Macintosh Inspector Library

A Guide to the BigFix® Macintosh Inspectors

BigFix, Inc.
Emeryville, CA

Last Modified: January 23, 2007
Version 6.0

BigFix, Inc.
1480 64th Street, Suite 200
Emeryville, CA 94608

Contents

Preface

The *Macintosh Inspector Library* is a guide to the ordinary phrases (known as Inspectors) of the **BigFix Relevance Language**™ for the Macintosh computer. Using this guide, you can write your own Relevance Expressions and use them to target actions to exactly those computers that need them. Both the **BES Console** and the **BigFix Development Environment** allow you to write **Fixlet**® messages and post them to **Fixlet Sites**. For more information on how these programs support the Relevance language, see the ***BigFix Enterprise Suite (BES) Console Operator's Guide*** and the ***BigFix Relevance Language Reference***.

Audience

This guide is for IT managers, product support groups and other people who want to write Fixlet messages.

IT managers will use the BigFix Enterprise Suite (BES) to keep a network of computers up to date and running smoothly without interruption.

To get the most out of this manual, it helps to have some experience with the BigFix Relevance Language.

Organization of this manual

For each Inspector in this library, there is a list of corresponding properties. The Inspectors are organized by category as follows:

- **Primitive Objects.** This chapter covers the basic data types supported by the language and describes the operations that can be applied to them.
- **World Objects.** This chapter covers the keywords used to create all the 'top' level objects of the world. The properties of these objects provide access to all levels of the machine state that can be inspected.
- **File System 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.
- **System Objects.** This chapter covers the keywords available for querying 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.
- **Site Objects.** This chapter covers the keywords that query the properties of Fixlet sites to which the client is subscribed.
- **Client Objects.** This chapter covers properties of the client application hosting the relevance evaluation.
- **Environment Objects.** This chapter provides access to environment variables.
- **User Objects.** This chapter covers the local and current user keywords. A Local User object is provided to access the user data of the local machine. Note that domain users are not available through this inspector.
- **Action Objects.** These are the keywords associated with properties available for inspection during the execution of BigFix Actions.
- **Networking Objects.** These are the keywords used to query the local network configuration.
- **Introspectors.** This chapter describes Inspectors that inspect the Inspector language itself.
- **Miscellaneous.** These are objects included chiefly for compatibility with other operating systems.
- **Key Phrases (Inspectors).** This chapter provides an alphabetical list of all the Inspector key phrases along with the form, context object type, and resulting object type.

Conventions Used in this manual

This document makes use of the following conventions and nomenclature:

Convention	Use
Bold Sans	A bold sans-serif font is used for Inspector headers.
`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 an object type. For instance to indicate the creation and usage of a particular object, you might see “absolute value of <integer>” which indicates that an integer is to follow the “absolute value of” keyphrase.
Italics	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 lists the first implementation for every relevant operating system.

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

Most Inspectors have equivalent implementations on other operating systems, allowing you to write cross-platform relevance expressions. There are exceptions, of course. To keep track of them for each Inspector and operating system, the debut BigFix version is listed at the end of the description, e.g.:

Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1

These are the abbreviations for some of the current operating systems:

Mac: the Macintosh version of BES.

Win: the Windows version of the BigFix Enterprise Suite (BES).

Lin: the Red Hat & Suse Linux version of BES.

Sol: the SUN Solaris operating system version of BES.

HPUX: the Hewlett-Packard Unix version of BES.

AIX: the AIX version of BES.

Introduction

This manual details the properties and operators of the BigFix Inspector keywords. Inspectors are the basis of the Relevance Language. They can be thought of as object-oriented representations of the underlying computer system. With Inspectors, you can write Relevance expressions that query all aspects of the computer. Inspectors are also used to produce substituted variables in action buttons, allowing you to target actions at well-specified subsets of your networked computers. They can also be used to create human-readable descriptions of any given computer system.

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, such as "character of <string>" or "character <integer> of <string>."

This document describes inspectors for the Macintosh Operating System. Contact your BigFix sales representative for information about Inspector Guides for other operating systems, including Windows, Solaris, HP, AIX, Red Hat and Suse Linux.

In the following pages, you will find tables defining the inspectors of the relevance language. The inspectors come in seven **forms** depending upon their context:

Form	Syntax required
Cast	<object> as keyword
Global	keyword
Named	keyword "*name*" of <object>
NamedGlobal	keyword "*name*"
Numbered	keyword *number* of <object>
NumberedGlobal	keyword *number*
Plain	keyword of <object>

These differ from one another in format and in the syntax they require. Except for Cast, 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 listed in the Keyword section later in this document.

Creation Methods are used to create objects of the specified type, and various **Properties** are defined for each object.

Operators list the binary and unary operations that can be performed with the given object type. Binary operators take two inputs and generate one output. The integer '+' (addition) operator is an example of a binary operation. Unary operators take a single input and generate a single output. The boolean 'Not' operation is an example of a unary operation.

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.

Boolean

Creation Methods

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

Key Phrase	*Form*	Description
<string> as boolean	*Cast*	Returns a boolean TRUE or FALSE from a string. The string must contain values of "TRUE" or "FALSE". Case is ignored. For example, "FalSe" as boolean = FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
bit <integer> of <integer>	*Numbered*	Return TRUE if the bit referenced by the integer is on. Bits are numbered starting with zero being the least significant. For example, bit 0 of 5 and bit 2 of 5 and not bit 1 of 5 = TRUE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
boolean <integer> of <array>	*Numbered*	Get, from an array, a boolean keyed by the specified integer. Mac:4.1
boolean <string>	*NamedGlobal*	Creates the boolean value of the <string>, e.g., • boolean "False" = FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
false	*PlainGlobal*	Creates a boolean with value FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
inexact of <floating point>	*Plain*	Returns TRUE if the calculation raised the inexact exception; that is, if some intermediate result could not be represented exactly. Win:4.1, Mac:4.1
infinite of <floating point>	*Plain*	Returns TRUE if the floating point number is infinite. Win:4.1, Mac:4.1

Key Phrase	*Form*	Description
invalid of <floating point>	*Plain*	Returns TRUE if the calculation raised the invalid exception; that is, if some part of the calculation a function was applied to a value outside its domain. Win:4.1, Mac:4.1
nan of <floating point>	*Plain*	Returns TRUE if the value is not a number. Win:4.1, Mac:4.1
normal of <floating point>	*Plain*	Returns TRUE if the value is a valid floating point number. Win:4.1, Mac:4.1
overflow of <floating point>	*Plain*	Returns TRUE if the calculation raised the overflow exception; that is, if some intermediate result was too large to be represented, but not an exact infinity. Win:4.1, Mac:4.1
powerpc	*PlainGlobal*	Returns TRUE if the cpu is a PowerPC, FALSE if it is a 68000 chip. Mac:4.1
true	*PlainGlobal*	Creates a boolean with value TRUE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
underflow of <floating point>	*Plain*	Returns TRUE if the calculation raised the underflow exception; that is, if some intermediate result was a nonzero value too small to be represented. Win:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<boolean> as string	*Cast*	<string>	Converts the boolean value to a string. The possible values returned are "True" and "False" with this exact case, e.g., • TRUE as string = "True". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Operators

Key phrase	Return Type	Description
<boolean> * <time range>	*<timed(time range, boolean)>*	Returns a time interval labeled with a boolean TRUE or FALSE. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<boolean> {cmp} <boolean>	*<boolean>*	Compare two boolean expressions. Returns another boolean, depending on the evaluation of the comparison: • {cmp} is one of: =, != . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<boolean> {op} <boolean>	*<boolean>*	Operates on two boolean expressions. Returns another boolean, depending on the evaluation of the operation, e.g., (True And True) = True. • {op} is one of: And, Or . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<floating point> {cmp} <floating point>	*<boolean>*	Compares two floating point numbers, where: • {cmp} is one of: =, <, <=. Win:4.1, Mac:4.1
<floating point> {cmp} <integer>	*<boolean>*	Compares a floating point number and an integer, where: • {cmp} is one of: =, <=, <. Win:4.1, Mac:4.1
<integer> {cmp} <floating point>	*<boolean>*	Compares an integer to a floating point number, where: • {cmp} is one of: =, <=, <. Win:4.1, Mac:4.1
<time interval> {cmp} <time interval>	*<boolean>*	Compare two time intervals, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time range> * <boolean>	*<timed(time range, boolean)>*	Returns a time interval labeled with the specified boolean, in the form of: • (<date> to <date>), <boolean>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Examples

- `bit 0 of (least integer + 1)`
  - Returns the least significant bit of the smallest possible integer, plus one.
- `infinite of (floating point "1"/ 0)`
  - Returns TRUE.
- `nan of (floating point "1.e-99999" * floating point "1.e999999")`
  - Returns TRUE.
- `overflow of (floating point "1.0e50000")`
  - Returns TRUE, since the number is too big to represent in floating point.

Integer

Integers are represented internally as 64-bit signed values.

Creation Methods

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

Key Phrase	*Form*	Description
<floating point> as integer	*Cast*	Rounds off and casts a floating point number as an integer. Win:6.0, Mac:6.0
<integer> as integer	*Cast*	Integer casting for completeness. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as integer	*Cast*	Converts from a string to an integer. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
absolute value of <integer>	*Plain*	Creates the positive value of the <integer> object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
cpu speed	*PlainGlobal*	Returns the clock speed of the cpu in Hz. Mac:4.1
day of <time>	*Plain*	The day of the month derived from the given time. Mac:4.1
day_of_week of <time>	*Plain*	The day of the week derived from the given time. Sunday = 0, etc. Mac:4.1

Key Phrase	***Form***	**Description**
gestalt <string>	*NamedGlobal*	This reads a 32 bit integer from the MacOS. The selector name is the four character OSType that chooses which item is being inspected. The interpretation of the result depends on the selector. It might represent an integer or a version, for instance. Mac:4.1
greatest integer	*PlainGlobal*	Creates the value 9,223,372,036,854,775,807. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hexadecimal integer <string>	*NamedGlobal*	Creates an integer from the provided hexadecimal value. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
hour of <time>	*Plain*	The hour derived from the given time. The range is from 0 to 23. Mac:4.1
integer <integer>	*NumberedGlobal*	Creates a global object with the given integer value, e.g., Integer 123. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
integer <string>	*NamedGlobal*	Creates a global object with the integer value given by a string, e.g., Integer "123" creates the value 123. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
integer ceiling of <floating point>	*Plain*	Returns the smallest integer not less than the floating point number. For example, ceiling of 2.1 = 3, ceiling of 2 = 2 and ceiling of -2.3 = -2. Win:6.0, Mac:6.0
integer floor of <floating point>	*Plain*	Returns the largest integer less than or equal to the floating point number. For example, floor of 2.8 = 2, floor of -2 = -2 and floor of -2.1 = -3. For nonnegative x, this is the same as the integer part of x. Win:6.0, Mac:6.0
keyboard type	*PlainGlobal*	Returns the keyboard type. Mac:4.1
least integer	*PlainGlobal*	Creates the value -9,223,372,036,854,775,808. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	***Form***	**Description**
length of <rope>	*Plain*	Creates an integer object corresponding to the number of bytes in the rope. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
length of <string>	*Plain*	Creates an integer object corresponding to the number of bytes in the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
lower bound of <integer range>	*Plain*	The low end of the integer range. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
machine type	*PlainGlobal*	Returns the machine type. The value is from a long enumeration of all Mac platforms. See the header file Gestalt.h. Mac:4.1
maximum of <integer>	*Plain*	Returns the maximum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
minimum of <integer>	*Plain*	Returns the minimum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
minute of <time>	*Plain*	The minute derived from the given time. The range is from 0 to 59. Mac:4.1
month of <time>	*Plain*	The month derived from the given time. January = 1, etc. Mac:4.1
nubus map	*PlainGlobal*	Returns the nubus map. Mac:4.1
physical ram	*PlainGlobal*	Returns the amount of physical ram in the computer. Mac:4.1
second of <time>	*Plain*	The second derived from the given time. The range is from 0 to 59. Mac:4.1
significant digits <integer> of <integer>	*Numbered*	Creates a number with <integer> significant digits (e.g.. significant digits 3 of 1235569 = 1240000). Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
sum of <integer>	*Plain*	Returns the sum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Description
upper bound of <integer range>	*Plain*	The high end of the integer range. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
year of <time>	*Plain*	The year derived from the given time. Mac:4.1

Properties

Integers are represented internally as 64-bit signed values.

Key Phrase	*Form*	Return Type	Description
<integer> as bit set	*Cast*	<bit set>	Returns the bits of the binary representation of the integer; bit zero is the least-significant bit. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<integer> as bits	*Cast*	<bit set>	Returns the bits of the binary representation of the integer; bit zero is the least-significant bit. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<integer> as floating point	*Cast*	<floating point>	Converts an integer into a floating point number. Win:4.1, Mac:4.1
<integer> as hexadecimal	*Cast*	<string>	Converts an integer into a hexadecimal string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
<integer> as integer	*Cast*	<integer>	Reflective cast for completeness. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer> as string	*Cast*	<string>	Converts an integer to a string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
absolute value of <integer>	*Plain*	<integer>	Returns the positive value of the integer. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
bit <integer> of <integer>	*Numbered*	<boolean>	Returns TRUE if the numbered bit is on. Bits are numbered starting at zero. Bit 0 is the least significant bit. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
maximum of <integer>	*Plain*	<integer>	Returns the maximum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
mean of <integer>	*Plain*	<floating point>	The mean of the integer(s). Win:5.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
minimum of <integer>	*Plain*	<integer>	Returns the minimum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
significant digits <integer> of <integer>	*Numbered*	<integer>	Returns a number with <integer> significant digits (e.g., significant digits 3 of 1235569 = 1240000). Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
standard deviation of <integer>	*Plain*	<floating point>	The standard deviation of the integer(s). Win:5.1, Mac:4.1
sum of <integer>	*Plain*	<integer>	Returns the sum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
unique value of <integer>	*Plain*	<integer with multiplicity>	Given a set of integers, returns the number of instances of each integer. Given (1,2,2,2,3), returns (1,3,1). Earlier versions of this Inspector returned the unique set of integers. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Operators

Key phrase	Return Type	Description
<floating point> {cmp} <integer>	*<boolean>*	Compares a floating point number and an integer, where: • {cmp} is one of: =, <=, <. Win:4.1, Mac:4.1
<floating point> {op} <integer>	*<floating point>*	Operates on a floating point number and an integer, returning a floating point number, where: • {op} is one of: +, -, *, /, And . Win:4.1, Mac:4.1
<hertz> {op} <integer>	*<hertz>*	Returns a hertz object operated on by the given integer, where: • {op} is one of: *, / . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer> * <time range>	*<timed(time range, integer)>*	Returns a tuple of a time interval and an integer. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Key phrase	Return Type	Description
<integer> {cmp} <floating point>	<*boolean*>	Compares an integer to a floating point number, where: • {cmp} is one of: =, <=, <. Win:4.1, Mac:4.1
<integer> {cmp} <integer>	<*boolean*>	Returns boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer> {op} <floating point>	<*floating point*>	Operates on an integer and a floating point number, returning a floating point number, where: • {op} is one of: -, +, *, /. Win:4.1, Mac:4.1
<integer> {op} <integer>	<*integer*>	Returns the integer solution to the equation, depending on the operator, where: • {op} is one of: +, -, *, /, mod . Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time range> * <integer>	<*timed(time range, integer)*>	Returns a time interval labeled with the specified integer, in the form of: • (<date> to <date>), <integer>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Examples

- `hexadecimal integer "A0"`
  - Returns 160.
- `255 as hexadecimal`
  - Returns the string "ff".
- `maximum of (7;2;4;5)`
  - Returns 7.
- `significant digits 3 of 1235569`
  - Returns 1240000.
- `21 mod 5`
  - Returns 1.

Integer Range

Specifies a range between two 64-bit signed integers.

Creation Methods

Key Phrase	*Form*	Description
distance of <selected server>	*Plain*	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:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
lower bound of <integer range>	*Plain*	<integer>	The low end of the integer range. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
upper bound of <integer range>	*Plain*	<integer>	The high end of the integer range. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Integer With Multiplicity

These Inspectors deal with arrays of integers, allowing you to pluck out unique numbers and count them. These objects are derived from integer types.

Creation Methods

Key Phrase	*Form*	Description
unique value of <integer>	*Plain*	Given a set of integers, returns the number of instances of each integer. Given (1,2,2,2,3), returns (1,3,1). Earlier versions of this Inspector returned the unique set of integers. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <integer with multiplicity>	*Plain*	<integer>	Returns the multiplicity (quantity) of each element in a multiple integer list. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Floating Point

The point type holds a floating-point number, with precision dependent on the computer. It also keeps track of the IEEE floating-point exceptions raised in a calculation and an estimate of the significance with which the number should be expressed when it is converted to a string. All arithmetic operations are carried out to the full precision of the computer; only conversions to string are affected by the estimated significance.

Creation Methods

Key Phrase	*Form*	Description
<integer> as floating point	*Cast*	Converts an integer into a floating point number. Win:4.1, Mac:4.1
<string> as floating point	*Cast*	Converts the contents of a string into a floating point number. Win:4.1, Mac:4.1
floating point <string>	*NamedGlobal*	Creates a floating point number from the provided string. Win:4.1, Mac:4.1
less significance <integer> of <floating point>	*Numbered*	Removes <integer> number of digits of significance from the floating point value. Win:4.1, Mac:4.1
mean of <floating point>	*Plain*	The mean of the floating point number(s). Win:5.1, Mac:4.1
mean of <integer>	*Plain*	The mean of the integer(s). Win:5.1, Mac:4.1
more significance <integer> of <floating point>	*Numbered*	Adds <integer> number of digits of significance to the floating point value. Win:4.1, Mac:4.1

Key Phrase	*Form*	Description
relative significance place <integer> of <floating point>	*Numbered*	The same floating point value, to be expressed to the given number of significant digits. Win:4.1, Mac:4.1
relative significance place of <floating point>	*Plain*	The base 10 logarithm of the quotient of the value and its significance place; approximately the number of significant digits to which the number should be expressed. Win:4.1, Mac:4.1
significance place <integer> of <floating point>	*Numbered*	The same floating point value, to be expressed to the given decimal place. Win:4.1, Mac:4.1
significance place of <floating point>	*Plain*	The base 10 logarithm of the significance threshold; approximately the number of digits to the left (positive) or right (negative) of the ones place to which the number should be expressed. Win:4.1, Mac:4.1
significance threshold of <floating point>	*Plain*	The difference between the given value and the next number expressed to the same significance level. For example, the significance threshold of 3 is 1, the significance threshold of 3.0 is 0.1, and the significance threshold of 3000 is 1000. Win:4.1, Mac:4.1
standard deviation of <floating point>	*Plain*	The standard deviation of the floating point number(s). Win:5.1, Mac:4.1
standard deviation of <integer>	*Plain*	The standard deviation of the integer(s). Win:5.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<floating point> as integer	*Cast*	<integer>	Rounds off and casts a floating point number as an integer. Win:6.0, Mac:6.0
<floating point> as scientific notation	*Cast*	<string>	Converts a floating point number into a string with scientific notation. Win:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
<floating point> as standard notation	*Cast*	<string>	Converts a floating point number into a string with standard notation. Win:4.1, Mac:4.1
<floating point> as string	*Cast*	<string>	Converts a floating point number into a string with standard notation. Win:4.1, Mac:4.1
divided by zero of <floating point>	*Plain*	<boolean>	Returns TRUE if the calculation raised the divide-by-zero exception; that is, if some part of the calculation produced an exact infinity. Win:4.1, Mac:4.1
finite of <floating point>	*Plain*	<boolean>	Returns TRUE if the floating point number is finite. Win:4.1, Mac:4.1
inexact of <floating point>	*Plain*	<boolean>	Returns TRUE if the calculation raised the inexact exception; that is, if some intermediate result could not be represented exactly. Win:4.1, Mac:4.1
infinite of <floating point>	*Plain*	<boolean>	Returns TRUE if the floating point number is infinite. Win:4.1, Mac:4.1
integer ceiling of <floating point>	*Plain*	<integer>	Returns the smallest integer not less than the floating point number. For example, ceiling of 2.1 = 3, ceiling of 2 = 2 and ceiling of -2.3 = -2. Win:6.0, Mac:6.0
integer floor of <floating point>	*Plain*	<integer>	Returns the largest integer less than or equal to the floating point number. For example, floor of 2.8 = 2, floor of -2 = -2 and floor of -2.1 = -3. For nonnegative x, this is the same as the integer part of x. Win:6.0, Mac:6.0
invalid of <floating point>	*Plain*	<boolean>	Returns TRUE if the calculation raised the invalid exception; that is, if some part of the calculation a function was applied to a value outside its domain. Win:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
less significance <integer> of <floating point>	*Numbered*	<floating point>	Removes <integer> number of digits of significance from the floating point value. Win:4.1, Mac:4.1
mean of <floating point>	*Plain*	<floating point>	The mean of the floating point number(s). Win:5.1, Mac:4.1
more significance <integer> of <floating point>	*Numbered*	<floating point>	Adds <integer> number of digits of significance to the floating point value. Win:4.1, Mac:4.1
nan of <floating point>	*Plain*	<boolean>	Returns TRUE if the value is not a number. Win:4.1, Mac:4.1
normal of <floating point>	*Plain*	<boolean>	Returns TRUE if the value is a valid floating point number. Win:4.1, Mac:4.1
overflow of <floating point>	*Plain*	<boolean>	Returns TRUE if the calculation raised the overflow exception; that is, if some intermediate result was too large to be represented, but not an exact infinity. Win:4.1, Mac:4.1
relative significance place <integer> of <floating point>	*Numbered*	<floating point>	The same floating point value, to be expressed to the given number of significant digits. Win:4.1, Mac:4.1
relative significance place of <floating point>	*Plain*	<floating point>	The base 10 logarithm of the quotient of the value and its significance place; approximately the number of significant digits to which the number should be expressed. Win:4.1, Mac:4.1
significance place <integer> of <floating point>	*Numbered*	<floating point>	The same floating point value, to be expressed to the given decimal place. Win:4.1, Mac:4.1
significance place of <floating point>	*Plain*	<floating point>	The base 10 logarithm of the significance threshold; approximately the number of digits to the left (positive) or right (negative) of the ones place to which the number should be expressed. Win:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
significance threshold of <floating point>	*Plain*	<floating point>	The difference between the given value and the next number expressed to the same significance level. For example, the significance threshold of 3 is 1, the significance threshold of 3.0 is 0.1, and the significance threshold of 3000 is 1000. Win:4.1, Mac:4.1
standard deviation of <floating point>	*Plain*	<floating point>	The standard deviation of the floating point number(s). Win:5.1, Mac:4.1
underflow of <floating point>	*Plain*	<boolean>	Returns TRUE if the calculation raised the underflow exception; that is, if some intermediate result was a nonzero value too small to be represented. Win:4.1, Mac:4.1

Operators

Key phrase	Return Type	Description
<floating point> {op} <floating point>	*<floating point>*	Operates on two floating point numbers, returning another floating point number, where: • {op} is one of: +, -, *, /. Win:4.1, Mac:4.1
<floating point> {op} <integer>	*<floating point>*	Operates on a floating point number and an integer, returning a floating point number, where: • {op} is one of: +, -, *, / . Win:4.1, Mac:4.1
<integer> {cmp} <floating point>	*<boolean>*	Compares an integer to a floating point number, where: • {cmp} is one of: =, <=, <, >, >=, !=. Win:4.1, Mac:4.1
<integer> {op} <floating point>	*<floating point>*	Operates on an integer and a floating point number, returning a floating point number, where: • {op} is one of: -, +, *, /. Win:4.1, Mac:4.1

Examples

- `4.5 as floating point`
  - Returns 4.5.
- `less significance 2 of floating point "5.115"`
  - Returns 5.1.
- `mean of integers(1;2;3;4;5)`
  - Returns 3.0.
- `more significance 2 of floating point "5.2"`
  - Returns 5.200.
- `significance place 2 of floating point "9123"`
  - Returns 9100.
- `significance place of floating point "9000"`
  - Returns 3.00.
- `standard deviation of integers(1;2;3;4;5)`
  - Returns 1.4.
- `floating point "600987.9" as scientific notation`
  - Returns 6.009879e+5.
- `floating point "6.009e8" as standard notation`
  - Returns 600900000.
- `finite of (floating point "1"/ 0)`
  - Returns FALSE.
- `infinite of (floating point "1"/ 0)`
  - Returns TRUE.
- `integer floor of ("-2.1" as floating point)`
  - Returns -3.
- `less significance 2 of floating point "5.115"`
  - Returns 5.1.
- `mean of floating points( "1.3";"2.5")`
  - Returns 1.90.

