>	<network address list>	*Plain*	win
subnet address of <network ip interface>	subnet addresses	<ipv4 address>	<network ip interface>	*Plain*	win
subnet mask of <network adapter interface>	subnet masks	<ipv4or6 address>	<network adapter interface>	*Plain*	win
subnet mask of <network adapter>	subnet masks	<ipv4 address>	<network adapter>	*Plain*	win
subnet mask of <network address list>	subnet masks	<ipv4 address>	<network address list>	*Plain*	win
subnet mask of <network ip interface>	subnet masks	<ipv4 address>	<network ip interface>	*Plain*	win
subscribe time of <site>	subscribe times	<time>	<site>	*Plain*	win
subscription of <event task trigger>	subscriptions	<string>	<event task trigger>	*Plain*	win
substring <(integer,	substrings	<substring>	<string>	*Index<(integer,*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
integer)> of <string>				*integer)>*	
substring <string> of <string>	substrings	<substring>	<string>	*Named*	core
substring after <string> of <string>	substrings after	<substring>	<string>	*Named*	core
substring before <string> of <string>	substrings before	<substring>	<string>	*Named*	core
substring between <string> of <string>	substrings between	<substring>	<string>	*Named*	core
substring separated by <string> of <string>	substrings separated by	<substring>	<string>	*Named*	core
suite mask of <operating system>	suite masks	<operating system suite mask>	<operating system>	*Plain*	win
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
synchronize permission of <access control entry>	synchronize permissions	<boolean>	<access control entry>	*Plain*	win
system category of <audit policy>	system categories	<audit policy category>	<audit policy>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
system event log	system event logs	<event log>	<world>	*PlainGlobal*	win
system file <string>	system files	<file>	<world>	*NamedGlobal*	win
system group	system groups	<security account>	<world>	*PlainGlobal*	win
system ini device file	system ini device files	<file>	<world>	*PlainGlobal*	win
system ini device file <string>	system ini device files	<file>	<world>	*NamedGlobal*	win
system interval of <power history>	system intervals	<system power interval>	<power history>	*Plain*	win
system language	system languages	<string>	<world>	*PlainGlobal*	win
system locale	system locales	<language>	<world>	*PlainGlobal*	win
system of <filesystem object>	systems	<boolean>	<filesystem object>	*Plain*	win
system policy of <audit policy subcategory>	system policies	<audit policy information>	<audit policy subcategory>	*Plain*	win
system ui language	system ui languages	<language>	<world>	*PlainGlobal*	win
system wow64 folder	system wow64 folders	<folder>	<world>	*PlainGlobal*	win
system x32 folder	system x32 folders	<folder>	<world>	*PlainGlobal*	win
system x64 folder	system x64 folders	<folder>	<world>	*PlainGlobal*	win
table <string> of <html>	tables	<html>	<html>	*Named*	core
table <string> of <string>	tables	<html>	<string>	*Named*	core
table of <html>	tables	<html>	<html>	*Plain*	core
table of <string>	tables	<html>	<string>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
target ip address of <port mapping>	target ip addresses	<ipv4 address>	<port mapping>	*Plain*	win
target ipv4or6 address of <port mapping>	target ipv4or6 addresses	<ipv4or6 address>	<port mapping>	*Plain*	win
target name of <port mapping>	target names	<string>	<port mapping>	*Plain*	win
task action type <integer>	task action types	<task action type>	<world>	*NumberedGlobal*	win
task folder <string>	task folders	<task folder>	<world>	*NamedGlobal*	win
task folder of <task folder>	task folders	<task folder>	<task folder>	*Plain*	win
task trigger type <integer>	task trigger types	<task trigger type>	<world>	*NumberedGlobal*	win
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
tcp	tcps	<internet protocol>	<world>	*PlainGlobal*	win
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 duplicate account flag of <user>	temporary duplicate account flags	<boolean>	<user>	*Plain*	win
temporary of <filesystem object>	temporarys	<boolean>	<filesystem object>	*Plain*	win
terminal bit <operating system suite mask>	terminal bits	<boolean>	<world>	*Index<operating system suite*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
				mask>Global	
terminal server user group	terminal server user groups	<security account>	<world>	*PlainGlobal*	win
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
thead of <html>	theads	<html>	<html>	*Plain*	core
thead of <string>	theads	<html>	<string>	*Plain*	core
thursday	thursdays	<day of week>	<world>	*PlainGlobal*	core
time <string>	times	<time>	<world>	*NamedGlobal*	core
time <time zone> of <time>	times	<time of day with time zone>	<time>	*Index<time zone>*	core
time generated of <event log record>	times generated	<time>	<event log record>	*Plain*	win
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 task trigger type	time task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
time value <integer> of	time values	<time>	<wmi select>	*Numbered*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<wmi select>					
time value of <wmi select>	time values	<time>	<wmi select>	*Plain*	win
time written of <event log record>	times written	<time>	<event log record>	*Plain*	win
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 <show message task action>	titles	<string>	<show message task action>	*Plain*	win
title of <string>	titles	<html>	<string>	*Plain*	core
to of <email task action>	tos	<string>	<email task action>	*Plain*	win
total duration of <application usage summary instance>	total durations	<time interval>	<application usage summary instance>	*Plain*	win
total duration of <application usage summary>	total durations	<time interval>	<application usage summary>	*Plain*	win
total processor core count	total processor core counts	<integer>	<world>	*PlainGlobal*	win
total run count of <application usage summary instance>	total run counts	<integer>	<application usage summary instance>	*Plain*	win
total run count of <application usage summary>	total run counts	<integer>	<application usage summary>	*Plain*	win
total size of <download storage folder>	total sizes	<integer>	<download storage folder>	*Plain*	win
total space of <drive>	total spaces	<integer>	<drive>	*Plain*	win

