Tivoli. *Endpoint Manager*

Version 8.1

Windows Mobile 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 Mobile Client Inspector Guide*** 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 Mobile (WM). Thousands of Inspectors have been created to expose the inner workings of Windows Mobile devices, including the hardware, file system and software.

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

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

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

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

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

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

Audience

This guide is for IT managers, product support groups and other people who want to use Inspectors to write Fixlet messages and Tasks for Windows Mobile-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 Mobile devices. Contact your Tivoli Endpoint Manager marketing representative for information about Inspector Guides for other operating systems, including Windows, Solaris, HPUX, Macintosh, and a variety of Linux operating systems.

Conventions Used in this manual

This document makes use of the following conventions and nomenclature:

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

Examples

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

- `concatenation of "light" & "year"`
- Returns "lightyear"

Versions

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

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

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

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

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

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

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

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

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

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

There are exceptions due to platform variations and introduction dates. 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

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
computer id	*PlainGlobal*	<integer>	This is a unique integer assigned to the computer by the BES system. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
computer name	*PlainGlobal*	<string>	Returns a string corresponding to the name of the computer as it appears on the network. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
custom site subscription effective date <string>	*NamedGlobal*	<time>	Returns the date the custom site (specified by <string>) was last subscribed or unsubscribed. It is used internally by BES to manage custom site subscriptions. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
dns name	*PlainGlobal*	<string>	Returns the DNS name of the computer. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
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
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
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
oem code page	*PlainGlobal*	<integer>	Returns an integer value of the Windows API GetOEMCP. Win, WM

Key Phrase	*Form*	Return Type	Description
parameter <string>	*NamedGlobal*	<string>	This Inspector is a synonym for the parameter <string> of <action>. It looks up the value of the action parameter specified by <string>. This is used in conjunction with the parameter set command. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
pending login	*PlainGlobal*	<boolean>	Installers may leave values in the registry that the operating system will execute when the next user logs in. Pending login can detect these registry entries. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pending restart	*PlainGlobal*	<boolean>	Returns TRUE if the operating system indicates that a restart needs to occur. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pending restart <string>	*NamedGlobal*	<boolean>	Immediately after issuing a command like 'Action requires restart "PatchGroupX"', the expression 'Pending restart "PatchGroupX"' will be true until the next restart. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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
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
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

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

Examples

- `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.

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

Key Phrase	*Form*	Description
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
application of <registry>	*Plain*	<application>	Iterates through the properly installed applications. See application. Win, WM
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

- `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
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
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
name of <registry key>	*Plain*	<string>	Returns the name of the key as a string. Win, WM
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.

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

- `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, etc. It also includes the keywords needed to identify and compare version information of files and patches. For more information on special Windows folders, see the Resources section at the end of this guide.

Filesystem

The filesystem object can be used to inspect various aspects of mounted file systems, including the format of the file system. Here are some of the possible format types:

- affs
- ext, ext2, ext2_old
- hpfs
- iso
- minix, minix_30, minix2, minix2_30
- msdos
- ncp
- nfs
- proc
- smb
- xenix
- sysv4, sysv2
- coh
- ufs
- xia

Creation Methods

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

Key Phrase	*Form*	Description
drive <string>	*NamedGlobal*	Returns the drive associated with the pathname specified by <string>. • On Windows computers, this returns a <drive> object. • On Macintosh computers, this returns a <volume> object. • On *nix computers, this returns a <filesystem> object. Lin, Sol, HPUX, AIX, , WM, Ubu

Examples

- `names of drives`

  Returns the names of the mounted drives.

Filesystem Object

Properties

Key Phrase	*Form*	Return Type	Description
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

Key Phrase	*Form*	Return Type	Description
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
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
descendant of <folder>	*Plain*	Returns a list of all the descendant files of the specified folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
download file <string>	*NamedGlobal*	This inspector is available in relevance subsitution action processing. It returns a file object with the given name from the named folder or the download folder. This is equivalent to 'file "name" of download folder'. The file should exist or the result will not exist. Win:7.2, Lin:7.2, Sol:7.2, HPUX:7.2, AIX:7.2, Mac:7.2, WM, Ubu
file <string>	*NamedGlobal*	Returns a filesystem object corresponding to the full pathname provided in <string>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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

Key Phrase	*Form*	Description
file of <folder>	*Plain*	Iterates through the files of a folder. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
find file <string> of <folder>	*Named*	Creates an object corresponding to the files of the folder that that match the wildcard <string> provided. A wildcard string uses an asterisk to stand for any number of characters (incuding zero), and a question mark to stand for exactly one character. Thus A??.txt would match All.txt and AXE.txt but not all.txt or a.txt. Win, Lin, Sol, HPUX, AIX, Mac:8.0, WM, Ubu
masthead of <site>	*Plain*	A copy of the masthead is maintained with the site data. This inspector returns a file object for the copy. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
system file <string>	*NamedGlobal*	Creates the file objects corresponding to the named file within the system folder. Win, WM
windows file <string>	*NamedGlobal*	Returns a file object corresponding to the relative pathname (within the Windows folder) provided. See file. Win, WM

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
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
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

Key Phrase	*Form*	Return Type	Description
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
sha1 of <file>	*Plain*	<string>	Returns the sha1 checksum of the file hex encoded as a 40 character long string. CAUTION: This Inspector maintains a handle to the specified file, so during its operation it may block any other applications that attempt to open the file. Inspectors open files as with both read and write sharing, so apps that open with compatibleaccess will not block. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
size of <file>	*Plain*	<integer>	Returns the size in bytes of a file. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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

Key Phrase	*Form*	Return Type	Description
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.

- `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.

- `file version of application "iexplore.exe" of the registry < "4"`
  - Test for older version of IE -- returns TRUE is version is less than 4.

- `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.

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

Key Phrase	*Form*	Description
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
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
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

Examples

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

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

- `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
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

Key Phrase	*Form*	Description
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
download folder	*PlainGlobal*	This inspector is available in relevance subsitution action processing. When the action is active, this inspector returns a folder object of __Download\, otherwise it returns a folder object of __Global\sitename\actionid\named. This inspector is designed for the prefetch process of action execution. • Macintosh Note: Prior to version 7.2, this Inspector referred to the system download folder on the Macintosh. That Inspector is referred to as ISS Download as of version 7.2. WM
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
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

Key Phrase	*Form*	Description
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

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

Key Phrase	*Form*	Return Type	Description
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
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

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

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

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.

- `install folder 11`

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

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 Content

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

Creation Methods

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

Properties

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

Operators

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

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

Examples

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

  Returns TRUE if either of the strings "emachines" or "eMachines" is found in the file.

Version

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

Creation Methods

Key Phrase	*Form*	Description
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
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 <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 <wince_web_browser>	*Plain*	Creates an object corresponding to the version of the specified WinCE web browser, typically some version of Internet Explorer. WM

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.

- `version of default web browser`
- Returns a version, such as 'iexplore.exe 4.1', corresponding to the current web browser.

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
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

Key Phrase	*Form*	Description
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

- `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

Key Phrase	*Form*	Description
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
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

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

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.

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
build number high of <operating system>	*Plain*	<integer>	Numeric representation of the most significant 16 bits of the build number. Win, WM

Key Phrase	*Form*	Return Type	Description
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
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
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, etcetera. 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

Key Phrase	*Form*	Return Type	Description
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
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

Key Phrase	*Form*	Return Type	Description
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
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
x64 of <operating system>	*Plain*	<boolean>	Returns TRUE if the current operating system is 64-bits. Win, WM

Examples

- `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, etcetera.

- `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, etcetera.

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

Processor

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

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
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, etcetera. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
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
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.

- `bit 18 of feature mask of main processor`
  - Returns TRUE if the processor ID feature is enabled on this processor.

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

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

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

Ram

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
available of <ram>	*Plain*	<integer>	Returns the total amount of RAM (in bytes) currently available on the Windows Mobile device. This is the same as for for the Windows client. WM
load of <ram>	*Plain*	<integer>	Returns the amount of memory being used on the Windows Mobile device as a percentage. 0 = no memory used, 100 = all memory used. This is the same as for the Windows client. WM

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

- `available of ram`

  Returns the number of bytes of RAM, such as 72351744.

- `load of ram`

  Returns a number from 0 to 100 indicating the percentage of RAM currently being used in the Windows Mobile device.

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

  Returns the size of RAM in megabytes.

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
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

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

Properties

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

Key Phrase	*Form*	Return Type	Description
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
setting <string> of <site>	*Named*	<setting>	Returns the setting whose name matches the string provided from the Fixlet site settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
setting of <site>	*Plain*	<setting>	Returns one or more settings from the site settings. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site tag of <site>	*Plain*	<string>	Returns the last component of the specified site's url, eg. ‘actionsite’, ‘enteprisesecurity’, etcetera. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
site version list of <site>	*Plain*	<site version list>	Returns the last gathered site version list (manyversion) of the specified site. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.0, Mac:7.1, WM, Ubu
subscribe time of <site>	*Plain*	<time>	Returns the time that the current machine began subscribing to the site. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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

Key Phrase	*Form*	Return Type	Description
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
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
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

Fixlet_header

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

Creation Methods

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

Properties

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

Examples

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

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

Client Objects

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

Client

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

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
administrator <string> of <client>	*Named*	<setting>	If the administrator named in the <string> is enabled on the given <client> computer, this property returns a setting with the given name and the value 'allow.' For instance, if the name of the administrator is joe_admin, then the client would return a setting object with the name 'joe_admin' and a value of 'allow'. Casting this as a string would return 'joe_admin=allow'. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
administrator of <client>	*Plain*	<setting>	Returns one or more settings each representing an administrator of the client. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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

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

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

Examples

- `names of settings of site "actionsite"`

  Returns the names of all the settings of the site named "actionsite".

Selected Server

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

Creation Methods

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

Properties

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

Key Phrase	*Form*	Return Type	Description
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

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
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
running of <application usage summary>	*Plain*	<boolean>	Returns TRUE if the specified application is currently running. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
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

License Objects

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

License

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
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

Key Phrase	*Form*	Return Type	Description
encryption certificate of <license>	*Plain*	<x509 certificate>	Provides the encryption certificate that is currently active and which will be used by clients to encrypt reports. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
evaluation of <license>	*Plain*	<boolean>	Returns TRUE if client is running an evaluation license. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
expiration date of <license>	*Plain*	<time>	Returns date when license will expire. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
expiration state of <license>	*Plain*	<string>	Returns a string, one of "Unrestricted", "Grace" or "Restricted". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
fips mode of <license>	*Plain*	<boolean>	Returns TRUE if the BES action masthead specifies that applications (the client, console, or web reports, depending on the context) in the deployment should operate in FIPS 140-2 compliant mode. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
gather url of <license>	*Plain*	<string>	Returns the gather URL for the deployment's main Action site as specified in the deployment masthead. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
maximum seat count of <license>	*Plain*	<integer>	Returns maximum seat count allowed by the license. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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
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

Key Phrase	*Form*	Return Type	Description
start date of <license>	*Plain*	<time>	The starting date specified for the BigFix license. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Examples

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

Windows Mobile Device Objects

These Inspectors retrieve information about Windows Mobile devices, such as smart phones which are being used as BES Clients.

Phone

These Inspectors return various pieces of information about the Windows mobile phone, including serial numbers, the owner's name, email address and more.

Creation Methods

Key Phrase	*Form*	Description
phone	*PlainGlobal*	Creates an object corresponding to the Client phone. This can be used to query other custome aspects of the device. WM

Properties

Key Phrase	*Form*	Return Type	Description
identifier of <phone>	*Plain*	<string>	Returns a string corresponding to the identifier of the specified phone. WM
manufacturer of <phone>	*Plain*	<string>	Returns a string corresponding to the manufacturer of the specified phone. WM
model of <phone>	*Plain*	<string>	Returns a string corresponding to the model of the specified phone. WM

Key Phrase	*Form*	Return Type	Description
operator name of <phone>	*Plain*	<string>	Returns a string corresponding to the operator name of the specified phone. WM
owner address of <phone>	*Plain*	<string>	Returns a string corresponding to the address of the owner of the specified phone. WM
owner company of <phone>	*Plain*	<string>	Returns a string corresponding to the company name of the owner of the specified phone. WM
owner email of <phone>	*Plain*	<string>	Returns a string corresponding to the email address of the owner of the specified phone. WM
owner name of <phone>	*Plain*	<string>	Returns a string corresponding to the name of the owner of the specified phone. WM
owner notes of <phone>	*Plain*	<string>	Returns a string containing the owner notes of the specified phone. WM
phone number of <phone>	*Plain*	<string>	Returns a string containing the phone number of the specified phone. WM
rated speed of <phone>	*Plain*	<hertz>	Returns a string corresponding to the rated speed of the specified phone. WM
revision of <phone>	*Plain*	<string>	Returns a string identifying the revision of the specified phone. WM
roaming status of <phone>	*Plain*	<string>	Returns a string identifying the roaming status of the specified phone. WM
serial number of <phone>	*Plain*	<string>	Returns a string corresponding to the serial number of the specified phone. WM
signal strength of <phone>	*Plain*	<integer>	Returns a string corresponding to the signal strength of the specified phone as a percentage. WM

Key Phrase	*Form*	Return Type	Description
subscriber number of <phone>	*Plain*	<string>	Returns a string corresponding to the subscriber number of the specified phone. WM
type of <phone>	*Plain*	<string>	Returns a string identifying the type of the specified phone. WM

Examples

- `identifier of phone`

  Returns a string identifying the brand of the phone, such as: 'Samsung Blackjack'.

- `manufacturer of phone`

  Returns a string containg the name of the phone's manufacturer, such as 'SAMSUNG Electronics'.

- `model of phone`

  Returns a string containing the model name of the phone, such as 'SAMSUNG MITs'.

- `operator name of phone`

  Returns a string containing the name of the Windows Mobile device service provider, such as 'ATT'.

- `owner address of phone`

  Returns a string containing the street address of the phone's owner, such as: '12345 Some Street, Denver, Colorado'.

- `owner company of phone`

  Returns a string containing the name of the company that owns the phone, such as 'ACME Inc'.

- `owner email of phone`

  Returns a string containing the email address of the phone's owner, such as: 'john.smith@mail.com'.

- `owner name of phone`

  Returns a string containing the name of the phone's owner, such as: 'John Smith'.

- `owner notes of phone`

  Returns a string containing notes written by the phone's owner, such as: 'These are my important notes'.

- `phone number of phone`

  Returns a string containing the phone number of the Windows Mobile device.

- `rated speed of phone`

  Returns a hertz object indicating the phone's rated speed, such as '419430400 hertz'.

- `revision of phone`

  Returns a string containing revision of the phone, such as 'i607UCGB4'.

- `roaming status of phone`

  Returns a string identifying the roaming status of the phone, such as 'Unavailable'.

- `serial number of phone`

  Returns a string containing the serial number of the phone, such as '35546001011618/1 04'.

- `signal strength of phone`

  Returns an integer between 0 and 100, indicating the strength of the phone connection as a percentage.

- `subscriber number of phone`

  Returns a string containing the phone number of the phone's subscriber.

- `type of phone`

  Returns a string identifying the type of the phone, such as: 'SmartPhone'.

Oma Csp

These are Windows Mobile Inspectors for Open Mobile Alliance (OMA) Configuration Service Providers (CSPs). They allow you to Inspect various features and security settings on a Windows Mobile device. Some of these Inspectors return XML strings that can be directly used in Actions to configure or provision Windows Mobile devices.