- `more significance 2 of floating point "5.2"`
  - Returns 5.200.
- `nan of (floating point "1.e-99999" * floating point "1.e999999")`
  - Returns TRUE.
- `overflow of (floating point "1.0e50000")`
  - Returns TRUE, since the number is too big to represent in floating point.
- `significance place 2 of floating point "9123"`
  - Returns 9100.
- `significance place of floating point "9000"`
  - Returns 3.00.

String

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
<boolean> as string	*Cast*	Operates on a boolean to return a string. Possible values are "True" and "False". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<floating point> as scientific notation	*Cast*	Converts a floating point number into a string with scientific notation. Win:4.1, Mac:4.1
<floating point> as standard notation	*Cast*	Converts a floating point number into a string with standard notation. Win:4.1, Mac:4.1

Key Phrase	*Form*	Description
<floating point> as string	*Cast*	Converts a floating point number into a string with standard notation. Win:4.1, Mac:4.1
<hertz> as string	*Cast*	Creates a string containing the number of hertz and the word hertz, e.g., (3 * hz) as string = "3 hertz". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer> as hexadecimal	*Cast*	Converts an integer into a hexadecimal string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
<integer> as string	*Cast*	Creates a string formatted with the integer provided. (-22) as string = "-22". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as hexadecimal	*Cast*	Converts a string to a hexadecimal number. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
<string> as lowercase	*Cast*	Creates a lowercase version of the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as string	*Cast*	Reflexive cast of string to string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as uppercase	*Cast*	Creates an uppercase version of the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time interval> as string	*Cast*	Returns a string formatted as • ddd days, HH:MM:SS.mmmmmm • For example, millisecond as string = " 00:00:00.001". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time zone> as string	*Cast*	Creates a string containing a time zone. See <time zone>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as local string	*Cast*	Creates a string containing a time. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as string	*Cast*	Creates a string containing a time. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as universal string	*Cast*	Creates a string containing a time. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Description
character <integer>	*NumberedGlobal*	Creates a string containing the single ASCII character for the decimal number provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
concatenation <string> of <string>	*Named*	Combines the second set of strings into a single string, separated by the first string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
concatenation of <string>	*Plain*	Combines the supplied strings into a single string, end-to-end. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
hexadecimal string <string>	*NamedGlobal*	Creates a string from the given hexadecimal value. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
machine name	*PlainGlobal*	Returns the machine name. The value is from a long enumeration of all Mac platforms. See the header file Gestalt.h or Apple's. Mac:4.1
string <string>	*NamedGlobal*	Creates a string matching the name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<string> as boolean	*Cast*	<boolean>	Returns a boolean value for the string. All possible capitalization's of "TRUE" and "FALSE" will convert successfully. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as floating point	*Cast*	<floating point>	Converts the contents of a string into a floating point number. Win:4.1, Mac:4.1
<string> as hexadecimal	*Cast*	<string>	Converts a string to a hexadecimal number. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
<string> as integer	*Cast*	<integer>	Returns an integer value for the string provided. If the string contains anything but ASCII digits, the conversion will fail. Use numeric value for more liberal parsing rules. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
<string> as local time	*Cast*	<time>	Returns a local time object from a properly formatted string. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as lowercase	*Cast*	<string>	Returns a lowercase version of the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as string	*Cast*	<string>	Returns the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time	*Cast*	<time>	Returns a time object from a properly formatted string. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time interval	*Cast*	<time interval>	Returns a time interval object from a properly formatted string. Expects strings formatted as • ddd days, HH:MM:SS.mmmmmm. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time zone	*Cast*	<time zone>	Returns a time zone object from a properly formatted string. See <time zone>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as universal time	*Cast*	<time>	Returns a universal time object from a properly formatted string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as uppercase	*Cast*	<string>	Returns an uppercase version of the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as version	*Cast*	<version>	Returns a version if the string can be parsed as a version. The first numeric set of characters delimited with period, comma or comma-space is returned. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
character <integer> of <string>	*Numbered*	<substring>	Returns a string of length 1 made by taking the character identified by <integer> from the string. Numbering begins at zero. Example, Character 1 of "HI" is "I". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
character of <string>	*Plain*	<substring>	Returns the characters from the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
concatenation <string> of <string>	*Named*	<string>	Concatenation <string1> of <string2> concatenates a list of strings indicated by string2, placing string1 between each. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
concatenation of <string>	*Plain*	<string>	Combines the supplied strings into a single string, end-to-end. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
first <integer> of <string>	*Numbered*	<substring>	Returns a substring containing the number of characters specified from the given string. For example, First 5 of "To be or not to be" is "To be". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
first <string> of <string>	*Named*	<substring>	Returns a substring containing the first occurrence of the name provided. See substring. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last <integer> of <string>	*Numbered*	<substring>	Returns a substring containing the number of characters specified. For example, Last 5 of "To be or not to be" is "to be". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last <string> of <string>	*Named*	<substring>	Returns a substring containing the last occurrence of the name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
length of <string>	*Plain*	<integer>	Returns the number of characters in the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
position <integer> of <string>	*Numbered*	<string position>	Returns a string position pointing to the character position specified. The first character is at position 0. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
position of <string>	*Plain*	<string position>	Returns the positions of the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
substring <string> of <string>	*Named*	<substring>	Iterates through the string returning all the substrings matching the name given. For example, number of substrings "be" of "to be or not to be" = 2. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
substring after <string> of <string>	*Named*	<substring>	Returns the substrings that come after the first string delimiter. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring before <string> of <string>	*Named*	<substring>	Returns the substrings that come before the first string delimiter. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring between <string> of <string>	*Named*	<substring>	Returns the substring in the second string found between two instances of the first string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring separated by <string> of <string>	*Named*	<substring>	Returns a substring (or set of substrings) delimited by the first string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
unique value of <string>	*Plain*	<string with multiplicity>	Given a list of strings, returns the count of each unique string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Operators

Key phrase	Return Type	Description
<rope> & <string>	<*rope*>	Concatenates a rope and a string, producing a rope. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<rope> contains <string>	<*boolean*>	Returns boolean TRUE if the rope contains the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> & <rope>	<*rope*>	Concatenates a rope and a string, returning a new rope. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> & <string>	<*string*>	Concatenates two strings, producing a new string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> {cmp} <string>	<*boolean*>	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Note

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

Examples

- `floating point "600987.9" as scientific notation`
  - Returns 6.009879e+5.
- `floating point "6.009e8" as standard notation`
  - Returns 600900000.
- `255 as hexadecimal`
  - Returns the string "ff".
- `concatenation of "light" & "year"`
  - Returns "lightyear".
- `4.5 as floating point`
  - Returns 4.5.
- `exists character whose (it is "z") of "Paul Cezanne"`
  - Returns True.
- `concatenation "/" of ("a" ; "b" ; "c" )`
  - Returns "a/b/c".
- `preceding text of last "ab" of "abracadabra" is "abracad"`
  - Returns True.
- `substrings after ":" of "definition: after the colon"`
  - Returns " after the colon".
- `substrings before "<--" of "the item pointed to <--"`
  - Returns "the item pointed to".
- `substrings between "*" of "the item *between* asterisks"`
  - Returns "between".
- `substrings separated by "," of "1,2,3"`
  - Returns the list of numbers separated by commas in the specified string.

- `multiplicities of unique values of ("steak"; "chop"; "rib"; "rib"; "rib")`
  - Returns the multiplicity of (the number of times) each string in the list is used, namely, 1,3,1.

• Note that the multiplicities are based on the alphabetic order of the strings (chop, rib, steak), not their position in the list.

String Position

String position works in combination with the string and substring data types. A string position is a point within a string. It can be compared to an integer (which it is derived from), but it also acts as a pointer within a string so that the preceding and following text can be extracted. A substring (a part of a larger string) is derived from a string object.

Creation Methods

Key Phrase	*Form*	Description
end of <substring>	*Plain*	Creates an object corresponding to the position in the string of the end of the substring. For example, end of first "be" of "To be or not to be" = 5. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
position <integer> of <string>	*Numbered*	Creates an index (zero based) into the string. For example, position 5 of "to be or not to be" = 5. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
position of <string>	*Plain*	Iterates through the string returning values for all possible positions within it. For example, number of positions of "hi" = 3. Note that the positions being counted here are 0, 1, and 2. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
start of <substring>	*Plain*	Creates the position of the substring within its containing string. For example, Start of substring "or" of "to be or not to be" = 6. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
following text of <string position>	*Plain*	<substring>	Returns the substring following the position in the string. For example, following text of position 5 of "0123456789" = "567890". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
preceding text of <string position>	*Plain*	<substring>	Returns the substring preceding the position in the string. For example, preceding text of position 5 of "0123456789" = "01234". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Note

String positions can be compared and combined with integers using the same operations that are available for integers.

Examples

- `preceding text of position 5 of "Four score and seven.."`
- Returns "Four".

Substring

A substring object is derived from a string object, so it has all the properties of a string. Substrings also have these additional properties:

Creation Methods

Key Phrase	*Form*	Description
character <integer> of <string>	*Numbered*	Creates the single character substring at the position given within the string. For example, character 2 of "abc" = "c". Note that numbering begins at zero. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
character of <string>	*Plain*	Iterates through the string (or substring) returning substrings that contain the individual characters of the string. For example, number of characters of string "abc" = 3. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Description
first <integer> of <string>	*Numbered*	Creates a substring for the given number of characters at the start of the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
first <string> of <string>	*Named*	Creates an object containing the first match of the given string. For example, first "be" of "to be or not to be" = "be". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
following text of <string position>	*Plain*	Creates an object containing the substring following the position in the string. For example, following text of position 5 of "0123456789" = "567890". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
following text of <substring>	*Plain*	Creates an object containing the string following the substring. For example, following text of last "." of "log.txt" = "txt". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last <integer> of <string>	*Numbered*	Creates an object containing a substring from the last part of the string containing the number of characters specified. For example, Last 5 of "To be or not to be" is "to be". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last <string> of <string>	*Named*	Creates a substring containing the last occurrence of the name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
preceding text of <string position>	*Plain*	Creates the substring preceding the position in the string. For example, preceding text of position 5 of "0123456789" = "01234". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
preceding text of <substring>	*Plain*	Creates an object containing the string preceding the substring. For example, preceding text of last "." of "log.txt" = "log". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
substring <string> of <string>	*Named*	Iterates through the string returning all the substrings matching the name given. For example, number of substrings "be" of "to be or not to be" = 2. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Description
substring after <string> of <string>	*Named*	Returns the substrings that come after the first string delimiter. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring before <string> of <string>	*Named*	Returns the substrings that come before the first string delimiter. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring between <string> of <string>	*Named*	Returns the substring in the second string found between two instances of the first string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring separated by <string> of <string>	*Named*	Returns a substring (or set of substrings) delimited by the first string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
end of <substring>	*Plain*	<string position>	Returns the position of the substring within its containing string. For example, end of first "be" of "to be or not to be" = 5. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
following text of <substring>	*Plain*	<substring>	Returns the string following the substring. For example, following text of last "." of "log.txt" = "txt". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
preceding text of <substring>	*Plain*	<substring>	Returns the string preceding the substring. For example, preceding text of last "." of "log.txt" = "log". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
start of <substring>	*Plain*	<string position>	Returns the position within the string of the substring. For example, start of substring "or" of "to be or not to be" = 6. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Note

All the string operators can also be applied to substrings.

Examples

- `substrings after ":" of "definition: after the colon"`
  - Returns " after the colon".
- `substrings before "<--" of "the item pointed to <--"`
  - Returns "the item pointed to".
- `substrings between "*" of "the item *between* asterisks"`
  - Returns "between".
- `substrings separated by "," of "1,2,3"`
  - Returns the list of numbers separated by commas in the specified string.

String With Multiplicity

These Inspectors deal with arrays of strings, allowing you to pluck out unique strings and count them.

Creation Methods

Key Phrase	*Form*	Description
unique value of <string>	*Plain*	Given a list of strings, returns the count of each unique string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <string with multiplicity>	*Plain*	<integer>	Returns the multiplicity (quantity) of each element in a multiple string list. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Rope

The rope object is a way to efficiently concatenate long strings. String literals in the Relevence language are limited to 512 characters, but internally, they can be any length. Ropes provide a technique for concatenating string literals that is memory-efficient. In general, the Fixlet author will not need to worry about ropes, but they are useful for increasing efficiency.

Creation Methods

Key Phrase	*Form*	Description
rope <string>	*NamedGlobal*	Creates a rope object from the given string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<rope> as string	*Cast*	<string>	Converts a rope into a string object. Once converted, all the other string properties are available. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
length of <rope>	*Plain*	<integer>	Returns the number of bytes in the rope. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Operators

Key phrase	Return Type	Description
<rope> & <rope>	<*rope*>	Concatenates two ropes into a new rope. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<rope> & <string>	<*rope*>	Concatenates a rope and a string, producing a rope. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
<rope> contains <string>	<*boolean*>	Returns TRUE if the rope contains the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> & <rope>	<*rope*>	Concatenates a rope and a string, returning a new rope. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1

Array

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

Creation Methods

Key Phrase	*Form*	Description
array <integer> of <array>	*Numbered*	An array from an array by index. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
array <integer> of <array>	*Numbered*	<array>	An array from an array by index. Mac:4.1
boolean <integer> of <array>	*Numbered*	<boolean>	Get, from an array, a boolean keyed by the specified integer. Mac:4.1
date <integer> of <array>	*Numbered*	<time>	Get, from an array, a date keyed by the specified integer. Mac:4.1
dictionary <integer> of <array>	*Numbered*	<dictionary>	Get, from an array, a dictionary keyed by the specified integer. Mac:4.1
integer <integer> of <array>	*Numbered*	<integer>	Get, from an array, an integer keyed by the specified integer. Mac:4.1
size of <array>	*Plain*	<integer>	The size of the given array. Mac:4.1
string <integer> of <array>	*Numbered*	<string>	Get, from an array, a string keyed by the specified integer. Mac:4.1
value of <array>	*Plain*	<osxvalue>	Values of the array. Mac:4.1

Examples

- `strings of values of array 0 of array "com.apple.iTunes" of dictionary of file "com.apple.help.plist" of preferences folder`
  - Returns any values of type string from the array, e.g. iTunes Help or file://localhost/Applications/iTunes.app/Contents/Resources/English.lproj/iTunes.Help/.

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

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

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

Bit Set

A small, numbered collection of bits that can be examined and manipulated.

Creation Methods

Key Phrase	*Form*	Description
<integer> as bit set	*Cast*	Returns the bits of the binary representation of the integer; bit zero is the least-significant bit. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<integer> as bits	*Cast*	Returns the bits of the binary representation of the integer; bit zero is the least-significant bit. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
bit <integer>	*NumberedGlobal*	Creates a <bit set> object representing the nth bit position as specified by the integer. The integer value must be between 0 and 63 corresponding to the bit position of interest. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
bit set <string>	*NamedGlobal*	Returns the bits of the binary number given by the string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
left shift <integer> of <bit	*Numbered*	A bit set which, at each position n >= delta, holds bit

Key Phrase	*Form*	Description
set>		n-delta of the original bit set, where delta is the given integer. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
right shift <integer> of <bit set>	*Numbered*	A bit set which, at each position n, holds bit n+delta of the original bit set, where delta is the given shift integer. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
<bit set> as integer	*Cast*	<integer>	Returns the integer whose binary representation matches the bit set. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<bit set> as string	*Cast*	<string>	Returns the bits (0s and 1s) in a string format. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
bit <integer> of <bit set>	*Numbered*	<boolean>	Returns the value of the bit at the given <integer> position in the set. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
least significant one bit of <bit set>	*Plain*	<integer>	Returns the least n such that bit n of the set is true. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
left shift <integer> of <bit set>	*Numbered*	<bit set>	A bit set which, at each position n >= delta, holds bit n-delta of the original bit set, where delta is the given integer. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
most significant one bit of <bit set>	*Plain*	<integer>	Returns the greatest n such that bit n of the set is true. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
one bit of <bit set>	*Plain*	<integer>	Returns the numbers n for which bit n of the set is true. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
right shift <integer> of <bit set>	*Numbered*	<bit set>	A bit set which, at each position n, holds bit n+delta of the original bit set, where delta is the given shift integer. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Operators

Key phrase	Return Type	Description
<bit set> - <bit set>	<*bit set*>	Returns the bits that are true in the left bit set and false in the right bit set. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<bit set> * <bit set>	<*bit set*>	Returns the intersection of the two bit sets. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<bit set> + <bit set>	<*bit set*>	Returns the union of the two sets. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<bit set> = <bit set>	<*boolean*>	Returns true if the corresponding bits of the two sets are equal. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<bit set> contains <bit set>	<*boolean*>	Returns false if -- for any n -- bit n of the left set is false, but bit n of the right set is true. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Examples

- `bit 0 of 5`
- Returns TRUE.

- `bit 3 of bit 3`
- Returns TRUE.

Regular Expression

These Inspectors let you use regular expressions (or regexes) in relevance statements. They use the boost library implementation of the 'POSIX-Extended' regular expression syntax, as documented at:
• http://www.boost.org/libs/regex/doc/syntax_extended.html.

Creation Methods

Key Phrase	*Form*	Description
case insensitive regex <string>	*NamedGlobal*	Creates a case-insensitive regular expression (regex) from the specified string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
case insensitive regular expression <string>	*NamedGlobal*	Same as case insensitive regex <string>. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Description
regex <string>	*NamedGlobal*	Creates a regex object from the given string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
regular expression <string>	*NamedGlobal*	Same as regex <string>. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Operators

Key phrase	Return Type	Description
<regular expression> = <string>	<*boolean*>	Returns TRUE if the regular expression is equal to the specified string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> = <regular expression>	<*boolean*>	Returns TRUE if the regular expression is equal to the specified string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> contains <regular expression>	<*boolean*>	Returns TRUE if the specified string contains the contents of the regular expression. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> ends with <regular expression>	<*boolean*>	Returns TRUE if the string ends with the contents of the regular expression. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> starts with <regular expression>	<*boolean*>	Returns TRUE if the string starts with the contents of the regular expression. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Examples

- `regex ".+match.+" = "We will win the match tonight"`
  - Returns TRUE.

- `regex ".*PATH.*" = line 1 of file "/etc/profile"`
  - Returns TRUE if the word "PATH" exists in line 1 of the given file.

Regular Expression Match

These Inspectors let you match regular expressions (or regexes) in relevance statements. They use the boost library implementation of the 'POSIX-Extended' regular expression syntax, as documented at:
• http://www.boost.org/libs/regex/doc/syntax_extended.html. These objects are derived from substring objects.

Creation Methods

Key Phrase	*Form*	Description
first match <regular expression> of <string>	*Indexed*	Creates an object containing the first match to the regular expression in the given string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
match <regular expression> of <string>	*Indexed*	Creates an object containing all the matches to the regular expression in the given string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
parenthesized part <integer> of <regular expression match>	*Numbered*	<substring>	Returns the nth parenthetical (given by <integer>) in the specified regular expression match. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
parenthesized part of <regular expression match>	*Plain*	<substring>	Returns the parenthetical part of the specified regular expression match. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Examples

- `first match (regex "to.+") of "just too hot to handle"`
- Returns "too hot to handle".

Undefined

The "undefined" type is used as the result type of Inspectors that never return a value.

Creation Methods

Key Phrase	*Form*	Description
error <string>	*NamedGlobal*	Always fails; if an error message is generated, it is based on the given string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Examples

- `if FALSE then 1 else error "my error message"`
- Returns the string: User-defined error: my error message.

Hertz

The hertz object is useful to measure clock cycles. It is used primarily to measure clock frequency by the speed of the processor Inspector. Hertz objects have a resolution of 1 hertz and are stored internally as a 64 bit signed integer.

Creation Methods

Key Phrase	*Form*	Description
absolute value of <hertz>	*Plain*	Creates a hertz object with a positive value. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
ghz	*PlainGlobal*	Creates a hertz object corresponding to 1 giga-hertz. For example, ghz = 1000*mhz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
greatest hz	*PlainGlobal*	Creates the largest hertz object that can be represented on the current machine. It returns the value 9,223,372,036,854,775,807 hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hz	*PlainGlobal*	Creates a hertz object corresponding to 1 hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
khz	*PlainGlobal*	Creates a hertz object corresponding to 1 kilohertz. For example, khz = 1000*hz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Description
least hz	*PlainGlobal*	Creates the largest negative hertz object that can be represented on the current machine. It returns the value -9,223,372,036,854,775,808 hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
mhz	*PlainGlobal*	Creates a hertz object corresponding to 1 megahertz. For example, mhz = 1000*khz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
significant digits <integer> of <hertz>	*Numbered*	Rounds up the value of a hertz object with <integer> significant digits. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<hertz> as string	*Cast*	<string>	Returns a string formatted "##### hertz". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
absolute value of <hertz>	*Plain*	<hertz>	Returns the positive value of the hertz object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
significant digits <integer> of <hertz>	*Numbered*	<hertz>	Returns the value of a hertz object with <integer> significant digits (e.g.. significant digits 3 of 1235569 = 1240000). Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Operators

Key phrase	Return Type	Description
- <hertz>	<*hertz*>	Returns the negative of the <hertz> value. Win:2.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
<hertz> {cmp} <hertz>	<*boolean*>	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<hertz> {op} <hertz>	<*hertz*>	Returns a hertz object equal to the result of the operation, where: • {op} is one of: +, -, mod . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key phrase	Return Type	Description
<hertz> {op} <integer>	*<hertz>*	Returns a hertz object equal to the result of the operation, where: • {op} is one of: *, / . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Examples

- `speed of processor > 3*ghz`
- Returns TRUE on machines faster than 3Ghz.

- `greatest hz`
- Returns a large positive value, such as 9223372036854775807 hertz.

- `least hz`
- Returns a large negative value, such as -9223372036854775808 hertz.

- `significant digits 3 of 1235569`
- Returns 1240000.

- `speed of processor`
- Returns the speed of the processor in hz, such as 3394000000 hertz for a 3.4 GHz computer.

Time

A time object is used to identify a point in time. Time objects are used to represent important properties of objects such as the modification time of a file. You can create time objects from literal strings. The format of the string is defined by the MIME standard. The difference between two Time objects may be calculated by subtracting them and yields time intervals. Time intervals may be added or subtracted from time objects to obtain time objects.

Creation Methods

Key Phrase	*Form*	Description
<string> as local time	*Cast*	Local time creates a time object by parsing the string literal provided. The time zone is optional. If not present, the local time zone is assumed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time	*Cast*	Parses the string. Time zone information must be provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Description
<string> as universal time	*Cast*	Parses the string. If time zone is not provided in the string, the universal time zone is assumed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
maximum of <time>	*Plain*	Returns the maximum time from a list of times. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
minimum of <time>	*Plain*	Returns the minimum time from a list of times. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
now	*PlainGlobal*	Creates an object for the current time. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
time <string>	*NamedGlobal*	The time inspector creates a time object by parsing the string literal provided. The zone info is required. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
universal time <string>	*NamedGlobal*	The universal time inspector returns a time object by parsing the string literal provided. The time zone is optional. If not present, universal time is assumed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<time> as local string	*Cast*	<string>	Returns a string in MIME format of the given time object. The format is: ddd, DD mmm YYYY HH:MM:SS sZZZZ. The string is formatted using the local time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as string	*Cast*	<string>	Same as above. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as universal string	*Cast*	<string>	Returns a string in MIME format of the given time object. The format is: • ddd, DD mmm YYYY HH:MM:SS +0000 • The string is formatted using the universal time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
date of <time>	*Plain*	<date>	The year, month and day derived from the given time. Mac:4.1

Key Phrase	*Form*	Return Type	Description
day of <time>	*Plain*	<integer>	The day of the month derived from the given time. Mac:4.1
day_of_week of <time>	*Plain*	<integer>	The day of the week derived from the given time. Sunday = 0, etc. Mac:4.1
hour of <time>	*Plain*	<integer>	The hour derived from the given time. The range is from 0 to 23. Mac:4.1
maximum of <time>	*Plain*	<time>	Returns the maximum time from a list of times. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
minimum of <time>	*Plain*	<time>	Returns the minimum time from a list of times. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
minute of <time>	*Plain*	<integer>	The minute derived from the given time. The range is from 0 to 59. Mac:4.1
month of <time>	*Plain*	<integer>	The month derived from the given time. January = 1, etc. Mac:4.1
second of <time>	*Plain*	<integer>	The second derived from the given time. The range is from 0 to 59. Mac:4.1
time of <time>	*Plain*	<hours>	The hour, minute and second derived from the given time. Mac:4.1
year of <time>	*Plain*	<integer>	The year derived from the given time. Mac:4.1

Operators

Key phrase	Return Type	Description
<time interval> & <time>	*<time range>*	Concatenates a time interval with a time, returning a time range of the form time1 to time2. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time range> & <time>	*<time range>*	Concatenates a time with a time range, producing a new time range, in the form of: • <date> to <date>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time> & <time interval>	*<time range>*	Concatenates a time and a time interval, producing a time range object. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time> & <time range>	*<time range>*	Concatenates a time and a time range, producing a new time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time> & <time>	*<time range>*	Concatenates two times into a time range, with the earliest date first and the latest date last. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time> {cmp} <time>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> {op} <time interval>	*<time>*	Returns a <time> corresponding to the operator, where: • {op} is one of: +, -. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Note

The string format for a time object is given by the MIME standard. When output as a string, the format used is:
ddd, DD mmm YYYY HH:MM:SS sZZZZ
where:

ddd	The day of the week. Abbreviations are Mon, Tue, Wed, Thu, Fri, Sat, Sun.
DD	The day of the month. A leading zero will be applied to make it two characters wide.
mmm	The Month. Abbreviations are Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.
YYYY	The year.
HH	The hour of the day. It is always output at two digits. Possible values run from 0 to 23. The digits 00 are used to designate midnight.
MM	The minutes of the hour. It is always output as two digits. Possible values run from 0 to 59.
SS	The seconds of the minute.
s	A single character representing whether the time is east or west of Greenwich. The value of + means east of Greenwich while the value - means west of Greenwich.
ZZZZ	The number of minutes east or west of Greenwich.

