Windows Inspector Library

A Guide to the BigFix Windows Inspectors

BigFix, Inc.
Emeryville, CA

Last Modified: August 18, 2007
Compatible with
BES 7.0

© 1998–2007 BigFix, Inc. All rights reserved.

BigFix®, Fixlet® and "Fix it before it fails"® are registered trademarks of BigFix, Inc. iprevention, Powered by BigFix, Relevance Engine, and related BigFix logos are trademarks of BigFix, Inc. All other product names, trade names, trademarks, and logos used in this documentation are the property of their respective owners. BigFix's use of any other company's trademarks, trade names, product names and logos or images of the same does not necessarily constitute: (1) an endorsement by such company of BigFix and its products, and (2) an endorsement of the company or its products by BigFix.

Except as set forth in the last sentence of this paragraph: (1) no part of this documentation may be reproduced, transmitted, or otherwise distributed in any form or by any means (electronic or otherwise) without the prior written consent of BigFix, Inc., and (2) you may not use this documentation for any purpose except in connection with your properly licensed use or evaluation of BigFix software and any other use, including for reverse engineering such software or creating derivative works thereof, is prohibited. If the license to the software which this documentation accompanies is terminated, you must immediately return this documentation to BigFix, Inc. and destroy all copies you may have. You may treat only those portions of this documentation specifically designated in the "Acknowledgements and Notices" section below as notices applicable to third party software in accordance with the terms of such notices.

All inquiries regarding the foregoing should be addressed to:
BigFix, Inc.
1480 64th Street, Suite 200
Emeryville, CA 94608

Contents

PREFACE 1

Audience ... 1
Organization of this manual ... 1
Conventions Used in this manual ... 3
Examples ... 3
Versions ... 4

INTRODUCTION 5

PRIMITIVE OBJECTS 6

Boolean ... 6
Integer ... 10
Integer Range ... 16
Integer With Multiplicity ... 17
Integer Set ... 18
Floating Point ... 20
String ... 27
String Position ... 37
Substring ... 38
String With Multiplicity ... 41
String Set ... 42
Rope ... 44
Bit Set ... 45
Regular Expression ... 47
Regular Expression Match ... 48
Undefined ... 49
Hertz ... 50
Time ... 52
Time Of Day ... 56
Time Zone ... 58
Time Of Day With Time Zone ... 59
Time Range ... 62
Time Interval ... 64
Date ... 68
Day Of Week ... 71
Day Of Month ... 73
Day Of Year ... 76
Month ... 79
Month And Year ... 83
Number Of Months ... 87
Year ... 88

WORLD OBJECTS 91

World ... 91

© 2007 by BigFix, Inc.

REGISTRY OBJECTS 122

REGISTRY 122
REGISTRY KEY 125
REGISTRY KEY VALUE 130
REGISTRY KEY VALUE TYPE 133

FILESYSTEM OBJECTS 135

FILESYSTEM OBJECT 135
FILE 137
APPLICATION 144
FOLDER 147
DRIVE 152
FILE SHORTCUT 155
FILE SECTION 156
FILE CONTENT 157
VERSION 158
FILE VERSION BLOCK 161
FILE LINE 166
XML DOM DOCUMENT 168
XML DOM NODE 169
APPLICATION USAGE SUMMARY 174

SYSTEM OBJECTS 176

BIOS 176
OPERATING SYSTEM 177
PROCESSOR 181
RAM 184
ACTIVE DEVICE 185
LICENSE 187
LOCAL MSSQL DATABASE 188
SERVICE 189
LANGUAGE 192
PRIMARY LANGUAGE 193

© 2007 by BigFix, Inc.

FIREWALL OBJECTS 194

FIREWALL 194
FIREWALL AUTHORIZED APPLICATION 196
FIREWALL PROFILE 197
FIREWALL PROFILE TYPE 200
FIREWALL POLICY 202
FIREWALL SCOPE 203
FIREWALL OPEN PORT 204
FIREWALL SERVICE 206
FIREWALL SERVICE TYPE 207
FIREWALL ICMP SETTINGS 209
FIREWALL REMOTE ADMIN SETTINGS 210
FIREWALL ACTION 211
FIREWALL LOCAL POLICY MODIFY STATE 212
FIREWALL RULE 213
FIREWALL SERVICE RESTRICTION 217
INTERNET PROTOCOL 218
IP VERSION 219

WMI OBJECTS 221

WMI 221
WMI SELECT 222
WMI OBJECT 225

SITE OBJECTS 226

SITE 226
SITE GROUP 229
SITE VERSION LIST 229
FIXLET 230
FIXLET_HEADER 231

CLIENT OBJECTS 233

CLIENT 233
SETTING 234
SELECTED SERVER 235
OPERATING SYSTEM PRODUCT TYPE 237
LOCAL GROUP 238
LOCAL GROUP MEMBER 239
EVENT LOG 240
EVENT LOG RECORD 242
EVENT LOG EVENT TYPE 245
CURRENT RELAY 246
ROOT SERVER 247

ENVIRONMENT OBJECTS 248

ENVIRONMENT 248
ENVIRONMENT VARIABLE 249

© 2007 by BigFix, Inc.

AUTHORIZATION OBJECTS 250

ACCESS CONTROL ENTRY 250
ACCESS CONTROL LIST 254
SECURITY DESCRIPTOR 259
SECURITY IDENTIFIER 261

USER OBJECTS 263

LOCAL USER 263
CURRENT USER 267
LOGGED ON USER 268

ACTION OBJECTS 270

ACTION 270

NETWORKING OBJECTS 273

NETWORK 273
NETWORK INTERFACE 274
NETWORK IP INTERFACE 275
NETWORK SHARE 276
NETWORK ADAPTER 278
NETWORK ADDRESS LIST 282
INTERNET CONNECTION FIREWALL 283
PORT MAPPING 283
CONNECTION 285
CONNECTION STATUS 286
MEDIA TYPE 289
IPV4 ADDRESS 291
IPV6 ADDRESS 294
ACTIVE DIRECTORY LOCAL COMPUTER 296
ACTIVE DIRECTORY SERVER 297
DISTINGUISHED NAME 297
DISTINGUISHED NAME COMPONENT 298

MICROSOFT IIS METABASE OBJECTS 300

METABASE 300
METABASE IDENTIFIER 301
METABASE KEY 302
METABASE TYPE 303
METABASE USER TYPE 304
METABASE VALUE 305

INTROSPECTORS 307

TYPE 307
PROPERTY 309
BINARY OPERATOR 311
UNARY OPERATOR 312
CAST 314

© 2007 by BigFix, Inc.

KEY PHRASES (INSPECTORS) 316

KEY PHRASES 316
CASTING OPERATORS 364

APPENDIX 370

FOLDERS ON WINDOWS MACHINES 370
PROCESSORS 371

INDEX 373

© 2007 by BigFix, Inc.

Preface

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

Audience

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

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

QA and other support teams will produce Fixlet messages for the BigFix Consumer Edition (BCE), designed to keep their users updated and their support calls to a minimum. Only those Inspectors marked with BCE are available for the consumer version. To get the most out of this manual, it helps to have some experience with the Windows Registry and the BigFix Relevance Language.

Organization of this manual

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

- **Primitive Objects.** This chapter covers the basic data types supported by the language and describes the operations that can be applied to them.
- **World Objects.** This chapter covers the keywords used to create all the 'top' level objects of the world. The properties of these objects provide access to all levels of the machine state that can be inspected.
- **Registry Objects.** This chapter covers the keywords for dealing with the Windows registry. Particular attention is paid to registered applications and their associated file extensions.
- **File System Objects.** This chapter covers the keywords for extracting information from the file system, like applications, drives, pathnames, folders, versions, etc. It includes the keywords dealing with applications that have registered themselves in the Windows registry. It also includes the keywords needed to identify and compare version information of files and applications.

© 2007 by BigFix, Inc.

- **System Objects.** This chapter covers the keywords available for querying the name and version of the operating sytem. It also includes the version information of the system Bios. This chapter also covers the keywords used to describe the vendors and types of the various processors that coexist in a typical computer system.
- **Firewall Objects.** This chapter details the firewall Inspectors that examine the authorized applications, policies, services, settings and more.
- **WMI Objects.** This chapter covers WMI objects that provide access to the WMI (Windows Management Interface) query facility.
- **Site Objects.** This chapter covers the keywords that query the properties of Fixlet sites to which the client is subscribed.
- **Client Objects.** This chapter covers the client inspectors, which allow access to properties of the client application hosting the relevance evaluation.
- **Environment Objects.** An environment object is provided to access environment variables. These are the same variables you are used to seeing in a DOS shell when you type the 'set' command. Note that you are inspecting the environment of the application executing the relevance clause, which may or may not match the environment of other applications on the computer.
- **Authorization Objects.** This section covers inspectors that retrieve security and access settings.
- **User Objects.** This chapter covers the local and current user keywords. A Local User object is provided to access the user data of the local machine. Note that domain users are not available through this inspector.
- **Action Objects.** These are the keywords associated with properties available for inspection during the execution of BigFix Actions.
- **Network Objects.** This chapter covers the keywords used to query the local network configuration.
- **Microsoft IIS Metabase Objects.** This section lists the inspectors for the Microsoft IIS Metabase, which is a repository for most IIS configuration values.
- **Introspectors.** This chapter is concerned with Inspectors that query the Inspectors themselves, looking at types, properties, operators and casts.
- **Keywords (Inspector List).** This chapter provides an alphabetical list of all the Inspector keywords along with the form, context object type, and resulting object type.

© 2007 by BigFix, Inc.

Conventions Used in this manual

This document makes use of the following conventions and nomenclature:

Convention	Use
Bold Sans	A bold sans-serif font is used for Inspector headers.
`Mono-space`	A mono-spaced font is used to indicate expressions in the Relevance Language.
{curly braces}	Braces are used to indicate the comparison {=, !=}or arithmetic operators {+, -} that are available for a binary operation.
<angle bracket>	Angle brackets are used to indicate an object type. For instance to indicate the creation and usage of a particular object, you might see “absolute value of <integer>” which indicates that an integer is to follow the “absolute value of” keyphrase.
Italics	An inspector form. Some inspectors are simple keywords. Others are a keyword in combination with another inspector. Still other forms allow iteration through object lists. Each form is defined below
Small print	The small print beneath the description of each Inspector lists the first implementation for every relevant operating system.

Examples

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

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

© 2007 by BigFix, Inc.

Versions

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

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

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

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

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

Sol: the SUN Solaris operating system version of BES.

HPUX: the Hewlett-Packard Unix version of BES.

AIX: the AIX version of BES.

Mac: the Macintosh version of BES.

© 2007 by BigFix, Inc.

Introduction

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

You will notice that many of the keywords of the language are not unique; they get their meaning from their context. Accordingly, their definitions often include a phrase to define the context of each Inspector.

This document describes inspectors for Windows 95/98/ME as well as Windows NT, 2000, XP and Vista. Only those Inspectors marked with BCE are available for the Bigfix Consumer Edition. Contact your BigFix sales representative for information about Inspector Guides for other operating systems, including Solaris, Mac, HPUX, AIX, Red Hat and Suse Linux.

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

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

These differ from one another in format and in the syntax they require. Except for Cast, these forms can be used to access both single objects and *lists* of objects by using the plural form of the keyword. The plurals are listed in the Keyword section later in this document.

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

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

© 2007 by BigFix, Inc.

Primitive Objects

The relevance language is based upon a comprehensive set of primitive objects. These primitives are the basic building blocks of the more complex objects to follow.

Boolean

Creation Methods

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

Key Phrase	*Form*	Description
<string> as boolean	*Cast*	Returns a boolean TRUE or FALSE from a string. The string must contain values of "TRUE" or "FALSE". Case is ignored. For example, "FalSe" as boolean = FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
bit <integer> of <integer>	*Numbered*	Return TRUE if the bit referenced by the integer is on. Bits are numbered starting with zero being the least significant. For example, bit 0 of 5 and bit 2 of 5 and not bit 1 of 5 = TRUE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
boolean <string>	*NamedGlobal*	Creates the boolean value of the <string>, e.g., • boolean "False" = FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
false	*PlainGlobal*	Creates a boolean with value FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

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

Properties

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

© 2007 by BigFix, Inc.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Operators

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

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<integer> {cmp} <floating point>	*<boolean>*	Compares an integer to a floating point number, where: • {cmp} is one of: =, <=, <. Win:4.1, Mac:4.1
<time interval> {cmp} <time interval>	*<boolean>*	Compare two time intervals, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time range> * <boolean>	*<timed(time range, boolean)>*	Returns a time interval labeled with the specified boolean, in the form of: • (<date> to <date>), <boolean>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Examples

- `bit 0 of (least integer + 1)`
  - Returns the least significant bit of the smallest possible integer, plus one.

- `conjunction of (current month = April; leap of year of current date)`
  - Returns TRUE during April of a leap year.

- `disjunction of (current day_of_week = Monday ;current day_of_week = Wednesday; current day_of_week = Friday)`
  - Returns TRUE on either Monday, Wednesday or Friday.

- `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.

© 2007 by BigFix, Inc.

Integer

Integers are represented internally as 64-bit signed values.

Creation Methods

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

Key Phrase	*Form*	Description
<floating point> as integer	*Cast*	Rounds off and casts a floating point number as an integer. Win:6.0, Mac:6.0
<integer> as integer	*Cast*	Integer casting for completeness. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as integer	*Cast*	Converts from a string to an integer. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
absolute value of <integer>	*Plain*	Creates the positive value of the <integer> object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
greatest integer	*PlainGlobal*	Creates the value 9,223,372,036,854,775,807. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hexadecimal integer <string>	*NamedGlobal*	Creates an integer from the provided hexadecimal value. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
integer <integer>	*NumberedGlobal*	Creates a global object with the given integer value, e.g., Integer 123. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
integer <string>	*NamedGlobal*	Creates a global object with the integer value given by a string, e.g., Integer "123" creates the value 123. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
integer ceiling of <floating point>	*Plain*	Returns the smallest integer not less than the floating point number. For example, ceiling of 2.1 = 3, ceiling of 2 = 2 and ceiling of -2.3 = -2. Win:6.0, Mac:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
integer floor of <floating point>	*Plain*	Returns the largest integer less than or equal to the floating point number. For example, floor of 2.8 = 2, floor of -2 = -2 and floor of -2.1 = -3. For nonnegative x, this is the same as the integer part of x. Win:6.0, Mac:6.0
least integer	*PlainGlobal*	Creates the value -9,223,372,036,854,775,808. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
length of <rope>	*Plain*	Creates an integer object corresponding to the number of bytes in the rope. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
length of <string>	*Plain*	Creates an integer object corresponding to the number of bytes in the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
lower bound of <integer range>	*Plain*	The low end of the integer range. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
maximum of <integer>	*Plain*	Returns the maximum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
minimum of <integer>	*Plain*	Returns the minimum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
numeric value of <string>	*Plain*	Creates an integer object containing the value of the first number contained in a string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
product of <integer>	*Plain*	Multiplies a list of integers, returning the product. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
significant digits <integer> of <integer>	*Numbered*	Creates a number with <integer> significant digits (e.g.. significant digits 3 of 1235569 = 1240000). Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
sum of <integer>	*Plain*	Returns the sum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
upper bound of <integer range>	*Plain*	The high end of the integer range. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Properties

Integers are represented internally as 64-bit signed values.

Key Phrase	*Form*	Return Type	Description
<integer> as bit set	*Cast*	<bit set>	Returns the bits of the binary representation of the integer; bit zero is the least-significant bit. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<integer> as bits	*Cast*	<bit set>	Returns the bits of the binary representation of the integer; bit zero is the least-significant bit. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<integer> as day_of_month	*Cast*	<day of month>	Cast an integer as a day of the month type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<integer> as floating point	*Cast*	<floating point>	Converts an integer into a floating point number. Win:4.1, Mac:4.1
<integer> as hexadecimal	*Cast*	<string>	Converts an integer into a hexadecimal string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
<integer> as integer	*Cast*	<integer>	Reflective cast for completeness. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer> as month	*Cast*	<month>	Returns the name of the nth month of the year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<integer> as string	*Cast*	<string>	Converts an integer to a string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer> as year	*Cast*	<year>	Casts an integer as a year type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
absolute value of <integer>	*Plain*	<integer>	Returns the positive value of the integer. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
bit <integer> of <integer>	*Numbered*	<boolean>	Returns TRUE if the numbered bit is on. Bits are numbered starting at zero. Bit 0 is the least significant bit. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
maximum of <integer>	*Plain*	<integer>	Returns the maximum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
mean of <integer>	*Plain*	<floating point>	The mean of the integer(s). Win:5.1, Mac:4.1
minimum of <integer>	*Plain*	<integer>	Returns the minimum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
product of <integer>	*Plain*	<integer>	Multiplies a list of integers, returning the product. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
set of <integer>	*Plain*	<integer set>	Creates a set from the given list of semicolon-separated integers. Win:7.0
significant digits <integer> of <integer>	*Numbered*	<integer>	Returns a number with <integer> significant digits (e.g., significant digits 3 of 1235569 = 1240000). Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
standard deviation of <integer>	*Plain*	<floating point>	The standard deviation of the integer(s). Win:5.1, Mac:4.1
sum of <integer>	*Plain*	<integer>	Returns the sum of a list of integers. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
unique value of <integer>	*Plain*	<integer with multiplicity>	Given a set of integers, returns the number of instances of each integer. Given (1,2,2,2,3), returns (1,3,1). Earlier versions of this Inspector returned the unique set of integers. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Operators

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

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<hertz> {op} <integer>	*<hertz>*	Returns a hertz object operated on by the given integer, where: • {op} is one of: *, / . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer set> contains <integer>	*<boolean>*	Returns TRUE if the specified set contains the given integer. Win:7.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<integer> * <time range>	*<timed(time range, integer)>*	Returns a tuple of a time interval and an integer. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<integer> {cmp} <floating point>	*<boolean>*	Compares an integer to a floating point number, where: • {cmp} is one of: =, <=, <. Win:4.1, Mac:4.1
<integer> {cmp} <integer>	*<boolean>*	Returns boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer> {cmp} <registry key value type>	*<boolean>*	Returns boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2
<integer> {cmp} <registry key value>	*<boolean>*	Returns boolean TRUE or FALSE, depending on the comparison operator, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2
<integer> {op} <floating point>	*<floating point>*	Operates on an integer and a floating point number, returning a floating point number, where: • {op} is one of: -, +, *, /. Win:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<integer> {op} <integer>	*<integer>*	Returns the integer solution to the equation, depending on the operator, where: • {op} is one of: +, -, *, /, mod . Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<number of months> {op} <integer>	*<number of months>*	Where {op} is one of: *, /. Win:6.0
<time range> * <integer>	*<timed(time range, integer)>*	Returns a time interval labeled with the specified integer, in the form of: • (<date> to <date>), <integer>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Examples

- `hexadecimal integer "A0"`
  - Returns 160.
- `integer ceiling of (15/8 as floating point)`
  - Returns 2.
- `integer floor of ("1.9" as floating point)`
  - Returns 1.
- `maximum of (sizes of files of folder "c:\")`
  - Returns the size of the largest file in the indicated folder.
- `minimum of (sizes of files of folder "c:\")`
  - Returns the size of the smallest file in the indicated folder.
- `numeric value of "string 123 xyz 45" = 123`
  - Returns TRUE.
- `product of (1;2;3)`
  - Returns 6.
- `sum of (sizes of files of folder "c:\")`
  - Returns the sum of the sizes of all files in the specified folder.
- `255 as hexadecimal`
  - Returns the string "ff".

© 2007 by BigFix, Inc.