Creation Methods

Key Phrase	*Form*	Description
oma csp	*PlainGlobal*	Creates a global object corresponding to the OMA CSP on the current Windows Mobile device. WM
oma csp <(string, string)>	*Index<(string, string)>Global*	Creates a global object corresponding the Open Mobile Alliance (OMA) Configuration Service Provider (CSP) parameter for the two specified oma csps parameter strings. WM
oma csp <(string, string, string)>	*Index<(string, string, string)>Global*	Returns the value of the Open Mobile Alliance (OMA) Configuration Service Provider (CSP) parameter for the three specified oma csps parameter strings. WM

Key Phrase	*Form*	Description
oma csp <(string, string, string, string)>	*Index<(string, string, string, string)>Global*	Returns the value of the Open Mobile Alliance (OMA) Configuration Service Provider (CSP) parameter for the four specified oma csps parameter strings. WM
oma csp <string>	*NamedGlobal*	Returns the value of the Open Mobile Alliance (OMA) Configuration Service Provider (CSP) parameter for the specified oma csps parameter string. This form of the oma csp Inspector takes a string that can contain any number of comma-separated parameters. WM

Properties

Key Phrase	*Form*	Return Type	Description
autorun policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current autorun policy from the SecurityPolicy Configuration Service Provider. • 0 indicates that applications are allowed to run automatically from the Multimedia Card when inserted. • 1 indicates that applications are restricted from autorunning. WM
block incoming calls of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'block incoming calls' status from the SecurityPolicy Configuration Service Provider. WM
block outgoing calls of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'block outgoing calls' status from the SecurityPolicy Configuration Service Provider. WM
bluetooth mode of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current bluetooth mode from the SecurityPolicy Configuration Service Provider. WM

Key Phrase	*Form*	Return Type	Description
bluetooth policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current bluetooth policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether a Bluetooth-enabled device will allow other devices to perform a search on the device. Possible values are: • 0 blocks other devices from searching. • 1 allows other devices to search. WM
boolean <string> of <oma csp>	*Named*	<boolean>	Returns the result of the specified OMA CSP query as a boolean value. WM
call waiting enabled of <oma csp>	*Plain*	<boolean>	Returns the current 'call waiting enabled' status (TRUE or FALSE) from the SecurityPolicy Configuration Service Provider. WM
construct xml <string> of <oma csp>	*Named*	<string>	Returns an XML snippet to query an OMA CSP based on the parameters passed in <string>. WM
desktop quick connect authentication policy of <oma csp>	*Plain*	<integer>	Returns the current 'desktop quick connect authentication' policy from the SecurityPolicy Configuration Service Provider. This setting indicates how device authentication will be handled when connecting to the desktop. Possible values are: • 0 User must authenticate the device upon connection, if the device lock is active. • 1 If user chooses quick connect, the desktop will uniquely identify the device and allow it to connect without requiring the user to manually unlock it. WM
drm security policy of <oma csp>	*Plain*	<integer>	Returns a bit-map integer corresponding to the current Digital Rights Management (DRM) security policy from the SecurityPolicy Configuration Service Provider. The given role bit-map indicates which DRM rights messages will be accepted by the DRM engine. WM

Key Phrase	*Form*	Return Type	Description
encrypt removable storage policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'encrypt removable storage' policy from the SecurityPolicy Configuration Service Provider. This setting indicates if the user is allowed to change mobile encryption settings for the removable storage media. Possible values are: • 0 indicates that the user is not allowed to change the encryption settings. • 1 indicates that the user can change the encryption settings. This is the default. WM
fixed dialing enabled of <oma csp>	*Plain*	<boolean>	Returns the current 'fixed dialing enabled' setting (TRUE or FALSE) from the SecurityPolicy Configuration Service Provider. WM
forward all calls enabled of <oma csp>	*Plain*	<boolean>	Returns the current 'forward all calls enabled' setting (TRUE or FALSE) from the SecurityPolicy Configuration Service Provider. WM
forward all calls of <oma csp>	*Plain*	<string>	Returns a string corresponding to the current 'forward all calls' setting from the SecurityPolicy Configuration Service Provider. WM
forward all calls timeout of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'forward all calls timeout' setting from the SecurityPolicy Configuration Service Provider. WM
forward all calls to of <oma csp>	*Plain*	<string>	Returns a string corresponding to the current 'forward all calls to' string from the SecurityPolicy Configuration Service Provider. WM
forward calls enabled when busy of <oma csp>	*Plain*	<boolean>	Returns the current 'forward calls enabled when busy' setting (TRUE or FALSE) from the SecurityPolicy Configuration Service Provider. WM
forward calls enabled when no answer of <oma csp>	*Plain*	<boolean>	Returns the current 'forward calls enabled when no answer' setting (TRUE or FALSE) from the SecurityPolicy Configuration Service Provider. WM

Key Phrase	*Form*	Return Type	Description
forward calls enabled when unavailable of <oma csp>	*Plain*	<boolean>	Returns the current 'forward calls enabled when unavailable' setting (TRUE or FALSE) from the SecurityPolicy Configuration Service Provider. WM
forward calls timeout when busy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'forward calls timeout when busy' setting from the SecurityPolicy Configuration Service Provider. WM
forward calls timeout when no answer of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'forward calls timeout when no answer' setting from the SecurityPolicy Configuration Service Provider. WM
forward calls timeout when unavailable of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'forward calls timeout when unavailable' setting from the SecurityPolicy Configuration Service Provider. WM
forward calls to when busy of <oma csp>	*Plain*	<string>	Returns a string corresponding to the current 'forward calls to when busy' setting from the SecurityPolicy Configuration Service Provider. WM
forward calls to when no answer of <oma csp>	*Plain*	<string>	Returns a string corresponding to the current 'forward calls to when no answer' setting from the SecurityPolicy Configuration Service Provider. WM
forward calls to when unavailable of <oma csp>	*Plain*	<string>	Returns a string corresponding to the current 'forward calls to when unavailable' setting from the SecurityPolicy Configuration Service Provider. WM
forward calls when busy of <oma csp>	*Plain*	<string>	Returns a string corresponding to the current 'forward calls when busy' setting from the SecurityPolicy Configuration Service Provider. WM
forward calls when no answer of <oma csp>	*Plain*	<string>	Returns a string corresponding to the current 'forward calls when no answer' setting from the SecurityPolicy Configuration Service Provider. WM
forward calls when unavailable of <oma csp>	*Plain*	<string>	Returns a string corresponding to the current 'forward calls when unavailable' setting from the SecurityPolicy Configuration Service Provider. WM

Key Phrase	*Form*	Return Type	Description
grant manager policy of <oma csp>	*Plain*	<integer>	Returns an integer bit-mask corresponding to the current 'grant manager' policy from the SecurityPolicy Configuration Service Provider. This setting grants the system administrative privileges held by the role manager to other security roles, without modifying metabase role assignments. The bit-mask describes which roles are granted system administrative privileges. WM
grant user authenticated policy of <oma csp>	*Plain*	<integer>	Returns an integer bit-mask corresponding to the current 'grant user authenticated' policy from the SecurityPolicy Configuration Service Provider. This setting grants privileges held by the User Authenticated role to other security roles without modifying metabase role assignments. The bit-mask describes which roles are granted system administrative privileges. WM
html message policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'html message' policy from the SecurityPolicy Configuration Service Provider. This setting specifies whether message transports will allow HTML messages. • 0 indicates that HTML messages are not allowed. • 1 indicates that HTML messages are allowed. WM
integer <string> of <oma csp>	*Named*	<integer>	Returns the result of the specified OMA CSP query as an integer value. WM
message authentication retry number policy of <oma csp>	*Plain*	<integer>	Returns a one-byte integer corresponding to the current 'message authentication retry number' policy from the SecurityPolicy Configuration Service Provider. This indicates the maximum number of times the user is allowed to try authenticating a Wireless Application Protocol (WAP) PIN-signed message. The default value is 3 for WM. Possible values are 1 through 256. WM

Key Phrase	*Form*	Return Type	Description
message encryption negotiation policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'message encryption negotiation' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether the Inbox application can negotiate the encryption algorithm in the case that a recipient's certificate doesn't support the specified encryption algorithm. Possible values are: • 0 doesn't allow negotiation. • 1 allows negotiation to a strong algorithm. • 2 allows negotiation to any algorithm. WM
network pin prompt policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'network personal identification number (PIN) prompt' policy from the SecurityPolicy Configuration Service Provider. This setting is used when an over-the-air (OTA) OMA Client Provisioning message is only signed with a network PIN. This setting indicates whether or not the user will be prompted to accept the device setting changes. Possible values are: • 0 indicates that the device will prompt the user. • 1 indicates that the user is not prompted. This is the default. WM
network type of <oma csp>	*Plain*	<string>	Returns the current 'network type' policy from the SecurityPolicy Configuration Service Provider. WM
obex enabled of <oma csp>	*Plain*	<boolean>	Returns the current 'obex enabled' policy from the SecurityPolicy Configuration Service Provider. This indicates whether or not the phone can exchange binary objects, either by infrared or bluetooth. WM
oma cp network pin policy of <oma csp>	*Plain*	<integer>	Returns the current 'oma cp network personal identification number (PIN)' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether the OMA network PIN-signed message will be accepted. The message's role bit-mask and the policy's role mask are ANDed together. If the result is non-zero, then the message will be accepted. WM

Key Phrase	*Form*	Return Type	Description
oma cp user network pin policy of <oma csp>	*Plain*	<integer>	Returns the current 'oma cp user network personal identification number (PIN)' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether the OMA user network PIN-signed message will be accepted. The message's role bit-mask and the policy's role mask are ANDed together. If the result is non-zero, then the message will be accepted. WM
oma cp user pin policy of <oma csp>	*Plain*	<integer>	Returns the current 'oma cp user personal identification number (PIN)' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether the OMA-user PIN or user MAC-signed message will be accepted. The message's role bit-mask and the policy's role mask are ANDed together. If the result is non-zero, then the message will be accepted. WM
ota provisioning policy of <oma csp>	*Plain*	<integer>	Returns an integer bit-mask corresponding to the current 'ota provisioning' policy from the SecurityPolicy Configuration Service Provider. This setting indicates which provisioning messages are accepted by the configuration host based on the role bit-maps assigned to the messages. This policy restricts the provisioning messages that come from the Push Router. A specified role bit-mask indicates system administrative privileges are provided to the given mask. WM
password required policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'password required' policy from the SecurityPolicy Configuration Service Provider. Possible values are: • 0 indicates that a password is required. This is the default. • A value other than 0 indicates that a password is not required. WM

Key Phrase	*Form*	Return Type	Description
privileged applications policy of <oma csp>	*Plain*	<integer>	Returns the current 'privileged applications' policy from the SecurityPolicy Configuration Service Provider. This setting indicates which security model has been implemented on the WM device. Possible values are: • 0 indicates that a two-tier security model is enabled. • 1 indicates that a one-tier security model is enabled. • Any value other than 1 is treated as 0. WM
process xml query <string> of <oma csp>	*Named*	<string>	This Inspector will take the value passed in <string> and then ask the system to process it. To use it, the value provided must be a valid OMA CSP XML query that is not trying to set a value (only queries are allowed). A typical use is to take the results of the 'construct xml query' Inspector and pass it in as the query string. WM
rapi policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current RAPI (Remote API) policy from the SecurityPolicy Configuration Service Provider. • 0 indicates that the ActiveSync service is shut down and RAPI calls are rejected. • 1 indicates that full access to ActiveSync is provided and RAPI calls are allowed without restrictions. • 2 indicates that access to ActiveSync is restricted to the User-Authenticated role. RAPI calls are then checked against this role mask before being granted. WM
security policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'security policy' policy from the SecurityPolicy Configuration Service Provider. WM
send caller id of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'send caller id' policy from the SecurityPolicy Configuration Service Provider. WM

Key Phrase	*Form*	Return Type	Description
service indication message policy of <oma csp>	*Plain*	<integer>	Returns an integer bit-mask corresponding to the current 'service indication message' policy from the SecurityPolicy Configuration Service Provider. An SI message is sent to WM 6 Standard to notify users of new services and service updates. This setting indicates whether SI messages are accepted in the form of a role bit-mask. WM
service loading message policy of <oma csp>	*Plain*	<integer>	Returns an integer bit-mask corresponding to the current 'service loading message' policy from the SecurityPolicy Configuration Service Provider. An SL message downloads new services to the WM device. This setting indicates whether SL messages are accepted in the form of a role bit-mask. WM
sharepoint access policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'sharepoint access' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether OMA SharePoint or UNC access is enabled through ActiveSync protocol to fetch documents. Possible values are: • 0 doesn't allow SharePoint or UNC file access. • 1 allows OMA to fetch documents on a corporate SharePoint site or UNC. WM
sl security policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'sl security' policy from the SecurityPolicy Configuration Service Provider. This setting indicates that the operator can override https to use http, or wsps to use wsp. Possible values are: • 0 use https or wsps. • 1 use http or wsp. This is the default value. WM

Key Phrase	*Form*	Return Type	Description
smime encryption algorithm policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'smime encryption algorithm' policy from the SecurityPolicy Configuration Service Provider. This setting indicates which algorithm is used to encrypt a message. Possible values are: • 0 specifies the default algorithm. • 1 is an invalid value. • 2 specifies the triple DES algorithm. • 3 specifies the DES algorithm. • 4 specifies the RC2 128-bit algorithm. • 5 specifies the RC2 64-bit algorithm. • 6 specifies the RC2 40-bit algorithm. WM
smime encryption policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'smime encryption' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether the Inbox application will send all messages encrypted. • 0 all messages must be encrypted. • 1 encrypting messages is optional. WM
smime signing algorithm policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'smime signing algorithm' policy from the SecurityPolicy Configuration Service Provider. This setting indicates which algorithm is used to sign a message. Possible values are: • 0 specifies the default algorithm. • 1 is an invalid value. • 2 specifies the SHA algorithm. • 3 specifies the MD5 algorithm. WM
smime signing policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'smime signing' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether the Inbox application will send all messaged signed. • 0 all messages must be signed. • 1 signing messages is optional. WM

Key Phrase	*Form*	Return Type	Description
software certificates policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'software certificates' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether software certificates can be used to sign outgoing messages. Possible values are: • 0 indicates that software certificates cannot be used to sign messages. • 1 indicates that software certificates can be used to sign messages. This is the default. WM
storage card encryption of <oma csp>	*Plain*	<boolean>	Returns the current 'storage card encryption' policy from the SecurityPolicy Configuration Service Provider. WM
string <string> of <oma csp>	*Named*	<string>	Returns the result of the specified OMA CSP query as a string value. WM
timezone of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current timezone policy from the SecurityPolicy Configuration Service Provider. WM
trusted provisioning server policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'trusted provisioning server' policy from the SecurityPolicy Configuration Service Provider. Possible values are: • 0 indicates that assigning TPS role assignment is disabled. • 1 indicates TPS role assignment is enabled and the TPS role can be assigned to mobile operators. This is the WM default. WM
trusted wap proxy policy of <oma csp>	*Plain*	<integer>	Returns an integer bit-map corresponding to the current 'trusted wap proxy' policy from the SecurityPolicy Configuration Service Provider. This setting indicates the level of permissions required to create, modify or delete a trusted proxy. The security roles that can have Trusted WAP Proxy level permissions are returned as a bit-mask. WM

Key Phrase	*Form*	Return Type	Description
unauthenticated message policy of <oma csp>	*Plain*	<integer>	Returns an integer bit-mask corresponding to the current 'unauthenticated message' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether to accept unsigned WAP messages processed by the default security provider in the Push Router, based on their origin. The message source must match one of the security roles specified by this policy. This setting indicates whether unauthenticated messages are accepted in the form of a role bit-mask. WM
unsigned applications policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'unsigned applications' policy from the SecurityPolicy Configuration Service Provider. The possible values are: • 0 indicates that unsigned apps are not allowed to run on the device. • 1 indicates that unsigned apps are allowed to run on the device. This is the default for WM. • Any value other than 1 is treated as 0. WM
unsigned cabs policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'unsigned CABS' policy from the SecurityPolicy Configuration Service Provider. This indicates whether unsigned .cab files can be installed on the device. Possible values are: • 0 is equivalent to having none of the role mask bits set and indicates that no unsigned .cab files can be installed. • A specified role bit-mask indicates accepted unsigned .cab files are installed with the given role mask. WM
unsigned prompt policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'unsigned prompt' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether a user must be prompted to accept or reject unsigned .exe, theme, .dll or .cab files. Possible values are: • 0 indicates that the user will be prompted. This is the WM default. • 1 indicates that the user will not be prompted. • Any value other than 1 is treated as 0. WM

Key Phrase	*Form*	Return Type	Description
unsigned themes policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'unsigned themes' policy from the SecurityPolicy Configuration Service Provider. Possible values are: • 0 is equivalent to having none of the role-mask bits set, and indicates that no unsigned Theme files can be installed. • A specified role bit-mask indicates accepted unsigned Theme files are installed with the given role mask. WM
value <string> of <oma csp>	*Named*	<string>	Returns the result of the specified OMA CSP query as a string value. WM
wsp push policy of <oma csp>	*Plain*	<integer>	Returns an integer corresponding to the current 'Wireless Session Protocol (WSP) push' policy from the SecurityPolicy Configuration Service Provider. This setting indicates whether WSP notifications from the WAP stack are routed. Possible values are: • 0 indicates that routing of WSP notifications is not allowed. • 1 indicates that Routing is allowed. This is the WM default. WM

Examples

- `value "URL" of oma csp ("BrowseFavorite", "Southridge Video Store")`

  Returns a URL corresponding to the specified browser favorite site, such as: 'http://www.southridgevideo.com'.