To create a string from a literal, use the format:**ddd,DD mmm YYYY HH:MM:SS zoneinfo**
where:

ddd	The optional day of the week. Abbreviations are Mon, Tue, Wed, Thu, Fri, Sat, Sun. Case is not important in these names. If provided it must be correct. For example, time "Sat, 19 jun 1998 00:00:00 +0000" will fail since June 19, 1998 was a Friday.
DD	The day of the month. One or two digits are allowed.
mmm	The Month. Abbreviations are Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. Case is not important.
YYYY	The year. A two, three or four digit year. If two digits are given a base of 1900 is assumed.
HH	The hour of the day. It is always input at two digits. Possible values run from 0 to 23. The digits 00 are used to designate midnight.
MM	The minutes of the hour. It is always output as two digits. Possible values run from 0 to 59.
SS	The seconds of the minute. Range from 0 to 59. This is optional. If seconds are not present, the preceding colon should also not be present
Zoneinfo	The time zone information. It is provided in one of these formats: Single character + or - followed by 4 digits. The 4 digits are interpreted as HHMM two digits of hours and two digits of minutes. Plus designates east of universal time while minus designates west of universal time. Three letters for the civilian name of the time zone. cdt, edt, mdt, pdt are the designations for central, eastern, mountain and pacific daylight savings time while cst, est, mst, pst are the designations for central, eastern, mountain and pacific standard time. gmt designates Greenwich mean time. A single letter military name of the time zone. Military time zones use single letters from a to z, excepting j. a-m represent offsets from universal time of -1 to -12 hours respectively. z represents 0 offset.

Examples

- `now`
- Returns the current time.

- `the hour of the time of the creation time of the application "iTunes.app" is 12`
- Returns TRUE if the specified application was created at noon.

Time Zone

Time zones are used in conjunction with the time object. Time zones have a resolution of 1 minute.

Creation Methods

Key Phrase	*Form*	Description
<string> as time zone	*Cast*	Creates a time zone object corresponding to the string provided. For example, "pst" as time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
local time zone	*PlainGlobal*	Creates a time zone object corresponding to the local time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
time zone <string>	*NamedGlobal*	Creates a time zone object corresponding to the string provided. For example, time zone "edt" as string = "-0400". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
universal time zone	*PlainGlobal*	Creates a time zone object corresponding to the universal time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<time zone> as string	*Cast*	<string>	Returns a string corresponding to the time zone object provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Operators

Key phrase	Return Type	Description
<time zone> {op} <time interval>	*<time zone>*	Returns a time zone object offset by a time interval, where: • {op} is one of: +, - . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Examples

- `local time zone - 2 * hour`
- Returns the time zone two hours away.

Time Range

These Inspectors provide tools for dealing and calculating with time-range types, which are of the form <time> to <time>, such as Tue, 18 Apr 2006 16:46:07 -0400 to Wed, 19 Apr 2006 16:46:07 -0400

Creation Methods

Key Phrase	*Form*	Description
final part <time interval> of <time range>	*Indexed*	Returns a time range with the specified interval, but ending on the final date of the time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
initial part <time interval> of <time range>	*Indexed*	Returns a time range starting with the first date of the time range and lasting for the specified interval. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
range after <time> of <time range>	*Indexed*	Returns a new time range, starting from the specified time and continuing through the end of the original range. The time must be within the range, or an error will result. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
range before <time> of <time range>	*Indexed*	Returns a new time range, starting from the original time in the specified range and continuting to the specified time. The time must be within the range, or an error will result. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<time range> as string	*Cast*	<string>	Casts a time range as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
end of <time range>	*Plain*	<time>	Returns the end date of a time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
final part <time interval> of <time range>	*Indexed*	<time range>	Returns a time range with the specified interval, but ending on the final date of the time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
initial part <time interval> of <time range>	*Indexed*	<time range>	Returns a time range starting with the first date of the time range and lasting for the specified interval. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
length of <time range>	*Plain*	<time interval>	Returns the time interval (in days, hours, minutes, seconds) between the start and end date of a time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
range after <time> of <time range>	*Indexed*	<time range>	Returns a new time range, starting from the specified time and continuing through the end of the original range. The time must be within the range, or an error will result. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
range before <time> of <time range>	*Indexed*	<time range>	Returns a new time range, starting from the original time in the specified range and continuting to the specified time. The time must be within the range, or an error will result. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
start of <time range>	*Plain*	<time>	Returns the starting date of a time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Operators

Key phrase	Return Type	Description
<time range> & <time range>	*<time range>*	Returns the smallest range that contains both ot the specified ranges (same as <time range> + <time range>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time range> & <time>	*<time range>*	Concatenates a time with a time range, producing a new time range, in the form of: • <date> to <date>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time range> * <time range>	*<time range>*	Returns the intersection of the two specified time ranges, if one exists. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time range> + <time range>	*<time range>*	Returns the smallest range that contains both ot the specified ranges (same as <time range> & <time range>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time> & <time range>	*<time range>*	Concatenates a time and a time range, producing a new time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Time Interval

Time intervals are used in conjunction with the time object. Time intervals have a resolution of 1 microsecond.

Creation Methods

Key Phrase	*Form*	Description
<string> as time interval	*Cast*	Returns a time interval object from a properly formatted string. Expects strings formatted as • ddd days, HH:MM:SS.mmmmmm. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
absolute value of <time interval>	*Plain*	Creates the positive value of a time interval. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
day	*PlainGlobal*	Creates a time interval corresponding to 1 day. For example, 2 * day = 48 * hour. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	***Form***	**Description**
greatest time interval	*PlainGlobal*	Creates the largest time interval that can be represented on the current machine. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hour	*PlainGlobal*	Creates a time interval corresponding to 1 hour. For example, day = 24 * hour. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
least time interval	*PlainGlobal*	Creates the largest negative time interval that can be represented on the current machine. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
maximum of <time interval>	*Plain*	Returns the maximum interval from a list of time intervals. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
microsecond	*PlainGlobal*	Creates a time interval corresponding to 1 microsecond. For example, 1000 * microsecond = 1 * millisecond. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
millisecond	*PlainGlobal*	Creates a time interval corresponding to 1 millisecond. For example, 1000 * millisecond = 1 * second. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
minimum of <time interval>	*Plain*	Returns the minimum interval from a list of time intervals. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
minute	*PlainGlobal*	Creates a time interval corresponding to 1 minute. For example, minute = 60 * second. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
second	*PlainGlobal*	Creates a time interval corresponding to 1 second. For example, 1000000 * microsecond = second. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
time interval <string>	*NamedGlobal*	Creates a time interval from the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
week	*PlainGlobal*	Creates a time interval corresponding to 1 week. For example, 7*day = 1*week. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<time interval> as string	*Cast*	<string>	Returns a string formatted as • ddd days, HH:MM:SS.mmmmmm • For example, millisecond as string = " 00:00:00.001". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
absolute value of <time interval>	*Plain*	<time interval>	Returns positive value of the time interval. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
maximum of <time interval>	*Plain*	<time interval>	Returns the maximum interval from a list of time intervals. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
minimum of <time interval>	*Plain*	<time interval>	Returns the minimum interval from a list of time intervals. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Operators

Key phrase	Return Type	Description
- <time interval>	*<time interval>*	The negative of a time interval. Win:2.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
<time interval> & <time>	*<time range>*	Concatenates a time interval with a time, returning a time range of the form time1 to time2. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time interval> {op} <integer>	*<time interval>*	Creates a time interval calculated as an integer operation on another time interval, where: • {op} is one of: *, / . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time interval> {op} <time interval>	*<time interval>*	Returns a calculated time interval, where: • {op} is one of: +, -, mod, / . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time interval> {op} <time zone>	*<time interval>*	Returns a calculated time interval, where: • {op} is one of: +, - . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> & <time interval>	*<time range>*	Concatenates a time and a time interval, producing a time range object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Hours

These Inspectors examine parts of a time field.

Creation Methods

Key Phrase	*Form*	Description
time of <time>	*Plain*	The hour, minute and second derived from the given time. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
hour of <hours>	*Plain*	<integer>	The hour portion of a time. Mac:4.1
minute of <hours>	*Plain*	<integer>	The minute portion of a time. Mac:4.1
second of <hours>	*Plain*	<integer>	The second portion of a time. Mac:4.1

Operators

Key phrase	Return Type	Description
<hours> {cmp} <hours>	<*boolean*>	Compares two hours, where: • {cmp} is one of: =, <, <=. Mac:4.1

Examples

- `hour of now`

  Returns the hour of the current time.

- `minute of now`

  Returns the minute of the current time.

- `second of now`

  Returns the second of the current time.

Date

These are the various Inspectors that access the date types.

Creation Methods

Key Phrase	*Form*	Description
date of <time>	*Plain*	The year, month and day derived from the given time. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
day of <date>	*Plain*	<integer>	The day of the month derived from the given date. Mac:4.1
day_of_week of <date>	*Plain*	<integer>	The day of the week, derived from the given date. Sunday=0, etc. • Note: This Inspector returns an integer on the Macintosh, but a <day of week> type (Monday, Tuesday, etc.) on all other operating systems. Mac:4.1
month of <date>	*Plain*	<integer>	The month derived from the given date. January = 1, etc. • Note: This is a Mac-only result. All other Oses return a <month> type (January, February, etc.). Mac:4.1
year of <date>	*Plain*	<integer>	Returns the year (as an integer), extracted from the given date. • Note: This inspector returns an <integer> on the Macintosh only. On other operating systems, it returns a <year>. Mac:4.1

Operators

Key phrase	Return Type	Description
<date> {cmp} <date>	<*boolean*>	Compares two dates, where: • {cmp} is one of: =, <, <=. Mac:4.1

World Objects

World

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

Properties

Key Phrase	*Form*	Return Type	Description
action	*PlainGlobal*	<action>	Returns the action currently being parsed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
action <integer>	*NumberedGlobal*	<action>	Returns the action matching the <integer> id. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
action lock state	*PlainGlobal*	<action lock state>	Returns the client action lock state. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
active action	*PlainGlobal*	<action>	Returns the action currently executing. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
application	*PlainGlobal*	<filesystem object>	Returns a list of the applications that have registered with LaunchServices. Mac:5.1
application <file signature>	*Global*	<filesystem object>	This is an iterated property that requires a name. It designates one or more files which are applications known to the Macintosh Desktop Database. The name will be the four character OSType called the creator. Mac:4.1

Key Phrase	*Form*	Return Type	Description
application <string>	*NamedGlobal*	<application>	Returns an application for the name provided. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
application usage summary	*PlainGlobal*	<application usage summary>	Returns an application usage summary containing information including the start time, duration and other statistics on client applications. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
application usage summary <string>	*NamedGlobal*	<application usage summary>	Returns the usage summary for the application specified in <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
binary operator <string>	*NamedGlobal*	<binary operator>	Typically used in the plural, returns the various possible binary inspectors that use the specified operators. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
binary operator returning <type>	*IndexedGlobal*	<binary operator>	Returns a list of binary operators that return the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
bit <integer>	*NumberedGlobal*	<bit set>	Returns TRUE or FALSE, corresponding to value of the bit specified by <integer>. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
bit set <string>	*NamedGlobal*	<bit set>	Returns the bits of the binary number given by the string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
boolean <string>	*NamedGlobal*	<boolean>	Returns a boolean. For example, boolean "TRUE". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
bundle <string>	*NamedGlobal*	<bundle>	Returns a bundle (CFBundle) by name. Mac:4.1
carbon process	*PlainGlobal*	<carbon proc>	Returns a list of the processes currently running. Mac:5.1
carbon process <integer>	*NumberedGlobal*	<carbon proc>	Returns the process object corresponding to the given integer process ID (pid). Mac:5.1

Key Phrase	*Form*	Return Type	Description
case insensitive regex <string>	*NamedGlobal*	<regular expression>	Returns a case-insensitive regular expression from the supplied string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
case insensitive regular expression <string>	*NamedGlobal*	<regular expression>	Same as case insensitive regex <string>. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
cast <string>	*NamedGlobal*	<cast>	Returns a list of the objects that can be cast into the type specified by <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
cast returning <type>	*IndexedGlobal*	<cast>	Returns a list of the objects that can be cast into the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
character <integer>	*NumberedGlobal*	<string>	Returns a string containing a single ASCII character. For example, character 90 = "Z". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
classic domain	*PlainGlobal*	<domain>	Returns a classic domain object. Mac:5.1
client	*PlainGlobal*	<client>	Returns the client object corresponding to the BigFix application evaluating the current relevance expression. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
client license	*PlainGlobal*	<license>	Global object containing client licensing information. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
component	*PlainGlobal*	<component>	Is an iterated property. The MacOS supports software "components", for example QuickTime codecs. This iterator can examine the components that are available. Mac:4.1
computer	*PlainGlobal*	<computer>	Refers to the computer itself. Mac:4.1
computer id	*PlainGlobal*	<integer>	This is a unique integer assigned to the computer by the BES system. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
computer name	*PlainGlobal*	<string>	Returns a string corresponding to the name of the computer as it appears on the network. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
country <string>	*NamedGlobal*	<country>	Maps a country script string (e.g. verAfrikaans) to its region code. Mac:4.1
cpu speed	*PlainGlobal*	<integer>	Returns the clock speed of the cpu in Hz. Mac:4.1
current site	*PlainGlobal*	<site>	Returns the current site object. See site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
current user	*PlainGlobal*	<current user>	Returns the current user if one is logged in to the desktop. • Note: For Unix, this returns a <user> type as of BES version 6.0. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
day	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 day. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
default web browser	*PlainGlobal*	<filesystem object>	Returns the application currently associated with HTML files. This is a Windows and Macintosh Inspector; it will fail gracefully under other operating systems rather than generate an error. Mac:5.1
dns name	*PlainGlobal*	<string>	Returns the DNS name of the computer. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
drive	*PlainGlobal*	<volume>	Returns the volume corresponding to the drive. Typically used to return a list of the drives (volumes, filesystems) on the client computer. Drives, volumes and filesystems are treated the same on the Macintosh and return a <volume> type. Mac:6.0
drive <integer>	*NumberedGlobal*	<volume>	Returns the volume corresponding to the numbered drive. Mac:6.0
drive <string>	*NamedGlobal*	<volume>	Returns the volume corresponding to the named drive. Mac:6.0
environment	*PlainGlobal*	<environment>	Returns an object corresponding to the currently defined set of environment variables. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
error <string>	*NamedGlobal*	<undefined>	Always fails; if an error message is generated, it is based on the given string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
false	*PlainGlobal*	<boolean>	Returns the boolean FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
file <string>	*NamedGlobal*	<file>	Returns a filesystem object corresponding to the full pathname provided in <string>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
file signature <string>	*NamedGlobal*	<file signature>	Turn a string into a file signature (four letters, e.g. FNDR). Mac:4.1
file type <string>	*NamedGlobal*	<file type>	The phrase 'file type' can be used to create a file type object from a string. Mac:4.1
filesystem	*PlainGlobal*	<volume>	Returns the volume corresponding to the filesystem. Typically used to return a list of the filesystems (drives, volumes) on the client computer. Drives, volumes and filesystems are treated the same on the Macintosh and return a <volume> type. Mac:6.0

Key Phrase	*Form*	Return Type	Description
filesystem <integer>	*NumberedGlobal*	<volume>	Returns the volume corresponding to the numbered filesystem. Mac:6.0
filesystem <string>	*NamedGlobal*	<volume>	Returns the volume corresponding to the named filesystem. Mac:6.0
floating point <string>	*NamedGlobal*	<floating point>	Creates a floating point number from the provided string. Win:4.1, Mac:4.1
folder <string>	*NamedGlobal*	<folder>	Returns a folder object for the name provided. See drive. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
full volume	*PlainGlobal*	<sound volume>	The maximum possible sound volume. Mac:4.1
gestalt <string>	*NamedGlobal*	<integer>	This reads a 32 bit integer from the MacOS. The selector name is the four character OSType that chooses which item is being inspected. The interpretation of the result depends on the selector. It might represent an integer or a version, for instance. Mac:4.1
ghz	*PlainGlobal*	<hertz>	Returns a Hertz object corresponding to 1 giga-hertz. See hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
greatest hz	*PlainGlobal*	<hertz>	Returns the largest hertz object that can be represented on this machine. See hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
greatest integer	*PlainGlobal*	<integer>	Returns the largest integer that can be represented on this machine. See integer. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
greatest time interval	*PlainGlobal*	<time interval>	Returns the greatest time interval representable. The value corresponds to 106751991 days, 04:00:54.775807. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
hexadecimal integer <string>	*NamedGlobal*	<integer>	Creates an integer from the provided hexadecimal value. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
hexadecimal string <string>	*NamedGlobal*	<string>	Creates a string from the given hexadecimal value. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
hfs file <string>	*NamedGlobal*	<file>	Returns the hierachical file system (HFS) file specified by <string>. Mac:5.1
hfs folder <string>	*NamedGlobal*	<folder>	Returns the hierachical file system (HFS) folder specified by <string>. Mac:5.1
hostname	*PlainGlobal*	<string>	Returns the standard host name, usually for the computer's network. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hour	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 hour. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hz	*PlainGlobal*	<hertz>	Returns a hertz object corresponding to 1 hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
integer <integer>	*NumberedGlobal*	<integer>	Returns an integer. The keyword is optional. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
integer <string>	*NamedGlobal*	<integer>	Returns integer for name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
internet plugins folder	*PlainGlobal*	<folder>	Returns the folder object corresponding to the internet plugins. Mac:5.1
iokit registry	*PlainGlobal*	<registryroot>	Returns the root of the registry; the place to start with the registry. Mac:4.1
ip address <string>	*NamedGlobal*	<ip address>	Returns the internal form of an ip address from the given string. Mac:4.1

Key Phrase	*Form*	Return Type	Description
ipv4 address <string>	*NamedGlobal*	<ipv4 address>	Returns an ip address for the string provided. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
khz	*PlainGlobal*	<hertz>	Returns a hertz object corresponding to 1 kilohertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
least hz	*PlainGlobal*	<hertz>	Returns the least hertz value that can be represented on this machine. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
least integer	*PlainGlobal*	<integer>	Returns the least integer value that can be represented on this machine. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
least time interval	*PlainGlobal*	<time interval>	Returns the least time interval that can be represented on this machine. The value corresponds to -106751991 days, 04:00:54.775808. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
left component <stereo volume>	*Global*	<sound volume>	Sound volume of the left channel. Mac:4.1
local domain	*PlainGlobal*	<domain>	Returns a local domain object. Mac:5.1
local time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
local time zone	*PlainGlobal*	<time zone>	Returns a time zone object corresponding to the local time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
logical ram	*PlainGlobal*	<integer>	Returns the amount of logical ram in the computer, physical plus virtual. Mac:4.1
machine name	*PlainGlobal*	<string>	Returns the machine name. The value is from a long enumeration of all Mac platforms. See the header file Gestalt.h or Apple's Website. Mac:4.1

Key Phrase	*Form*	Return Type	Description
main gather service	*PlainGlobal*	<nothing>	Returns "no such object" -- included for compatibility with other operating systems. Mac:5.1
main processor	*PlainGlobal*	<processor>	Returns the processor object corresponding to the main processor. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
mhz	*PlainGlobal*	<hertz>	Returns a hertz object corresponding to 1 megahertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
microsecond	*PlainGlobal*	<time interval>	Returns a time interval corresponding to .000001 seconds. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
millisecond	*PlainGlobal*	<time interval>	Returns a time interval corresponding to .001 seconds. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
minute	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 minute. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
network	*PlainGlobal*	<network>	Returns an object containing properties of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
network domain	*PlainGlobal*	<domain>	Returns a network domain object. Mac:5.1
now	*PlainGlobal*	<time>	Returns the current time as a time object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
nubus map	*PlainGlobal*	<integer>	Returns the nubus map. Mac:4.1
on appropriate disk domain	*PlainGlobal*	<domain>	Returns one of the Macintosh domains. In most cases, this is the equivalent of kOnAppropriateDisk. On Mac OS X, this constant is used instead of the constant kOnSytemDisk to indicate any disk. For more information, see the Apple documentation on Carbon domain constants. Mac:5.1

Key Phrase	*Form*	Return Type	Description
on system disk domain	*PlainGlobal*	<domain>	Returns the OnSystemDisk domain. Mac:5.1
operating system	*PlainGlobal*	<operating system>	Returns the operating system object. See operating system. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
pending restart	*PlainGlobal*	<boolean>	Returns TRUE if the operating system indicates that a restart needs to occur. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
physical ram	*PlainGlobal*	<integer>	Returns the amount of physical ram in the computer. Mac:4.1
posix file <string>	*NamedGlobal*	<file>	Returns the POSIX file specified by <string>. Mac:5.1
posix folder <string>	*NamedGlobal*	<folder>	Returns the POSIX folder specified by <string>. Mac:5.1
powerpc	*PlainGlobal*	<boolean>	Returns TRUE if the cpu is a PowerPC, FALSE if it is a 68000 chip. Mac:4.1
preference <string>	*NamedGlobal*	<preference>	The named set of preferences. Mac:4.1
primary internet connection	*PlainGlobal*	<network ip interface>	This contains information about the current internet connection. Mac:5.1

Key Phrase	*Form*	Return Type	Description
process	*PlainGlobal*	<process>	Returns all process objects currently running. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
process <integer>	*NumberedGlobal*	<process>	Returns the process object corresponding to the given integer pid. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
processor	*PlainGlobal*	<processor>	Returns all the processor objects defined on the machine. See processor. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
property <string>	*NamedGlobal*	<property>	Typically used in the plural, returns the "line" Inspector properties. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property returning <type>	*IndexedGlobal*	<property>	Produces a list of the Inspector properties that return the specified <type>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
quickdraw version	*PlainGlobal*	<version>	Returns the version of QuickDraw installed. Mac:4.1
ram	*PlainGlobal*	<ram>	Returns a ram object for inspecting the properties of Random Access Memory installed on the machine. See ram. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
regapp	*PlainGlobal*	<filesystem object>	Returns all the applications registered with LaunchServices, and is included for compatibility with Windows (same as application). Mac:5.1
regapp <string>	*NamedGlobal*	<filesystem object>	Returns an application object for the name provided. On a Macintosh, returns a file that has been registered with Launch Services. See application and regapp. Mac:5.1
regex <string>	*NamedGlobal*	<regular expression>	Creates a regex object from the given string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
registered application <string>	*NamedGlobal*	<filesystem object>	Returns the named application (which must be registered with LaunchServices). Same as application. Mac:4.1

Key Phrase	*Form*	Return Type	Description
registry	*PlainGlobal*	<dummy type>	Windows only, but included for compatibility with other operating systems. Mac:5.1
regular expression <string>	*NamedGlobal*	<regular expression>	Same as regex <string>. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
relay service	*PlainGlobal*	<nothing>	Returns FALSE. Included for compatibility with Windows Inspectors. Mac:5.1
rom version	*PlainGlobal*	<version>	Returns the version of the system ROM. Mac:4.1
rope <string>	*NamedGlobal*	<rope>	Creates a rope object from the given string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
running application	*PlainGlobal*	<filesystem object>	This Inspector returns the applications that are currently running. This is basically the same as carbon processes, and returns a <filesystem object> on the Macintosh, but an <application> on Windows and Linux. Mac:5.1
scsibus	*PlainGlobal*	<scsibus>	An iterated property. When used without a number and not iterated it means SCSI bus 0. Mac:4.1
scsibus <integer>	*NumberedGlobal*	<scsibus>	Returns a SCSI bus with the given number. Mac:4.1
scsidevice	*PlainGlobal*	<scsidevice>	An iterated property. It is derived from calls to the MacOS. Mac:4.1
scsidevice <integer>	*NumberedGlobal*	<scsidevice>	Returns a SCSI device with the given number. Mac:4.1
second	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 second. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
selected server	*PlainGlobal*	<selected server>	The BES Server or BES Relay to which the agent reports. Returned as the "selected server" type. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
service <string>	*NamedGlobal*	<dummy>	A dummy inspector to provide compatibility with Windows and Unix. Mac:5.1
site	*PlainGlobal*	<site>	Returns all the site objects that are currently loaded into memory. See site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site <string>	*NamedGlobal*	<site>	Returns a site object for the name provided. The name is the URL of the site location. See site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
string <string>	*NamedGlobal*	<string>	Returns a string for the name provided. The keyword string is optional. For example, string "hi" = "hi". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
system beep volume	*PlainGlobal*	<stereo volume>	The system beep volume. Mac:4.1
system domain	*PlainGlobal*	<domain>	Returns a system domain object. Mac:5.1
system version	*PlainGlobal*	<version>	Returns the version of MacOS. Mac:4.1
time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
time interval <string>	*NamedGlobal*	<time interval>	Creates a time interval from the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
time zone <string>	*NamedGlobal*	<time zone>	Returns a time zone object for the name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
true	*PlainGlobal*	<boolean>	Returns the boolean TRUE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
unary operator <string>	*NamedGlobal*	<unary operator>	Typically used in the plural, this inspector returns a list of objects that use the specified operator. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Key Phrase	*Form*	Return Type	Description
unary operator returning <type>	*IndexedGlobal*	<unary operator>	Returns a list of the unary operator inspectors (such as negative) that return the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
universal time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
universal time zone	*PlainGlobal*	<time zone>	Returns a time zone object corresponding to the universal time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
user domain	*PlainGlobal*	<domain>	Returns a user domain object. • NOTE: The "user domain" refers to the root user, not the currently logged in user. Mac:5.1
version <string>	*NamedGlobal*	<version>	Short hand for 'file version'. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
virtual memory	*PlainGlobal*	<boolean>	Returns TRUE if virtual memory is turned on. Mac:4.1
volume	*PlainGlobal*	<volume>	An iterated property. Examines all currently mounted volumes which will include the startup volume, CD-ROM, disk images and other removable media and file sharing volumes. Mac:4.1
volume <integer>	*NumberedGlobal*	<volume>	A numbered property returning the specified volume. Mac:4.1
volume <string>	*NamedGlobal*	<volume>	A numbered property returning the specified volume. Mac:4.1
week	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 week. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
zero volume	*PlainGlobal*	<sound volume>	The quietest sound volume. Mac:4.1

Examples

- `exists application "iTunes.app"`
  - Returns TRUE if there is at least one copy of iTunes known to the system.