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
traverse permission of <access control entry>	traverse permissions	<boolean>	<access control entry>	*Plain*	win
trigger of <task definition>	triggers	<task trigger>	<task definition>	*Plain*	win
trigger string of <scheduled task>	trigger strings	<string>	<scheduled task>	*Plain*	win
true	trues	<boolean>	<world>	*PlainGlobal*	core
trustee of <access control entry>	trustees	<security identifier>	<access control entry>	*Plain*	win
trustee type of <access control entry>	trustee types	<integer>	<access control entry>	*Plain*	win
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
tunnel of <network adapter>	tunnels	<boolean>	<network adapter>	*Plain*	win
tuple string item <integer> of <string>	tuple string items	<string>	<string>	*Numbered*	core
tuple string item of <string>	tuple string items	<string>	<string>	*Plain*	core
two digit hour of <time of day with time zone>	two digit hours	<string>	<time of day with time zone>	*Plain*	core
two digit hour of <time of day>	two digit hours	<string>	<time of day>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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 <distinguished name component>	types	<string>	<distinguished name component>	*Plain*	core
type of <drive>	types	<string>	<drive>	*Plain*	win
type of <firewall profile>	types	<firewall profile type>	<firewall profile>	*Plain*	win
type of <firewall service>	types	<firewall service type>	<firewall service>	*Plain*	win
type of <license>	types	<string>	<license>	*Plain*	win
type of <metabase value>	types	<metabase type>	<metabase value>	*Plain*	win
type of <network adapter>	types	<integer>	<network adapter>	*Plain*	win
type of <network share>	types	<integer>	<network share>	*Plain*	win
type of <processor>	types	<integer>	<processor>	*Plain*	win
type of <registry key value>	types	<registry key value type>	<registry key value>	*Plain*	win
type of <site>	types	<string>	<site>	*Plain*	win
type of <smbios structure>	types	<integer>	<smbios structure>	*Plain*	win
type of <smbios value>	types	<string>	<smbios value>	*Plain*	win
type of <task action>	types	<task action type>	<task action>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
type of <task trigger>	types	<task trigger type>	<task trigger>	*Plain*	win
type of <wmi select>	types	<integer>	<wmi select>	*Plain*	win
udp	udps	<internet protocol>	<world>	*PlainGlobal*	win
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
underflow of <floating point>	underflows	<boolean>	<floating point>	*Plain*	core
unicast responses to multicast broadcast disabled of <firewall profile>	unicast responses to multicast broadcast disableds	<boolean>	<firewall profile>	*Plain*	win
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unique value of <hertz>	unique values	<hertz with multiplicity>	<hertz>	*Plain*	core
unique value of <integer>	unique values	<integer with multiplicity>	<integer>	*Plain*	core
unique value of <ipv4 address>	unique values	<ipv4 address with multiplicity>	<ipv4 address>	*Plain*	core
unique value of <ipv4or6 address>	unique values	<ipv4or6 address with multiplicity>	<ipv4or6 address>	*Plain*	core
unique value of <ipv6 address>	unique values	<ipv6 address with multiplicity>	<ipv6 address>	*Plain*	core
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unique value of <version>	unique values	<version with multiplicity>	<version>	*Plain*	core
unique value of <year>	unique values	<year with multiplicity>	<year>	*Plain*	core
universal time <string>	universal times	<time>	<world>	*NamedGlobal*	core
universal time zone	universal time zones	<time zone>	<world>	*PlainGlobal*	core
unix of <operating system>	unixes	<boolean>	<operating system>	*Plain*	win
unknown state of <running task>	unknown states	<boolean>	<running task>	*Plain*	win
unknown state of <scheduled task>	unknown states	<boolean>	<scheduled task>	*Plain*	win
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*	win
up of <network adapter>	ups	<boolean>	<network adapter>	*Plain*	win
up of <network ip interface>	ups	<boolean>	<network ip interface>	*Plain*	win
upload progress of <client>	upload progresses	<string>	<client>	*Plain*	win
upnp firewall service type	upnp firewall service types	<firewall service type>	<world>	*PlainGlobal*	win
ups of <power level>	upss	<boolean>	<power level>	*Plain*	win