- `value "TAPI_FORWARD_ADDRESS" of oma csp ("Tapi","Busy","Voice")`

  Returns a string containing the forwarding telephone number for the specified parameters, such as '5551212'.

- `integer "TAPI_BARRING_OUT" of oma csp "Tapi"`

  Returns an integer corresponding to the value of the chosen parameter.

- `process xml query (construct xml "TAPI_FORWARD_ENABLED" of oma csp "Tapi,Busy,Voice") of oma csp`

  Processes the given xml query and returns the resulting xml, in this case: <wap-provisioningdoc> <characteristic type="Tapi"> <characteristic type="Busy"> <characteristic type="Voice"> <parm-query

name="TAPI_FORWARD_ENABLED"/> </characteristic> </characteristic> </characteristic> </wap-provisioningdoc>.

- `value "TAPI_FORWARD_ADDRESS" of oma csp "Tapi,Busy,Voice"`

  Returns a string containing the forwarding telephone number for the specified parameters.

- `autorun policy of oma csp`

  Returns 0 or 1, depending on whether apps are allowed to run automatically or not. This is the same as: integer "2" of oma csp "SecurityPolicy".

- `block incoming calls of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "TAPI_BARRING_IN" of oma csp "Tapi".

- `block outgoing calls of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "TAPI_BARRING_OUT" of oma csp "Tapi".

- `bluetooth mode of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "BtMode" of oma csp "Bluetooth".

- `bluetooth policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4135" of oma csp "SecurityPolicy".

- `boolean "TAPI_FORWARD_ENABLED" of oma csp "Tapi,Busy,Voice"`

  Returns the boolean value of the specified OMA CSP query.

- `call waiting enabled of oma csp`

  Returns TRUE if call waiting is enabled.

- `construct xml "TAPI_FORWARD_ENABLED" of oma csp "Tapi,Busy,Voice"`

  Returns a snippet of XML like the following: <wap-provisioningdoc> <characteristic type="Tapi"> <characteristic type="Busy"> <characteristic type="Voice"> <parm-query name="TAPI_FORWARD_ENABLED"/> </characteristic> </characteristic> </characteristic> </wap-provisioningdoc>.

- `construct xml "TAPI_FORWARD_ADDRESS" of oma csp "Tapi,Busy,Voice"`

  Returns a snippet of XML like the following: <wap-provisioningdoc> <characteristic type="Tapi"> <characteristic type="Busy"> <characteristic type="Voice"> <parm-query name="TAPI_FORWARD_ADDRESS" /> </characteristic> </characteristic> </characteristic> </wap-provisioningdoc>.

- `desktop quick connect authentication policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4146" of oma csp "SecurityPolicy".

- `drm security policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4129" of oma csp "SecurityPolicy".

- `encrypt removable storage policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4134" of oma csp "SecurityPolicy".

- `fixed dialing enabled of oma csp`

  Returns TRUE if fixed dialing is enabled. This is the same as: boolean "TAPI_FIXEDDIAL_ENABLED" of oma csp "Tapi".

- `forward all calls enabled of oma csp`

  Returns TRUE if call forwarding is enabled. This is the same as: boolean "TAPI_FORWARD_ENABLED" of oma csp ("Tapi","Unconditional","Voice").

- `forward all calls of oma csp`

  Returns a string of the form: 'Forward calls to 5551212 when Unconditional after 20 seconds'.

- `forward all calls timeout of oma csp`

  Returns an integer such as 20. This is the same as: integer "TAPI_FORWARD_TIMEOUT" of oma csp ("Tapi","Unconditional","Voice").

- `forward all calls to of oma csp`

  Returns a telephone number as a string. This is the same as: string "TAPI_FORWARD_ADDRESS" of oma csp ("Tapi","Unconditional","Voice").

- `forward calls enabled when busy of oma csp`

  Returns TRUE if the call forwarding is enabled when busy.

- `forward calls enabled when no answer of <oma csp`

  Returns TRUE if the call forwarding is enabled when there is no answer.

- `forward calls enabled when unavailable of oma csp`

  Returns TRUE if the call forwarding is enabled when the user is unreachable.

- `forward calls timeout when busy of oma csp`

  Returns an integer corresponding to the specified timeout.

- `forward calls timeout when no answer of oma csp`

  Returns an integer corresponding to the specified timeout.

- `forward calls timeout when unavailable of oma csp`

  Returns an integer corresponding to the specified timeout.

- `forward calls to when busy of oma csp`

  Returns a string corresponding to the forwarding phone number when busy.

- `forward calls to when no answer of oma csp`

  Returns a string corresponding to the forwarding phone number when there is no answer.

- `forward calls to when unavailable of oma csp`

  Returns a string corresponding to the forwarding phone number when the user is unreachable.

- `forward calls when busy of oma csp`

  Returns a string of the form: 'Forward calls to 5551212 when Busy after 5 seconds'.

- `forward calls when no answer of oma csp`

  Returns a string of the form: 'Forward calls to 5551212 when No-Reply after 25 seconds'.

- `forward calls when unavailable of oma csp`

  Returns a string of the form: 'Forward calls to 5551212 when Not-Reachable after 5 seconds'.

- `grant manager policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4119" of oma csp "SecurityPolicy".

- `grant user authenticated policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4120" of oma csp "SecurityPolicy".

- `html message policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4136" of oma csp "SecurityPolicy".

- `integer "TAPI_BARRING_OUT" of oma csp "Tapi"`

  Returns an integer corresponding to the current status of the specified Security Policy string constant.

- `message authentication retry number policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4105" of oma csp "SecurityPolicy".

- `message encryption negotiation policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4144" of oma csp "SecurityPolicy".

- `network pin prompt policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4132" of oma csp "SecurityPolicy".

- `network type of oma csp`

  Returns a network type as a string, such as 'ATT'. This is the same as: string "TAPI_FORWARD_ADDRESS" of oma csp ("Tapi","Unconditional","Voice").

- `obex enabled of oma csp`

  Returns TRUE if object exchange protocol is enabled. This is the same as: boolean "IsEnabled" of oma csp ("Obex","HKLM\Software\Microsoft\Obex").

- `oma cp network pin policy of oma csp`

  Returns an integer bit-mask corresponding to the current status of the given Security Policy, such as '3200'. This is the same as: integer "4141" of oma csp "SecurityPolicy".

- `oma cp user network pin policy of oma csp`

  Returns an integer bit-mask corresponding to the current status of the given Security Policy, such as '3200'. This is the same as: integer "4143" of oma csp "SecurityPolicy".

- `oma cp user pin policy of oma csp`

  Returns an integer bit-mask corresponding to the current status of the given Security Policy, such as '3200'. This is the same as: integer "4142" of oma csp "SecurityPolicy".

- `ota provisioning policy of oma csp`

  Returns an integer bit-mask corresponding to the current status of the given Security Policy, such as '3728'. This is the same as: integer "4111" of oma csp "SecurityPolicy".

- `password required policy of oma csp`

  Returns a 0 if a password is required. This is the same as: integer "4131" of oma csp "SecurityPolicy".

- `privileged applications policy of oma csp`

  Returns a 1 if a one-tier security model is enabled, otherwise, a two-tier model is used. This is the same as: integer "4123" of oma csp "SecurityPolicy".

- `process xml query (construct xml "TAPI_FORWARD_ADDRESS" of oma csp "Tapi,Busy,Voice") of oma csp`

  Returns a string containing an XML snippet such as: '<wap-provisioningdoc> <characteristic type="Tapi"> <characteristic type="Busy"> <characteristic type="Voice"> <parm-query name="TAPI_FORWARD_ADDRESS" value="5551212" /> </characteristic> </characteristic> </characteristic> </wap-provisioningdoc>'.

- `rapi policy of oma csp`

  Returns an integer (0-2) corresponding to the current status of the given Security Policy, such as '3200'. This is the same as: integer "4097" of oma csp "SecurityPolicy".

- `security policy of oma csp`

  Returns an integer corresponding to the current status of the given Security Policy. This is the same as: integer "4124" of oma csp "SecurityPolicy".

- `send caller id of oma csp`

  Returns the 'send caller id' policy as an integer. This is the same as: integer "TAPI_SEND_CALLID" of oma csp "Tapi".

- `service indication message policy of oma csp`

  Returns an integer bit-mask corresponding to the current status of the given Security Policy. This is the same as: integer "4109" of oma csp "SecurityPolicy".

- `service loading message policy of oma csp`

  Returns an integer bit-mask corresponding to the current status of the given Security Policy. This is the same as: integer "4108" of oma csp "SecurityPolicy".

- `sharepoint access policy of oma csp`

  Returns an integer (0 or 1) corresponding to the current status of the given Security Policy. This is the same as: integer "4145" of oma csp "SecurityPolicy".3073.

- `sl security policy of oma csp`

  Returns an integer (0 or 1) corresponding to the current status of the given Security Policy. This is the same as: integer "4124" of oma csp "SecurityPolicy".

- `smime encryption algorithm policy of oma csp`

  Returns an integer (0-6) corresponding to the current status of the given Security Policy. This is the same as: integer "4140" of oma csp "SecurityPolicy".

- `smime encryption policy of oma csp`

  Returns an integer (0 or 1) corresponding to the current status of the given Security Policy. This is the same as: integer "4138" of oma csp "SecurityPolicy".

- `smime signing algorithm policy of oma csp`

  Returns an integer (0-3) corresponding to the current status of the given Security Policy. This is the same as: integer "4139" of oma csp "SecurityPolicy".

- `smime signing policy of oma csp`

  Returns an integer (0 or 1) corresponding to the current status of the given Security Policy. This is the same as: integer "4137" of oma csp "SecurityPolicy".

- `software certificates policy of oma csp`

  Returns an integer (0 or 1) corresponding to the current status of the given Security Policy. This is the same as: integer "4127" of oma csp "SecurityPolicy".

- `storage card encryption of oma csp`

  Returns a boolean TRUE if the storage card is encrypted.

- `string "TAPI_FORWARD_ADDRESS" of oma csp ("Tapi","Unconditional","Voice")`

  Returns a string containing the forwarding phone number currently set up for the specified parameters.

- `timezone of oma csp`

  Returns an integer corresponding to the time zone set for the phone. This is equivalent to: integer "TimeZone" of oma csp "Clock".

- `trusted provisioning server policy of oma csp`

  Returns an integer (0 or 1) corresponding to the current status of the given Security Policy. This is the same as: integer "4104" of oma csp "SecurityPolicy".

- `trusted wap proxy policy of oma csp`

  Returns an integer bit-map descripting the current trusted wap proxy policy.

- `unauthenticated message policy of oma csp`

  Returns an integer, such as '64', corresponding to the current unauthenticated message policy.

- `unsigned applications policy of oma csp`

  Returns 1 if unsigned apps are allowed to run.

- `unsigned cabs policy of oma csp`

  Returns an integer bit-mask defining the roles for accepting unsigned cab files.

- `unsigned prompt policy of oma csp`

  Returns a 1 if the user is to be prompted before accepting certain unsigned files.

- `unsigned themes policy of oma csp`

  Returns an integer bit-mask defining the roles for accepting unsigned theme files.

- `value "TAPI_FORWARD_ADDRESS" of oma csp ("Tapi","Unconditional","Voice")`

  This phrase returns the forwarding number, such as '5551212', for the given parameters.

- `wsp push policy of oma csp`

  Returns 1 if routing of WSP notifications is allowed.

Wince Network Connection Detail

These Inspectors return detailed information about the Windows Embedded CE network connections on the Windows Mobile device. For more information about these Inspectors, refer to the MSDN article titled CONNMGR_CONNECTION_DETAILED_STATUS. These Inspectors require that the Mobile device be running WinCE .NET 4.2 or later.