- `maximum of (7;2;4;5)`
  - Returns 7.
- `minimum of (sizes of files of folder "c:\")`
  - Returns the size of the smallest file in the indicated folder.
- `significant digits 3 of 1235569`
  - Returns 1240000.
- `sum of (sizes of files of folder "c:\")`
  - Returns the sum of the sizes of all files in the specified folder.
- `set of (1;2;3) contains 3`
  - Returns TRUE.
- `(July-current month) < 2*month`
  - Returns TRUE when the current date is between June and July.
- `21 mod 5`
  - Returns 1.

Integer Range

Specifies a range between two 64-bit signed integers.

Creation Methods

Key Phrase	*Form*	Description
distance of <selected server>	*Plain*	The distance, in IP gateway hops, to the server. Among servers with the same priority, closer servers are preferred. Returns an integer range, since the exact distance may not be known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

Properties

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

© 2007 by BigFix, Inc.

Integer With Multiplicity

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

Creation Methods

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

Properties

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

Examples

- `unique values of (1;2;3;3)`
- Returns a list of the count of each integer, namely 1,1,2.

- `multiplicities of unique values of (1;2;3;3)`
- Returns the multiplicity of (the number of times) each number in the list is used, namely, 1,1,2.

© 2007 by BigFix, Inc.

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.

Creation Methods

Key Phrase	*Form*	Description
set of <integer>	*Plain*	Creates a set from the given integers (usually plural) Win:7.0
union of <integer set>	*Plain*	Returns a set of integers equal to the union of the specified sets. Win:7.0

Properties

Key Phrase	*Form*	Return Type	Description
element of <integer set>	*Plain*	<integer>	Typically used in the plural, returns the individual elements of the specified set of integers. Win:7.0
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
size of <integer set>	*Plain*	<integer>	Returns the number of elements in the specified set. Win:7.0
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

© 2007 by BigFix, Inc.

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
<integer set> * <integer set>	*<integer set>*	Returns the intersection of the two specified sets. Win:7.0
<integer set> + <integer set>	*<integer set>*	Returns the union of the specified sets. Win:7.0
<integer set> = <integer set>	*<boolean>*	Returns TRUE if the specified sets have identical contents. Win:7.0
<integer set> contains <integer set>	*<boolean>*	Returns TRUE if the first set contains all the elements of the second set. Win:7.0
<integer set> contains <integer>	*<boolean>*	Returns TRUE if the specified set contains the given integer. Win:7.0

Examples

- `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 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.

© 2007 by BigFix, Inc.

- `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:4.1, Mac:4.1
<string> as floating point	*Cast*	Converts the contents of a string into a floating point number. Win:4.1, Mac:4.1
floating point <string>	*NamedGlobal*	Creates a floating point number from the provided string. Win:4.1, Mac:4.1
less significance <integer> of <floating point>	*Numbered*	Removes <integer> number of digits of significance from the floating point value. Win:4.1, Mac:4.1
mean of <floating point>	*Plain*	The mean of the floating point number(s). Win:5.1, Mac:4.1
mean of <integer>	*Plain*	The mean of the integer(s). Win:5.1, Mac:4.1
more significance <integer> of <floating point>	*Numbered*	Adds <integer> number of digits of significance to the floating point value. Win:4.1, Mac:4.1
relative significance place <integer> of <floating point>	*Numbered*	The same floating point value, to be expressed to the given number of significant digits. Win:4.1, Mac:4.1

© 2007 by BigFix, Inc.

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

Properties

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

© 2007 by BigFix, Inc.

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

© 2007 by BigFix, Inc.

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

© 2007 by BigFix, Inc.

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

Operators

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

© 2007 by BigFix, Inc.

Examples

- `4.5 as floating point`
  - Returns 4.5.
- `less significance 2 of floating point "5.115"`
  - Returns 5.1.
- `mean of floating points( "1.3";"2.5")`
  - Returns 1.90.
- `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.
- `15/2 as integer`
  - Returns 7.
- `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.

© 2007 by BigFix, Inc.

- `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.

© 2007 by BigFix, Inc.

String

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

Creation Methods

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

Key Phrase	***Form***	**Description**
<boolean> as string	*Cast*	Operates on a boolean to return a string. Possible values are "True" and "False". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<date> as string	*Cast*	Cast a date type as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<floating point> as scientific notation	*Cast*	Converts a floating point number into a string with scientific notation. Win:4.1, Mac:4.1
<floating point> as standard notation	*Cast*	Converts a floating point number into a string with standard notation. Win:4.1, Mac:4.1
<floating point> as string	*Cast*	Converts a floating point number into a string with standard notation. Win:4.1, Mac:4.1
<hertz> as string	*Cast*	Creates a string containing the number of hertz and the word hertz, e.g., (3 * hz) as string = "3 hertz". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<integer> as hexadecimal	*Cast*	Converts an integer into a hexadecimal string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
<integer> as string	*Cast*	Creates a string formatted with the integer provided. (-22) as string = "-22". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as hexadecimal	*Cast*	Converts a string to a hexadecimal number. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
<string> as left trimmed string	*Cast*	Trims the leading spaces from a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as lowercase	*Cast*	Creates a lowercase version of the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as right trimmed string	*Cast*	Trims the trailing spaces from a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as string	*Cast*	Reflexive cast of string to string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as trimmed string	*Cast*	Trims the leading and trailing spaces off of the specified string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as uppercase	*Cast*	Creates an uppercase version of the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time interval> as string	*Cast*	Returns a string formatted as • ddd days, HH:MM:SS.mmmmmm • For example, millisecond as string = " 00:00:00.001". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time zone> as string	*Cast*	Creates a string containing a time zone. See <time zone>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as local string	*Cast*	Creates a string containing a time. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as string	*Cast*	Creates a string containing a time. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as universal string	*Cast*	Creates a string containing a time. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
character <integer>	*NumberedGlobal*	Creates a string containing the single ASCII character for the decimal number provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
concatenation <string> of <string>	*Named*	Combines the second set of strings into a single string, separated by the first string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
concatenation of <string>	*Plain*	Combines the supplied strings into a single string, end-to-end. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
hexadecimal string <string>	*NamedGlobal*	Creates a string from the given hexadecimal value. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
parameter <string>	*NamedGlobal*	This Inspector is a synonym for the parameter <string> of <action>. It looks up the value of the action parameter specified by <string>. This is used in conjunction with the parameter set command. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
string <string>	*NamedGlobal*	Creates a string matching the name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
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
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
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

Properties

Key Phrase	*Form*	Return Type	Description
<string> as boolean	*Cast*	<boolean>	Returns a boolean value for the string. All possible capitalization's of "TRUE" and "FALSE" will convert successfully. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
<string> as date	*Cast*	<date>	Casts a string as a date type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as day_of_month	*Cast*	<day of month>	Casts a string as a day of the month (eg. 28). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as day_of_week	*Cast*	<day of week>	Casts a string as a day of the week. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as floating point	*Cast*	<floating point>	Converts the contents of a string into a floating point number. Win:4.1, Mac:4.1
<string> as hexadecimal	*Cast*	<string>	Converts a string to a hexadecimal number. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
<string> as integer	*Cast*	<integer>	Returns an integer value for the string provided. If the string contains anything but ASCII digits, the conversion will fail. Use numeric value for more liberal parsing rules. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<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
<string> as left trimmed string	*Cast*	<string>	Trims the leading spaces from a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as local time	*Cast*	<time>	Returns a local time object from a properly formatted string. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as 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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as lowercase	*Cast*	<string>	Returns a lowercase version of the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as month	*Cast*	<month>	Converts a string into a month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
<string> as right trimmed string	*Cast*	<string>	Trims the trailing spaces from a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as site version list	*Cast*	<site version list>	Converts a string into a site version list. Win:7.0
<string> as string	*Cast*	<string>	Returns the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time	*Cast*	<time>	Returns a time object from a properly formatted string. See <time>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time interval	*Cast*	<time interval>	Returns a time interval object from a properly formatted string. Expects strings formatted as • ddd days, HH:MM:SS.mmmmmm. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time zone	*Cast*	<time zone>	Returns a time zone object from a properly formatted string. See <time zone>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time_of_day	*Cast*	<time of day>	Converts a string to a time_of_day type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as trimmed string	*Cast*	<string>	Trims the leading and trailing spaces off of the specified string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as universal time	*Cast*	<time>	Returns a universal time object from a properly formatted string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as universal zoned time_of_day	*Cast*	<time of day with time zone>	Converts a string into a universal zoned time of day. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as uppercase	*Cast*	<string>	Returns an uppercase version of the string provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
<string> as version	*Cast*	<version>	Returns a version if the string can be parsed as a version. The first numeric set of characters delimited with period, comma or comma-space is returned. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
<string> as windows display time	*Cast*	<time>	Returns a Windows display time object from a properly formatted string. See <Time>. Win:1.2
<string> as year	*Cast*	<year>	Converts a string into a year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as zoned time_of_day	*Cast*	<time of day with time zone>	Converts a string into a zoned time of day. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
character <integer> of <string>	*Numbered*	<substring>	Returns a string of length 1 made by taking the character identified by <integer> from the string. Numbering begins at zero. Example, Character 1 of "HI" is "I". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
character of <string>	*Plain*	<substring>	Returns the characters from the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
concatenation <html> of <string>	*Indexed*	<html>	This is an additional overload of the existing html concatenation operators with the same conventions (string arguments are escaped). Win:7.0
concatenation <string> of <string>	*Named*	<string>	Concatenation <string1> of <string2> concatenates a list of strings indicated by string2, placing string1 between each. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
concatenation of <string>	*Plain*	<string>	Combines the supplied strings into a single string, end-to-end. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
escape of <string>	*Plain*	<string>	Returns a string containing a \\ for every \ character found. Useful for setting registry key values to strings in regset action commands. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
expand environment string of <string>	*Plain*	<string>	Uses the Windows ExpandEnvironmentStrings API to translate a string containing special Windows environment variables. For example, %windir%\my.dll might expand to c:\winnt\my.dll. Win:1.2
first <integer> of <string>	*Numbered*	<substring>	Returns a substring containing the number of characters specified from the given string. For example, First 5 of "To be or not to be" is "To be". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
first <string> of <string>	*Named*	<substring>	Returns a substring containing the first occurrence of the name provided. See substring. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last <integer> of <string>	*Numbered*	<substring>	Returns a substring containing the number of characters specified. For example, Last 5 of "To be or not to be" is "to be". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last <string> of <string>	*Named*	<substring>	Returns a substring containing the last occurrence of the name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
length of <string>	*Plain*	<integer>	Returns the number of characters in the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
numeric value of <string>	*Plain*	<integer>	Returns an integer for the first numeric value in the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
position <integer> of <string>	*Numbered*	<string position>	Returns a string position pointing to the character position specified. The first character is at position 0. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
position of <string>	*Plain*	<string position>	Returns the positions of the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
set of <string>	*Plain*	<string set>	Creates a set from the given list of semicolon-separated strings. Win:7.0

© 2007 by BigFix, Inc.

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

Operators

Key phrase	Return Type	Description
<rope> & <string>	*<rope>*	Concatenates a rope and a string, producing a rope. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<rope> contains <string>	*<boolean>*	Returns boolean TRUE if the rope contains the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string set> contains <string>	*<boolean>*	Win:7.0
<string> & <rope>	*<rope>*	Concatenates a rope and a string, returning a new rope. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> & <string>	*<string>*	Concatenates two strings, producing a new string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

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

Note

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

Examples

- `floating point "600987.9" as scientific notation`
  - Returns 6.009879e+5.
- `floating point "6.009e8" as standard notation`
  - Returns 600900000.
- `255 as hexadecimal`
  - Returns the string "ff".
- `concatenation ":" of (names of files of folder "c:\")`
  - Returns a single string with the names of each file in the specified path separated by a colon.
- `concatenation of "light" & "year"`
  - Returns "lightyear".
- `"01 Apr 2020" as date`
  - Returns Wed, 01 Apr 2020.
- `Tue as day_of_week`
  - Returns Tuesday.
- `4.5 as floating point`
  - Returns 4.5.
- `exists character whose (it is "z") of "Paul Cezanne"`
  - Returns True.

© 2007 by BigFix, Inc.

- `concatenation "/" of ("a" ; "b" ; "c" )`
  - Returns "a/b/c".

- `concatenation of (name of it & ":") of files of folder "c:\"`
  - Returns a single string with the names of each file in the specified path separated by a colon.

- `first 2 of pathname of regapp "bigfix.exe" as lowercase = "c:"`
  - Returns true if BigFix is installed on drive C:.

- `preceding text of last "ab" of "abracadabra" is "abracad"`
  - Returns True.

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

- `substrings before "<--" of "the item pointed to <--"`
  - Returns "the item pointed to".

- `substrings between "*" of "the item *between* asterisks"`
  - Returns "between".

- `substrings separated by "," of "1,2,3"`
  - Returns the list of numbers separated by commas in the specified string.

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

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

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

© 2007 by BigFix, Inc.

String Position

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

Note

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

Examples

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

Substring

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

Creation Methods

Key Phrase	*Form*	Description
character <integer> of <string>	*Numbered*	Creates the single character substring at the position given within the string. For example, character 2 of "abc" = "c". Note that numbering begins at zero. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
character of <string>	*Plain*	Iterates through the string (or substring) returning substrings that contain the individual characters of the string. For example, number of characters of string "abc" = 3. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
first <integer> of <string>	*Numbered*	Creates a substring for the given number of characters at the start of the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
first <string> of <string>	*Named*	Creates an object containing the first match of the given string. For example, first "be" of "to be or not to be" = "be". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
following text of <string position>	*Plain*	Creates an object containing the substring following the position in the string. For example, following text of position 5 of "0123456789" = "567890". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
following text of <substring>	*Plain*	Creates an object containing the string following the substring. For example, following text of last "." of "log.txt" = "txt". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last <integer> of <string>	*Numbered*	Creates an object containing a substring from the last part of the string containing the number of characters specified. For example, Last 5 of "To be or not to be" is "to be". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last <string> of <string>	*Named*	Creates a substring containing the last occurrence of the name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
preceding text of <string position>	*Plain*	Creates the substring preceding the position in the string. For example, preceding text of position 5 of "0123456789" = "01234". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
preceding text of <substring>	*Plain*	Creates an object containing the string preceding the substring. For example, preceding text of last "." of "log.txt" = "log". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
substring <string> of <string>	*Named*	Iterates through the string returning all the substrings matching the name given. For example, number of substrings "be" of "to be or not to be" = 2. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
substring after <string> of <string>	*Named*	Returns the substrings that come after the first string delimiter. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring before <string> of <string>	*Named*	Returns the substrings that come before the first string delimiter. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring between <string> of <string>	*Named*	Returns the substring in the second string found between two instances of the first string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
substring separated by <string> of <string>	*Named*	Returns a substring (or set of substrings) delimited by the first string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

© 2007 by BigFix, Inc.

Properties

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

Note

All the string operators can also be applied to substrings.

Examples

- `first 2 of pathname of regapp "bigfix.exe" as lowercase = "c:"`
  - Returns true if BigFix is installed on drive C:.
- `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.

© 2007 by BigFix, Inc.

String With Multiplicity

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

Creation Methods

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

Properties

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

Examples

- `unique values of ("steak"; "chop"; "rib"; "rib"; "rib")`
  - Returns the unique values of the multiple strings in alphabetical order, namely chop, rib, steak.

- `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.

© 2007 by BigFix, Inc.

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.

Creation Methods

Key Phrase	*Form*	Description
union of <string set>	*Plain*	Returns a set of strings equal to the union of the specified sets. Win:7.0

Properties

Key Phrase	*Form*	Return Type	Description
element of <string set>	*Plain*	<string>	Typically used in the plural, returns the individual elements of the specified set of strings. Win:7.0
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
size of <string set>	*Plain*	<integer>	Returns the number of elements in the specified set. Win:7.0
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

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
<string set> * <string set>	*<string set>*	Returns the intersection of the specified sets. Win:7.0

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<string set> + <string set>	*<string set>*	Returns the union of the specified sets. Win:7.0
<string set> = <string set>	*<boolean>*	Returns TRUE if the specified sets have identical contents. Win:7.0
<string set> contains <string set>	*<boolean>*	Returns TRUE if the first set contains all the elements of the second set. Win:7.0
<string set> contains <string>	*<boolean>*	Win:7.0

Examples

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

© 2007 by BigFix, Inc.

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

Properties

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

Operators

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

© 2007 by BigFix, Inc.

Bit Set

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
<bit set> as integer	*Cast*	<integer>	Returns the integer whose binary representation matches the bit set. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
<bit set> as string	*Cast*	<string>	Returns the bits (0s and 1s) in a string format. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
bit <integer> of <bit set>	*Numbered*	<boolean>	Returns the value of the bit at the given <integer> position in the set. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
least significant one bit of <bit set>	*Plain*	<integer>	Returns the least n such that bit n of the set is true. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
left shift <integer> of <bit set>	*Numbered*	<bit set>	A bit set which, at each position n >= delta, holds bit n-delta of the original bit set, where delta is the given integer. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
most significant one bit of <bit set>	*Plain*	<integer>	Returns the greatest n such that bit n of the set is true. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
one bit of <bit set>	*Plain*	<integer>	Returns the numbers n for which bit n of the set is true. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
right shift <integer> of <bit set>	*Numbered*	<bit set>	A bit set which, at each position n, holds bit n+delta of the original bit set, where delta is the given shift integer. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Operators

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

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<bit set> contains <bit set>	*<boolean>*	Returns false if -- for any n -- bit n of the left set is false, but bit n of the right set is true. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Examples

- `bit 0 of 5`
  - Returns TRUE.
- `bit 3 of bit 3`
  - Returns TRUE.

Regular Expression

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

Creation Methods

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

Operators

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

© 2007 by BigFix, Inc.

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

Examples

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

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

Regular Expression Match

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

Creation Methods

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

© 2007 by BigFix, Inc.

Properties

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

Examples

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

Undefined

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

Creation Methods

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

Examples

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

© 2007 by BigFix, Inc.

Hertz

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

Creation Methods

Key Phrase	*Form*	Description
absolute value of <hertz>	*Plain*	Creates a hertz object with a positive value. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
ghz	*PlainGlobal*	Creates a hertz object corresponding to 1 giga-hertz. For example, ghz = 1000*mhz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
greatest hz	*PlainGlobal*	Creates the largest hertz object that can be represented on the current machine. It returns the value 9,223,372,036,854,775,807 hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hz	*PlainGlobal*	Creates a hertz object corresponding to 1 hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
khz	*PlainGlobal*	Creates a hertz object corresponding to 1 kilohertz. For example, khz = 1000*hz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
least hz	*PlainGlobal*	Creates the largest negative hertz object that can be represented on the current machine. It returns the value -9,223,372,036,854,775,808 hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
mhz	*PlainGlobal*	Creates a hertz object corresponding to 1 megahertz. For example, mhz = 1000*khz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
significant digits <integer> of <hertz>	*Numbered*	Rounds up the value of a hertz object with <integer> significant digits. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Properties

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

Operators

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

Examples

- `speed of processor > 3*ghz`
  - Returns TRUE on machines faster than 3Ghz.
- `greatest hz`
  - Returns a large positive value, such as 9223372036854775807 hertz.

© 2007 by BigFix, Inc.

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

- `significant digits 2 of speed of processor/mhz/ 1000 as floating point`
  - Returns a floating point representation of the processor speed in GHz, such as 3.4 ghz.

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

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

Time

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

Creation Methods

Key Phrase	*Form*	Description
<string> as local time	*Cast*	Local time creates a time object by parsing the string literal provided. The time zone is optional. If not present, the local time zone is assumed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as time	*Cast*	Parses the string. Time zone information must be provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<string> as universal time	*Cast*	Parses the string. If time zone is not provided in the string, the universal time zone is assumed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<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:1.2
maximum of <time>	*Plain*	Returns the maximum time from a list of times. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

© 2007 by BigFix, Inc.

