Tivoli. *Endpoint Manager*

Version 8.1

Core Inspector Guide

IBM®

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

© Copyright IBM Corporation 2003, 2011.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Part One

Introduction

The ***Tivoli Endpoint Manager Core Inspector Library*** is a guide to the core ordinary phrases (known as Inspectors) of the **Relevance Language**™ as they apply to all BigFix Client Platforms. Using this guide along with the relevant Client Inspector Guides, you can write your own Relevance Expressions and use them to target actions to precisely those Clients that need them. The core Inspectors also complement the **Session** Inspectors – phrases that interrogate the BigFix database within the Console or Web Reports environment.

Using these Inspectors, you can write **Fixlet**® messages and post them to **Fixlet Sites**. You also use the core Inspectors to write Dashboards or Wizards for the session context. For more information on how to write Relevance expressions, see the ***Tivoli Endpoint Manager Console Operator's Guide*** and the ***Tivoli Endpoint Manager Relevance Language Reference***.

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

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

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

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

Audience

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

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

This manual details the properties and operators of the Core BigFix Inspector keywords. These include those Inspectors common to the various BigFix Clients as well as the Session Inspectors. This document should be used along with specific client platform ***Inspector Guides*** (for Windows, Mac or various Unix clients) to provide a complete library reference to Client-based Relevance scripting. This document should also be used in concert with the ***Session Inspector Guide*** to analyze the BigFix database in a Dashboard or Web Report.

Conventions Used in this manual

This document makes use of the following conventions and nomenclature:

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

Examples

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

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

Versions

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Forms

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

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

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

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

Part Two

Inspectors

Primitive Objects

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

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
boolean <string>	*NamedGlobal*	Creates the boolean value of the <string>, for example, • boolean "False" = FALSE. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
conjunction of <boolean>	*Plain*	This inspector performs a serial AND on all its boolean arguments: • conjunction of (true; true; true) -> TRUE • conjunction of (true; true; false) -> FALSE. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
disjunction of <boolean>	*Plain*	This inspector performs a serial OR on all its boolean arguments: • disjunction of (false; false; false) -> FALSE • disjunction of (false; false; true) -> TRUE. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
false	*PlainGlobal*	Creates a boolean with value FALSE. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
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, Lin:8.0, , WM, Ubu
infinite of <floating point>	*Plain*	Returns TRUE if the floating point number is infinite. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, , WM, Ubu
nan of <floating point>	*Plain*	Returns TRUE if the value is not a number. Win, Lin:8.0, Mac, WM, Ubu
normal of <floating point>	*Plain*	Returns TRUE if the value is a valid floating point number. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, , WM, Ubu
true	*PlainGlobal*	Creates a boolean with value TRUE. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin:8.0, , WM, Ubu

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, for example: • TRUE as string = "True". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
conjunction of <boolean>	*Plain*	<boolean>	This inspector performs a serial AND on all its boolean arguments: • conjunction of (true; true; true) -> TRUE • conjunction of (true; true; false) -> FALSE. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
disjunction of <boolean>	*Plain*	<boolean>	This inspector performs a serial OR on all its boolean arguments: • disjunction of (false; false; false) -> FALSE • disjunction of (false; false; true) -> TRUE. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<boolean> {cmp} <boolean>	*<boolean>*	Compare two boolean expressions. Returns another boolean, depending on the evaluation of the comparison: • {cmp} is one of: =, != . Win, Lin, Sol, HPUX, AIX, Mac, WM
<boolean> {op} <boolean>	*<boolean>*	Operates on two boolean expressions. Returns another boolean, depending on the evaluation of the operation, for example, (True And True) = True. • {op} is one of: And, Or . Win, Lin, Sol, HPUX, AIX, Mac, WM
<floating point> {cmp} <floating point>	*<boolean>*	Compares two floating point numbers, where: • {cmp} is one of: =, <, <=. Win, Lin:8.0, Mac, WM
<floating point> {cmp} <integer>	*<boolean>*	Compares a floating point number and an integer, where: • {cmp} is one of: =, <=, <. Win, Lin:8.0, Mac, WM
<integer> {cmp} <floating point>	*<boolean>*	Compares an integer to a floating point number, where: • {cmp} is one of: =, <=, <. Win, Lin:8.0, Mac, WM
<time interval> {cmp} <time interval>	*<boolean>*	Compare two time intervals, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, Lin, Sol, HPUX, AIX, Mac, WM

Key phrase	Return Type	Description
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin:8.0, Mac, WM, Ubu
<integer> as integer	*Cast*	Integer casting for completeness. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as integer	*Cast*	Converts from a string to an integer. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
absolute value of <integer>	*Plain*	Creates the positive value of the <integer> object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
greatest integer	*PlainGlobal*	Creates the value 9,223,372,036,854,775,807. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
hexadecimal integer <string>	*NamedGlobal*	Creates an integer from the provided hexadecimal value. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
integer <integer>	*NumberedGlobal*	Creates a global object with the given integer value, for example, Integer 123. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
integer <string>	*NamedGlobal*	Creates a global object with the integer value given by a string, for example, Integer "123" creates the value 123. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu
integer to <integer>	*NumberedGlobal*	Returns a list of integers from zero up to the specified integer. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
least integer	*PlainGlobal*	Creates the value -9,223,372,036,854,775,808. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
length of <rope>	*Plain*	Creates an integer object corresponding to the number of bytes in the rope. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
length of <string>	*Plain*	Creates an integer object corresponding to the number of bytes in the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
maximum of <integer>	*Plain*	Returns the maximum of a list of integers. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
minimum of <integer>	*Plain*	Returns the minimum of a list of integers. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
numeric value of <string>	*Plain*	Creates an integer object containing the value of the first number contained in a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
product of <integer>	*Plain*	Multiplies a list of integers, returning the product. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
significant digits <integer> of <integer>	*Numbered*	Creates a number with <integer> significant digits (for example. significant digits 3 of 1235569 = 1240000). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
sum of <integer>	*Plain*	Returns the sum of a list of integers. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> as bits	*Cast*	<bit set>	Returns the bits of the binary representation of the integer; bit zero is the least-significant bit. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> as day_of_month	*Cast*	<day of month>	Cast an integer as a day of the month type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<integer> as floating point	*Cast*	<floating point>	Converts an integer into a floating point number. Win, Lin:8.0, Mac, WM, Ubu
<integer> as hexadecimal	*Cast*	<string>	Converts an integer into a hexadecimal string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> as integer	*Cast*	<integer>	Reflexive cast for completeness. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> as month	*Cast*	<month>	Returns the name of the nth month of the year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<integer> as string	*Cast*	<string>	Converts an integer to a string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> as year	*Cast*	<year>	Casts an integer as a year type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
absolute value of <integer>	*Plain*	<integer>	Returns the positive value of the integer. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
extrema of <integer>	*Plain*	<(integer, integer)>	Returns the minimum and maximum extreme values of the given list of <integer> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <integer>	*Plain*	<integer>	Returns the maximum of a list of integers. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
mean of <integer>	*Plain*	<floating point>	The mean of the integer(s). Win, Mac
minimum of <integer>	*Plain*	<integer>	Returns the minimum of a list of integers. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
product of <integer>	*Plain*	<integer>	Multiplies a list of integers, returning the product. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
set of <integer>	*Plain*	<integer set>	Creates a set from the given list of semicolon-separated integers. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
significant digits <integer> of <integer>	*Numbered*	<integer>	Returns a number with <integer> significant digits (for example, significant digits 3 of 1235569 = 1240000). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
standard deviation of <integer>	*Plain*	<floating point>	The standard deviation of the integer(s). Win, Mac
sum of <integer>	*Plain*	<integer>	Returns the sum of a list of integers. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
unique value of <integer>	*Plain*	<integer with multiplicity>	Returns the unique values of a given list of <integer> types, removing duplicates and sorting by value. Win, Lin:6.0, Sol:6.0, HPUX, AIX, Mac, WM, Ubu

Operators

Key phrase	Return Type	Description
- <integer>	*<integer>*	Creates the negative of the integer provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key phrase	Return Type	Description
<floating point> {cmp} <integer>	*<boolean>*	Compares a floating point number and an integer, where: • {cmp} is one of: =, <=, <. Win, Lin:8.0, Mac, WM
<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, Lin:8.0, Mac, WM
<hertz> {op} <integer>	*<hertz>*	Returns a hertz object operated on by the given integer, where: • {op} is one of: *, / . Win, Lin, Sol, HPUX, AIX, Mac, WM
<integer set> contains <integer>	*<boolean>*	Returns TRUE if the specified set contains the given integer. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<integer> * <number of months>	*<number of months>*	Multiply a number of months by an integer, producing a new number of months. This is a typical technique to create a value of this type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<integer> * <time range>	*<timed(time range, integer)>*	Returns a tuple of a time interval and an integer. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> {cmp} <floating point>	*<boolean>*	Compares an integer to a floating point number, where: • {cmp} is one of: =, <=, <. Win, Mac, WM
<integer> {cmp} <integer>	*<boolean>*	Returns boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, Lin, Sol, HPUX, AIX, Mac, WM
<integer> {cmp} <registry key value type>	*<boolean>*	Returns boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, WM
<integer> {cmp} <registry key value>	*<boolean>*	Returns boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, WM

Key phrase	Return Type	Description
<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, Lin:8.0, Mac, WM
<integer> {op} <integer>	*<integer>*	Returns the integer solution to the equation, depending on the operator, where: • {op} is one of: +, -, *, /, mod . Win, Lin, Sol, HPUX, AIX, Mac, WM
<number of months> {op} <integer>	*<number of months>*	Where {op} is one of: *, /. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Examples

- `hexadecimal integer "A0"`
  - Returns 160.
- `numeric value of "string 123 xyz 45" = 123`
  - Returns TRUE.
- `255 as hexadecimal`
  - Returns the string "ff".
- `maximum of (7;2;4;5)`
  - Returns 7.
- `set of (3; 2; 2; 1; -1) contains 4`
  - Returns FALSE.
- `set of (3; 2; 2; 1; -1) contains set of (2; -1)`
  - Returns TRUE.
- `set of (3; 2; 2; 1; -1) = set of (2; -1)`
  - Returns FALSE.
- `significant digits 3 of 1235569`
  - Returns 1240000.

- `set of (1;2;3) contains 3`
  - Returns TRUE.

- `21 mod 5`
  - Returns 1.

Integer with Multiplicity

These Inspectors deal with arrays of integers, allowing you to extract unique numbers and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <integer>	*Plain*	Returns the unique values of a given list of integers, removing duplicates and sorting by value. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Integer Set

These Inspectors deal with sets of integers, which are essentially lists or arrays with integer elements. Think of them as mathematical sets: you can compare them, subtract them from other sets and form the union and intersection of multiple sets.

- Note: These Inspectors are not available on SUSE Linux systems.

Creation Methods

Key Phrase	*Form*	Description
intersection of <integer set>	*Plain*	Returns a set of integers equal to the intersection of the specified sets. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
set of <integer>	*Plain*	Creates a set from the given integers (usually plural). Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
union of <integer set>	*Plain*	Returns a set of integers equal to the union of the specified sets. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
element of <integer set>	*Plain*	<integer>	Returns the unique elements of the specified <integer set>, removing duplicates and sorting by value. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
intersection of <integer set>	*Plain*	<integer set>	Returns a set of integers equal to the intersection of the specified sets, in numeric order and with redundant elements stripped out. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
size of <integer set>	*Plain*	<integer>	Returns the number of unique elements in the specified set. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
union of <integer set>	*Plain*	<integer set>	Returns a set of integers equal to the union of the specified sets, in numeric order and with redundant elements stripped out. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<integer set> - <integer set>	*<integer set>*	Subtracts the elements in the second set from the elements in the first. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<integer set> * <integer set>	*<integer set>*	Returns the intersection of the two specified sets. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<integer set> + <integer set>	*<integer set>*	Returns the union of the specified sets. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<integer set> = <integer set>	*<boolean>*	Returns TRUE if the specified sets have identical contents. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Key phrase	Return Type	Description
<integer set> contains <integer set>	*<boolean>*	Returns TRUE if the first set contains all the elements of the second set. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<integer set> contains <integer>	*<boolean>*	Returns TRUE if the specified set contains the given integer. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Examples

- `elements of intersection of (set of (3;2;1); set of (2;3;4))`
  - Returns the list: 2,3.
- `intersection of (set of (-1; 2); set of (2; 1; 3); set of (2; 5))`
  - Returns 2.
- `set of (3; 2; 2; 1; -1) contains 3`
  - Returns TRUE.
- `set of (3; 2; 2; 1; -1) contains set of (2; -1)`
  - Returns TRUE.
- `set of (3; 2; 2; 1; -1) = set of (2; -1)`
  - Returns FALSE.
- `elements of union of (set of (1;2;3); set of (2;3;4); set of (5;6))`
  - Returns the list: 1,2,3,4,5,6.
- `elements of set of (5;7;2;3;3;3)`
  - Returns the integers 2,3,5,7.
- `elements of intersection of (set of (3;2;1); set of (2;3;4))`
  - Returns the list: 2,3.
- `intersection of (set of (-1; 2); set of (2; 1; 3); set of (2; 5))`
  - Returns 2.
- `size of set of (3; 2; 2; 1; -1)`
  - Returns 4, the number of unique elements in the defined set.
- `elements of union of (set of (1;2;3); set of (2;3;4))`
  - Returns the list: 1,2,3,4.

- `elements of (set of (1;2;3) - set of (2;3;4))`
  - Returns 1.
- `elements of (set of (1;2;3) * set of (2;3;4))`
  - Returns the list: 2,3.
- `elements of (set of (1;2;3) * set of (2;3;4) * set of (3;4;5))`
  - Returns 3.
- `elements of (set of (1;2;3) + set of (2;3;4))`
  - Returns the list: 1,2,3,4.
- `set of (3; 2; 2; 1; -1) = set of (2; -1)`
  - Returns FALSE.
- `set of (1;2;3) contains set of (2;3)`
  - Returns TRUE.
- `set of (1;2;3) contains 2`
  - Returns TRUE.

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, Lin:8.0, Mac, WM, Ubu
<string> as floating point	*Cast*	Converts the contents of a string into a floating point number. Win, Lin:8.0, Mac, WM, Ubu
floating point <floating point>	*Index<floating point>Global*	Creates a floating point type object from the specified floating point number. Win:7.2, Lin:8.0, Mac:7.2, WM, Ubu
floating point <string>	*NamedGlobal*	Creates a floating point number from the provided string. Win, Lin:8.0, Mac, WM, Ubu

Key Phrase	*Form*	Description
less significance <integer> of <floating point>	*Numbered*	Removes <integer> number of digits of significance from the floating point value. Win, Lin:8.0, Mac, WM, Ubu
maximum of <floating point>	*Plain*	Returns the maximum value from a list of <floating point> types. Win:7.1, Lin:8.0, Mac:7.1, WM, Ubu
mean of <floating point>	*Plain*	The mean of the floating point number(s). Win, Mac
mean of <integer>	*Plain*	The mean of the integer(s). Win, Mac
minimum of <floating point>	*Plain*	Returns the minimum value from a list of <floating point> types. Win:7.1, Lin:8.0, Mac:7.1, WM, Ubu
more significance <integer> of <floating point>	*Numbered*	Adds <integer> number of digits of significance to the floating point value. Win, Lin:8.0, Mac, WM, Ubu
product of <floating point>	*Plain*	Multiplies a list of floating point numbers, returning a floating point product. Win:8.0, Lin:8.0, Mac:8.0, Ubu
relative significance place <integer> of <floating point>	*Numbered*	The same floating point value, to be expressed to the given number of significant digits. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu
significance place <integer> of <floating point>	*Numbered*	The same floating point value, to be expressed to the given decimal place. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu

Key Phrase	*Form*	Description
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, Lin:8.0, Mac, WM, Ubu
standard deviation of <floating point>	*Plain*	The standard deviation of the floating point number(s). Win, Mac
standard deviation of <integer>	*Plain*	The standard deviation of the integer(s). Win, Mac
sum of <floating point>	*Plain*	Returns the sum of a list of floating point numbers. Win:8.0, Lin:8.0, Mac:8.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<floating point> as floating point	*Cast*	<floating point>	This casting operator is added for completeness. It takes a floating point number and casts it as a floating point number. It facilitates automatic relevance generation where the software is not aware of the input types. Win:7.2, Lin:8.0, Mac:7.2, WM, Ubu
<floating point> as integer	*Cast*	<integer>	Rounds off and casts a floating point number as an integer. Win, Lin:8.0, Mac, WM, Ubu
<floating point> as scientific notation	*Cast*	<string>	Converts a floating point number into a string with scientific notation. Win, Lin:8.0, Mac, WM, Ubu
<floating point> as standard notation	*Cast*	<string>	Converts a floating point number into a string with standard notation. Win, Lin:8.0, Mac, WM, Ubu
<floating point> as string	*Cast*	<string>	Converts a floating point number into a string with standard notation. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, , WM, Ubu

Key Phrase	*Form*	Return Type	Description
extrema of <floating point>	*Plain*	<(floating point, floating point)>	Returns the minimum and maximum extreme values of the given list of <floating point> types. Win:7.1, Lin:8.0, Mac:7.1, WM, Ubu
finite of <floating point>	*Plain*	<boolean>	Returns TRUE if the floating point number is finite. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, , WM, Ubu
infinite of <floating point>	*Plain*	<boolean>	Returns TRUE if the floating point number is infinite. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, , WM, Ubu
less significance <integer> of <floating point>	*Numbered*	<floating point>	Removes <integer> number of digits of significance from the floating point value. Win, Lin:8.0, Mac, WM, Ubu
maximum of <floating point>	*Plain*	<floating point>	Returns the maximum value from a list of <floating point> types. Win:7.1, Lin:8.0, Mac:7.1, WM, Ubu
mean of <floating point>	*Plain*	<floating point>	The mean of the floating point number(s). Win, Mac
minimum of <floating point>	*Plain*	<floating point>	Returns the minimum value from a list of <floating point> types. Win:7.1, Lin:8.0, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
more significance <integer> of <floating point>	*Numbered*	<floating point>	Adds <integer> number of digits of significance to the floating point value. Win, Lin:8.0, Mac, WM, Ubu
nan of <floating point>	*Plain*	<boolean>	Returns TRUE if the value is not a number. Win, Lin:8.0, Mac, WM, Ubu
normal of <floating point>	*Plain*	<boolean>	Returns TRUE if the value is a valid floating point number. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, , WM, Ubu
product of <floating point>	*Plain*	<floating point>	Multiplies a list of floating point numbers, returning a floating point product. Win:8.0, Lin:8.0, Mac:8.0, Ubu
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, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu
significance place <integer> of <floating point>	*Numbered*	<floating point>	The same floating point value, to be expressed to the given decimal place. Win, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu
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, Lin:8.0, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
standard deviation of <floating point>	*Plain*	<floating point>	The standard deviation of the floating point number(s). Win, Mac
sum of <floating point>	*Plain*	<floating point>	Returns the sum of a list of floating point numbers. Win:8.0, Lin:8.0, Mac:8.0, Ubu
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, Lin:8.0, , WM, Ubu
unique value of <floating point>	*Plain*	<floating point with multiplicity>	Returns the unique values of a given list of <floating point> types, removing duplicates and sorting by value. Win:7.1, Lin:8.0, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
- <floating point>	*<floating point>*	Change the sign of a floating point number. Win, Lin:8.0, Mac, WM, Ubu
<floating point> {op} <floating point>	*<floating point>*	Operates on two floating point numbers, returning another floating point number, where: • {op} is one of: +, -, *, /. Win, Lin:8.0, Mac, WM
<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, Lin:8.0, Mac, WM
<integer> {cmp} <floating point>	*<boolean>*	Compares an integer to a floating point number, where: • {cmp} is one of: =, <=, <, >, >=, !=. Win, Lin:8.0, Mac, WM
<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, Lin:8.0, Mac, WM

Examples

- `"4.5" as floating point`
  - Returns 4.5.
- `floating point (floating point "5.2")`
  - Returns a floating point object of 5.2.
- `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.