Creation Methods

Key Phrase	*Form*	Description
network connection	*PlainGlobal*	Creates a global object corresponding to the WinCE network connection. WM

Properties

Key Phrase	*Form*	Return Type	Description
adapter name of <wince network connection detail>	*Plain*	<string>	Returns a string corresponding to the null-terminated name of the adapter for the given WinCE network connection. If no adapter name is available, the Inspector returns NULL. WM
description of <wince network connection detail>	*Plain*	<string>	Returns a string corresponding to the null-terminated description of the given WinCE network connection. If no adapter name is available, the Inspector returns NULL. WM
destination network of <wince network connection detail>	*Plain*	<string>	Returns a string containing the GUID of the destination network for the specified WinCE connection. WM
flags of <wince network connection detail>	*Plain*	<string>	Returns a string containing one or more connection options for the specified WinCE network connection. These flags include: billed by time, always on, or suspend and resume. The constants for these flags are explained in greater detail in the MSDN article on the Connection Manager Connection Options Constants. WM
ip addresses of <wince network connection detail>	*Plain*	<string>	Returns a string containing the available IP addresses for the specified WinCE network connection. If no addresses are available, this Inspector returns NULL. WM
last connected of <wince network connection detail>	*Plain*	<string>	Returns a string containing the last time that the connection was established for the specified WinCE network connection. WM
secure of <wince network connection detail>	*Plain*	<boolean>	Returns a boolean describing the security level of the current connection for the specified WinCE network. If TRUE, the connection is secure. WM
signal quality of <wince network connection detail>	*Plain*	<integer>	Returns the signal quality of the specified WinCE network connection. This is an integer between 0 and 255, with 255 indicating the best signal quality. WM

Key Phrase	*Form*	Return Type	Description
source network of <wince network connection detail>	*Plain*	<string>	Returns a string containing the GUID of the source network for the specified WinCE connection. WM
status of <wince network connection detail>	*Plain*	<string>	Returns the status of the specified WinCE network connection. This is a string indicating whether the connection is established, suspended, disconnected, waiting, failed or more. These are explained in greater detail in the MSDN article on the Connection Manager Status Constants. WM
type of <wince network connection detail>	*Plain*	<string>	Returns the type of the specified WinCE network connection. This is a string indicating a cellular, NIC, Bluetooth, Unimodem, VPN, Proxy or PC connection. These are explained in greater detail in the MSDN article on the Connection Manager Connection Type Constants. WM

Examples

- `adapter name of network connection "My Wifi"`
  - Returns a string such as "TNETW12511".
- `description of network connection "My Wifi"`
  - Returns a string such as: My Wifi.
- `destination network of network connection "My Wifi Network"`
  - Returns a string such as: IID_DestNetInternet.
- `flags of network connection "My Wifi"`
  - Returns a string such as 'Always On'.
- `ip addresses of network connection "My Wifi"`
  - Returns a string containing the available IP address(es) for the given network, such as 'localhost (192.168.1.104)'.
- `last connected of network connection "My Wifi"`
  - Returns a string containing the last connetion time, such as '7/13/2009 12:35:00 AM'.
- `secure of network connection "My Wifi"`
  - Returns TRUE if the WinCE network connection is secure.

- `signal quality of network connection "My Wifi"`

  Returns an integer between 0 and 255, indicating the strength of the specified network connection.

- `source network of network connection "My Wifi"`

  Returns a string containing the GUID of the source network for the specified connection, such as: 'Unknown'.

- `status of network connection "My Wifi"`

  Returns a string containing the current network connection status, such as: 'Connected'.

- `type of network connection "My Wifi"`

  Returns a string, such as 'Wifi', indicating the network type.

Wince_web_browser

These Inspectors return information about the Windows Embedded CE browser on the specified device. Typically, this is a version of Internet Explorer that has been optimized for operation on the small display of a Windows Mobile device.

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
default web browser	*PlainGlobal*	Creates a global object corresponding to the current default web browser on the Windows CE device. Windows Embedded CE uses IE, which has been optimized for WinCE devices. WM

Properties

Key Phrase	*Form*	Return Type	Description
version of <wince_web_browser>	*Plain*	<version>	Returns the version of the current web browser on the Windows CE device. WM

Examples

- `default web browser`

  Returns a value such as: "iexplore.exe" "" "" "" "".

- `version of default web browser`

  Returns a version, such as 'iexplore.exe 4.1', corresponding to the current web browser.

Base_battery

This is an abstract type from which <battery> and <backup battery> are derived.

Properties

Key Phrase	*Form*	Return Type	Description
full life of <base_battery>	*Plain*	<time interval>	For the specified Windows Mobile battery, this Inspector returns a time interval corresponding to the number of seconds of battery life when at full charge. Base battery is an abstract type that can refer to either the main "battery" or the "backup battery". WM
life of <base_battery>	*Plain*	<time interval>	For the specified Windows Mobile battery, this Inspector returns a time interval corresponding to the number of seconds of battery life remaining. Base battery is an abstract type that can refer to either the main "battery" or the "backup battery". WM
life percent of <base_battery>	*Plain*	<integer>	For the specified Windows Mobile battery, this Inspector returns an integer corresponding to the percentage of full battery charge remaining. This is a value in the range 0 to 100. Base battery is an abstract type that can refer to either the main "battery" or the "backup battery". WM
millivolts of <base_battery>	*Plain*	<integer>	For the specified Windows Mobile battery, this Inspector returns an integer corresponding to the amount of battery voltage in millivolts (mV). This is a value in the range of 0 to 65,535. Base battery is an abstract type that can refer to either the main "battery" or the "backup battery". WM
status of <base_battery>	*Plain*	<string>	Returns a string corresponding to the current status of the battery. This is one of the following: Charging, High, Low, Critical, No battery or Unknown. Base battery is an abstract type that can refer to either the main "battery" or the "backup battery". WM

Examples

- `full life of battery`

  Returns a time interval for the battery life, such as 3:45:00.

- `life of backup battery`

  Returns a time interval denoting the remaining backup battery life, such as '0:04:03'. This is the same return type used for the main battery life.

- `life percent of battery`

  Returns a number from 0 to 100 indicating the percentage of life left in the battery.

- `millivolts of backup battery`

  Returns an integer corresponding to the backup battery voltage (in mV).

- `status of battery`

  Returns a string indicating the current battery status, such as: 'High'. A similar string applies to the backup battery as well.

Battery

These Inspectors return information about the battery in the Windows Mobile device, including items such as the type, charge and lifetime.

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

Creation Methods

Key Phrase	*Form*	Description
battery	*PlainGlobal*	Creates an inspectable object corresponding to the main battery of the Windows Mobile device. WM

Properties

Key Phrase	*Form*	Return Type	Description
ac of <battery>	*Plain*	<string>	Returns a string detailing the AC power status of the specified Windows Mobile device battery. This can include offline, online or backup. For more information, see the MSDN article on SYSTEM_POWER_STATUS_EX. WM
average interval of <battery>	*Plain*	<integer>	Returns an integer corresponding to the time constant in milliseconds (ms) used for integrating the average battery current in milliamps. WM

Key Phrase	*Form*	Return Type	Description
average milliamps of <battery>	*Plain*	<integer>	Returns an integer corresponding to the short-term average current drain of the Windows Mobile device (in milliamps). This number is in the range of 0 to 32,767 when charging and 0 to –32,768 when discharging. WM
chemistry of <battery>	*Plain*	<string>	This Inspector returns a string describing the type of chemistry used by the specified Windows Mobile battery. It can include alkaline, nicad, lithium and others. For details, see the MSDN article on SYSTEM_POWER_STATUS_EX2. WM
milliamps of <battery>	*Plain*	<integer>	Returns an integer corresponding to the instantaneous current drain of the Windows Mobile device (in milliamps). This number is in the range of 0 to 32,767 when charging and 0 to –32,768 when discharging. WM
milliamps per hour of <battery>	*Plain*	<integer>	Returns an integer corresponding to the long-term cumulative average discharge in milliamperes per hour (mA/H). This number can have a value in the range of 0 to –32,768. This value can be reset by charging or changing the batteries. WM
temperature of <battery>	*Plain*	<floating point>	For this specified Windows Mobile device battery, this Inspector returns a floating point number corresponding to the battery temperature in degrees Celsius. It can be in the range of –3,276.8 to 3,276.7 in increments of 0.1 degrees Celsius. WM

Examples

- `battery`

  Returns a value such as: "High" "41" "00:00:00" "00:00:00" "3708" "Offline" "Lithium-ion" "-386" "0" "0" "0" "37".

- `ac of battery`

  Returns a string such as "Online".

- `ac of backup battery`

  Returns a string such as "Online".

- `average interval of battery`

  Returns a number corresponding to the time in milliseconds used to average the battery current.

- `average milliamps of battery`

  Returns a number corresponding to the current drain of the main battery.

- `average milliamps of backup battery`

  Returns a number corresponding to the current drain of the backup battery.

- `chemistry of battery`

  Returns a sting such as: Lithium-ion. These strings are also used to describe the backup battery.

- `milliamps of battery`

  Returns a signed integer, such as -376, corresponding to the battery drain.

- `milliamps per hour of battery`

  Returns a signed integer, such as -53, corresponding to the cumulative average battery drain per hour.

- `temperature of battery`

  Returns a floating point number corresponding to the temperature of the battery.

Backup_battery

These Inspectors return information about the backup battery in the Windows Mobile device, similar to the information obtainable from the battery Inspectors.

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

Creation Methods

Key Phrase	*Form*	Description
backup battery	*PlainGlobal*	Creates an inspectable object corresponding to the backup battery of the Windows Mobile device. The backup battery takes over should the main battery run out of charge. WM

Examples

- `backup battery`

  Returns a value such as: "High" "41" "00:00:00" "00:00:00" "3708" "Offline" "Lithium-ion" "-386" "0" "0" "0" "37".

Gps

These are inspectors for Mobile Windows to interrogate the Global Positioning Service (GPS) device on the phone. The information available, as well as latitude and longitude, includes heading, altitude and more. The majority of the fields in this structure are translations from the fields defined by the National Marine Electronics Association (NMEA).

Creation Methods

Key Phrase	*Form*	Description
gps	*PlainGlobal*	Creates a global object corresponding to the GPS on the Windows Mobile device. WM

Properties

Key Phrase	*Form*	Return Type	Description
altitude of <gps>	*Plain*	<string>	Returns a string containing the altitude (in meters) of the Windows Mobile device, as determined by the onboard GPS. WM
enabled of <gps>	*Plain*	<boolean>	Returns TRUE if the Global Positioning Service (GPS) on the Windows Mobile device is enabled. WM
full status of <gps>	*Plain*	<string>	Returns a string containing the full status of the Windows Mobile device, as determined by the onboard GPS. It is a concatenation of all the inspectable items, with the general form 'feature: {value} units', each separated by a space. The full string looks like 'Name: {name} Status: {ON/OFF} Last sample time: {sample time} Latitude: {latitude} degrees Longitude: {longitude} degrees Heading: {heading} degrees Speed: {speed} knots Altitude: {altitude} m.'. WM
heading of <gps>	*Plain*	<string>	Returns a string containing the heading in degrees (a heading of zero is true north) of the Windows Mobile device, as determined by the onboard GPS. WM
latitude of <gps>	*Plain*	<string>	Returns a string containing the latitude (in degrees) of the Windows Mobile device, as determined by the onboard GPS. Positive numbers indicate the northern latitudes. WM

Key Phrase	*Form*	Return Type	Description
longitude of <gps>	*Plain*	<string>	Returns a string containing the longitude (in degrees) of the Windows Mobile device, as determined by the onboard GPS. Positive numbers indicate east longitudes. WM
name of <gps>	*Plain*	<string>	Returns a string containing the human-readable name of the embedded GPS of the Windows Mobile device. It might, for example, be something like 'Acme GPS Card, version 3.4.'. WM
sample time of <gps>	*Plain*	<time>	Returns a time value containing the current sample time used by the onboard GPS of the Windows Mobile device. WM
speed of <gps>	*Plain*	<string>	Returns a string containing the speed (in knots) of the Windows Mobile device, as determined by the onboard GPS. WM

Examples

- `altitude of gps`
  - Returns a string such as "150.000".

- `enabled of gps`
  - Returns TRUE if the GPS is enabled.

- `full status of gps`
  - Returns a string of the form: 'Name: QualComm GpsOne Card, version 0.0 Status: ON Last sample time: Mon, 13 Jul 2009 12:50:05 -0800 Latitude: 32.99205 degrees Longitude: -117.05468 degrees Heading: Not available Speed: 0.0000 knots Altitude: 150.000 m'.

- `heading of gps`
  - Returns a string containing the heading of the mobile device in degrees, such as '90' for due east.

- `latitude of gps`
  - Returns a string containing latitude of the Win Mobile device, as determined by the GPS, such as '32.99205'.

- `longitude of gps`
  - Returns a string representing the current longitude of the Windows Mobile device, as indicated by the GPS, such as '-117.05468'.

- `name of gps`

  Returns a string containing the model name of the GPS, such as 'QualComm GpsOne Card, version 0.0'.

- `sample time of gps`

  Returns a time object according to the GPS, such as: 'Mon, 13 Jul 2009 12:50:05 -0800'.

- `speed of gps`

  Returns a string indicating the speed of the device in knots, such as: '43.3420'.

Authorization Objects

These inspectors retrieve security and access settings.

Client_cryptography

These Inspectors expose cryptographic properties exclusive to the client.

Creation Methods

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

Properties

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

X509 Certificate

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

Creation Methods

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

User Objects

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

Logged On User

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

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

Creation Methods

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

Properties

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

Key Phrase	*Form*	Return Type	Description
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

Action Objects

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

Action

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

Creation Methods

Key Phrase	*Form*	Description
action	*PlainGlobal*	Creates an action object corresponding to the BigFix Action currently being parsed. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
action <integer>	*NumberedGlobal*	Creates an action object matching the <integer> id. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
active action	*PlainGlobal*	Creates an action object corresponding to the currently executing action. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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
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

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

Examples

- `parameter "action issue date" of action`

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

Networking Objects

This chapter includes the various networking Inspectors.

Network

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
adapter of <network>	*Plain*	<network adapter>	Returns the one or more network adapter objects of the network. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
dns server of <network>	*Plain*	<network address list>	Returns a list of DNS servers used by the local computer. Win, WM
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
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

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

Key Phrase	*Form*	Description
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
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 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

Examples

- `names of ip interfaces of network`

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

- `addresses of ip interfaces of network`

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

- `address of ip interface whose (loopback of it = false) of network = "192.168.127.127"`

  Returns TRUE if the given IP address doesn't have loopback.

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 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

Key Phrase	*Form*	Return Type	Description
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>	*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

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 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
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
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
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

Key Phrase	*Form*	Return Type	Description
maximum transmission unit of <network adapter>	*Plain*	<integer>	The maximum transmission unit (MTU) size, in bytes, of the specified adapter. Win:7.0, WM
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
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
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
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

Key Phrase	*Form*	Return Type	Description
wins enabled of <network adapter>	*Plain*	<boolean>	Returns TRUE if WINS is enabled on the network adapter. Win, WM

Ipv4 Address

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

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

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

Creation Methods

Key Phrase	*Form*	Description
address of <network adapter>	*Plain*	Returns the ip address of the network adapter. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:7.1, WM, Ubu
address of <network 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

Key Phrase	*Form*	Description
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
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

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, 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
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

Key Phrase	*Form*	Description
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