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

Properties

Key Phrase	*Form*	Return Type	Description
<time> as local string	*Cast*	<string>	Returns a string in MIME format of the given time object. The format is: ddd, DD mmm YYYY HH:MM:SS sZZZZ. The string is formatted using the local time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as string	*Cast*	<string>	Same as above. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<time> as universal string	*Cast*	<string>	Returns a string in MIME format of the given time object. The format is: • ddd, DD mmm YYYY HH:MM:SS +0000 • The string is formatted using the universal time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
date <time zone> of <time>	*Indexed*	<date>	Returns the date adjusted for the specified time zone. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
maximum of <time>	*Plain*	<time>	Returns the maximum time from a list of times. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
minimum of <time>	*Plain*	<time>	Returns the minimum time from a list of times. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
time <time zone> of <time>	*Indexed*	<time of day with time zone>	Adjusts the specified time to the given time zone. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Operators

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

© 2007 by BigFix, Inc.

Note

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

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

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

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

© 2007 by BigFix, Inc.

Examples

- `maximum of (modification times of files of folder "temp" of windows folder)`
  - Returns the latest time stamp from the files in the windows temporary folder.

- `minimum of (modification times of files of folder "temp" of windows folder)`
  - Returns the latest earliest stamp from the files in the windows temporary folder.

- `now`
  - Returns the current time.

- `time "Sat, 01 Jan 2000 00:00:00 -0400" & now`
  - Returns a time range from the beginning of the millennia to now, eg:

• Sat, 01 Jan 2000 00:00:00 -0400 to Sat, 08 Apr 2006 20:39:51 -0400.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
midnight	*PlainGlobal*	Returns 00:00:00 as a time of day object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
noon	*PlainGlobal*	Returns 12:00:00 as a time of day object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
time of <time of day with time zone>	*Plain*	Returns the time of day, without the time zone information. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
time_of_day <string>	*NamedGlobal*	Creates a time of day object out of the given string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
<time of day> as string	*Cast*	<string>	Casts the time of day as a string type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
hour_of_day of <time of day>	*Plain*	<integer>	Returns the hour section of the 'time of day' object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
minute_of_hour of <time of day>	*Plain*	<integer>	Returns the 'minutes after the hour' section of the 'time of day' object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
second_of_minute of <time of day>	*Plain*	<integer>	Extracts the 'seconds after the minute' section of the 'tim of day' object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
two digit hour of <time of day>	*Plain*	<string>	Extracts the 2-digit hour from the time of day. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
two digit minute of <time of day>	*Plain*	<string>	Extracts the 2-digit minute from the time of day. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
two digit second of <time of day>	*Plain*	<string>	Extracts the 2-digit second from the time of day. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Operators

Key phrase	Return Type	Description
<time of day> - <time of day>	*<time interval>*	Subtracts two times of day, returning a time interval. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<time of day> {cmp} <time of day>	*<boolean>*	Compares two times of day, where {cmp} is one of: <, <=, =. Win:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Time Zone

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

Creation Methods

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

Properties

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

Operators

Key phrase	Return Type	Description
<time 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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<time zone> {op} <time interval>	*<time zone>*	Returns a time zone object offset by a time interval, where: • {op} is one of: +, - . Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Examples

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

Time 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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as universal zoned time_of_day	*Cast*	Converts a string into a universal zoned time of day. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as zoned time_of_day	*Cast*	Converts a string into a zoned time of day. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current time_of_day	*PlainGlobal*	Returns the current time of day in the local time zone. Win:7.0
current time_of_day <time zone>	*IndexedGlobal*	Returns the current time of day in the specified time zone. Win:7.0
time <time zone> of <time>	*Indexed*	Converts the specified time to the given time zone. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
time of <time of day with time zone>	*Plain*	<time of day>	Returns the time of day, without the time zone information. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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
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
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
zone of <time of day with time zone>	*Plain*	<time zone>	Returns the zone associated with the specified time. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<time of day with time zone> = <time of day with time zone>	*<boolean>*	Compares two times of day with time zone. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Examples

- `12:00:00 -4000 as universal zoned time_of_day`
  - Returns 04:00:00 +0000.
- `time (time zone "+0000") of now`
  - Returns the time in Greenwich, England.
- `hour_of_day of time (universal time zone) of now`
  - Returns the hour of day in Greenwich, England.
- `minute_of_hour of time (local time zone) of now`
  - Returns the current minute past the hour.
- `time zone "+0000" & time (universal time zone) of now`
  - Returns the time in Greenwich, England.

© 2007 by BigFix, Inc.

Time Range

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
<time range> as string	*Cast*	<string>	Casts a time range as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
end of <time range>	*Plain*	<time>	Returns the end date of a time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
final part <time interval> of <time range>	*Indexed*	<time range>	Returns a time range with the specified interval, but ending on the final date of the time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

© 2007 by BigFix, Inc.

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

Operators

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

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<time range> + <time range>	*<time range>*	Returns the smallest range that contains both ot the specified ranges (same as <time range> & <time range>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
<time> & <time range>	*<time range>*	Concatenates a time and a time range, producing a new time range. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Examples

- `(week & now) * (day & now)`
  - Returns a one-day time range (from yesterday to today).
- `(week & now) + (day & now)`
  - Returns a one-week time range (from a week ago to today).

Time Interval

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

Creation Methods

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

© 2007 by BigFix, Inc.

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

© 2007 by BigFix, Inc.

Properties

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

Operators

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

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0
<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:6.0
<time> & <time interval>	*<time range>*	Concatenates a time and a time interval, producing a time range object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Examples

- `maximum of ("00:00:00" as time interval; "01:01:01"as time interval)`
  - Returns 01:01:10.

- `minimum of ("00:00:00" as time interval; "01:01:01"as time interval)`
  - Returns 00:00:00.

- `time interval "01:00:00" & now`
  - Returns a one-hour time range ending now, eg. Fri, 07 Apr 2006 12:36:10 -0400 to Fri, 07 Apr 2006 13:36:10 -0400.

© 2007 by BigFix, Inc.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current date	*PlainGlobal*	Returns the current date in the format: • Day of week, Day Month Year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
date <time zone> of <time>	*Indexed*	Returns the date adjusted for the specified time zone. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<date> as string	*Cast*	<string>	Cast a date type as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_month of <date>	*Plain*	<day of month>	Extracts the day of the month from the specified date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_week of <date>	*Plain*	<day of week>	Extracts the day of the week (Monday, Tuesday, etc.) from the specified date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_year of <date>	*Plain*	<day of year>	Extracts the day of year from the specified date, in the 'Month Day' format. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month of <date>	*Plain*	<month>	Returns the month derived from the given date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
month_and_year of <date>	*Plain*	<month and year>	Formats the specified date in month year format, eg. March 2012. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year of <date>	*Plain*	<year>	Returns the year, extracted from the given date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Operators

Key phrase	Return Type	Description
<date> - <date>	*<time interval>*	Subtracts two dates to produce a time interval. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<date> & <time of day with time zone>	*<time>*	Concatenates a date with a time and a time zone for a complete time stamp. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<date> {op} <number of months>	*<date>*	Adds or subtracts a specified number of months to a given date, where {op} is one of: -, +. Win:6.0
<date> {op} <time interval>	*<date>*	Add or subtract a time interval to a date, producing a new date, where {op} is one of: -, +. Win:6.0
<number of months> + <date>	*<date>*	Adds a number of months to a date, returning a new date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<time interval> + <date>	*<date>*	Adds a time interval (days, hours, minutes, seconds) to a date to create a new date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Examples

- `01 Apr 2020 as date`
  - Returns Wed, 01 Apr 2020.
- `april 1 of 2020`
  - Returns Wed, 01 Apr 2020..
- `date "09 Apr 2006"`
  - Returns Sun, 09 Apr 2006.
- `now - time "Sat, 01 Jan 2000 00:00:00 -0000"`
  - Returns the number of days, hours, minutes and seconds since the turn of the millennia, eg: 2288 days, 17:53:06.
- `current date + 14*month`
  - Returns the date 14 months from today, eg. Mon, 07 Jan 2008.
- `current date + time interval "7 days"`
  - Returns the date a week from now.

Day Of Week

These Inspectors provide tools for dealing and calculating with day-of-week types, which include Monday, Tuesday, etc. 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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current day_of_week	*PlainGlobal*	Retruns the current day of the week, eg. Monday, Tuesday, etc. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_week <string>	*NamedGlobal*	Converts the given string value to a day of week type, eg. Monday, Tuesday, etc. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_week of <date>	*Plain*	Extracts the day of the week from the specified date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
friday	*PlainGlobal*	Returns Friday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
monday	*PlainGlobal*	Returns the day of week object for Monday. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
saturday	*PlainGlobal*	Returns Saturday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
sunday	*PlainGlobal*	Returns Sunday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
thursday	*PlainGlobal*	Returns Thursday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
tuesday	*PlainGlobal*	Returns Tuesday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
wednesday	*PlainGlobal*	Returns Wednesday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<day of week> as string	*Cast*	<string>	Casts the day of week as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<day of week> as three letters	*Cast*	<string>	Casts the day of week as a three-letter abbreviation (Mon, Tue, etc.). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Operators

Key phrase	Return Type	Description
<day of week> - <day of week>	*<time interval>*	Subtract two day of week types (Monday, Tuesday, etc.) to produce a time interval. The answer cannot exceed 6 days. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<day of week> = <day of week>	*<boolean>*	Compares two days of the week and returns a boolean TRUE or FALSE. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Examples

- `Tue as day_of_week`
  - Returns Tuesday.
- `day_of_week "Tuesday"`
  - Returns Tuesday as a 'day of week' object.
- `Saturday as three letters`
  - Returns Sat.
- `Friday - Wednesday`
  - Returns 2.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as day_of_month	*Cast*	Casts a string as a day of month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current day_of_month	*PlainGlobal*	Returns the current day of the month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day of <day of year>	*Plain*	Returns the day of the month of the specified date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
day_of_month <integer>	*NumberedGlobal*	Converts the given integer to a day of month type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_month <string>	*NamedGlobal*	Converts the given string value (must be an integer from 1-31) to a day of month type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_month of <date>	*Plain*	Extracts the day of the month from the specified date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<day of month> as integer	*Cast*	<integer>	Cast a day of month type as an integer. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<day of month> as string	*Cast*	<string>	Cast a day of month type as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<day of month> as two digits	*Cast*	<string>	Cast a day of month type as a two-digit number. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<day of month> & <month and year>	*<date>*	Concatenate a day of month with a month and year type to produce a complete date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<day of month> {cmp} <day of month>	*<boolean>*	Compare two day of month types, where {cmp} is one of: <, <=, =. Win:6.0

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<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:6.0
<month and year> & <day of month>	*<date>*	Concatenates a month and year with a day of month to produce a complete date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month> & <day of month>	*<day of year>*	Concatenates a month and a day of the month to produce a day of year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Examples

- `day of July 4`
  - Returns 4.
- `day_of_month of current date`
  - Returns the current day of the month.
- `current day_of_month as integer`
  - Returns the day of the month as an integer.
- `(day_of_month 2) & june of 2008`
  - Returns Mon, 02 Jun 2008.
- `(day_of_month 2) & june`
  - Returns June 2.

© 2007 by BigFix, Inc.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
august <integer>	*NumberedGlobal*	Returns the nth day of August as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current day_of_year	*PlainGlobal*	Retruns the current day of the year, in a Month Day format. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_year of <date>	*Plain*	Extracts the day of year from the specified date, in the 'Month Day' format. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
december <integer>	*NumberedGlobal*	Returns the nth day of December as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
february <integer>	*NumberedGlobal*	Returns the nth day of February as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
january <integer>	*NumberedGlobal*	Returns the nth day of January as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
july <integer>	*NumberedGlobal*	Returns the nth day of July as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
june <integer>	*NumberedGlobal*	Returns the nth day of June as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
march <integer>	*NumberedGlobal*	Returns the nth day of March as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
may <integer>	*NumberedGlobal*	Returns the nth day of May as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
november <integer>	*NumberedGlobal*	Returns the nth day of November as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
october <integer>	*NumberedGlobal*	Returns the nth day of October as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
september <integer>	*NumberedGlobal*	Returns the nth day of September as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day of <day of year>	*Plain*	<day of month>	Returns the day of the month of the specified date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month of <day of year>	*Plain*	<month>	Returns the month portion of the given date (in month day format). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0
<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:6.0
<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:6.0
<month and year> & <day of year>	*<date>*	Concatenates a month and year with a day of year to produce a complete date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<year> & <day of year>	*<date>*	Concatenates a year with the day of the year, returning a full date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Examples

- `year 2020 & april 1`
- Returns Sun, 01 Apr 2007.

© 2007 by BigFix, Inc.

Month

These Inspectors provide tools for dealing and calculating with month types, which are of the form January, February, etc. 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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as month	*Cast*	Converts a string into a month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
april	*PlainGlobal*	Returns april as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
august	*PlainGlobal*	Returns August as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current month	*PlainGlobal*	Returns the current month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
december	*PlainGlobal*	Returns December as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
february	*PlainGlobal*	Returns February as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
january	*PlainGlobal*	Returns January as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
july	*PlainGlobal*	Returns July as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
june	*PlainGlobal*	Returns June as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
march	*PlainGlobal*	Returns March as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
may	*PlainGlobal*	Returns May as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month <integer>	*NumberedGlobal*	Returns the month type corresponding to the given <integer>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
month <string>	*NamedGlobal*	Returns a month type corresponding to the given <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month of <date>	*Plain*	Returns the month of the given date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month of <day of year>	*Plain*	Returns the month portion of the given date (in month day format). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month of <month and year>	*Plain*	Returns the month portion of the given date (in month year format). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
november	*PlainGlobal*	Returns November as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
october	*PlainGlobal*	Returns October as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
september	*PlainGlobal*	Returns September as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<month> as integer	*Cast*	<integer>	Converts the given month into an integer (1-12). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month> as string	*Cast*	<string>	Converts the given month into a string value. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month> as three letters	*Cast*	<string>	Converts the given month into a 3-letter string (Jan, Feb, etc.). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month> as two digits	*Cast*	<string>	Converts the month into a two digit number (01 - 12). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month> & <day of month>	*<day of year>*	Concatenates a month and a day of the month to produce a day of year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month> & <year>	*<month and year>*	Returns a date (in month year format) from the concatenation of a month and a year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month> {cmp} <month>	*<boolean>*	Compares the values of two months, where {cmp} is one of: <, <=, =. Win:6.0
<month> {op} <number of months>	*<month>*	Adds or subtracts a number of months from the given month. Here {op} is one of: -, +. Win:6.0
<number of months> + <month>	*<month>*	Adds a number of months to the given month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<year> & <month>	*<month and year>*	Returns a date (in month year format) from the concatenation of a month and a year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Examples

- `5 as month`
  - Returns May.
- `january as month - 1 * month`
  - Returns December (one month before January).
- `current month + 2*month`
  - Returns the name of the month, two months from today.

© 2007 by BigFix, Inc.

- `month 9`
  - Returns September.
- `month "jun"`
  - Returns June.
- `month of current date`
  - Returns the current month, eg. September.
- `month of (day_of_year of (current date + 40*day))`
  - Returns the name of the month 40 days from today, eg. October.
- `january as three letters`
  - Returns Jan.
- `january as two digits`
  - Returns 01.
- `december - current month`
  - Returns the number of months left until december. If the current month is April, it returns 8 months.
- `December 3 & "2032" as year`
  - Returns Fri, 03 Dec 2032.
- `July <= current month`
  - Returns true in the second half of the year, when the month is greater than or equal to July.
- `current month + 2*month`
  - Returns the name of the month two months from now. If it's currently January, this would return March.
- `year 2134 & april`
  - Returns April 2134.

© 2007 by BigFix, Inc.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
august of <integer>	*Plain*	Creates a date (in month year format) corresponding to August of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current month_and_year	*PlainGlobal*	Returns the current date in month year format, eg. January 2012. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
december of <integer>	*Plain*	Creates a date (in month year format) corresponding to December of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
february of <integer>	*Plain*	Creates a date (in month year format) corresponding to February of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
january of <integer>	*Plain*	Creates a date (in month year format) corresponding to January of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
july of <integer>	*Plain*	Creates a date (in month year format) corresponding to July of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
june of <integer>	*Plain*	Creates a date (in month year format) corresponding to June of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
march of <integer>	*Plain*	Creates a date (in month year format) corresponding to March of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
may of <integer>	*Plain*	Creates a date (in month year format) corresponding to May of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
month_and_year of <date>	*Plain*	Formats the specified date in month year format, eg. March 2012. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
november of <integer>	*Plain*	Creates a date (in month year format) corresponding to November of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
october of <integer>	*Plain*	Creates a date (in month year format) corresponding to October of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
september of <integer>	*Plain*	Creates a date (in month year format) corresponding to September of the specified year (as an <integer>). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
first <day of week> of <month and year>	*Indexed*	<date>	Finds the specific date corresponding to the first day of the week (eg. Friday) for a given month and year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
first friday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Friday of any given month and year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
first monday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Monday of any given month and year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
first saturday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Saturday of any given month and year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
first sunday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Sunday of any given month and year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
first thursday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Thursday of any given month and year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
first tuesday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Tuesday of any given month and year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
first wednesday of <month and year>	*Plain*	<date>	Finds the date corresponding to the first Wednesday of any given month and year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
length of <month and year>	*Plain*	<time interval>	Returns the number of days in the specified month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month of <month and year>	*Plain*	<month>	Returns the name of the month corresponding to the given date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year of <month and year>	*Plain*	<year>	Returns the year portion of the specified date (in month year format). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month and year> - <month and year>	*<number of months>*	Subtracts two dates (in month year format), returning a number of months. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month and year> {cmp} <month and year>	*<boolean>*	Compares two dates (in month year format), where {cmp} is one of: <, <=, =. Win:6.0

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<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:6.0
<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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Examples

- `month_and_year of current date`
  - Returns the current date formatted as month year, eg. April, 2006.
- `first monday of april of 2020`
  - Returns the date of the first Monday in April 2020, which is Mon, 06 Apr 2020.
- `length of (month "February" & year "2004")`
  - Returns 29.
- `month of date "Sun, 02 Apr 2006" + 2*month`
  - Returns June.
- `year of current date`
  - Returns the current year, eg. 2006.
- `January of 2020 - current month_and_year`
  - Returns a time interval measured to the nearest month, such as 13 years, 9 months.
- `january of 2009 < current month_and_year`
  - Evaluates to TRUE when the current date is later than the specified date. This phrase could serve as an expiration flag.
- `current month_and_year + 18*month`
  - Gives a date 18 months ahead of the current date, eg. October 2007.

© 2007 by BigFix, Inc.

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
month	*PlainGlobal*	Returns the specified number of months. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year	*PlainGlobal*	Returns the specified number of years as a <number of months> type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Operators

Key phrase	Return Type	Description
- <number of months>	*<number of months>*	Creates the negative of the specified number of months. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<number of months> {cmp} <number of months>	*<boolean>*	Compare two numbers of months, where {cmp} is one of: <, <=, =. Win:6.0
<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:6.0
<number of months> + <year>	*<year>*	Returns the year after adding the specified number of months. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

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:6.0

Examples

- `24*month`
- Returns 2 years.

- `year 1984 + 264*month`
- Returns 2006.