- `bit 0 of 5`
  - Returns TRUE.

- `bit set "101" as integer`
  - Returns 5.

- `bundle "com.apple.BigFixEnterprise"`
  - Returns a bundle to hand to something that needs one. It doesn't print out, per se.

- `name of computer`
  - Returns the name of the computer, e.g. "MacBES".

- `exists (country "verChina")`
  - Returns TRUE if there exists a Chinese script string.

- `cpu speed > 2000000000`
  - Returns TRUE if the CPU speed is greater than two gigahertz.

- `drives`
  - Returns a list of drives, eg., /, /Users/MyUserName, etc.

- `if FALSE then 1 else error "my error message"`
  - Returns the string: User-defined error: my error message.

- `filesystems`
  - Returns a list of filesystems, eg., /, /Users/MyUserName, etc.

- `gestalt "ram" > 33554432`
  - Returns TRUE if the installed RAM is greater than the amount specified.

- `hexadecimal integer "A0"`
  - Returns 160.

- `address of primary internet connection is ip address "33.33.33.33"`
  - Compares the primary internet address to the specified string.

- `physical ram > 33554432`
  - Returns TRUE if there is more than the specified amount of RAM.

- `exists scsidevices whose (vendor of it is "Apple") of scsibuses`
  - Returns TRUE if an Apple SCSI device exists on the SCSI bus.

- `exists scsidevice whose (vendor of it is "Apple") of scsibus 1`
  - Here the scsibus is specified and the scsidevice is iterated over vendors.

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.

Filesystem Object

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

Creation Methods

Key Phrase	*Form*	Description
application	*PlainGlobal*	Returns a list of the applications that have registered with LaunchServices. Mac:5.1
default web browser	*PlainGlobal*	Returns the application (typically the browser) currently associated with HTML files. Mac:5.1
file of <carbon proc>	*Plain*	Returns a filesystem object corresponding to the given process. Mac:5.1
item <string> of <folder>	*Named*	Returns the named item (file or folder) from the specified folder. Mac:5.1
item ending in <string> of <folder>	*Named*	Returns a list of items (files/folders) ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for “*xxxx”. Mac:6.0
item of <folder>	*Plain*	Returns a list of the items (file or folder) in the specified folder. Mac:4.1
regapp	*PlainGlobal*	Returns all the applications registered with LaunchServices, and is included for compatibility with Windows (same as application). Mac:5.1

Key Phrase	*Form*	Description
regapp <string>	*NamedGlobal*	Returns an application object for the name provided. On a Macintosh, returns a file that has been registered with Launch Services. See application and regapp. Mac:4.1
registered application <string>	*NamedGlobal*	Returns the named application (which must be registered with LaunchServices). Same as application. Mac:4.1
running application	*PlainGlobal*	This Inspector returns the applications that are currently running. This is basically the same as carbon processes. Mac:5.1
sibling item <string> of <filesystem object>	*Named*	The named sibling of a filesystem object (file, folder). Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
<filesystem object> as file	*Cast*	<file>	Returns a file or nothing (if, for example, the filesystem object was a folder). Mac:4.1
<filesystem object> as folder	*Cast*	<folder>	Returns a folder or nothing. Mac:4.1
<filesystem object> as string	*Cast*	<string>	Casts a filesystem object as a string. Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
ancestor of <filesystem object>	*Plain*	<folder>	Returns all ancestor folders (recursive parent folders) of the given filesystem object (file or folder). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
backup time of <filesystem object>	*Plain*	<time>	The date of the last backup of the specified filesystem object (such as a file or folder). This corresponds to what is shown in the "Get Info" box for this object. Mac:4.1
bundle version of <filesystem object>	*Plain*	<version>	Returns the version of the filesystem object corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac:5.1

Key Phrase	*Form*	Return Type	Description
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:6.0, Mac:4.1
hfs path of <filesystem object>	*Plain*	<string>	Returns the path to a filesystem object in HFS terms (colons as delimiters). Mac:4.1
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
name of <filesystem object>	*Plain*	<string>	This returns the name of the file or folder. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
parent folder of <filesystem object>	*Plain*	<folder>	The folder containing the specified file or folder. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
pathname of <filesystem object>	*Plain*	<string>	Returns the full pathname of the specified file or folder (filesystem object) as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
posix path of <filesystem object>	*Plain*	<string>	The POSIX file path for the file or folder. Mac:4.1
sibling file <string> of <filesystem object>	*Named*	<file>	A named property. A file in the same folder as the specified file object. Mac:4.1
sibling folder <string> of <filesystem object>	*Named*	<folder>	A named property. A folder in the same folder as the specified file object. Mac:4.1
sibling item <string> of <filesystem object>	*Named*	<filesystem object>	The named sibling of a filesystem object (file, folder). Mac:5.1
version of <filesystem object>	*Plain*	<version>	This returns the version information from "vers" resource 1 of the file. It is usually present in applications, and may exist in data files as well. It corresponds to what appears in the "Get Info" box for the specified filesystem object. Mac:4.1

Operators

Examples

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

- `application "iTunes.app" as file`
  - Fails because that is in fact a folder.

- `creation time of file "System" of System Folder > time "3 jan 1998 00:00+0000"`
  - Returns TRUE if the creation time of the system file is newer than the specified date.

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

- `posix paths of items whose (name of it starts with "i") of applications folder`
  - Returns a list of the paths of applications starting with "i", such as /Applications/iCal.app, /Applications/iChat.app or /Applications/iTunes.app.

- `exists sibling file "iTunes.app" of application "iChat.app"`
  - Returns TRUE if both applications are in the same folder.

- `version of primary application "ttxt" is greater than "1.3"`
  - Returns TRUE if the default application for "ttxt" is more recent than version 1.3.

File

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

Creation Methods

See application objects for additional creation methods

Key Phrase	*Form*	Description
<filesystem object> as file	*Cast*	Returns a file or nothing (if, for example, the filesystem object was a folder). Mac:4.1
descendant of <folder>	*Plain*	Returns a list of all the descendant files of the specified folder. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:6.0
file <string>	*NamedGlobal*	Returns a filesystem object corresponding to the full pathname provided in <string>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
file <string> of <folder>	*Named*	Creates the file objects corresponding to the named file within the folder. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
file ending in <string> of <folder>	*Named*	Returns a list of files ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for “*xxxx”. Mac:6.0
file of <folder>	*Plain*	Iterates through the files of a folder. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hfs file <string>	*NamedGlobal*	Returns the hierachical file system (HFS) file specified by <string>. Mac:5.1
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
posix file <string>	*NamedGlobal*	Returns the POSIX file specified by <string>. Mac:5.1

Key Phrase	*Form*	Description
relative file <string> of <folder>	*Named*	Returns the file with the path specified by <string> relative to the given <folder>. Mac:5.1
relative hfs file <string> of <folder>	*Named*	Returns the HFS file with the path specified by <string> relative to the given <folder>. Mac:5.1
relative posix file <string> of <folder>	*Named*	Returns the POSIX file with the path specified by <string> relative to the given <folder>. Mac:5.1
sibling file <string> of <filesystem object>	*Named*	A named property. A file in the same folder as the specified file object. Mac:4.1

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.

Properties

Key Phrase	*Form*	Return Type	Description
alias of <file>	*Plain*	<boolean>	Returns TRUE if the file is an alias for another file. Mac:4.1
byte <integer> of <file>	*Numbered*	<integer>	Returns the numeric value of the byte located at the offset specified by number within the file. Byte 0 of the file is the first byte. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
content of <file>	*Plain*	<file content>	Returns an object that can be used to search for a string in the file. See content. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
data fork of <file>	*Plain*	<datafork>	Returns information about the data fork of the specified file. Mac:4.1

Key Phrase	*Form*	Return Type	Description
dictionary of <file>	*Plain*	<dictionary>	Returns the dictionary object for the specified file. Mac:4.1
drive of <file>	*Plain*	<volume>	Returns the drive (as a <volume>) associated with the specified file. This is a Macintosh-only Inspector. Mac:5.1
filesystem of <file>	*Plain*	<volume>	Returns the volume corresponding to the filesystem of the specified file. Mac:6.0
key <string> of <file>	*Named*	<string>	Returns a key and its value from the given structured text file. It iterates over lines that start with the key name (as specified by <string>) followed by an = or : character. When searching, white space is ignored. Win:4.1, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
length of <file>	*Plain*	<integer>	The total length of the data and resource forks. Mac:4.1
line <integer> of <file>	*Numbered*	<file line>	Returns the nth line (specified by <integer>) from the given file. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
line containing <string> of <file>	*Named*	<file line>	Returns all lines from the given file that contain the specified string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
line of <file>	*Plain*	<file line>	Iterates over all the lines of the specified file. NOTE: lines are truncated to 1023 characters. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
line starting with <string> of <file>	*Named*	<file line>	Same as line <string> of <file>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
locked of <file>	*Plain*	<boolean>	Returns TRUE if the file is locked. Mac:4.1
resource fork of <file>	*Plain*	<resfork>	Returns information about the resource fork of the file. Mac:4.1

Key Phrase	*Form*	Return Type	Description
section <string> of <file>	*Named*	<file section>	Returns a named section of a file. Useful for locating sections of 'ini' files. Section names are delimited by square bracket characters '[section name]'. See examples below. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
sha1 of <file>	*Plain*	<string>	Returns the sha1 checksum of the file hex encoded as a 40 character long string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
size of <file>	*Plain*	<integer>	Returns the size in bytes of a file. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
version <integer> of <file>	*Numbered*	<version>	Returns the nth version information from the "vers" resource of the given file. Typically n=1, but other information may be stored in "vers" resources greater than 1. Mac:4.1
version of <file>	*Plain*	<version>	Synonym for file version of <file>. Win:1.2, Mac:4.1
visible of <file>	*Plain*	<boolean>	Returns TRUE if the file is visible. Mac:4.1
volume of <file>	*Plain*	<volume>	The volume containing the file. Mac:4.1

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. Consider using the “find file” Inspector which allows you to filter set of files by using the wildcard.

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

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

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

- `relative file "Safari.app/Contents/MacOS/Safari” of applications folder`
  - Returns the concatenation of the specified folder and the given path, /Applications/Safari.app/Contents/MacOS/Safari.

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

- `Length of data fork of file "Microsoft Word" of folder "Microsoft Office X" of Applications folder is greater than 100000`
  - Returns TRUE if the data fork of the system file is greater than the specified amount.

- `locked of file "this file" of folder "this folder"`
  - Returns TRUE if the specified file is locked.

- `version 0 of file "fname" = "1.0"`
  - Checks for the zero version, if one exists.

- `volume of (application "iTunes.app" as folder) is volume of system folder`
  - Returns TRUE if iTunes is installed on the same volume as the system folder.

Folder

For every folder that exists in the file system, you can create a folder object. The properties of this object allow you to examine many aspects of the system. Folder objects are derived from filesystem 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.

Creation Methods

Key Phrase	*Form*	Description
<filesystem object> as folder	*Cast*	Returns a folder or nothing. Mac:4.1
ancestor of <filesystem object>	*Plain*	Returns all ancestor folders (recursive parent folders) of the given filesystem object (file or folder). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
apple extras folder of <domain>	*Plain*	Returns the apple extras folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
apple menu items folder of <domain>	*Plain*	Returns the apple menu items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
application support folder of <domain>	*Plain*	Returns the application support folder of the specified OS X domain, typically /Library/Application Support. If the domain is not specified, it defaults to the system domain. Mac:5.1
applications folder of <domain>	*Plain*	Returns the applications folder of the specified OS X domain, typically /Applications. If the domain is not specified, it defaults to the system domain. Mac:5.1
assistants folder of <domain>	*Plain*	Returns the assistants folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
audio folder of <domain>	*Plain*	Returns the audio folder of the specified OS X domain, typically /Library/Audio. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
cache folder of <domain>	*Plain*	Returns the cache folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
carbon folder of <domain>	*Plain*	Returns the carbon folder of the specified OS X domain, typically /Library/Carbon. If the domain is not specified, it defaults to the system domain. Mac:5.1
chewable items folder of <domain>	*Plain*	Returns the chewable items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
classic folder of <domain>	*Plain*	Returns the classic folder of the specified OS X domain, typically the /System Folder. If the domain is not specified, it defaults to the system domain. Mac:5.1
client folder of <site>	*Plain*	Creates an object corresponding to the folder on the client where site data is gathered. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
color sync folder of <domain>	*Plain*	Returns the color sync folder of the specified OS X domain, typically /System/Library/ColorSync. If the domain is not specified, it defaults to the system domain. Mac:5.1
colorsync profiles folder of <domain>	*Plain*	Returns the colorsync profiles folder of the specified OS X domain, typically /System/Library/ColorSync/Profiles. If the domain is not specified, it defaults to the system domain. Mac:5.1
component folder of <domain>	*Plain*	Returns the component folder of the specified OS X domain, typically /System/Library/Components. If the domain is not specified, it defaults to the system domain. Mac:5.1
contextual menu items folder of <domain>	*Plain*	Returns the contextual menu items folder of the specified OS X domain, typically /Library/Contextual Menu Items. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
control panels folder of <domain>	*Plain*	Returns the control panels folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
control strip modules folder of <domain>	*Plain*	Returns the control strip modules folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
core services folder of <domain>	*Plain*	Returns the core services folder of the specified OS X domain, typically /System/Library/CoreServices. If the domain is not specified, it defaults to the system domain. Mac:5.1
current user folder of <domain>	*Plain*	Returns the current user folder of the specified OS X domain, typically found at /Users/username. If the domain is not specified, it defaults to the system domain. Mac:5.1
desktop folder of <domain>	*Plain*	Returns the desktop folder of the specified OS X domain, typically /Users/Username/Desktop. If the domain is not specified, it defaults to the system domain. Mac:5.1
developer docs folder of <domain>	*Plain*	Returns the developer docs folder of the specified OS X domain, typically found at /Developer/Documentation. If the domain is not specified, it defaults to the system domain. Mac:5.1
developer folder of <domain>	*Plain*	Returns the developer folder of the specified OS X domain, typically found at /Developer. If the domain is not specified, it defaults to the system domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
developer help folder of <domain>	*Plain*	Returns the help folder of the specified OS X domain, typically /Developer/Documentation/Help. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
disabled control panels folder of <domain>	*Plain*	Returns the control panels folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled extensions folder of <domain>	*Plain*	Returns the extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled shutdown items folder of <domain>	*Plain*	Returns the shutdown folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled startup items folder of <domain>	*Plain*	Returns the startup items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled system extensions folder of <domain>	*Plain*	Returns the systems extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
documentation folder of <domain>	*Plain*	Returns the documentation folder for the given OS X domain, typically found at /Library/Documentation. If the domain is not specified, it defaults to the system domain. Mac:5.1
documents folder of <domain>	*Plain*	Returns the documents folder for the specified OS X domain, typically found at /User/Username/Documents. If the domain is not specified, it defaults to the system domain. Mac:5.1
domain library folder of <domain>	*Plain*	Returns the domain library folder of the specified OS X domain, typically found at /Library. If the domain is not specified, it defaults to the system domain. Mac:5.1
domain top folder of <domain>	*Plain*	Returns the top folder of the specified OS X domain, typically found at /System. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
download folder of <domain>	*Plain*	Returns the download folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
extensions folder of <domain>	*Plain*	Returns the extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
favorites folder of <domain>	*Plain*	Returns the favorites folder of the specified OS X domain, typically /Users/username/Library/Favorites. If the domain is not specified, it defaults to the system domain. Mac:5.1
folder <string>	*NamedGlobal*	Creates a folder object for the named folder. This is a global property. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
folder <string> of <folder>	*Named*	Creates a folder object for the named sub-folder. Trailing slashes should be omitted from the name. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
folder ending in <string> of <folder>	*Named*	Returns a list of folders ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for “*xxxx”. Mac:6.0
folder of <folder>	*Plain*	Iterates through the sub-folders of the folder object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
fonts folder of <domain>	*Plain*	Returns the font folder of the specified OS X domain, typically /System/Library/Fonts. If the domain is not specified, it defaults to the system domain. Mac:5.1
framework <string> of <domain>	*Named*	Returns a folder of the form "/<domain>/Library/Frameworks/<string>.framework”. • Note: This Inspector appends .framework for you, so don't provide it. The framework inspector needs a domain, and without it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
framework folder of <domain>	*Plain*	Returns the framework folder of the specified OS X domain, typically /System/Library/Frameworks. If the domain is not specified, it defaults to the system domain. Mac:5.1
help folder of <domain>	*Plain*	Returns the help folder of the specified OS X domain, typically /Library/Documentation/Help. If the domain is not specified, it defaults to the system domain. Mac:5.1
hfs folder <string>	*NamedGlobal*	Returns the hierachical file system (HFS) folder specified by <string>. Mac:5.1
internet plugins folder	*PlainGlobal*	Returns the folder object corresponding to the internet plugins. Mac:5.1
internet plugins folder of <domain>	*Plain*	Returns the internet plugins folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
kernel extensions folder of <domain>	*Plain*	Returns the kernel extensions folder of the specified OS X domain, typically /System/Library/Extensions. If the domain is not specified, it defaults to the system domain. Mac:5.1
locales folder of <domain>	*Plain*	Returns the locales folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
location manager modules folder of <domain>	*Plain*	Returns the location manager modules folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
location manager preferences folder of <domain>	*Plain*	Returns the location manager preferences folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
locations folder of <domain>	*Plain*	Returns the locations folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
macos read me folder of <domain>	*Plain*	Returns the Mac OS read me folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
modem scripts folder of <domain>	*Plain*	Returns the modem scripts folder of the specified OS X domain, typically /System/Library/Modem Scripts. If the domain is not specified, it defaults to the system domain. Mac:5.1
parent folder of <filesystem object>	*Plain*	The folder containing the specified file or folder. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
posix folder <string>	*NamedGlobal*	Returns the POSIX folder specified by <string>. Mac:5.1
preferences folder of <domain>	*Plain*	Returns the preferences folder of the specified OS X domain, typically /Users/username/Library/Preferences. If the domain is not specified, it defaults to the system domain. Mac:5.1
printer descriptions folder of <domain>	*Plain*	Returns the printer descriptions folder of the specified OS X domain, typically /System/Library/Printers/PPDs. If the domain is not specified, it defaults to the system domain. Mac:5.1
printer drivers folder of <domain>	*Plain*	Returns the printer drivers folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
printers folder of <domain>	*Plain*	Returns the printers folder of the specified OS X domain, typically /System/Library/Printers. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
printmonitor documents folder of <domain>	*Plain*	Returns the printmonitor documents folder of the specified OS X domain, typically /Library/Printers/PrintMonitor Documents. If the domain is not specified, it defaults to the system domain. Mac:5.1
private framework folder of <domain>	*Plain*	Returns the private framework folder of the specified OS X domain, typically /System/Library/PrivateFrameworks. If the domain is not specified, it defaults to the system domain. Mac:5.1
quicktime folder of <domain>	*Plain*	Returns the quicktime folder of the specified OS X domain, typically /System/Library/QuickTime. If the domain is not specified, it defaults to the system domain. Mac:5.1
receipts folder of <domain>	*Plain*	Returns the receipts folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
relative folder <string> of <folder>	*Named*	Returns the folder with the path specified by <string> relative to the given <folder>. Mac:5.1
relative hfs folder <string> of <folder>	*Named*	Returns the HFS folder with the path specified by <string> relative to the given <folder>. Mac:5.1
relative posix folder <string> of <folder>	*Named*	Returns the POSIX folder with the path specified by <string> relative to the given <folder>. Mac:5.1
scripting additions folder of <domain>	*Plain*	Returns the scripting additions folder of the specified OS X domain, typically /System/Library/Scripting Additions. If the domain is not specified, it defaults to the system domain. Mac:5.1
shared folder of <domain>	*Plain*	Returns the shared folder of the specified OS X domain, typically /Users/Shared. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
shared libraries folder of <domain>	*Plain*	Returns the shared libraries folder of the specified OS X domain, typically /System/Library/CFMSupport. If the domain is not specified, it defaults to the system domain. Mac:5.1
shutdown items folder of <domain>	*Plain*	Returns the shutdown items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
sibling folder <string> of <filesystem object>	*Named*	A named property. A folder in the same folder as the specified file object. Mac:4.1
sound folder of <domain>	*Plain*	Returns the sound folder of the specified OS X domain, typically /System/Library/Sound. If the domain is not specified, it defaults to the system domain. Mac:5.1
speech folder of <domain>	*Plain*	Returns the speech folder of the specified OS X domain, typically /System/Library/Speech. If the domain is not specified, it defaults to the system domain. Mac:5.1
startup items folder of <domain>	*Plain*	Returns the startup items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
system folder of <domain>	*Plain*	Returns the system folder of the specified OS X domain, typically /System. If the domain is not specified, it defaults to the system domain. Mac:5.1
temporary items folder of <domain>	*Plain*	Returns the temporary items folder of the specified OS X domain, typically /private/tmp/. If the domain is not specified, it defaults to the system domain. Mac:5.1
text encodings folder of <domain>	*Plain*	Returns the text encodings folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Description
themes folder of <domain>	*Plain*	Returns the themes folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
user temp folder of <domain>	*Plain*	Returns the user temp folder of the specified OS X domain, typically /private/tmp/uid where uid is the user ID number. If the domain is not specified, it defaults to the system domain. Mac:5.1
users folder of <domain>	*Plain*	Returns the users folder of the specified OS X domain, typically /Users. If the domain is not specified, it defaults to the system domain. Mac:5.1
utilities folder of <domain>	*Plain*	Returns the utilities folder of the specified OS X domain, typically /Applications/Utilities. If the domain is not specified, it defaults to the system domain. Mac:5.1
voices folder of <domain>	*Plain*	Returns the voices folder of the specified OS X domain, typically /System/Library/Speech/Voices. If the domain is not specified, it defaults to the system domain. Mac:5.1
volume settings folder of <domain>	*Plain*	Returns the volume settings folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
bundle of <folder>	*Plain*	<bundle>	Returns a bundle from a folder (if it has one like application folders do). Mac:4.1
bundle version of <folder>	*Plain*	<version>	Returns the version of the folder corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac:5.1

Key Phrase	*Form*	Return Type	Description
descendant of <folder>	*Plain*	<file>	Returns a list of all the descendant files of the specified folder. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:6.0
drive of <folder>	*Plain*	<volume>	Returns the drive associated with the specified folder. Mac:5.1
file <string> of <folder>	*Named*	<file>	Returns a file object for the named file located in the folder. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
file ending in <string> of <folder>	*Named*	<file>	Returns a list of files ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for “*xxxx”. Mac:6.0
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
filesystem of <folder>	*Plain*	<volume>	Returns the volume corresponding to the filesystem of the specified folder. Mac:6.0
folder <string> of <folder>	*Named*	<folder>	Returns a folder object for the named sub-folder. Trailing slashes should be omitted from the name. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
folder ending in <string> of <folder>	*Named*	<folder>	Returns a list of folders ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for “*xxxx”. Mac:6.0
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
item <string> of <folder>	*Named*	<filesystem object>	Returns the named item (file or folder) from the specified folder. Mac:5.1

Key Phrase	*Form*	Return Type	Description
item ending in <string> of <folder>	*Named*	<filesystem object>	Returns a list of items (files/folders) ending in "xxxx". Typically used to identify a dotted extension. Equivalent to a wildcard search for “*xxxx”. Mac:6.0
item of <folder>	*Plain*	<filesystem object>	Returns a list of the items (file or folder) in the specified folder. Mac:4.1
relative file <string> of <folder>	*Named*	<file>	Returns the file with the path specified by <string> relative to the given <folder>. Mac:5.1
relative folder <string> of <folder>	*Named*	<folder>	Returns the folder with the path specified by <string> relative to the given <folder>. Mac:5.1
relative hfs file <string> of <folder>	*Named*	<file>	Returns the HFS file with the path specified by <string> relative to the given <folder>. Mac:5.1
relative hfs folder <string> of <folder>	*Named*	<folder>	Returns the HFS folder with the path specified by <string> relative to the given <folder>. Mac:5.1
relative posix file <string> of <folder>	*Named*	<file>	Returns the POSIX file with the path specified by <string> relative to the given <folder>. Mac:5.1
relative posix folder <string> of <folder>	*Named*	<folder>	Returns the POSIX folder with the path specified by <string> relative to the given <folder>. Mac:5.1
version of <folder>	*Plain*	<version>	Returns a version for a folder representing an application. Mac:4.1
volume of <folder>	*Plain*	<volume>	The volume containing the specified folder. Mac:4.1

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 very careful not to iterate through folders that contain lots of files. Counting files in folders that contains lots of files can be slow. Always try to use the most efficient techniques to minimize the client overhead.