uptime of <operating	uptimes	<time interval>	<operating	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
system>			system>		
uri of <task registration info>	uris	<string>	<task registration info>	*Plain*	win
url of <site>	urls	<string>	<site>	*Plain*	win
use count of <network share>	use counts	<integer>	<network share>	*Plain*	win
use limit of <network share>	use limits	<integer>	<network share>	*Plain*	win
user	users	<user>	<world>	*PlainGlobal*	win
user <string>	users	<user>	<world>	*NamedGlobal*	win
user comment of <user>	user comments	<string>	<user>	*Plain*	win
user id of <logon task trigger>	user ids	<string>	<logon task trigger>	*Plain*	win
user id of <session state change task trigger>	user ids	<string>	<session state change task trigger>	*Plain*	win
user id of <task principal>	user ids	<string>	<task principal>	*Plain*	win
user id of <user>	user ids	<integer>	<user>	*Plain*	win
user interval of <activity history>	user intervals	<system power interval>	<activity history>	*Plain*	win
user language	user languages	<string>	<world>	*PlainGlobal*	win
user locale	user locales	<language>	<world>	*PlainGlobal*	win
user object count of <process>	user object counts	<integer>	<process>	*Plain*	win
user of <logged on user>	users	<user>	<logged on user>	*Plain*	win
user of <process>	users	<security identifier>	<process>	*Plain*	win
user privilege of <user>	user privileges	<boolean>	<user>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
user sid of <event log record>	user sids	<security identifier>	<event log record>	*Plain*	win
user time of <process>	user times	<time interval>	<process>	*Plain*	win
user type of <metabase value>	user types	<metabase user type>	<metabase value>	*Plain*	win
user ui language	user ui languages	<language>	<world>	*PlainGlobal*	win
usual name of <property>	usual names	<string>	<property>	*Plain*	core
v1 compatibility of <task settings>	v1 compatibilities	<boolean>	<task settings>	*Plain*	win
v2 compatibility of <task settings>	v2 compatibilities	<boolean>	<task settings>	*Plain*	win
value <string> of <file version block>	values	<string>	<file version block>	*Named*	win
value <string> of <registry key>	values	<registry key value>	<registry key>	*Named*	win
value <string> of <smbios structure>	values	<smbios value>	<smbios structure>	*Named*	win
value of <distinguished name component>	values	<string>	<distinguished name component>	*Plain*	core
value of <environment variable>	values	<string>	<environment variable>	*Plain*	win
value of <fixlet_header>	values	<string>	<fixlet_header>	*Plain*	win
value of <metabase key>	values	<metabase value>	<metabase key>	*Plain*	win
value of <registry key>	values	<registry key value>	<registry key>	*Plain*	win
value of <setting>	values	<string>	<setting>	*Plain*	win
value of <smbios structure>	values	<smbios value>	<smbios structure>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
value of <task named value pair>	values	<string>	<task named value pair>	*Plain*	win
value query of <event task trigger>	value queries	<task named value pair>	<event task trigger>	*Plain*	win
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*	win
variable of <environment>	variables	<environment variable>	<environment>	*Plain*	win
variable of <file>	variables	<string>	<file>	*Plain*	win
vendor name of <processor>	vendor names	<string>	<processor>	*Plain*	win
version <string>	versions	<version>	<world>	*NamedGlobal*	core
version block <integer> of <file>	version blocks	<file version block>	<file>	*Numbered*	win
version block <string> of <file>	version blocks	<file version block>	<file>	*Named*	win
version block of <file>	version blocks	<file version block>	<file>	*Plain*	win
version of <application usage summary instance>	versions	<version>	<application usage summary instance>	*Plain*	win
version of <bios>	versions	<string>	<bios>	*Plain*	win
version of <current relay>	versions	<version>	<current relay>	*Plain*	win
version of <file>	versions	<version>	<file>	*Plain*	win
version of <operating system>	versions	<version>	<operating system>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
version of <service>	versions	<version>	<service>	*Plain*	win
version of <site>	versions	<integer>	<site>	*Plain*	win
version of <task registration info>	versions	<string>	<task registration info>	*Plain*	win
version string <string> of <module>	version strings	<string>	<module>	*Named*	core