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<string> as year	*Cast*	Converts a string into a year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current year	*PlainGlobal*	Returns the current year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year <integer>	*NumberedGlobal*	Creates a year object from the specified integer. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year <string>	*NamedGlobal*	Creates a year object from the specified string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year of <date>	*Plain*	The year derived from the given date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year of <month and year>	*Plain*	Returns the year portion of the specified date (in month year format). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
<year> as integer	*Cast*	<integer>	Casts a year as an integer. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<year> as string	*Cast*	<string>	Casts a year as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
leap of <year>	*Plain*	<boolean>	Returns a flag indicating whether or not the specified year is a leap year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
length of <year>	*Plain*	<time interval>	Returns the number of day in the specified year. Leap years have 366 days. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<month> & <year>	*<month and year>*	Returns a date (in month year format) from the concatenation of a month and a year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<year> - <year>	*<number of months>*	Subtracts two years and produces a time interval marked in months and years. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<year> & <day of year>	*<date>*	Concatenates a year with the day of the year, returning a full date. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<year> & <month>	*<month and year>*	Returns a date (in month year format) from the concatenation of a month and a year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
<year> {cmp} <year>	*<boolean>*	Compares two years, where {cmp} is one of: <, <=, =. Win:6.0

© 2007 by BigFix, Inc.

Examples

- `length of year "2008"`
  - Returns 366.
- `year 2020 - year 2008`
  - Returns 12 years.
- `year 2080 > current year`
  - Returns TRUE until the year 2080.

© 2007 by BigFix, Inc.