Examples

- `color sync folder of system domain`
  - Returns /System/Library/ColorSync.
- `color sync folder of local domain`
  - Returns /Library/ColorSync.
- `application support folder of user domain`
  - Returns /var/root/Library/Application Support.
- `framework "MyPrivate" of local domain`
  - Returns /Library/Frameworks/MyPrivate.framework.
- `parent folder of application "Terminal.app" is Utilities folder`
  - TRUE if the specified application is in the Utilities folder.
- `exists (bundle of applications folder)`
  - Typically returns TRUE.
- `files ending in ".a" of folder "/usr/lib"`
  - Returns a list of files with the specified ending, such as: /usr/lib/libcpp_kext.a, /usr/lib/libfl.a, /usr/lib/libioabc.a, etc.

Volume

The following Inspectors refer to the mounted drive volumes.

Creation Methods

Key Phrase	*Form*	Description
drive	*PlainGlobal*	Returns the volume corresponding to the drive. Typically used to return a list of the drives (volumes, filesystems) on the client computer. Drives, volumes and filesystems are treated the same on the Macintosh and return a <volume> type. Mac:6.0
drive <integer>	*NumberedGlobal*	Returns the volume corresponding to the numbered drive. Mac:6.0
drive <string>	*NamedGlobal*	Returns the volume corresponding to the named drive. Mac:6.0
drive of <file>	*Plain*	Returns the drive associated with the specified file. Mac:5.1
drive of <folder>	*Plain*	Returns the drive associated with the specified folder. Mac:5.1
filesystem	*PlainGlobal*	Returns the volume corresponding to the filesystem. Typically used to return a list of the filesystems (drives, volumes) on the client computer. Drives, volumes and filesystems are treated the same on the Macintosh and all return a <volume> type. Mac:6.0
filesystem <integer>	*NumberedGlobal*	Returns the volume corresponding to the numbered filesystem. Mac:6.0
filesystem <string>	*NamedGlobal*	Returns the volume corresponding to the named filesystem. Mac:6.0
filesystem of <file>	*Plain*	Returns the filesystem (volume or drive) containing the specified file. Drive, volume and filesystem are treated the same on the Mac. Mac:6.0

Key Phrase	*Form*	Description
filesystem of <folder>	*Plain*	Returns the filesystem (volume or drive) containing the specified folder. Drive, volume and filesystem are treated the same on the Mac. Mac:6.0
volume	*PlainGlobal*	An iterated property. Examines all currently mounted volumes which will include the startup volume, CD-ROM, floppies and other removable media and file sharing volumes. Mac:4.1
volume <integer>	*NumberedGlobal*	A numbered property. Volume 1 is the startup volume. Others follow in sequence. Mac:4.1
volume <string>	*NamedGlobal*	A named property representing a volume. Mac:4.1
volume of <file>	*Plain*	The volume containing the file. Mac:4.1
volume of <folder>	*Plain*	The volume containing the specified folder. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
allocation block count of <volume>	*Plain*	<integer>	The number of allocation blocks, used or free, on the volume. Mac:4.1
directory count of <volume>	*Plain*	<integer>	The number of directories on the volume. Mac:4.1
file count of <volume>	*Plain*	<integer>	The number of files on the volume. Mac:4.1
flag of <volume>	*Plain*	<integer>	Returns the Mac-specific volume flags.More info on these flags can be found at the Apple developer site, under Carbon > Reference > File Manager. Mac:6.0

Key Phrase	*Form*	Return Type	Description
free percent of <volume>	*Plain*	<integer>	Returns the percentage of room available on the specified volume. Mac:6.0
free space of <volume>	*Plain*	<integer>	The number of free bytes on the volume. Mac:4.1
init date of <volume>	*Plain*	<time>	Returns the initialization date of a disk volume. Mac:4.1
modification time of <volume>	*Plain*	<time>	Creates a time object corresponding to the time the volume was last modified. Mac:5.1
name of <volume>	*Plain*	<string>	The name of the volume. Mac:4.1
size of <volume>	*Plain*	<integer>	Returns a number corresponding to the total number of bytes (used and unused) on the specified volume. Mac:6.0
total space of <volume>	*Plain*	<integer>	The total amount of space, used and free, on the volume. Mac:4.1
used percent of <volume>	*Plain*	<integer>	Returns the used percentage of room on the specified volume (or drive or filesystem). Mac:6.0
used space of <volume>	*Plain*	<integer>	Returns a number corresponding to the used bytes on the specified volume. Mac:6.0

Operators

Key phrase	Return Type	Description
<volume> = <volume>	<*boolean*>	Compare two volumes. Mac:4.1

Examples

- `drives`
  - Returns a list of drives, eg., /, /Users/MyUserName, etc.

- `filesystems`
  - Returns a list of filesystems, eg., /, /Users/MyUserName, etc.

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

- `free percentage of volume 1`
  - Returns a number between 0 and 100.

- `init date of volume of system folder`
  - Returns the date the system folder was initialized.

- `size of volume 1`
  - Returns the size of the specified volume in bytes, eg. 159697911808.

- `used percentage of volume 2`
  - Returns a number between 0 and 100.

File Section

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

Creation Methods

Key Phrase	*Form*	Description
section <string> of <file>	*Named*	Creates a file section for the name given. A case-insensitive search is performed to locate the named section in the file. Searching through files for configuration data can be a slow process. Particularly for large ini files. In this case you may want to find another method that requires less computation. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
key <string> of <file section>	*Named*	<string>	Returns a string containing the value for the name provided. A case-insensitive search is performed through the section of the file. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0

Note

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

File Content

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

Creation Methods

Key Phrase	*Form*	Description
<file content> as lowercase	*Cast*	Returns the contents of the file as lower case characters. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<file content> as uppercase	*Cast*	Returns the contents of the file as upper case characters. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
content of <file>	*Plain*	Creates a content object for a file. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<file content> as lowercase	*Cast*	<file content>	Returns a lowercase version of the content provided. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<file content> as uppercase	*Cast*	<file content>	Returns an uppercase version of the content provided. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Operators

Key phrase	Return Type	Description
<file content> contains <string>	*<boolean>*	Returns TRUE if the string is located in the content provided. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Note

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

Examples

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

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

Version

This is the numeric method of indicating the file version, which is compact, convenient and fast. It makes use of a short string to define the version number.

Creation Methods

Key Phrase	*Form*	Description
<string> as version	*Cast*	Turns a string into a version object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
bundle version of <bundle>	*Plain*	Returns the version of the bundle corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac:5.1
bundle version of <filesystem object>	*Plain*	Returns the version of the filesystem object corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac:5.1
bundle version of <folder>	*Plain*	Returns the version of the folder corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac:5.1
quickdraw version	*PlainGlobal*	Returns the version of QuickDraw installed. Mac:4.1
release of <operating system>	*Plain*	The full version number of the operating system, of the form <major>.<minor>.<bug revision>. Mac:4.1
rom version	*PlainGlobal*	Returns the version of the system ROM. Mac:4.1

Key Phrase	*Form*	Description
system version	*PlainGlobal*	Returns the version of MacOS. Mac:4.1
version <integer> of <file>	*Numbered*	Returns the nth version information from the "vers" resource of the given file. Typically n=1, but other information may be stored in "vers" resources greater than 1. Mac:4.1
version <string>	*NamedGlobal*	Creates a version object corresponding to the name provided. Syntax: version "1.2". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
version of <bundle>	*Plain*	Version of the given bundle. Mac:4.1
version of <component>	*Plain*	The version of the component, as determined by the component maker. Mac:4.1
version of <file>	*Plain*	Shorthand for file version of <file>. Win:1.2, Mac:4.1
version of <filesystem object>	*Plain*	This returns the version information from "vers" resource 1 of the file. It is usually present in applications, and may exist in data files as well. It corresponds to what appears in the "Get Info" box for the specified filesystem object. Mac:4.1
version of <folder>	*Plain*	Returns a version for a folder representing an application. Mac:4.1
version of <scsibus>	*Plain*	Version of the SCSI bus. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<version> as string	*Cast*	<string>	Turns a version type into a string of the form "1.2.3.4". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
bug revision of <version>	*Plain*	<integer>	If the stage is present the bug revision is the number after the stage. If absent, zero is implied. Mac:4.1
major revision of <version>	*Plain*	<integer>	The number before the first period in the version string. Mac:4.1
minor revision of <version>	*Plain*	<integer>	The number immediately after the first period in the version string. If absent, zero is implied. Mac:4.1

Operators

Key phrase	Return Type	Description
<version> {cmp} <string>	<*boolean*>	Returns a boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<version> {cmp} <version>	<*boolean*>	Returns a boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

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

- `release of operating system`
- Returns a version number, e.g. 10.3.8.

- `version of bundle of folder "iTunes.app" of applications folder`
  - Returns the version of the specified folder bundle, such as 4.2.

- `major revision of version of file "name" > 4`
  - Returns TRUE if the major revision number is greater than the specified number.

Domain

Mac OS X defines several file system domains to control access to system resources on multi-user systems. These include the User, Local, Network, Classic and System domains. The domain for a given resource or folder determines its accessibility to the user. For example, while a user-installed font is only available to that user, an administrator-installed font is available to all network users. The following Inspectors allow folder access to be parceled out according to domain.
• NOTE: The "user domain" refers to the root user, not the currently logged in user.

Creation Methods

Key Phrase	*Form*	Description
classic domain	*PlainGlobal*	Returns a classic domain object. Mac:5.1
local domain	*PlainGlobal*	Returns a local domain object. Mac:5.1
network domain	*PlainGlobal*	Returns a network domain object. Mac:5.1
on appropriate disk domain	*PlainGlobal*	Returns one of the Macintosh domains. In most cases, this is the equivalent of kOnAppropriateDisk. On Mac OS X, this constant is used instead of the constant kOnSytemDisk to indicate any disk. For more information, see the Apple documentation on Carbon domain constants. Mac:5.1
on system disk domain	*PlainGlobal*	Returns the OnSystemDisk domain. Mac:5.1
system domain	*PlainGlobal*	Returns a system domain object. Mac:5.1
user domain	*PlainGlobal*	Returns a user domain object. • NOTE: The "user domain" refers to the root user, not the currently logged in user. Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
apple extras folder of <domain>	*Plain*	<folder>	Returns the apple extras folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
apple menu items folder of <domain>	*Plain*	<folder>	Returns the apple menu items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
application support folder of <domain>	*Plain*	<folder>	Returns the application support folder of the specified OS X domain, typically /Library/Application Support. If the domain is not specified, it defaults to the system domain. Mac:5.1
applications folder of <domain>	*Plain*	<folder>	Returns the applications folder of the specified OS X domain, typically /Applications. If the domain is not specified, it defaults to the system domain. Mac:5.1
assistants folder of <domain>	*Plain*	<folder>	Returns the assistants folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
audio folder of <domain>	*Plain*	<folder>	Returns the audio folder of the specified OS X domain, typically /Library/Audio. If the domain is not specified, it defaults to the system domain. Mac:5.1
cache folder of <domain>	*Plain*	<folder>	Returns the cache folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
carbon folder of <domain>	*Plain*	<folder>	Returns the carbon folder of the specified OS X domain, typically /Library/Carbon. If the domain is not specified, it defaults to the system domain. Mac:5.1
chewable items folder of <domain>	*Plain*	<folder>	Returns the chewable items folder of the specified OS X domain. If the domain is not specified, it

Key Phrase	*Form*	Return Type	Description
			defaults to the system domain. Mac:5.1
classic folder of <domain>	*Plain*	<folder>	Returns the classic folder of the specified OS X domain, typically the /System Folder. If the domain is not specified, it defaults to the system domain. Mac:5.1
color sync folder of <domain>	*Plain*	<folder>	Returns the color sync folder of the specified OS X domain, typically /System/Library/ColorSync. If the domain is not specified, it defaults to the system domain. Mac:5.1
colorsync profiles folder of <domain>	*Plain*	<folder>	Returns the colorsync profiles folder of the specified OS X domain, typically /System/Library/ColorSync/Profiles. If the domain is not specified, it defaults to the system domain. Mac:5.1
component folder of <domain>	*Plain*	<folder>	Returns the component folder of the specified OS X domain, typically /System/Library/Components. If the domain is not specified, it defaults to the system domain. Mac:5.1
contextual menu items folder of <domain>	*Plain*	<folder>	Returns the contextual menu items folder of the specified OS X domain, typically /Library/Contextual Menu Items. If the domain is not specified, it defaults to the system domain. Mac:5.1
control panels folder of <domain>	*Plain*	<folder>	Returns the control panels folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
control strip modules folder of <domain>	*Plain*	<folder>	Returns the control strip modules folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
core services folder of <domain>	*Plain*	<folder>	Returns the core services folder of the specified OS X domain, typically /System/Library/CoreServices.

Key Phrase	*Form*	Return Type	Description
			If the domain is not specified, it defaults to the system domain. Mac:5.1
current user folder of <domain>	*Plain*	<folder>	Returns the current user folder of the specified OS X domain, typically found at /Users/username. If the domain is not specified, it defaults to the system domain. Mac:5.1
desktop folder of <domain>	*Plain*	<folder>	Returns the desktop folder of the specified OS X domain, typically /Users/Username/Desktop. If the domain is not specified, it defaults to the system domain. Mac:5.1
developer docs folder of <domain>	*Plain*	<folder>	Returns the developer docs folder of the specified OS X domain, typically found at /Developer/Documentation. If the domain is not specified, it defaults to the system domain. Mac:5.1
developer folder of <domain>	*Plain*	<folder>	Returns the developer folder of the specified OS X domain, typically found at /Developer. If the domain is not specified, it defaults to the system domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
developer help folder of <domain>	*Plain*	<folder>	Returns the help folder of the specified OS X domain, typically /Developer/Documentation/Help. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled control panels folder of <domain>	*Plain*	<folder>	Returns the control panels folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled extensions folder of <domain>	*Plain*	<folder>	Returns the extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled shutdown	*Plain*	<folder>	Returns the shutdown folder of the specified OS X

Key Phrase	*Form*	Return Type	Description
items folder of <domain>			domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled startup items folder of <domain>	*Plain*	<folder>	Returns the startup items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
disabled system extensions folder of <domain>	*Plain*	<folder>	Returns the systems extensions folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
documentation folder of <domain>	*Plain*	<folder>	Returns the documentation folder for the given OS X domain, typically found at /Library/Documentation. If the domain is not specified, it defaults to the system domain. Mac:5.1
documents folder of <domain>	*Plain*	<folder>	Returns the documents folder for the specified OS X domain, typically found at /User/Username/Documents. If the domain is not specified, it defaults to the system domain. Mac:5.1
domain library folder of <domain>	*Plain*	<folder>	Returns the domain library folder of the specified OS X domain, typically found at /Library. If the domain is not specified, it defaults to the system domain. Mac:5.1
domain top folder of <domain>	*Plain*	<folder>	Returns the top folder of the specified OS X domain, typically found at /System. If the domain is not specified, it defaults to the system domain. Mac:5.1
download folder of <domain>	*Plain*	<folder>	Returns the download folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
extensions folder of	*Plain*	<folder>	Returns the extensions folder of the specified OS X

Key Phrase	*Form*	Return Type	Description
<domain>			domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
favorites folder of <domain>	*Plain*	<folder>	Returns the favorites folder of the specified OS X domain, typically /Users/username/Library/Favorites. If the domain is not specified, it defaults to the system domain. Mac:5.1
fonts folder of <domain>	*Plain*	<folder>	Returns the font folder of the specified OS X domain, typically /System/Library/Fonts. If the domain is not specified, it defaults to the system domain. Mac:5.1
framework <string> of <domain>	*Named*	<folder>	Returns a folder of the form "/System/Library/Frameworks/<string>.framework”. • Note: This Inspector appends .framework for you, so don't provide it. The framework inspector needs a domain, and without it defaults to the system domain. Mac:5.1
framework folder of <domain>	*Plain*	<folder>	Returns the framework folder of the specified OS X domain, typically /System/Library/Frameworks. If the domain is not specified, it defaults to the system domain. Mac:5.1
help folder of <domain>	*Plain*	<folder>	Returns the help folder of the specified OS X domain, typically /Library/Documentation/Help. If the domain is not specified, it defaults to the system domain. Mac:5.1
internet plugins folder of <domain>	*Plain*	<folder>	Returns the internet plugins folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
kernel extensions folder of <domain>	*Plain*	<folder>	Returns the kernel extensions folder of the specified OS X domain, typically /System/Library/Extensions. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Return Type	Description
locales folder of <domain>	*Plain*	<folder>	Returns the locales folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
location manager modules folder of <domain>	*Plain*	<folder>	Returns the location manager modules folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
location manager preferences folder of <domain>	*Plain*	<folder>	Returns the location manager preferences folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
locations folder of <domain>	*Plain*	<folder>	Returns the locations folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
macos read me folder of <domain>	*Plain*	<folder>	Returns the Mac OS read me folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
modem scripts folder of <domain>	*Plain*	<folder>	Returns the modem scripts folder of the specified OS X domain, typically /System/Library/Modem Scripts. If the domain is not specified, it defaults to the system domain. Mac:5.1
preferences folder of <domain>	*Plain*	<folder>	Returns the preferences folder of the specified OS X domain, typically /Users/username/Library/Preferences. If the domain is not specified, it defaults to the system domain. Mac:5.1
printer descriptions folder of <domain>	*Plain*	<folder>	Returns the printer descriptions folder of the specified OS X domain, typically /System/Library/Printers/PPDs. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Return Type	Description
printer drivers folder of <domain>	*Plain*	<folder>	Returns the printer drivers folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
printers folder of <domain>	*Plain*	<folder>	Returns the printers folder of the specified OS X domain, typically /System/Library/Printers. If the domain is not specified, it defaults to the system domain. Mac:5.1
printmonitor documents folder of <domain>	*Plain*	<folder>	Returns the printmonitor documents folder of the specified OS X domain, typically /Library/Printers/PrintMonitor Documents. If the domain is not specified, it defaults to the system domain. Mac:5.1
private framework folder of <domain>	*Plain*	<folder>	Returns the private framework folder of the specified OS X domain, typically /System/Library/PrivateFrameworks. If the domain is not specified, it defaults to the system domain. Mac:5.1
quicktime folder of <domain>	*Plain*	<folder>	Returns the quicktime folder of the specified OS X domain, typically /System/Library/QuickTime. If the domain is not specified, it defaults to the system domain. Mac:5.1
receipts folder of <domain>	*Plain*	<folder>	Returns the receipts folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
scripting additions folder of <domain>	*Plain*	<folder>	Returns the scripting additions folder of the specified OS X domain, typically /System/Library/Scripting Additions. If the domain is not specified, it defaults to the system domain. Mac:5.1
shared folder of <domain>	*Plain*	<folder>	Returns the shared folder of the specified OS X domain, typically /Users/Shared. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Return Type	Description
shared libraries folder of <domain>	*Plain*	<folder>	Returns the shared libraries folder of the specified OS X domain, typically /System/Library/CFMSupport. If the domain is not specified, it defaults to the system domain. Mac:5.1
shutdown items folder of <domain>	*Plain*	<folder>	Returns the shutdown items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
sound folder of <domain>	*Plain*	<folder>	Returns the sound folder of the specified OS X domain, typically /System/Library/Sound. If the domain is not specified, it defaults to the system domain. Mac:5.1
speech folder of <domain>	*Plain*	<folder>	Returns the speech folder of the specified OS X domain, typically /System/Library/Speech. If the domain is not specified, it defaults to the system domain. Mac:5.1
startup items folder of <domain>	*Plain*	<folder>	Returns the startup items folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
system folder of <domain>	*Plain*	<folder>	Returns the system folder of the specified OS X domain, typically /System. If the domain is not specified, it defaults to the system domain. Mac:5.1
temporary items folder of <domain>	*Plain*	<folder>	Returns the temporary items folder of the specified OS X domain, typically /private/tmp/. If the domain is not specified, it defaults to the system domain. Mac:5.1
text encodings folder of <domain>	*Plain*	<folder>	Returns the text encodings folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1
themes folder of <domain>	*Plain*	<folder>	Returns the themes folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1

Key Phrase	*Form*	Return Type	Description
user temp folder of <domain>	*Plain*	<folder>	Returns the user temp folder of the specified OS X domain, typically /private/tmp/uid where uid is the user ID number. If the domain is not specified, it defaults to the system domain. Mac:5.1
users folder of <domain>	*Plain*	<folder>	Returns the users folder of the specified OS X domain, typically /Users. If the domain is not specified, it defaults to the system domain. Mac:5.1
utilities folder of <domain>	*Plain*	<folder>	Returns the utilities folder of the specified OS X domain, typically /Applications/Utilities. If the domain is not specified, it defaults to the system domain. Mac:5.1
voices folder of <domain>	*Plain*	<folder>	Returns the voices folder of the specified OS X domain, typically /System/Library/Speech/Voices. If the domain is not specified, it defaults to the system domain. Mac:5.1
volume settings folder of <domain>	*Plain*	<folder>	Returns the volume settings folder of the specified OS X domain. If the domain is not specified, it defaults to the system domain. Mac:5.1

Bundle

A bundle object is used to inspect software installations as managed by the "software depot."

Creation Methods

Key Phrase	*Form*	Description
bundle <string>	*NamedGlobal*	Returns a bundle (CFBundle) by name. Mac:4.1
bundle of <folder>	*Plain*	Returns a bundle from a folder (if it has one like application folders do). Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
bundle version of <bundle>	*Plain*	<version>	Returns the version of the bundle corresponding to the CFBundleVersion string, as distinct from the CFBundleShortVersionString. Mac:5.1
creator of <bundle>	*Plain*	<file signature>	The creator 4-letter code of the bundle (e.g., FNDR for Finder). Mac:4.1
global dictionary of <bundle>	*Plain*	<dictionary>	The bundle's information dictionary. Mac:4.1
local dictionary of <bundle>	*Plain*	<dictionary>	The bundle’s localized information dictionary. Mac:4.1
type of <bundle>	*Plain*	<file type>	Returns the file type of the bundle. Can be used only to compare equality between file types. Mac:4.1
version of <bundle>	*Plain*	<version>	Version of the given bundle. Mac:4.1

Examples

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

  Returns TRUE if the specified types are the same.