Floating Point with Multiplicity

These Inspectors deal with floating point arrays, allowing you to extract unique floating point numbers and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <floating point>	*Plain*	Returns the unique values of a given list of <floating point> types, removing duplicates and sorting by value. Win:7.1, Lin:8.0, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <floating point with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <floating point> types. Win:7.1, Lin:8.0, Mac:7.1, WM, Ubu

String

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

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

Creation Methods

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

Key Phrase	*Form*	Description
<boolean> as string	*Cast*	Operates on a boolean to return a string. Possible values are "True" and "False". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<date> as string	*Cast*	Cast a date type as a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<floating point> as scientific notation	*Cast*	Converts a floating point number into a string with scientific notation. Win, Lin:8.0, Mac, WM, Ubu
<floating point> as standard notation	*Cast*	Converts a floating point number into a string with standard notation. Win, Lin:8.0, Mac, WM, Ubu
<floating point> as string	*Cast*	Converts a floating point number into a string with standard notation. Win, Lin:8.0, Mac, WM, Ubu
<hertz> as string	*Cast*	Creates a string containing the number of hertz and the word hertz, for example, (3 * hz) as string = "3 hertz". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> as hexadecimal	*Cast*	Converts an integer into a hexadecimal string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> as string	*Cast*	Creates a string formatted with the integer provided. (-22) as string = "-22". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
<string> as hexadecimal	*Cast*	Converts a string to a hexadecimal number. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as left trimmed string	*Cast*	Trims the leading spaces from a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as lowercase	*Cast*	Creates a lowercase version of the string provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as right trimmed string	*Cast*	Trims the trailing spaces from a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as string	*Cast*	Reflexive cast of string to string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as trimmed string	*Cast*	Trims the leading and trailing spaces off of the specified string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as uppercase	*Cast*	Creates an uppercase version of the string provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time zone> as string	*Cast*	Creates a string containing a time zone. See <time zone>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time> as local string	*Cast*	Creates a string containing a time. See <time>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time> as string	*Cast*	Creates a string containing a time. See <time>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time> as universal string	*Cast*	Creates a string containing a time. See <time>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<undefined> as string	*Cast*	Casts the 'undefined' error as a string. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
character <integer>	*NumberedGlobal*	Creates a string containing the single ASCII character for the decimal number provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
concatenation <string> of <string>	*Named*	This inspector concatenates the string items in the second argument with a separator defined by the string argument. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
concatenation of <string>	*Plain*	Combines the supplied strings into a single string, end-to-end. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
hexadecimal string <string>	*NamedGlobal*	Creates a string from the given hexadecimal value. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
regex escape of <string>	*Plain*	Creates a new string that escapes the special characters .[]{}()*+?|^$,-. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
string <string>	*NamedGlobal*	Creates a string matching the name provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
tuple string item <integer> of <string>	*Numbered*	Parses the given string as if it were a tuple result string and returns the string which is the nth item of the tuple. The tuple Inspectors used to be session-only, but as of version 8.0, they are core Inspectors, available to all platforms at all times. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, , WM:8.0, Ubu
tuple string item of <string>	*Plain*	Parses the given string as if it were a tuple result string and iterates over the <string> objects that make up the tuple. The tuple Inspectors used to be session-only, but as of version 8.0, they are core Inspectors, available to all platforms at all times. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, , WM:8.0, Ubu
two digit hour of <time of day with time zone>	*Plain*	Returns the hour of the zoned time of day as text, with values less than 10 having a leading zero. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
two digit minute of <time of day with time zone>	*Plain*	Returns the minute of the zoned time of day as text, with values less than 10 having a leading zero. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
two digit second of <time of day with time zone>	*Plain*	Returns the second of the zoned time of day as text, with values less than 10 having a leading zero. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<string> as boolean	*Cast*	<boolean>	Returns a boolean value for the string. All possible capitalization's of "TRUE" and "FALSE" will convert successfully. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as date	*Cast*	<date>	Casts a string as a date type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as day_of_month	*Cast*	<day of month>	Casts a string as a day of the month (eg. 28). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as day_of_week	*Cast*	<day of week>	Casts a string as a day of the week. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as floating point	*Cast*	<floating point>	Converts the contents of a string into a floating point number. Win, Lin:8.0, Mac, WM, Ubu
<string> as hexadecimal	*Cast*	<string>	Converts a string to a hexadecimal number. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as html	*Cast*	<html>	Casts a string into html. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as ipv4or6 address	*Cast*	<ipv4or6 address>	Converts a string version of an IP address into the flexible ipv4or6 address format. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<string> as ipv6 address	*Cast*	<ipv6 address>	Converts a string representations of an IPv6 address (with colons and/or dots) as an IPv6 address type. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
<string> as left trimmed string	*Cast*	<string>	Trims the leading spaces from a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as local time	*Cast*	<time>	Returns a local time object from a properly formatted string. See <time>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
<string> as local zoned time_of_day	*Cast*	<time of day with time zone>	Converts a string to a time of day with local time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as lowercase	*Cast*	<string>	Returns a lowercase version of the string provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as month	*Cast*	<month>	Converts a string into a month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as right trimmed string	*Cast*	<string>	Trims the trailing spaces from a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as site version list	*Cast*	<site version list>	Converts a string into a site version list. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.0, Mac:7.1, WM, Ubu
<string> as string	*Cast*	<string>	Returns the string provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as time	*Cast*	<time>	Returns a time object from a properly formatted string. See <time>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as time zone	*Cast*	<time zone>	Returns a time zone object from a properly formatted string. See <time zone>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as time_of_day	*Cast*	<time of day>	Converts a string to a time_of_day type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as trimmed string	*Cast*	<string>	Trims the leading and trailing spaces off of the specified string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as universal time	*Cast*	<time>	Returns a universal time object from a properly formatted string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as universal zoned time_of_day	*Cast*	<time of day with time zone>	Converts a string into a universal zoned time of day. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
<string> as uppercase	*Cast*	<string>	Returns an uppercase version of the string provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as windows display time	*Cast*	<time>	Returns a Windows display time object from a properly formatted string. See <Time>. Win, WM
<string> as year	*Cast*	<year>	Converts a string into a year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as zoned time_of_day	*Cast*	<time of day with time zone>	Converts a string into a zoned time of day. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
abbr <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <abbr> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
abbr of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <abbr> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
acronym <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <acronym> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
acronym of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <acronym> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
address <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <address> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
address of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <address> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
anchor <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <a> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
anchor of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <a> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
b <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <b> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
b of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <b> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
base <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <base> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
base of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <base> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
big <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <big> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
big of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <big> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
blockquote <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <blockquote> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
blockquote of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <blockquote> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
body <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <body> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
body of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <body> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
caption <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <caption> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
caption of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <caption> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
character of <string>	*Plain*	<substring>	Returns the characters from the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
cite <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <cite> (citation) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
cite of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <cite> (citation) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
code <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <code> (fixed-width font) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
code of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <code> (fixed-width font) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
col <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <col> (column) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
col of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <col> (column) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
colgroup <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <colgroup> (column group) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
colgroup of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <colgroup> (column group) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
concatenation <html> of <string>	*Index<html>*	<html>	This inspector concatenates the string items in the second argument with a separator defined by the first argument. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, Ubu
concatenation <string> of <string>	*Named*	<string>	This inspector concatenates the string items in the second argument with a separator defined by the string argument. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
concatenation of <string>	*Plain*	<string>	Combines the supplied strings into a single string, end-to-end. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
dd <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <dd> (definition) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dd of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <dd> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
definition list <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <dl> (definition) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
definition list of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <dl> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
del <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <del> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
del of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <del> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dfn <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <dfn> (definition) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
dfn of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <dfn> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
div <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a (division or section) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
div of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a (division or section) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dt <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <dt> (definition) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dt of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <dt> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
em <string> of <string>	*Named*	<html>	Emphasize the specified string inside an <em string> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
em of <string>	*Plain*	<html>	Emphasize the specified string inside an tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
first <string> of <string>	*Named*	<substring>	Returns a substring containing the first occurrence of the name provided. See substring. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
h1 <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <h1> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h1 of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <h1> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
h2 <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <h2> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h2 of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <h2> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h3 <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <h3> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h3 of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <h3> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h4 <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <h4> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h4 of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <h4> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h5 <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <h5> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h5 of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <h5> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h6 <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <h6> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h6 of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <h6> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
head <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <head> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
head of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <head> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <html> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <html> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html tag <string> of <string>	*Named*	<html>	Returns an HTML snippet with the second string enclosed in a tag specified by the first string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ins <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <ins> (insert) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ins of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <ins> (insert) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
italic <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <i> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
italic of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <i> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
kbd <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <kbd> (keyboard entry) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
kbd of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <kbd> (keyboard entry) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
last <string> of <string>	*Named*	<substring>	Returns a substring containing the last occurrence of the name provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
length of <string>	*Plain*	<integer>	Returns the number of characters in the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
li <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a (list) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
li of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a (list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
link <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <link> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
link of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <link> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
meta <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <meta> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
meta of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <meta> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
numeric value of <string>	*Plain*	<integer>	Returns an integer for the first numeric value in the string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
ol <string> of <string>	*Named*	<html>	Creates an ordered list out of <string2> with an optional style specified by <string1>. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ol of <string>	*Plain*	<html>	Creates an ordered list out of the <string>. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ordered list <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an tag, where the tag is modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
ordered list of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
p <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <p> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
p of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <p> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
position of <string>	*Plain*	<string position>	Returns the positions of the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
pre <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in an <pre> (preformatted) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
pre of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in an <pre> (preformatted) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
q <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <q> (quotation) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
q of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <q> (quotation) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
regex escape of <string>	*Plain*	<string>	Creates a new string that escapes the special characters .[]{}()*+?|^$,-. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
samp <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <samp> (sample) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
samp of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <samp> (sample) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
set of <string>	*Plain*	<string set>	Creates a set from the given list of semicolon-separated strings. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
small <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <small> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
small of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <small> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
span <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
span of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
strong <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
strong of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sub <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <sub> (subscript) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sub of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <sub> (subscript) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
substring <(integer, integer)> of <string>	*Index<(integer, integer)>*	<substring>	Returns a substring of the specified string as defined by the start and length. The first integer is the start and the second integer is the length. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	*Form*	Return Type	Description
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
substring after <string> of <string>	*Named*	<substring>	Returns the substrings that come after the first string delimiter. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
substring before <string> of <string>	*Named*	<substring>	Returns the substrings that come before the first string delimiter. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
substring between <string> of <string>	*Named*	<substring>	Returns the substring in the second string found between two instances of the first string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
substring separated by <string> of <string>	*Named*	<substring>	Returns a substring (or set of substrings) delimited by the first string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
sup <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <sup> (superscript) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sup of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <sup> (superscript) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
table <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <table> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
table of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <table> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tbody <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <tbody> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tbody of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <tbody> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
td <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <td> (table cell) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
td of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <td> (table cell) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tfoot <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <tfoot> (table foot) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tfoot of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <tfoot> (table foot) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
th <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <th> (table header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
th of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <th> (table header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
thead <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <thead> (table header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
thead of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <thead> (table header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
title <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <title> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
title of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <title> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tr <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <tr> (table row) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
tr of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <tr> (table row) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tt <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <tt> (teletype font) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tt of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <tt> (teletype font) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tuple string item <integer> of <string>	*Numbered*	<string>	Parses the given string as if it were a tuple result string and returns the string which is the nth item of the tuple. The tuple Inspectors used to be session-only, but as of version 8.0, they are core Inspectors, available to all platforms at all times. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, , WM:8.0, Ubu
tuple string item of <string>	*Plain*	<string>	Parses the given string as if it were a tuple result string and iterates over the <string> objects that make up the tuple. The tuple Inspectors used to be session-only, but as of version 8.0, they are core Inspectors, available to all platforms at all times. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, , WM:8.0, Ubu
ul <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a (unordered list) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ul of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a (unordered list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
unique value of <string>	*Plain*	<string with multiplicity>	Returns the unique values of a given list of <string> types, removing duplicates and sorting by value. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
unordered list <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a (unordered list item) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
unordered list of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a (unordered list item) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
var <string> of <string>	*Named*	<html>	Returns an HTML snippet with the specified string enclosed in a <var> (variable type) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
var of <string>	*Plain*	<html>	Returns an HTML snippet containing the specified string enclosed in a <var> (variable type) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Operators

Key phrase	Return Type	Description
<html> & <string>	*<html>*	Concatenates a string with an HTML file, returning a new HTML file. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
<rope> & <string>	*<rope>*	Concatenates a rope and a string, producing a rope. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<rope> contains <string>	*<boolean>*	Returns boolean TRUE if the rope contains the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string set> contains <string>	*<boolean>*	Returns TRUE if the specified set of strings contains the given string. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<string> & <html>	*<html>*	Concatenates a string with an HTML file, returning a new HTML file. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
<string> & <rope>	*<rope>*	Concatenates a rope and a string, returning a new rope. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> & <string>	*<string>*	Concatenates two strings, producing a new string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> {cmp} <string>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, Lin, Sol, HPUX, AIX, Mac, WM

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".
- `html tag "i" of "italic string"`
  - Returns italic string.
- `preceding text of last "ab" of "abracadabra" is "abracad"`
  - Returns True.
- `regex escape of "[square] or {curly}"`
  - Returns \[square\] or \{curly\}.
- `substring (1,2) of "abcdefg"`
  - Returns "bc".
- `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.

- `set of ("foo";"bar") contains "foo"`

  Returns TRUE.

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.

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

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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 a part of a larger string and has all the properties of a string. Substrings also have the following methods and properties.

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

Creation Methods

Key Phrase	*Form*	Description
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
first <integer> of <string>	*Numbered*	Creates a substring for the given number of characters at the start of the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
last <string> of <string>	*Named*	Creates a substring containing the last occurrence of the name provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
substring after <string> of <string>	*Named*	Returns the substrings that come after the first string delimiter. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
substring before <string> of <string>	*Named*	Returns the substrings that come before the first string delimiter. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
substring between <string> of <string>	*Named*	Returns the substring in the second string found between two instances of the first string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
substring separated by <string> of <string>	*Named*	Returns a substring (or set of substrings) delimited by the first string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
following text of <substring>	*Plain*	<substring>	Returns the string following the substring. For example, following text of last "." of "log.txt" = "txt". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
preceding text of <substring>	*Plain*	<substring>	Returns the string preceding the substring. For example, preceding text of last "." of "log.txt" = "log". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

NOTE: All the string operators can also be applied to substrings.

Examples

- `substrings after ":" of "definition: after the colon"`
  - Returns " after the colon".

- `substrings after (character 3 of it) of "woe-is-me"`
  - Returns 'is' and 'me', since character 3 (zero-based) is a dash and this Inspector returns the next substrings separated by dashes.

- `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 extract unique strings and count them.

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

Creation Methods

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

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

String Set

These Inspectors deal with sets of strings, which are essentially lists or arrays with string elements. Think of them as mathematical sets: you can compare them, subtract them from other sets and form the union and intersection of multiple sets.

- Note: These Inspectors are not available on SUSE Linux systems.

Creation Methods

Key Phrase	*Form*	Description
intersection of <string set>	*Plain*	Returns a set of strings equal to the intersection of the specified sets. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
set of <string>	*Plain*	Creates a set from the given strings (usually plural). Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
union of <string set>	*Plain*	Returns a set of strings equal to the union of the specified sets. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
element of <string set>	*Plain*	<string>	Returns the unique elements of the specified <string set>, removing duplicates and sorting by value. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
intersection of <string set>	*Plain*	<string set>	Returns a set of strings equal to the intersection of the specified sets, aphabetized and with redundant elements stripped out. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
size of <string set>	*Plain*	<integer>	Returns the number of unique elements in the specified set. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
union of <string set>	*Plain*	<string set>	Returns a set of strings equal to the union of the specified sets, alphabetized and with redundant elements stripped out. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<string set> - <string set>	*<string set>*	Subtracts the elements in the second set from the elements in the first. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<string set> * <string set>	*<string set>*	Returns the intersection of the specified sets. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<string set> + <string set>	*<string set>*	Returns the union of the specified sets. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<string set> = <string set>	*<boolean>*	Returns TRUE if the specified sets have identical contents. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<string set> contains <string set>	*<boolean>*	Returns TRUE if the first set contains all the elements of the second set. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
<string set> contains <string>	*<boolean>*	Returns TRUE if the specified set of strings contains the given string. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu

Examples

- `elements of intersection of (set of ("to";"be"); set of ("or";"not";"to";"be"))`
  - Returns the list: be,to.

- `elements of union of (set of ("to";"be"); set of ("or";"not";"to";"be"))`
  - Returns the list: be,not,or,to.

- `elements of set of ("beta";"beta";"alpha";"gamma";"beta")`
  - Returns the strings alpha, beta, gamma.

- `elements of intersection of (set of ("to";"be"); set of ("or";"not";"to";"be"))`
  - Returns the list: be,to.

- `size of set of ("to";"be"; "or"; "not"; "to"; "be")`
  - Returns 4, the number of unique strings in the set.

- `elements of union of (set of ("to";"be"); set of ("or";"not";"to";"be"))`
  - Returns the list: be,not,or,to.

- `elements of (set of ("to";"be";"or") - set of ("not";"to";"be"))`
  - Returns or.

- `elements of (set of ("fee";"fie";"foe") - set of ("fee") - set of("foe"))`
  - Return "fie.".

- `elements of (set of ("to";"be";"or") * set of ("not";"to";"be"))`
  - Returns the list: be,to.

- `elements of (set of ("lime";"pie") * set of ("pie";"face") * set of("pie";"in";"sky"))`
  - Returns "pie.".

- `elements of (set of ("to";"be";"or") + set of ("not";"to";"be"))`
  - Returns the list: be,not,or,to.

- `set of ("to";"be";"or";"not") contains set of ("to";"be")`
  - Returns TRUE.

- `set of ("foo";"bar") contains "foo"`
  - Returns TRUE.

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

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

Operators

Key phrase	Return Type	Description
<rope> & <rope>	*<rope>*	Concatenates two ropes into a new rope. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<rope> & <string>	*<rope>*	Concatenates a rope and a string, producing a rope. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<rope> contains <string>	*<boolean>*	Returns TRUE if the rope contains the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> & <rope>	*<rope>*	Concatenates a rope and a string, returning a new rope. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<integer> as bits	*Cast*	Returns the bits of the binary representation of the integer; bit zero is the least-significant bit. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
bit set <string>	*NamedGlobal*	Returns the bits of the binary number given by the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
left shift <integer> of <bit set>	*Numbered*	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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<bit set> as integer	*Cast*	<integer>	Returns the integer whose binary representation matches the bit set. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<bit set> as string	*Cast*	<string>	Returns the bits (0s and 1s) in a string format. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
bit <integer> of <bit set>	*Numbered*	<boolean>	Returns the value of the bit at the given <integer> position in the set. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
least significant one bit of <bit set>	*Plain*	<integer>	Returns the least n such that bit n of the set is true. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
most significant one bit of <bit set>	*Plain*	<integer>	Returns the greatest n such that bit n of the set is true. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
one bit of <bit set>	*Plain*	<integer>	Returns the numbers n for which bit n of the set is true. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<bit set> * <bit set>	*<bit set>*	Returns the intersection of the two bit sets. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<bit set> + <bit set>	*<bit set>*	Returns the union of the two sets. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<bit set> = <bit set>	*<boolean>*	Returns true if the corresponding bits of the two sets are equal. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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 Wikipedia using the search term "posix-extended regex".

Creation Methods

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

Operators

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

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 Wikipedia using the search term "posix-extended regex".

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

Creation Methods

Key Phrase	*Form*	Description
first match <regular expression> of <string>	*Index<regular expression>*	Creates an object containing the first match to the regular expression in the given string. Win, Lin, Sol, HPUX, AIX, Mac, Ubu
match <regular expression> of <string>	*Index<regular expression>*	Creates an object containing all the matches to the regular expression in the given string. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

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, Lin, Sol, HPUX, AIX, Mac, Ubu
parenthesized part of <regular expression match>	*Plain*	<substring>	Returns the parenthetical part of the specified regular expression match. Win, Lin, Sol, HPUX, AIX, Mac, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<undefined> as string	*Cast*	<string>	Casts the 'undefined' error as a string. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
ghz	*PlainGlobal*	Creates a hertz object corresponding to 1 giga-hertz. For example, ghz = 1000*mhz. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
hz	*PlainGlobal*	Creates a hertz object corresponding to 1 hertz. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
khz	*PlainGlobal*	Creates a hertz object corresponding to 1 kilohertz. For example, khz = 1000*hz. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
maximum of <hertz>	*Plain*	Returns the maximum value from a list of <hertz> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
mhz	*PlainGlobal*	Creates a hertz object corresponding to 1 megahertz. For example, mhz = 1000*khz. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
minimum of <hertz>	*Plain*	Returns the minimum value from a list of <hertz> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
significant digits <integer> of <hertz>	*Numbered*	Rounds up the value of a hertz object with <integer> significant digits. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<hertz> as string	*Cast*	<string>	Returns a string formatted "##### hertz". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
absolute value of <hertz>	*Plain*	<hertz>	Returns the positive value of the hertz object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
extrema of <hertz>	*Plain*	<(hertz, hertz)>	Returns the minimum and maximum extreme values of the given list of <hertz> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <hertz>	*Plain*	<hertz>	Returns the maximum value from a list of <hertz> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <hertz>	*Plain*	<hertz>	Returns the minimum value from a list of <hertz> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
significant digits <integer> of <hertz>	*Numbered*	<hertz>	Returns the value of a hertz object with <integer> significant digits (for example. significant digits 3 of 1235569 = 1240000). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
unique value of <hertz>	*Plain*	<hertz with multiplicity>	Returns the unique values of a given list of <hertz> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

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

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.

Hertz with Multiplicity

These Inspectors deal with hertz arrays, allowing you to extract unique hertz values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <hertz>	*Plain*	Returns the unique values of a given list of <hertz> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <hertz with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <hertz> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
<string> as time	*Cast*	Parses the string. Time zone information must be provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as universal time	*Cast*	Parses the string. If time zone is not provided in the string, the universal time zone is assumed. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<string> as windows display time	*Cast*	Parses the string. If time zone is not provided in the string, the current time zone in effect at the given time is assumed. Win, WM
maximum of <time>	*Plain*	Returns the maximum time from a list of times. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
minimum of <time>	*Plain*	Returns the minimum time from a list of times. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
now	*PlainGlobal*	Creates an object for the current time. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
time <string>	*NamedGlobal*	The time inspector creates a time object by parsing the string literal provided. The zone info is required. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
windows display time <string>	*NamedGlobal*	Creates an object for a string that may match the time shown in the Windows file system. Win, WM

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time> as string	*Cast*	<string>	Same as above. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
date <time zone> of <time>	*Index<time zone>*	<date>	Returns the date adjusted for the specified time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <time>	*Plain*	<(time, time)>	Returns the minimum and maximum extreme values of the given list of <time> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <time>	*Plain*	<time>	Returns the maximum time from a list of times. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
minimum of <time>	*Plain*	<time>	Returns the minimum time from a list of times. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
time <time zone> of <time>	*Index<time zone>*	<time of day with time zone>	Adjusts the specified time to the given time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
unique value of <time>	*Plain*	<time with multiplicity>	Returns the unique values of a given list of <time> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time> & <time interval>	*<time range>*	Concatenates a time and a time interval, producing a time range object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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

NOTE:

The string format for a time object is given by the MIME standard. When output as a string, the format 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.

Time with Multiplicity

These Inspectors deal with time arrays, allowing you to extract unique time values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <time>	*Plain*	Returns the unique values of a given list of <time> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <time with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <time> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Time of Day

These Inspectors provide tools for dealing and calculating with time-of-day types, which are of the form HH:MM:SS, as in 12:59:59.

Creation Methods

Key Phrase	*Form*	Description
<string> as time_of_day	*Cast*	Converts a string to a time_of_day type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <time of day>	*Plain*	Returns the maximum value from a list of <time of day> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
midnight	*PlainGlobal*	Returns 00:00:00 as a time of day object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
minimum of <time of day>	*Plain*	Returns the minimum value from a list of <time of day> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
noon	*PlainGlobal*	Returns 12:00:00 as a time of day object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
time of <time of day with time zone>	*Plain*	Returns the time of day, without the time zone information. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
time_of_day <string>	*NamedGlobal*	Creates a time of day object out of the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<time of day> as string	*Cast*	<string>	Casts the time of day as a string type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <time of day>	*Plain*	<(time of day, time of day)>	Returns the minimum and maximum extreme values of the given list of <time of day> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
hour_of_day of <time of day>	*Plain*	<integer>	Returns the hour section of the 'time of day' object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <time of day>	*Plain*	<time of day>	Returns the maximum value from a list of <time of day> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <time of day>	*Plain*	<time of day>	Returns the minimum value from a list of <time of day> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minute_of_hour of <time of day>	*Plain*	<integer>	Returns the 'minutes after the hour' section of the 'time of day' object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
second_of_minute of <time of day>	*Plain*	<integer>	Extracts the 'seconds after the minute' section of the 'tim of day' object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
two digit hour of <time of day>	*Plain*	<string>	Extracts the 2-digit hour from the time of day. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
two digit minute of <time of day>	*Plain*	<string>	Extracts the 2-digit minute from the time of day. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
two digit second of <time of day>	*Plain*	<string>	Extracts the 2-digit second from the time of day. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
unique value of <time of day>	*Plain*	<time of day with multiplicity>	Returns the unique values of a given list of <time of day> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<time of day> - <time of day>	*<time interval>*	Subtracts two times of day, returning a time interval. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time of day> & <time zone>	*<time of day with time zone>*	Concatenates a time of day with a time zone, returning a time of day with time zone type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time of day> {cmp} <time of day>	*<boolean>*	Compares two times of day, where {cmp} is one of: <, <=, =. Win, WM
<time zone> & <time of day>	*<time of day with time zone>*	Concatenates a time of day with a time zone, returning a time of day with time zone type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Time of Day with Multiplicity

These Inspectors deal with time-of-day arrays, allowing you to extract unique time-of-day values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <time of day>	*Plain*	Returns the unique values of a given list of <time of day> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <time of day with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <time of day> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
local time zone	*PlainGlobal*	Creates a time zone object corresponding to the local time zone. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
time zone <string>	*NamedGlobal*	Creates a time zone object corresponding to the string provided. For example, time zone "edt" as string = "-0400". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
universal time zone	*PlainGlobal*	Creates a time zone object corresponding to the universal time zone. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<time zone> as string	*Cast*	<string>	Returns a string corresponding to the time zone object provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
unique value of <time zone>	*Plain*	<time zone with multiplicity>	Returns the unique values of a given list of <time zone> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<time of day> & <time zone>	*<time of day with time zone>*	Concatenates a time of day with a time zone, returning a time of day with time zone type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time zone> & <time of day with time zone>	*<time of day with time zone>*	Converts a 'time of day with time zone' to the time in the specified time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time zone> & <time of day>	*<time of day with time zone>*	Concatenates a time of day with a time zone, returning a time of day with time zone type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time zone> {op} <time interval>	*<time zone>*	Returns a time zone object offset by a time interval, where: • {op} is one of: +, - . Win, Lin, Sol, HPUX, AIX, Mac

Examples

- `local time zone - 2 * hour`

  Returns the time zone two hours away.