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
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
ac of <battery>	acs	<string>	<battery>	*Plain*	wm
accessed time of <filesystem object>	accessed times	<time>	<filesystem object>	*Plain*	wm
action	actions	<action>	<world>	*PlainGlobal*	wm
action <integer>	actions	<action>	<world>	*NumberedGlobal*	wm
action lock state	action lock states	<action lock state>	<world>	*PlainGlobal*	wm
active action	active actions	<action>	<world>	*PlainGlobal*	wm
active of <action>	actives	<boolean>	<action>	*Plain*	wm
active of <logged on user>	actives	<boolean>	<logged on user>	*Plain*	wm
active start time of <action>	active start times	<time>	<action>	*Plain*	wm
adapter name of <wince network connection detail>	adapter names	<string>	<wince network connection detail>	*Plain*	wm
adapter of <network>	adapters	<network adapter>	<network>	*Plain*	wm
address list of <network adapter>	address lists	<network address list>	<network adapter>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
address of <network adapter>	addresses	<ipv4 address>	<network adapter>	*Plain*	wm
address of <network address list>	addresses	<ipv4 address>	<network address list>	*Plain*	wm
address of <network ip interface>	addresses	<ipv4 address>	<network ip interface>	*Plain*	wm
administrator <string> of <client>	administrators	<setting>	<client>	*Named*	wm
administrator of <client>	administrators	<setting>	<client>	*Plain*	wm
altitude of <gps>	altitudes	<string>	<gps>	*Plain*	wm
ancestor of <filesystem object>	ancestors	<folder>	<filesystem object>	*Plain*	wm
ansi code page	ansi code pages	<integer>	<world>	*PlainGlobal*	wm
apparent registration server time	apparent registration server times	<time>	<world>	*PlainGlobal*	wm
application <string>	applications	<application>	<world>	*NamedGlobal*	wm
application <string> of <folder>	applications	<application>	<folder>	*Named*	wm
application <string> of <registry key>	applications	<application>	<registry key>	*Named*	wm
application <string> of <registry>	applications	<application>	<registry>	*Named*	wm
application folder <string> of <registry key>	application folders	<folder>	<registry key>	*Named*	wm
application folder <string> of <registry>	application folders	<folder>	<registry>	*Named*	wm
application folder of <registry key>	application folders	<folder>	<registry key>	*Plain*	wm
application of <registry key>	applications	<application>	<registry key>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
application of <registry>	applications	<application>	<registry>	*Plain*	wm
application usage summary	application usage summaries	<application usage summary>	<world>	*PlainGlobal*	wm
application usage summary <string>	application usage summaries	<application usage summary>	<world>	*NamedGlobal*	wm
april	aprils	<month>	<world>	*PlainGlobal*	core
april <integer>	aprils	<day of year>	<world>	*NumberedGlobal*	core
april <integer> of <integer>	aprils	<date>	<integer>	*Numbered*	core
april of <integer>	aprils	<month and year>	<integer>	*Plain*	core
archive of <filesystem object>	archives	<boolean>	<filesystem object>	*Plain*	wm
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
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
autorun policy of <oma csp>	autorun policies	<integer>	<oma csp>	*Plain*	wm
available of <ram>	availables	<integer>	<ram>	*Plain*	wm
average interval of <battery>	average intervals	<integer>	<battery>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
average milliamps of <battery>	average milliampses	<integer>	<battery>	*Plain*	wm
backup battery	backup batteries	<backup_battery>	<world>	*PlainGlobal*	wm
battery	batteries	<battery>	<world>	*PlainGlobal*	wm
bes license	bes licenses	<license>	<world>	*PlainGlobal*	wm
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
block incoming calls of <oma csp>	block incoming callses	<integer>	<oma csp>	*Plain*	wm
block outgoing calls of <oma csp>	block outgoing callses	<integer>	<oma csp>	*Plain*	wm
bluetooth mode of <oma csp>	bluetooth modes	<integer>	<oma csp>	*Plain*	wm
bluetooth policy of <oma csp>	bluetooth policies	<integer>	<oma csp>	*Plain*	wm
boolean <string>	booleans	<boolean>	<world>	*NamedGlobal*	core
boolean <string> of <oma csp>	booleans	<boolean>	<oma csp>	*Named*	wm
brand id of <processor>	brand ids	<integer>	<processor>	*Plain*	wm
brand string of <processor>	brand strings	<string>	<processor>	*Plain*	wm
broadcast address of <network ip interface>	broadcast addresses	<ipv4 address>	<network ip interface>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
broadcast support of <network ip interface>	broadcast supports	<boolean>	<network ip interface>	*Plain*	wm
build number high of <operating system>	build number highs	<integer>	<operating system>	*Plain*	wm
build number low of <operating system>	build number lows	<integer>	<operating system>	*Plain*	wm
byte <integer> of <file>	bytes	<integer>	<file>	*Numbered*	wm
call waiting enabled of <oma csp>	call waiting enableds	<boolean>	<oma csp>	*Plain*	wm
cast <string>	casts	<cast>	<world>	*NamedGlobal*	core
cast from of <type>	casts from	<cast>	<type>	*Plain*	core
cast returning <type>	casts returning	<cast>	<world>	*Index<type>Global*	core
character <integer>	characters	<string>	<world>	*NumberedGlobal*	core
character <integer> of <string>	characters	<substring>	<string>	*Numbered*	core
character of <string>	characters	<substring>	<string>	*Plain*	core
chemistry of <battery>	chemistries	<string>	<battery>	*Plain*	wm
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 string of <network adapter>	cidr strings	<string>	<network adapter>	*Plain*	wm
cidr string of <network address list>	cidr strings	<string>	<network address list>	*Plain*	wm
cidr string of <network ip interface>	cidr strings	<string>	<network ip interface>	*Plain*	wm
client	clients	<client>	<world>	*PlainGlobal*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
client cryptography	client cryptographies	<client_cryptography>	<world>	*PlainGlobal*	wm
client folder of <site>	client folders	<folder>	<site>	*Plain*	wm
client license	client licenses	<license>	<world>	*PlainGlobal*	wm
codepage of <file version block>	codepages	<string>	<file version block>	*Plain*	wm
common name of <license>	common names	<string>	<license>	*Plain*	wm
competition size of <selected server>	competition sizes	<integer>	<selected server>	*Plain*	wm
competition weight of <selected server>	competition weights	<integer>	<selected server>	*Plain*	wm
complete time of <action>	complete times	<time>	<action>	*Plain*	wm
component <integer> of <site version list>	components	<integer>	<site version list>	*Numbered*	core
compressed of <filesystem object>	compresseds	<boolean>	<filesystem object>	*Plain*	wm
computer id	computer ids	<integer>	<world>	*PlainGlobal*	wm
computer name	computer names	<string>	<world>	*PlainGlobal*	wm
concatenation <string> of <string>	concatenations	<string>	<string>	*Named*	core
concatenation of <string>	concatenations	<string>	<string>	*Plain*	core
conjunction of <boolean>	conjunctions	<boolean>	<boolean>	*Plain*	core
constrained of <action>	constraineds	<boolean>	<action>	*Plain*	wm
construct xml <string> of <oma csp>	construct xmls	<string>	<oma csp>	*Named*	wm
content of <file>	contents	<file content>	<file>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
controller of <action lock state>	controllers	<string>	<action lock state>	*Plain*	wm
creation time of <filesystem object>	creation times	<time>	<filesystem object>	*Plain*	wm
csd version of <operating system>	csd versions	<string>	<operating system>	*Plain*	wm
csidl folder <integer>	csidl folders	<folder>	<world>	*NumberedGlobal*	wm
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 month	current months	<month>	<world>	*PlainGlobal*	core
current month_and_year	current months_and_years	<month and year>	<world>	*PlainGlobal*	core
current relay	current relays	<current relay>	<world>	*PlainGlobal*	wm
current site	current sites	<site>	<world>	*PlainGlobal*	wm
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*	wm
current year	current years	<year>	<world>	*PlainGlobal*	core
custom site subscription effective date <string>	custom site subscription effective dates	<time>	<world>	*NamedGlobal*	wm
date <string>	dates	<date>	<world>	*NamedGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
date <time zone> of <time>	dates	<date>	<time>	*Index<time zone>*	core
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
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*	wm
default web browser	default web browsers	<wince_web_browser>	<world>	*PlainGlobal*	wm
descendant folder of <folder>	descendant folders	<folder>	<folder>	*Plain*	wm
descendant of <folder>	descendants	<file>	<folder>	*Plain*	wm
description of <network adapter>	descriptions	<string>	<network adapter>	*Plain*	wm
description of <wince network connection detail>	descriptions	<string>	<wince network connection detail>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
desired encrypt report of <client_cryptography>	desired encrypt reports	<boolean>	<client_cryptography>	*Plain*	wm
desktop quick connect authentication policy of <oma csp>	desktop quick connect authentication policies	<integer>	<oma csp>	*Plain*	wm
destination network of <wince network connection detail>	destination networks	<string>	<wince network connection detail>	*Plain*	wm
dhcp enabled of <network adapter>	dhcp enableds	<boolean>	<network adapter>	*Plain*	wm
dhcp server of <network adapter>	dhcp servers	<ipv4 address>	<network adapter>	*Plain*	wm
direct object type of <property>	direct object types	<type>	<property>	*Plain*	core
disjunction of <boolean>	disjunctions	<boolean>	<boolean>	*Plain*	core
distance of <selected server>	distances	<integer range>	<selected server>	*Plain*	wm
divided by zero of <floating point>	divided by zeroes	<boolean>	<floating point>	*Plain*	core
dns name	dns names	<string>	<world>	*PlainGlobal*	wm
dns server of <network adapter>	dns servers	<network address list>	<network adapter>	*Plain*	wm
dns server of <network>	dns servers	<network address list>	<network>	*Plain*	wm
dns suffix of <network adapter>	dns suffixes	<string>	<network adapter>	*Plain*	wm
download file <string>	download files	<file>	<world>	*NamedGlobal*	wm
download folder	download folders	<folder>	<world>	*PlainGlobal*	wm
download path <string>	download paths	<string>	<world>	*NamedGlobal*	wm
drive	drives	<drive>	<world>	*PlainGlobal*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
drive <string>	drives	<drive>	<world>	*NamedGlobal*	wm
drive of <filesystem object>	drives	<drive>	<filesystem object>	*Plain*	wm
drm security policy of <oma csp>	drm security policies	<integer>	<oma csp>	*Plain*	wm
effective date of <action lock state>	effective dates	<time>	<action lock state>	*Plain*	wm
effective date of <setting>	effective dates	<time>	<setting>	*Plain*	wm
element of <integer set>	elements	<integer>	<integer set>	*Plain*	core
element of <string set>	elements	<string>	<string set>	*Plain*	core
email address of <license>	email addresses	<string>	<license>	*Plain*	wm
enabled of <gps>	enableds	<boolean>	<gps>	*Plain*	wm
enabled of <setting>	enableds	<boolean>	<setting>	*Plain*	wm
encrypt removable storage policy of <oma csp>	encrypt removable storage policies	<integer>	<oma csp>	*Plain*	wm
encrypt report failure message of <client_cryptography>	encrypt report failure messages	<string>	<client_cryptography>	*Plain*	wm
encrypt report of <client_cryptography>	encrypt reports	<boolean>	<client_cryptography>	*Plain*	wm
encrypted message policy of <oma csp>	encrypted message policies	<integer>	<oma csp>	*Plain*	wm
encryption certificate of <license>	encryption certificates	<x509 certificate>	<license>	*Plain*	wm
end of <substring>	ends	<string position>	<substring>	*Plain*	core
end of <time range>	ends	<time>	<time range>	*Plain*	core
error <string>	errors	<undefined>	<world>	*NamedGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
escape of <string>	escapes	<string>	<string>	*Plain*	wm
evaluation of <license>	evaluations	<boolean>	<license>	*Plain*	wm
executable file format of <file>	executable file formats	<string>	<file>	*Plain*	wm
expiration date of <action lock state>	expiration dates	<time>	<action lock state>	*Plain*	wm
expiration date of <license>	expiration dates	<time>	<license>	*Plain*	wm
expiration state of <license>	expiration states	<string>	<license>	*Plain*	wm
extended family of <processor>	extended families	<integer>	<processor>	*Plain*	wm
extended model of <processor>	extended models	<integer>	<processor>	*Plain*	wm
extrema of <date>	extremas	<(date, date)>	<date>	*Plain*	core
extrema of <day of month>	extremas	<(day of month, day of month)>	<day of month>	*Plain*	core
extrema of <day of year>	extremas	<(day of year, day of year)>	<day of year>	*Plain*	core
extrema of <floating point>	extremas	<(floating point, floating point)>	<floating point>	*Plain*	core
extrema of <hertz>	extremas	<(hertz, hertz)>	<hertz>	*Plain*	core
extrema of <integer>	extremas	<(integer, integer)>	<integer>	*Plain*	core
extrema of <ipv4 address>	extremas	<(ipv4 address, ipv4 address)>	<ipv4 address>	*Plain*	core
extrema of <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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
false	falses	<boolean>	<world>	*PlainGlobal*	core
family name of <processor>	family names	<string>	<processor>	*Plain*	wm
family of <network interface>	families	<integer>	<network interface>	*Plain*	wm
family of <processor>	families	<integer>	<processor>	*Plain*	wm
feature mask of <processor>	feature masks	<integer>	<processor>	*Plain*	wm
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*	wm
file <string> of <folder>	files	<file>	<folder>	*Named*	wm
file extension <string> of <registry>	file extensions	<registry key>	<registry>	*Named*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
file of <folder>	files	<file>	<folder>	*Plain*	wm
file system type of <drive>	file system types	<string>	<drive>	*Plain*	wm
file type <string> of <registry>	file types	<registry key>	<registry>	*Named*	wm
file version of <file>	file versions	<version>	<file>	*Plain*	wm
file_supports_encryption of <drive>	file_supports_encryptions	<boolean>	<drive>	*Plain*	wm
file_supports_object_ids of <drive>	file_supports_object_idss	<boolean>	<drive>	*Plain*	wm
file_supports_reparse_points of <drive>	file_supports_reparse_pointss	<boolean>	<drive>	*Plain*	wm
file_supports_sparse_files of <drive>	file_supports_sparse_filess	<boolean>	<drive>	*Plain*	wm
file_volume_quotas of <drive>	file_volume_quotass	<boolean>	<drive>	*Plain*	wm
final part <time interval> of <time range>	final parts	<time range>	<time range>	*Index<time interval>*	core
find file <string> of <folder>	find files	<file>	<folder>	*Named*	wm
finite of <floating point>	finites	<boolean>	<floating point>	*Plain*	core
fips mode of <license>	fips modes	<boolean>	<license>	*Plain*	wm
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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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*	wm
first saturday of <month and year>	first saturdays	<date>	<month and year>	*Plain*	core
first start time of <application usage summary>	first start times	<time>	<application usage summary>	*Plain*	wm
first sunday of <month and year>	first sundays	<date>	<month and year>	*Plain*	core
first thursday of <month and year>	first thursdays	<date>	<month and year>	*Plain*	core