File Type

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

Creation Methods

Key Phrase	*Form*	Description
file type <string>	*NamedGlobal*	The phrase 'file type' can be used to create a file type object from a string. Mac:4.1
type of <bundle>	*Plain*	Returns the file type of the bundle. Can be used only to compare equality between file types. Mac:4.1
type of <file>	*Plain*	Refers to the file type of the specified file. Mac:4.1

Operators

Key phrase	Return Type	Description
<file type> = <file type>	*<boolean>*	Compare two file types. Mac:4.1

Examples

- `creator of "that file" of "that folder" is file type "ttxt"`

  Returns TRUE if "that file" is a TeachText file.

- `type of "that file" is type of "this file"`

  Compares the types of two files.

File Signature

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

Creation Methods

Key Phrase	*Form*	Description
creator of <bundle>	*Plain*	The creator 4-letter code of the bundle (e.g., FNDR for Finder). Mac:4.1
creator of <file>	*Plain*	This refers to the four-character identifier used in all MacOS files. The creator tells what application should open the specified file. Mac:4.1

Operators

Examples

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

Component

Components are handled by the MacOS Component Manager. These Inspectors provide access to the various software components available through the Component Manager. The information derived from a component is placed there by its maker. What it means is determined by the maker.

Creation Methods

Key Phrase	*Form*	Description
component	*PlainGlobal*	Is an iterated property. The MacOS supports software "components", for example QuickTime codecs. This iterator can examine the components that are available. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
info of <component>	*Plain*	<string>	Returns information about the component, as determined by the creator of the component. Mac:4.1
maker of <component>	*Plain*	<string>	Returns the name of the maker of the specified component. Mac:4.1
name of <component>	*Plain*	<string>	The name of the component, as determined by its maker. Mac:4.1
subtype of <component>	*Plain*	<string>	The subtype of the component, as determined by the component maker. Mac:4.1
type of <component>	*Plain*	<string>	The type of the component, as determined by the component maker. Mac:4.1
version of <component>	*Plain*	<version>	The version of the component, as determined by the component maker. Mac:4.1

Datafork

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

Creation Methods

Key Phrase	*Form*	Description
data fork of <file>	*Plain*	Returns information about the data fork of the specified file. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
length of <datafork>	*Plain*	<integer>	The logical length of the data fork of the file. Mac:4.1
size of <datafork>	*Plain*	<integer>	Returns the size of the specified datafork. Mac:5.1

Resfork

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

Creation Methods

Key Phrase	*Form*	Description
resource fork of <file>	*Plain*	Returns information about the resource fork of the file. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
length of <resfork>	*Plain*	<integer>	The logical length of the resource fork of the file. Mac:4.1
size of <resfork>	*Plain*	<integer>	Returns the size of the resource fork. Mac:5.1

Dictionary

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

Creation Methods

Key Phrase	*Form*	Description
dictionary <integer> of <array>	*Numbered*	Get, from an array, a dictionary keyed by the specified integer. Mac:4.1
dictionary <string> of <dictionary>	*Named*	Get, from a dictionary, a dictionary keyed by the specified string. Mac:4.1
dictionary <string> of <preference>	*Named*	Get, from a preference, a dictionary keyed by the specified string. Mac:4.1
dictionary of <file>	*Plain*	Returns the dictionary object for the specified file, if it exists. Mac:4.1
dictionary of <osxvalue>	*Plain*	Cast the osxvalue (essentially untyped) to a dictionary. Mac:4.1
dictionary of <registrynode>	*Plain*	Returns a dictionary from a node in the IORegistry. Mac:4.1
dictionary of <registryroot>	*Plain*	Returns a dictionary from the root of the registry. Mac:4.1
global dictionary of <bundle>	*Plain*	The bundle's information dictionary. Mac:4.1
local dictionary of <bundle>	*Plain*	The bundle’s localized information dictionary. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
array <string> of <dictionary>	*Named*	<array>	Get, from a dictionary, an array keyed by the specified string. Mac:4.1
boolean <string> of <dictionary>	*Named*	<boolean>	Get, from a dictionary, a boolean keyed by the string. Mac:4.1
cstring <string> of <dictionary>	*Named*	<string>	Get, from a dictionary, the string keyed by the specified string. Works for objects in a dictionary of type String or Data. If the type is Data, any null terminator will be stripped. Mac:6.0
data <string> of <dictionary>	*Named*	<string>	Returns the specified dictionary data as a string. Mac:6.0
date <string> of <dictionary>	*Named*	<time>	Get, from a dictionary, a date keyed by the specified string. Mac:4.1
dictionary <string> of <dictionary>	*Named*	<dictionary>	Get, from a dictionary, a dictionary keyed by the specified string. Mac:4.1
entry of <dictionary>	*Plain*	<dictionaryentry>	A key-value pair of a dictionary. Mac:4.1
integer <string> of <dictionary>	*Named*	<integer>	Get, from a dictionary, an integer keyed by the specified string. Mac:4.1
key of <dictionary>	*Plain*	<string>	The keys of a dictionary. Mac:4.1
size of <dictionary>	*Plain*	<integer>	The size of the given dictionary. Mac:4.1
string <string> of <dictionary>	*Named*	<string>	Get, from a dictionary, the string keyed by the specified string. Works for objects in a dictionary of type String or Data. If the type is Data, any null terminator will be returned as part of the string. Use cstring if you expect a Data object containing a null terminated string. Mac:4.1

Examples

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

  Returns TRUE if the specified dictionary exists.

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

  Returns TRUE if the specified dictionary exists.

- `exists dictionary "Timer" of dictionary "SUCheckSchedulerTag" of dictionary "com.apple.SoftwareUpdate" of dictionary "AbsoluteSchedule" of preference "com.apple.scheduler"`

  Returns TRUE if the specified dictionary exists.

- `exists dictionary of file "com.apple.scheduler.plist" of preferences folder`

  Returns TRUE if the specified dictionary exists.

- `keys of entries of dictionary of usb plane of iokit registry`

  Returns the specified keys, e.g.: Device Speed, iSerialNumber, bNumConfigurations, IOUserClientClass, AAPL, current-available, bDeviceClass, USB Product Name, IOCFPlugInTypes.

- `keys of entries of dictionary of iokit registry`

  Returns the specified keys, e.g.: IOKitBuildVersion, IONDRVFrameBufferGeneration, IOConsoleUsers, IOKitDiagnostics, IORegistryPlanes, IOMaximumMappedIOByteCount, IOCatalogue.

- `string "CFBundleVersion" of global dictionary of bundle of folder "iTunes.app" of applications folder`

  Returns the specified bundle version.

- `keys of entries of local dictionary of bundle of folder "iTunes.app" of applications folder`

  Returns information in the default system langauge for bundles that include localized resources, such as Movie File, iTunes Remote Library URL, MP2 Audio File, WAVE Audio File, MP3 Audio File, CFBundleHelpBookName, Equalizer Settings, Tunes Database File, etc...

- `boolean "trash-full" of dictionary of file "com.apple.dock.plist" of preferences folder`

  Returns TRUE if the given flag is set.

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

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

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

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

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

Country

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

Creation Methods

Key Phrase	*Form*	Description
country <string>	*NamedGlobal*	Maps a country script string (e.g. verAfrikaans) to its region code. Mac:4.1

Operators

Examples

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

Osxvalue

These Inspectors provide access to the elements of a dictionary.

Creation Methods

Key Phrase	*Form*	Description
value of <array>	*Plain*	Values of the array. Mac:4.1
value of <dictionaryentry>	*Plain*	Values of the dictionary entry. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
array of <osxvalue>	*Plain*	<array>	Casts the osxvalue (which is essentially untyped) to an array. Mac:4.1
boolean of <osxvalue>	*Plain*	<boolean>	Casts the osxvalue (which is essentially untyped) to a boolean. Mac:4.1
cstring of <osxvalue>	*Plain*	<string>	Strips off any trailing null from an osxvalue, returning a proper string from the dictionary. • NOTE: In Version 6.0.9.54, the plural inspector is misspelled as csrings. Mac:6.0
data of <osxvalue>	*Plain*	<string>	Casts the specified osxvalue to a string. Mac:6.0
date of <osxvalue>	*Plain*	<time>	Casts the osxvalue (essentially untyped) to a date. Mac:4.1
dictionary of <osxvalue>	*Plain*	<dictionary>	Casts the osxvalue (essentially untyped) to a dictionary. Mac:4.1
integer of <osxvalue>	*Plain*	<integer>	Casts the osxvalue (essentially untyped) to an integer. Mac:4.1

Key Phrase	*Form*	Return Type	Description
string of <osxvalue>	*Plain*	<string>	The string of the osxvalue, cast to a string if necessary. Mac:4.1
type of <osxvalue>	*Plain*	<string>	The type of the osxvalue (e.g. boolean, string, integer, data, date, array, dictionary, or unknown type). Mac:4.1

Examples

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

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

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

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

Preference

These Inspectors provide access to application preference files.

Creation Methods

Key Phrase	*Form*	Description
preference <string>	*NamedGlobal*	The named set of preferences. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
array <string> of <preference>	*Named*	<array>	Get, from a preference file, an array keyed by the specified string. Mac:4.1
boolean <string> of <preference>	*Named*	<boolean>	Get, from a preference file, a boolean keyed by the specified string. Mac:4.1
date <string> of <preference>	*Named*	<time>	Get, from a preference, a date keyed by the specified string. Mac:4.1
dictionary <string> of <preference>	*Named*	<dictionary>	Get, from a preference, a dictionary keyed by the specified string. Mac:4.1
integer <string> of <preference>	*Named*	<integer>	Get, from a preference, an integer keyed by the specified string. Mac:4.1
string <string> of <preference>	*Named*	<string>	Get, from a preference, the string keyed by the specified string. Mac:4.1

Examples

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

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

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

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

File Line

A file line is a string from a text file.

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:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
line containing <string> of <file>	*Named*	Returns the line with the specified search string in the given file. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
line of <file>	*Plain*	Returns the lines of a specified file. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
line starting with <string> of <file>	*Named*	Returns a line from the given file beginning with the specified phrase. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
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:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
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:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

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:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
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:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Key Phrase	*Form*	Return Type	Description
previous line of <file line>	*Plain*	<file line>	Returns the line before the nth line in a file, provided n>1. You may repeat this command up to three times. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Application Usage Summary

To enable these Inspectors, you first need to create the client setting _BESClient_UsageManager_EnableAppUsageSummary and initialize it to 1. You can also limit the summary to a subset of applications by creating _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 enable summaries 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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
application usage summary <string>	*NamedGlobal*	Returns the usage summary for the application specified in <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
last start time of <application usage summary>	*Plain*	<time>	Returns the last time this specified application was started. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Key Phrase	*Form*	Return Type	Description
last time seen of <application usage summary>	*Plain*	<time>	Returns the last time this specified application was seen running. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
name of <application usage summary>	*Plain*	<string>	Returns the names of the applications that are currently enabled for usage summaries. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
running of <application usage summary>	*Plain*	<boolean>	Returns TRUE if the specified application is currently running. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
total duration of <application usage summary>	*Plain*	<time interval>	Returns the total elapsed time that the specified application has been running. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

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 compatiblity.

Bios

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

Creation Methods

Key Phrase	*Form*	Description
bios	*PlainGlobal*	This Windows-only Inspector creates the Bios object. On a non-Windows operating system, it returns FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
<bios> as string	*Cast*	<string>	This Windows-only Inspector returns a string that is the concatenation of the BIOS name and date. On a non-Windows operating system, it returns FALSE. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
date of <bios>	*Plain*	<string>	This Windows-only Inspector returns the date string stored in the bios. This string is formatted as MM/DD/YY. On a non-Windows operating system, it returns FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
version of <bios>	*Plain*	<string>	This Windows-only Inspector returns the version string stored in the bios. This string may not exist. The format depends upon your BIOS manufacturer. On a non-Windows operating system, it returns FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

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:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
architecture of <operating system>	*Plain*	<string>	Returns the architecture of the operating system. This is the value of the 'machine' element of the utsname structure obtained by calling uname. Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:6.0
boot time of <operating system>	*Plain*	<time>	Returns the time of the last restart. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
build number of <operating system>	*Plain*	<integer>	Returns the integer build number of the operating system. • Note: On the Macintosh, this returns a string as of BES version 5.1. Win:1.2, Mac:4.1
build number of <operating system>	*Plain*	<string>	Returns the integer build number (as a string) of the operating system. Mac:5.1
build of <operating system>	*Plain*	<string>	Returns a string corresponding to the build number of the OS. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
name of <operating system>	*Plain*	<string>	Returns the name of the operating system as a string. Names might include Win98, WinNT, etc. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
release of <operating system>	*Plain*	<version>	Information about the release of the operating system, formatted as a <version> on the Macintosh, but a <string> on Unix and Windows for BES 6.0. Mac:5.1
uptime of <operating system>	*Plain*	<time interval>	Returns a time interval that represents the elapsed time since the operating system was last booted. • Note: Depending on the Laptop, this interval may not include time spent in hibernation. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:6.0

Examples

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

- `build number of operating system = 67306684`
- Returns TRUE if the build number = 040304BC in hexadecimal.

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.

Creation Methods

Key Phrase	*Form*	Description
main processor	*PlainGlobal*	Creates the object associated with the 'Primary' processor. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
processor	*PlainGlobal*	Iterates through the processors in the system. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
family name of <processor>	*Plain*	<string>	Returns the family name of the CPU, dependent on the type of client computer, for instance Pentium, Sparc, PowerPC G4, etc. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
speed of <processor>	*Plain*	<hertz>	Returns the speed of the processor in Hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:1.2, Sol:3.1, AIX:4.1, Mac:4.1
type of <processor>	*Plain*	<string>	Numeric type of the CPU. Values include: • 0 - standard • 1 - overdrive • 2 - dual CPU capable • 3 - reserved. Sol:6.0, AIX:6.0, Mac:5.1

Examples

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

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

Ram

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

Creation Methods

Key Phrase	*Form*	Description
ram	*PlainGlobal*	Creates the object that can be accessed to inspect the amount of ram on the machine. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
size of <ram>	*Plain*	<integer>	Returns the number of bytes of random access memory on the current machine. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Examples

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

License

These Inspectors are available only through the BigFix Enterprise System. They inspect the properties of the client's BigFix license.

Creation Methods

Key Phrase	*Form*	Description
client license	*PlainGlobal*	Creates the global object containing client licensing information. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
evaluation of <license>	*Plain*	<boolean>	Returns TRUE if client is running an evaluation license. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
expiration date of <license>	*Plain*	<time>	Returns date when license will expire. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
expiration state of <license>	*Plain*	<string>	Returns a string, one of "Unrestricted", "Grace" or "Restricted". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
maximum seat count of <license>	*Plain*	<integer>	Returns maximum seat count allowed by the license. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
seat count state of <license>	*Plain*	<string>	Returns one of "Unrestricted", "Grace" or "Restricted". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
seat of <license>	*Plain*	<integer>	The license number assigned to the client. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Process

When Unix launches a process, it assigns it a unique number called the process ID, or pid. The process object lets you inspect the properties of any running process, including its name and pid.

Creation Methods

Key Phrase	*Form*	Description
process	*PlainGlobal*	Returns all process objects currently running. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
process <integer>	*NumberedGlobal*	Returns the process object corresponding to the given integer pid. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
id of <process>	*Plain*	<integer>	Returns the integer ID of the specified process. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
name of <process>	*Plain*	<string>	Returns the name (as a string) of the specified process. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
pid of <process>	*Plain*	<integer>	Returns the integer process ID for the specified process. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
process id of <process>	*Plain*	<integer>	Returns the integer process ID for the specified process. Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

Examples

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

Computer

These Inspectors provide access to the name of the computer.

Creation Methods

Key Phrase	*Form*	Description
computer	*PlainGlobal*	Refers to the computer itself. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
name of <computer>	*Plain*	<string>	The name of the computer. Mac:4.1

Examples

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

Registryroot

These are the Inspectors for the planes of the IOKit Registry

Creation Methods

Key Phrase	*Form*	Description
iokit registry	*PlainGlobal*	Returns the root of the IOKit registry. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
audio plane of <registryroot>	*Plain*	<registrynode>	The audio plane of IOKit's registry. Mac:4.1
devicetree plane of <registryroot>	*Plain*	<registrynode>	Returns the device tree plane of IOKit's registry. The device tree contains extensive information about devices in the system. Mac:4.1
dictionary of <registryroot>	*Plain*	<dictionary>	Returns a dictionary from the root of the IOKit registry. Mac:4.1
firewire plane of <registryroot>	*Plain*	<registrynode>	The firewire plane of IOKit's registry. Mac:4.1
node <string> of <registryroot>	*Named*	<registrynode>	The named node of the root of the IOKit's registry. Mac:4.1
power plane of <registryroot>	*Plain*	<registrynode>	The power plane of IOKit's registry. Mac:4.1
service plane of <registryroot>	*Plain*	<registrynode>	The service plane of IOKit's registry. Mac:4.1
usb plane of <registryroot>	*Plain*	<registrynode>	The usb plane of IOKit's registry. Mac:4.1

Examples

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

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

Registrynode

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

Creation Methods

Key Phrase	*Form*	Description
audio plane of <registryroot>	*Plain*	The audio plane of IOKit's registry. Mac:4.1
devicetree plane of <registryroot>	*Plain*	Returns the device tree plane of IOKit's registry. Mac:4.1
firewire plane of <registryroot>	*Plain*	The firewire plane of IOKit's registry. Mac:4.1
node <string> of <registrynode>	*Named*	The named node of the given node of the IOKit registry. Mac:4.1
node <string> of <registryroot>	*Named*	The named node of the root of the IOKit's registry. Mac:4.1
node of <registrynode>	*Plain*	The nodes of the given node of the IOKit registry. Mac:4.1
power plane of <registryroot>	*Plain*	The power plane of IOKit's registry. Mac:4.1
service plane of <registryroot>	*Plain*	The service plane of IOKit's registry. Mac:4.1
usb plane of <registryroot>	*Plain*	The usb plane of IOKit's registry. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
classname of <registrynode>	*Plain*	<string>	The class name of the IOKit registry node. Mac:4.1
dictionary of <registrynode>	*Plain*	<dictionary>	Returns a dictionary from a node in the IORegistry. Mac:4.1
name of <registrynode>	*Plain*	<string>	Name of the given IOKit registry node. Mac:4.1
node <string> of <registrynode>	*Named*	<registrynode>	The named node of the given node of the IOKit registry. Mac:4.1
node of <registrynode>	*Plain*	<registrynode>	The nodes of the given node of the IOKit registry. Mac:4.1
path of <registrynode>	*Plain*	<string>	Path of the node in the IOKit registry. Mac:4.1

Examples

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

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

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

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

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

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

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

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

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

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

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

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

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

Scsibus

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

Creation Methods

Key Phrase	*Form*	Description
scsibus	*PlainGlobal*	An iterated property. When used without a number and not iterated it means SCSI bus 0. Mac:4.1
scsibus <integer>	*NumberedGlobal*	Returns a SCSI bus with the given number. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
scsidevice <integer> of <scsibus>	*Numbered*	<scsidevice>	Returns a SCSI device with the given number. Mac:4.1
scsidevice of <scsibus>	*Plain*	<scsidevice>	Returns a SCSI device associated with the given bus. Mac:4.1
version of <scsibus>	*Plain*	<version>	Version of the SCSI bus. Mac:4.1
wide16 scsi of <scsibus>	*Plain*	<boolean>	Whether wide16 is available for the given SCSI bus. Mac:4.1
wide32 scsi of <scsibus>	*Plain*	<boolean>	Whether wide32 is available for the given SCSI bus. Mac:4.1

Scsidevice

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

Creation Methods

Key Phrase	*Form*	Description
scsidevice	*PlainGlobal*	An iterated property. It is derived from calls to the MacOS. Mac:4.1
scsidevice <integer>	*NumberedGlobal*	Returns a SCSI device with the given number. Mac:4.1
scsidevice <integer> of <scsibus>	*Numbered*	Returns a SCSI device with the given number. Mac:4.1
scsidevice of <scsibus>	*Plain*	Returns a SCSI device associated with the given bus. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
product of <scsidevice>	*Plain*	<string>	The product string for the given SCSI device. Mac:4.1
revision of <scsidevice>	*Plain*	<string>	The revision of the SCSI device. Mac:4.1
type of <scsidevice>	*Plain*	<string>	Returns a SCSI device type, such as: DISK, TAPE, PRINTER, CPU, WORM, CDROM, SCAN, DISK, or UNKNOWN. Mac:4.1
vendor of <scsidevice>	*Plain*	<string>	Vendor string for given SCSI device. Mac:4.1

Sound Volume

These Inspectors are for inspecting the sound volume settings.

Creation Methods

Key Phrase	*Form*	Description
full volume	*PlainGlobal*	The maximum possible sound volume. Mac:4.1
zero volume	*PlainGlobal*	The quietest sound volume. Mac:4.1

Operators

Key phrase	Return Type	Description
<integer> * <sound volume>	*<sound volume>*	Multiply an integer by a sound volume. Mac:4.1
<sound volume> / <sound volume>	*<integer>*	Returns an integer representing the ratio of the two sound volumes. Mac:4.1
<sound volume> {cmp} <sound volume>	*<boolean>*	Compares two sound volumes, where: • {cmp} is one of: =, <, <=. Mac:4.1

Key phrase	Return Type	Description
<sound volume> {cmp} <stereo volume>	*<boolean>*	Compares a sound and a stereo volume, where: • {cmp} is one of: =, <, <=. Mac:4.1
<sound volume> {op} <integer>	*<sound volume>*	Operates on a sound volume and an integer, returning a sound volume, where: • {op} is one of: *, /. Mac:4.1
<sound volume> {op} <sound volume>	*<sound volume>*	Operates on two sound volumes, returning another sound volume, where: • {op} is one of: +, -. Mac:4.1
<sound volume> {op} <stereo volume>	*<stereo volume>*	Operates on a sound volume and a stereo volume, returning a stereo volume, where: • {op} is one of: +, -. Mac:4.1
<stereo volume> {cmp} <sound volume>	*<boolean>*	Compares a stereo and a sound volume, where: • {cmp} is one of: =, <, <=. Mac:4.1
<stereo volume> {op} <sound volume>	*<stereo volume>*	Operates on a stereo volume and a sound volume, returning a stereo volume, where: • {op} is one of: +, -. Mac:4.1
<stereo volume> {op} <stereo volume>	*<stereo volume>*	Operates on two stereo volumes, returning another stereo volume, where: • {op} is one of: +, -. Mac:4.1

Examples

- `number of zero volume`
- Returns a number corresponding to the lowest possible sound volume.

Stereo Volume

These Inspectors are for inspecting the stereo volume settings.

Creation Methods

Key Phrase	*Form*	Description
default sound volume	*PlainGlobal*	The default sound volume. Mac:4.1
system beep volume	*PlainGlobal*	The system beep volume. Mac:4.1

Operators

Key phrase	Return Type	Description
<integer> * <stereo volume>	*<stereo volume>*	Multiply an integer by a stereo volume. Mac:4.1
<stereo volume> {cmp} <stereo volume>	*<boolean>*	Compares two stereo volumes, where: • {cmp} is one of: =, <, <=. Mac:4.1
<stereo volume> {op} <integer>	*<stereo volume>*	Operates on a stereo volume and an integer, returning a stereo volume, where: • {op} is one of: *, /. Mac:4.1

Examples

- `number of system beep volume`
- Returns the system beep volume.

Usb

These are the Universal Serial Bus Inspectors.

Creation Methods

Properties

Carbon Proc

The Carbon Process Manager is a Macintosh API for that allows both forward and backward compatibility between code written for Mac OS 9 and Mac OS X. These Inspectors access the Carbon Process functionality. The Carbon proc Inspector allows you to access all processes that register with the Carbon Process Manager. This is different from the set of BSD Unix processes in that not all BSD processes use Carbon, e.g., daemon processes are not shown. Apps that run in Classic mode only run in the Classic process and so do not show up individually as in the Unix process list. However, they do show up in the Carbon version as they register with the Carbon Process Manager as individual entries. The Carbon processes also allow you to get information such as the file path.