volatile attribute of <metabase value>	volatile attributes	<boolean>	<metabase value>	*Plain*	win
wait timeout of <task idle settings>	wait timeouts	<time interval>	<task idle settings>	*Plain*	win
waiting for download of <action>	waiting for downloads	<boolean>	<action>	*Plain*	win
wake on lan subnet cidr string	wake on lan subnet cidr strings	<string>	<world>	*PlainGlobal*	win
wake to run of <task settings>	wake to runs	<boolean>	<task settings>	*Plain*	win
wakeonlan enabled of <network adapter>	wakeonlan enableds	<boolean>	<network adapter>	*Plain*	win
warning event log event type	warning event log event types	<event log event type>	<world>	*PlainGlobal*	win
wednesday	wednesdays	<day of week>	<world>	*PlainGlobal*	core
week	weeks	<time interval>	<world>	*PlainGlobal*	core
weekly task trigger type	weekly task trigger types	<task trigger type>	<world>	*PlainGlobal*	win
weeks interval of <weekly task trigger>	weeks intervals	<time interval>	<weekly task trigger>	*Plain*	win
weight of <selected server>	weights	<integer>	<selected server>	*Plain*	win
well known account <integer>	well known accounts	<security account>	<world>	*NumberedGlobal*	win
win32 exit code of	win32 exit	<integer>	<service>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
<service>	codes				
win32 running service	win32 running services	<service>	<world>	*PlainGlobal*	win
win32 service	win32 services	<service>	<world>	*PlainGlobal*	win
win32 type of <service>	win32 types	<boolean>	<service>	*Plain*	win
windows display time <string>	windows display times	<time>	<world>	*NamedGlobal*	core
windows file <string>	windows files	<file>	<world>	*NamedGlobal*	win
windows folder	windows folders	<folder>	<world>	*PlainGlobal*	win
windows of <operating system>	windowses	<boolean>	<operating system>	*Plain*	win
windows server count of <bes product>	windows server counts	<integer>	<bes product>	*Plain*	win
wins enabled of <network adapter>	wins enableds	<boolean>	<network adapter>	*Plain*	win
winsock2 supported of <network>	winsock2 supporteds	<boolean>	<network>	*Plain*	win
wmi	wmis	<wmi>	<world>	*PlainGlobal*	win
wmi <string>	wmis	<wmi>	<world>	*NamedGlobal*	win
working directory of <exec task action>	working directories	<string>	<exec task action>	*Plain*	win
working set size of <process>	working set sizes	<integer>	<process>	*Plain*	win
workstation count of <bes product>	workstation counts	<integer>	<bes product>	*Plain*	win
workstation trust account flag of <user>	workstation trust account flags	<boolean>	<user>	*Plain*	win
wow64 of <process>	wow64s	<boolean>	<process>	*Plain*	win
write attributes permission of <access control entry>	write attributes permissions	<boolean>	<access control entry>	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
write dac permission of <access control entry>	write dac permissions	<boolean>	<access control entry>	*Plain*	win
write extended attributes permission of <access control entry>	write extended attributes permissions	<boolean>	<access control entry>	*Plain*	win
write owner permission of <access control entry>	write owner permissions	<boolean>	<access control entry>	*Plain*	win
write permission of <access control entry>	write permissions	<boolean>	<access control entry>	*Plain*	win
write permission of <network share>	write permissions	<boolean>	<network share>	*Plain*	win
x32 application <string>	x32 applications	<application>	<world>	*NamedGlobal*	win
x32 file <string>	x32 files	<file>	<world>	*NamedGlobal*	win
x32 folder <string>	x32 folders	<folder>	<world>	*NamedGlobal*	win
x32 registry	x32 registries	<registry>	<world>	*PlainGlobal*	win
x64 application <string>	x64 applications	<application>	<world>	*NamedGlobal*	win
x64 file <string>	x64 files	<file>	<world>	*NamedGlobal*	win
x64 folder <string>	x64 folders	<folder>	<world>	*NamedGlobal*	win
x64 of <operating system>	x64s	<boolean>	<operating system>	*Plain*	win
x64 registry	x64 registries	<registry>	<world>	*PlainGlobal*	win
x64 variable <string> of <environment>	x64 variables	<environment variable>	<environment>	*Named*	win
x64 variable of <environment>	x64 variables	<environment variable>	<environment>	*Plain*	win
xml document of <file>	xml documents	<xml dom document>	<file>	*Plain*	win
xml document of <string>	xml documents	<xml dom document>	<string>	*Plain*	core
xml of <scheduled task>	xmls	<string>	<scheduled	*Plain*	win