World Objects

World

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

Properties

Key Phrase	*Form*	Return Type	Description
action	*PlainGlobal*	<action>	Returns the action currently being parsed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
action <integer>	*NumberedGlobal*	<action>	Returns the action matching the <integer> id. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
action lock state	*PlainGlobal*	<action lock state>	Returns the client action lock state. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
active action	*PlainGlobal*	<action>	Returns the action currently executing. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
active device	*PlainGlobal*	<active device>	Returns a list of all active devices found using the Configuration Manager SetupDiGetClassDevs NT API. Win:1.2
active device file	*PlainGlobal*	<file>	Under Windows NT, returns a list of file objects corresponding the list returned from the Windows NT EnumDeviceDrivers() function. Win:1.2
active device file <string>	*NamedGlobal*	<file>	Under Windows NT, returns a file object corresponding to the name provided. See file. Win:1.2
all firewall scope	*PlainGlobal*	<firewall scope>	Returns the scope of computers that allow ALL traffic through the firewall, corresponding to the Microsoft enumerated type NET_FW_SCOPE_ALL. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
allow firewall action	*PlainGlobal*	<firewall action>	Returns a value corresponding to the NET_FW_ACTION enumerated type, which specifies the action for a rule or default setting. This is a Vista Inspector. For more information, see the MSDN Library entry for NET_FW_ACTION. Win:7.0
ansi code page	*PlainGlobal*	<integer>	Returns an integer value of the Windows API GetACP. Win:4.1
any ip version	*PlainGlobal*	<ip version>	Returns a type corresponding to the Microsoft enumerated value NET_FW_IP_VERSION_ANY. Win:5.1
apparent registration server time	*PlainGlobal*	<time>	Shorthand for 'now of registration server'. When the client registers with the server, the server passes its current time back to the client. The client starts a stop watch at that time. The apparent registration server time is the time the server passed back to the client, plus the elapsed time on the stop watch. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
application <string>	*NamedGlobal*	<application>	Returns an application for the name provided. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
application event log	*PlainGlobal*	<event log>	Returns the object corresponding to the application event log, which records certain application events, such as the failure of MS SQL to access a database. Win:6.0
application usage summary	*PlainGlobal*	<application usage summary>	Returns an application usage summary containing information including the start time, duration and other statistics on client applications. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
application usage summary <string>	*NamedGlobal*	<application usage summary>	Returns the usage summary for the application specified in <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
april	*PlainGlobal*	<month>	Returns april as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
april <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of april as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
audit failure event log event type	*PlainGlobal*	<event log event type>	Returns an object corresponding to an audit failure -- an event related to the failed execution of an action. Win:6.0
audit success event log event type	*PlainGlobal*	<event log event type>	Returns an object corresponding to an audit success in an event log. Win:6.0
august	*PlainGlobal*	<month>	Returns August as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
august <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of August as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
backoffice bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the backoffice bit of the Windows operating system suite mask is set. Win:6.0
bes license	*PlainGlobal*	<license>	Synonym for 'client license'. Win:7.0
binary operator <string>	*NamedGlobal*	<binary operator>	Typically used in the plural, returns the various possible binary inspectors that use the specified operators. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
binary operator returning <type>	*IndexedGlobal*	<binary operator>	Returns a list of binary operators that return the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
bit <integer>	*NumberedGlobal*	<bit set>	Returns TRUE or FALSE, corresponding to value of the bit specified by <integer>. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
bit set <string>	*NamedGlobal*	<bit set>	Returns the bits of the binary number given by the string. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
blade bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the blade bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
block firewall action	*PlainGlobal*	<firewall action>	Returns a value corresponding to the NET_FW_ACTION enumerated type, which specifies the action for a rule or default setting. This is a Vista Inspector. For more information, see the MSDN Library entry for NET_FW_ACTION. Win:7.0
boolean <string>	*NamedGlobal*	<boolean>	Returns a boolean. For example, boolean "TRUE". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
case insensitive regex <string>	*NamedGlobal*	<regular expression>	Returns a case-insensitive regular expression from the supplied string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
case insensitive regular expression <string>	*NamedGlobal*	<regular expression>	Same as case insensitive regex <string>. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
cast <string>	*NamedGlobal*	<cast>	Returns a list of the objects that can be cast into the type specified by <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
cast returning <type>	*IndexedGlobal*	<cast>	Returns a list of the objects that can be cast into the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
character <integer>	*NumberedGlobal*	<string>	Returns a string containing a single ASCII character. For example, character 90 = "Z". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
client	*PlainGlobal*	<client>	Returns the client object corresponding to the BigFix application evaluating the current relevance expression. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
client license	*PlainGlobal*	<license>	Global object containing client licensing information. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
communications bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the communications bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
computer id	*PlainGlobal*	<integer>	This is a unique integer assigned to the computer by the BES system. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
computer name	*PlainGlobal*	<string>	Returns a string corresponding to the name of the computer as it appears on the network. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
connection status <integer>	*NumberedGlobal*	<connection status>	Returns the connection status based on its integer value. This Inspector is included to take advantage of new (or undocumented) additions to the status values. Win:5.0
connection status authenticating	*PlainGlobal*	<connection status>	Returns the value NCS_AUTHENTICATING: The connection is waiting for authentication to occur. Win:5.0
connection status authentication failed	*PlainGlobal*	<connection status>	Returns the value NCS_AUTHENTICATION_FAILED: Authentication has failed on this connection. Win:5.0
connection status authentication succeeded	*PlainGlobal*	<connection status>	Returns the value NCS_AUTHENTICATION_SUCCEEDED: Authentication has succeeded on this connection. Win:5.0
connection status connected	*PlainGlobal*	<connection status>	Returns the value NCS_CONNECTED: The connection is in a connected state. Win:5.0
connection status connecting	*PlainGlobal*	<connection status>	Returns the value NCS_CONNECTING: The connection is in the process of connecting. Win:5.0
connection status disconnected	*PlainGlobal*	<connection status>	Returns the value NCS_DISCONNECTED: The connection is disconnected. Win:5.0
connection status disconnecting	*PlainGlobal*	<connection status>	Returns the value NCS_DISCONNECTING: The connection is in the process of disconnecting. Win:5.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
connection status hardware disabled	*PlainGlobal*	<connection status>	Returns the value NCS_HARDWARE_DISABLED: The hardware for the connection is present, but is not enabled. Win:5.0
connection status hardware malfunction	*PlainGlobal*	<connection status>	Returns the value NCS_HARDWARE_MALFUNCTION: A malfunction has occurred in the hardware for the connection. Win:5.0
connection status media disconnected	*PlainGlobal*	<connection status>	Returns the value NCS_MEDIA_DISCONNECTED: The media, for example the network cable, is disconnected. Win:5.0
connection status no hardware present	*PlainGlobal*	<connection status>	Returns the value NCS_NO_HARDWARE_PRESENT: The hardware for the connection, for example network interface card (NIC), is not present. Win:6.0
csidl folder <integer>	*NumberedGlobal*	<folder>	Returns the csidl folder corresponding to the specified integer. The windows SHGetSpecialFolderLocation API is used to look up paths to special folders, which are identified by passing the specified integer as the second argument of the API call. These values and their meaning are described in the windows ShlObj.h include file found in the development sdk. • Note that some of these folders do not exist in the Local System context. Win:7.0
current date	*PlainGlobal*	<date>	Returns the current date in the format: • Day of week, Day Month Year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current day_of_month	*PlainGlobal*	<day of month>	Returns the current day of the month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current day_of_week	*PlainGlobal*	<day of week>	Retruns the current day of the week, eg. Monday, Tuesday, etc. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
current day_of_year	*PlainGlobal*	<day of year>	Retruns the current day of the year, in a Month Day format. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current firewall profile type	*PlainGlobal*	<firewall profile type>	Retrieves the type of firewall profile that is currently in effect. Win:5.1
current month	*PlainGlobal*	<month>	Returns the current month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current month_and_year	*PlainGlobal*	<month and year>	Returns the current date in month year format, eg. January 2012. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
current relay	*PlainGlobal*	<current relay>	Returns an object corresponding to the server or relay that the client last registered with. This may be a BES Relay or the BES root server. Win:7.0
current site	*PlainGlobal*	<site>	Returns the current site object. See site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
current time_of_day	*PlainGlobal*	<time of day with time zone>	Returns the current time of day in the local time zone. Win:7.0
current time_of_day <time zone>	*IndexedGlobal*	<time of day with time zone>	Returns the current time of day in the specified time zone. Win:7.0
current user	*PlainGlobal*	<current user>	Returns the current user if one is logged in to the desktop. • Note: For Unix, this returns a <user> type as of BES version 6.0. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
current user	*PlainGlobal*	<logged on user>	Returns the active, console (local) user, if logged on. Otherwise does not exist. Win:7.0
current year	*PlainGlobal*	<year>	Returns the current year. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
custom firewall scope	*PlainGlobal*	<firewall scope>	Returns the custom firewall scope, corresponding to the Microsoft enumerated type: NET_FW_SCOPE_CUSTOM. Win:5.1
custom site subscription effective date <string>	*NamedGlobal*	<time>	Returns the date the custom site (specified by <string>) was last subscribed or unsubscribed. It is used internally by BES to manage custom site subscriptions. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
datacenter bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the datacenter bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0
date <string>	*NamedGlobal*	<date>	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:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 day. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
day_of_month <integer>	*NumberedGlobal*	<day of month>	Converts the given integer to a day of month type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_month <string>	*NamedGlobal*	<day of month>	Converts the given string value (must be an integer from 1-31) to a day of month type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
day_of_week <string>	*NamedGlobal*	<day of week>	Converts the given string value to a day of week type, eg. Monday, Tuesday, etc. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
december	*PlainGlobal*	<month>	Returns December as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
december <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of December as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
default web browser	*PlainGlobal*	<application>	Returns the application currently associated with HTML files. This is a Windows and Macintosh inspector; it will fail gracefully under other operating systems, rather than generate an error. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
dmi	*PlainGlobal*	<dmi>	Creates the global dmi object. If no dmi information is available, creation of the object will fail. Win:1.2, Lin:4.1
dns name	*PlainGlobal*	<string>	Returns the DNS name of the computer. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
domain firewall profile type	*PlainGlobal*	<firewall profile type>	Creates a domain firewall profile type for comparison. Win:5.1
domain user	*PlainGlobal*	<local user>	Returns all of the users that are members of the domain for which the machine is a user. Win:4.1
domain user <string>	*NamedGlobal*	<local user>	Returns the local user object corresponding to the specified name. Win:4.1
drive	*PlainGlobal*	<drive>	Returns all the drive objects defined on the machine. • Note: For Unix, this Inspector returns a <filesystem> object as of version 6.0 of BES. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1
drive <string>	*NamedGlobal*	<drive>	Returns a drive object for the name provided. • Note: For Unix, this Inspector returns a <filesystem> object as of version 6.0 of BES. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
embedded nt bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the embedded nt bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0
embedded restricted bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the embedded restricted bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
enterprise bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the enterprise bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0
environment	*PlainGlobal*	<environment>	Returns an object corresponding to the currently defined set of environment variables. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
error <string>	*NamedGlobal*	<undefined>	Always fails; if an error message is generated, it is based on the given string. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
error event log event type	*PlainGlobal*	<event log event type>	Returns an object corresponding to an error in the event log, such as the failure of a service to start. Win:6.0
event log <string>	*NamedGlobal*	<event log>	Returns the named event log, which contains historical information that help to track down system and security problems. There are several distinct logs that you can specify, including: • Application log: records application events • Security log: recirds global or local policy audit events • System log: records OS events. Win:6.0
event log event type <integer>	*NumberedGlobal*	<event log event type>	Returns an event type object corresponding to the specified number. The enumerated types include: • 1: error event • 2: warning event • 4: information event • 8: audit success event • 16: audit failure event. Win:6.0
false	*PlainGlobal*	<boolean>	Returns the boolean FALSE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
february	*PlainGlobal*	<month>	Returns February as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
february <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of February as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
file <string>	*NamedGlobal*	<file>	Returns a filesystem object corresponding to the full pathname provided in <string>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
file_and_print firewall service type	*PlainGlobal*	<firewall service type>	Returns the global service type for file and print sharing, corresponding to the Microsoft enumerated type: NET_FW_SERVICE_FILE_AND_PRINT. Win:6.0
firewall	*PlainGlobal*	<firewall>	Returns the global firewall object for this computer. Win:5.1
firewall action <integer>	*NumberedGlobal*	<firewall action>	Returns the Nth enumerated type from the Firewall Action interface. This provides an alternative way to access the firewall action variables and requires Vista. For more information, see the MSDN Library entry for NET_FW_ACTION. Win:7.0
firewall local policy modify state <integer>	*NumberedGlobal*	<firewall local policy modify state>	Returns a value corresponding to the NET_FW_MODIFY_STATE enumerated type, which specifies the effect of modifications to the current policy. The <integer> specifies the zero-based enumeration, and provides an alternative access method. For more information, see the MSDN Library entry for NET_FW_MODIFY_STATE. Win:7.0
firewall profile type <integer>	*NumberedGlobal*	<firewall profile type>	Returns the firewall profile type corresponding to the given integer: • 0: Domain • 1: Standard • 2: Current. Win:5.1
firewall scope <integer>	*NumberedGlobal*	<firewall scope>	Returns the scope of addresses from which a port can listen. Win:5.1
firewall service type <integer>	*NumberedGlobal*	<firewall service type>	Returns the firewall service type specified by <integer>. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
floating point <string>	*NamedGlobal*	<floating point>	Creates a floating point number from the provided string. Win:4.1, Mac:4.1
folder <string>	*NamedGlobal*	<folder>	Returns a folder object for the name provided. See drive. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
friday	*PlainGlobal*	<day of week>	Returns Friday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
full wmi <string>	*NamedGlobal*	<wmi>	Returns a wmi object which can retrieve all values, including system values. Win:3.0
ghz	*PlainGlobal*	<hertz>	Returns a Hertz object corresponding to 1 giga-hertz. See hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
gp override firewall local policy modify state	*PlainGlobal*	<firewall local policy modify state>	Returns a value corresponding to the NET_FW_MODIFY_STATE enumerated type, which specifies the effect of modifications to the current policy. For more information, see the MSDN Library entry for NET_FW_MODIFY_STATE. Win:7.0
greatest hz	*PlainGlobal*	<hertz>	Returns the largest hertz object that can be represented on this machine. See hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
greatest integer	*PlainGlobal*	<integer>	Returns the largest integer that can be represented on this machine. See integer. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
greatest time interval	*PlainGlobal*	<time interval>	Returns the greatest time interval representable. The value corresponds to 106751991 days, 04:00:54.775807. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hexadecimal integer <string>	*NamedGlobal*	<integer>	Creates an integer from the provided hexadecimal value. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
hexadecimal string <string>	*NamedGlobal*	<string>	Creates a string from the given hexadecimal value. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
hostname	*PlainGlobal*	<string>	Returns the standard host name, usually for the computer's network. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hour	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 hour. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
hyperthreading capable	*PlainGlobal*	<boolean>	Returns a boolean TRUE if the agent is able to detect that the processor is capable of running with hyperthreading enabled. Win:6.0
hyperthreading enabled	*PlainGlobal*	<boolean>	Returns TRUE if the machine is running with hyperthreading enabled, a method in which each physical processor on the machine presents itself as multiple logical processors to the operating system. Win:5.0
hz	*PlainGlobal*	<hertz>	Returns a hertz object corresponding to 1 hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
inbound blocked firewall local policy modify state	*PlainGlobal*	<firewall local policy modify state>	Returns a value corresponding to the NET_FW_MODIFY_STATE enumerated type, which specifies the effect of modifications to the current policy. For more information, see the MSDN Library entry for NET_FW_MODIFY_STATE. Win:7.0
information event log event type	*PlainGlobal*	<event log event type>	Returns an object corresponding to an information event -- An informational event which is generally related to a successful action. Win:6.0
install folder <integer>	*NumberedGlobal*	<folder>	Returns a folder object corresponding to the number provided. The placement of some system folders can be found using numbers that have been associated with those folders. See folder. Win:1.2
integer <integer>	*NumberedGlobal*	<integer>	Returns an integer. The keyword is optional. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
integer <string>	*NamedGlobal*	<integer>	Returns integer for name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
internet protocol <integer>	*NumberedGlobal*	<internet protocol>	Returns the firewall internet protocol specified by the given integer. These correspond to the Microsoft enumerated types: • NET_FW_IP_PROTOCOL_TCP • NET_FW_IP_PROTOCOL_UDP. Win:5.1
ip version <integer>	*NumberedGlobal*	<ip version>	Returns the the IP version for the specified integer. Win:5.1
ipv4	*PlainGlobal*	<ip version>	Provides a comparison value for a firewall ip version inspector. Win:5.1
ipv4 address <string>	*NamedGlobal*	<ipv4 address>	Returns an ip address for the string provided. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ipv6	*PlainGlobal*	<ip version>	Provides a comparison value for a firewall ip version inspector. Win:5.1
ipv6 address <string>	*NamedGlobal*	<ipv6 address>	Converts a string representations of an IPv6 address (with colons and/or dots) as an IPv6 address type. Win:7.0
january	*PlainGlobal*	<month>	Returns January as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
january <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of January as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
july	*PlainGlobal*	<month>	Returns July as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
july <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of July as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
june	*PlainGlobal*	<month>	Returns June as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
june <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of June as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
khz	*PlainGlobal*	<hertz>	Returns a hertz object corresponding to 1 kilohertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
least hz	*PlainGlobal*	<hertz>	Returns the least hertz value that can be represented on this machine. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
least integer	*PlainGlobal*	<integer>	Returns the least integer value that can be represented on this machine. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
least time interval	*PlainGlobal*	<time interval>	Returns the least time interval that can be represented on this machine. The value corresponds to -106751991 days, 04:00:54.775808. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
local administrator	*PlainGlobal*	<boolean>	Returns the boolean TRUE if the user belongs to the local administrator group. Also returns TRUE for Win9x and WinME. Win:1.2
local group	*PlainGlobal*	<local group>	Returns local groups defined on the local computer using the windows NetLocalGroupEnum API. Several local groups are defined simply by a default operating system install, and have names such as Administrators, Backup Operators, Guests, Network Configuration Operators, Power users, Users, etc. Some software applications also define local groups in order to help manage protections. Win:6.0
local group <string>	*NamedGlobal*	<local group>	Returns a local group corresponding to the given name, such as Adminstrator, Guests, etc. Win:6.0
local mssql database	*PlainGlobal*	<local mssql database>	Returns local MSSQL database objects. Win:1.2
local mssql database <string>	*NamedGlobal*	<local mssql database>	Returns the local Microsoft SQL (MSSQL) database object identified by the name provided. Win:1.2
local subnet firewall scope	*PlainGlobal*	<firewall scope>	Returns the local subnet firewall scope, corresponding to the Microsoft enumerated type: NET_FW_SCOPE_LOCAL_SUBNET. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
local time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
local time zone	*PlainGlobal*	<time zone>	Returns a time zone object corresponding to the local time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
local user	*PlainGlobal*	<local user>	Returns all local users of the machine. Win:1.2
local user <string>	*NamedGlobal*	<local user>	Returns the named local user. Win:1.2
logged on user	*PlainGlobal*	<logged on user>	Returns the user logged on to this BES Client. This Inspector iterates through all logged-on users, using Terminal Services, ACLs, and on Win 9x, the registry. Win:7.0
logical processor count	*PlainGlobal*	<integer>	Returns the number of logical processors available per physical processor. This can be interpreted as the number of hyperthreads that could be enabled on the machine. On a machine with 2 physical processors, each with 2 possible hyperthreads per processor, the 'physical processor count' and the 'logical processor count' would both return 2, while the 'number of processors' would return 4, since there are a total of 4 logical processors available for work. With hyperthreading turned off, the 'number of processors', 'logical processor count' and 'physical processor count' would all be 2. Disabling one of those processors will then give 'number of processors'=1, 'logical processor count'=2, and 'physical processor count'=1. If the number of processors / physical processor count != logical processor count, you can turn on hyperthreading. Win:6.0
main gather service	*PlainGlobal*	<service>	Returns a service object for the main gathering service, typically located on the main server. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
main processor	*PlainGlobal*	<processor>	Returns the processor object corresponding to the main processor. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
march	*PlainGlobal*	<month>	Returns March as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
march <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of March as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
may	*PlainGlobal*	<month>	Returns May as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
may <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of May as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
media type <integer>	*NumberedGlobal*	<media type>	Returns the media type based on its integer value. This Inspector is included to take advantage of new (or undocumented) additions to the media types. Win:5.0
media type bridge	*PlainGlobal*	<media type>	Returns the value NCM_BRIDGE: Bridged connection. Win:5.0
media type direct	*PlainGlobal*	<media type>	Returns the value NCM_DIRECT: Direct serial connection through a serial port. Win:5.0
media type isdn	*PlainGlobal*	<media type>	Returns the value NCM_ISDN: Connection is through an integrated services digital network (ISDN) line. Win:5.0
media type lan	*PlainGlobal*	<media type>	Returns the value NCM_LAN: Connection is to a local area network (LAN). Win:5.0
media type phone	*PlainGlobal*	<media type>	Returns the value NCM_PHONE: Dial-up connection over a conventional phone line. Win:5.0
media type pppoe	*PlainGlobal*	<media type>	Returns the value NCM_PPPOE: Point-to-Point protocol (PPP) over Ethernet. Win:5.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
media type shared access host lan	*PlainGlobal*	<media type>	Returns the value NCM_SHAREDACCESSHOST_LAN: Shared connection to a LAN. Win:5.0
media type shared access host ras	*PlainGlobal*	<media type>	Returns the value NCM_SHAREDACCESSHOST_RAS: Shared connection to a remote or wide area network (WAN). Win:5.0
media type tunnel	*PlainGlobal*	<media type>	Returns the value NCM_TUNNEL: Virtual private network (VPN) connection. Win:5.0
metabase	*PlainGlobal*	<metabase>	Returns the IIS metabase object. Win:4.1
mhz	*PlainGlobal*	<hertz>	Returns a hertz object corresponding to 1 megahertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
microsecond	*PlainGlobal*	<time interval>	Returns a time interval corresponding to .000001 seconds. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
midnight	*PlainGlobal*	<time of day>	Returns 00:00:00 as a time of day object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
millisecond	*PlainGlobal*	<time interval>	Returns a time interval corresponding to .001 seconds. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
minute	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 minute. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
module <string>	*NamedGlobal*	<module>	For BigFix internal use only. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
monday	*PlainGlobal*	<day of week>	Returns the day of week object for Monday. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month	*PlainGlobal*	<number of months>	Returns the specified number of months. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
month <integer>	*NumberedGlobal*	<month>	Returns the month type corresponding to the given <integer>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
month <string>	*NamedGlobal*	<month>	Returns a month type corresponding to the given <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
native registry	*PlainGlobal*	<registry>	On 32 bit versions of windows, this returns the same as registry32 and registry. On 64 bit versions of windows, this returns the same as registry64. Win:6.0
network	*PlainGlobal*	<network>	Returns an object containing properties of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
network share	*PlainGlobal*	<network share>	Creates a network shared object. Win:4.1
network share <string>	*NamedGlobal*	<network share>	Creates a named network shared object. Win:4.1
none firewall service type	*PlainGlobal*	<firewall service type>	Returns the no firewall service type, corresponding to the Microsoft enumerated type: NET_FW_SERVICE_NONE. Win:6.0
noon	*PlainGlobal*	<time of day>	Returns 12:00:00 as a time of day object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
november	*PlainGlobal*	<month>	Returns November as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
november <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of November as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
now	*PlainGlobal*	<time>	Returns the current time as a time object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
nt domain controller product type	*PlainGlobal*	<operating system product type>	Returns an object corresponding to OS product type of nt domain controller. Win:6.0
nt server product type	*PlainGlobal*	<operating system product type>	Returns an object corresponding to OS product type of nt server. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
nt workstation product type	*PlainGlobal*	<operating system product type>	Returns an object corresponding to OS product type of nt workstation. Win:6.0
october	*PlainGlobal*	<month>	Returns October as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
october <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of October as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
oem code page	*PlainGlobal*	<integer>	Returns an integer value of the Windows API GetOEMCP. Win:4.1
ok firewall local policy modify state	*PlainGlobal*	<firewall local policy modify state>	Returns a value corresponding to the NET_FW_MODIFY_STATE enumerated type, which specifies the effect of modifications to the current policy. For more information, see the MSDN Library entry for NET_FW_MODIFY_STATE. Win:7.0
operating system	*PlainGlobal*	<operating system>	Returns the operating system object. See operating system. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
operating system product type <integer>	*NumberedGlobal*	<operating system product type>	Returns an object corresponding to the numbered OS product type. Win:6.0
parameter <string>	*NamedGlobal*	<string>	This Inspector is a synonym for the parameter <string> of <action>. It looks up the value of the action parameter specified by <string>. This is used in conjunction with the parameter set command. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
pending login	*PlainGlobal*	<boolean>	Installers may leave values in the registry that the operating system will execute when the next user logs in. Pending login can detect these registry entries. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
pending restart	*PlainGlobal*	<boolean>	Returns TRUE if the operating system indicates that a restart needs to occur. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
pending restart <string>	*NamedGlobal*	<boolean>	Immediately after issuing a command like 'Action requires restart "PatchGroupX”', the expression 'Pending restart "PatchGroupX”' will be true until the next restart. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
personal bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the personal bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0
physical processor count	*PlainGlobal*	<integer>	Returns the number of physical processors on the machine. Note that 'number of processors' returns the number of logical processors. To determine the number of logical processors per physical processor, use 'number of processors / physical processor count'. Win:5.0
private firewall profile type	*PlainGlobal*	<firewall profile type>	Retrieves the enumerated variable corresponding to private profile type. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
processor	*PlainGlobal*	<processor>	Returns all the processor objects defined on the machine. See processor. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
processor <integer>	*NumberedGlobal*	<processor>	Returns a processor object for the numbered processor. Processors are numbered from 1. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
property <string>	*NamedGlobal*	<property>	Typically used in the plural, returns the "line" Inspector properties. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property returning <type>	*IndexedGlobal*	<property>	Produces a list of the Inspector properties that return the specified <type>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
public firewall profile type	*PlainGlobal*	<firewall profile type>	Retrieves the enumerated variable corresponding to public profile type. This profile type is used for public internet access points. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
ram	*PlainGlobal*	<ram>	Returns a ram object for inspecting the properties of Random Access Memory installed on the machine. See ram. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
random access memory	*PlainGlobal*	<ram>	Same as above. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1
recent application	*PlainGlobal*	<application>	Returns all the application objects that have recently been executing on the machine. See application. Win:1.2, Lin:6.0
recent application <string>	*NamedGlobal*	<application>	Returns an application for the name provided it has recently executed. The name is assumed to be the last part of an executable file name. Win:1.2, Lin:6.0
regapp	*PlainGlobal*	<application>	Returns all the application objects defined under the 'App Paths' key of the registry. • Note: This Inspector returns a <filesystem> object on the Macintosh. Win:1.2
regapp <string>	*NamedGlobal*	<application>	Returns an application object for the name provided. On a Macintosh, returns a file that has been registered with Launch Services. See application and regapp. Win:1.2
regex <string>	*NamedGlobal*	<regular expression>	Creates a regex object from the given string. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
registry	*PlainGlobal*	<registry>	Returns a registry object. Win:1.2
regular expression <string>	*NamedGlobal*	<regular expression>	Same as regex <string>. Win:6.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
relay service	*PlainGlobal*	<service>	Returns a service object for the relay component of BES. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
remote desktop firewall service type	*PlainGlobal*	<firewall service type>	Returns the remote desktop firewall service type, corresponding to the Microsoft enumerated type: NET_FW_SERVICE_REMOTE_DESKTOP. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
root server	*PlainGlobal*	<root server>	Returns an object representing the root BES Server to which the client last registered. Win:7.0
rope <string>	*NamedGlobal*	<rope>	Creates a rope object from the given string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
rsop computer wmi	*PlainGlobal*	<wmi>	Provides access to the Resultant Set of Policy (RSoP) WMI classes via the RsopLoggingModeProvider. This is used to examine the state of the GPO (Group Policy object) security policies. Win:7.0
rsop user wmi <security identifier>	*IndexedGlobal*	<wmi>	Each user has its own RSoP (Resultant Set of Policy) namespace based on the user's Security Identifier (SID). This Inspector returns the namespace specified by the <security identifier>. Win:7.0
running application	*PlainGlobal*	<application>	Returns all the application objects that are currently executing on the machine. See application. Win:1.2, Lin:6.0
running application <string>	*NamedGlobal*	<application>	Returns an application for the name provided it is currently executing. The name is assumed to be the last part of an executable file name. Win:1.2, Lin:6.0
running service	*PlainGlobal*	<service>	Returns all the running service objects. Win:1.2
running service <string>	*NamedGlobal*	<service>	Returns the running service object matching the name provided. Win:1.2
saturday	*PlainGlobal*	<day of week>	Returns Saturday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
second	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 second. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
security event log	*PlainGlobal*	<event log>	Returns a security event log, which records global or local group policy events. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
selected server	*PlainGlobal*	<selected server>	The BES Server or BES Relay to which the agent reports. Returned as the "selected server" type. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
september	*PlainGlobal*	<month>	Returns September as an object of type month. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
september <integer>	*NumberedGlobal*	<day of year>	Returns the nth day of September as a 'day of year' type (month day). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
service	*PlainGlobal*	<service>	Returns all the service objects. Win:1.2
service <string>	*NamedGlobal*	<service>	Returns the service object matching the name provided regardless of its running state. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
single user ts bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the single user ts bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0
site	*PlainGlobal*	<site>	Returns all the site objects that are currently loaded into memory. See site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site <string>	*NamedGlobal*	<site>	Returns a site object for the name provided. The name is the URL of the site location. See site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site version list <string>	*NamedGlobal*	<site version list>	Returns a textual representation of a site version list ("manyversion"). Win:7.0
small business bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the small business bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0
small business restricted bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the small business restricted bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
standard firewall profile type	*PlainGlobal*	<firewall profile type>	Returns the Standard firewall profile type. This is a global property. Win:5.1
string <string>	*NamedGlobal*	<string>	Returns a string for the name provided. The keyword string is optional. For example, string "hi" = "hi". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
sunday	*PlainGlobal*	<day of week>	Returns Sunday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
system event log	*PlainGlobal*	<event log>	Returns a system event log, which records OS or component events, such as the failure of a bootup service. Win:6.0
system file <string>	*NamedGlobal*	<file>	Returns a file object corresponding to the relative pathname provided. Win:1.2
system ini device file	*PlainGlobal*	<file>	Returns a list of file objects corresponding to all the device files loaded as a result of a device= lines of the system.ini file. See file. Win:1.2
system ini device file <string>	*NamedGlobal*	<file>	Returns a file object corresponding to a device file loaded as a result of a device= line of the system.ini file. Win:1.2
system language	*PlainGlobal*	<string>	Returns the language of the system as a string. It is identified using the GetSystemDefaultLangID() system call. See the language keyword of the application object for a list of possible language value. Win:1.2, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1
system locale	*PlainGlobal*	<language>	Determines which bitmap fonts, and OEM, ANSI, and MAC code pages are defaults for the system. This only affects applications that are not fully Unicode. Win:4.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1
system ui language	*PlainGlobal*	<language>	Determines the default language of menus and dialogs, messages and help files. Win:4.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
system wow64 folder	*PlainGlobal*	<folder>	Returns a filesystem object corresponding to a "Windows On Windows 64" system folder, which does not exist on 32-bit Windows. You can find out more about the WOW64 system folder at the Microsoft site: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/getsystemwow64directory.asp. Win:6.0
system x32 folder	*PlainGlobal*	<folder>	Returns a filesystem object corresponding to a 32-bit system folder. On a 32-bit machine, this is equivalent to the normal system folder. Win:6.0
system x64 folder	*PlainGlobal*	<folder>	Returns a filesystem object corresponding to a 64-bit system folder. This is the same as the system folder, but with file system redirection disabled. For more information about file redirection, see the Microsoft site http://msdn.microsoft.com/library/default.asp?url=/library/en-us/win64/win64/file_system_redirector.asp. Win:6.0
tcp	*PlainGlobal*	<internet protocol>	Returns an internet protocol corresponding to the Microsoft enumerated type: NET_FW_IP_PROTOCOL_TCP. Win:5.1
terminal bit <operating system suite mask>	*IndexedGlobal*	<boolean>	Returns TRUE if the terminal bit of the Suite Mask (a part of the Windows OS version) is set. Win:6.0
thursday	*PlainGlobal*	<day of week>	Returns Thursday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
time interval <string>	*NamedGlobal*	<time interval>	Creates a time interval from the string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
time zone <string>	*NamedGlobal*	<time zone>	Returns a time zone object for the name provided. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
time_of_day <string>	*NamedGlobal*	<time of day>	Creates a time of day object out of the given string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
total processor core count	*PlainGlobal*	<integer>	Returns an integer corresponding to the total number of processor cores. Win:6.0
true	*PlainGlobal*	<boolean>	Returns the boolean TRUE. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
tuesday	*PlainGlobal*	<day of week>	Returns Tuesday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
udp	*PlainGlobal*	<internet protocol>	Returns an internet protocol corresponding to the Microsoft enumerated type: NET_FW_IP_PROTOCOL_UDP. Win:5.1
unary operator <string>	*NamedGlobal*	<unary operator>	Typically used in the plural, this inspector returns a list of objects that use the specified operator. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
unary operator returning <type>	*IndexedGlobal*	<unary operator>	Returns a list of the unary operator inspectors (such as negative) that return the specified type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
universal time <string>	*NamedGlobal*	<time>	Returns a time object for the name provided. See time. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
universal time zone	*PlainGlobal*	<time zone>	Returns a time zone object corresponding to the universal time zone. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
upnp firewall service type	*PlainGlobal*	<firewall service type>	Returns the UPnP (Universal Plug and Play) firewall service type, corresponding to the Microsoft enumerated type: NET_FW_SERVICE_UPNP. • Note: UPnP is not the same as PnP. UPnP is used for network connectivity via TCP/IP to various devices (scanners, printers, etc.). Win:5.1
user language	*PlainGlobal*	<string>	Returns the language of the system as a string. It is identified by using the GetUserDefaultLangId() system call. See the language keyword of the application object for a list of possible language value. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
user locale	*PlainGlobal*	<language>	Determines which settings are used for formatting dates, times, currency, and numbers as a default for each user. Also determines the sort order for sorting text. Win:4.1
user ui language	*PlainGlobal*	<language>	Non-MUI: Same as system UI Language. • MUI: Determines the language of menus and dialogs, messages, and help files. Win:4.1
version <string>	*NamedGlobal*	<version>	Short hand for 'file version'. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
warning event log event type	*PlainGlobal*	<event log event type>	Returns an object corresponding to a warning in the event log. Warnings can be used to prevent future system problems. Win:6.0
wednesday	*PlainGlobal*	<day of week>	Returns Wednesday as a day of week object. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
week	*PlainGlobal*	<time interval>	Returns a time interval corresponding to 1 week. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
windows display time <string>	*NamedGlobal*	<time>	Returns a string that may match the time shown in the Windows file system. Win:1.2
windows file <string>	*NamedGlobal*	<file>	Returns a file object corresponding to the relative pathname (within the Windows folder) provided. See file. Win:1.2
windows folder	*PlainGlobal*	<folder>	Returns a folder object of the Windows folder This is operating system dependent. Under Win98 this is usually c:\Windows. Win:1.2
wmi	*PlainGlobal*	<wmi>	Returns the wmi object corresponding to the "root\cimv2" namespace. Win:3.0
wmi <string>	*NamedGlobal*	<wmi>	Returns the wmi object corresponding to the namespace string provided. Win:3.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
x32 application <string>	*NamedGlobal*	<application>	Returns an object corresponding to the 32-bit application specified by <string>. Win:6.0
x32 file <string>	*NamedGlobal*	<file>	Returns an object corresponding to a 32 bit file with name specified by <string>. Win:6.0
x32 folder <string>	*NamedGlobal*	<folder>	Returns a filesystem object corresponding to a 32-bit folder. Win:6.0
x32 registry	*PlainGlobal*	<registry>	Returns a 32-bit registry object. This Inspector is equivalent to the ordinary registry Inspector. Win:6.0
x64 application <string>	*NamedGlobal*	<application>	Returns an object corresponding to the 64-bit application specified by <string>. On a 32-bit computer, this is equivalent to a normal application Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. Win:6.0
x64 file <string>	*NamedGlobal*	<file>	Returns an object corresponding to a 64 bit file with pathname specified by <string>. On a 32-bit computer, this is equivalent to a normal file Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. Win:6.0
x64 folder <string>	*NamedGlobal*	<folder>	Returns a filesystem object corresponding to a 64-bit folder with the given pathname. On a 32-bit computer, this is equivalent to a normal folder Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. This action is transitive: any resulting filesystem objects will also have redirection disabled. For example, "pathnames of files of x64 folder <path>" will disable redirection when locating the folder, iterating over the files in the folder and calculating pathnames. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
x64 registry	*PlainGlobal*	<registry>	Returns a 64-bit registry object. This Inspector is for 64-bit computers only; there is no 64-bit registry on a 32-bit computer. • Note that "x64 registry" and "native registry" on 64-bit machines do NOT provide the same view as the 64-bit version of regedit (the "physical" view). If you try to access the physical location of the 32-bit view keys using a 64-bit view, it will be mapped back to the equivalent location in the 64-bit view. Win:6.0
year	*PlainGlobal*	<number of months>	Returns the specified number of years as a <number of months> type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year <integer>	*NumberedGlobal*	<year>	Creates a year object from the specified integer. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
year <string>	*NamedGlobal*	<year>	Creates a year object from the specified string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
zoned time_of_day <string>	*NamedGlobal*	<time of day with time zone>	Returns a 'time of day with time zone' object from the specified string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Examples

- `ansi code page = 1252`
  - Returns TRUE on English Windows systems.

- `description of record (oldest record number of it) of application event log`
  - Returns a description of the oldest record in the application event log.

- `binary operators "&"`
  - Returns a list of all the concatentation inspectors available.

- `binary operators returning (type "integer")`
  - Returns a list of binary operators that return an integer, including +, -, *, /, %, and other combinations.

- `bit 0 of 5`
  - Returns TRUE.

© 2007 by BigFix, Inc.

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

- `casts "integer"`
  - Returns a list of the objects that can be cast as integers, eg., <string> as integer, <integer> as integer, etc.