first tuesday of <month and year>	first tuesdays	<date>	<month and year>	*Plain*	core
first wednesday of <month and year>	first wednesdays	<date>	<month and year>	*Plain*	core
fixed dialing enabled of <oma csp>	fixed dialing enableds	<boolean>	<oma csp>	*Plain*	wm
fixlet of <site>	fixlets	<fixlet>	<site>	*Plain*	wm
flags of <wince network connection detail>	flagses	<string>	<wince network connection detail>	*Plain*	wm
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*	wm
folder <string> of <drive>	folders	<folder>	<drive>	*Named*	wm
folder <string> of <folder>	folders	<folder>	<folder>	*Named*	wm
folder of <folder>	folders	<folder>	<folder>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
following text of <string position>	following texts	<substring>	<string position>	*Plain*	core
following text of <substring>	following texts	<substring>	<substring>	*Plain*	core
forward all calls enabled of <oma csp>	forward all calls enableds	<boolean>	<oma csp>	*Plain*	wm
forward all calls of <oma csp>	forward all callses	<string>	<oma csp>	*Plain*	wm
forward all calls timeout of <oma csp>	forward all calls timeouts	<integer>	<oma csp>	*Plain*	wm
forward all calls to of <oma csp>	forward all calls tos	<string>	<oma csp>	*Plain*	wm
forward calls enabled when busy of <oma csp>	forward calls enabled when busys	<boolean>	<oma csp>	*Plain*	wm
forward calls enabled when no answer of <oma csp>	forward calls enabled when no answers	<boolean>	<oma csp>	*Plain*	wm
forward calls enabled when unavailable of <oma csp>	forward calls enabled when unavailables	<boolean>	<oma csp>	*Plain*	wm
forward calls timeout when busy of <oma csp>	forward calls timeout when busys	<integer>	<oma csp>	*Plain*	wm
forward calls timeout when no answer of <oma csp>	forward calls timeout when no answers	<integer>	<oma csp>	*Plain*	wm
forward calls timeout when unavailable of <oma csp>	forward calls timeout when unavailables	<integer>	<oma csp>	*Plain*	wm
forward calls to when busy of <oma csp>	forward calls to when busys	<string>	<oma csp>	*Plain*	wm
forward calls to when no answer of <oma csp>	forward calls to when no answers	<string>	<oma csp>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
forward calls to when unavailable of <oma csp>	forward calls to when unavailables	<string>	<oma csp>	*Plain*	wm
forward calls when busy of <oma csp>	forward calls when busys	<string>	<oma csp>	*Plain*	wm
forward calls when no answer of <oma csp>	forward calls when no answers	<string>	<oma csp>	*Plain*	wm
forward calls when unavailable of <oma csp>	forward calls when unavailables	<string>	<oma csp>	*Plain*	wm
free space of <drive>	free spaces	<integer>	<drive>	*Plain*	wm
friday	fridays	<day of week>	<world>	*PlainGlobal*	core
friendly name of <network adapter>	friendly names	<string>	<network adapter>	*Plain*	wm
fs_case_is_preserved of <drive>	fs_case_is_preserveds	<boolean>	<drive>	*Plain*	wm
fs_case_sensitive of <drive>	fs_case_sensitives	<boolean>	<drive>	*Plain*	wm
fs_file_compression of <drive>	fs_file_compressions	<boolean>	<drive>	*Plain*	wm
fs_persistent_acls of <drive>	fs_persistent_aclss	<boolean>	<drive>	*Plain*	wm
fs_unicode_stored_on_disk of <drive>	fs_unicode_stored_on_disks	<boolean>	<drive>	*Plain*	wm
fs_vol_is_compressed of <drive>	fs_vol_is_compresseds	<boolean>	<drive>	*Plain*	wm
full life of <base_battery>	full lives	<time interval>	<base_battery>	*Plain*	wm
full status of <gps>	full statuses	<string>	<gps>	*Plain*	wm
gateway address <integer> of <selected server>	gateway addresses	<ipv4or6 address>	<selected server>	*Numbered*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
gateway address of <selected server>	gateway addresses	<ipv4or6 address>	<selected server>	*Plain*	wm
gateway list of <network adapter>	gateway lists	<network address list>	<network adapter>	*Plain*	wm
gateway of <network adapter>	gateways	<ipv4 address>	<network adapter>	*Plain*	wm
gather schedule authority of <site>	gather schedule authoritys	<string>	<site>	*Plain*	wm
gather schedule time interval of <site>	gather schedule time intervals	<time interval>	<site>	*Plain*	wm
gather url of <license>	gather urls	<string>	<license>	*Plain*	wm
ghz	ghzs	<hertz>	<world>	*PlainGlobal*	core
gps	gpses	<gps>	<world>	*PlainGlobal*	wm
grant manager policy of <oma csp>	grant manager policies	<integer>	<oma csp>	*Plain*	wm
grant user authenticated policy of <oma csp>	grant user authenticated policies	<integer>	<oma csp>	*Plain*	wm
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*	wm
group leader of <action>	group leaders	<boolean>	<action>	*Plain*	wm
header <string> of <fixlet>	headers	<fixlet_header>	<fixlet>	*Named*	wm
header of <fixlet>	headers	<fixlet_header>	<fixlet>	*Plain*	wm
heading of <gps>	headings	<string>	<gps>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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*	wm
host name of <root server>	host names	<string>	<root server>	*Plain*	wm
hostname	hostnames	<string>	<world>	*PlainGlobal*	wm
hour	hours	<time interval>	<world>	*PlainGlobal*	core
hour_of_day of <time of day with time zone>	hours_of_day	<integer>	<time of day with time zone>	*Plain*	core
hour_of_day of <time of day>	hours_of_day	<integer>	<time of day>	*Plain*	core
html message policy of <oma csp>	html message policies	<integer>	<oma csp>	*Plain*	wm
hyperthreading capable	hyperthreading capables	<boolean>	<world>	*PlainGlobal*	wm
hyperthreading enabled	hyperthreading enableds	<boolean>	<world>	*PlainGlobal*	wm
hz	hzs	<hertz>	<world>	*PlainGlobal*	core
ia64 of <operating system>	ia64s	<boolean>	<operating system>	*Plain*	wm
id of <action>	ids	<integer>	<action>	*Plain*	wm
id of <file version block>	ids	<string>	<file version block>	*Plain*	wm
id of <fixlet>	ids	<integer>	<fixlet>	*Plain*	wm
id of <root server>	ids	<integer>	<root server>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
id of <site group>	ids	<integer>	<site group>	*Plain*	wm
identifier of <phone>	identifiers	<string>	<phone>	*Plain*	wm
index type of <property>	index types	<type>	<property>	*Plain*	core
inexact of <floating point>	inexacts	<boolean>	<floating point>	*Plain*	core
infinite of <floating point>	infinites	<boolean>	<floating point>	*Plain*	core
initial part <time interval> of <time range>	initial parts	<time range>	<time range>	*Index<time interval>*	core
install folder <integer>	install folders	<folder>	<world>	*NumberedGlobal*	wm
integer <integer>	integers	<integer>	<world>	*NumberedGlobal*	core
integer <string>	integers	<integer>	<world>	*NamedGlobal*	core
integer <string> of <oma csp>	integers	<integer>	<oma csp>	*Named*	wm
integer ceiling of <floating point>	integer ceilings	<integer>	<floating point>	*Plain*	core
integer floor of <floating point>	integer floors	<integer>	<floating point>	*Plain*	core
interface <integer> of <network>	interfaces	<network interface>	<network>	*Numbered*	wm
interface of <network>	interfaces	<network interface>	<network>	*Plain*	wm
intersection of <integer set>	intersections	<integer set>	<integer set>	*Plain*	core
intersection of <string set>	intersections	<string set>	<string set>	*Plain*	core
invalid before of <x509 certificate>	invalid befores	<time>	<x509 certificate>	*Plain*	core
invalid of <floating point>	invalids	<boolean>	<floating point>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
ip address of <selected server>	ip addresses	<ipv4or6 address>	<selected server>	*Plain*	wm
ip addresses of <wince network connection detail>	ip addressess	<string>	<wince network connection detail>	*Plain*	wm
ip interface <integer> of <network>	ip interfaces	<network ip interface>	<network>	*Numbered*	wm
ip interface of <network>	ip interfaces	<network ip interface>	<network>	*Plain*	wm
ipv4 address <string>	ipv4 addresses	<ipv4 address>	<world>	*NamedGlobal*	core
ipv4 part of <ipv6 address>	ipv4 parts	<ipv4 address>	<ipv6 address>	*Plain*	core
ipv6 address <string>	ipv6 addresses	<ipv6 address>	<world>	*NamedGlobal*	core
ipv6 address of <network adapter>	ipv6 addresses	<ipv6 address>	<network adapter>	*Plain*	wm
ipv6 dns server of <network adapter>	ipv6 dns servers	<ipv6 address>	<network adapter>	*Plain*	wm
january	januarys	<month>	<world>	*PlainGlobal*	core
january <integer>	januarys	<day of year>	<world>	*NumberedGlobal*	core
january <integer> of <integer>	januarys	<date>	<integer>	*Numbered*	core
january of <integer>	januarys	<month and year>	<integer>	*Plain*	core
july	julys	<month>	<world>	*PlainGlobal*	core
july <integer>	julys	<day of year>	<world>	*NumberedGlobal*	core
july <integer> of <integer>	julys	<date>	<integer>	*Numbered*	core
july of <integer>	julys	<month and year>	<integer>	*Plain*	core
june	junes	<month>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
june <integer>	junes	<day of year>	<world>	*NumberedGlobal*	core
june <integer> of <integer>	junes	<date>	<integer>	*Numbered*	core
june of <integer>	junes	<month and year>	<integer>	*Plain*	core
key <string> of <registry key>	keys	<registry key>	<registry key>	*Named*	wm
key <string> of <registry>	keys	<registry key>	<registry>	*Named*	wm
key of <registry key>	keys	<registry key>	<registry key>	*Plain*	wm
khz	khzs	<hertz>	<world>	*PlainGlobal*	core
language of <file version block>	languages	<string>	<file version block>	*Plain*	wm
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*	wm
last child of <xml dom node>	last children	<xml dom node>	<xml dom node>	*Plain*	core
last connected of <wince network connection detail>	last connecteds	<string>	<wince network connection detail>	*Plain*	wm
last gather time of <site>	last gather times	<time>	<site>	*Plain*	wm
last start time of <application usage summary>	last start times	<time>	<application usage summary>	*Plain*	wm
last time seen of <application usage summary>	last times seen	<time>	<application usage summary>	*Plain*	wm
latitude of <gps>	latitudes	<string>	<gps>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
leap of <year>	leaps	<boolean>	<year>	*Plain*	core
lease expires of <network adapter>	leases expire	<time>	<network adapter>	*Plain*	wm
lease obtained of <network adapter>	leases obtained	<time>	<network adapter>	*Plain*	wm
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 <month and year>	lengths	<time interval>	<month and year>	*Plain*	core
length of <rope>	lengths	<integer>	<rope>	*Plain*	core
length of <string>	lengths	<integer>	<string>	*Plain*	core
length of <time range>	lengths	<time interval>	<time range>	*Plain*	core
length of <year>	lengths	<time interval>	<year>	*Plain*	core
less significance <integer> of <floating point>	less significances	<floating point>	<floating point>	*Numbered*	core
life of <base_battery>	lives	<time interval>	<base_battery>	*Plain*	wm
life percent of <base_battery>	life percents	<integer>	<base_battery>	*Plain*	wm
line <integer> of <file>	lines	<file line>	<file>	*Numbered*	wm
line containing <string> of <file>	lines containing	<file line>	<file>	*Named*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
line number of <file line>	line numbers	<integer>	<file line>	*Plain*	wm
line of <file>	lines	<file line>	<file>	*Plain*	wm
line starting with <string> of <file>	lines starting with	<file line>	<file>	*Named*	wm
link speed of <network adapter>	link speeds	<integer>	<network adapter>	*Plain*	wm
load of <ram>	loads	<integer>	<ram>	*Plain*	wm
local time <string>	local times	<time>	<world>	*NamedGlobal*	core
local time zone	local time zones	<time zone>	<world>	*PlainGlobal*	core
location of <filesystem object>	locations	<string>	<filesystem object>	*Plain*	wm
lock string of <action lock state>	lock strings	<string>	<action lock state>	*Plain*	wm
locked of <action lock state>	lockeds	<boolean>	<action lock state>	*Plain*	wm
logged on user	logged on users	<logged on user>	<world>	*PlainGlobal*	wm
logical processor count	logical processor counts	<integer>	<world>	*PlainGlobal*	wm
longitude of <gps>	longitudes	<string>	<gps>	*Plain*	wm
loopback of <network ip interface>	loopbacks	<boolean>	<network ip interface>	*Plain*	wm
mac address of <network adapter>	mac addresses	<string>	<network adapter>	*Plain*	wm
main processor	main processors	<processor>	<world>	*PlainGlobal*	wm
major version of <operating system>	major versions	<integer>	<operating system>	*Plain*	wm
manufacturer of <phone>	manufacturers	<string>	<phone>	*Plain*	wm
march	marchs	<month>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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*	wm
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 <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 <ipv6 address>	maxima	<ipv6 address>	<ipv6 address>	*Plain*	core
maximum of <month and year>	maxima	<month and year>	<month and year>	*Plain*	core
maximum of <month>	maxima	<month>	<month>	*Plain*	core
maximum of <number of months>	maxima	<number of months>	<number of months>	*Plain*	core
maximum of <site version list>	maxima	<site version list>	<site version list>	*Plain*	core
maximum of <time interval>	maxima	<time interval>	<time interval>	*Plain*	core
maximum of <time of day>	maxima	<time of day>	<time of day>	*Plain*	core
maximum of <time>	maxima	<time>	<time>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
maximum of <version>	maxima	<version>	<version>	*Plain*	core
maximum of <year>	maxima	<year>	<year>	*Plain*	core
maximum seat count of <license>	maximum seat counts	<integer>	<license>	*Plain*	wm
maximum transmission unit of <network adapter>	maximum transmission units	<integer>	<network adapter>	*Plain*	wm
may	mays	<month>	<world>	*PlainGlobal*	core
may <integer>	mays	<day of year>	<world>	*NumberedGlobal*	core
may <integer> of <integer>	mays	<date>	<integer>	*Numbered*	core
may of <integer>	mays	<month and year>	<integer>	*Plain*	core
member of <site group>	members	<boolean>	<site group>	*Plain*	wm
message authentication retry number policy of <oma csp>	message authentication retry number policies	<integer>	<oma csp>	*Plain*	wm
message encryption negotiation policy of <oma csp>	message encryption negotiation policies	<integer>	<oma csp>	*Plain*	wm
mhz	mhzs	<hertz>	<world>	*PlainGlobal*	core
microsecond	microseconds	<time interval>	<world>	*PlainGlobal*	core
midnight	midnights	<time of day>	<world>	*PlainGlobal*	core
milliamps of <battery>	milliampses	<integer>	<battery>	*Plain*	wm
milliamps per hour of <battery>	milliamps per hours	<integer>	<battery>	*Plain*	wm
millisecond	milliseconds	<time interval>	<world>	*PlainGlobal*	core
millivolts of <base_battery>	millivoltses	<integer>	<base_battery>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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 <ipv6 address>	minima	<ipv6 address>	<ipv6 address>	*Plain*	core
minimum of <month and year>	minima	<month and year>	<month and year>	*Plain*	core
minimum of <month>	minima	<month>	<month>	*Plain*	core
minimum of <number of months>	minima	<number of months>	<number of months>	*Plain*	core
minimum of <site version list>	minima	<site version list>	<site version list>	*Plain*	core
minimum of <time interval>	minima	<time interval>	<time interval>	*Plain*	core
minimum of <time of day>	minima	<time of day>	<time of day>	*Plain*	core
minimum of <time>	minima	<time>	<time>	*Plain*	core
minimum of <version>	minima	<version>	<version>	*Plain*	core
minimum of <year>	minima	<year>	<year>	*Plain*	core
minor version of <operating system>	minor versions	<integer>	<operating system>	*Plain*	wm
minute	minutes	<time interval>	<world>	*PlainGlobal*	core