Time Zone with Multiplicity

These Inspectors deal with time zone arrays, allowing you to extract unique time zone values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <time zone>	*Plain*	Returns the unique values of a given list of <time zone> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <time zone with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <time zone> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Time of Day with Time Zone

These Inspectors provide tools for dealing and calculating with time-of-day-with-time-zone types, which are of the form HH:MM:SS +ZZZZ, as in 12:59:59 -0400.

Creation Methods

Key Phrase	*Form*	Description
<string> as local zoned time_of_day	*Cast*	Converts a string to a time of day with local time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as universal zoned time_of_day	*Cast*	Converts a string into a universal zoned time of day. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as zoned time_of_day	*Cast*	Converts a string into a zoned time of day. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
current time_of_day	*PlainGlobal*	Returns the current time of day in the local time zone. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
current time_of_day <time zone>	*Index<time zone>Global*	Returns the current time of day in the specified time zone. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
time <time zone> of <time>	*Index<time zone>*	Converts the specified time to the given time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
zoned time_of_day <string>	*NamedGlobal*	Creates a 'zoned time of day' out of a string object in the form of HH:MM:SS +ZZZZ. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<time of day with time zone> as string	*Cast*	<string>	Converts a 'time of day with time zone' object into a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
hour_of_day of <time of day with time zone>	*Plain*	<integer>	Returns the hour section of the 'time of day with time zone' object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
minute_of_hour of <time of day with time zone>	*Plain*	<integer>	Returns the 'minutes after the hour' section of the 'time of day with time zone' object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
second_of_minute of <time of day with time zone>	*Plain*	<integer>	Returns the 'seconds after the minute' section of the 'time of day with time zone' object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
time of <time of day with time zone>	*Plain*	<time of day>	Returns the time of day, without the time zone information. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
two digit hour of <time of day with time zone>	*Plain*	<string>	Returns the hour of the zoned time of day as text, with values less than 10 having a leading zero. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
two digit minute of <time of day with time zone>	*Plain*	<string>	Returns the minute of the zoned time of day as text, with values less than 10 having a leading zero. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
two digit second of <time of day with time zone>	*Plain*	<string>	Returns the second of the zoned time of day as text, with values less than 10 having a leading zero. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, WM, Ubu
unique value of <time of day with time zone>	*Plain*	<time of day with time zone with multiplicity>	Returns the unique values of a given list of <time of day with time zone> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
zone of <time of day with time zone>	*Plain*	<time zone>	Returns the zone associated with the specified time. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<date> & <time of day with time zone>	*<time>*	Concatenates a date with a time and a time zone for a complete time stamp. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time of day with time zone> - <time of day with time zone>	*<time interval>*	Subtracts two times of day (including time zones), returning a time interval. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time of day with time zone> & <time zone>	*<time of day with time zone>*	Concatenates a 'time of day with a time zone' and another time zone. The 'time of day with time zone' object that is produced is adjusted to fit the appended time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time of day with time zone> = <time of day with time zone>	*<boolean>*	Compares two times of day with time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time zone> & <time of day with time zone>	*<time of day with time zone>*	Converts a 'time of day with time zone' to the time in the specified time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Time of Day with Time Zone with Multiplicity

These Inspectors deal with time-of-day-with-time-zone arrays, allowing you to extract unique time-of-day-with-time-zone values and count them.

Type Derivation: This object type is derived from the <time of day with time zone> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	*Form*	Description
unique value of <time of day with time zone>	*Plain*	Returns the unique values of a given list of <time of day with time zone> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <time of day with time zone with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <time of day with time zone> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

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>	*Index<time interval>*	Returns a time range with the specified interval, but ending on the final date of the time range. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
initial part <time interval> of <time range>	*Index<time interval>*	Returns a time range starting with the first date of the time range and lasting for the specified interval. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
range after <time> of <time range>	*Index<time>*	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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
range before <time> of <time range>	*Index<time>*	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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<time range> as string	*Cast*	<string>	Casts a time range as a string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
end of <time range>	*Plain*	<time>	Returns the end date of a time range. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
final part <time interval> of <time range>	*Index<time interval>*	<time range>	Returns a time range with the specified interval, but ending on the final date of the time range. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
initial part <time interval> of <time range>	*Index<time interval>*	<time range>	Returns a time range starting with the first date of the time range and lasting for the specified interval. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
range after <time> of <time range>	*Index<time>*	<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
range before <time> of <time range>	*Index<time>*	<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
start of <time range>	*Plain*	<time>	Returns the starting date of a time range. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
unique value of <time range>	*Plain*	<time range with multiplicity>	Returns the unique values of a given list of <time range> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key phrase	Return Type	Description
<time range> * <time range>	*<time range>*	Returns the intersection of the two specified time ranges, if one exists. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time range> + <time range>	*<time range>*	Returns the smallest range that contains both ot the specified ranges (same as <time range> & <time range>). Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time range> = <time range>	*<boolean>*	Compares two time range types and returns TRUE if they are equal. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
<time> & <time range>	*<time range>*	Concatenates a time and a time range, producing a new time range. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Time Range with Multiplicity

These Inspectors deal with time-range arrays, allowing you to extract unique time-range values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <time range>	*Plain*	Returns the unique values of a given list of <time range> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <time range with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <time range> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
absolute value of <time interval>	*Plain*	Creates the positive value of a time interval. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
day	*PlainGlobal*	Creates a time interval corresponding to 1 day. For example, 2 * day = 48 * hour. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
greatest time interval	*PlainGlobal*	Creates the largest time interval that can be represented on the current machine. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
hour	*PlainGlobal*	Creates a time interval corresponding to 1 hour. For example, day = 24 * hour. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
least time interval	*PlainGlobal*	Creates the largest negative time interval that can be represented on the current machine. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
maximum of <time interval>	*Plain*	Returns the maximum interval from a list of time intervals. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
microsecond	*PlainGlobal*	Creates a time interval corresponding to 1 microsecond. For example, 1000 * microsecond = 1 * millisecond. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
millisecond	*PlainGlobal*	Creates a time interval corresponding to 1 millisecond. For example, 1000 * millisecond = 1 * second. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
minimum of <time interval>	*Plain*	Returns the minimum interval from a list of time intervals. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
minute	*PlainGlobal*	Creates a time interval corresponding to 1 minute. For example, minute = 60 * second. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
second	*PlainGlobal*	Creates a time interval corresponding to 1 second. For example, 1000000 * microsecond = second. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
sum of <time interval>	*Plain*	Returns the sum of the supplied time intervals. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
time interval <string>	*NamedGlobal*	Creates a time interval from the string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
week	*PlainGlobal*	Creates a time interval corresponding to 1 week. For example, 7*day = 1*week. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
absolute value of <time interval>	*Plain*	<time interval>	Returns positive value of the time interval. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
extrema of <time interval>	*Plain*	<(time interval, time interval)>	Returns the minimum and maximum extreme values of the given list of <time interval> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <time interval>	*Plain*	<time interval>	Returns the maximum interval from a list of time intervals. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
minimum of <time interval>	*Plain*	<time interval>	Returns the minimum interval from a list of time intervals. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
sum of <time interval>	*Plain*	<time interval>	Returns the sum of the supplied time intervals. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
unique value of <time interval>	*Plain*	<time interval with multiplicity>	Returns the unique values of a given list of <time interval> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
- <time interval>	*<time interval>*	The negative of a time interval. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<time interval> & <time>	*<time range>*	Concatenates a time interval with a time, returning a time range of the form time1 to time2. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<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, Lin, Sol, HPUX, AIX, Mac, WM
<time interval> {op} <time interval>	*<time interval>*	Returns a calculated time interval, where: • {op} is one of: +, -, mod, / . Win, Lin, Sol, HPUX, AIX, Mac, WM
<time interval> {op} <time zone>	*<time interval>*	Returns a calculated time interval, where: • {op} is one of: +, - . Win, Lin, Sol, HPUX, AIX, Mac, WM
<time interval> + <time of day with time zone>	*<time of day with time zone>*	Adds a time interval (days, hours, minutes, seconds) to a time of the day with time zone to create a new time of the day with time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time interval> + <time of day>	*<time of day>*	Adds a time interval (days, hours, minutes, seconds) to a time of the day to create a new time of the day. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time of day with time zone> {op} <time interval>	*<time of day with time zone>*	Adds or subtracts a time interval and a specified 'time of day with time zone' object, where {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<time of day> {op} <time interval>	*<time of day>*	Adds or subtracts a time interval to provide a new time of day. Here {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<time> & <time interval>	*<time range>*	Concatenates a time and a time interval, producing a time range object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Examples

- `sum of ((hour*4); (minute*3); (second*5))`

  Returns 04:03:05.

Time Interval with Multiplicity

These Inspectors deal with time-interval arrays, allowing you to extract unique time-interval values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <time interval>	*Plain*	Returns the unique values of a given list of <time interval> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <time interval with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <time interval> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Date

These are the various Inspectors that access the date types.

Creation Methods

Key Phrase	*Form*	Description
<string> as date	*Cast*	Casts a string as a date type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
april <integer> of <integer>	*Numbered*	Returns the nth day of april and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
august <integer> of <integer>	*Numbered*	Returns the nth day of August and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
current date	*PlainGlobal*	Returns the current date in the format: • Day of week, Day Month Year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
date <string>	*NamedGlobal*	Converts the given string into a date. The string should be of the form 'Day Month Year' and the returned date will be of the form 'Day of week, Day Month Year'. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
date <time zone> of <time>	*Index<time zone>*	Returns the date adjusted for the specified time zone. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
december <integer> of <integer>	*Numbered*	Returns the nth day of December and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
february <integer> of <integer>	*Numbered*	Returns the nth day of February and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
january <integer> of <integer>	*Numbered*	Returns the nth day of January and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
july <integer> of <integer>	*Numbered*	Returns the nth day of July and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
june <integer> of <integer>	*Numbered*	Returns the nth day of June and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
march <integer> of <integer>	*Numbered*	Returns the nth day of March and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <date>	*Plain*	Returns the maximum value from a list of <date> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
may <integer> of <integer>	*Numbered*	Returns the nth day of May and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
minimum of <date>	*Plain*	Returns the minimum value from a list of <date> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
november <integer> of <integer>	*Numbered*	Returns the nth day of November and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
october <integer> of <integer>	*Numbered*	Returns the nth day of October and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
september <integer> of <integer>	*Numbered*	Returns the nth day of September and the specified year as a date (day of week, month day year). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<date> as string	*Cast*	<string>	Cast a date type as a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_month of <date>	*Plain*	<day of month>	Extracts the day of the month from the specified date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_week of <date>	*Plain*	<day of week>	Extracts the day of the week (Monday, Tuesday, etcetera) from the specified date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_year of <date>	*Plain*	<day of year>	Extracts the day of year from the specified date, in the 'Month Day' format. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <date>	*Plain*	<(date, date)>	Returns the minimum and maximum extreme values of the given list of <date> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <date>	*Plain*	<date>	Returns the maximum value from a list of <date> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <date>	*Plain*	<date>	Returns the minimum value from a list of <date> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
month of <date>	*Plain*	<month>	Returns the month derived from the given date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
month_and_year of <date>	*Plain*	<month and year>	Formats the specified date in month year format, eg. March 2012. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
unique value of <date>	*Plain*	<date with multiplicity>	Returns the unique values and counts of a given list of <date> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
year of <date>	*Plain*	<year>	Returns the year, extracted from the given date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<date> - <date>	*<time interval>*	Subtracts two dates to produce a time interval. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<date> & <time of day with time zone>	*<time>*	Concatenates a date with a time and a time zone for a complete time stamp. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<date> {cmp} <date>	*<boolean>*	Compares two dates, where: • {cmp} is one of: =, <, <=. Win, Lin, Sol, HPUX, AIX, Mac, WM
<date> {op} <number of months>	*<date>*	Adds or subtracts a specified number of months to a given date, where {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<date> {op} <time interval>	*<date>*	Add or subtract a time interval to a date, producing a new date, where {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<format> + <date>	*<format>*	Takes a date and embeds it into the format specification. Win:8.0
<number of months> + <date>	*<date>*	Adds a number of months to a date, returning a new date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time interval> + <date>	*<date>*	Adds a time interval (days, hours, minutes, seconds) to a date to create a new date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time of day with time zone> & <date>	*<time>*	Concatenates a 'time of day with time zone' object with a date object to produce a time object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Date with Multiplicity

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

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <date>	*Plain*	Returns the unique values and counts of a given list of <date> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <date with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <date> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Day of Week

These Inspectors provide tools for dealing and calculating with day-of-week types, which include Monday, Tuesday, and more. This set of Inspectors includes each day as a self-named object.

Creation Methods

Key Phrase	*Form*	Description
<string> as day_of_week	*Cast*	Casts a string as a day of the week. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
current day_of_week	*PlainGlobal*	Returns the current day of the week, eg. Monday, Tuesday, etcetera. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_week <string>	*NamedGlobal*	Converts the given string value to a day of week type, eg. Monday, Tuesday, etcetera. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_week of <date>	*Plain*	Extracts the day of the week from the specified date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
friday	*PlainGlobal*	Returns Friday as a day of week object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
monday	*PlainGlobal*	Returns the day of week object for Monday. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
saturday	*PlainGlobal*	Returns Saturday as a day of week object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
sunday	*PlainGlobal*	Returns Sunday as a day of week object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
thursday	*PlainGlobal*	Returns Thursday as a day of week object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
tuesday	*PlainGlobal*	Returns Tuesday as a day of week object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
wednesday	*PlainGlobal*	Returns Wednesday as a day of week object. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<day of week> as string	*Cast*	<string>	Casts the day of week as a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of week> as three letters	*Cast*	<string>	Casts the day of week as a three-letter abbreviation (Mon, Tue, etcetera.). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
unique value of <day of week>	*Plain*	<day of week with multiplicity>	Returns the unique values of a given list of <day of week> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<day of week> - <day of week>	*<time interval>*	Subtract two day of week types (Monday, Tuesday, etcetera.) to produce a time interval. The answer cannot exceed 6 days. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key phrase	Return Type	Description
<day of week> {op} <time interval>	*<day of week>*	Add or subtract a time interval from a day of the week to produce a new day of week. Here {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<day of week> = <day of week>	*<boolean>*	Compares two days of the week and returns a boolean TRUE or FALSE. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<format> + <day of week>	*<format>*	Takes a 'day of week' and embeds it into the format specification. Win:8.0
<time interval> + <day of week>	*<day of week>*	Adds a time interval (days, hours, minutes, seconds) to a day of the week to create a new day of the week. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Day of Week with Multiplicity

These Inspectors deal with day-of-week arrays, allowing you to extract unique day-of-week values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <day of week>	*Plain*	Returns the unique values of a given list of <day of week> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <day of week with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <day of week> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Day of Month

These Inspectors provide tools for dealing and calculating with day-of-month types, which are numbers from 1-31.

Creation Methods

Key Phrase	*Form*	Description
<integer> as day_of_month	*Cast*	Cast an integer as a day of the month type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as day_of_month	*Cast*	Casts a string as a day of month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
current day_of_month	*PlainGlobal*	Returns the current day of the month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day of <day of year>	*Plain*	Returns the day of the month of the specified date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_month <integer>	*NumberedGlobal*	Converts the given integer to a day of month type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_month <string>	*NamedGlobal*	Converts the given string value (must be an integer from 1-31) to a day of month type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_month of <date>	*Plain*	Extracts the day of the month from the specified date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <day of month>	*Plain*	Returns the maximum value from a list of <day of month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <day of month>	*Plain*	Returns the minimum value from a list of <day of month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<day of month> as integer	*Cast*	<integer>	Cast a day of month type as an integer. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of month> as string	*Cast*	<string>	Cast a day of month type as a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
<day of month> as two digits	*Cast*	<string>	Cast a day of month type as a two-digit number. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <day of month>	*Plain*	<(day of month, day of month)>	Returns the minimum and maximum extreme values of the given list of <day of month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <day of month>	*Plain*	<day of month>	Returns the maximum value from a list of <day of month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <day of month>	*Plain*	<day of month>	Returns the minimum value from a list of <day of month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
unique value of <day of month>	*Plain*	<day of month with multiplicity>	Returns the unique values of a given list of <day of month> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<day of month> - <day of month>	*<time interval>*	Subtract two day of month types, producing a time interval. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of month> & <month and year>	*<date>*	Concatenate a day of month with a month and year type to produce a complete date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of month> & <month>	*<day of year>*	Concatenate a day of month with a month type to produce a day of year (eg. April 20). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of month> {cmp} <day of month>	*<boolean>*	Compare two day of month types, where {cmp} is one of: <, <=, =. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<day of month> {op} <time interval>	*<day of month>*	Add or subtract a time interval from a day of month to produce a new day of month. Here {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<month and year> & <day of month>	*<date>*	Concatenates a month and year with a day of month to produce a complete date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key phrase	Return Type	Description
<month> & <day of month>	*<day of year>*	Concatenates a month and a day of the month to produce a day of year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time interval> + <day of month>	*<day of month>*	Adds a time interval (days, hours, minutes, seconds) to a day of the month to create a new day of the month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Day of Month with Multiplicity

These Inspectors deal with day-of-month arrays, allowing you to extract unique day-of-month values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <day of month>	*Plain*	Returns the unique values of a given list of <day of month> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <day of month with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <day of month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Day of Year

These Inspectors provide tools for dealing and calculating with day-of-month types, which are of the form Sun, 01 Apr 2007.

Creation Methods

Key Phrase	*Form*	Description
april <integer>	*NumberedGlobal*	Returns the nth day of april as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
august <integer>	*NumberedGlobal*	Returns the nth day of August as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
current day_of_year	*PlainGlobal*	Returns the current day of the year, in a Month Day format. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day_of_year of <date>	*Plain*	Extracts the day of year from the specified date, in the 'Month Day' format. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
december <integer>	*NumberedGlobal*	Returns the nth day of December as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
february <integer>	*NumberedGlobal*	Returns the nth day of February as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
january <integer>	*NumberedGlobal*	Returns the nth day of January as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
july <integer>	*NumberedGlobal*	Returns the nth day of July as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
june <integer>	*NumberedGlobal*	Returns the nth day of June as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
march <integer>	*NumberedGlobal*	Returns the nth day of March as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <day of year>	*Plain*	Returns the maximum value from a list of <day of year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
may <integer>	*NumberedGlobal*	Returns the nth day of May as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
minimum of <day of year>	*Plain*	Returns the minimum value from a list of <day of year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
november <integer>	*NumberedGlobal*	Returns the nth day of November as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
october <integer>	*NumberedGlobal*	Returns the nth day of October as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
september <integer>	*NumberedGlobal*	Returns the nth day of September as a 'day of year' type (month day). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<day of year> as string	*Cast*	<string>	Casts a day of the year as a string type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
day of <day of year>	*Plain*	<day of month>	Returns the day of the month of the specified date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <day of year>	*Plain*	<(day of year, day of year)>	Returns the minimum and maximum extreme values of the given list of <day of year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <day of year>	*Plain*	<day of year>	Returns the maximum value from a list of <day of year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <day of year>	*Plain*	<day of year>	Returns the minimum value from a list of <day of year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
month of <day of year>	*Plain*	<month>	Returns the month portion of the given date (in month day format). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
unique value of <day of year>	*Plain*	<day of year with multiplicity>	Returns the unique values of a given list of <day of year> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<day of year> - <day of year>	*<time interval>*	Subtracts two days of the year to produce a time interval. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of year> & <month and year>	*<date>*	Concatenates a day of the year with a month and year to create a complete date type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of year> & <year>	*<date>*	Concatenates a day of the year with a year to create a complete date type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of year> {cmp} <day of year>	*<boolean>*	Compares two days of the year, producing a boolean TRUE or FALSE, where {cmp} is one of: <, <=, =. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<day of year> {op} <number of months>	*<day of year>*	Add or subtract a number of months to a day of the year to produce a new day of the year. Here {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<day of year> {op} <time interval>	*<day of year>*	Add or subtract a time interval to a day of the year to produce a new day of the year. Here {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<month and year> & <day of year>	*<date>*	Concatenates a month and year with a day of year to produce a complete date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<number of months> + <day of year>	*<day of year>*	Adds a number of months to a day of the year (July 4, say) to produce another day of the year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<time interval> + <day of year>	*<day of year>*	Adds a time interval (days, hours, minutes, seconds) to a day of the year to create a new day of the year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<year> & <day of year>	*<date>*	Concatenates a year with the day of the year, returning a full date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Day of Year with Multiplicity

These Inspectors deal with day-of-year arrays, allowing you to extract unique day-of-year values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <day of year>	*Plain*	Returns the unique values of a given list of <day of year> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <day of year with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <day of year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Month

These Inspectors provide tools for dealing and calculating with month types, which are of the form January, February, and more. This set of Inspectors includes each month as a self-named object.

Creation Methods

Key Phrase	*Form*	Description
<integer> as month	*Cast*	Returns the name of the nth month of the year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<string> as month	*Cast*	Converts a string into a month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
april	*PlainGlobal*	Returns april as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
august	*PlainGlobal*	Returns August as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
current month	*PlainGlobal*	Returns the current month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
december	*PlainGlobal*	Returns December as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
february	*PlainGlobal*	Returns February as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
january	*PlainGlobal*	Returns January as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
july	*PlainGlobal*	Returns July as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
june	*PlainGlobal*	Returns June as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
march	*PlainGlobal*	Returns March as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <month>	*Plain*	Returns the maximum value from a list of <month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
may	*PlainGlobal*	Returns May as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
minimum of <month>	*Plain*	Returns the minimum value from a list of <month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
month <integer>	*NumberedGlobal*	Returns the month type corresponding to the given <integer>. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
month <string>	*NamedGlobal*	Returns a month type corresponding to the given <string>. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
month of <date>	*Plain*	Returns the month of the given date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
month of <day of year>	*Plain*	Returns the month portion of the given date (in month day format). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
month of <month and year>	*Plain*	Returns the month portion of the given date (in month year format). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
november	*PlainGlobal*	Returns November as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
october	*PlainGlobal*	Returns October as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
september	*PlainGlobal*	Returns September as an object of type month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<month> as integer	*Cast*	<integer>	Converts the given month into an integer (1-12). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month> as string	*Cast*	<string>	Converts the given month into a string value. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month> as three letters	*Cast*	<string>	Converts the given month into a 3-letter string (Jan, Feb, etcetera.). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month> as two digits	*Cast*	<string>	Converts the month into a two digit number (01 - 12). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <month>	*Plain*	<(month, month)>	Returns the minimum and maximum extreme values of the given list of <month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <month>	*Plain*	<month>	Returns the maximum value from a list of <month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <month>	*Plain*	<month>	Returns the minimum value from a list of <month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
unique value of <month>	*Plain*	<month with multiplicity>	Returns the unique values of a given list of <month> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<day of month> & <month>	*<day of year>*	Concatenate a day of month with a month type to produce a day of year (eg. April 20). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month> - <month>	*<number of months>*	Subtracts two months, returning a positive number of months. If the first month is earlier than the second, it assumes the year has rolled over. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month> & <day of month>	*<day of year>*	Concatenates a month and a day of the month to produce a day of year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month> & <year>	*<month and year>*	Returns a date (in month year format) from the concatenation of a month and a year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month> {cmp} <month>	*<boolean>*	Compares the values of two months, where {cmp} is one of: <, <=, =. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<month> {op} <number of months>	*<month>*	Adds or subtracts a number of months from the given month. Here {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<number of months> + <month>	*<month>*	Adds a number of months to the given month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<year> & <month>	*<month and year>*	Returns a date (in month year format) from the concatenation of a month and a year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Month with Multiplicity

These Inspectors deal with month arrays, allowing you to extract unique month values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <month>	*Plain*	Returns the unique values of a given list of <month> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <month with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Month and Year

These Inspectors provide tools for dealing and calculating with month-and-year types, which are of the form month of year, eg., January of 2007.