- `device names of connections whose (status of it = connection status connected) of network`
  - Returns the names of the connected network devices.

- `pathname of csidl folder 26`
  - Returns the path corresponding to CSIDL folder 26 (the application shared data folder, CSIDL_APPDATA).

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

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

- `current firewall profile type = firewall profile type 3`
  - Returns TRUE.

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

- `number of processors / physical processor count != logical processor count`
  - Returns TRUE if hyperthreading has not yet been turned on.

- `paths of network shares`
  - Returns a list of the paths currently being shared over the network.

© 2007 by BigFix, Inc.

Registry Objects

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

Registry

These are the Inspectors that expose the Windows registry.

Creation Methods

Key Phrase	*Form*	Description
native registry	*PlainGlobal*	On 32 bit versions of windows, this returns the same as registry32 and registry. On 64 bit versions of windows, this returns the same as registry64. Win:6.0
registry	*PlainGlobal*	Creates an object for accessing the registry. Win:1.2
x32 registry	*PlainGlobal*	Returns a 32-bit registry object. This Inspector is equivalent to the ordinary registry Inspector. Win:6.0
x64 registry	*PlainGlobal*	Returns a 64-bit registry object. This Inspector is for 64-bit computers only; there is no 64-bit registry on a 32-bit computer. • Note that "x64 registry" and "native registry" on 64-bit machines do NOT provide the same view as the 64-bit version of regedit (the "physical" view). If you try to access the physical location of the 32-bit view keys using a 64-bit view, it will be mapped back to the equivalent location in the 64-bit view. Win:6.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
application <string> of <registry>	*Named*	<application>	Returns an application object matching name provided. See application. Win:1.2
application folder <string> of <registry>	*Named*	<folder>	Returns the folder containing the matching name provided. See application. The application does not have to exist. The folder has to exist. Win:1.2
application of <registry>	*Plain*	<application>	Iterates through the properly installed applications. See application. Win:1.2
current user key <logged on user> of <registry>	*Indexed*	<registry key>	On 2000/2003/XP/Vista, this Inspector returns RegOpenCurrentUser. Under NT4, it uses HKEY_USERS\<sid>. On these systems, the Inspector may fail if run in a non-privileged context. Under Windows 9x, it returns HKEY_USERS\<username> if it exists. Otherwise it uses HKEY_USERS\Default. Win:7.0
device key <string> of <registry>	*Named*	<registry key>	Iterates through all the keys known to the configuration manager for active devices whose "DeviceDesc" matches the name provided. See registry key. Win:1.2
device key of <registry>	*Plain*	<registry key>	Iterates through all the keys known to the configuration manager for active devices. See registry key. Win:1.2
file extension <string> of <registry>	*Named*	<registry key>	Returns a key associated with the named extension. See registry key. Win:1.2
file type <string> of <registry>	*Named*	<registry key>	Returns a key associated the named file type. See registry key. Win:1.2
key <string> of <registry>	*Named*	<registry key>	Returns a key associated with the name provided. See registry key. Win:1.2

© 2007 by BigFix, Inc.

Examples

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

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

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

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

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

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

- `file type "txtfile" of the registry`

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

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

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

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

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

© 2007 by BigFix, Inc.

Registry Key

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

Creation Methods

Key Phrase	*Form*	Description
current user key <logged on user> of <registry>	*Indexed*	On 2000/2003/XP/Vista, this Inspector returns RegOpenCurrentUser. Under NT4, it uses HKEY_USERS\<sid>. On these systems, the Inspector may fail if run in a non-privileged context. Under Windows 9x, it returns HKEY_USERS\<username> if it exists. Otherwise it uses HKEY_USERS\Default. Win:7.0
device key <string> of <registry>	*Named*	Iterates through all the keys known to the configuration manager for active devices whose "DeviceDesc" matches the name provided. Win:1.2
device key of <registry>	*Plain*	Iterates through all the keys known to the configuration manager for active devices. Win:1.2
driver key of <active device>	*Plain*	The key identified by adding the value of 'driver key value name of active device' to HKLM\System\CurrentControlSet\Control\Class\. Win:1.2
driver key of <registry key>	*Plain*	Uses the value of "Driver" of the key to indirectly return a key corresponding to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\<value of Driver>. Win:1.2
file extension <string> of <registry>	*Named*	Creates a key object provided the registry indicates support for the named file extension. Win:1.2
file type <string> of <registry>	*Named*	Creates a key object provided the registry indicates support for the named file type. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
key <string> of <registry key>	*Named*	Creates an object for the named sub-key of the key. Win:1.2
key <string> of <registry>	*Named*	Creates an object for the named key. The name may be a full path to a key of the form "HKEY_CLASSES_ROOT\Fixlet.Pool\". Win:1.2
key of <registry key>	*Plain*	Iterates through the sub-keys of a key. Win:1.2

Properties

Key Phrase	*Form*	Return Type	Description
application <string> of <registry key>	*Named*	<application>	Returns the application associated with the named command. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file type. Win:1.2
application folder <string> of <registry key>	*Named*	<folder>	Returns the parent folder associated with the named application. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file type. Win:1.2
application folder of <registry key>	*Plain*	<folder>	Returns the parent folder associated with the named application. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file type. Win:1.2
application of <registry key>	*Plain*	<application>	Returns the application associated with the "open" command. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file extension. Win:1.2
default value of <registry key>	*Plain*	<registry key value>	Returns the unnamed value associated with a key as a string. It does not necessarily exist. Win:1.2
driver key of <registry key>	*Plain*	<registry key>	Normally used as a property of a device key. Looks up the value of "Driver" of the key provided to indirectly return another key corresponding to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\<value of Driver>. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
key <string> of <registry key>	*Named*	<registry key>	Returns a key for the named sub-key. Win:1.2
key of <registry key>	*Plain*	<registry key>	Iterates through the sub-keys of the key. Win:1.2
name of <registry key>	*Plain*	<string>	Returns the name of the key as a string. Win:1.2
security descriptor of <registry key>	*Plain*	<security descriptor>	Specifies the security descriptor associated with the specified registry key. Win:4.1
value <string> of <registry key>	*Named*	<registry key value>	Returns the named value stored under the key. See registry key value. Win:1.2
value of <registry key>	*Plain*	<registry key value>	Iterates through values stored under a key. Win:1.2

Note

The terminology of keys, values, default values and values that have names and data is chosen to match the convention's used by the Windows registry editor as well as the API's provided by the Windows operating system for accessing this information.
Top branches of the Windows registry include:
HKEY_CLASSES_ROOT
HKEY_LOCAL_MACHINE
HKEY_CURRENT_USER
HKEY_USERS
HKEY_CURRENT_CONFIG
HKEY_DYN_DATA
HKEY_PERFORMANCE_DATA (NT)
The trailing slashes on registry key names are optional.

© 2007 by BigFix, Inc.

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

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

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

Examples

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

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

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

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

- `file type "txtfile" of the registry`

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

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

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

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

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

© 2007 by BigFix, Inc.

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

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

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

- `exists default value of key "AppEvents\EventLabels\AppGPFault" of key "HKEY_CURRENT_USER" of registry`
  - Returns TRUE if the key exists and has a default value.

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

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

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

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

© 2007 by BigFix, Inc.

Registry Key Value

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

Creation Methods

Key Phrase	*Form*	Description
default value of <registry key>	*Plain*	Every key may have a default or unnamed value. This inspector returns the default value of the key. This value has the same properties as any other registry key value except that it does not have a name property. Win:1.2
value <string> of <registry key>	*Named*	Creates an object with the value of the key. The name property of the value will match the name provided. Win:1.2
value of <registry key>	*Plain*	Creates an object with all the values of a key. Win:1.2

Properties

Key Phrase	*Form*	Return Type	Description
<registry key value> as application	*Cast*	<application>	If the data stored in the value is a string and it is the full pathname of an application that exists on disk, the application object is returned. Win:1.2
<registry key value> as file	*Cast*	<file>	If the data stored in the value is a string and it is the full pathname of a file that exists on disk, the file object is returned. Win:1.2
<registry key value> as folder	*Cast*	<folder>	If the data stored in the value is a string and it is the full pathname of a folder that exists on disk, the folder object is returned. Win:1.2
<registry key value> as integer	*Cast*	<integer>	Returns the value stored in the registry entry provided it can be fully represented as an integer. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
<registry key value> as string	*Cast*	<string>	Returns a string if the data of the value is of type REG_SZ. Win:1.2
<registry key value> as system file	*Cast*	<file>	If the data stored in the value is a string and it is a relative pathname from the system folder of a file that exists on disk, the corresponding file object is returned. Win:1.2
<registry key value> as time	*Cast*	<time>	If the data stored in the value is a string in MIME compliant date format, this property will return a time object. If the data stored is a binary value and is 16 or more bytes in length, its first 16 bytes are interpreted as a SYSTEMTIME and the corresponding time object is returned. See time. Win:1.2
name of <registry key value>	*Plain*	<string>	Returns the name of the value as a string. (see escape of <string> for more information). Win:1.2
size of <registry key value>	*Plain*	<integer>	Returns the size of the data as an integer. Win:1.2
type of <registry key value>	*Plain*	<registry key value type>	Returns the type of the data of the value. See type of value of key or registry. Win:1.2

Operators

Key phrase	Return Type	Description
<registry key value> {cmp} <integer>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2
<registry key value> {cmp} <registry key value>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2

© 2007 by BigFix, Inc.

Key phrase	Return Type	Description
<registry key value> {cmp} <string>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2

Note

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

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

Examples

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

- `exists default value of key "AppEvents\EventLabels\AppGPFault" of key "HKEY_CURRENT_USER" of registry`
  - Returns TRUE if the key exists and has a default value.

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

© 2007 by BigFix, Inc.

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

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

- `type of default value of key "AppEvents\EventLabels\AppGPFault" of key "HKEY_CURRENT_USER of registry = "REG_SZ"`
  - Returns TRUE if the default type of the specified key is REG_SZ.

Registry Key Value Type

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

Creation Methods

Key Phrase	*Form*	Description
type of <registry key value>	*Plain*	Creates an integer designating the type of data stored in the registry key value. See the registry MS documentation for these numeric values, which correspond to the enumerated constants discussed in the "<registry key value type> as string" property. Win:1.2

Properties

Key Phrase	*Form*	Return Type	Description
<registry key value type> as string	*Cast*	<string>	Returns the type of value as a string. One of REG_SZ, REG_NONE, REG_DWORD, REG_LINK, REG_BINARY, REG_MULTI_SZ, REG_EXPAND_SZ, REG_RESOURCE_LIST, REG_DWORD_LITTLE_ENDIAN, REG_DWORD_BIG_ENDIAN, REG_FULL_RESOURCE_DESCRIPTOR, REG_RESOURCE_REQUIREMENTS_LIST. Win:1.2

© 2007 by BigFix, Inc.

Operators

Key phrase	Return Type	Description
<registry key value type> {cmp} <integer>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2
<registry key value type> {cmp} <registry key value type>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2
<registry key value type> {cmp} <string>	*<boolean>*	Returns a boolean TRUE/FALSE depending on the result of the comparison, where: • {cmp} is one of: =, !=, <, <=, >, >= . Win:1.2

© 2007 by BigFix, Inc.

Filesystem Objects

This chapter covers the keywords for extracting information from the file system, like files, drives, pathnames, folders, etc. It also includes the keywords needed to identify and compare version information of files and patches.

Filesystem Object

Properties

Key Phrase	*Form*	Return Type	Description
accessed time of <filesystem object>	*Plain*	<time>	When the filesystem object (file or folder) was last accessed. Some file systems maintain this property. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
ancestor of <filesystem object>	*Plain*	<folder>	Returns all ancestor folders (recursive parent folders) of the given filesystem object (file or folder). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
archive of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the Archive bit is turned on for the specified file or folder (filesystem object). This bit is often used by backup software. Win:6.0
compressed of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (filesystem object) has been compressed. Win:6.0
creation time of <filesystem object>	*Plain*	<time>	The date and time of creation of the specified file or folder. This corresponds to what is shown in the "Get Info" box. Win:6.0, Mac:4.1
drive of <filesystem object>	*Plain*	<drive>	Returns the drive associated with the specified file or folder (filesystem object). Win:6.0
hidden of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (filesystem object) is marked as hidden. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
location of <filesystem object>	*Plain*	<string>	Returns the name of the directory in which the file or folder (filesystem object) is located. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
modification time of <filesystem object>	*Plain*	<time>	The date and time of latest modification of the file. This corresponds to what is shown in the "Get Info" box. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
name of <filesystem object>	*Plain*	<string>	This returns the name of the file or folder. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
normal of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (filesystem object) is 'normal'. Win:6.0
offline of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (the filesystem object) is marked as 'offline'. Win:6.0
parent folder of <filesystem object>	*Plain*	<folder>	The folder containing the specified file or folder. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
pathname of <filesystem object>	*Plain*	<string>	Returns the full pathname of the specified file or folder (filesystem object) as a string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
readonly of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (the filesystem object) is marked as read-only. Win:6.0
system of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (the filesystem object) is marked as a system folder. Win:6.0
temporary of <filesystem object>	*Plain*	<boolean>	Returns TRUE if the file or folder (the filesystem object) is marked as a temporary folder. Win:6.0

© 2007 by BigFix, Inc.

Examples

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

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

File

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

Creation Methods

See application objects for additional creation methods

Key Phrase	*Form*	Description
<registry key value> as file	*Cast*	If the value contains a string and the string points to an existing file, a file object is returned. Win:1.2
<registry key value> as system file	*Cast*	If the value contains a string and the string points to an file, a file object is returned. Relative paths are interpreted relative to the system folder. Win:1.2
active device file	*PlainGlobal*	Under Windows NT, returns a list of file objects corresponding the list returned from the Windows NT EnumDeviceDrivers() function. Win:1.2
active device file <string>	*NamedGlobal*	Under Windows NT, returns a file object corresponding to the name provided. Names provided need only match the last component of the file. This inspector uses the Windows NT EnumDeviceDrivers() function. Win:1.2
descendant of <folder>	*Plain*	Returns a list of all the descendant files of the specified folder. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
file <string>	*NamedGlobal*	Returns a filesystem object corresponding to the full pathname provided in <string>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
file <string> of <folder>	*Named*	Creates the file objects corresponding to the named file within the folder. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
file of <folder>	*Plain*	Iterates through the files of a folder. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
file of <service>	*Plain*	This returns the executable file associated with the given <service> under Windows 2000 operating systems. Win:3.0
find file <string> of <folder>	*Named*	Creates an object corresponding to the files of the folder that that match the wildcard <string> provided. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
masthead of <site>	*Plain*	A copy of the masthead is maintained with the site data. This inspector returns a file object for the copy. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
system file <string>	*NamedGlobal*	Creates the file objects corresponding to the named file within the system folder. Win:1.2
system ini device file	*PlainGlobal*	Creates a list of file objects corresponding to all the device= lines of the system.ini file. Note that items whose pathnames start with '*' are not placed into this list. Win:1.2
system ini device file <string>	*NamedGlobal*	Creates a file object corresponding to a device file loaded as a result of a device= line of the system.ini file. The name provided should match the last component of the full path. Win:1.2
windows file <string>	*NamedGlobal*	Returns a file object corresponding to the relative pathname (within the Windows folder) provided. See file. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
x32 file <string>	*NamedGlobal*	Returns an object corresponding to a 32 bit file with name specified by <string>. Win:6.0
x64 file <string>	*NamedGlobal*	Returns an object corresponding to a 64 bit file with pathname specified by <string>. On a 32-bit computer, this is equivalent to a normal file Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. Win:6.0

Note

File systems that do not maintain the creation or last accessed times will often return the last modification time when queried for the creation or last accessed times or files. Modification times are preserved when files are copied. Thus, it is not uncommon to see a file that appears to have been modified before it was created.

Properties

Key Phrase	*Form*	Return Type	Description
<file> as string	*Cast*	<string>	Creates a string containing the full pathname of the specified file. See <file>. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1
byte <integer> of <file>	*Numbered*	<integer>	Returns the numeric value of the byte located at the offset specified by number within the file. Byte 0 of the file is the first byte. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
content of <file>	*Plain*	<file content>	Returns an object that can be used to search for a string in the file. See content. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
executable file format of <file>	*Plain*	<string>	Returns a four-byte string containing the format specifier for the specified file. Win:4.1
file version of <file>	*Plain*	<version>	Returns the file version extracted from the file's resource block. See version. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
first raw version block of <file>	*Plain*	<file version block>	Returns the first version block directly from a PE file. If the first block is sufficient for your purposes, use this version inspector for best speed. Win:4.1
key <string> of <file>	*Named*	<string>	Returns a key and its value from the given structured text file. It iterates over lines that start with the key name (as specified by <string>) followed by an = or : character. When searching, white space is ignored. Win:4.1, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
line <integer> of <file>	*Numbered*	<file line>	Returns the nth line (specified by <integer>) from the given file. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
line containing <string> of <file>	*Named*	<file line>	Returns all lines from the given file that contain the specified string. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
line of <file>	*Plain*	<file line>	Iterates over all the lines of the specified file. NOTE: lines are truncated to 1023 characters. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
line starting with <string> of <file>	*Named*	<file line>	Same as line <string> of <file>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:5.1
only raw version block of <file>	*Plain*	<file version block>	Returns the only version block directly from a PE file. Win:4.1
only version block of <file>	*Plain*	<file version block>	Most files only have 1 version block. This property allows language independent access when there is only one version block present. The result is the same as 'version block 1'. Win:1.2
product version of <file>	*Plain*	<version>	Returns the product version extracted from the file's resource block. See version. Win:1.2
raw file version of <file>	*Plain*	<version>	Returns the file version directly from a PE file. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
raw product version of <file>	*Plain*	<version>	Returns the product version directly from a PE file. Win:4.1
raw version block <integer> of <file>	*Numbered*	<file version block>	Returns the numbered version block directly from a PE file. Win:4.1
raw version block <string> of <file>	*Named*	<file version block>	Returns the named version block directly from a PE file. Win:4.1
raw version block of <file>	*Plain*	<file version block>	Returns the version block directly from a PE file. Win:4.1
raw version of <file>	*Plain*	<version>	Returns the version directly from a PE file. Win:4.1
section <string> of <file>	*Named*	<file section>	Returns a named section of a file. Useful for locating sections of 'ini' files. Section names are delimited by square bracket characters '[section name]'. See examples below. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
security descriptor of <file>	*Plain*	<security descriptor>	Specifies the security descriptor associated with the specified file. Win:4.1
sha1 of <file>	*Plain*	<string>	Returns the sha1 checksum of the file hex encoded as a 40 character long string. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
shortcut of <file>	*Plain*	<file shortcut>	Returns the properties and locates the target of a file shortcut: • pathname (string) • start in pathname (string) • argument string (string) • icon pathname (string) • icon index (integer). Win:1.2
size of <file>	*Plain*	<integer>	Returns the size in bytes of a file. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
variable of <file>	*Plain*	<string>	Returns the names of variables contained in an INF style file, in the format [section].name=value. Win:4.1, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
version block <integer> of <file>	*Numbered*	<file version block>	You can identify the particular version block you want to access by ordinal number. Win:1.2
version block <string> of <file>	*Named*	<file version block>	You can identify the particular version block you are looking up by name. The name you provide should match the id string of the version block. Win:1.2
version block of <file>	*Plain*	<file version block>	Iterates through the version blocks of a file. Win:1.2
version of <file>	*Plain*	<version>	Synonym for file version of <file>. Win:1.2, Mac:4.1
xml document of <file>	*Plain*	<xml dom document>	Returns an XML Document Object Model (DOM) for the specified file. Win:5.1

Note

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

Note

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

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

© 2007 by BigFix, Inc.

Examples

- `Number of find files "siteico*.bmp" of client folder of current site = 3`
  - Returns TRUE if there are 3 files matching the wildcard pattern siteico*.bmp.