Key Phrase	Plural	Creates a	From a	*Form*	Ref
			task>		
xml of <task definition>	xmls	<string>	<task definition>	*Plain*	win
xml of <task registration info>	xmls	<string>	<task registration info>	*Plain*	win
xml of <task settings>	xmls	<string>	<task settings>	*Plain*	win
xpath <(string, string)> of <xml dom node>	xpaths	<xml dom node>	<xml dom node>	*Index<(string, string)>*	core
xpath <string> of <xml dom node>	xpaths	<xml dom node>	<xml dom node>	*Named*	core
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>
<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>
<discretionary access control list> as string	<string>	<discretionary access control list>
<environment variable> as string	<string>	<environment variable>
<file content> as lowercase	<file content>	<file content>
<file content> as uppercase	<file content>	<file content>
<file> as string	<string>	<file>

Key Phrase	Creates a	From a
<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>
<format> as string	<string>	<format>
<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>
<ipv4or6 address> as compressed string	<string>	<ipv4or6 address>

Key Phrase	Creates a	From a
<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>
<ipv6 address> as string with leading zeros	<string>	<ipv6 address>

Key Phrase	Creates a	From a
<ipv6 address> as string with leading zeros with zone index	<string>	<ipv6 address>
<ipv6 address> as string with zone index	<string>	<ipv6 address>
<language> as string	<string>	<language>
<local group member> as string	<string>	<local group member>
<metabase identifier> as integer	<integer>	<metabase identifier>
<metabase identifier> as string	<string>	<metabase identifier>
<metabase type> as integer	<integer>	<metabase type>
<metabase type> as string	<string>	<metabase type>
<metabase user type> as integer	<integer>	<metabase user type>
<metabase user type> as string	<string>	<metabase user type>
<metabase value> as integer	<integer>	<metabase value>
<metabase value> as string	<string>	<metabase value>
<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>

Key Phrase	Creates a	From a
<primary language> as string	<string>	<primary language>
<property> as string	<string>	<property>
<registry key value type> as string	<string>	<registry key value type>
<registry key value> as application	<application>	<registry key value>
<registry key value> as file	<file>	<registry key value>
<registry key value> as folder	<folder>	<registry key value>
<registry key value> as integer	<integer>	<registry key value>
<registry key value> as string	<string>	<registry key value>
<registry key value> as system file	<file>	<registry key value>
<registry key value> as time	<time>	<registry key value>
<rope> as string	<string>	<rope>
<security descriptor> as string	<string>	<security descriptor>
<security identifier> as string	<string>	<security identifier>
<service> as string	<string>	<service>
<setting> as string	<string>	<setting>
<site version list> as string	<string>	<site version list>
<smbios value> as string	<string>	<smbios value>
<string> as boolean	<boolean>	<string>
<string> as date	<date>	<string>
<string> as day_of_month	<day of month>	<string>

Key Phrase	Creates a	From a
<string> as day_of_week	<day of week>	<string>
<string> as floating point	<floating point>	<string>
<string> as hexadecimal	<string>	<string>
<string> as html	<html>	<string>
<string> as integer	<integer>	<string>
<string> as ipv4or6 address	<ipv4or6 address>	<string>
<string> as left trimmed string	<string>	<string>
<string> as local time	<time>	<string>
<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>

Key Phrase	Creates a	From a
<string> as windows display time	<time>	<string>
<string> as year	<year>	<string>
<string> as zoned time_of_day	<time of day with time zone>	<string>
<system access control list> as string	<string>	<system access control list>
<task action> as com handler task action	<com handler task action>	<task action>
<task action> as email task action	<email task action>	<task action>
<task action> as exec task action	<exec task action>	<task action>
<task action> as show message task action	<show message task action>	<task action>
<task trigger> as boot task trigger	<boot task trigger>	<task trigger>
<task trigger> as daily task trigger	<daily task trigger>	<task trigger>
<task trigger> as event task trigger	<event task trigger>	<task trigger>
<task trigger> as idle task trigger	<idle task trigger>	<task trigger>
<task trigger> as logon task trigger	<logon task trigger>	<task trigger>
<task trigger> as monthly task trigger	<monthly task trigger>	<task trigger>
<task trigger> as monthlydow task trigger	<monthlydow task trigger>	<task trigger>
<task trigger> as registration task trigger	<registration task trigger>	<task trigger>
<task trigger> as session state change task trigger	<session state change task trigger>	<task trigger>
<task trigger> as time task trigger	<time task trigger>	<task trigger>
<task trigger> as weekly task trigger	<weekly task trigger>	<task trigger>