Creation Methods

Key Phrase	*Form*	Description
april of <integer>	*Plain*	Creates a date (in month year format) corresponding to april of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
august of <integer>	*Plain*	Creates a date (in month year format) corresponding to August of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
current month_and_year	*PlainGlobal*	Returns the current date in month year format, eg. January 2012. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
december of <integer>	*Plain*	Creates a date (in month year format) corresponding to December of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
february of <integer>	*Plain*	Creates a date (in month year format) corresponding to February of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
january of <integer>	*Plain*	Creates a date (in month year format) corresponding to January of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
july of <integer>	*Plain*	Creates a date (in month year format) corresponding to July of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
june of <integer>	*Plain*	Creates a date (in month year format) corresponding to June of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
march of <integer>	*Plain*	Creates a date (in month year format) corresponding to March of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <month and year>	*Plain*	Returns the maximum value from a list of <month and year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
may of <integer>	*Plain*	Creates a date (in month year format) corresponding to May of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
minimum of <month and year>	*Plain*	Returns the minimum value from a list of <month and year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
month_and_year of <date>	*Plain*	Formats the specified date in month year format, eg. March 2012. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
november of <integer>	*Plain*	Creates a date (in month year format) corresponding to November of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
october of <integer>	*Plain*	Creates a date (in month year format) corresponding to October of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
september of <integer>	*Plain*	Creates a date (in month year format) corresponding to September of the specified year (as an <integer>). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<month and year> as string	*Cast*	<string>	Casts a date (in month year format) as a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <month and year>	*Plain*	<(month and year, month and year)>	Returns the minimum and maximum extreme values of the given list of <month and year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
first <day of week> of <month and year>	*Index<day of week>*	<date>	Finds the specific date corresponding to the first day of the week (eg. Friday) for a given month and year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
first friday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Friday of any given month and year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
first monday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Monday of any given month and year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
first saturday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Saturday of any given month and year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
first sunday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Sunday of any given month and year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
first thursday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Thursday of any given month and year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
first tuesday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Tuesday of any given month and year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
first wednesday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Wednesday of any given month and year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
length of <month and year>	*Plain*	<time interval>	Returns the number of days in the specified month. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <month and year>	*Plain*	<month and year>	Returns the maximum value from a list of <month and year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <month and year>	*Plain*	<month and year>	Returns the minimum value from a list of <month and year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
month of <month and year>	*Plain*	<month>	Returns the name of the month corresponding to the given date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
unique value of <month and year>	*Plain*	<month and year with multiplicity>	Returns the unique values of a given list of <month and year> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
year of <month and year>	*Plain*	<year>	Returns the year portion of the specified date (in month year format). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<day of month> & <month and year>	*<date>*	Concatenate a day of month with a month and year type to produce a complete date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<day of year> & <month and year>	*<date>*	Concatenates a day of the year with a month and year to create a complete date type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month and year> - <month and year>	*<number of months>*	Subtracts two dates (in month year format), returning a number of months. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month and year> {cmp} <month and year>	*<boolean>*	Compares two dates (in month year format), where {cmp} is one of: <, <=, =. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<month and year> {op} <number of months>	*<month and year>*	Adds or subtracts a number of months from a given date (in month year format), where {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<number of months> + <month and year>	*<month and year>*	Adds a number of months to a given date (in month year format) producing a new date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Month and Year with Multiplicity

These Inspectors deal with month-and-year arrays, allowing you to extract unique month-and-year values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <month and year>	*Plain*	Returns the unique values of a given list of <month and year> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <month and year with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <month and year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Number of Months

These Inspectors provide tools for dealing and calculating with number-of-month types, which are similar to integers, but wih yearly roll-over.

Creation Methods

Key Phrase	*Form*	Description
maximum of <number of months>	*Plain*	Returns the maximum value from a list of <number of months> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <number of months>	*Plain*	Returns the minimum value from a list of <number of months> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
month	*PlainGlobal*	Returns the specified number of months. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
year	*PlainGlobal*	Returns the specified number of years as a <number of months> type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<number of months> as string	*Cast*	<string>	Converts a number of months type into a string type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <number of months>	*Plain*	<(number of months, number of months)>	Returns the minimum and maximum extreme values of the given list of <number of months> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <number of months>	*Plain*	<number of months>	Returns the maximum value from a list of <number of months> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <number of months>	*Plain*	<number of months>	Returns the minimum value from a list of <number of months> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
unique value of <number of months>	*Plain*	<number of months with multiplicity>	Returns the unique values of a given list of <number of month> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
- <number of months>	*<number of months>*	Creates the negative of the specified number of months. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<number of months> {cmp} <number of months>	*<boolean>*	Compare two numbers of months, where {cmp} is one of: <, <=, =. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<number of months> {op} <number of months>	*<number of months>*	Multiplies or divides a number of months by an integer. Here {op} is one of: *, /. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM
<number of months> + <year>	*<year>*	Returns the year after adding the specified number of months. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key phrase	Return Type	Description
<year> {op} <number of months>	*<year>*	Adds or subtracts the specified number of months to derive a new year. Here {op} is one of: -, +. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM

Number of Months with Multiplicity

These Inspectors deal with number-of-month arrays, allowing you to extract unique number-of-month values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <number of months>	*Plain*	Returns the unique values of a given list of <number of month> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <number of months with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <number of month> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Year

These Inspectors provide tools for dealing and calculating with year types, which are of the form YYYY, as in 2008.

Creation Methods

Key Phrase	*Form*	Description
<integer> as year	*Cast*	Casts an integer as a year type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
<string> as year	*Cast*	Converts a string into a year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
current year	*PlainGlobal*	Returns the current year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
maximum of <year>	*Plain*	Returns the maximum value from a list of <year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <year>	*Plain*	Returns the minimum value from a list of <year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
year <integer>	*NumberedGlobal*	Creates a year object from the specified integer. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
year <string>	*NamedGlobal*	Creates a year object from the specified string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
year of <date>	*Plain*	The year derived from the given date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
year of <month and year>	*Plain*	Returns the year portion of the specified date (in month year format). Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<year> as integer	*Cast*	<integer>	Casts a year as an integer. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<year> as string	*Cast*	<string>	Casts a year as a string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
extrema of <year>	*Plain*	<(year, year)>	Returns the minimum and maximum extreme values of the given list of <year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
leap of <year>	*Plain*	<boolean>	Returns a flag indicating whether or not the specified year is a leap year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
length of <year>	*Plain*	<time interval>	Returns the number of day in the specified year. Leap years have 366 days. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
maximum of <year>	*Plain*	<year>	Returns the maximum value from a list of <year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <year>	*Plain*	<year>	Returns the minimum value from a list of <year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
unique value of <year>	*Plain*	<year with multiplicity>	Returns the unique values of a given list of <year> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<day of year> & <year>	*<date>*	Concatenates a day of the year with a year to create a complete date type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<month> & <year>	*<month and year>*	Returns a date (in month year format) from the concatenation of a month and a year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<year> - <year>	*<number of months>*	Subtracts two years and produces a time interval marked in months and years. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<year> & <day of year>	*<date>*	Concatenates a year with the day of the year, returning a full date. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<year> & <month>	*<month and year>*	Returns a date (in month year format) from the concatenation of a month and a year. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM, Ubu
<year> {cmp} <year>	*<boolean>*	Compares two years, where {cmp} is one of: <, <=, =. Win, Lin, Sol, HPUX, AIX, Mac:7.1, WM

Year with Multiplicity

These Inspectors deal with year arrays, allowing you to extract unique year values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <year>	*Plain*	Returns the unique values of a given list of <year> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <year with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <year> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

World Objects

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

World

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

Properties

Key Phrase	*Form*	Return Type	Description
boolean <string>	*NamedGlobal*	<boolean>	Returns a boolean. For example, boolean "TRUE". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
character <integer>	*NumberedGlobal*	<string>	Returns a string containing a single ASCII character. For example, character 90 = "Z". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
false	*PlainGlobal*	<boolean>	Returns the boolean FALSE. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
greatest integer	*PlainGlobal*	<integer>	Returns the largest integer that can be represented on this machine. See integer. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
hexadecimal integer <string>	*NamedGlobal*	<integer>	Creates an integer from the provided hexadecimal value. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
hexadecimal string <string>	*NamedGlobal*	<string>	Creates a string from the given hexadecimal value. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
integer <integer>	*NumberedGlobal*	<integer>	Returns an integer. The keyword is optional. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
integer <string>	*NamedGlobal*	<integer>	Returns integer for name provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
integer in <(integer, integer)>	*Index<(integer, integer)>Global*	<integer>	Returns a list of the integers contained between the endpoints (inclusive). The order is preserved, from least to greatest or vice-versa. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
integer in <(integer, integer, integer)>	*Index<(integer, integer, integer)>Global*	<integer>	Returns a list of the integers contained between the endpoints (inclusive) specified by the first two integers, with a step size specified by the third integer. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
integer to <integer>	*NumberedGlobal*	<integer>	Returns a list of integers from zero up to the specified integer. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
least integer	*PlainGlobal*	<integer>	Returns the least integer value that can be represented on this machine. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
local time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
now	*PlainGlobal*	<time>	Returns the current time as a time object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
string <string>	*NamedGlobal*	<string>	Returns a string for the name provided. The keyword string is optional. For example, string "hi" = "hi". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Return Type	Description
true	*PlainGlobal*	<boolean>	Returns the boolean TRUE. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
universal time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
windows display time <string>	*NamedGlobal*	<time>	Returns a string that may match the time shown in the Windows file system. Win, WM

Examples

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

- `integers in (2,-1)`
- Returns 2, 1, 0, -1.

- `integers in (1,6,2)`
- Returns 1,3,5.

- `integers to 2`
- Returns 0,1,2.

Filesystem Objects

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

Version

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

Creation Methods

Key Phrase	*Form*	Description
<string> as version	*Cast*	Turns a string into a version object. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
<version> as version	*Cast*	Reflexive cast of version. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, WM, Ubu
maximum of <version>	*Plain*	Returns the maximum value from a list of <version> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <version>	*Plain*	Returns the minimum value from a list of <version> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
pad of <version>	*Plain*	Creates a version object which is padded with zero values. Win, WM
version <string>	*NamedGlobal*	Creates a version object corresponding to the name provided. Syntax: version "1.2". Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
<version> as version	*Cast*	<version>	Reflexive cast of version. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, WM, Ubu
extrema of <version>	*Plain*	<(version, version)>	Returns the minimum and maximum extreme values of the given list of <version> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <version>	*Plain*	<version>	Returns the maximum value from a list of <version> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <version>	*Plain*	<version>	Returns the minimum value from a list of <version> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
pad of <version>	*Plain*	<version>	Returns a version object which is padded with zero values. Win, WM

Key Phrase	*Form*	Return Type	Description
unique value of <version>	*Plain*	<version with multiplicity>	Returns the unique values of a given list of <version> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM
<version> {cmp} <version>	<*boolean*>	Returns a boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win, Lin, Sol, HPUX, AIX, Mac, WM

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

Examples

- `pad of version "1.2" = version "1.2.0.0"`
- Returns TRUE.

- `extrema of (version "1.1"; version "2.3"; version "0.9")`
- Returns the minimum and maximum values of the set: 0.9, 2.3.

Version with Multiplicity

These Inspectors deal with version arrays, allowing you to extract unique version values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <version>	*Plain*	Returns the unique values of a given list of <version> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <version with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <version> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Xml Dom Document

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

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

Creation Methods

Key Phrase	*Form*	Description
owner document of <xml dom node>	*Plain*	Returns the name of the document that contains the specified node. Win, WM
xml document of <string>	*Plain*	Returns an XML document object from the given <string>, typically a file name. Win, WM

Xml Dom Node

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

Creation Methods

Key Phrase	*Form*	Description
attribute <integer> of <xml dom node>	*Numbered*	Returns the numbered attribute of the specified XML DOM node. Win, WM
attribute <string> of <xml dom node>	*Named*	Returns the named attribute of the specified node. Win, WM
attribute of <xml dom node>	*Plain*	Returns the attribute(s) of the specified XML DOM node. Win, WM
child node <integer> of <xml dom node>	*Numbered*	Returns the child node by number. Win, WM
child node of <xml dom node>	*Plain*	By chaining this Inspector, you can find the child nodes of any given node. Win, WM
first child of <xml dom node>	*Plain*	Returns the first child node in the specified node. When applied to an XML DOM file, it returns the first node in the file. Win, WM
last child of <xml dom node>	*Plain*	Returns the last child node in the specified node. When applied to an XML DOM file, it returns the last node in the file. Win, WM
next sibling of <xml dom node>	*Plain*	Returns the next child node after the current one. Win, WM
parent node of <xml dom node>	*Plain*	Returns the parent node of the specified node. The top of the hierarchy is the document itself, so a phrase such as "exists parent node of xml dom document" will return FALSE. Win, WM
previous sibling of <xml dom node>	*Plain*	Returns the child node before the one specified. Win, WM

Key Phrase	*Form*	Description
select <string> of <xml dom node>	*Named*	Uses an Xpath string to specify an XML DOM node. For instance, to select all elements BBB which are children of the root element AAA, use: • selects "/AAA/BBB" of xml dom document <string>. Win, WM
xpath <(string, string)> of <xml dom node>	*Index<(string, string)>*	The iterated named property xpaths (<namespace>, <query>) provides a way of specifying the namespaces for the query. If the XML document you are querying over uses namespaces, you must use them in the query and use this property. Win, WM
xpath <string> of <xml dom node>	*Named*	Returns an iterated list of matching xml dom nodes, given the xpath query specified by <string>. Win, WM

Properties

Key Phrase	*Form*	Return Type	Description
<xml dom node> as text	*Cast*	<string>	Casts an xml document object module node as text. Win, WM
<xml dom node> as xml	*Cast*	<string>	Casts an xml document object module node as xml. Win, WM
attribute <integer> of <xml dom node>	*Numbered*	<xml dom node>	Returns the numbered attribute of the specified XML DOM node. Win, WM
attribute <string> of <xml dom node>	*Named*	<xml dom node>	Returns the named attribute of the specified node. Win, WM
attribute of <xml dom node>	*Plain*	<xml dom node>	Returns the attribute(s) of the specified XML DOM node. Win, WM
child node <integer> of <xml dom node>	*Numbered*	<xml dom node>	Returns the child node by number. Win, WM
child node of <xml dom node>	*Plain*	<xml dom node>	By chaining this Inspector, you can find the child nodes of any given node. Win, WM

Key Phrase	*Form*	Return Type	Description
first child of <xml dom node>	*Plain*	<xml dom node>	Returns the first child node in the specified node. When applied to an XML DOM file, it returns the first node in the file. Win, WM
last child of <xml dom node>	*Plain*	<xml dom node>	Returns the last child node in the specified node. When applied to an XML DOM file, it returns the last node in the file. Win, WM
next sibling of <xml dom node>	*Plain*	<xml dom node>	Returns the next child node after the current one. Win, WM
node name of <xml dom node>	*Plain*	<string>	Returns the name of the specified XML DOM node as a string. Win, WM
node type of <xml dom node>	*Plain*	<integer>	Returns the numeric node type of the specified Document Object Module (DOM) node, 1-12 as shown in the creation Inspector. Win, WM
node value of <xml dom node>	*Plain*	<string>	Returns the node value, which varies depending on the node type. If the standard interface produces a null type, the Inspector throws NoSuchObject. Win, WM
owner document of <xml dom node>	*Plain*	<xml dom document>	Returns a document belonging to the owner of the specified node. Win, WM
parent node of <xml dom node>	*Plain*	<xml dom node>	Returns the parent node of the specified node. Win, WM
previous sibling of <xml dom node>	*Plain*	<xml dom node>	Returns the child node before the one specified. Win, WM
select <string> of <xml dom node>	*Named*	<xml dom node>	Uses an Xpath string to specify an XML DOM node. For instance, to select all elements BBB which are children of the root element AAA, use: • selects "/AAA/BBB" of xml dom document <string>. Win, WM

Key Phrase	*Form*	Return Type	Description
xpath <(string, string)> of <xml dom node>	*Index<(string, string)>*	<xml dom node>	The iterated named property xpaths (<namespace>, <query>) provides a way of specifying the namespaces for the query. If the XML document you are querying over uses namespaces, you must use them in the query and use this property. Win, WM
xpath <string> of <xml dom node>	*Named*	<xml dom node>	Returns an iterated list of matching xml dom nodes, given the xpath query specified by <string>. Win, WM

NOTE:

Some of the examples in this section refer to test.xml, a structured file like this:

```
<?xml version ="1.0"?>
  <a:Books xmlns:a="x-schema:bookschema.xml" >
    <a:Book>
      <title>Presenting XML</title>
      <author>Richard Light</author>
    </a:Book>
  </a:Books>
```

Site Objects

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

Site Version List

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

Creation Methods

Key Phrase	*Form*	Description
<string> as site version list	*Cast*	Converts a string into a site version list. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.0, Mac:7.1, WM, Ubu
maximum of <site version list>	*Plain*	Returns the maximum value from a list of <site version list> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Description
minimum of <site version list>	*Plain*	Returns the minimum value from a list of <site version list> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
site version list <string>	*NamedGlobal*	Returns a textual representation of a site version list ("manyversion"). Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.0, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<site version list> as string	*Cast*	<string>	Converts a site version list to a string. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.0, Mac:7.1, WM, Ubu
component <integer> of <site version list>	*Numbered*	<integer>	Returns the nth element (as specified by <integer>) of the given site version list. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.0, Mac:7.1, WM, Ubu
extrema of <site version list>	*Plain*	<(site version list, site version list)>	Returns the minimum and maximum extreme values of the given list of <site version list> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <site version list>	*Plain*	<site version list>	Returns the maximum value from a list of <site version list> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <site version list>	*Plain*	<site version list>	Returns the minimum value from a list of <site version list> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
unique value of <site version list>	*Plain*	<site version list with multiplicity>	Returns the unique values of a given list of <site version list> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<site version list> {cmp} <site version list>	*<boolean>*	Compares two site version lists, component by component, where {cmp} is one of: <, <=, =. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM

Site Version List with Multiplicity

These Inspectors deal with site-version-list arrays, allowing you to extract unique site-version-list values and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <site version list>	*Plain*	Returns the unique values of a given list of <site version list> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <site version list with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <site version list> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Formatting Objects

These Inspectors make it easy for you to format HTML statements.

Html

This type helps you to author HTML commands to create customized content for the BigFix Console and Web Reports. They allow construction of HTML snippets that can be used to display BigFix data elements in a browser.