- `names of find files "*.exe" of windows folder`
  - Returns a list of the names of all the executable programs in the Windows folder.

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

- `exists windows file "command.com"`
  - Verifies the existence of the named file in the Windows folder.

- `byte 0 of file "C:\test.txt"`
  - Returns the first byte in the specified file.

- `content of file "oeminfo.ini" of system folder as lowercase contains "dell"`
  - Returns TRUE if the specified file contains the string "dell" anywhere in the file.

- `executable file format of client`
  - Returns a string like "PE%00%00".

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

- `product version of file "qna.exe" of parent folder of regapp "bigfix.exe" = product version of regapp "bigfix.exe"`
  - Verifies the existence a co-executable located in the same folder with the proper version.

- `product version of regapp "bigfix.exe" > version "1.0.21"`
  - Returns TRUE if the application has a version of 1.0.22 or higher, and FALSE if the application has a version of 1.0.21 or less.

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

© 2007 by BigFix, Inc.

- `version of file (pathname of shortcut of file "BigFix.lnk" of folder "c:\Documents and Settings\All Users\Desktop")`
  - Returns the version number of the application to which the shortcut points.

Application

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

Creation Methods

Key Phrase	*Form*	Description
<registry key value> as application	*Cast*	If the value is of type string, and the string is a full pathname to an executable that exists on disk, an application object is created. Win:1.2
application <string>	*NamedGlobal*	Creates an application object for the name provided. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
application <string> of <folder>	*Named*	As with the file object, you can create an application object by naming it relative to its parent folder. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
application <string> of <registry key>	*Named*	Creates the application object associated with the named command. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file type. Win:1.2
application <string> of <registry>	*Named*	Creates the application object associated with the name provided. The name provided must be the name of a sub-key of the 'App Paths' registry key. See notes. Win:1.2
application of <registry key>	*Plain*	Creates the application object associated with the "open" command. Normally used with a sub-key of key HKEY_CLASSES_ROOT whose name is a file extension. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
application of <registry>	*Plain*	Iterates through the 'App Paths' registry key creating objects for the applications that exist. See notes. Win:1.2
default web browser	*PlainGlobal*	Returns the application object (typically the web browser) associated with HTML documents. Win:3.0
recent application	*PlainGlobal*	Iterates through the list of recently executed applications, creating application objects. This includes the list of all currently running applications. Win:1.2, Lin:6.0
recent application <string>	*NamedGlobal*	If named application has been executed recently, this inspector creates an application object. Only specify the last component of the filename. Win:1.2, Lin:6.0
regapp	*PlainGlobal*	Iterates through the applications of the registry. The applications will be those associated with the sub-keys of the 'App Paths' registry key. See notes. Win:1.2
regapp <string>	*NamedGlobal*	Returns an application object for the name provided. On a Macintosh, returns a file that has been registered with Launch Services. See application and regapp. Win:1.2
running application	*PlainGlobal*	Iterates through the list of running applications. Win:1.2, Lin:6.0
running application <string>	*NamedGlobal*	If the named application is currently executing then this inspector creates an application object. Only specify the last component of the file name. Win:1.2, Lin:6.0
x32 application <string>	*NamedGlobal*	Returns an object corresponding to the 32-bit application specified by <string>. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
x64 application <string>	*NamedGlobal*	Returns an object corresponding to the 64-bit application specified by <string>. On a 32-bit computer, this is equivalent to a normal application Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. Win:6.0

Note

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

Examples

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

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

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

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

- `name of application "print" of key "HKEY_CLASSES_ROOT\.gif" of registry`

Returns the name of the application currently responsible for printing gif files.

- `name of application "print" of key "HKEY_CLASSES_ROOT\giffile" of the registry`

The same as above when the default value of the key HKEY_CLASSES_ROOT\.gif contains giffile. These two examples demonstrate the method used by Windows to maintain file associations in the registry.

- `application of key "HKEY_CLASSES_ROOT\mailto" of the registry`

This example returns the application responsible for handling mailto requests in your web browser.

© 2007 by BigFix, Inc.

- `names of regapps`

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

- `byte 0 of regapp "bigfix.exe" = 77`

TRUE if the first byte in the specified file is ASCII 77.

- `regapp "IEXPLORE.EXE"`

Returns the application object associated with the named registry key. Checks to see if the executable exists and if so, returns the application object. Case is ignored.

- `exists running application whose (name of it as lowercase is "winword.exe")`

Returns TRUE if Microsoft Word is currently executing.

Folder

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

Creation Methods

Key Phrase	*Form*	Description
<registry key value> as folder	*Cast*	If the value in the registry is a string, and the string points to an existing folder, a folder object is returned. Win:1.2
ancestor of <filesystem object>	*Plain*	Returns all ancestor folders (recursive parent folders) of the given filesystem object (file or folder). Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
application folder <string> of <registry key>	*Named*	Synonym for pathname of parent folder of regapp <string>. Win:1.2
application folder <string> of <registry>	*Named*	Creates a folder object for the name given. Name is used to search through AppPaths of the registry. Application doesn't have to exist. Folder must exist. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
application folder of <registry key>	*Plain*	Creates a folder object for the name given. If the registry key has a "shell\open\command\" subkey and the unnamed value points to an executable, this will return the parent folder of the executable if the application and folder exist. Win:1.2
client folder of <site>	*Plain*	Creates an object corresponding to the folder on the client where site data is gathered. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
csidl folder <integer>	*NumberedGlobal*	Returns the csidl folder corresponding to the specified integer. The windows SHGetSpecialFolderLocation API is used to look up paths to special folders, which are identified by passing the specified integer as the second argument of the API call. These values and their meaning are described in the windows ShlObj.h include file found in the development sdk. • Note that some of these folders do not exist in the Local System context. Win:7.0
descendant folder of <folder>	*Plain*	Returns the descendant folders, recursively, of the given folder. The folder equivalent of "descendants of <folder>". Win:7.0
folder <string>	*NamedGlobal*	Creates a folder object for the named folder. This is a global property. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
folder <string> of <drive>	*Named*	Creates a folder object for the name provided if it exists on the drive provided. Win:1.2
folder <string> of <folder>	*Named*	Creates a folder object for the named sub-folder. Trailing slashes should be omitted from the name. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
folder of <folder>	*Plain*	Iterates through the sub-folders of the folder object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
install folder <integer>	*NumberedGlobal*	Creates a folder object corresponding to the number provided. The placement of some system folders can be found using numbers that have been associated with those folders. See notes. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
parent folder of <filesystem object>	*Plain*	The folder containing the specified file or folder. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:4.1
root folder of <drive>	*Plain*	Creates a folder object for the root of the given drive. Win:1.2
system wow64 folder	*PlainGlobal*	Returns a filesystem object corresponding to a "Windows On Windows 64" system folder, which does not exist on 32-bit Windows. You can find out more about the WOW64 system folder at the Microsoft site: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/getsystemwow64directory.asp. Win:6.0
system x32 folder	*PlainGlobal*	Returns a filesystem object corresponding to a 32-bit system folder. On a 32-bit machine, this is equivalent to the normal system folder. Win:6.0
system x64 folder	*PlainGlobal*	Returns a filesystem object corresponding to a 64-bit system folder. This is the same as the system folder, but with file system redirection disabled. For more information about file redirection, see the Microsoft site http://msdn.microsoft.com/library/default.asp?url=/library/en-us/win64/win64/file_system_redirector.asp. Win:6.0
windows folder	*PlainGlobal*	Creates a folder object of the Windows folder. This is operating system dependent. Under Win98 this is usually c:\Windows. Win:1.2
x32 folder <string>	*NamedGlobal*	Returns a filesystem object corresponding to a 32-bit folder with the specified pathname. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
x64 folder <string>	*NamedGlobal*	Returns a filesystem object corresponding to a 64-bit folder with the given pathname. On a 32-bit computer, this is equivalent to a normal folder Inspector, but on a 64-bit machine, this Inspector returns an object that has filesystem redirection disabled. This action is transitive: any resulting filesystem objects will also have redirection disabled. For example, "pathnames of files of x64 folder <path>" will disable redirection when locating the folder, iterating over the files in the folder and calculating pathnames. Win:6.0

Properties

Key Phrase	*Form*	Return Type	Description
application <string> of <folder>	*Named*	<application>	Returns an application object for the named file located in the folder. See application. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
descendant folder of <folder>	*Plain*	<folder>	Returns the descendant folders, recursively, of the given folder. The folder equivalent of "descendants of <folder>". Win:7.0
descendant of <folder>	*Plain*	<file>	Returns a list of all the descendant files of the specified folder. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:6.0
file <string> of <folder>	*Named*	<file>	Returns a file object for the named file located in the folder. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
file of <folder>	*Plain*	<file>	Iterates through the files of a folder returning file objects. When combined with a whose clause you can select files with specific properties. See file. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
find file <string> of <folder>	*Named*	<file>	Iterates through the files of a folder returning file objects whose name matches the search string provided in the name parameter. See example below. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
folder <string> of <folder>	*Named*	<folder>	Returns a folder object for the named sub-folder. Trailing slashes should be omitted from the name. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
folder of <folder>	*Plain*	<folder>	Iterates through the folders of a folder returning folder objects. When combined with a whose clause, you can select folders with specific properties. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
security descriptor of <folder>	*Plain*	<security descriptor>	Specifies the security descriptor associated with the specified folder. Win:4.1

Note

Folder and file names may be case sensitive. Use “as uppercase” or “as lowercase” if you don’t know the actual case when making comparisons. Be careful not to iterate through folders that contain a large number of files. Counting files in such a folder can be slow. Always try to use the most efficient techniques to minimize the client overhead. Consider using the “find file” Inspector which allows you to filter sets of files by using the wildcard.

Examples

- `pathname of csidl folder 26`
- Returns the path corresponding to CSIDL folder 26 (the application shared data folder, CSIDL_APPDATA).

- `exists folder "c:\program files"`
- Checks for the existence of the program files folder.

- `exists folder "fonts" of the windows folder`
- Returns TRUE if fonts is a subdirectory of the Windows directory.

- `install folder 11`
- Returns a folder object for system folder identified with this number.

- `exists file whose (name of it contains ".pdf") of folder "name"`
- Returns TRUE if some file in the folder has a name including the string ".pdf".

© 2007 by BigFix, Inc.

Drive

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

Creation Methods

Key Phrase	*Form*	Description
drive	*PlainGlobal*	Iterates through all valid drives on the system. Win:1.2
drive <string>	*NamedGlobal*	Creates the drive object for the name specified. Win:1.2
drive of <filesystem object>	*Plain*	Returns the drive associated with the specified file or folder (filesystem object). Win:6.0

Properties

Key Phrase	*Form*	Return Type	Description
file system type of <drive>	*Plain*	<string>	Value as reported by GetVolumeInformation. Win:1.2
file_supports_encryption of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
file_supports_object_ids of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
file_supports_reparse_points of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
file_supports_sparse_files of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
file_volume_quotas of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	***Form***	**Return Type**	**Description**
folder <string> of <drive>	*Named*	<folder>	Returns a folder object corresponding to the name given provided that folder exists on the drive. Win:1.2
free space of <drive>	*Plain*	<integer>	Returns the number of unused bytes of storage for the drive. (Only available for fixed disks). Win:1.2
fs_case_is_preserved of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
fs_case_sensitive of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
fs_file_compression of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
fs_persistent_acls of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
fs_unicode_stored_on_disk of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
fs_vol_is_compressed of <drive>	*Plain*	<boolean>	TRUE if bit is returned by GetVolumeInformation. Win:1.2
name of <drive>	*Plain*	<string>	Returns the name of the drive. Names look like 'c:' and 'D:'. Win:1.2
numeric type of <drive>	*Plain*	<integer>	Returns the type of drive as an integer. Win:1.2
root folder of <drive>	*Plain*	<folder>	Returns the folder corresponding to the root of the drive. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
total space of <drive>	*Plain*	<integer>	Returns the size in bytes of the drive. (Only available for fixed disks). Win:1.2
type of <drive>	*Plain*	<string>	Returns the type of drive as a string. Win:1.2

Note

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

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

Examples

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

© 2007 by BigFix, Inc.

- `type of drive of the system folder = "DRIVE_FIXED"`
- Returns TRUE if the system folder is on a fixed disk drive.

File Shortcut

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

Creation Methods

Key Phrase	*Form*	Description
shortcut of <file>	*Plain*	Creates a shortcut object for the file. If the file is not a shortcut, this property does not exist. Win:1.2

Properties

Key Phrase	*Form*	Return Type	Description
argument string of <file shortcut>	*Plain*	<string>	Returns the arguments that are passed to the application to which the shortcut points when the user attempts to open the shortcut. Win:1.2
icon index of <file shortcut>	*Plain*	<integer>	The index number of the icon in the file containing the icon associated with the shortcut. Win:1.2
icon pathname of <file shortcut>	*Plain*	<string>	The full path name of the file containing the icon associated with the shortcut. Win:1.2
pathname of <file shortcut>	*Plain*	<string>	Returns the full path name of the object to which the shortcut points. Win:1.2
start in pathname of <file shortcut>	*Plain*	<string>	Returns the full path name the system sets the current directory when the user launches the shortcut. Win:1.2

© 2007 by BigFix, Inc.

Examples

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

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

File Section

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

Creation Methods

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

Properties

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

Note

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

© 2007 by BigFix, Inc.

Examples

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

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

File Content

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

Operators

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

Note

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

Examples

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

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

Version

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

Creation Methods

Key Phrase	*Form*	Description
<string> as version	*Cast*	Turns a string into a version object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1
<version> as version	*Cast*	Reflexive cast of version. Win:1.2
file version of <file>	*Plain*	Creates a version object associated with the FILEVERSION property of the file. Win:1.2
pad of <version>	*Plain*	Creates a version object which is padded with zero values. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
product version of <file>	*Plain*	Creates a version object associated with the PRODUCTVERSION property of the file. Win:1.2
raw file version of <file>	*Plain*	Same as file version, but allows a workaround for anomalous behavior on Windows systems with the Windows language pack installed (the MUI). Win:4.1
raw product version of <file>	*Plain*	Same as product version, but allows a workaround for anomalous behavior on Windows systems with the Windows language pack installed (the MUI). Win:4.1
raw version of <file>	*Plain*	Same as version, but allows a workaround for anomalous behavior on Windows systems with the Windows language pack installed (the MUI). Win:4.1
version <string>	*NamedGlobal*	Creates a version object corresponding to the name provided. Syntax: version "1.2". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
version of <current relay>	*Plain*	Returns a version object that is the version of the server or relay that the client last registered with. This may be a BES Relay or the BES root server. Win:7.0
version of <file>	*Plain*	Shorthand for file version of <file>. Win:1.2, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<version> as string	*Cast*	<string>	Turns a version type into a string of the form "1.2.3.4". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
<version> as version	*Cast*	<version>	Reflexive cast of version. Win:1.2
pad of <version>	*Plain*	<version>	Returns a version object which is padded with zero values. Win:1.2

© 2007 by BigFix, Inc.

Operators

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

Note

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

Examples

- `"MyApp 1.2" as version = version "1.2"`

▸ The version cast looks through the string until it identifies something that can be interpreted as a version. This is convenient for extracting version numbers out of strings that contain both version numbers and textual description.

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

▸ Returns TRUE if the dll has the specified version number.

- `pad of version "1.2" = version "1.2.0.0"`

▸ Returns TRUE.

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

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

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

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

© 2007 by BigFix, Inc.

- `product version of regapp "bigfix.exe" > version "1.0.21"`
  - Returns TRUE if the application has a version of 1.0.22 or higher, and FALSE if the application has a version of 1.0.21 or less.

- `version of file "Winsock.dll" of windows folder = "4.0.0.1111"`
  - The plain version inspector is a shorthand for file version.

- `version of regapp "bigfix.exe" as string = "1.0.45.0"`
  - Returns TRUE if the BigFix application has the specified version.

File Version Block

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

Creation Methods

Key Phrase	*Form*	Description
first raw version block of <file>	*Plain*	Returns the first version block directly from a PE file. If the first block is sufficient for your purposes, use this version inspector for best speed. Win:4.1
only raw version block of <file>	*Plain*	Returns the only version block directly from a PE file. Win:4.1
only version block of <file>	*Plain*	Most applications only have 1 version block. This inspector allows language independent access when there is only one version block present. Win:1.2
raw version block <integer> of <file>	*Numbered*	Returns the numbered version block directly from a PE file. Win:4.1
raw version block <string> of <file>	*Named*	Returns the named version block directly from a PE file. Win:4.1
raw version block of <file>	*Plain*	Returns the version block directly from a PE file. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
version block <integer> of <file>	*Numbered*	You can identify the particular version block you are looking up by ordinal number. 'Version block 1' is equivalent to 'Only Version block'. Win:1.2
version block <string> of <file>	*Named*	You can identify the particular version block you are looking up by name. The name you provide should match the id string of the version block. Win:1.2
version block of <file>	*Plain*	Iterates through the version blocks of a file. Win:1.2

Properties

Key Phrase	*Form*	Return Type	Description
codepage of <file version block>	*Plain*	<string>	A string representation of the codepage portion of the id of this version block. See notes for known codepage strings. For example, 'Unicode'. Win:1.2
id of <file version block>	*Plain*	<string>	A string representation containing both the language and codepage of this version block. The format is 8 hex digits, 4 of the codepage concatenated with 4 of the language. For example, '040904b0'. See notes for known values. Win:1.2
language of <file version block>	*Plain*	<string>	A string representation of the language portion of the id of this version block. For example, 'English (United States)'. See notes for known values. Win:1.2
value <string> of <file version block>	*Named*	<string>	Returns a string corresponding to the name provided. Values have names such as 'CompanyName', 'FileDescription', 'FileVersion'. Win:1.2

© 2007 by BigFix, Inc.

Note

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

value:

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

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

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

id:

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

language:

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

© 2007 by BigFix, Inc.

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

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

© 2007 by BigFix, Inc.

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

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

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

Examples

- `value "CompanyName" of version block 1 of regapp "bigfix.exe" = "BigFix Inc."`
  - Returns TRUE if the "CompanyName" value of the given file's version block equals the specified string.

- `exists version block "040904B0" of regapp "bigfix.exe"`
  - Returns TRUE if the designated version block exists. The case of the name of the version block is ignored.

- `codepage of only version block of regapp "bigfix.exe" is "Unicode"`
  - Returns TRUE if the version block codepage for the specified file is unicode.

- `id of only version block of regapp "bigfix.exe" is "040904b0"`
  - Returns TRUE if the given file's version block id is the specified string.

- `language of version block 1 of regapp "bigfix.exe" = "English (United States)"`
  - Returns TRUE if the given file's version block language is as specified.

- `value "FileVersion" of version block 1 of regapp "bigfix.exe" as version`
  - When casting a string value to a version, the parser skips through the string until it identifies something that can be interpreted as a version. This is convenient for extracting version numbers from strings containing added text.

© 2007 by BigFix, Inc.

File Line

A file line is a string from a text file.

Creation Methods

Key Phrase	*Form*	Description
line <integer> of <file>	*Numbered*	Returns the nth line in a file. A file line is just a string, except that you can use the additional properties "next line" and "previous line". Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
line containing <string> of <file>	*Named*	Returns the line with the specified search string in the given file. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
line of <file>	*Plain*	Returns the lines of a specified file. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
line starting with <string> of <file>	*Named*	Returns a line from the given file beginning with the specified phrase. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
next line of <file line>	*Plain*	Returns the line after the specified line in a file (provided that it is not the last line). This Inspector can be chained indefinitely, eg., next line of next line of …. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
previous line of <file line>	*Plain*	Returns the line before the nth line in a file, provided n>1. You may repeat this command up to three times. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
line number of <file line>	*Plain*	<integer>	Returns the line number of a given line. Can be used to locate specific lines in a file. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
next line of <file line>	*Plain*	<file line>	Returns the line after the specified line in a file (provided that it is not the last line). This Inspector can be chained indefinitely, eg., next line of next line of …. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1
previous line of <file line>	*Plain*	<file line>	Returns the line before the nth line in a file, provided n>1. You may repeat this command up to three times. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:5.1

Examples

- `line 2 of (file "printmon.inf" of system folder)`
  - Returns the second line of the specified file.