Creation Methods

Key Phrase	*Form*	Description
carbon process	*PlainGlobal*	Returns a list of the processes currently running. Mac:5.1
carbon process <integer>	*NumberedGlobal*	Returns the process object corresponding to the given integer process ID (pid). Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
file of <carbon proc>	*Plain*	<filesystem object>	Returns a filesystem object corresponding to the given process. Mac:5.1
id of <carbon proc>	*Plain*	<integer>	Returns the integer ID of the specified carbon process. Mac:5.1
name of <carbon proc>	*Plain*	<string>	Returns the name of the specified carbon process. Mac:5.1

Key Phrase	*Form*	Return Type	Description
pid of <carbon proc>	*Plain*	<integer>	Returns the process ID of the specified carbon process. Mac:5.1
process id of <carbon proc>	*Plain*	<integer>	Returns the integer ID of the specified carbon process. Mac:5.1

Examples

- `names of carbon processes`

Returns a list of processes, such as loginwindow, Dock, SystemUIServer, Finder, iTunesHelper, System Events, Terminal, DashboardClient, etc.

- `files of carbon processes`

Returns a list of file pathnames, such as /System/Library/CoreServcies/loginwindow.app/Contents/MacOS/loginwindow, /System/Library/CoreServcies/Dock.app/Contents/MacOS/Dock, /System/Library/CoreServcies/SystemUIServer.app/Contents/MacOS/SystemUIServer, /System/Library/CoreServcies/Finder.app/Contents/MacOS/Finder, etc.

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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site	*PlainGlobal*	Iterates through all the sites. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
fixlet of <site>	*Plain*	<fixlet>	Iterates through the Fixlet messages of the specified site. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
gather schedule authority of <site>	*Plain*	<string>	Returns a string corresponding to the authority of the site schedule, e.g.: Publisher, Custom, Manual or Disabled. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
gather schedule time interval of <site>	*Plain*	<time interval>	Returns the time interval between automatic gathering of site content. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
group <integer> of <site>	*Numbered*	<site group>	Returns an object corresponding to the numbered group of the specified site. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
last gather time of <site>	*Plain*	<time>	Returns the time of last successful gathering from the site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
name of <site>	*Plain*	<string>	The name of the site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
relevant fixlet of <site>	*Plain*	<fixlet>	Iterates through the Relevant Fixlet messages for the specified site. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
setting <string> of <site>	*Named*	<setting>	Returns the setting whose name matches the string provided from the Fixlet site settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
setting of <site>	*Plain*	<setting>	Returns one or more settings from the site settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site tag of <site>	*Plain*	<string>	Returns the last component of the specified site's url, eg. ‘actionsite’, ‘enteprisesecurity’, etc. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
subscribe time of <site>	*Plain*	<time>	Returns the time that the current machine began subscribing to the site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
type of <site>	*Plain*	<string>	Returns one of the following 4 literal strings: • Master Action Site • Operator Site • Custom Site • Fixlet Site. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
version of <site>	*Plain*	<integer>	Returns the version number of the site content. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Note

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

Examples

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

- `masthead path of site "BESSupport"`
  - Typically returns a string such as: Macintosh HD:private:var:root:Library:Preferences:BigFix Enterprise:BES Support:__Local:Masthead.

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

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

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

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

Site Group

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

Creation Methods

Key Phrase	*Form*	Description
group <integer> of <site>	*Numbered*	Returns an object corresponding to the numbered group of the specified site. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
member of <site group>	*Plain*	<boolean>	Returns TRUE if the current computer is a member of the specified group. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

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:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
relevant fixlet of <site>	*Plain*	Iterates over all the relevant Fixlet messages in the specified site. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0

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:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
header of <fixlet>	*Plain*	<fixlet_header>	Iterates over all the headers of the Fixlet message. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
id of <fixlet>	*Plain*	<integer>	Returns the ID number of the specified Fixlet message. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
relevance of <fixlet>	*Plain*	<boolean>	Returns a boolean TRUE or False, depending on the Relevance of the specified Fixlet message. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0

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:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
header of <fixlet>	*Plain*	Iterates over all the headers of the Fixlet message. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0

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:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
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:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0

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 organization of the BES Client computers.

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. Client objects are derived from file objects, so they share all the file properties.

Creation Methods

Key Phrase	*Form*	Description
client	*PlainGlobal*	Returns the client object corresponding to the BigFix application evaluating the current relevance expression. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

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:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
administrator of <client>	*Plain*	<setting>	Creates a setting object corresponding to the administrators of the given <client>. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
process owner of <client>	*Plain*	<client process owner>	The name of the owner of the BigFix client. Mac:4.1
setting <string> of <client>	*Named*	<setting>	Returns a client setting whose name matches the string provided from the client settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
setting of <client>	*Plain*	<setting>	Returns one or more settings from the client settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
version of <client>	*Plain*	<string>	The product version of the BES application (BESClient or QnA). This Mac-only Inspector is derived from a filesystem object and returns a string. Mac:5.1

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 are assigned by site authors. Settings of the client are assigned by the BES console operator.

Creation Methods

Key Phrase	*Form*	Description
administrator <string> of <client>	*Named*	Creates a setting with the given name on the given <client> computer. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
administrator of <client>	*Plain*	Creates a setting object consisting of the administrator for the given <client> computer. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
setting <string> of <client>	*Named*	Returns the setting whose name matches the string provided from the client settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
setting <string> of <site>	*Named*	Returns the setting whose name matches the string provided from the site settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
setting of <client>	*Plain*	Returns one or more settings from the client settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
setting of <site>	*Plain*	Returns one or more settings from the site settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<setting> as string	*Cast*	<string>	Returns a string formatted as <name>=<value> for the setting. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
effective date of <setting>	*Plain*	<time>	Returns the date when the setting was last modified. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
name of <setting>	*Plain*	<string>	Returns the name of the setting. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
value of <setting>	*Plain*	<string>	Returns the value of the setting. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Examples

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

Selected Server

The BES Server or BES Relay to which the BigFix agent or client 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:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

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:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
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:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
distance of <selected server>	*Plain*	<integer range>	The distance, in IP gateway hops, to the server. Among servers with the same priority, closer servers are preferred. Returns an integer range, since the exact distance may not be known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
gateway address <integer> of <selected server>	*Numbered*	<ipv4 address>	The ip address of a gateway between the agent and the selected server at the given distance from the agent, if known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
gateway address of <selected server>	*Plain*	<ipv4 address>	All known ip addresses of gateways between the agent and the selected server. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
ip address of <selected server>	*Plain*	<ipv4 address>	The ip address to which reports are sent. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
name of <selected server>	*Plain*	<string>	The DNS name of the server, if known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
port number of <selected server>	*Plain*	<integer>	The port number to which reports are sent. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
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:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
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:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Client Process Owner

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

Creation Methods

Key Phrase	*Form*	Description
process owner of <client>	*Plain*	Owner of the agent process. Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<client process owner> as string	*Cast*	<string>	Owner of the agent process. Mac:4.1
long name of <client process owner>	*Plain*	<string>	The long name of the client process owner. Mac:4.1
name of <client process owner>	*Plain*	<string>	Owner of the agent process. Mac:4.1
short name of <client process owner>	*Plain*	<string>	The short name of the client process owner. Mac:4.1

Examples

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

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

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

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

Environment Objects

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

Environment

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

Creation Methods

Key Phrase	*Form*	Description
environment	*PlainGlobal*	Creates the one and only environment object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
variable <string> of <environment>	*Named*	<environment variable>	Returns an environment variable that matches the given name. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
variable of <environment>	*Plain*	<environment variable>	Iterates through all the environment variables defined. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

Examples

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

Environment Variable

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

Creation Methods

Key Phrase	*Form*	Description
variable <string> of <environment>	*Named*	Creates the variable of the environment matching the name provided. The capitalization of the name is ignored. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
variable of <environment>	*Plain*	Iterates through all the environment variables defined. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
<environment variable> as string	*Cast*	<string>	Casting the variable as a string yields a string containing the variable name and the value of the variable separated by ' = '. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
name of <environment variable>	*Plain*	<string>	Returns the name of the variable. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
value of <environment variable>	*Plain*	<string>	Returns the value of the variable. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

Examples

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

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

User Objects

Current User

This object exists if the user is logged in to the current machine.

Creation Methods

Key Phrase	*Form*	Description
current user	*PlainGlobal*	Creates the current user object if one is logged in to the desktop. Win:1.2, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
name of <current user>	*Plain*	<string>	Returns the name of the current user. Win:1.2, Mac:4.1

Examples

- `name of current user = "bigfix"`
- Returns true if BigFix is the current user object.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
action <integer>	*NumberedGlobal*	Creates an action object matching the <integer> id. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
active action	*PlainGlobal*	Creates an action object corresponding to the currently executing action. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
active of <action>	*Plain*	<boolean>	Returns TRUE if the action is currently running (active). Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
active start time of <action>	*Plain*	<time>	Returns the time the action started. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
complete time of <action>	*Plain*	<time>	Returns the time the action completed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
constrained of <action>	*Plain*	<boolean>	Returns TRUE if action is unable to run yet. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
id of <action>	*Plain*	<integer>	Returns the action id of the action. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Return Type	Description
last change time of <action>	*Plain*	<time>	Returns the time when the action state last changed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
origin fixlet id of <action>	*Plain*	<integer>	Returns the Fixlet id that contained the action. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
pending of <action>	*Plain*	<boolean>	Returns TRUE if action is available to run. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
pending time of <action>	*Plain*	<time>	Returns the time the action became pending. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
waiting for download of <action>	*Plain*	<boolean>	Returns TRUE if client is waiting for mirroring server to have downloads required by the action. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

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:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
interface <integer> of <network>	*Numbered*	<network interface>	Returns the particular interface of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
interface of <network>	*Plain*	<network interface>	Returns all the interfaces of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ip interface <integer> of <network>	*Numbered*	<network ip interface>	Returns the particular ip interface of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ip interface of <network>	*Plain*	<network ip interface>	Returns all the ip interfaces of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
link interface <integer> of <network>	*Numbered*	<network link interface>	Returns the specified network link interface of the given network. Mac:5.1
link interface of <network>	*Plain*	<network link interface>	Returns the network link interface of the specified network. Mac:5.1

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:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
interface of <network>	*Plain*	Creates an object with all the interfaces of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
family name of <network interface>	*Plain*	<string>	Returns the name of the interface family. There are about 35 of these, but the most common is AF_INET, or normal IP interface. AF_LINK is used for wifi devices and AF_INET6 is for IP v6 addresses. There are specific inspectors for AF_INET (ip interface) and AF_LINK (link interface). The family name of inspector returns this type as a string corresponding to the #define value in the header file. Mac:5.1
family of <network interface>	*Plain*	<integer>	Returns an family designator of the address family (i.e., 2=AFI_NET). Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
name of <network interface>	*Plain*	<string>	Returns the name of the network interface object. Mac:5.1
up of <network interface>	*Plain*	<boolean>	Returns TRUE if the specified network interface is currently working. Interfaces like wifi may be turned it off to save power, but this Inspector will still tell you if it is active. Mac:5.1

Examples

- `names of interfaces of network`
- Returns a list of the network interface names, e.g., 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.

Creation Methods

Key Phrase	*Form*	Description
ip interface <integer> of <network>	*Numbered*	Creates an object with the specified ip interface of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ip interface of <network>	*Plain*	Creates an object or an object list (using the plural keyword) with all the ip interfaces of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
primary internet connection	*PlainGlobal*	This contains information about the current internet connection. Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
address of <network ip interface>	*Plain*	<ipv4 address>	Returns the ip address of the ip interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
broadcast address of <network ip interface>	*Plain*	<ipv4 address>	Returns the broadcast address of the interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
broadcast support of <network ip interface>	*Plain*	<boolean>	Indicates that broadcast messages are supported by the ip interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
loopback of <network	*Plain*	<boolean>	Indicates that the particular network ip interface

Key Phrase	*Form*	Return Type	Description
ip interface>			is a loopback interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
multicast support of <network ip interface>	*Plain*	<boolean>	Indicates that multicast messages are supported by the ip interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
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:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
subnet address of <network ip interface>	*Plain*	<ipv4 address>	The subnet to which the interface belongs. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
subnet mask of <network ip interface>	*Plain*	<ipv4 address>	The subnet mask of the interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

Examples

- `names of ip interfaces of network`
  - Returns a list of the names of the network IP interfaces, e.g., lo0, en0.

- `addresses of ip interfaces of network`
  - Returns a list of the IP addresses of the network IP interfaces, e.g., 127.0.0.1, 192.168.1.100, etc.

- `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 Link Interface

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

Creation Methods

Key Phrase	*Form*	Description
link interface <integer> of <network>	*Numbered*	Returns the specified network link interface of the given network. Mac:5.1
link interface of <network>	*Plain*	Returns the network link interface of the specified network. Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
mac address of <network link interface>	*Plain*	<string>	Returns the MAC address of the specified network link interface. Mac:5.1

Examples

- `names of link interfaces of network`
  - Returns a list of the names of the network link interfaces, e.g., lo0, gif0, stf0, en0, en1, fw0, etc.

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

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

Ipv4 Address

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

Creation Methods

Key Phrase	*Form*	Description
address of <network ip interface>	*Plain*	Creates an object with the ip address of the interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
broadcast address of <network ip interface>	*Plain*	Creates an object with the broadcast address of the interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
gateway address <integer> of <selected server>	*Numbered*	The ip address of a gateway between the agent and the selected server at the given distance from the agent, if known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
gateway address of <selected server>	*Plain*	All known ip addresses of gateways between the agent and the selected server. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ip address of <selected server>	*Plain*	The ip address to which reports are sent. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ipv4 address <string>	*NamedGlobal*	Creates an object with an ip address for the string provided. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
subnet address of <network ip interface>	*Plain*	Creates an object with the subnet address of the network interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
subnet mask of <network ip interface>	*Plain*	Creates an object with the address bitwise ANDed with the subnet mask. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
<ipv4 address> as string	*Cast*	<string>	Converts the ipv4 address to a string. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

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:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
<ipv4 address> {cmp} <string>	<*boolean*>	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

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.
- `ipv4 address "192.168.100.1"`
  - Returns the four-byte ip address 192.168.100.1.

Active Directory Local Computer

These are the Active Directory Inspectors for the local computer.

Creation Methods

Key Phrase	*Form*	Description
local computer of <active directory server>	*Plain*	Represents your computer within the Active Directory. Win:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
distinguished name error message of <active directory local computer>	*Plain*	<string>	Active Directory error if unable to get the distinguished name (this is for debugging purposes). Win:4.1, Mac:5.1
distinguished name of <active directory local computer>	*Plain*	<string>	Returns the computer's fully qualified active directory name in the distinguished name format, for instance, 'CN=ALBATROSS, CN=Computers, DC=bigfix, DC=com'. Win:4.1, Mac:5.1

Examples

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

Active Directory Server

These are the Active Directory Server Inspectors.

Creation Methods

Key Phrase	*Form*	Description
active directory	*PlainGlobal*	Returns an object containing the properties of the Active Directory to which your machine is attached. Win:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
local computer of <active directory server>	*Plain*	<active directory local computer>	Represents your computer within the Active Directory. Win:4.1, Mac:5.1

Introspectors

These Inspectors look into the currently installed relevance engine to retrieve information about specific Inspectors.

Type

Some Inspectors look at the Relevance language itself, inspecting the Inspectors, so to speak. There are several aspects to view, including the types, properties, casts and operators. This group of Inspectors looks at the various type options available from the Relevance language.

Creation Methods

Key Phrase	*Form*	Description
direct object type of <property>	*Plain*	The type (if any) required after the keyword "of" in an expression using the property. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
index type of <property>	*Plain*	The type (if any) required before or without the keyword "of" in an expression using the property. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
left operand type of <binary operator>	*Plain*	The type required before the operator in an expression. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
operand type of <cast>	*Plain*	The type required before the keyword "as" in an expression using the cast. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
operand type of <unary operator>	*Plain*	The type required in an expression using the operator. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
parent of <type>	*Plain*	The types (if any) whose properties are inherited by this type. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
result type of <binary operator>	*Plain*	The type that the operator produces. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
result type of <property>	*Plain*	The type that the property produces. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
result type of <unary operator>	*Plain*	The type that the operator produces. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Key Phrase	*Form*	Description
right operand type of <binary operator>	*Plain*	The type required after the operator in an expression. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
type	*PlainGlobal*	The inspector types. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
type <string>	*NamedGlobal*	The type with the given name. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<type> as string	*Cast*	<string>	A string indicating the type. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
cast from of <type>	*Plain*	<cast>	Returns the casts that can be created from the specified <type>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
name of <type>	*Plain*	<string>	A string naming the type. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
parent of <type>	*Plain*	<type>	The types (if any) whose properties are inherited by this type. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
property <string> of <type>	*Named*	<property>	Returns the Inspector property of the specified string and type. Typically there is more than one property, so this is often used in the plural. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property of <type>	*Plain*	<property>	Returns the Inspector property of the specified type. Typically there is more than one property, so this is often used in the plural. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property returning <type> of <type>	*Indexed*	<property>	Returns Inspectors of the form <type> of <type>. Typically there is more than one property, so this is often used in the plural. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
size of <type>	*Plain*	<integer>	The number of bytes used in the internal representation of an object of the given type. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Operators

Key phrase	Return Type	Description
<type> = <type>	<*boolean*>	Returns TRUE if both expressions denote the same type. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Property

Some Inspectors look at the Relevance language itself, inspecting the Inspectors, so to speak. There are several aspects to view, including the types, properties, casts and operators. This group of Inspectors looks at the properties available from the Relevance language.

Creation Methods

Key Phrase	*Form*	Description
property	*PlainGlobal*	The inspectors invoked with phrases, but without the keyword "as". Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
property <string>	*NamedGlobal*	Returns the first inspector property whose name matches the given string. Note that there may be more than one property with a given name. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property <string> of <type>	*Named*	Returns the Inspector property of the specified string and type. Typically there is more than one property, so this is often used in the plural. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property of <type>	*Plain*	Returns the Inspector property of the specified type. Typically there is more than one property, so this is typically used in the plural. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property returning <type>	*IndexedGlobal*	Produces a list of the Inspector properties that return the "file" type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property returning <type> of <type>	*Indexed*	Returns Inspectors of the form <type> of <type>. Typically there is more than one property, so this is often used in the plural. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<property> as string	*Cast*	<string>	A short description of the use of the property. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
direct object type of <property>	*Plain*	<type>	The type (if any) required after the keyword "of" in an expression using the property. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
index type of <property>	*Plain*	<type>	The type (if any) required before or without the keyword "of" in an expression using the property. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
multivalued of <property>	*Plain*	<boolean>	Can the property have more than one value for a single input?. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
plural name of <property>	*Plain*	<string>	The name of the property, in the plural. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
result type of <property>	*Plain*	<type>	The type that the property produces. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
singular name of <property>	*Plain*	<string>	The name of the property, in the singular. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
usual name of <property>	*Plain*	<string>	Returns the usual name of the specified property. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Binary Operator

Some Inspectors look at the Relevance language itself, inspecting the Inspectors, so to speak. There are several aspects to view, including the types, properties, casts and operators. This group of Inspectors looks at the various binary operators available from the Relevance language.

Creation Methods

Key Phrase	*Form*	Description
binary operator	*PlainGlobal*	The inspectors that have two parameters, and are invoked with punctuation marks or reserved phrases. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
binary operator <string>	*NamedGlobal*	Typically used in the plural, returns the various possible binary inspectors that use the specified operators. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
binary operator returning <type>	*IndexedGlobal*	Returns a list of binary operators that return the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<binary operator> as string	*Cast*	<string>	A short description of the use of the operator. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
left operand type of <binary operator>	*Plain*	<type>	The type required before the operator in an expression. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
name of <binary operator>	*Plain*	<string>	A phrase naming the operator. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
result type of <binary operator>	*Plain*	<type>	The type that the operator produces. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
right operand type of <binary operator>	*Plain*	<type>	The type required after the operator in an expression. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
symbol of <binary operator>	*Plain*	<string>	A phrase or punctuation mark used to invoke the operator. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Unary Operator

Some Inspectors look at the Relevance language itself, inspecting the Inspectors, so to speak. There are several aspects to view, including the types, properties, casts and operators. This group of Inspectors looks at the various unary operators available from the Relevance language.

Creation Methods

Key Phrase	*Form*	Description
unary operator	*PlainGlobal*	The inspectors that have one parameter, and are invoked with punctuation marks or reserved phrases. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
unary operator <string>	*NamedGlobal*	Typically used in the plural, this inspector returns a list of objects that use the specified operator. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
unary operator returning <type>	*IndexedGlobal*	Returns a list of the unary operator inspectors (such as negative) that return the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<unary operator> as string	*Cast*	<string>	A short description of the use of the operator. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
name of <unary operator>	*Plain*	<string>	A phrase naming the operator. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
operand type of <unary operator>	*Plain*	<type>	The type required in an expression using the operator. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
result type of <unary operator>	*Plain*	<type>	The type that the operator produces. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
symbol of <unary operator>	*Plain*	<string>	A phrase or punctuation mark used to invoke the operator. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Cast

Some Inspectors look at the Relevance language itself, inspecting the Inspectors, so to speak. There are several aspects to view, including the types, properties, casts and operators. This group of Inspectors looks at the various casting operations available from the Relevance language.

Creation Methods

Key Phrase	*Form*	Description
cast	*PlainGlobal*	The inspectors invoked using the keyword "as.". Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
cast <string>	*NamedGlobal*	Returns a list of the objects that can be cast into the type specified by <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
cast from of <type>	*Plain*	Returns the casts that can be created from the specified <type>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
cast returning <type>	*IndexedGlobal*	Returns a list of the objects that can be cast into the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<cast> as string	*Cast*	<string>	A short description of the use of the cast. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
name of <cast>	*Plain*	<string>	The phrase used after the keyword "as" in an expression using the cast. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
operand type of <cast>	*Plain*	<type>	The type required before the keyword "as" in an expression using the cast. Win:5.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Miscellaneous

Dummy

These Inspectors are place holders for compatability with Windows clients

Creation Methods

Key Phrase	*Form*	Description
service <string>	*NamedGlobal*	A dummy inspector to provide compatibility with Windows. Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
state of <dummy>	*Plain*	<string>	A dummy inspector to provide compatibility with other operating systems. Mac:5.1

Dummy Type

These Inspectors are place holders for compatability with Windows clients

Creation Methods

Key Phrase	*Form*	Description
registry	*PlainGlobal*	No meaning on the Macintosh. Included for compatability with other Agents. Mac:4.1

Properties

Nothing

These Inspectors are place holders for compatability with Windows clients

Creation Methods

Key Phrase	*Form*	Description
main gather service	*PlainGlobal*	Returns FALSE. Included for compatability with Windows clients. Mac:4.1
relay service	*PlainGlobal*	Returns FALSE. Included for compatability with Windows clients. Mac:4.1

Key Phrases (Inspectors)

This chapter provides an alphabetical list of the Inspector keywords and their casting operators. Both lists include the context object type (From an object), and the resulting object type (Creates an object). These lists are not all-inclusive; they only include those Inspectors that are relevant to the context of the current Guide.

Key phrases

This is a list of the key phrases relevant to this document, sorted alphabetically.