Creation Methods

Key Phrase	*Form*	Description
<html> as html	*Cast*	Casts an html string into an html type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
<string> as html	*Cast*	Casts a string into html. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
abbr <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <abbr> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
abbr <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <abbr> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
abbr of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <abbr> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
abbr of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <abbr> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
acronym <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <acronym> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
acronym <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <acronym> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
acronym of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <acronym> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
acronym of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <acronym> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
address <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <address> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
address <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <address> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
address of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <address> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
address of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <address> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
anchor <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <a> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
anchor <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <a> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
anchor of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <a> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
anchor of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <a> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
b <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <b> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
b <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <b> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
b of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <b> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
b of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <b> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
base <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <base> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
base <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <base> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
base of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <base> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
base of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <base> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
big <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <big> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
big <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <big> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
big of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <big> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
big of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <big> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
blockquote <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <blockquote> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
blockquote <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <blockquote> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
blockquote of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <blockquote> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
blockquote of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <blockquote> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
body <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <body> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
body <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <body> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
body of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <body> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
body of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <body> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
br	*PlainGlobal*	Creates an HTML tag to output a line break. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
br <string>	*NamedGlobal*	Creates an HTML tag with an included modifier, such as class. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
caption <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <caption> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
caption <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <caption> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
caption of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <caption> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
caption of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <caption> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
cite <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <cite> (citation) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
cite <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <cite> (citation) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
cite of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <cite> (citation) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
cite of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <cite> (citation) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
code <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <code> (fixed-width font) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
code <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <code> (fixed-width font) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
code of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <code> (fixed-width font) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
code of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <code> (fixed-width font) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
col <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <col> (column) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
col <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <col> (column) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
col of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <col> (column) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
col of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <col> (column) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
colgroup <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <colgroup> (column group) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
colgroup <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <colgroup> (column group) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
colgroup of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <colgroup> (column group) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
colgroup of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <colgroup> (column group) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
concatenation <html> of <html>	*Index<html>*	This inspector concatenates the html items in the second argument with a separator defined by the first argument. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, Ubu
concatenation <html> of <string>	*Index<html>*	This inspector concatenates the string items in the second argument with a separator defined by the first argument. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, Ubu
concatenation <string> of <html>	*Named*	This inspector concatenates the html items in the second argument with a separator defined by the string argument. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
concatenation of <html>	*Plain*	Concatenates the items in the specified HTML string, end-to-end. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dd <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <dd> (definition) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dd <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <dd> (definition) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dd of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <dd> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dd of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <dd> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
definition list <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <dl> (definition) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
definition list <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <dl> (definition) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
definition list of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <dl> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
definition list of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <dl> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
del <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <del> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
del <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <del> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
del of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <del> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
del of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <del> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dfn <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <dfn> (definition) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dfn <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <dfn> (definition) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dfn of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <dfn> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
dfn of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <dfn> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
div <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a (division or section) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
div <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a (division or section) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
div of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a (division or section) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
div of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a (division or section) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dt <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <dt> (definition) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dt <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <dt> (definition) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dt of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <dt> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dt of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <dt> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
em <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an (emphasized) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
em <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an (emphasized) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
em of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an (emphasized) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
em of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an (emphasized) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h1 <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <h1> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h1 <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <h1> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h1 of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <h1> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h1 of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <h1> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h2 <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <h2> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h2 <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <h2> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h2 of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <h2> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h2 of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <h2> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h3 <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <h3> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
h3 <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <h3> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h3 of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <h3> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h3 of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <h3> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h4 <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <h4> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h4 <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <h4> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h4 of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <h4> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h4 of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <h4> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h5 <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <h5> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h5 <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <h5> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h5 of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <h5> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h5 of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <h5> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
h6 <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <h6> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h6 <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <h6> (header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h6 of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <h6> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h6 of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <h6> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
head <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <head> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
head <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <head> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
head of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <head> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
head of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <head> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
hr	*PlainGlobal*	Creates a horizontal line tag . Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
hr <string>	*NamedGlobal*	Creates a horizontal line tag with an option specfied by the string <hr string/>. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html <string>	*NamedGlobal*	Returns the html version of the specified string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <html> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
html <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <html> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html concatenation <string> of <html>	*Named*	Returns an html-formatted concatenation of the specified html segments, embedded in <html attr></html> tags, where the html attribute is specified by the <string>. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html concatenation of <html>	*Plain*	Returns an html-formatted concatenation of the specified html segments. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <html> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <html> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html tag <(string, html)>	*Index<(string, html)>Global*	Returns an HTML snippet containing the specified html enclosed in an html tag specified by string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html tag <(string, html attribute list)>	*Index<(string, html attribute list)>Global*	Returns an HTML snippet containing a HTML attribute list enclosed in an html tag specified by the string. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
html tag <(string, html attribute list, html)>	*Index<(string, html attribute list, html)>Global*	Returns an HTML snippet enclosed in a tag specified by the string, modified by the html attribute list and bracketing the html argument. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html tag <(string, html attribute list, string)>	*Index<(string, html attribute list, string)>Global*	Returns an HTML snippet enclosed in a tag specified by the first string, modified by the html attribute list and bracketing the second string argument. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html tag <(string, string)>	*Index<(string, string)>Global*	Returns an HTML snippet containing the second string enclosed in an html tag specified by the first string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html tag <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a tag specified by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
html tag <string> of <string>	*Named*	Returns an HTML snippet with the second string enclosed in a tag specified by the first string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ins <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <ins> (insert) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ins <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <ins> (insert) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ins of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <ins> (insert) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ins of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <ins> (insert) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
italic <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <i> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
italic <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <i> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
italic of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <i> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
italic of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <i> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
kbd <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <kbd> (keyboard entry) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
kbd <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <kbd> (keyboard entry) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
kbd of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <kbd> (keyboard entry) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
kbd of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <kbd> (keyboard entry) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
li <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a (list) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
li <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a (list) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
li of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a (list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
li of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a (list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
link <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <link> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
link <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <link> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
link of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <link> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
link of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <link> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
meta <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <meta> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
meta <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <meta> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
meta of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <meta> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
meta of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <meta> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ol <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an (ordered list) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ol <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an (ordered list) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ol of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an (ordered list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ol of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an (ordered list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ordered list <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ordered list <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an tag, where the tag is modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ordered list of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ordered list of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
p <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <p> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
p <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <p> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
p of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <p> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
p of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <p> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
pre <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in an <pre> (preformatted) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
pre <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in an <pre> (preformatted) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
pre of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in an <pre> (preformatted) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
pre of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in an <pre> (preformatted) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
q <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <q> (quotation) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
q <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <q> (quotation) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
q of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <q> (quotation) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
q of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <q> (quotation) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
samp <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <samp> (sample) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
samp <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <samp> (sample) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
samp of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <samp> (sample) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
samp of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <samp> (sample) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
small <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <small> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
small <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <small> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
small of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <small> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
small of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <small> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
span <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
span <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
span of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
span of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
strong <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
strong <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
strong of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
strong of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sub <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <sub> (subscript) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sub <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <sub> (subscript) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sub of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <sub> (subscript) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sub of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <sub> (subscript) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sup <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <sup> (superscript) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
sup <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <sup> (superscript) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sup of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <sup> (superscript) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sup of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <sup> (superscript) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
table <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <table> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
table <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <table> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
table of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <table> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
table of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <table> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tbody <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <tbody> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tbody <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <tbody> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tbody of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <tbody> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tbody of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <tbody> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
td <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <td> (table cell) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
td <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <td> (table cell) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
td of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <td> (table cell) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
td of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <td> (table cell) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tfoot <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <tfoot> (table foot) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tfoot <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <tfoot> (table foot) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tfoot of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <tfoot> (table foot) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tfoot of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <tfoot> (table foot) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
th <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <th> (table header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
th <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <th> (table header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
th of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <th> (table header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
th of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <th> (table header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
thead <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <thead> (table header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
thead <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <thead> (table header) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
thead of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <thead> (table header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
thead of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <thead> (table header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
title <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <title> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
title <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <title> tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
title of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <title> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
title of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <title> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tr <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <tr> (table row) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tr <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <tr> (table row) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
tr of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <tr> (table row) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tr of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <tr> (table row) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tt <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <tt> (teletype font) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tt <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <tt> (teletype font) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tt of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <tt> (teletype font) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tt of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <tt> (teletype font) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ul <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a (unordered list) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ul <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a (unordered list) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ul of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a (unordered list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ul of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a (unordered list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
unordered list <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a (unordered list item) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Description
unordered list <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a (unordered list item) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
unordered list of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a (unordered list item) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
unordered list of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a (unordered list item) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
var <string> of <html>	*Named*	Returns an HTML snippet with the specified html enclosed in a <var> (variable type) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
var <string> of <string>	*Named*	Returns an HTML snippet with the specified string enclosed in a <var> (variable type) tag modified by the first given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
var of <html>	*Plain*	Returns an HTML snippet containing the specified html enclosed in a <var> (variable type) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
var of <string>	*Plain*	Returns an HTML snippet containing the specified string enclosed in a <var> (variable type) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<html> as html	*Cast*	<html>	Casts an html string into an html type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
<html> as string	*Cast*	<string>	Casts an html string into a string type. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
abbr <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <abbr> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
abbr of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <abbr> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
acronym <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <acronym> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
acronym of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <acronym> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
address <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <address> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
address of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <address> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
anchor <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <a> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
anchor of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <a> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
b <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <b> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
b of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <b> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
base <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <base> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
base of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <base> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
big <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <big> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
big of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <big> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
blockquote <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <blockquote> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
blockquote of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <blockquote> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
body <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <body> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
body of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <body> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
caption <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <caption> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
caption of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <caption> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
cite <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <cite> (citation) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
cite of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <cite> (citation) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
code <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <code> (fixed-width font) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
code of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <code> (fixed-width font) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
col <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <col> (column) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
col of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <col> (column) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
colgroup <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <colgroup> (column group) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
colgroup of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <colgroup> (column group) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
concatenation <html> of <html>	*Index<html>*	<html>	This inspector concatenates the html items in the second argument with a separator defined by the first argument. Win:7.0, Lin:7.0, Sol:7.0, HPUX:7.0, AIX:7.0, Mac:7.1, Ubu
concatenation <string> of <html>	*Named*	<html>	This inspector concatenates the html items in the second argument with a separator defined by the string argument. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
concatenation of <html>	*Plain*	<html>	Concatenates the items in the specified HTML string, end-to-end. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dd <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <dd> (definition) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dd of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <dd> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
definition list <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <dl> (definition) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
definition list of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <dl> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
del <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <del> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
del of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <del> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dfn <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <dfn> (definition) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dfn of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <dfn> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
div <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a (division or section) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
div of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a (division or section) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dt <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <dt> (definition) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
dt of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <dt> (definition) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
em <string> of <html>	*Named*	<html>	Emphasize the specified HTML code inside an <em string> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
em of <html>	*Plain*	<html>	Emphasize the specified HTML code inside an tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h1 <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <h1> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
h1 of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <h1> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h2 <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <h2> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h2 of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <h2> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h3 <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <h3> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h3 of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <h3> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h4 <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <h4> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h4 of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <h4> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h5 <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <h5> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h5 of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <h5> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h6 <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <h6> (header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
h6 of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <h6> (header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
head <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <head> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
head of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <head> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <html> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html concatenation <string> of <html>	*Named*	<html>	Returns an html-formatted concatenation of the specified html segments, embedded in <html attr></html> tags, where the html attribute is specified by the <string>. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html concatenation of <html>	*Plain*	<html>	Returns an html-formatted concatenation of the specified html segments. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <html> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
html tag <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a tag specified by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ins <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <ins> (insert) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ins of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <ins> (insert) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
italic <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <i> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
italic of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <i> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
kbd <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <kbd> (keyboard entry) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
kbd of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <kbd> (keyboard entry) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
li <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a (list) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
li of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a (list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
link <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <link> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
link of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <link> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
meta <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <meta> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
meta of <html>	*Plain*	<html>	Encloses the specified html in a <meta> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ol <string> of <html>	*Named*	<html>	Creates an ordered list out of the <html> with an optional style specified by <string>. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ol of <html>	*Plain*	<html>	Creates an ordered list out of the <html> content. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ordered list <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
ordered list of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
p <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <p> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
p of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <p> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
pre <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in an <pre> (preformatted) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
pre of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in an <pre> (preformatted) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
q <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <q> (quotation) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
q of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <q> (quotation) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
samp <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <samp> (sample) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
samp of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <samp> (sample) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
small <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <small> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
small of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <small> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
span <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
span of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
strong <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
strong of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sub <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <sub> (subscript) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sub of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <sub> (subscript) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sup <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <sup> (superscript) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
sup of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <sup> (superscript) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
table <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <table> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
table of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <table> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tbody <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <tbody> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
tbody of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <tbody> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
td <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <td> (table cell) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
td of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <td> (table cell) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tfoot <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <tfoot> (table foot) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tfoot of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <tfoot> (table foot) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
th <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <th> (table header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
th of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <th> (table header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
thead <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <thead> (table header) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
thead of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <thead> (table header) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
title <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <title> tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
title of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <title> tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Key Phrase	*Form*	Return Type	Description
tr <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <tr> (table row) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tr of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <tr> (table row) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tt <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <tt> (teletype font) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
tt of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <tt> (teletype font) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ul <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a (unordered list) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
ul of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a (unordered list) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
unordered list <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a (unordered list item) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
unordered list of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a (unordered list item) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
var <string> of <html>	*Named*	<html>	Returns an HTML snippet with the specified html enclosed in a <var> (variable type) tag and attributes supplied by the given string. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
var of <html>	*Plain*	<html>	Returns an HTML snippet containing the specified html enclosed in a <var> (variable type) tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Operators

Key phrase	Return Type	Description
<html> & <html>	*<html>*	Concatenates two HTML files into a new one. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
<html> & <string>	*<html>*	Concatenates a string with an HTML file, returning a new HTML file. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu
<string> & <html>	*<html>*	Concatenates a string with an HTML file, returning a new HTML file. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Examples

- `(p of "hello") as html`
  - Returns hello .

- `"<p>hello</p>" as html`
  - Note that this will convert special characters, which may not be what you want. This example returns <p>hello</p>.

- `abbr "title='etcetera'" of "etc."`
  - Returns <abbr title='etcetera'>etc.</abbr>.

- `acronym "class='myclass'" of "TLA"`
  - Returns <acronym class='myclass'>TLA</acronym>.

- `address "class='myclass'" of "John Smith, Chicago, Illinois"`
  - Returns <address class='myclass'>John Smith, Chicago, Illinois</address>.

- `anchor "href='mylink'" of "click here"`
  - Returns click here.

- `head of (base "href='html://www.mybaseurl/'" of "")`
  - Returns <head><base='html://www.mybaseurl/'></base></head>

- .

- `big "class='myclass'" of "important"`
  - Returns <big class='myclass'>important</big>.

- `body "bgcolor='red'" of "my body text"`
  - Returns <body bgcolor='red'>my body text</body>.

- `br "class='clear'"`
  - Returns <br class='clear'/>.

- `code of "print 'hello world'"`
  - Returns <code>print 'hello world'</code>, which will display the contents in a fixed-width font such as courier.

- `concatenation ( html "<br>" ) of names of bes fixlets`
  - Returns a list of Fixlet names separated by line breaks: BES Clients in Seat Count Grace Mode
BES Clients Restricted by Seat Count….

- `concatenation "---" of links of bes fixlets`
  - Returns a list of Fixlet links separated by three dashes: <A href="linkid:openfixlet(1,1)">BES Clients in Seat Count Grace Mode</A>---<A href="linkid:openfixlet(1,2">BES Clients Restricted by Seat Count</A>….

- `concatenation of links of bes fixlets`
  - Returns a list of links derived from the BES Fixlets, such as <A href="linkid:openfixlet(1,1)">BES Clients in Seat Count Grace Mode</A><A href="linkid:openfixlet(1,2">BES Clients Restricted by Seat Count</A>….

- `dt of "salt" & dd of "NaCl"`
  - Returns <dt>salt</dt><dd>NaCl</dd>.

- `definition list of (dt of "hour" & dd of "60 minutes")`
  - Returns <dl><dt>hour</dt><dd>60 minutes</dd></dl>.

- `del of "old" & "new"`
  - Returns ~~old~~new, which will render "old" in a strikethrough font.

- `div "style='color:#AABBCC;'" of "hello"`
  - Returns hello .

- `html tag ("body", "hello")`
  - Returns <body>hello</body>.

- `html tag ( "p", attr list of ( ( "class", "myclass"); ( "align", "left" ) ), html "html <i>snippet</i>" )`
  - Returns html snippet .

- `html tag ( "p", attr list of ( ( "class", "myclass"); ( "align", "left" ) ), "formatted text" )`
  - Returns formatted text .

- `html tag ( "b", "bold text" )`
  - Returns bold text.

- `html tag "i" of html "italic <b>string</b>"`
  - Returns italic string.

- `html tag "body" of "hello"`
  - Returns <body>hello</body>.

- `"type" & kbd of " hello " & "into the box"`
  - Returns type<kbd> hello </kbd>into the box.

- `ordered list of "item 1"`
  - Returns item 1.

- `p "align='center'" of "hello"`
  - Returns hello .

- `unordered list of "item 1"`
  - item 1.

- `concatenation ( html "<div class=%22separator%22 />" ) of links of bes fixlets`
  - Returns a list of Fixlet links separated by an HTML statement: <A href="linkid:openfixlet(1,1)">BES Clients in Seat Count Grace Mode</A> <A href="linkid:openfixlet(1,2">BES Clients Restricted by Seat Count</A>...

- `html "lang='en-US'" of concatenation of (head of "my head"; body of "my content" )`
  - Returns: <html lang='en-US'><head>my head</head><body>my content</body></html>.

- `html concatenation of (head of "my head"; body of "my content" )`
-
  - Returns the html snippet: <head>my head</head><body>my content</body>.

- `html of (head of (title of "my title") &"my content")`
  - Returns <html><head><title>my title</title></head>my content</html>
- .

- `html tag "i" of html "italic <b>string</b>"`
  - Returns italic string.

- `ol "style=square" of (li "class=bold" of "first item" & li "class=bold" of "second item")`
  - Returns <ol style=square><li class=bold>first item
<li class=bold>second item.
- .

Html Attribute List

This type helps you to author HTML commands to create customized content for the BigFix Console and Web Reports. It is designed to create an HTML attribute=value snippet such as href="site.com".

Creation Methods

Key Phrase	*Form*	Description
attr list of <(string, string)>	*Plain*	Creates an HTML snippet from the supplied strings, where the first string is the tag and the second string is the value of the tag. Win, Lin, Sol, HPUX, AIX, Mac:7.1, Ubu

Examples

- `html tag ("a", attr lists of ("href", "mysite.com/image.gif"))`
- Returns <a href="mysite.com/image.gif"/>.

Format

Format inspectors make it easier for content authors to create localizable content. Similar in concept to the C printf function, they allow you to embed arguments into a formatting string, which is followed by the argument values themselves. The arguments are numbered and enclosed in curly brackets {0} and the values to be substituted are preceded by a plus sign (+). For example:

- Q: format "At {1}, user {0} is logged on." + name of current user + now
- A: At 9/27/2010 3:03:26 PM, user Scott is logged on.

Creation Methods

Key Phrase	*Form*	Description
format <string>	*NamedGlobal*	Returns the specified string into a format type which can be used to print formatted variables. Win:8.0

Properties

Key Phrase	*Form*	Return Type	Description
<format> as string	*Cast*	<string>	Returns the formatted output as a string type. Win:8.0

Operators

Key phrase	Return Type	Description
<format> + <format>	<*format*>	This Inspector allows you to embed a format inside another one. Win:8.0
<format> + <integer>	<*format*>	Takes an integer and embeds it into the format specification. Win:8.0
<format> + <string>	<*format*>	Takes a string and embeds it into the format specification. Win:8.0
<format> + <time interval>	<*format*>	Takes a time interval and embeds it into the format specification. Win:8.0
<format> + <time of day>	<*format*>	Takes a time of day and embeds it into the format specification. Win:8.0
<format> + <time>	<*format*>	Takes a time and embeds it into the format specification. Win:8.0

Examples

- `format "Logged in: {0}" + (format "user {0}" + name of current user)`
-
- Returns
- Logged in: user Scott.

- `format "Today's date is {0}." + now`
- Returns: Today's date is 9/27/2010 3:40:27 PM.

Authorization Objects

These inspectors retrieve security and access settings.

Cryptography

This is a global object that has several properties that expose the state of the cryptography controls. BigFix uses cryptographic functions throughout the BigFix Platform. Every time an operator logs in to BigFix, creates a new user, starts an action or subscribes to new content, authentication and signature routines are executed using cryptographic libraries based on the FIPS 140-2 standard.

Creation Methods

Key Phrase	*Form*	Description
cryptography	*PlainGlobal*	A global object that implements the FIPS 140-2 standard for secure signing and authentication throughout the BigFix application. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
desired fips mode of <cryptography>	*Plain*	<boolean>	Returns TRUE if the application (the client, console, or web reports, depending on the context) tried to enter FIPS 140-2 compliant mode. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, Ubu
fips mode failure message of <cryptography>	*Plain*	<string>	Returns the error message returned by the cryptographic library if the application (the client, console, or web reports, depending on the context) tried to enter FIPS 140-2 compliant mode and failed. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, Ubu
fips mode of <cryptography>	*Plain*	<boolean>	Returns TRUE if the application (the client, console, or web reports, depending on the context) is operating in FIPS 140-2 mode (the mode provided by openssl). FIPS mode limits the set of ciphers and SSL protocols that can be used in the cryptographic library. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, Ubu

X509 Certificate

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

Properties

Key Phrase	*Form*	Return Type	Description
invalid before of <x509 certificate>	*Plain*	<time>	Returns the date on which the certificate first becomes valid. This is useful for examining encryption certificates, where the 'invalid before date' is the time when the encryption credentials were generated. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
sha1 of <x509 certificate>	*Plain*	<string>	Returns the SHA1 hash of the given certificate, which uniquely identifies it. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Networking Objects

This chapter includes the various networking Inspectors.

Ip Version

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

Creation Methods

Key Phrase	*Form*	Description
any ip version	*PlainGlobal*	Returns a type corresponding to the Microsoft enumerated value NET_FW_IP_VERSION_ANY. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ip version <integer>	*NumberedGlobal*	Returns the the IP version for the <integer> port corresponding to the Microsoft enumerated types: • 0: NET_FW_IP_VERSION_V4 • 1: NET_FW_IP_VERSION_V6 • 2: NET_FW_IP_VERSION_ANY • 3: NET_FW_IP_VERSION_MAX. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	*Form*	Description
ip version of <ipv4or6 address>	*Plain*	Returns the IP version of the specified ipv4or6 address. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv4	*PlainGlobal*	Provides a comparison value for a firewall or other ip version inspector. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv6	*PlainGlobal*	Provides a comparison value for a firewall or other ip version inspector. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<ip version> as string	*Cast*	<string>	Converts an IP version to a string. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Operators

Key phrase	Return Type	Description
<ip version> = <ip version>	*<boolean>*	Compares two IP versions for equality. Win, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Ipv4 Address

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

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

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

Creation Methods

Key Phrase	*Form*	Description
ipv4 address <string>	*NamedGlobal*	Creates an object with an ip address for the string provided. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Key Phrase	*Form*	Description
ipv4 part of <ipv4or6 address>	*Plain*	Returns the lowest 32-bits of the IPv4or6 address as an IPv4 address. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv4 part of <ipv6 address>	*Plain*	Returns the lowest 32-bits of the IPv6 address as an IPv4 address. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <ipv4 address>	*Plain*	Returns the maximum value from a list of <ipv4 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <ipv4 address>	*Plain*	Returns the minimum value from a list of <ipv4 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<ipv4 address> as ipv4or6 address	*Cast*	<ipv4or6 address>	Converts an IPv4 address into the more inclusive ipv4or6 type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4 address> as ipv6 address	*Cast*	<ipv6 address>	Returns the specified IPv4 address embedded in a IPv6 address space as defined by RFC 4291 section 2.5.5.2: IPv4-Mapped IPv6 Address. In this scheme the lowest 32 bits of the IPv6 address contain the IPv4 address, the next higher 16 bits are all 1 (ffff) and the remaining bits are all 0. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
<ipv4 address> as string	*Cast*	<string>	Converts the ipv4 address to a string. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
extrema of <ipv4 address>	*Plain*	<(ipv4 address, ipv4 address)>	Returns the minimum and maximum extreme values of the given list of <ipv4 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <ipv4 address>	*Plain*	<ipv4 address>	Returns the maximum value from a list of <ipv4 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <ipv4 address>	*Plain*	<ipv4 address>	Returns the minimum value from a list of <ipv4 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
unique value of <ipv4 address>	*Plain*	<ipv4 address with multiplicity>	Returns the unique values of a given list of <ipv4 address> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Examples

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

Ipv4 Address with Multiplicity

These Inspectors deal with ipv4 address arrays, allowing you to extract unique ipv4 addresses and count them.