Key Phrase	Creates a	From a
<time interval> as string	<string>	<time interval>
<time of day with time zone> as string	<string>	<time of day with time zone>
<time of day> as string	<string>	<time of day>
<time range> as string	<string>	<time range>
<time zone> as string	<string>	<time zone>
<time> as local string	<string>	<time>
<time> as string	<string>	<time>
<time> as universal string	<string>	<time>
<type> as string	<string>	<type>
<unary operator> as string	<string>	<unary operator>
<undefined> as string	<string>	<undefined>
<version> as string	<string>	<version>
<version> as version	<version>	<version>
<wmi object> as string	<string>	<wmi object>
<wmi select> as string	<string>	<wmi select>
<xml dom node> as text	<string>	<xml dom node>
<xml dom node> as xml	<string>	<xml dom node>
<year> as integer	<integer>	<year>
<year> as string	<string>	<year>

Part Three

Resources

Processors

On Windows machines, including mobile devices, the Vendor Name, Family, Type, Model, Extended Family, Extended Model and stepping are calculated using the CPUID instruction. The results depend upon the processor and the vendor of the processor. The Inspectors return values based upon the Intel specification for the CPUID instruction. Other vendors or older processors may behave differently. An attempt is made to identify the Family and Family name for processors that do not support the CPUID instruction. You can depend upon the vendor name to distinguish the different vendors except that early versions of the 80486 from AMD are completely indistinguishable from an Intel processor. In this case "AmbiguousCPU" is returned for the vendor name. A complete list of bit values returned by the feature masks property is available in the Intel documentation. These can be found online at the Intel web site.

The speed is measured using a timed sequence of instructions. The speed returned may differ from the expected amount by a couple of MHz.

The CPUID instruction is executed with 1 in the EAX register to compute:

Stepping	Bits 0-3
Model	Bits 4-7
Family	Bits 8-11
Type	Bits 12-13
Extended Model	Bits 16-19
Extended Family	Bits 20-23

Numeric values returned for family of processor and string values returned by family name of processor are computed using the table below. For an unidentified family name, the “brand string” is returned, if available.

Vendor Name	Family Name	Family	Model	Extended Family
GenuineIntel	8086	0		
	80286	2		
	80386	3		
	80486	4		
	Pentium	5		
	Pentium Pro	6	0-2	
	Pentium II	6	38418	
	Pentium III	6	7 or greater	
	Pentium 4	15	0	0
AuthenticAMD	486	4		
	K5	5	0-5	
	K6	5	6 or greater	
	Athlon	6	1,2,4	
	Duron	6	3	
CyrixInstead	MediaGX	4		
	6x86	5	2	
	GXm	5	Not 2	
	6x86MX	6		
CentaurHauls	C6	5		

Folders on Windows Devices

On Windows machines, including mobile devices, numeric identifiers can be used to locate a wide range of system folders. INF files are used to install system software components and device drives. INF files contain sections entitled DestinationDirs. This section is used with the corresponding CopyFiles section to specify destination locations for files placed on the system during the install. The Number identifies the directory. The numbers are sometimes called LDIDs and sometimes called DIRIDs. We call them install folders. Below is a table of install folders and the method Tivoli Endpoint Manager uses to calculate the location.

Install folder#	Name	Calculated using
10	LDID_WIN	GetWindowsDirectory()
11	LDID_SYS	GetSystemDirectory()
12	LDID_IOS	GetSystemDirectory() + "\IOSUBSYS"
13	LDID_CMD	GetWindowsDirectory() + "\COMMAND"
14	LDID_CPL	GetPathFromCSIDL(CSIDL_CONTROLS)
15	LDID_PRINT	GetPathFromCSIDL(CSIDL_PRINTERS)
17	LDID_INF	GetWindowsDirectory() + "\INF"
18	LDID_HELP	GetWindowsDirectory() + "\HELP"
19	LDID_WINADMIN	*Registered Setup folder "WinAdminDir"
20	LDID_FONTS	GetPathFromCSIDL(CSIDL_ CSIDL_FONTS)
21	LDID_VIEWERS	GetSystemDirectory() + "\VIEWERS"
22	LDID_VMM32	GetSystemDirectory() + "\VMM32"
23	LDID_COLOR	*Registered Setup folder "ICMPath"
24	LDID_APPS	*Registered Setup folder "AppsDir"
25	LDID_SHARED	*Registered Setup folder "SharedDir"
26	LDID_WINBOOT	*Registered Setup folder "WinBootDir"
27	LDID_MACHINE	*Registered Setup folder "MachineDir"
28	LDID_HOST_WINBOOT	*Registered Setup folder "HostWinBootDir"
29	LDID_BOOT	*Registered Setup folder "BootDir"
30	LDID_BOOT_HOST	*Registered Setup folder "BootHost"
31	LDID_OLD_WINBOOT	*Registered Setup folder "OldWinBootDir"
32	LDID_OLD_WIN	*Registered Setup folder "OldWinDir"
33	LDID_OLD_DOS	*Registered Setup folder "OldDosDir"