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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
multiplicity of <day of month with multiplicity>	multiplicities	<integer>	<day of month with multiplicity>	*Plain*	core
multiplicity of <day of week with multiplicity>	multiplicities	<integer>	<day of week with multiplicity>	*Plain*	core
multiplicity of <day of year with multiplicity>	multiplicities	<integer>	<day of year with multiplicity>	*Plain*	core
multiplicity of <floating point with multiplicity>	multiplicities	<integer>	<floating point with multiplicity>	*Plain*	core
multiplicity of <hertz with multiplicity>	multiplicities	<integer>	<hertz with multiplicity>	*Plain*	core
multiplicity of <integer with multiplicity>	multiplicities	<integer>	<integer with multiplicity>	*Plain*	core
multiplicity of <ipv4 address with multiplicity>	multiplicities	<integer>	<ipv4 address with multiplicity>	*Plain*	core
multiplicity of <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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
multiplicity of <time interval with multiplicity>	multiplicities	<integer>	<time interval with multiplicity>	*Plain*	core
multiplicity of <time of day with multiplicity>	multiplicities	<integer>	<time of day with multiplicity>	*Plain*	core
multiplicity of <time of day with time zone with multiplicity>	multiplicities	<integer>	<time of day with time zone with multiplicity>	*Plain*	core
multiplicity of <time range with multiplicity>	multiplicities	<integer>	<time range with multiplicity>	*Plain*	core
multiplicity of <time with multiplicity>	multiplicities	<integer>	<time with multiplicity>	*Plain*	core
multiplicity of <time zone with multiplicity>	multiplicities	<integer>	<time zone with multiplicity>	*Plain*	core
multiplicity of <version with multiplicity>	multiplicities	<integer>	<version with multiplicity>	*Plain*	core
multiplicity of <year with multiplicity>	multiplicities	<integer>	<year with multiplicity>	*Plain*	core
multivalued of <property>	multivalueds	<boolean>	<property>	*Plain*	core
name of <application usage summary>	names	<string>	<application usage summary>	*Plain*	wm
name of <binary operator>	names	<string>	<binary operator>	*Plain*	core
name of <cast>	names	<string>	<cast>	*Plain*	core
name of <drive>	names	<string>	<drive>	*Plain*	wm
name of <filesystem object>	names	<string>	<filesystem object>	*Plain*	wm
name of <fixlet_header>	names	<string>	<fixlet_header>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
name of <gps>	names	<string>	<gps>	*Plain*	wm
name of <logged on user>	names	<string>	<logged on user>	*Plain*	wm
name of <network adapter>	names	<string>	<network adapter>	*Plain*	wm
name of <operating system>	names	<string>	<operating system>	*Plain*	wm
name of <registry key value>	names	<string>	<registry key value>	*Plain*	wm
name of <registry key>	names	<string>	<registry key>	*Plain*	wm
name of <selected server>	names	<string>	<selected server>	*Plain*	wm
name of <setting>	names	<string>	<setting>	*Plain*	wm
name of <site>	names	<string>	<site>	*Plain*	wm
name of <type>	names	<string>	<type>	*Plain*	core
name of <unary operator>	names	<string>	<unary operator>	*Plain*	core
nan of <floating point>	nans	<boolean>	<floating point>	*Plain*	core
native registry	native registries	<registry>	<world>	*PlainGlobal*	wm
network	networks	<network>	<world>	*PlainGlobal*	wm
network connection	network connections	<wince network connection detail>	<world>	*PlainGlobal*	wm
network pin prompt policy of <oma csp>	network pin prompt policies	<integer>	<oma csp>	*Plain*	wm
network type of <oma csp>	network types	<string>	<oma csp>	*Plain*	wm
next line of <file line>	next lines	<file line>	<file line>	*Plain*	wm
next sibling of <xml dom node>	next siblings	<xml dom node>	<xml dom node>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
noon	noons	<time of day>	<world>	*PlainGlobal*	core
normal of <filesystem object>	normals	<boolean>	<filesystem object>	*Plain*	wm
normal of <floating point>	normals	<boolean>	<floating point>	*Plain*	core
november	novembers	<month>	<world>	*PlainGlobal*	core
november <integer>	novembers	<day of year>	<world>	*NumberedGlobal*	core
november <integer> of <integer>	novembers	<date>	<integer>	*Numbered*	core
november of <integer>	novembers	<month and year>	<integer>	*Plain*	core
now	nows	<time>	<world>	*PlainGlobal*	core
numeric type of <drive>	numeric types	<integer>	<drive>	*Plain*	wm
numeric value of <string>	numeric values	<integer>	<string>	*Plain*	core
obex enabled of <oma csp>	obex enableds	<boolean>	<oma csp>	*Plain*	wm
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*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
offer accepted of <action>	offer accepteds	<boolean>	<action>	*Plain*	wm
offer of <action>	offers	<boolean>	<action>	*Plain*	wm
offline of <filesystem object>	offlines	<boolean>	<filesystem object>	*Plain*	wm
oma cp network pin policy of <oma csp>	oma cp network pin policies	<integer>	<oma csp>	*Plain*	wm
oma cp user network pin policy of <oma csp>	oma cp user network pin policies	<integer>	<oma csp>	*Plain*	wm
oma cp user pin policy of <oma csp>	oma cp user pin policies	<integer>	<oma csp>	*Plain*	wm
oma csp	oma csps	<oma csp>	<world>	*PlainGlobal*	wm
oma csp <(string, string)>	oma csps	<oma csp>	<world>	*Index<(string, string)>Global*	wm
oma csp <(string, string, string)>	oma csps	<oma csp>	<world>	*Index<(string, string, string)>Global*	wm
oma csp <(string, string, string, string)>	oma csps	<oma csp>	<world>	*Index<(string, string, string, string)>Global*	wm
oma csp <string>	oma csps	<oma csp>	<world>	*NamedGlobal*	wm
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*	wm
only version block of <file>	only version blocks	<file version block>	<file>	*Plain*	wm
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*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
operating system product type <integer>	operating system product types	<operating system product type>	<world>	*NumberedGlobal*	wm
operator name of <phone>	operator names	<string>	<phone>	*Plain*	wm
organization of <license>	organizations	<string>	<license>	*Plain*	wm
origin fixlet id of <action>	origin fixlet ids	<integer>	<action>	*Plain*	wm
ota provisioning policy of <oma csp>	ota provisioning policies	<integer>	<oma csp>	*Plain*	wm
overflow of <floating point>	overflows	<boolean>	<floating point>	*Plain*	core
owner address of <phone>	owner addresses	<string>	<phone>	*Plain*	wm
owner company of <phone>	owner companies	<string>	<phone>	*Plain*	wm
owner document of <xml dom node>	owner documents	<xml dom document>	<xml dom node>	*Plain*	core
owner email of <phone>	owner emails	<string>	<phone>	*Plain*	wm
owner name of <phone>	owner names	<string>	<phone>	*Plain*	wm
owner notes of <phone>	owner noteses	<string>	<phone>	*Plain*	wm
pad of <version>	pads	<version>	<version>	*Plain*	core
parameter <string>	parameters	<string>	<world>	*NamedGlobal*	wm
parameter <string> of <action>	parameters	<string>	<action>	*Named*	wm
parent folder of <filesystem object>	parent folders	<folder>	<filesystem object>	*Plain*	wm
parent node of <xml dom node>	parent nodes	<xml dom node>	<xml dom node>	*Plain*	core
parent of <type>	parents	<type>	<type>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
password required policy of <oma csp>	password required policies	<integer>	<oma csp>	*Plain*	wm
pathname of <filesystem object>	pathnames	<string>	<filesystem object>	*Plain*	wm
pem encoded certificate of <file>	pem encoded certificates	<x509 certificate>	<file>	*Plain*	wm
pending login	pending logins	<boolean>	<world>	*PlainGlobal*	wm
pending login of <action>	pending logins	<boolean>	<action>	*Plain*	wm
pending of <action>	pendings	<boolean>	<action>	*Plain*	wm
pending restart	pending restarts	<boolean>	<world>	*PlainGlobal*	wm
pending restart <string>	pending restarts	<boolean>	<world>	*NamedGlobal*	wm
pending restart of <action>	pending restarts	<boolean>	<action>	*Plain*	wm
pending time of <action>	pending times	<time>	<action>	*Plain*	wm
performance counter frequency of <operating system>	performance counter frequencies	<hertz>	<operating system>	*Plain*	wm
performance counter of <operating system>	performance counters	<integer>	<operating system>	*Plain*	wm
phone	phones	<phone>	<world>	*PlainGlobal*	wm
phone number of <phone>	phone numbers	<string>	<phone>	*Plain*	wm
physical processor count	physical processor counts	<integer>	<world>	*PlainGlobal*	wm
platform id of <operating system>	platform ids	<integer>	<operating system>	*Plain*	wm
plural name of <property>	plural names	<string>	<property>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
point to point of <network ip interface>	point to points	<boolean>	<network ip interface>	*Plain*	wm
port number of <selected server>	port numbers	<integer>	<selected server>	*Plain*	wm
position <integer> of <string>	positions	<string position>	<string>	*Numbered*	core
position of <string>	positions	<string position>	<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*	wm
previous sibling of <xml dom node>	previous siblings	<xml dom node>	<xml dom node>	*Plain*	core
primary language of <language>	primary languages	<primary language>	<language>	*Plain*	wm
primary wins server of <network adapter>	primary wins servers	<ipv4 address>	<network adapter>	*Plain*	wm
priority of <selected server>	priorities	<integer>	<selected server>	*Plain*	wm
privileged applications policy of <oma csp>	privileged applications policies	<integer>	<oma csp>	*Plain*	wm
process xml query <string> of <oma csp>	process xml queries	<string>	<oma csp>	*Named*	wm
processor	processors	<processor>	<world>	*PlainGlobal*	wm
processor <integer>	processors	<processor>	<world>	*NumberedGlobal*	wm
product info numeric of <operating system>	product info numerics	<integer>	<operating system>	*Plain*	wm
product info string of <operating system>	product info strings	<string>	<operating system>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
product of <integer>	products	<integer>	<integer>	*Plain*	core
product type of <operating system>	product types	<operating system product type>	<operating system>	*Plain*	wm
product version of <file>	product versions	<version>	<file>	*Plain*	wm
property <string>	properties	<property>	<world>	*NamedGlobal*	core
property <string> of <type>	properties	<property>	<type>	*Named*	core
property of <type>	properties	<property>	<type>	*Plain*	core
property returning <type>	properties returning	<property>	<world>	*Index<type>Global*	core
property returning <type> of <type>	properties returning	<property>	<type>	*Index<type>*	core
ram	rams	<ram>	<world>	*PlainGlobal*	wm
random access memory	random access memories	<ram>	<world>	*PlainGlobal*	wm
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
rapi policy of <oma csp>	rapi policies	<integer>	<oma csp>	*Plain*	wm
rated speed of <phone>	rated speeds	<hertz>	<phone>	*Plain*	wm
raw file version of <file>	raw file versions	<version>	<file>	*Plain*	wm
raw product version of <file>	raw product versions	<version>	<file>	*Plain*	wm
raw version block <integer> of <file>	raw version blocks	<file version block>	<file>	*Numbered*	wm
raw version block <string> of <file>	raw version blocks	<file version block>	<file>	*Named*	wm
raw version block of <file>	raw version blocks	<file version block>	<file>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
raw version of <file>	raw versions	<version>	<file>	*Plain*	wm
readonly of <filesystem object>	readonlys	<boolean>	<filesystem object>	*Plain*	wm
recent application	recent applications	<application>	<world>	*PlainGlobal*	wm
recent application <string>	recent applications	<application>	<world>	*NamedGlobal*	wm
regapp	regapps	<application>	<world>	*PlainGlobal*	wm
regapp <string>	regapps	<application>	<world>	*NamedGlobal*	wm
registrar number of <license>	registrar numbers	<integer>	<license>	*Plain*	wm
registry	registries	<registry>	<world>	*PlainGlobal*	wm
relative significance place <integer> of <floating point>	relative significance places	<floating point>	<floating point>	*Numbered*	core
relative significance place of <floating point>	relative significance places	<floating point>	<floating point>	*Plain*	core
release of <operating system>	releases	<string>	<operating system>	*Plain*	wm
relevance of <fixlet>	relevances	<boolean>	<fixlet>	*Plain*	wm
relevant fixlet of <site>	relevant fixlets	<fixlet>	<site>	*Plain*	wm
remote of <logged on user>	remotes	<boolean>	<logged on user>	*Plain*	wm
result type of <binary operator>	result types	<type>	<binary operator>	*Plain*	core
result type of <property>	result types	<type>	<property>	*Plain*	core
result type of <unary operator>	result types	<type>	<unary operator>	*Plain*	core
revision of <phone>	revisions	<string>	<phone>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
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
roaming status of <phone>	roaming statuses	<string>	<phone>	*Plain*	wm
root folder of <drive>	root folders	<folder>	<drive>	*Plain*	wm
root server	root servers	<root server>	<world>	*PlainGlobal*	wm
rope <string>	ropes	<rope>	<world>	*NamedGlobal*	core
running application	running applications	<application>	<world>	*PlainGlobal*	wm
running application <string>	running applications	<application>	<world>	*NamedGlobal*	wm
running of <application usage summary>	runnings	<boolean>	<application usage summary>	*Plain*	wm
sample time of <gps>	sample times	<time>	<gps>	*Plain*	wm
saturday	saturdays	<day of week>	<world>	*PlainGlobal*	core
seat count state of <license>	seat count states	<string>	<license>	*Plain*	wm
seat of <license>	seats	<integer>	<license>	*Plain*	wm
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*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
secure of <wince network connection detail>	secures	<boolean>	<wince network connection detail>	*Plain*	wm
security policy of <oma csp>	security policies	<integer>	<oma csp>	*Plain*	wm
select <string> of <xml dom node>	selects	<xml dom node>	<xml dom node>	*Named*	core
selected server	selected servers	<selected server>	<world>	*PlainGlobal*	wm
send caller id of <oma csp>	send caller ids	<integer>	<oma csp>	*Plain*	wm
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
serial number of <phone>	serial numbers	<string>	<phone>	*Plain*	wm
service indication message policy of <oma csp>	service indication message policies	<integer>	<oma csp>	*Plain*	wm
service loading message policy of <oma csp>	service loading message policies	<integer>	<oma csp>	*Plain*	wm
service pack major version of <operating system>	service pack major versions	<integer>	<operating system>	*Plain*	wm
service pack minor version of <operating system>	service pack minor versions	<integer>	<operating system>	*Plain*	wm
set of <integer>	sets	<integer set>	<integer>	*Plain*	core
set of <string>	sets	<string set>	<string>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
setting <string> of <client>	settings	<setting>	<client>	*Named*	wm
setting <string> of <site>	settings	<setting>	<site>	*Named*	wm
setting of <client>	settings	<setting>	<client>	*Plain*	wm
setting of <site>	settings	<setting>	<site>	*Plain*	wm
sha1 of <file>	sha1s	<string>	<file>	*Plain*	wm
sha1 of <x509 certificate>	sha1s	<string>	<x509 certificate>	*Plain*	core
sharepoint access policy of <oma csp>	sharepoint access policies	<integer>	<oma csp>	*Plain*	wm
signal quality of <wince network connection detail>	signal qualities	<integer>	<wince network connection detail>	*Plain*	wm
signal strength of <phone>	signal strengths	<integer>	<phone>	*Plain*	wm
signed mail policy of <oma csp>	signed mail policies	<integer>	<oma csp>	*Plain*	wm
significance place <integer> of <floating point>	significance places	<floating point>	<floating point>	*Numbered*	core
significance place of <floating point>	significance places	<floating point>	<floating point>	*Plain*	core
significance threshold of <floating point>	significance thresholds	<floating point>	<floating point>	*Plain*	core
significant digits <integer> of <hertz>	significant digitss	<hertz>	<hertz>	*Numbered*	core
significant digits <integer> of <integer>	significant digitss	<integer>	<integer>	*Numbered*	core
singular name of <property>	singular names	<string>	<property>	*Plain*	core
site	sites	<site>	<world>	*PlainGlobal*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
site <string>	sites	<site>	<world>	*NamedGlobal*	wm
site number of <license>	site numbers	<integer>	<license>	*Plain*	wm
site tag of <site>	site tags	<string>	<site>	*Plain*	wm
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*	wm
size of <file>	sizes	<integer>	<file>	*Plain*	wm
size of <integer set>	sizes	<integer>	<integer set>	*Plain*	core
size of <ram>	sizes	<integer>	<ram>	*Plain*	wm
size of <registry key value>	sizes	<integer>	<registry key value>	*Plain*	wm
size of <string set>	sizes	<integer>	<string set>	*Plain*	core
size of <type>	sizes	<integer>	<type>	*Plain*	core
sl security policy of <oma csp>	sl security policies	<integer>	<oma csp>	*Plain*	wm
smime encryption algorithm policy of <oma csp>	smime encryption algorithm policies	<integer>	<oma csp>	*Plain*	wm
smime encryption policy of <oma csp>	smime encryption policies	<integer>	<oma csp>	*Plain*	wm
smime signing algorithm policy of <oma csp>	smime signing algorithm policies	<integer>	<oma csp>	*Plain*	wm
smime signing policy of <oma csp>	smime signing policies	<integer>	<oma csp>	*Plain*	wm
software certificates policy of <oma csp>	software certificates policies	<integer>	<oma csp>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
source network of <wince network connection detail>	source networks	<string>	<wince network connection detail>	*Plain*	wm
speed of <gps>	speeds	<string>	<gps>	*Plain*	wm
speed of <processor>	speeds	<hertz>	<processor>	*Plain*	wm
start date of <license>	start dates	<time>	<license>	*Plain*	wm
start of <substring>	starts	<string position>	<substring>	*Plain*	core
start of <time range>	starts	<time>	<time range>	*Plain*	core
status of <action>	statuss	<string>	<action>	*Plain*	wm
status of <base_battery>	statuses	<string>	<base_battery>	*Plain*	wm
status of <network adapter>	statuses	<integer>	<network adapter>	*Plain*	wm
status of <wince network connection detail>	statuses	<string>	<wince network connection detail>	*Plain*	wm
stepping of <processor>	steppings	<integer>	<processor>	*Plain*	wm
storage card encryption of <oma csp>	storage card encryptions	<boolean>	<oma csp>	*Plain*	wm
string <string>	strings	<string>	<world>	*NamedGlobal*	core
string <string> of <oma csp>	strings	<string>	<oma csp>	*Named*	wm
subnet address of <network adapter>	subnet addresses	<ipv4 address>	<network adapter>	*Plain*	wm
subnet address of <network address list>	subnet addresses	<ipv4 address>	<network address list>	*Plain*	wm
subnet address of <network ip interface>	subnet addresses	<ipv4 address>	<network ip interface>	*Plain*	wm
subnet mask of <network adapter>	subnet masks	<ipv4 address>	<network adapter>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
subnet mask of <network address list>	subnet masks	<ipv4 address>	<network address list>	*Plain*	wm
subnet mask of <network ip interface>	subnet masks	<ipv4 address>	<network ip interface>	*Plain*	wm
subscribe time of <site>	subscribe times	<time>	<site>	*Plain*	wm
subscriber number of <phone>	subscriber numbers	<string>	<phone>	*Plain*	wm
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*	wm
sum of <integer>	sums	<integer>	<integer>	*Plain*	core
sunday	sundays	<day of week>	<world>	*PlainGlobal*	core
symbol of <binary operator>	symbols	<string>	<binary operator>	*Plain*	core
symbol of <unary operator>	symbols	<string>	<unary operator>	*Plain*	core
system file <string>	system files	<file>	<world>	*NamedGlobal*	wm
system language	system languages	<string>	<world>	*PlainGlobal*	wm
system locale	system locales	<language>	<world>	*PlainGlobal*	wm
system of <filesystem object>	systems	<boolean>	<filesystem object>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
system ui language	system ui languages	<language>	<world>	*PlainGlobal*	wm
system wow64 folder	system wow64 folders	<folder>	<world>	*PlainGlobal*	wm
system x32 folder	system x32 folders	<folder>	<world>	*PlainGlobal*	wm
system x64 folder	system x64 folders	<folder>	<world>	*PlainGlobal*	wm
temperature of <battery>	temperatures	<floating point>	<battery>	*Plain*	wm
temporary of <filesystem object>	temporarys	<boolean>	<filesystem object>	*Plain*	wm
thursday	thursdays	<day of week>	<world>	*PlainGlobal*	core
time <string>	times	<time>	<world>	*NamedGlobal*	core
time <time zone> of <time>	times	<time of day with time zone>	<time>	*Index<time zone>*	core
time interval <string>	time intervals	<time interval>	<world>	*NamedGlobal*	core
time of <time of day with time zone>	times	<time of day>	<time of day with time zone>	*Plain*	core
time zone <string>	time zones	<time zone>	<world>	*NamedGlobal*	core
time_of_day <string>	times_of_day	<time of day>	<world>	*NamedGlobal*	core
timezone of <oma csp>	timezones	<integer>	<oma csp>	*Plain*	wm
total duration of <application usage summary>	total durations	<time interval>	<application usage summary>	*Plain*	wm
total processor core count	total processor core counts	<integer>	<world>	*PlainGlobal*	wm
total run count of <application usage summary>	total run counts	<integer>	<application usage summary>	*Plain*	wm
total space of <drive>	total spaces	<integer>	<drive>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
true	trues	<boolean>	<world>	*PlainGlobal*	core
trusted provisioning server policy of <oma csp>	trusted provisioning server policies	<integer>	<oma csp>	*Plain*	wm
trusted wap proxy policy of <oma csp>	trusted wap proxy policies	<integer>	<oma csp>	*Plain*	wm
tuesday	tuesdays	<day of week>	<world>	*PlainGlobal*	core
tuple string item <integer> of <string>	tuple string items	<string>	<string>	*Numbered*	core
tuple string item of <string>	tuple string items	<string>	<string>	*Plain*	core
two digit hour of <time of day with time zone>	two digit hours	<string>	<time of day with time zone>	*Plain*	core
two digit hour of <time of day>	two digit hours	<string>	<time of day>	*Plain*	core
two digit minute of <time of day with time zone>	two digit minutes	<string>	<time of day with time zone>	*Plain*	core
two digit minute of <time of day>	two digit minutes	<string>	<time of day>	*Plain*	core
two digit second of <time of day with time zone>	two digit seconds	<string>	<time of day with time zone>	*Plain*	core
two digit second of <time of day>	two digit seconds	<string>	<time of day>	*Plain*	core
type of <drive>	types	<string>	<drive>	*Plain*	wm
type of <network adapter>	types	<integer>	<network adapter>	*Plain*	wm
type of <phone>	types	<string>	<phone>	*Plain*	wm
type of <processor>	types	<integer>	<processor>	*Plain*	wm
type of <registry key value>	types	<registry key value type>	<registry key value>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
type of <site>	types	<string>	<site>	*Plain*	wm
type of <wince network connection detail>	types	<string>	<wince network connection detail>	*Plain*	wm
unary operator <string>	unary operators	<unary operator>	<world>	*NamedGlobal*	core
unary operator returning <type>	unary operators returning	<unary operator>	<world>	*Index<type>Global*	core
unauthenticated message policy of <oma csp>	unauthenticated message policies	<integer>	<oma csp>	*Plain*	wm
underflow of <floating point>	underflows	<boolean>	<floating point>	*Plain*	core
union of <integer set>	unions	<integer set>	<integer set>	*Plain*	core
union of <string set>	unions	<string set>	<string set>	*Plain*	core
unique value of <date>	unique values	<date with multiplicity>	<date>	*Plain*	core
unique value of <day of month>	unique values	<day of month with multiplicity>	<day of month>	*Plain*	core
unique value of <day of week>	unique values	<day of week with multiplicity>	<day of week>	*Plain*	core
unique value of <day of year>	unique values	<day of year with multiplicity>	<day of year>	*Plain*	core
unique value of <floating point>	unique values	<floating point with multiplicity>	<floating point>	*Plain*	core
unique value of <hertz>	unique values	<hertz with multiplicity>	<hertz>	*Plain*	core
unique value of <integer>	unique values	<integer with multiplicity>	<integer>	*Plain*	core
unique value of <ipv4 address>	unique values	<ipv4 address with multiplicity>	<ipv4 address>	*Plain*	core
unique value of <ipv6 address>	unique values	<ipv6 address with multiplicity>	<ipv6 address>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unique value of <month and year>	unique values	<month and year with multiplicity>	<month and year>	*Plain*	core
unique value of <month>	unique values	<month with multiplicity>	<month>	*Plain*	core
unique value of <number of months>	unique values	<number of months with multiplicity>	<number of months>	*Plain*	core
unique value of <site version list>	unique values	<site version list with multiplicity>	<site version list>	*Plain*	core
unique value of <string>	unique values	<string with multiplicity>	<string>	*Plain*	core
unique value of <time interval>	unique values	<time interval with multiplicity>	<time interval>	*Plain*	core
unique value of <time of day with time zone>	unique values	<time of day with time zone with multiplicity>	<time of day with time zone>	*Plain*	core
unique value of <time of day>	unique values	<time of day with multiplicity>	<time of day>	*Plain*	core
unique value of <time range>	unique values	<time range with multiplicity>	<time range>	*Plain*	core
unique value of <time zone>	unique values	<time zone with multiplicity>	<time zone>	*Plain*	core
unique value of <time>	unique values	<time with multiplicity>	<time>	*Plain*	core
unique value of <version>	unique values	<version with multiplicity>	<version>	*Plain*	core
unique value of <year>	unique values	<year with multiplicity>	<year>	*Plain*	core
universal time <string>	universal times	<time>	<world>	*NamedGlobal*	core
universal time zone	universal time zones	<time zone>	<world>	*PlainGlobal*	core
unsigned applications policy of <oma csp>	unsigned applications policies	<integer>	<oma csp>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unsigned cabs policy of <oma csp>	unsigned cabs policies	<integer>	<oma csp>	*Plain*	wm
unsigned prompt policy of <oma csp>	unsigned prompt policies	<integer>	<oma csp>	*Plain*	wm
unsigned themes policy of <oma csp>	unsigned themes policies	<integer>	<oma csp>	*Plain*	wm
url of <site>	urls	<string>	<site>	*Plain*	wm
user language	user languages	<string>	<world>	*PlainGlobal*	wm
user locale	user locales	<language>	<world>	*PlainGlobal*	wm
user ui language	user ui languages	<language>	<world>	*PlainGlobal*	wm
usual name of <property>	usual names	<string>	<property>	*Plain*	core
value <string> of <file version block>	values	<string>	<file version block>	*Named*	wm
value <string> of <oma csp>	values	<string>	<oma csp>	*Named*	wm
value <string> of <registry key>	values	<registry key value>	<registry key>	*Named*	wm
value of <fixlet_header>	values	<string>	<fixlet_header>	*Plain*	wm
value of <registry key>	values	<registry key value>	<registry key>	*Plain*	wm
value of <setting>	values	<string>	<setting>	*Plain*	wm
variable of <file>	variables	<string>	<file>	*Plain*	wm
vendor name of <processor>	vendor names	<string>	<processor>	*Plain*	wm
version <string>	versions	<version>	<world>	*NamedGlobal*	core
version block <integer> of <file>	version blocks	<file version block>	<file>	*Numbered*	wm
version block <string> of <file>	version blocks	<file version block>	<file>	*Named*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
version block of <file>	version blocks	<file version block>	<file>	*Plain*	wm
version of <current relay>	versions	<version>	<current relay>	*Plain*	wm
version of <file>	versions	<version>	<file>	*Plain*	wm
version of <site>	versions	<integer>	<site>	*Plain*	wm
version of <wince_web_browser>	versions	<version>	<wince_web_browser>	*Plain*	wm
version string <string> of <module>	version strings	<string>	<module>	*Named*	core
waiting for download of <action>	waiting for downloads	<boolean>	<action>	*Plain*	wm
wake on lan subnet cidr string	wake on lan subnet cidr strings	<string>	<world>	*PlainGlobal*	wm
wakeonlan enabled of <network adapter>	wakeonlan enableds	<boolean>	<network adapter>	*Plain*	wm
wap signed message policy of <oma csp>	wap signed message policies	<integer>	<oma csp>	*Plain*	wm
wednesday	wednesdays	<day of week>	<world>	*PlainGlobal*	core
week	weeks	<time interval>	<world>	*PlainGlobal*	core
weight of <selected server>	weights	<integer>	<selected server>	*Plain*	wm
windows display time <string>	windows display times	<time>	<world>	*NamedGlobal*	core
windows file <string>	windows files	<file>	<world>	*NamedGlobal*	wm
windows folder	windows folders	<folder>	<world>	*PlainGlobal*	wm
wins enabled of <network adapter>	wins enableds	<boolean>	<network adapter>	*Plain*	wm
winsock2 supported of <network>	winsock2 supporteds	<boolean>	<network>	*Plain*	wm