Type Derivation: This object type is derived from the <ipv4 address> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	*Form*	Description
unique value of <ipv4 address>	*Plain*	Returns the unique values of a given list of <ipv4 address> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <ipv4 address with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <ipv4 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Ipv6 Address

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

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

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

Creation Methods

Key Phrase	*Form*	Description
<ipv4 address> as ipv6 address	*Cast*	Returns the specified IPv4 address embedded in a IPv6 address space as defined by RFC 4291 section 2.5.5.2: IPv4-Mapped IPv6 Address. In this scheme the lowest 32 bits of the IPv6 address contain the IPv4 address, the next higher 16 bits are all 1 (ffff) and the remaining bits are all 0. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
<string> as ipv6 address	*Cast*	Converts a string representations of an IPv6 address (with colons and/or dots) as an IPv6 address type. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
ipv6 address <string>	*NamedGlobal*	Converts a string representations of an IPv6 address (with colons and/or dots) into an IPv6 address type. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <ipv6 address>	*Plain*	Returns the maximum value from a list of <ipv6 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <ipv6 address>	*Plain*	Returns the minimum value from a list of <ipv6 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<ipv6 address> as compressed string	*Cast*	<string>	Similar to casting as a string, but with double colons used to represent multiple zero 16-bit parts (RFC 4291 section 2.2, item 2). Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
<ipv6 address> as compressed string with ipv4	*Cast*	<string>	Similar to casting as a string, but with both colon-compression and standard IPv4 representation for the low 32-bits. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
<ipv6 address> as compressed string with ipv4 with zone index	*Cast*	<string>	Similar to casting as a string, but with both colon-compression and standard IPv4 representation for the low 32-bits as well as any zone index. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv6 address> as compressed string with zone index	*Cast*	<string>	Similar to casting as a string, but with both colon-compression and a zone index. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv6 address> as ipv4or6 address	*Cast*	<ipv4or6 address>	Casts an IPv6 address as a more flexible ipv4or6 address. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv6 address> as string	*Cast*	<string>	Returns a string representation of the IPv6 address in the form x:x:x:x:x:x:x:x, where each x represents one to four hexadecimal digits of the eight 16-bit address fields. Leading zeros in an individual field are omitted. See RFC 4291 section 2.2, item 1. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
<ipv6 address> as string with ipv4	*Cast*	<string>	Similar to casting as a string, but with the lowest 32 bits of the address in the standard IPv4 representation (RFC 4291 section 2.2, item 3). Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
<ipv6 address> as string with ipv4 with zone index	*Cast*	<string>	Similar to casting as a string, but with standard IPv4 representation for the low 32-bits as well as any zone index. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv6 address> as string with leading zeros	*Cast*	<string>	Similar to casting as a string, but leading zeros in individual fields are preserved. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
<ipv6 address> as string with leading zeros with zone index	*Cast*	<string>	Converts an IPv6 address into a string with all fields packed with leading zeroes as well as a terminating zone index, if any. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv6 address> as string with zone index	*Cast*	<string>	Converts an IPv6 address into a string with a terminating slash-delimited zone index, if any. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
extrema of <ipv6 address>	*Plain*	<(ipv6 address, ipv6 address)>	Returns the minimum and maximum extreme values of the given list of <ipv6 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Key Phrase	*Form*	Return Type	Description
hexadecet <integer> of <ipv6 address>	*Numbered*	<integer>	Returns one of the eight 16-bit parts (hexadecets) of an IPv6 address specified by <integer>. Here zero refers to the highest hexadecet (network or big-endian order). Thus hexadecet 0 refers to the most-significant 16-bits of the 128 bit IPv6 address. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
ipv4 part of <ipv6 address>	*Plain*	<ipv4 address>	Returns the lowest 32-bits of the IPv6 address as an IPv4 address. • As of version 8.0, this Inspector type is derived from an <ipv4or6 address> type. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
maximum of <ipv6 address>	*Plain*	<ipv6 address>	Returns the maximum value from a list of <ipv6 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
minimum of <ipv6 address>	*Plain*	<ipv6 address>	Returns the minimum value from a list of <ipv6 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu
unique value of <ipv6 address>	*Plain*	<ipv6 address with multiplicity>	Returns the unique values of a given list of <ipv6 address> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Operators

Key phrase	Return Type	Description
<ipv6 address> {cmp} <ipv6 address>	<*boolean*>	Where {cmp} is one of: <, <=, =. Win:7.0, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM

Ipv6 Address with Multiplicity

These Inspectors deal with ipv6 address arrays, allowing you to extract unique ipv6 addresses and count them.

Type Derivation: This object type is derived from the <ipv6 address> type and therefore shares the same properties as that type.

Creation Methods

Key Phrase	*Form*	Description
unique value of <ipv6 address>	*Plain*	Returns the unique values of a given list of <ipv6 address> types, removing duplicates and sorting by value. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <ipv6 address with multiplicity>	*Plain*	<integer>	Sorts the list and returns the multiplicity, or count, of each unique element in the specified list of multiple <ipv6 address> types. Win:7.1, Lin:7.1, Sol:7.1, HPUX:7.1, AIX:7.1, Mac:7.1, WM, Ubu

Ipv4or6 Address

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

Creation Methods

Key Phrase	*Form*	Description
<ipv4 address> as ipv4or6 address	*Cast*	Converts an IPv4 address into the more flexible ipv4or6 type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv6 address> as ipv4or6 address	*Cast*	Casts an IPv6 address as a more flexible ipv4or6 address. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<string> as ipv4or6 address	*Cast*	Converts a string version of an IP address into the flexible ipv4or6 address format. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	*Form*	Description
ipv4or6 address <string>	*NamedGlobal*	Returns an ipv4or6 address object as described by the given string. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
maximum of <ipv4or6 address>	*Plain*	Returns the maximum of the specified list of <ipv4or6 address> types. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
minimum of <ipv4or6 address>	*Plain*	Returns the minimum of the specified list of <ipv4or6 address> types. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<ipv4or6 address> as compressed string	*Cast*	<string>	Compresses the IP address, using double colons to stand in for repeated zeroes in the address. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4or6 address> as compressed string with ipv4	*Cast*	<string>	Similar to casting to a string, but with both colon-compression and standard IPv4 representation for the low 32-bits. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4or6 address> as compressed string with ipv4 with zone index	*Cast*	<string>	Creates a compressed string (replacing multiple zeroes with double colons) with the lower 32 bits in IPv4 format. It includes the zone index (on Windows, typically a number) following a percent sign (%). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4or6 address> as compressed string with zone index	*Cast*	<string>	Creates a compressed string (replacing multiple zeroes with double colons) and includes the zone index (on Windows, typically a number) following a percent sign (%). Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4or6 address> as string	*Cast*	<string>	Casts an IP address (v4 or 6) as a string type. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4or6 address> as string with ipv4	*Cast*	<string>	Converts an address to a string with the lower 32 bits in IPv4 format. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4or6 address> as string with ipv4 with zone index	*Cast*	<string>	Converts an address to a string with the lower 32 bits in IPv4 format as well as any zone index. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrase	*Form*	Return Type	Description
<ipv4or6 address> as string with leading zeros	*Cast*	<string>	Converts an IP address into a string with all fields packed with leading zeroes. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4or6 address> as string with leading zeros with zone index	*Cast*	<string>	Converts an IP address into a string with all fields packed with leading zeroes, as well as any zone index. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
<ipv4or6 address> as string with zone index	*Cast*	<string>	Converts an IP address into a string with a zone index, if any. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
extrema of <ipv4or6 address>	*Plain*	<(ipv4or6 address, ipv4or6 address)>	Returns the minimum and maximum of the specified list of ipv4or6 addresses. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
hexadecet <integer> of <ipv4or6 address>	*Numbered*	<integer>	Returns one of the eight 16-bit parts (hexadecets) of an ipv4or6 address specified by <integer>. Here zero refers to the highest hexadecet (network or big-endian order). Thus hexadecet 0 refers to the most-significant 16-bits of the 128 bit ipv4or6 address. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ip version of <ipv4or6 address>	*Plain*	<ip version>	Returns the IP version of the specified ipv4or6 address. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
ipv4 part of <ipv4or6 address>	*Plain*	<ipv4 address>	Returns the lowest 32-bits of the IPv4or6 address as an IPv4 address. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
maximum of <ipv4or6 address>	*Plain*	<ipv4or6 address>	Returns the maximum of the specified list of <ipv4or6 address> types. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
minimum of <ipv4or6 address>	*Plain*	<ipv4or6 address>	Returns the minimum of the specified list of <ipv4or6 address> types. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
unique value of <ipv4or6 address>	*Plain*	<ipv4or6 address with multiplicity>	Returns the unique values of a given list of addresses (ipv4or6), removing duplicates and sorting by value. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Operators

Key phrase	Return Type	Description
<ipv4or6 address> {cmp} <ipv4or6 address>	*<boolean>*	Where {cmp} is one of: <, <=, =. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0
<ipv4or6 address> {cmp} <string>	*<boolean>*	Where {cmp} is one of: <, <=, =. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0
<string> {cmp} <ipv4or6 address>	*<boolean>*	Where {cmp} is one of: <, <=, =. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0

Examples

- `("0:0:0:0:0:ffff:c0a8:16a" as ipv4or6 address) as compressed string`

  Returns "::ffff:c0a8:16a", a compressed string with a double colon standing in for leading zeroes.

- `("0:0:0:0:0:ffff:c0a8:16a" as ipv4or6 address) as compressed string with ipv4`

  Returns "::ffff:192.168.1.106", a compressed string (leading zeroes collapsed to ::) with a four-byte IPv4 address as the lower 32 bits.

Ipv4or6 Address with Multiplicity

These Inspectors deal with arrays of ipv4or6 addresses, allowing you to extract unique addresses and count them.

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

Creation Methods

Key Phrase	*Form*	Description
unique value of <ipv4or6 address>	*Plain*	Returns the unique values of a given list of ipv4or6 addresses, removing duplicates and sorting by value. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
multiplicity of <ipv4or6 address with multiplicity>	*Plain*	<integer>	Returns the multiplicity (how many entries of each unique address) of a list of ipv4or6 addresses. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Distinguished Name

These Inspectors refer to the distinguished name (DN) as defined by the Microsoft Active Directory service. See the core inspector guide for a list of properties of a distinguished name object.

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

Creation Methods

Key Phrase	*Form*	Description
distinguished name <string>	*NamedGlobal*	Interprets a distinguishedname in the textual format described by RFC 2253 and used by the LDAP/Active Directory. Win:7.0

Properties

Key Phrase	*Form*	Return Type	Description
component <integer> of <distinguished name>	*Numbered*	<distinguished name component>	Returns the component (as specified by <integer>) of the distinguished name. Win:7.0
component of <distinguished name>	*Plain*	<distinguished name component>	Used as a plural, returns a list of the components of the given distinguished name. Win:7.0

Distinguished Name Component

These Inspectors refer to the various components of the distinguished name (DN) as defined by the Microsoft Active Directory API. A DN is a sequence of relative distinguished names (RDN) connected by commas. An RDN is an attribute with an associated value in the form attribute=value; normally expressed in a UTF-8 string format. For more information, see the MSDN article on Distinguished Names.

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

Creation Methods

Key Phrase	*Form*	Description
component <integer> of <distinguished name>	*Numbered*	Returns the component (as specified by <integer>) of the distinguished name. Win:7.0

Key Phrase	*Form*	Description
component of <distinguished name>	*Plain*	Used as a plural, returns a list of the components of the given distinguished name. Win:7.0

Properties

Key Phrase	*Form*	Return Type	Description
type of <distinguished name component>	*Plain*	<string>	Returns the type of the specified component of the distinguished name. Win:7.0
value of <distinguished name component>	*Plain*	<string>	Returns the value of the specified component of the distinguished name. Win:7.0

Examples

- `component 0 of distinguished name "CN=BIGFOOT,CN=Computers,DC=devlan,DC=bigfix,DC=com"`

  Returns "CN=BIGFOOT".

- `(type of it, value of it) of components of distinguished name "CN=BIGFOOT,CN=Computers,DC=devlan,DC=bigfix,DC=com"`

  Returns the list:

  - CN, BIGFOOT
  - CN, Computers
  - DC, devlan
  - DC, bigfix
  - DC, com.

- `value of component 0 of distinguished name "CN=BIGFOOT,CN=Computers,DC=devlan,DC=bigfix,DC=com"`

  Returns "BIGFOOT".

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
index type of <property>	*Plain*	The type (if any) required before or without the keyword "of" in an expression using the property. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
left operand type of <binary operator>	*Plain*	The type required before the operator in an expression. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
operand type of <cast>	*Plain*	The type required before the keyword "as" in an expression using the cast. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
operand type of <unary operator>	*Plain*	The type required in an expression using the operator. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
parent of <type>	*Plain*	The types (if any) whose properties are inherited by this type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
result type of <binary operator>	*Plain*	The type that the binary operator produces. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
result type of <cast>	*Plain*	The type that the casting operator produces. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
result type of <property>	*Plain*	The type that the property produces. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
result type of <unary operator>	*Plain*	The type that the unary operator produces. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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

Properties

Key Phrase	*Form*	Return Type	Description
<type> as string	*Cast*	<string>	A string indicating the type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
cast from of <type>	*Plain*	<cast>	Returns the casts that can be created from the specified <type>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
name of <type>	*Plain*	<string>	A string naming the type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
parent of <type>	*Plain*	<type>	The types (if any) whose properties are inherited by this type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
property returning <type> of <type>	*Index<type>*	<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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
size of <type>	*Plain*	<integer>	The number of bytes used in the internal representation of an object of the given type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Operators

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

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
property returning <type>	*Index<type>Global*	Produces a list of the Inspector properties that return the "file" type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
property returning <type> of <type>	*Index<type>*	Returns Inspectors of the form <type> of <type>. Typically there is more than one property, so this is often used in the plural. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<property> as string	*Cast*	<string>	A short description of the use of the property. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
dependency known of <property>	*Plain*	<boolean>	This introspector returns TRUE if a property's dependency is known. If it isn't, then properties containing the expression can't be fingerprinted, an optimization that exempts expressions from re-evaluation if their dependencies haven't changed. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu
direct object type of <property>	*Plain*	<type>	The type (if any) required after the keyword "of" in an expression using the property. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
index type of <property>	*Plain*	<type>	The type (if any) required before or without the keyword "of" in an expression using the property. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
multivalued of <property>	*Plain*	<boolean>	Can the property have more than one value for a single input?. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
plural name of <property>	*Plain*	<string>	The name of the property, in the plural. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
result type of <property>	*Plain*	<type>	The type that the property produces. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
singular name of <property>	*Plain*	<string>	The name of the property, in the singular. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
usual name of <property>	*Plain*	<string>	Returns the usual name of the specified property. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
binary operator <string>	*NamedGlobal*	Typically used in the plural, returns the various possible binary inspectors that use the specified operators. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
binary operator returning <type>	*Index<type>Global*	Returns a list of binary operators that return the specified type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

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

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
unary operator <string>	*NamedGlobal*	Typically used in the plural, this inspector returns a list of objects that use the specified operator. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
unary operator returning <type>	*Index<type>Global*	Returns a list of the unary operator inspectors (such as negative) that return the specified type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

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

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, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
cast <string>	*NamedGlobal*	Returns a list of the objects that can be cast into the type specified by <string>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
cast from of <type>	*Plain*	Returns the casts that can be created from the specified <type>. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
cast returning <type>	*Index<type>Global*	Returns a list of the objects that can be cast into the specified type. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu

Properties

Key Phrase	*Form*	Return Type	Description
<cast> as string	*Cast*	<string>	A short description of the use of the cast. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
name of <cast>	*Plain*	<string>	The phrase used after the keyword "as" in an expression using the cast. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
operand type of <cast>	*Plain*	<type>	The type required before the keyword "as" in an expression using the cast. Win, Lin, Sol, HPUX, AIX, Mac, WM, Ubu
result type of <cast>	*Plain*	<type>	The type that the casting operator produces. Win:8.0, Lin:8.0, Sol:8.0, HPUX:8.0, AIX:8.0, Mac:8.0, Ubu

Key Phrases (Inspectors)

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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
abbr <string> of <html>	abbrs	<html>	<html>	*Named*	core
abbr <string> of <string>	abbrs	<html>	<string>	*Named*	core
abbr of <html>	abbrs	<html>	<html>	*Plain*	core
abbr of <string>	abbrs	<html>	<string>	*Plain*	core
absolute value of <hertz>	absolute values	<hertz>	<hertz>	*Plain*	core
absolute value of <integer>	absolute values	<integer>	<integer>	*Plain*	core
absolute value of <time interval>	absolute values	<time interval>	<time interval>	*Plain*	core
acronym <string> of <html>	acronyms	<html>	<html>	*Named*	core
acronym <string> of <string>	acronyms	<html>	<string>	*Named*	core
acronym of <html>	acronyms	<html>	<html>	*Plain*	core
acronym of <string>	acronyms	<html>	<string>	*Plain*	core
address <string> of <html>	addresss	<html>	<html>	*Named*	core
address <string> of <string>	addresss	<html>	<string>	*Named*	core
address of <html>	addresss	<html>	<html>	*Plain*	core
address of <string>	addresss	<html>	<string>	*Plain*	core
anchor <string> of <html>	anchors	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
anchor <string> of <string>	anchors	<html>	<string>	*Named*	core
anchor of <html>	anchors	<html>	<html>	*Plain*	core
anchor of <string>	anchors	<html>	<string>	*Plain*	core
any ip version	any ip versions	<ip version>	<world>	*PlainGlobal*	core
april	aprils	<month>	<world>	*PlainGlobal*	core
april <integer>	aprils	<day of year>	<world>	*NumberedGlobal*	core
april <integer> of <integer>	aprils	<date>	<integer>	*Numbered*	core
april of <integer>	aprils	<month and year>	<integer>	*Plain*	core
attribute <integer> of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Numbered*	core
attribute <string> of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Named*	core
attribute of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Plain*	core
august	augusts	<month>	<world>	*PlainGlobal*	core
august <integer>	augusts	<day of year>	<world>	*NumberedGlobal*	core
august <integer> of <integer>	augusts	<date>	<integer>	*Numbered*	core
august of <integer>	augusts	<month and year>	<integer>	*Plain*	core
b <string> of <html>	bs	<html>	<html>	*Named*	core
b <string> of <string>	bs	<html>	<string>	*Named*	core
b of <html>	bs	<html>	<html>	*Plain*	core
b of <string>	bs	<html>	<string>	*Plain*	core
base <string> of <html>	bases	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
base <string> of <string>	bases	<html>	<string>	*Named*	core
base of <html>	bases	<html>	<html>	*Plain*	core
base of <string>	bases	<html>	<string>	*Plain*	core
big <string> of <html>	bigs	<html>	<html>	*Named*	core
big <string> of <string>	bigs	<html>	<string>	*Named*	core
big of <html>	bigs	<html>	<html>	*Plain*	core
big of <string>	bigs	<html>	<string>	*Plain*	core
binary operator <string>	binary operators	<binary operator>	<world>	*NamedGlobal*	core
binary operator returning <type>	binary operators returning	<binary operator>	<world>	*Index<type>Global*	core
bit <integer>	bits	<bit set>	<world>	*NumberedGlobal*	core
bit <integer> of <bit set>	bits	<boolean>	<bit set>	*Numbered*	core
bit <integer> of <integer>	bits	<boolean>	<integer>	*Numbered*	core
bit set <string>	bit sets	<bit set>	<world>	*NamedGlobal*	core
blockquote <string> of <html>	blockquotes	<html>	<html>	*Named*	core
blockquote <string> of <string>	blockquotes	<html>	<string>	*Named*	core
blockquote of <html>	blockquotes	<html>	<html>	*Plain*	core
blockquote of <string>	blockquotes	<html>	<string>	*Plain*	core
body <string> of <html>	bodys	<html>	<html>	*Named*	core
body <string> of <string>	bodys	<html>	<string>	*Named*	core
body of <html>	bodys	<html>	<html>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
body of <string>	bodys	<html>	<string>	*Plain*	core
boolean <string>	booleans	<boolean>	<world>	*NamedGlobal*	core
br	brs	<html>	<world>	*PlainGlobal*	core
br <string>	brs	<html>	<world>	*NamedGlobal*	core
caption <string> of <html>	captions	<html>	<html>	*Named*	core
caption <string> of <string>	captions	<html>	<string>	*Named*	core
caption of <html>	captions	<html>	<html>	*Plain*	core
caption of <string>	captions	<html>	<string>	*Plain*	core
case insensitive regex <string>	case insensitive regexes	<regular expression>	<world>	*NamedGlobal*	regx
case insensitive regular expression <string>	case insensitive regular expressions	<regular expression>	<world>	*NamedGlobal*	regx
cast <string>	casts	<cast>	<world>	*NamedGlobal*	core
cast from of <type>	casts from	<cast>	<type>	*Plain*	core
cast returning <type>	casts returning	<cast>	<world>	*Index<type>Global*	core
character <integer>	characters	<string>	<world>	*NumberedGlobal*	core
character <integer> of <string>	characters	<substring>	<string>	*Numbered*	core
character of <string>	characters	<substring>	<string>	*Plain*	core
child node <integer> of <xml dom node>	child nodes	<xml dom node>	<xml dom node>	*Numbered*	core
child node of <xml dom node>	child nodes	<xml dom node>	<xml dom node>	*Plain*	core
cite <string> of <html>	cites	<html>	<html>	*Named*	core
cite <string> of <string>	cites	<html>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
cite of <html>	cites	<html>	<html>	*Plain*	core
cite of <string>	cites	<html>	<string>	*Plain*	core
code <string> of <html>	codes	<html>	<html>	*Named*	core
code <string> of <string>	codes	<html>	<string>	*Named*	core
code of <html>	codes	<html>	<html>	*Plain*	core
code of <string>	codes	<html>	<string>	*Plain*	core
col <string> of <html>	cols	<html>	<html>	*Named*	core
col <string> of <string>	cols	<html>	<string>	*Named*	core
col of <html>	cols	<html>	<html>	*Plain*	core
col of <string>	cols	<html>	<string>	*Plain*	core
colgroup <string> of <html>	colgroups	<html>	<html>	*Named*	core
colgroup <string> of <string>	colgroups	<html>	<string>	*Named*	core
colgroup of <html>	colgroups	<html>	<html>	*Plain*	core
colgroup of <string>	colgroups	<html>	<string>	*Plain*	core
component <integer> of <distinguished name>	components	<distinguished name component>	<distinguished name>	*Numbered*	core
component <integer> of <site version list>	components	<integer>	<site version list>	*Numbered*	core
component of <distinguished name>	components	<distinguished name component>	<distinguished name>	*Plain*	core
concatenation <html> of <html>	concatenations	<html>	<html>	*Index<html>*	core
concatenation <html> of <string>	concatenations	<html>	<string>	*Index<html>*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
concatenation <string> of <html>	concatenations	<html>	<html>	*Named*	core
concatenation <string> of <string>	concatenations	<string>	<string>	*Named*	core
concatenation of <html>	concatenations	<html>	<html>	*Plain*	core
concatenation of <string>	concatenations	<string>	<string>	*Plain*	core
conjunction of <boolean>	conjunctions	<boolean>	<boolean>	*Plain*	core
cryptography	cryptographies	<cryptography>	<world>	*PlainGlobal*	core
current date	current dates	<date>	<world>	*PlainGlobal*	core
current day_of_month	current days_of_month	<day of month>	<world>	*PlainGlobal*	core
current day_of_week	current days_of_week	<day of week>	<world>	*PlainGlobal*	core
current day_of_year	current days_of_year	<day of year>	<world>	*PlainGlobal*	core
current month	current months	<month>	<world>	*PlainGlobal*	core
current month_and_year	current months_and_years	<month and year>	<world>	*PlainGlobal*	core
current time_of_day	current times_of_day	<time of day with time zone>	<world>	*PlainGlobal*	core
current time_of_day <time zone>	current times_of_day	<time of day with time zone>	<world>	*Index<time zone>Global*	core
current year	current years	<year>	<world>	*PlainGlobal*	core
date <string>	dates	<date>	<world>	*NamedGlobal*	core
date <time zone> of <time>	dates	<date>	<time>	*Index<time zone>*	core