*Registered Setup folders are stored in the Windows registry under the key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SETUP
```

An adjustable set of target locations has been added to the Windows Registry under the key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SETUP\VarLDID
```

Each value stored under this key is a string whose name is the VarLDID and whose value contains a path to a folder. For example, if the value named 28701 contains C:\Program Files, then `install folder "28701"` would return a folder corresponding to that location.

DMI Inspectors

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

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

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

SMBIOS Inspectors

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

- `bios_version of bios_information of dmi`

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

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

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

bios_information	
String	vendor
String	bios_version
Word	bios_starting_address_segment
String	bios_release_date
Byte	bios_rom_size
QWord	bios_characteristics
Byte	system_bios_major_release
Byte	system_bios_minor_release
Byte	embedded_controller_firmware_major_release
Byte	embedded_controller_firmware_minor_release
system_information	
String	manufacturer
String	product_name
String	version

String	serial_number
BinaryString	uuid
Byte	wake_up_type
String	sku_number
String	family
base_board_information	
String	manufacturer
String	product
String	version
String	serial_number
String	asset_tag
Byte	feature_flags
String	location_in_chassis
Word	chassis_handle
Byte	board_type
Byte	number_of_contained_object_handles
system_enclosure_or_chassis	
String	manufacturer
Byte	type
String	version
String	serial_number
String	asset_tag_number
Byte	bootup_state
Byte	power_supply_state
Byte	thermal_state
Byte	security_status
DWord	oem_defined
Byte	height
Byte	number_of_power_cords
Byte	contained_element_count
Byte	contained_element_record_length
processor_information	
String	socket_designation
Byte	processor_type
Byte	processor_family
String	processor_manufacturer
QWord	processor_id
String	processor_version
Byte	voltage
Word	external_clock
Word	max_speed
Word	current_speed
Byte	status

Byte	processor_upgrade
Word	l1_cache_handle
Word	l2_cache_handle
Word	l3_cache_handle
String	serial_number
String	asset_tag
String	part_number
Byte	core_count
Byte	core_enabled
Byte	thread_count
Word	processor_characteristics
Word	processor_family_2
memory_controller_information	
Byte	error_detecting_method
Byte	error_correcting_capability
Byte	supported_interleave
Byte	current_interleave
Byte	maximum_memory_module_size
Word	supported_speeds
Word	supported_memory_types
Byte	memory_module_voltage
Byte	number_of_associated_memory_slots
memory_module_information	
String	socket_designation
Byte	bank_connections
Byte	current_speed
Word	current_memory_type
Byte	installed_size
Byte	enabled_size
Byte	error_status
cache_information	
String	socket_designation
Word	cache_configuration
Word	maximum_cache_size
Word	installed_size
Word	supported_sram_type
Word	current_sram_type
Byte	cache_speed
Byte	error_correction_type
Byte	system_cache_type
Byte	associativity
port_connector_information	
String	internal_reference_designator

Byte	internal_connector_type
String	external_reference_designator
Byte	external_connector_type
Byte	port_type
system_slots	
String	slot_designation
Byte	slot_type
Byte	slot_data_bus_width
Byte	current_usage
Byte	slot_length
Word	slot_id
Byte	slot_characteristics_1
Byte	slot_characteristics_2
Word	segment_group_number
Byte	bus_number
Byte	device_function_number
on_board_devices_information	
Byte	device_type
String	description_string
oem_strings	
Byte	count
MultipleString	string
system_configuration_options	
Byte	count
MultipleString	string
bios_language_information	
Byte	installable_languages
Byte	flags
BinaryString	reserved
String	current_language
group_associations	
String	group_name
Byte	item_type
Word	item_handle
physical_memory_array	
Byte	location
Byte	use
Byte	memory_error_correction
DWord	maximum_capacity
Word	memory_error_information_handle
Word	number_of_memory_devices
memory_device	
Word	memory_array_handle