Key Phrase	Plural	Creates a	From a	*Form*	Ref
wsp push policy of <oma csp>	wsp push policies	<integer>	<oma csp>	*Plain*	wm
x32 registry	x32 registries	<registry>	<world>	*PlainGlobal*	wm
x64 of <operating system>	x64s	<boolean>	<operating system>	*Plain*	wm
x64 registry	x64 registries	<registry>	<world>	*PlainGlobal*	wm
xml document of <file>	xml documents	<xml dom document>	<file>	*Plain*	wm
xml document of <string>	xml documents	<xml dom document>	<string>	*Plain*	core
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>
<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>
<file content> as lowercase	<file content>	<file content>
<file content> as uppercase	<file content>	<file content>
<file> as string	<string>	<file>
<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>

Key Phrase	Creates a	From a
<floating point> as string	<string>	<floating point>
<hertz> as string	<string>	<hertz>
<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>
<ipv4 address> as ipv6 address	<ipv6 address>	<ipv4 address>
<ipv4 address> as string	<string>	<ipv4 address>
<ipv6 address> as compressed string	<string>	<ipv6 address>
<ipv6 address> as compressed string with ipv4	<string>	<ipv6 address>
<ipv6 address> as string	<string>	<ipv6 address>
<ipv6 address> as string with ipv4	<string>	<ipv6 address>
<ipv6 address> as string with leading zeros	<string>	<ipv6 address>
<language> as string	<string>	<language>
<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>

Key Phrase	Creates a	From a
<month> as two digits	<string>	<month>
<number of months> as string	<string>	<number of months>
<operating system> as string	<string>	<operating system>
<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>
<setting> as string	<string>	<setting>
<site version list> as string	<string>	<site version list>
<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 integer	<integer>	<string>
<string> as left trimmed string	<string>	<string>
<string> as local time	<time>	<string>
<string> as local zoned time_of_day	<time of day with time zone>	<string>
<string> as lowercase	<string>	<string>
<string> as month	<month>	<string>
<string> as right trimmed string	<string>	<string>
<string> as site version list	<site version list>	<string>
<string> as string	<string>	<string>
<string> as time	<time>	<string>
<string> as time interval	<time interval>	<string>
<string> as time zone	<time zone>	<string>
<string> as time_of_day	<time of day>	<string>
<string> as trimmed string	<string>	<string>
<string> as universal time	<time>	<string>
<string> as universal zoned time_of_day	<time of day with time zone>	<string>
<string> as uppercase	<string>	<string>
<string> as version	<version>	<string>
<string> as windows display time	<time>	<string>
<string> as year	<year>	<string>

Key Phrase	Creates a	From a
<string> as zoned time_of_day	<time of day with time zone>	<string>
<time interval> as string	<string>	<time interval>
<time of day with time zone> as string	<string>	<time of day with time zone>
<time of day> as string	<string>	<time of day>
<time range> as string	<string>	<time range>
<time zone> as string	<string>	<time zone>
<time> as local string	<string>	<time>
<time> as string	<string>	<time>
<time> as universal string	<string>	<time>
<type> as string	<string>	<type>
<unary operator> as string	<string>	<unary operator>
<version> as string	<string>	<version>
<version> as version	<version>	<version>
<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.

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.

Part Five

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Y