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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
del <string> of <html>	dels	<html>	<html>	*Named*	core
del <string> of <string>	dels	<html>	<string>	*Named*	core
del of <html>	dels	<html>	<html>	*Plain*	core
del of <string>	dels	<html>	<string>	*Plain*	core
dependency known of <property>	dependencies known	<boolean>	<property>	*Plain*	core
desired fips mode of <cryptography>	desired fips modes	<boolean>	<cryptography>	*Plain*	core
dfn <string> of <html>	dfns	<html>	<html>	*Named*	core
dfn <string> of <string>	dfns	<html>	<string>	*Named*	core
dfn of <html>	dfns	<html>	<html>	*Plain*	core
dfn of <string>	dfns	<html>	<string>	*Plain*	core
direct object type of <property>	direct object types	<type>	<property>	*Plain*	core
disjunction of <boolean>	disjunctions	<boolean>	<boolean>	*Plain*	core
distinguished name <string>	distinguished names	<distinguished name>	<world>	*NamedGlobal*	core
div <string> of <html>	divs	<html>	<html>	*Named*	core
div <string> of <string>	divs	<html>	<string>	*Named*	core
div of <html>	divs	<html>	<html>	*Plain*	core
div of <string>	divs	<html>	<string>	*Plain*	core
divided by zero of <floating point>	divided by zeroes	<boolean>	<floating point>	*Plain*	core
dt <string> of <html>	dts	<html>	<html>	*Named*	core
dt <string> of <string>	dts	<html>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
dt of <html>	dts	<html>	<html>	*Plain*	core
dt of <string>	dts	<html>	<string>	*Plain*	core
element of <integer set>	elements	<integer>	<integer set>	*Plain*	core
element of <string set>	elements	<string>	<string set>	*Plain*	core
em <string> of <html>	ems	<html>	<html>	*Named*	core
em <string> of <string>	ems	<html>	<string>	*Named*	core
em of <html>	ems	<html>	<html>	*Plain*	core
em of <string>	ems	<html>	<string>	*Plain*	core
end of <substring>	ends	<string position>	<substring>	*Plain*	core
end of <time range>	ends	<time>	<time range>	*Plain*	core
error <string>	errors	<undefined>	<world>	*NamedGlobal*	core
extrema of <date>	extremas	<(date, date)>	<date>	*Plain*	core
extrema of <day of month>	extremas	<(day of month, day of month)>	<day of month>	*Plain*	core
extrema of <day of year>	extremas	<(day of year, day of year)>	<day of year>	*Plain*	core
extrema of <floating point>	extremas	<(floating point, floating point)>	<floating point>	*Plain*	core
extrema of <hertz>	extremas	<(hertz, hertz)>	<hertz>	*Plain*	core
extrema of <integer>	extremas	<(integer, integer)>	<integer>	*Plain*	core
extrema of <ipv4 address>	extremas	<(ipv4 address, ipv4 address)>	<ipv4 address>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
extrema of <ipv4or6 address>	extremas	<(ipv4or6 address, ipv4or6 address)>	<ipv4or6 address>	*Plain*	core
extrema of <ipv6 address>	extremas	<(ipv6 address, ipv6 address)>	<ipv6 address>	*Plain*	core
extrema of <month and year>	extremas	<(month and year, month and year)>	<month and year>	*Plain*	core
extrema of <month>	extremas	<(month, month)>	<month>	*Plain*	core
extrema of <number of months>	extremas	<(number of months, number of months)>	<number of months>	*Plain*	core
extrema of <site version list>	extremas	<(site version list, site version list)>	<site version list>	*Plain*	core
extrema of <time interval>	extremas	<(time interval, time interval)>	<time interval>	*Plain*	core
extrema of <time of day>	extremas	<(time of day, time of day)>	<time of day>	*Plain*	core
extrema of <time>	extremas	<(time, time)>	<time>	*Plain*	core
extrema of <version>	extremas	<(version, version)>	<version>	*Plain*	core
extrema of <year>	extremas	<(year, year)>	<year>	*Plain*	core
false	falses	<boolean>	<world>	*PlainGlobal*	core
february	februarys	<month>	<world>	*PlainGlobal*	core
february <integer>	februarys	<day of year>	<world>	*NumberedGlobal*	core
february <integer> of <integer>	februarys	<date>	<integer>	*Numbered*	core
february of <integer>	februarys	<month and year>	<integer>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
final part <time interval> of <time range>	final parts	<time range>	<time range>	*Index<time interval>*	core
finite of <floating point>	finites	<boolean>	<floating point>	*Plain*	core
fips mode failure message of <cryptography>	fips mode failure messages	<string>	<cryptography>	*Plain*	core
fips mode of <cryptography>	fips modes	<boolean>	<cryptography>	*Plain*	core
first <day of week> of <month and year>	firsts	<date>	<month and year>	*Index<day of week>*	core
first <integer> of <string>	firsts	<substring>	<string>	*Numbered*	core
first <string> of <string>	firsts	<substring>	<string>	*Named*	core
first child of <xml dom node>	first children	<xml dom node>	<xml dom node>	*Plain*	core
first friday of <month and year>	first fridays	<date>	<month and year>	*Plain*	core
first match <regular expression> of <string>	first matches	<regular expression match>	<string>	*Index<regular expression>*	regx
first monday of <month and year>	first mondays	<date>	<month and year>	*Plain*	core
first saturday of <month and year>	first saturdays	<date>	<month and year>	*Plain*	core
first sunday of <month and year>	first sundays	<date>	<month and year>	*Plain*	core
first thursday of <month and year>	first thursdays	<date>	<month and year>	*Plain*	core
first tuesday of <month and year>	first tuesdays	<date>	<month and year>	*Plain*	core
first wednesday of <month and year>	first wednesdays	<date>	<month and year>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
floating point <floating point>	floating points	<floating point>	<world>	*Index<floating point>Global*	core
floating point <string>	floating points	<floating point>	<world>	*NamedGlobal*	core
following text of <string position>	following texts	<substring>	<string position>	*Plain*	core
following text of <substring>	following texts	<substring>	<substring>	*Plain*	core
format <string>	formats	<format>	<world>	*NamedGlobal*	core
friday	fridays	<day of week>	<world>	*PlainGlobal*	core
ghz	ghzs	<hertz>	<world>	*PlainGlobal*	core
greatest hz	greatest hzs	<hertz>	<world>	*PlainGlobal*	core
greatest integer	greatest integers	<integer>	<world>	*PlainGlobal*	core
greatest time interval	greatest time intervals	<time interval>	<world>	*PlainGlobal*	core
h1 <string> of <html>	h1s	<html>	<html>	*Named*	core
h1 <string> of <string>	h1s	<html>	<string>	*Named*	core
h1 of <html>	h1s	<html>	<html>	*Plain*	core
h1 of <string>	h1s	<html>	<string>	*Plain*	core
h2 <string> of <html>	h2s	<html>	<html>	*Named*	core
h2 <string> of <string>	h2s	<html>	<string>	*Named*	core
h2 of <html>	h2s	<html>	<html>	*Plain*	core
h2 of <string>	h2s	<html>	<string>	*Plain*	core
h3 <string> of <html>	h3s	<html>	<html>	*Named*	core
h3 <string> of <string>	h3s	<html>	<string>	*Named*	core
h3 of <html>	h3s	<html>	<html>	*Plain*	core
h3 of <string>	h3s	<html>	<string>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
h4 <string> of <html>	h4s	<html>	<html>	*Named*	core
h4 <string> of <string>	h4s	<html>	<string>	*Named*	core
h4 of <html>	h4s	<html>	<html>	*Plain*	core
h4 of <string>	h4s	<html>	<string>	*Plain*	core
h5 <string> of <html>	h5s	<html>	<html>	*Named*	core
h5 <string> of <string>	h5s	<html>	<string>	*Named*	core
h5 of <html>	h5s	<html>	<html>	*Plain*	core
h5 of <string>	h5s	<html>	<string>	*Plain*	core
h6 <string> of <html>	h6s	<html>	<html>	*Named*	core
h6 <string> of <string>	h6s	<html>	<string>	*Named*	core
h6 of <html>	h6s	<html>	<html>	*Plain*	core
h6 of <string>	h6s	<html>	<string>	*Plain*	core
head <string> of <html>	heads	<html>	<html>	*Named*	core
head <string> of <string>	heads	<html>	<string>	*Named*	core
head of <html>	heads	<html>	<html>	*Plain*	core
head of <string>	heads	<html>	<string>	*Plain*	core
hexadecet <integer> of <ipv4or6 address>	hexadecets	<integer>	<ipv4or6 address>	*Numbered*	core
hexadecet <integer> of <ipv6 address>	hexadecets	<integer>	<ipv6 address>	*Numbered*	core
hexadecimal integer <string>	hexadecimal integers	<integer>	<world>	*NamedGlobal*	core
hexadecimal string <string>	hexadecimal strings	<string>	<world>	*NamedGlobal*	core
hour	hours	<time interval>	<world>	*PlainGlobal*	core

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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
html tag <string> of <string>	html tags	<html>	<string>	*Named*	core
hz	hzs	<hertz>	<world>	*PlainGlobal*	core
index type of <property>	index types	<type>	<property>	*Plain*	core
inexact of <floating point>	inexacts	<boolean>	<floating point>	*Plain*	core
infinite of <floating point>	infinites	<boolean>	<floating point>	*Plain*	core
initial part <time interval> of <time range>	initial parts	<time range>	<time range>	*Index<time interval>*	core
ins <string> of <html>	inss	<html>	<html>	*Named*	core
ins <string> of <string>	inss	<html>	<string>	*Named*	core
ins of <html>	inss	<html>	<html>	*Plain*	core
ins of <string>	inss	<html>	<string>	*Plain*	core
integer <integer>	integers	<integer>	<world>	*NumberedGlobal*	core
integer <string>	integers	<integer>	<world>	*NamedGlobal*	core
integer ceiling of <floating point>	integer ceilings	<integer>	<floating point>	*Plain*	core
integer floor of <floating point>	integer floors	<integer>	<floating point>	*Plain*	core
integer in <(integer, integer)>	integers in	<integer>	<world>	*Index<(integer, integer)>Global*	core
integer in <(integer, integer, integer)>	integers in	<integer>	<world>	*Index<(integer, integer, integer)>Global*	core
integer to <integer>	integers to	<integer>	<world>	*NumberedGlobal*	core
intersection of <integer set>	intersections	<integer set>	<integer set>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
intersection of <string set>	intersections	<string set>	<string set>	*Plain*	core
invalid before of <x509 certificate>	invalid befores	<time>	<x509 certificate>	*Plain*	core
invalid of <floating point>	invalids	<boolean>	<floating point>	*Plain*	core
ip version <integer>	ip versions	<ip version>	<world>	*NumberedGlobal*	core
ip version of <ipv4or6 address>	ip versions	<ip version>	<ipv4or6 address>	*Plain*	core
ipv4	ipv4s	<ip version>	<world>	*PlainGlobal*	core
ipv4 address <string>	ipv4 addresses	<ipv4 address>	<world>	*NamedGlobal*	core
ipv4 part of <ipv4or6 address>	ipv4 parts	<ipv4 address>	<ipv4or6 address>	*Plain*	core
ipv4 part of <ipv6 address>	ipv4 parts	<ipv4 address>	<ipv6 address>	*Plain*	core
ipv4or6 address <string>	ipv4or6 addresses	<ipv4or6 address>	<world>	*NamedGlobal*	core
ipv6	ipv6s	<ip version>	<world>	*PlainGlobal*	core
ipv6 address <string>	ipv6 addresses	<ipv6 address>	<world>	*NamedGlobal*	core
italic <string> of <html>	italics	<html>	<html>	*Named*	core
italic <string> of <string>	italics	<html>	<string>	*Named*	core
italic of <html>	italics	<html>	<html>	*Plain*	core
italic of <string>	italics	<html>	<string>	*Plain*	core
january	januarys	<month>	<world>	*PlainGlobal*	core
january <integer>	januarys	<day of year>	<world>	*NumberedGlobal*	core
january <integer> of <integer>	januarys	<date>	<integer>	*Numbered*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
january of <integer>	januarys	<month and year>	<integer>	*Plain*	core
july	julys	<month>	<world>	*PlainGlobal*	core
july <integer>	julys	<day of year>	<world>	*NumberedGlobal*	core
july <integer> of <integer>	julys	<date>	<integer>	*Numbered*	core
july of <integer>	julys	<month and year>	<integer>	*Plain*	core
june	junes	<month>	<world>	*PlainGlobal*	core
june <integer>	junes	<day of year>	<world>	*NumberedGlobal*	core
june <integer> of <integer>	junes	<date>	<integer>	*Numbered*	core
june of <integer>	junes	<month and year>	<integer>	*Plain*	core
kbd <string> of <html>	kbds	<html>	<html>	*Named*	core
kbd <string> of <string>	kbds	<html>	<string>	*Named*	core
kbd of <html>	kbds	<html>	<html>	*Plain*	core
kbd of <string>	kbds	<html>	<string>	*Plain*	core
khz	khzs	<hertz>	<world>	*PlainGlobal*	core
last <integer> of <string>	lasts	<substring>	<string>	*Numbered*	core
last <string> of <string>	lasts	<substring>	<string>	*Named*	core
last child of <xml dom node>	last children	<xml dom node>	<xml dom node>	*Plain*	core
leap of <year>	leaps	<boolean>	<year>	*Plain*	core
least hz	least hzs	<hertz>	<world>	*PlainGlobal*	core
least integer	least integers	<integer>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
least significant one bit of <bit set>	least significant one bits	<integer>	<bit set>	*Plain*	core
least time interval	least time intervals	<time interval>	<world>	*PlainGlobal*	core
left operand type of <binary operator>	left operand types	<type>	<binary operator>	*Plain*	core
left shift <integer> of <bit set>	left shifts	<bit set>	<bit set>	*Numbered*	core
length of <month and year>	lengths	<time interval>	<month and year>	*Plain*	core
length of <rope>	lengths	<integer>	<rope>	*Plain*	core
length of <string>	lengths	<integer>	<string>	*Plain*	core
length of <time range>	lengths	<time interval>	<time range>	*Plain*	core
length of <year>	lengths	<time interval>	<year>	*Plain*	core
less significance <integer> of <floating point>	less significances	<floating point>	<floating point>	*Numbered*	core
li <string> of <html>	lis	<html>	<html>	*Named*	core
li <string> of <string>	lis	<html>	<string>	*Named*	core
li of <html>	lis	<html>	<html>	*Plain*	core
li of <string>	lis	<html>	<string>	*Plain*	core
link <string> of <html>	links	<html>	<html>	*Named*	core
link <string> of <string>	links	<html>	<string>	*Named*	core
link of <html>	links	<html>	<html>	*Plain*	core
link of <string>	links	<html>	<string>	*Plain*	core
local time <string>	local times	<time>	<world>	*NamedGlobal*	core
local time zone	local time zones	<time zone>	<world>	*PlainGlobal*	core
march	marchs	<month>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
march <integer>	marchs	<day of year>	<world>	*NumberedGlobal*	core
march <integer> of <integer>	marchs	<date>	<integer>	*Numbered*	core
march of <integer>	marchs	<month and year>	<integer>	*Plain*	core
match <regular expression> of <string>	matches	<regular expression match>	<string>	*Index<regular expression>*	regx
maximum of <date>	maxima	<date>	<date>	*Plain*	core
maximum of <day of month>	maxima	<day of month>	<day of month>	*Plain*	core
maximum of <day of year>	maxima	<day of year>	<day of year>	*Plain*	core
maximum of <floating point>	maxima	<floating point>	<floating point>	*Plain*	core
maximum of <hertz>	maxima	<hertz>	<hertz>	*Plain*	core
maximum of <integer>	maxima	<integer>	<integer>	*Plain*	core
maximum of <ipv4 address>	maxima	<ipv4 address>	<ipv4 address>	*Plain*	core
maximum of <ipv4or6 address>	maxima	<ipv4or6 address>	<ipv4or6 address>	*Plain*	core
maximum of <ipv6 address>	maxima	<ipv6 address>	<ipv6 address>	*Plain*	core
maximum of <month and year>	maxima	<month and year>	<month and year>	*Plain*	core
maximum of <month>	maxima	<month>	<month>	*Plain*	core
maximum of <number of months>	maxima	<number of months>	<number of months>	*Plain*	core
maximum of <site version list>	maxima	<site version list>	<site version list>	*Plain*	core
maximum of <time interval>	maxima	<time interval>	<time interval>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
maximum of <time of day>	maxima	<time of day>	<time of day>	*Plain*	core
maximum of <time>	maxima	<time>	<time>	*Plain*	core
maximum of <version>	maxima	<version>	<version>	*Plain*	core
maximum of <year>	maxima	<year>	<year>	*Plain*	core
may	mays	<month>	<world>	*PlainGlobal*	core
may <integer>	mays	<day of year>	<world>	*NumberedGlobal*	core
may <integer> of <integer>	mays	<date>	<integer>	*Numbered*	core
may of <integer>	mays	<month and year>	<integer>	*Plain*	core
mean of <floating point>	means	<floating point>	<floating point>	*Plain*	core
mean of <integer>	means	<floating point>	<integer>	*Plain*	core
meta <string> of <html>	metas	<html>	<html>	*Named*	core
meta <string> of <string>	metas	<html>	<string>	*Named*	core
meta of <html>	metas	<html>	<html>	*Plain*	core
meta of <string>	metas	<html>	<string>	*Plain*	core
mhz	mhzs	<hertz>	<world>	*PlainGlobal*	core
microsecond	microseconds	<time interval>	<world>	*PlainGlobal*	core
midnight	midnights	<time of day>	<world>	*PlainGlobal*	core
millisecond	milliseconds	<time interval>	<world>	*PlainGlobal*	core
minimum of <date>	minima	<date>	<date>	*Plain*	core
minimum of <day of month>	minima	<day of month>	<day of month>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
minimum of <day of year>	minima	<day of year>	<day of year>	*Plain*	core
minimum of <floating point>	minima	<floating point>	<floating point>	*Plain*	core
minimum of <hertz>	minima	<hertz>	<hertz>	*Plain*	core
minimum of <integer>	minima	<integer>	<integer>	*Plain*	core
minimum of <ipv4 address>	minima	<ipv4 address>	<ipv4 address>	*Plain*	core
minimum of <ipv4or6 address>	minima	<ipv4or6 address>	<ipv4or6 address>	*Plain*	core
minimum of <ipv6 address>	minima	<ipv6 address>	<ipv6 address>	*Plain*	core
minimum of <month and year>	minima	<month and year>	<month and year>	*Plain*	core
minimum of <month>	minima	<month>	<month>	*Plain*	core
minimum of <number of months>	minima	<number of months>	<number of months>	*Plain*	core
minimum of <site version list>	minima	<site version list>	<site version list>	*Plain*	core
minimum of <time interval>	minima	<time interval>	<time interval>	*Plain*	core
minimum of <time of day>	minima	<time of day>	<time of day>	*Plain*	core
minimum of <time>	minima	<time>	<time>	*Plain*	core
minimum of <version>	minima	<version>	<version>	*Plain*	core
minimum of <year>	minima	<year>	<year>	*Plain*	core
minute	minutes	<time interval>	<world>	*PlainGlobal*	core
minute_of_hour of <time of day with time zone>	minutes_of_hour	<integer>	<time of day with time zone>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
minute_of_hour of <time of day>	minutes_of_hour	<integer>	<time of day>	*Plain*	core
module <string>	modules	<module>	<world>	*NamedGlobal*	core
monday	mondays	<day of week>	<world>	*PlainGlobal*	core
month	months	<number of months>	<world>	*PlainGlobal*	core
month <integer>	months	<month>	<world>	*NumberedGlobal*	core
month <string>	months	<month>	<world>	*NamedGlobal*	core
month of <date>	months	<month>	<date>	*Plain*	core
month of <day of year>	months	<month>	<day of year>	*Plain*	core
month of <month and year>	months	<month>	<month and year>	*Plain*	core
month_and_year of <date>	months_and_years	<month and year>	<date>	*Plain*	core
more significance <integer> of <floating point>	more significances	<floating point>	<floating point>	*Numbered*	core
most significant one bit of <bit set>	most significant one bits	<integer>	<bit set>	*Plain*	core
multiplicity of <date with multiplicity>	multiplicities	<integer>	<date with multiplicity>	*Plain*	core
multiplicity of <day of month with multiplicity>	multiplicities	<integer>	<day of month with multiplicity>	*Plain*	core
multiplicity of <day of week with multiplicity>	multiplicities	<integer>	<day of week with multiplicity>	*Plain*	core
multiplicity of <day of year with multiplicity>	multiplicities	<integer>	<day of year with multiplicity>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
multiplicity of <floating point with multiplicity>	multiplicities	<integer>	<floating point with multiplicity>	*Plain*	core
multiplicity of <hertz with multiplicity>	multiplicities	<integer>	<hertz with multiplicity>	*Plain*	core
multiplicity of <integer with multiplicity>	multiplicities	<integer>	<integer with multiplicity>	*Plain*	core
multiplicity of <ipv4 address with multiplicity>	multiplicities	<integer>	<ipv4 address with multiplicity>	*Plain*	core
multiplicity of <ipv4or6 address with multiplicity>	multiplicities	<integer>	<ipv4or6 address with multiplicity>	*Plain*	core
multiplicity of <ipv6 address with multiplicity>	multiplicities	<integer>	<ipv6 address with multiplicity>	*Plain*	core
multiplicity of <month and year with multiplicity>	multiplicities	<integer>	<month and year with multiplicity>	*Plain*	core
multiplicity of <month with multiplicity>	multiplicities	<integer>	<month with multiplicity>	*Plain*	core
multiplicity of <number of months with multiplicity>	multiplicities	<integer>	<number of months with multiplicity>	*Plain*	core
multiplicity of <site version list with multiplicity>	multiplicities	<integer>	<site version list with multiplicity>	*Plain*	core
multiplicity of <string with multiplicity>	multiplicities	<integer>	<string with multiplicity>	*Plain*	core
multiplicity of <time interval with multiplicity>	multiplicities	<integer>	<time interval with multiplicity>	*Plain*	core
multiplicity of <time of day with multiplicity>	multiplicities	<integer>	<time of day with multiplicity>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
multiplicity of <time of day with time zone with multiplicity>	multiplicities	<integer>	<time of day with time zone with multiplicity>	*Plain*	core
multiplicity of <time range with multiplicity>	multiplicities	<integer>	<time range with multiplicity>	*Plain*	core
multiplicity of <time with multiplicity>	multiplicities	<integer>	<time with multiplicity>	*Plain*	core
multiplicity of <time zone with multiplicity>	multiplicities	<integer>	<time zone with multiplicity>	*Plain*	core
multiplicity of <version with multiplicity>	multiplicities	<integer>	<version with multiplicity>	*Plain*	core
multiplicity of <year with multiplicity>	multiplicities	<integer>	<year with multiplicity>	*Plain*	core
multivalued of <property>	multivalueds	<boolean>	<property>	*Plain*	core
name of <binary operator>	names	<string>	<binary operator>	*Plain*	core
name of <cast>	names	<string>	<cast>	*Plain*	core
name of <type>	names	<string>	<type>	*Plain*	core
name of <unary operator>	names	<string>	<unary operator>	*Plain*	core
nan of <floating point>	nans	<boolean>	<floating point>	*Plain*	core
next sibling of <xml dom node>	next siblings	<xml dom node>	<xml dom node>	*Plain*	core
node name of <xml dom node>	node names	<string>	<xml dom node>	*Plain*	core
node type of <xml dom node>	node types	<integer>	<xml dom node>	*Plain*	core
node value of <xml dom node>	node values	<string>	<xml dom node>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
noon	noons	<time of day>	<world>	*PlainGlobal*	core
normal of <floating point>	normals	<boolean>	<floating point>	*Plain*	core
november	novembers	<month>	<world>	*PlainGlobal*	core
november <integer>	novembers	<day of year>	<world>	*NumberedGlobal*	core
november <integer> of <integer>	novembers	<date>	<integer>	*Numbered*	core
november of <integer>	novembers	<month and year>	<integer>	*Plain*	core
now	nows	<time>	<world>	*PlainGlobal*	core
numeric value of <string>	numeric values	<integer>	<string>	*Plain*	core
october	octobers	<month>	<world>	*PlainGlobal*	core
october <integer>	octobers	<day of year>	<world>	*NumberedGlobal*	core
october <integer> of <integer>	octobers	<date>	<integer>	*Numbered*	core
october of <integer>	octobers	<month and year>	<integer>	*Plain*	core
ol <string> of <html>	ols	<html>	<html>	*Named*	core
ol <string> of <string>	ols	<html>	<string>	*Named*	core
ol of <html>	ols	<html>	<html>	*Plain*	core
ol of <string>	ols	<html>	<string>	*Plain*	core
one bit of <bit set>	one bits	<integer>	<bit set>	*Plain*	core
operand type of <cast>	operand types	<type>	<cast>	*Plain*	core
operand type of <unary operator>	operand types	<type>	<unary operator>	*Plain*	core
ordered list <string> of <html>	ordered lists	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
ordered list <string> of <string>	ordered lists	<html>	<string>	*Named*	core
ordered list of <html>	ordered lists	<html>	<html>	*Plain*	core
ordered list of <string>	ordered lists	<html>	<string>	*Plain*	core
overflow of <floating point>	overflows	<boolean>	<floating point>	*Plain*	core
owner document of <xml dom node>	owner documents	<xml dom document>	<xml dom node>	*Plain*	core
p <string> of <html>	ps	<html>	<html>	*Named*	core
p <string> of <string>	ps	<html>	<string>	*Named*	core
p of <html>	ps	<html>	<html>	*Plain*	core
p of <string>	ps	<html>	<string>	*Plain*	core
pad of <version>	pads	<version>	<version>	*Plain*	core
parent node of <xml dom node>	parent nodes	<xml dom node>	<xml dom node>	*Plain*	core
parent of <type>	parents	<type>	<type>	*Plain*	core
parenthesized part <integer> of <regular expression match>	parenthesized parts	<substring>	<regular expression match>	*Numbered*	regx
parenthesized part of <regular expression match>	parenthesized parts	<substring>	<regular expression match>	*Plain*	regx
plural name of <property>	plural names	<string>	<property>	*Plain*	core
position <integer> of <string>	positions	<string position>	<string>	*Numbered*	core
position of <string>	positions	<string position>	<string>	*Plain*	core
pre <string> of <html>	pres	<html>	<html>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
pre <string> of <string>	pres	<html>	<string>	*Named*	core
pre of <html>	pres	<html>	<html>	*Plain*	core
pre of <string>	pres	<html>	<string>	*Plain*	core
preceding text of <string position>	preceding texts	<substring>	<string position>	*Plain*	core
preceding text of <substring>	preceding texts	<substring>	<substring>	*Plain*	core
previous sibling of <xml dom node>	previous siblings	<xml dom node>	<xml dom node>	*Plain*	core
product of <floating point>	products	<floating point>	<floating point>	*Plain*	core
product of <integer>	products	<integer>	<integer>	*Plain*	core
property <string>	properties	<property>	<world>	*NamedGlobal*	core
property <string> of <type>	properties	<property>	<type>	*Named*	core
property of <type>	properties	<property>	<type>	*Plain*	core
property returning <type>	properties returning	<property>	<world>	*Index<type>Global*	core
property returning <type> of <type>	properties returning	<property>	<type>	*Index<type>*	core
q <string> of <html>	qs	<html>	<html>	*Named*	core
q <string> of <string>	qs	<html>	<string>	*Named*	core
q of <html>	qs	<html>	<html>	*Plain*	core
q of <string>	qs	<html>	<string>	*Plain*	core
range after <time> of <time range>	ranges after	<time range>	<time range>	*Index<time>*	core
range before <time> of <time range>	ranges before	<time range>	<time range>	*Index<time>*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
regex <string>	regexes	<regular expression>	<world>	*NamedGlobal*	regx
regex escape of <string>	regex escapes	<string>	<string>	*Plain*	regx
regular expression <string>	regular expressions	<regular expression>	<world>	*NamedGlobal*	regx
relative significance place <integer> of <floating point>	relative significance places	<floating point>	<floating point>	*Numbered*	core
relative significance place of <floating point>	relative significance places	<floating point>	<floating point>	*Plain*	core
result type of <binary operator>	result types	<type>	<binary operator>	*Plain*	core
result type of <cast>	result types	<type>	<cast>	*Plain*	core
result type of <property>	result types	<type>	<property>	*Plain*	core
result type of <unary operator>	result types	<type>	<unary operator>	*Plain*	core
right operand type of <binary operator>	right operand types	<type>	<binary operator>	*Plain*	core
right shift <integer> of <bit set>	right shifts	<bit set>	<bit set>	*Numbered*	core
rope <string>	ropes	<rope>	<world>	*NamedGlobal*	core
samp <string> of <html>	samps	<html>	<html>	*Named*	core
samp <string> of <string>	samps	<html>	<string>	*Named*	core
samp of <html>	samps	<html>	<html>	*Plain*	core
samp of <string>	samps	<html>	<string>	*Plain*	core
saturday	saturdays	<day of week>	<world>	*PlainGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
second	seconds	<time interval>	<world>	*PlainGlobal*	core
second_of_minute of <time of day with time zone>	seconds_of_minute	<integer>	<time of day with time zone>	*Plain*	core
second_of_minute of <time of day>	seconds_of_minute	<integer>	<time of day>	*Plain*	core
select <string> of <xml dom node>	selects	<xml dom node>	<xml dom node>	*Named*	core
september	septembers	<month>	<world>	*PlainGlobal*	core
september <integer>	septembers	<day of year>	<world>	*NumberedGlobal*	core
september <integer> of <integer>	septembers	<date>	<integer>	*Numbered*	core
september of <integer>	septembers	<month and year>	<integer>	*Plain*	core
set of <integer>	sets	<integer set>	<integer>	*Plain*	core
set of <string>	sets	<string set>	<string>	*Plain*	core
sha1 of <x509 certificate>	sha1s	<string>	<x509 certificate>	*Plain*	core
significance place <integer> of <floating point>	significance places	<floating point>	<floating point>	*Numbered*	core
significance place of <floating point>	significance places	<floating point>	<floating point>	*Plain*	core
significance threshold of <floating point>	significance thresholds	<floating point>	<floating point>	*Plain*	core
significant digits <integer> of <hertz>	significant digitss	<hertz>	<hertz>	*Numbered*	core
significant digits <integer> of <integer>	significant digitss	<integer>	<integer>	*Numbered*	core
singular name of <property>	singular names	<string>	<property>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
site version list <string>	site version lists	<site version list>	<world>	*NamedGlobal*	core
size of <integer set>	sizes	<integer>	<integer set>	*Plain*	core
size of <string set>	sizes	<integer>	<string set>	*Plain*	core
size of <type>	sizes	<integer>	<type>	*Plain*	core
small <string> of <html>	smalls	<html>	<html>	*Named*	core
small <string> of <string>	smalls	<html>	<string>	*Named*	core
small of <html>	smalls	<html>	<html>	*Plain*	core
small of <string>	smalls	<html>	<string>	*Plain*	core
span <string> of <html>	spans	<html>	<html>	*Named*	core
span <string> of <string>	spans	<html>	<string>	*Named*	core
span of <html>	spans	<html>	<html>	*Plain*	core
span of <string>	spans	<html>	<string>	*Plain*	core
standard deviation of <floating point>	standard deviations	<floating point>	<floating point>	*Plain*	core
standard deviation of <integer>	standard deviations	<floating point>	<integer>	*Plain*	core
start of <substring>	starts	<string position>	<substring>	*Plain*	core
start of <time range>	starts	<time>	<time range>	*Plain*	core
string <string>	strings	<string>	<world>	*NamedGlobal*	core
strong <string> of <html>	strongs	<html>	<html>	*Named*	core
strong <string> of <string>	strongs	<html>	<string>	*Named*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
strong of <html>	strongs	<html>	<html>	*Plain*	core
strong of <string>	strongs	<html>	<string>	*Plain*	core
sub <string> of <html>	subs	<html>	<html>	*Named*	core
sub <string> of <string>	subs	<html>	<string>	*Named*	core
sub of <html>	subs	<html>	<html>	*Plain*	core
sub of <string>	subs	<html>	<string>	*Plain*	core
substring <(integer, integer)> of <string>	substrings	<substring>	<string>	*Index<(integer, integer)>*	core
substring <string> of <string>	substrings	<substring>	<string>	*Named*	core
substring after <string> of <string>	substrings after	<substring>	<string>	*Named*	core
substring before <string> of <string>	substrings before	<substring>	<string>	*Named*	core
substring between <string> of <string>	substrings between	<substring>	<string>	*Named*	core
substring separated by <string> of <string>	substrings separated by	<substring>	<string>	*Named*	core
sum of <floating point>	sums	<floating point>	<floating point>	*Plain*	core
sum of <integer>	sums	<integer>	<integer>	*Plain*	core
sum of <time interval>	sums	<time interval>	<time interval>	*Plain*	core
sunday	sundays	<day of week>	<world>	*PlainGlobal*	core
sup <string> of <html>	sups	<html>	<html>	*Named*	core
sup <string> of <string>	sups	<html>	<string>	*Named*	core
sup of <html>	sups	<html>	<html>	*Plain*	core
sup of <string>	sups	<html>	<string>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
symbol of <binary operator>	symbols	<string>	<binary operator>	*Plain*	core
symbol of <unary operator>	symbols	<string>	<unary operator>	*Plain*	core
table <string> of <html>	tables	<html>	<html>	*Named*	core
table <string> of <string>	tables	<html>	<string>	*Named*	core
table of <html>	tables	<html>	<html>	*Plain*	core
table of <string>	tables	<html>	<string>	*Plain*	core
tbody <string> of <html>	tbodys	<html>	<html>	*Named*	core
tbody <string> of <string>	tbodys	<html>	<string>	*Named*	core
tbody of <html>	tbodys	<html>	<html>	*Plain*	core
tbody of <string>	tbodys	<html>	<string>	*Plain*	core
td <string> of <html>	tds	<html>	<html>	*Named*	core
td <string> of <string>	tds	<html>	<string>	*Named*	core
td of <html>	tds	<html>	<html>	*Plain*	core
td of <string>	tds	<html>	<string>	*Plain*	core
tfoot <string> of <html>	tfoots	<html>	<html>	*Named*	core
tfoot <string> of <string>	tfoots	<html>	<string>	*Named*	core
tfoot of <html>	tfoots	<html>	<html>	*Plain*	core
tfoot of <string>	tfoots	<html>	<string>	*Plain*	core
th <string> of <html>	ths	<html>	<html>	*Named*	core
th <string> of <string>	ths	<html>	<string>	*Named*	core