Word	memory_error_information_handle
Word	total_width
Word	data_width
Word	size
Byte	form_factor
Byte	device_set
String	device_locator
String	bank_locator
Byte	memory_type
Word	type_detail
Word	speed
String	manufacturer
String	serial_number
String	asset_tag
String	part_number
Byte	attributes
b32_bit_memory_error_information	
Byte	error_type
Byte	error_granularity
Byte	error_operation
DWord	vendor_syndrome
DWord	memory_array_error_address
DWord	device_error_address
DWord	error_resolution
memory_array_mapped_address	
DWord	starting_address
DWord	ending_address
Word	memory_array_handle
Byte	partition_width
memory_device_mapped_address	
DWord	starting_address
DWord	ending_address
Word	memory_device_handle
Word	memory_array_mapped_address_handle
Byte	partition_row_position
Byte	interleave_position
Byte	interleaved_data_depth
built_in_pointing_device	
Byte	type
Byte	interface
Byte	number_of_buttons
portable_battery	
String	location

String	manufacturer
String	manufacture_date
String	serial_number
String	device_name
Byte	device_chemistry
Word	design_capacity
Word	design_voltage
String	sbds_version_number
Byte	maximum_error_in_battery_data
Word	sbds_serial_number
Word	sbds_manufacture_date
String	sbds_device_chemistry
Byte	design_capacity_multiplier
DWord	oem_specific
system_reset	
Byte	capabilities
Word	reset_count
Word	reset_limit
Word	timer_interval
Word	timeout
hardware_security	
Byte	hardware_security_settings
system_power_controls	
Byte	next_scheduled_power_on_month
Byte	next_scheduled_power_on_day_of_month
Byte	next_scheduled_power_on_hour
Byte	next_scheduled_power_on_minute
Byte	next_scheduled_power_on_second
voltage_probe	
String	description
Byte	location_and_status
Word	maximum_value
Word	minimum_value
Word	resolution
Word	tolerance
Word	accuracy
DWord	oem_defined
Word	nominal_value
cooling_device	
Word	temperature_probe_handle
Byte	device_type_and_status
Byte	cooling_unit_group
DWord	oem_defined

Word	nominal_speed
temperature_probe	
String	description
Byte	location_and_status
Word	maximum_value
Word	minimum_value
Word	resolution
Word	tolerance
Word	accuracy
DWord	oem_defined
Word	nominal_value
electrical_current_probe	
String	description
Byte	location_and_status
Word	maximum_value
Word	minimum_value
Word	resolution
Word	tolerance
Word	accuracy
DWord	oem_defined
Word	nominal_value
out_of_band_remote_access	
String	manufacturer_name
Byte	connections
system_boot_information	
BinaryString	reserved
b64_bit_memory_error_information	
Byte	error_type
Byte	error_granularity
Byte	error_operation
DWord	vendor_syndrome
QWord	memory_array_error_address
QWord	device_error_address
DWord	error_resolution
management_device	
String	description
Byte	type
DWord	address
Byte	address_type
management_device_component	
String	description
Word	management_device_handle
Word	component_handle

Word	threshold_handle
management_device_threshold_data	
Word	lower_threshold_non_critical
Word	upper_threshold_non_critical
Word	lower_threshold_critical
Word	upper_threshold_critical
Word	lower_threshold_non_recoverable
Word	upper_threshold_non_recoverable
memory_channel	
Byte	channel_type
Byte	maximum_channel_load
Byte	memory_device_count
Byte	memory_device_load
Word	memory_device_handle
ipmi_device_information	
Byte	interface_type
Byte	ipmi_specification_revision
Byte	i2c_slave_address
Byte	nv_storage_device_address
QWord	base_address
system_power_supply	
Byte	power_unit_group
String	location
String	device_name
String	manufacturer
String	serial_number
String	asset_tag_number
String	model_part_number
String	revision_level
Word	max_power_capacity
Word	power_supply_characteristics
Word	input_voltage_probe_handle
Word	cooling_device_handle
Word	input_current_probe_handle
additional_information	
Byte	number_of_additional_information_entries
onboard_devices_extended_information	
Byte	reference_designation
Byte	device_type
Byte	device_type_instance
Word	segment_group_number
Byte	bus_number
Byte	device_function_number

Part Four

Notices

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

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

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

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

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

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

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

provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Corporation

2Z4A/101

11400 Burnet Road

Austin, TX 78758 U.S.A.

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

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

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

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

COPYRIGHT LICENSE:

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

TRADEMARKS:

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

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

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

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

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

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

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

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

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

Part Five

Index

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y