Key Phrase	Plural	Creates a	From a	*Form*
absolute value of <hertz>	absolute values	<hertz>	<hertz>	*Plain*
absolute value of <integer>	absolute values	<integer>	<integer>	*Plain*
absolute value of <time interval>	absolute values	<time interval>	<time interval>	*Plain*
action	actions	<action>	<world>	*PlainGlobal*
action <integer>	actions	<action>	<world>	*NumberedGlobal*
action lock state	action lock states	<action lock state>	<world>	*PlainGlobal*
active action	active actions	<action>	<world>	*PlainGlobal*
active of <action>	actives	<boolean>	<action>	*Plain*
active start time of <action>	active start times	<time>	<action>	*Plain*
address of <network ip interface>	addresses	<ipv4 address>	<network ip interface>	*Plain*
administrator <string> of <client>	administrators	<setting>	<client>	*Named*
administrator of <client>	administrators	<setting>	<client>	*Plain*
ancestor of <filesystem object>	ancestors	<folder>	<filesystem object>	*Plain*
apparent registration server time	apparent registration server times	<time>	<world>	*PlainGlobal*
application <string>	applications	<application>	<world>	*NamedGlobal*
application usage summary	application usage summaries	<application usage summary>	<world>	*PlainGlobal*
application usage summary <string>	application usage summaries	<application usage summary>	<world>	*NamedGlobal*

Key Phrase	Plural	Creates a	From a	*Form*
binary operator <string>	binary operators	<binary operator>	<world>	*NamedGlobal*
binary operator returning <type>	binary operators returning	<binary operator>	<world>	*IndexedGlobal*
bit <integer>	bits	<bit set>	<world>	*NumberedGlobal*
bit <integer> of <bit set>	bits	<boolean>	<bit set>	*Numbered*
bit <integer> of <integer>	bits	<boolean>	<integer>	*Numbered*
bit set <string>	bit sets	<bit set>	<world>	*NamedGlobal*
boolean <string>	booleans	<boolean>	<world>	*NamedGlobal*
boot time of <operating system>	boot times	<time>	<operating system>	*Plain*
broadcast address of <network ip interface>	broadcast addresses	<ipv4 address>	<network ip interface>	*Plain*
broadcast support of <network ip interface>	broadcast supports	<boolean>	<network ip interface>	*Plain*
build number of <operating system>	build numbers	<integer>	<operating system>	*Plain*
byte <integer> of <file>	bytes	<integer>	<file>	*Numbered*
case insensitive regex <string>	case insensitive regexes	<regular expression>	<world>	*NamedGlobal*
case insensitive regular expression <string>	case insensitive regular expressions	<regular expression>	<world>	*NamedGlobal*
cast <string>	casts	<cast>	<world>	*NamedGlobal*
cast from of <type>	casts from	<cast>	<type>	*Plain*
cast returning <type>	casts returning	<cast>	<world>	*IndexedGlobal*
character <integer>	characters	<string>	<world>	*NumberedGlobal*
character <integer> of <string>	characters	<substring>	<string>	*Numbered*
character of <string>	characters	<substring>	<string>	*Plain*
client	clients	<client>	<world>	*PlainGlobal*
client folder of <site>	client folders	<folder>	<site>	*Plain*
client license	client licenses	<license>	<world>	*PlainGlobal*
competition size of <selected server>	competition sizes	<integer>	<selected server>	*Plain*
competition weight of <selected server>	competition weights	<integer>	<selected server>	*Plain*
complete time of <action>	complete times	<time>	<action>	*Plain*

Key Phrase	Plural	Creates a	From a	*Form*
computer id	computer ids	<integer>	<world>	*PlainGlobal*
computer name	computer names	<string>	<world>	*PlainGlobal*
concatenation <string> of <string>	concatenations	<string>	<string>	*Named*
concatenation of <string>	concatenations	<string>	<string>	*Plain*
constrained of <action>	constraineds	<boolean>	<action>	*Plain*
content of <file>	contents	<file content>	<file>	*Plain*
controller of <action lock state>	controllers	<string>	<action lock state>	*Plain*
creation time of <filesystem object>	creation times	<time>	<filesystem object>	*Plain*
current site	current sites	<site>	<world>	*PlainGlobal*
current user	current users	<current user>	<world>	*PlainGlobal*
custom site subscription effective date <string>	custom site subscription effective dates	<time>	<world>	*NamedGlobal*
date of <bios>	dates	<string>	<bios>	*Plain*
day	days	<time interval>	<world>	*PlainGlobal*
day_of_week of <date>	days_of_week	<integer>	<date>	*Plain*
descendant of <folder>	descendants	<file>	<folder>	*Plain*
direct object type of <property>	direct object types	<type>	<property>	*Plain*
distance of <selected server>	distances	<integer range>	<selected server>	*Plain*
distinguished name error message of <active directory local computer>	distinguished name error messages	<string>	<active directory local computer>	*Plain*
distinguished name of <active directory local computer>	distinguished names	<string>	<active directory local computer>	*Plain*
divided by zero of <floating point>	divided by zeroes	<boolean>	<floating point>	*Plain*
dns name	dns names	<string>	<world>	*PlainGlobal*
effective date of <action lock state>	effective dates	<time>	<action lock state>	*Plain*
effective date of <setting>	effective dates	<time>	<setting>	*Plain*
end of <substring>	ends	<string position>	<substring>	*Plain*
end of <time range>	ends	<time>	<time range>	*Plain*
environment	environments	<environment>	<world>	*PlainGlobal*

Key Phrase	Plural	Creates a	From a	*Form*
error <string>	errors	<undefined>	<world>	*NamedGlobal*
evaluation of <license>	evaluations	<boolean>	<license>	*Plain*
expiration date of <action lock state>	expiration dates	<time>	<action lock state>	*Plain*
expiration date of <license>	expiration dates	<time>	<license>	*Plain*
expiration state of <license>	expiration states	<string>	<license>	*Plain*
false	falses	<boolean>	<world>	*PlainGlobal*
family name of <processor>	family names	<string>	<processor>	*Plain*
family of <network interface>	families	<integer>	<network interface>	*Plain*
file <string>	files	<file>	<world>	*NamedGlobal*
file <string> of <folder>	files	<file>	<folder>	*Named*
file of <folder>	files	<file>	<folder>	*Plain*
final part <time interval> of <time range>	final parts	<time range>	<time range>	*Indexed*
finite of <floating point>	finites	<boolean>	<floating point>	*Plain*
first <integer> of <string>	firsts	<substring>	<string>	*Numbered*
first <string> of <string>	firsts	<substring>	<string>	*Named*
first match <regular expression> of <string>	first matches	<regular expression match>	<string>	*Indexed*
first start time of <application usage summary>	first start times	<time>	<application usage summary>	*Plain*
fixlet of <site>	fixlets	<fixlet>	<site>	*Plain*
floating point <string>	floating points	<floating point>	<world>	*NamedGlobal*
folder <string>	folders	<folder>	<world>	*NamedGlobal*
folder <string> of <folder>	folders	<folder>	<folder>	*Named*
folder of <folder>	folders	<folder>	<folder>	*Plain*
following text of <string position>	following texts	<substring>	<string position>	*Plain*
following text of <substring>	following texts	<substring>	<substring>	*Plain*
gateway address <integer> of <selected server>	gateway addresses	<ipv4 address>	<selected server>	*Numbered*

Key Phrase	Plural	Creates a	From a	*Form*
gateway address of <selected server>	gateway addresses	<ipv4 address>	<selected server>	*Plain*
gather schedule authority of <site>	gather schedule authoritys	<string>	<site>	*Plain*
gather schedule time interval of <site>	gather schedule time intervals	<time interval>	<site>	*Plain*
ghz	ghzs	<hertz>	<world>	*PlainGlobal*
greatest hz	greatest hzs	<hertz>	<world>	*PlainGlobal*
greatest integer	greatest integers	<integer>	<world>	*PlainGlobal*
greatest time interval	greatest time intervals	<time interval>	<world>	*PlainGlobal*
group <integer> of <site>	groups	<site group>	<site>	*Numbered*
header <string> of <fixlet>	headers	<fixlet_header>	<fixlet>	*Named*
header of <fixlet>	headers	<fixlet_header>	<fixlet>	*Plain*
hexadecimal integer <string>	hexadecimal integers	<integer>	<world>	*NamedGlobal*
hexadecimal string <string>	hexadecimal strings	<string>	<world>	*NamedGlobal*
hostname	hostnames	<string>	<world>	*PlainGlobal*
hour	hours	<time interval>	<world>	*PlainGlobal*
hz	hzs	<hertz>	<world>	*PlainGlobal*
id of <action>	ids	<integer>	<action>	*Plain*
id of <fixlet>	ids	<integer>	<fixlet>	*Plain*
id of <site group>	ids	<integer>	<site group>	*Plain*
index type of <property>	index types	<type>	<property>	*Plain*
inexact of <floating point>	inexacts	<boolean>	<floating point>	*Plain*
infinite of <floating point>	infinites	<boolean>	<floating point>	*Plain*
initial part <time interval> of <time range>	initial parts	<time range>	<time range>	*Indexed*
integer <integer>	integers	<integer>	<world>	*NumberedGlobal*
integer <string>	integers	<integer>	<world>	*NamedGlobal*
integer ceiling of <floating point>	integer ceilings	<integer>	<floating point>	*Plain*
integer floor of <floating point>	integer floors	<integer>	<floating point>	*Plain*
interface <integer> of <network>	interfaces	<network interface>	<network>	*Numbered*

Key Phrase	Plural	Creates a	From a	*Form*
interface of <network>	interfaces	<network interface>	<network>	*Plain*
invalid of <floating point>	invalids	<boolean>	<floating point>	*Plain*
ip address of <selected server>	ip addresses	<ipv4 address>	<selected server>	*Plain*
ip interface <integer> of <network>	ip interfaces	<network ip interface>	<network>	*Numbered*
ip interface of <network>	ip interfaces	<network ip interface>	<network>	*Plain*
ipv4 address <string>	ipv4 addresses	<ipv4 address>	<world>	*NamedGlobal*
key <string> of <file section>	keys	<string>	<file section>	*Named*
key <string> of <file>	keys	<string>	<file>	*Named*
khz	khzs	<hertz>	<world>	*PlainGlobal*
last <integer> of <string>	lasts	<substring>	<string>	*Numbered*
last <string> of <string>	lasts	<substring>	<string>	*Named*
last change time of <action>	last change times	<time>	<action>	*Plain*
last gather time of <site>	last gather times	<time>	<site>	*Plain*
last start time of <application usage summary>	last start times	<time>	<application usage summary>	*Plain*
last time seen of <application usage summary>	last times seen	<time>	<application usage summary>	*Plain*
least hz	least hzs	<hertz>	<world>	*PlainGlobal*
least integer	least integers	<integer>	<world>	*PlainGlobal*
least significant one bit of <bit set>	least significant one bits	<integer>	<bit set>	*Plain*
least time interval	least time intervals	<time interval>	<world>	*PlainGlobal*
left operand type of <binary operator>	left operand types	<type>	<binary operator>	*Plain*
left shift <integer> of <bit set>	left shifts	<bit set>	<bit set>	*Numbered*
length of <rope>	lengths	<integer>	<rope>	*Plain*
length of <string>	lengths	<integer>	<string>	*Plain*
length of <time range>	lengths	<time interval>	<time range>	*Plain*
less significance <integer> of <floating point>	less significances	<floating point>	<floating point>	*Numbered*
line <integer> of <file>	lines	<file line>	<file>	*Numbered*

Key Phrase	Plural	Creates a	From a	*Form*
line containing <string> of <file>	lines containing	<file line>	<file>	*Named*
line number of <file line>	line numbers	<integer>	<file line>	*Plain*
line of <file>	lines	<file line>	<file>	*Plain*
line starting with <string> of <file>	lines starting with	<file line>	<file>	*Named*
local computer of <active directory server>	local computers	<active directory local computer>	<active directory server>	*Plain*
local time <string>	local times	<time>	<world>	*NamedGlobal*
local time zone	local time zones	<time zone>	<world>	*PlainGlobal*
lock string of <action lock state>	lock strings	<string>	<action lock state>	*Plain*
locked of <action lock state>	lockeds	<boolean>	<action lock state>	*Plain*
loopback of <network ip interface>	loopbacks	<boolean>	<network ip interface>	*Plain*
lower bound of <integer range>	lower bounds	<integer>	<integer range>	*Plain*
main processor	main processors	<processor>	<world>	*PlainGlobal*
masthead of <site>	mastheads	<file>	<site>	*Plain*
match <regular expression> of <string>	matches	<regular expression match>	<string>	*Indexed*
maximum of <integer>	maxima	<integer>	<integer>	*Plain*
maximum of <time interval>	maxima	<time interval>	<time interval>	*Plain*
maximum of <time>	maxima	<time>	<time>	*Plain*
maximum seat count of <license>	maximum seat counts	<integer>	<license>	*Plain*
mean of <floating point>	means	<floating point>	<floating point>	*Plain*
mean of <integer>	means	<floating point>	<integer>	*Plain*
member of <site group>	members	<boolean>	<site group>	*Plain*
mhz	mhzs	<hertz>	<world>	*PlainGlobal*
microsecond	microseconds	<time interval>	<world>	*PlainGlobal*
millisecond	milliseconds	<time interval>	<world>	*PlainGlobal*
minimum of <integer>	minima	<integer>	<integer>	*Plain*
minimum of <time interval>	minima	<time interval>	<time interval>	*Plain*
minimum of <time>	minima	<time>	<time>	*Plain*

Key Phrase	Plural	Creates a	From a	*Form*
minute	minutes	<time interval>	<world>	*PlainGlobal*
modification time of <filesystem object>	modification times	<time>	<filesystem object>	*Plain*
month of <date>	months	<integer>	<date>	*Plain*
more significance <integer> of <floating point>	more significances	<floating point>	<floating point>	*Numbered*
most significant one bit of <bit set>	most significant one bits	<integer>	<bit set>	*Plain*
multicast support of <network ip interface>	multicast supports	<boolean>	<network ip interface>	*Plain*
multiplicity of <integer with multiplicity>	multiplicities	<integer>	<integer with multiplicity>	*Plain*
multiplicity of <string with multiplicity>	multiplicities	<integer>	<string with multiplicity>	*Plain*
multivalued of <property>	multivalueds	<boolean>	<property>	*Plain*
name of <application usage summary>	names	<string>	<application usage summary>	*Plain*
name of <binary operator>	names	<string>	<binary operator>	*Plain*
name of <cast>	names	<string>	<cast>	*Plain*
name of <current user>	names	<string>	<current user>	*Plain*
name of <environment variable>	names	<string>	<environment variable>	*Plain*
name of <filesystem object>	names	<string>	<filesystem object>	*Plain*
name of <fixlet_header>	names	<string>	<fixlet_header>	*Plain*
name of <operating system>	names	<string>	<operating system>	*Plain*
name of <selected server>	names	<string>	<selected server>	*Plain*
name of <setting>	names	<string>	<setting>	*Plain*
name of <site>	names	<string>	<site>	*Plain*
name of <type>	names	<string>	<type>	*Plain*
name of <unary operator>	names	<string>	<unary operator>	*Plain*
nan of <floating point>	nans	<boolean>	<floating point>	*Plain*
network	networks	<network>	<world>	*PlainGlobal*

Key Phrase	Plural	Creates a	From a	*Form*
next line of <file line>	next lines	<file line>	<file line>	*Plain*
normal of <floating point>	normals	<boolean>	<floating point>	*Plain*
now	nows	<time>	<world>	*PlainGlobal*
one bit of <bit set>	one bits	<integer>	<bit set>	*Plain*
operand type of <cast>	operand types	<type>	<cast>	*Plain*
operand type of <unary operator>	operand types	<type>	<unary operator>	*Plain*
operating system	operating systems	<operating system>	<world>	*PlainGlobal*
origin fixlet id of <action>	origin fixlet ids	<integer>	<action>	*Plain*
overflow of <floating point>	overflows	<boolean>	<floating point>	*Plain*
parameter <string> of <action>	parameters	<string>	<action>	*Named*
parent folder of <filesystem object>	parent folders	<folder>	<filesystem object>	*Plain*
parent of <type>	parents	<type>	<type>	*Plain*
parenthesized part <integer> of <regular expression match>	parenthesized parts	<substring>	<regular expression match>	*Numbered*
parenthesized part of <regular expression match>	parenthesized parts	<substring>	<regular expression match>	*Plain*
pathname of <filesystem object>	pathnames	<string>	<filesystem object>	*Plain*
pending login	pending logins	<boolean>	<world>	*PlainGlobal*
pending login of <action>	pending logins	<boolean>	<action>	*Plain*
pending of <action>	pendings	<boolean>	<action>	*Plain*
pending restart	pending restarts	<boolean>	<world>	*PlainGlobal*
pending restart <string>	pending restarts	<boolean>	<world>	*NamedGlobal*
pending restart of <action>	pending restarts	<boolean>	<action>	*Plain*
pending time of <action>	pending times	<time>	<action>	*Plain*
plural name of <property>	plural names	<string>	<property>	*Plain*
point to point of <network ip interface>	point to points	<boolean>	<network ip interface>	*Plain*
port number of <selected server>	port numbers	<integer>	<selected server>	*Plain*

Key Phrase	Plural	Creates a	From a	*Form*
position <integer> of <string>	positions	<string position>	<string>	*Numbered*
position of <string>	positions	<string position>	<string>	*Plain*
preceding text of <string position>	preceding texts	<substring>	<string position>	*Plain*
preceding text of <substring>	preceding texts	<substring>	<substring>	*Plain*
previous line of <file line>	previous lines	<file line>	<file line>	*Plain*
priority of <selected server>	priorities	<integer>	<selected server>	*Plain*
processor	processors	<processor>	<world>	*PlainGlobal*
property <string>	properties	<property>	<world>	*NamedGlobal*
property <string> of <type>	properties	<property>	<type>	*Named*
property of <type>	properties	<property>	<type>	*Plain*
property returning <type>	properties returning	<property>	<world>	*IndexedGlobal*
property returning <type> of <type>	properties returning	<property>	<type>	*Indexed*
ram	rams	<ram>	<world>	*PlainGlobal*
range after <time> of <time range>	ranges after	<time range>	<time range>	*Indexed*
range before <time> of <time range>	ranges before	<time range>	<time range>	*Indexed*
regex <string>	regexes	<regular expression>	<world>	*NamedGlobal*
regular expression <string>	regular expressions	<regular expression>	<world>	*NamedGlobal*
relative significance place <integer> of <floating point>	relative significance places	<floating point>	<floating point>	*Numbered*
relative significance place of <floating point>	relative significance places	<floating point>	<floating point>	*Plain*
relevance of <fixlet>	relevances	<boolean>	<fixlet>	*Plain*
relevant fixlet of <site>	relevant fixlets	<fixlet>	<site>	*Plain*
result type of <binary operator>	result types	<type>	<binary operator>	*Plain*
result type of <property>	result types	<type>	<property>	*Plain*
result type of <unary operator>	result types	<type>	<unary operator>	*Plain*
right operand type of <binary operator>	right operand types	<type>	<binary operator>	*Plain*
right shift <integer> of <bit set>	right shifts	<bit set>	<bit set>	*Numbered*

Key Phrase	Plural	Creates a	From a	*Form*
rope <string>	ropes	<rope>	<world>	*NamedGlobal*
running of <application usage summary>	runnings	<boolean>	<application usage summary>	*Plain*
seat count state of <license>	seat count states	<string>	<license>	*Plain*
seat of <license>	seats	<integer>	<license>	*Plain*
second	seconds	<time interval>	<world>	*PlainGlobal*
section <string> of <file>	sections	<file section>	<file>	*Named*
selected server	selected servers	<selected server>	<world>	*PlainGlobal*
setting <string> of <client>	settings	<setting>	<client>	*Named*
setting <string> of <site>	settings	<setting>	<site>	*Named*
setting of <client>	settings	<setting>	<client>	*Plain*
setting of <site>	settings	<setting>	<site>	*Plain*
sha1 of <file>	sha1s	<string>	<file>	*Plain*
significance place <integer> of <floating point>	significance places	<floating point>	<floating point>	*Numbered*
significance place of <floating point>	significance places	<floating point>	<floating point>	*Plain*
significance threshold of <floating point>	significance thresholds	<floating point>	<floating point>	*Plain*
significant digits <integer> of <hertz>	significant digitss	<hertz>	<hertz>	*Numbered*
significant digits <integer> of <integer>	significant digitss	<integer>	<integer>	*Numbered*
singular name of <property>	singular names	<string>	<property>	*Plain*
site	sites	<site>	<world>	*PlainGlobal*
site <string>	sites	<site>	<world>	*NamedGlobal*
site tag of <site>	site tags	<string>	<site>	*Plain*
size of <file>	sizes	<integer>	<file>	*Plain*
size of <ram>	sizes	<integer>	<ram>	*Plain*
size of <type>	sizes	<integer>	<type>	*Plain*
speed of <processor>	speeds	<hertz>	<processor>	*Plain*
standard deviation of <floating point>	standard deviations	<floating point>	<floating point>	*Plain*
standard deviation of <integer>	standard deviations	<floating point>	<integer>	*Plain*

Key Phrase	Plural	Creates a	From a	*Form*
start of <substring>	starts	<string position>	<substring>	*Plain*
start of <time range>	starts	<time>	<time range>	*Plain*
status of <action>	statuss	<string>	<action>	*Plain*
string <string>	strings	<string>	<world>	*NamedGlobal*
subnet address of <network ip interface>	subnet addresses	<ipv4 address>	<network ip interface>	*Plain*
subnet mask of <network ip interface>	subnet masks	<ipv4 address>	<network ip interface>	*Plain*
subscribe time of <site>	subscribe times	<time>	<site>	*Plain*
substring <string> of <string>	substrings	<substring>	<string>	*Named*
substring after <string> of <string>	substrings after	<substring>	<string>	*Named*
substring before <string> of <string>	substrings before	<substring>	<string>	*Named*
substring between <string> of <string>	substrings between	<substring>	<string>	*Named*
substring separated by <string> of <string>	substrings separated by	<substring>	<string>	*Named*
sum of <integer>	sums	<integer>	<integer>	*Plain*
symbol of <binary operator>	symbols	<string>	<binary operator>	*Plain*
symbol of <unary operator>	symbols	<string>	<unary operator>	*Plain*
time <string>	times	<time>	<world>	*NamedGlobal*
time interval <string>	time intervals	<time interval>	<world>	*NamedGlobal*
time zone <string>	time zones	<time zone>	<world>	*NamedGlobal*
total duration of <application usage summary>	total durations	<time interval>	<application usage summary>	*Plain*
total run count of <application usage summary>	total run counts	<integer>	<application usage summary>	*Plain*
true	trues	<boolean>	<world>	*PlainGlobal*
type of <processor>	types	<integer>	<processor>	*Plain*
type of <site>	types	<string>	<site>	*Plain*
unary operator <string>	unary operators	<unary operator>	<world>	*NamedGlobal*
unary operator returning <type>	unary operators returning	<unary operator>	<world>	*IndexedGlobal*

Key Phrase	Plural	Creates a	From a	*Form*
underflow of <floating point>	underflows	<boolean>	<floating point>	*Plain*
unique value of <integer>	unique values	<integer with multiplicity>	<integer>	*Plain*
unique value of <string>	unique values	<string with multiplicity>	<string>	*Plain*
universal time <string>	universal times	<time>	<world>	*NamedGlobal*
universal time zone	universal time zones	<time zone>	<world>	*PlainGlobal*
upper bound of <integer range>	upper bounds	<integer>	<integer range>	*Plain*
uptime of <operating system>	uptimes	<time interval>	<operating system>	*Plain*
url of <site>	urls	<string>	<site>	*Plain*
usual name of <property>	usual names	<string>	<property>	*Plain*
value of <environment variable>	values	<string>	<environment variable>	*Plain*
value of <fixlet_header>	values	<string>	<fixlet_header>	*Plain*
value of <setting>	values	<string>	<setting>	*Plain*
variable <string> of <environment>	variables	<environment variable>	<environment>	*Named*
variable of <environment>	variables	<environment variable>	<environment>	*Plain*
version <string>	versions	<version>	<world>	*NamedGlobal*
version of <bios>	versions	<string>	<bios>	*Plain*
version of <file>	versions	<version>	<file>	*Plain*
version of <site>	versions	<integer>	<site>	*Plain*
waiting for download of <action>	waiting for downloads	<boolean>	<action>	*Plain*
week	weeks	<time interval>	<world>	*PlainGlobal*
weight of <selected server>	weights	<integer>	<selected server>	*Plain*
year of <date>	years	<integer>	<date>	*Plain*

Casting Operators

The casting operators allow you to convert one type to another.This is a list of the casting operators sorted by key phrase.

Key Phrase	Creates a	From a
<action lock state> as string	<string>	<action lock state>
<binary operator> as string	<string>	<binary operator>
<bios> as string	<string>	<bios>
<bit set> as integer	<integer>	<bit set>
<bit set> as string	<string>	<bit set>
<boolean> as boolean	<boolean>	<boolean>
<boolean> as string	<string>	<boolean>
<cast> as string	<string>	<cast>
<environment variable> as string	<string>	<environment variable>
<file content> as lowercase	<file content>	<file content>
<file content> as uppercase	<file content>	<file content>
<floating point> as integer	<integer>	<floating point>
<floating point> as scientific notation	<string>	<floating point>
<floating point> as standard notation	<string>	<floating point>
<floating point> as string	<string>	<floating point>
<hertz> as string	<string>	<hertz>
<integer> as bit set	<bit set>	<integer>
<integer> as bits	<bit set>	<integer>
<integer> as floating point	<floating point>	<integer>
<integer> as hexadecimal	<string>	<integer>
<integer> as integer	<integer>	<integer>
<integer> as string	<string>	<integer>
<ipv4 address> as string	<string>	<ipv4 address>

Key Phrase	Creates a	From a
<operating system> as string	<string>	<operating system>
<property> as string	<string>	<property>
<rope> as string	<string>	<rope>
<setting> as string	<string>	<setting>
<string> as boolean	<boolean>	<string>
<string> as floating point	<floating point>	<string>
<string> as hexadecimal	<string>	<string>
<string> as integer	<integer>	<string>
<string> as local time	<time>	<string>
<string> as lowercase	<string>	<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 universal time	<time>	<string>
<string> as uppercase	<string>	<string>
<string> as version	<version>	<string>
<time interval> as string	<string>	<time interval>
<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>

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z