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

Key Phrase	Plural	Creates a	From a	*Form*	Ref
true	trues	<boolean>	<world>	*PlainGlobal*	core
tt <string> of <html>	tts	<html>	<html>	*Named*	core
tt <string> of <string>	tts	<html>	<string>	*Named*	core
tt of <html>	tts	<html>	<html>	*Plain*	core
tt of <string>	tts	<html>	<string>	*Plain*	core
tuesday	tuesdays	<day of week>	<world>	*PlainGlobal*	core
tuple string item <integer> of <string>	tuple string items	<string>	<string>	*Numbered*	core
tuple string item of <string>	tuple string items	<string>	<string>	*Plain*	core
two digit hour of <time of day with time zone>	two digit hours	<string>	<time of day with time zone>	*Plain*	core
two digit hour of <time of day>	two digit hours	<string>	<time of day>	*Plain*	core
two digit minute of <time of day with time zone>	two digit minutes	<string>	<time of day with time zone>	*Plain*	core
two digit minute of <time of day>	two digit minutes	<string>	<time of day>	*Plain*	core
two digit second of <time of day with time zone>	two digit seconds	<string>	<time of day with time zone>	*Plain*	core
two digit second of <time of day>	two digit seconds	<string>	<time of day>	*Plain*	core
type of <distinguished name component>	types	<string>	<distinguished name component>	*Plain*	core
ul <string> of <html>	uls	<html>	<html>	*Named*	core
ul <string> of <string>	uls	<html>	<string>	*Named*	core
ul of <html>	uls	<html>	<html>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
ul of <string>	uls	<html>	<string>	*Plain*	core
unary operator <string>	unary operators	<unary operator>	<world>	*NamedGlobal*	core
unary operator returning <type>	unary operators returning	<unary operator>	<world>	*Index<type>Global*	core
underflow of <floating point>	underflows	<boolean>	<floating point>	*Plain*	core
union of <integer set>	unions	<integer set>	<integer set>	*Plain*	core
union of <string set>	unions	<string set>	<string set>	*Plain*	core
unique value of <date>	unique values	<date with multiplicity>	<date>	*Plain*	core
unique value of <day of month>	unique values	<day of month with multiplicity>	<day of month>	*Plain*	core
unique value of <day of week>	unique values	<day of week with multiplicity>	<day of week>	*Plain*	core
unique value of <day of year>	unique values	<day of year with multiplicity>	<day of year>	*Plain*	core
unique value of <floating point>	unique values	<floating point with multiplicity>	<floating point>	*Plain*	core
unique value of <hertz>	unique values	<hertz with multiplicity>	<hertz>	*Plain*	core
unique value of <integer>	unique values	<integer with multiplicity>	<integer>	*Plain*	core
unique value of <ipv4 address>	unique values	<ipv4 address with multiplicity>	<ipv4 address>	*Plain*	core
unique value of <ipv4or6 address>	unique values	<ipv4or6 address with multiplicity>	<ipv4or6 address>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unique value of <ipv6 address>	unique values	<ipv6 address with multiplicity>	<ipv6 address>	*Plain*	core
unique value of <month and year>	unique values	<month and year with multiplicity>	<month and year>	*Plain*	core
unique value of <month>	unique values	<month with multiplicity>	<month>	*Plain*	core
unique value of <number of months>	unique values	<number of months with multiplicity>	<number of months>	*Plain*	core
unique value of <site version list>	unique values	<site version list with multiplicity>	<site version list>	*Plain*	core
unique value of <string>	unique values	<string with multiplicity>	<string>	*Plain*	core
unique value of <time interval>	unique values	<time interval with multiplicity>	<time interval>	*Plain*	core
unique value of <time of day with time zone>	unique values	<time of day with time zone with multiplicity>	<time of day with time zone>	*Plain*	core
unique value of <time of day>	unique values	<time of day with multiplicity>	<time of day>	*Plain*	core
unique value of <time range>	unique values	<time range with multiplicity>	<time range>	*Plain*	core
unique value of <time zone>	unique values	<time zone with multiplicity>	<time zone>	*Plain*	core
unique value of <time>	unique values	<time with multiplicity>	<time>	*Plain*	core
unique value of <version>	unique values	<version with multiplicity>	<version>	*Plain*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
unique value of <year>	unique values	<year with multiplicity>	<year>	*Plain*	core
universal time <string>	universal times	<time>	<world>	*NamedGlobal*	core
universal time zone	universal time zones	<time zone>	<world>	*PlainGlobal*	core
unordered list <string> of <html>	unordered lists	<html>	<html>	*Named*	core
unordered list <string> of <string>	unordered lists	<html>	<string>	*Named*	core
unordered list of <html>	unordered lists	<html>	<html>	*Plain*	core
unordered list of <string>	unordered lists	<html>	<string>	*Plain*	core
usual name of <property>	usual names	<string>	<property>	*Plain*	core
value of <distinguished name component>	values	<string>	<distinguished name component>	*Plain*	core
var <string> of <html>	vars	<html>	<html>	*Named*	core
var <string> of <string>	vars	<html>	<string>	*Named*	core
var of <html>	vars	<html>	<html>	*Plain*	core
var of <string>	vars	<html>	<string>	*Plain*	core
version <string>	versions	<version>	<world>	*NamedGlobal*	core
version string <string> of <module>	version strings	<string>	<module>	*Named*	core
wednesday	wednesdays	<day of week>	<world>	*PlainGlobal*	core
week	weeks	<time interval>	<world>	*PlainGlobal*	core
windows display time <string>	windows display times	<time>	<world>	*NamedGlobal*	core

Key Phrase	Plural	Creates a	From a	*Form*	Ref
xml document of <string>	xml documents	<xml dom document>	<string>	*Plain*	core
xpath <(string, string)> of <xml dom node>	xpaths	<xml dom node>	<xml dom node>	*Index<(string, string)>*	core
xpath <string> of <xml dom node>	xpaths	<xml dom node>	<xml dom node>	*Named*	core
year	years	<number of months>	<world>	*PlainGlobal*	core
year <integer>	years	<year>	<world>	*NumberedGlobal*	core
year <string>	years	<year>	<world>	*NamedGlobal*	core
year of <date>	years	<year>	<date>	*Plain*	core
year of <month and year>	years	<year>	<month and year>	*Plain*	core
zone of <time of day with time zone>	zones	<time zone>	<time of day with time zone>	*Plain*	core
zoned time_of_day <string>	zoned times_of_day	<time of day with time zone>	<world>	*NamedGlobal*	core

Casting Operators

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

Key Phrase	Creates a	From a
<binary operator> as string	<string>	<binary operator>
<bit set> as integer	<integer>	<bit set>
<bit set> as string	<string>	<bit set>
<boolean> as boolean	<boolean>	<boolean>
<boolean> as string	<string>	<boolean>
<cast> as string	<string>	<cast>
<date> as string	<string>	<date>
<day of month> as integer	<integer>	<day of month>
<day of month> as string	<string>	<day of month>
<day of month> as two digits	<string>	<day of month>
<day of week> as string	<string>	<day of week>
<day of week> as three letters	<string>	<day of week>
<day of year> as string	<string>	<day of year>
<floating point> as floating point	<floating point>	<floating point>
<floating point> as integer	<integer>	<floating point>
<floating point> as scientific notation	<string>	<floating point>
<floating point> as standard notation	<string>	<floating point>
<floating point> as string	<string>	<floating point>
<format> as string	<string>	<format>
<hertz> as string	<string>	<hertz>
<html> as html	<html>	<html>

Key Phrase	Creates a	From a
<html> as string	<string>	<html>
<integer> as bit set	<bit set>	<integer>
<integer> as bits	<bit set>	<integer>
<integer> as day_of_month	<day of month>	<integer>
<integer> as floating point	<floating point>	<integer>
<integer> as hexadecimal	<string>	<integer>
<integer> as integer	<integer>	<integer>
<integer> as month	<month>	<integer>
<integer> as string	<string>	<integer>
<integer> as year	<year>	<integer>
<ip version> as string	<string>	<ip version>
<ipv4 address> as ipv4or6 address	<ipv4or6 address>	<ipv4 address>
<ipv4 address> as ipv6 address	<ipv6 address>	<ipv4 address>
<ipv4 address> as string	<string>	<ipv4 address>
<ipv4or6 address> as compressed string	<string>	<ipv4or6 address>
<ipv4or6 address> as compressed string with ipv4	<string>	<ipv4or6 address>
<ipv4or6 address> as compressed string with ipv4 with zone index	<string>	<ipv4or6 address>
<ipv4or6 address> as compressed string with zone index	<string>	<ipv4or6 address>
<ipv4or6 address> as ipv6 address	<ipv4or6 address>	<ipv4or6 address>
<ipv4or6 address> as string	<string>	<ipv4or6 address>
<ipv4or6 address> as string with ipv4	<string>	<ipv4or6 address>
<ipv4or6 address> as string with ipv4 with zone index	<string>	<ipv4or6 address>

Key Phrase	Creates a	From a
<ipv4or6 address> as string with leading zeros	<string>	<ipv4or6 address>
<ipv4or6 address> as string with leading zeros with zone index	<string>	<ipv4or6 address>
<ipv4or6 address> as string with zone index	<string>	<ipv4or6 address>
<ipv6 address> as compressed string	<string>	<ipv6 address>
<ipv6 address> as compressed string with ipv4	<string>	<ipv6 address>
<ipv6 address> as compressed string with ipv4 with zone index	<string>	<ipv6 address>
<ipv6 address> as compressed string with zone index	<string>	<ipv6 address>
<ipv6 address> as ipv4or6 address	<ipv4or6 address>	<ipv6 address>
<ipv6 address> as string	<string>	<ipv6 address>
<ipv6 address> as string with ipv4	<string>	<ipv6 address>
<ipv6 address> as string with ipv4 with zone index	<string>	<ipv6 address>
<ipv6 address> as string with leading zeros	<string>	<ipv6 address>
<ipv6 address> as string with leading zeros with zone index	<string>	<ipv6 address>
<ipv6 address> as string with zone index	<string>	<ipv6 address>
<month and year> as string	<string>	<month and year>
<month> as integer	<integer>	<month>
<month> as string	<string>	<month>
<month> as three letters	<string>	<month>
<month> as two digits	<string>	<month>
<number of months> as string	<string>	<number of months>

Key Phrase	Creates a	From a
<property> as string	<string>	<property>
<rope> as string	<string>	<rope>
<site version list> as string	<string>	<site version list>
<string> as boolean	<boolean>	<string>
<string> as date	<date>	<string>
<string> as day_of_month	<day of month>	<string>
<string> as day_of_week	<day of week>	<string>
<string> as floating point	<floating point>	<string>
<string> as hexadecimal	<string>	<string>
<string> as html	<html>	<string>
<string> as integer	<integer>	<string>
<string> as ipv4or6 address	<ipv4or6 address>	<string>
<string> as ipv6 address	<ipv4or6 address>	<string>
<string> as left trimmed string	<string>	<string>
<string> as local time	<time>	<string>
<string> as local zoned time_of_day	<time of day with time zone>	<string>
<string> as lowercase	<string>	<string>
<string> as month	<month>	<string>
<string> as right trimmed string	<string>	<string>
<string> as site version list	<site version list>	<string>
<string> as string	<string>	<string>
<string> as time	<time>	<string>
<string> as time interval	<time interval>	<string>

Key Phrase	Creates a	From a
<string> as time zone	<time zone>	<string>
<string> as time_of_day	<time of day>	<string>
<string> as trimmed string	<string>	<string>
<string> as universal time	<time>	<string>
<string> as universal zoned time_of_day	<time of day with time zone>	<string>
<string> as uppercase	<string>	<string>
<string> as version	<version>	<string>
<string> as windows display time	<time>	<string>
<string> as year	<year>	<string>
<string> as zoned time_of_day	<time of day with time zone>	<string>
<time interval> as string	<string>	<time interval>
<time of day with time zone> as string	<string>	<time of day with time zone>
<time of day> as string	<string>	<time of day>
<time range> as string	<string>	<time range>
<time zone> as string	<string>	<time zone>
<time> as local string	<string>	<time>
<time> as string	<string>	<time>
<time> as universal string	<string>	<time>
<type> as string	<string>	<type>
<unary operator> as string	<string>	<unary operator>
<undefined> as string	<string>	<undefined>
<version> as string	<string>	<version>
<version> as version	<version>	<version>

Key Phrase	Creates a	From a
<xml dom node> as text	<string>	<xml dom node>
<xml dom node> as xml	<string>	<xml dom node>
<year> as integer	<integer>	<year>
<year> as string	<string>	<year>

Part Three

Notices

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

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

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

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

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

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

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

provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Corporation

2Z4A/101

11400 Burnet Road

Austin, TX 78758 U.S.A.

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

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

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

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

COPYRIGHT LICENSE:

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

TRADEMARKS:

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

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

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

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

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

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

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

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

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

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z