- `lines containing "Signature]" of file "mmdriver.inf" of system folder`
  - Returns a list corresponding to the top-level sections involving signatures in the specified .inf file.

- `lines of file "mmdriver.inf" of system folder`
  - Returns a list of the lines of the specified file.

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

- `next line of line containing "[mciavi]" of file "mmdriver.inf" of system folder`
  - Returns the line after the one containing "[mciavi]", which is a string such as "1:MSVFW32.DLL".

- `previous line of previous line of previous line of line containing "[mciavi]" of file "mmdriver.inf" of system folder`
  - Returns the line 3 lines previous to the one containing the specified phrase in the given file.

© 2007 by BigFix, Inc.

- `line number of line containing "[mciavi]" of file "mmdriver.inf" of system folder`
  - Returns the line number of the specified line in the given file.

- `next line of line containing "[mciavi]" of file "mmdriver.inf" of system folder`
  - Returns the line after the one containing "[mciavi]", which is a string such as "1:MSVFW32.DLL".

- `previous line of previous line of previous line of line containing "[mciavi]" of file "mmdriver.inf" of system folder`
  - Returns the line 3 lines previous to the one containing the specified phrase in the given file.

Xml Dom Document

These are the Inspectors for the XML Document Object Module (DOM) for specified XML files. XML dom document objects are derived from XML dom node objects. 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
owner document of <xml dom node>	*Plain*	Returns the name of the document that contains the specified node. Win:5.1
xml document of <file>	*Plain*	Returns the XML Document Object Module (DOM) for the specified file. Win:5.1
xml document of <string>	*Plain*	Returns an XML document object from the given <string>, typically a file name. Win:6.0

Examples

- `xml document of file "c:\test.xml" as xml`
  - Returns the test.xml document in a form like: <?xml version="1.0"?>%0d%0a<a:Books xmlns:a="x-schema:bookschema.xml">%0d%0a%09<a:Book>%0d%0a%09%09<title>Presenting XML</title>%0d%0a%09%09<author>Richard Light</author>%0d%0a%09</a:Book>%0d%0a</a:Books>%0d%0a.

© 2007 by BigFix, Inc.

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:5.1
attribute <string> of <xml dom node>	*Named*	Returns the named attribute of the specified node. Win:5.1
attribute of <xml dom node>	*Plain*	Returns the attribute(s) of the specified XML DOM node. Win:5.1
child node <integer> of <xml dom node>	*Numbered*	Returns the child node by number. Win:5.1
child node of <xml dom node>	*Plain*	By chaining this Inspector, you can find the child nodes of any given node. Win:5.1
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:5.1
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:5.1
next sibling of <xml dom node>	*Plain*	Returns the next child node after the current one. Win:5.1
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:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
previous sibling of <xml dom node>	*Plain*	Returns the child node before the one specified. Win:5.1
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:6.0
xpath <(string, string)> of <xml dom node>	*Indexed*	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:6.0
xpath <string> of <xml dom node>	*Named*	Returns an iterated list of matching xml dom nodes, given the xpath query specified by <string>. Win:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<xml dom node> as text	*Cast*	<string>	Casts an xml document object module node as text. Win:6.0
<xml dom node> as xml	*Cast*	<string>	Casts an xml document object module node as xml. Win:6.0
attribute <integer> of <xml dom node>	*Numbered*	<xml dom node>	Returns the numbered attribute of the specified XML DOM node. Win:5.1
attribute <string> of <xml dom node>	*Named*	<xml dom node>	Returns the named attribute of the specified node. Win:5.1
attribute of <xml dom node>	*Plain*	<xml dom node>	Returns the attribute(s) of the specified XML DOM node. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
child node <integer> of <xml dom node>	*Numbered*	<xml dom node>	Returns the child node by number. Win:5.1
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:5.1
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:5.1
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:5.1
next sibling of <xml dom node>	*Plain*	<xml dom node>	Returns the next child node after the current one. Win:5.1
node name of <xml dom node>	*Plain*	<string>	Returns the name of the specified XML DOM node as a string. Win:5.1
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:5.1
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:5.1
owner document of <xml dom node>	*Plain*	<xml dom document>	Returns a document belonging to the owner of the specified node. Win:5.1
parent node of <xml dom node>	*Plain*	<xml dom node>	Returns the parent node of the specified node. Win:5.1
previous sibling of <xml dom node>	*Plain*	<xml dom node>	Returns the child node before the one specified. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
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:6.0
xpath <(string, string)> of <xml dom node>	*Indexed*	<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:6.0
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:6.0

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>
```

© 2007 by BigFix, Inc.

Examples

- `node names of child nodes of child node 1 of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns a list of the names of the children of the first node in the specified document.

- `node names of child nodes of child node 1 of last child of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns the names of the specified chain of child nodes.

- `node name of next sibling of first child of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns the name of the second node of the specified file.

- `xpath ( "xmlns:a='x-schema:bookschema.xml'", "/a:Books/a:Book" ) of xml document of file "c:\test.xml" as xml`
  - Returns an xml dom node such as <a:Book xmlns:a="x-schema:bookschema.xml">%0d%0a%09<title>Presenting XML</title>%0d%0a%09<author>Richard Light</author>%0d%0a</a:Book>.

- `node value of attribute 0 of child node 0 of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns the value of the first attribute of the first node of the specified document.

- `node value of attribute "xmlns" of child node 1 of xml document of (file "icsxml\cmnicfg.xml" of system folder)`
  - Returns the value of the named attribute (xmlns) of the specified file.

- `node names of attributes of child nodes of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns the names of the attributes of each node in the specified file.

- `node names of child nodes of child node 1 of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns a list of the names of the children of the first node in the specified document.

- `node names of child nodes of child node 1 of last child of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns the names of the specified chain of child nodes.

- `node name of next sibling of first child of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns the name of the second node of the specified file.

© 2007 by BigFix, Inc.

- `node names of child nodes of xml document of file "C:\WINDOWS\system32\icsxml\cmnicfg.xml"`
  - Returns a list of the names of each node in the specified XML document.

- `node types of child nodes of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns a list of numeric types for each of the nodes in the specified XML document.

- `node value of first child of xml document of file "icsxml\cmnicfg.xml" of system folder`
  - Returns the value of the first node in the specified file. If the first statement of the file is <xml version="1.0">, for instance, the name would be "xml" and the value would be version="1.0".

Application Usage Summary

To enable these Inspectors, you first need to create the client setting _BESClient_UsageManager_EnableAppUsageSummary and initialize it to 1. You can also limit the summary to a subset of applications by creating _BESClient_UsageManager_EnableAppUsageSummaryApps and initializing it to a list of apps to include (or exclude). The value of this setting should look like +:app1:app2:app3: to add apps to the scope, and -:app1:app2: to exclude apps. The case is ignored. For instance, to enable summaries on the Word application, use the value +:winword.exe:.

Creation Methods

Key Phrase	*Form*	Description
application usage summary	*PlainGlobal*	Returns an application usage summary containing information including the start time, duration and other statistics on client applications. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
application usage summary <string>	*NamedGlobal*	Returns the usage summary for the application specified in <string>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
first start time of <application usage summary>	*Plain*	<time>	Returns the start time of the specified application since the computer was configured to track it, regardless of reboots. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

© 2007 by BigFix, Inc.

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

Examples

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

© 2007 by BigFix, Inc.

System Objects

These are the keywords available for querying various aspects of the system, including the name and version of the operating sytem. This chapter also covers the keywords used to describe the vendors and types of the various processors that coexist in a typical computer system. Some of these Inspectors are system-specific, but are included to provide cross-platform compatiblity.

Bios

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

Creation Methods

Key Phrase	*Form*	Description
bios	*PlainGlobal*	Returns the date of the bios if it exists, or <unknown> if it does not exist. This is a Windows-only command. On a non-Windows system, bios returns False. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:5.1

Properties

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

© 2007 by BigFix, Inc.

Examples

- `date of bios = "09/16/97"`
  - Returns TRUE if the BIOS date matches the value provided.

- `following text of last "/" of date of bios`
  - Returns the year of the bios as a string. For "09/07/99" it would return "99".

- `preceding text of first "/" of date of bios`
  - Returns the month of the bios date as a string.

- `preceding text of first "/" of following text of first "/" of date of bios`
  - Returns the day of the bios date as a string.

- `version of bios as lowercase contains "phoenix"`
  - Returns TRUE if the version string contains "phoenix", "PHOENIX" or "Phoenix".

Operating System

The operating system object provides access to several important properties of the system.

Creation Methods

Key Phrase	*Form*	Description
operating system	*PlainGlobal*	Creates the global operating system object. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<operating system> as string	*Cast*	<string>	Returns a string containing the name of the operating system concatenated with the release. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
boot time of <operating system>	*Plain*	<time>	Returns the time of the last restart. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
build number high of <operating system>	*Plain*	<integer>	Numeric representation of the most significant 16 bits of the build number. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
build number low of <operating system>	*Plain*	<integer>	Numeric representation of the least significant 16 bits of the build number. Win:1.2
build number of <operating system>	*Plain*	<integer>	Returns the integer build number of the operating system. • Note: On the Macintosh, this returns a string as of BES version 5.1. Win:1.2, Mac:4.1
csd version of <operating system>	*Plain*	<string>	Returns the Corrective Service Disk version of the operating system. The szCSDVersion as returned by the GetVersionEx system call. The format varies depending on the installed service packs. For WinNT it contains a string such as "Service Pack 3", for Win95 it can contain a string such as "B". Win:1.2
ia64 of <operating system>	*Plain*	<boolean>	Returns TRUE iff the BES Client is running on Itanium. Win:7.0
major version of <operating system>	*Plain*	<integer>	Returns integer which is the dwMajorVersion returned by the GetVersionEx system call. Note that while the WinNT major version tracks the release (3 for 3.51, 4 for 4.0, and 5.0 for Windows 2000, 5.1 for Windows XP), the major version for Win95 and Win98 is always 4. Win:1.2
metric <integer> of <operating system>	*Numbered*	<integer>	This inspector uses the windows GetSystemMetrics API. The integer constants and their meaning are defined by Microsoft. For example, the integer 87 indicates that the operating system is a media center edition. The integer constants are defined in WinUser.h. Win:6.0
minor version of <operating system>	*Plain*	<integer>	Numeric representation of the minor version of the operating system. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
name of <operating system>	*Plain*	<string>	Returns the name of the operating system as a string. Names might include Win98, WinNT, etc. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
performance counter frequency of <operating system>	*Plain*	<hertz>	The rate at which the performance counter is being incremented (per second). Win:1.2
performance counter of <operating system>	*Plain*	<integer>	Retrieves a 64-bit performance counter value. Win:1.2
platform id of <operating system>	*Plain*	<integer>	Returns the dwPlatformId as returned by the GetVersionEx system call. Possible values are 1 (Win95/95) and 2 (WinNT). Win:1.2
product type of <operating system>	*Plain*	<operating system product type>	Returns the product type of the operating system, which includes Workstations, Domain Controllers and Servers. Win:6.0
release of <operating system>	*Plain*	<string>	Information about the release of the operating system, typically formatted as <Major version>.<Minor version>. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
service pack major version of <operating system>	*Plain*	<integer>	Returns the major version number of the current service pack of the specified OS. Win:6.0
service pack minor version of <operating system>	*Plain*	<integer>	Returns the minor version number of the current service pack of the specified OS. Win:6.0
suite mask of <operating system>	*Plain*	<operating system suite mask>	Returns the bit-mapped suite mask for the operating system, which contains further fine-grain information about the version. Win:6.0
uptime of <operating system>	*Plain*	<time interval>	Returns a time interval that represents the elapsed time since the operating system was last booted. • Note: Depending on the Laptop, this interval may not include time spent in hibernation. Win:5.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1, Mac:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
x64 of <operating system>	*Plain*	<boolean>	Returns TRUE if the current operating system is 64-bits. Win:6.0

Examples

- `now - boot time of operating system > week`
  - Returns TRUE if the computer hasn't been rebooted for over a week.
- `build number high of operating system = 1027`
  - Returns TRUE if the high word of the build number = 0403 hex.
- `build number low of operating system = 1212`
  - Returns TRUE if the low word of the build number = 04BC hex.
- `build number of operating system = 67306684`
  - Returns TRUE if the build number = 040304BC in hexadecimal.
- `csd version of the operating system = "B"`
  - Returns TRUE on a Win95 System with Corrective Service Disk version = "B".
- `major version of operating system = 4`
  - Returns TRUE if the major version (before the dot) is 4, such as 4.1, 4.2, etc.
- `Metric 87 of operating system`
  - Returns TRUE if the OS is a Media Center Edition.
- `minor version of operating system = 0`
  - Returns TRUE if the minor part of a version number (after the dot) is 0, such as 4.0, 5.0, etc.
- `name of operating system = "WinXP"`
  - Returns TRUE on a WinXP System.
- `platform id of operating system = 1`
  - Returns TRUE on a Win95 System.
- `release of operating system = "OSR2.1"`
  - Returns TRUE if the Win95 computer is running under operating system release 2.5.

© 2007 by BigFix, Inc.

Processor

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

Creation Methods

Key Phrase	*Form*	Description
main processor	*PlainGlobal*	Creates the object associated with the 'Primary' processor. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
processor	*PlainGlobal*	Iterates through the processors in the system. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
processor <integer>	*NumberedGlobal*	Creates the processor object for the number specified. The first processor is processor number 1. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1

Properties

Key Phrase	*Form*	Return Type	Description
brand id of <processor>	*Plain*	<integer>	This inspector returns the integer known as the brand id, returned from the assembly language cpuid extended instruction. Win:6.0
brand string of <processor>	*Plain*	<string>	Returns the vendor-defined brand names for newer processors. Win:1.2
extended family of <processor>	*Plain*	<integer>	Integer representing the extended family of CPU. See the notes for the meaning of these numbers. Win:1.2
extended model of <processor>	*Plain*	<integer>	Integer representing the extended model of CPU. See the notes for the meaning of these numbers. Win:1.2
family name of <processor>	*Plain*	<string>	Returns the family name of the CPU, dependent on the type of client computer, for instance Pentium, Sparc, PowerPC G4, etc. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
family of <processor>	*Plain*	<integer>	Returns an integer representing the family of the CPU. See the notes for the meaning of these numbers. • Note: As of BES 6.0, this Inspector returns a string on Unix computers. Win:1.2, Lin:3.1, Sol:4.1, AIX:4.1
feature mask of <processor>	*Plain*	<integer>	Returns the feature flags from the CPUID instruction. The feature mask contains bits that identify extra features the processor may provide such as MMX support or if the Processor ID is enabled on the processor. Win:1.2
model of <processor>	*Plain*	<integer>	Returns the model number of the CPU. • Note: On Unix computers, this Inspector returns a <string> as of BES 6.0. Win:1.2, Lin:3.1, Sol:4.1, HPUX:4.0, AIX:4.1
speed of <processor>	*Plain*	<hertz>	Returns the speed of the processor in Hertz. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
stepping of <processor>	*Plain*	<integer>	Returns the stepping number of the processor. This item can be helpful in identifying very specific processor features or limitations. Win:1.2, Lin:3.1
type of <processor>	*Plain*	<integer>	Numeric type of the CPU. Values include: • 0 - standard • 1 - overdrive • 2 - dual CPU capable • 3 - reserved • Note: this Inspector returns a <string> type as of BES version 6.0 on Unix machines and version 5.1 on the Macintosh. Win:1.2, Sol:3.1, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
vendor name of <processor>	*Plain*	<string>	The manufacturer of the CPU. Names include: • GenuineIntel • AuthenticAMD • CyrixInstead • CentaurHauls • AmbiguousCPU. Win:1.2, Lin:3.1

Examples

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

- `family name of main processor = "Pentium III"`
  - Returns TRUE for a computer with a Pentium III cpu.

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

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

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

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

© 2007 by BigFix, Inc.

Ram

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

Creation Methods

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

Properties

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

Examples

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

© 2007 by BigFix, Inc.

Active Device

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

Creation Methods

Key Phrase	*Form*	Description
active device	*PlainGlobal*	Creates a list of all active devices found using the Configuration Manager SetupDiGetClassDevs NT API. The locations of active devices are relative to the Windows Directory. Win:1.2

Properties

Key Phrase	*Form*	Return Type	Description
class of <active device>	*Plain*	<string>	Returns the name of the class of the active device provided. Win:1.2
description of <active device>	*Plain*	<string>	Returns the description of the active device provided. Win:1.2
driver key of <active device>	*Plain*	<registry key>	The key identified by adding the value of 'driver key value name of active device' to HKLM\System\CurrentControlSet\Control\Class\. Win:1.2
driver key value name of <active device>	*Plain*	<string>	Returns the driver key value name of the active device provided. Win:1.2
friendly name of <active device>	*Plain*	<string>	Returns the friendly name of the active device. Win:1.2
hardware id of <active device>	*Plain*	<string>	Returns the hardware id of the active device provided. Win:1.2
location information of <active device>	*Plain*	<string>	Returns a string containing information about the bus location of the device. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
manufacturer of <active device>	*Plain*	<string>	Returns the manufactures string of the active device. Win:1.2
problem id of <active device>	*Plain*	<integer>	Configuration manager defined number describing device installation or use problems. Returned by call to CM_Get_DevInst_Status. Win:1.2
service key value name of <active device>	*Plain*	<string>	Returns the service key value name. Win:1.2
status of <active device>	*Plain*	<integer>	Configuration manager defined status bits conveying device driver status. Win:1.2

Examples

- `number of active devices`
  - Returns the number of active devices as determined by the Configuration Manager.

- `exists active device file "system32\ntoskrnl.exe"`
  - Returns TRUE if ntoskrnl.exe exists in the System32 folder of the Windows folder.

- `exists active device file "C:\WINNT\System32\ntoskrnl.exe"`
  - Returns FALSE because this Inspector path is relative to the Windows directory.

- `exists active device whose (class of it = "Display")`
  - Returns TRUE if there is an active device named "Display".

- `(description of it) of active devices whose ( class of it ="Display")`
  - Provides a list of the descriptions of the active display devices.

- `exists service key value name whose (it = "PGPdisk" ) of active devices`
  - Returns TRUE if the designated service key value name exists on this system.

© 2007 by BigFix, Inc.

License

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

Creation Methods

Key Phrase	*Form*	Description
bes license	*PlainGlobal*	Synonym for 'client license'. Win:7.0
client license	*PlainGlobal*	Creates the global object containing client licensing information. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
common name of <license>	*Plain*	<string>	Returns the name of the person (such as John Smith) who requested the action site license. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1
email address of <license>	*Plain*	<string>	Returns the email address of the person (such as John_Smith@bigcorp.com) who requested the action site license. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1
evaluation of <license>	*Plain*	<boolean>	Returns TRUE if client is running an evaluation license. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
expiration date of <license>	*Plain*	<time>	Returns date when license will expire. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
expiration state of <license>	*Plain*	<string>	Returns a string, one of "Unrestricted", "Grace" or "Restricted". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
gather url of <license>	*Plain*	<string>	Returns the gather URL for the deployment's main Action site as specified in the deployment masthead. Win:7.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
maximum seat count of <license>	*Plain*	<integer>	Returns maximum seat count allowed by the license. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
organization of <license>	*Plain*	<string>	Returns the organization of the person (such as Bigcorp, Inc.) who requested the action site license. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1
registrar number of <license>	*Plain*	<integer>	A unique number assigned to the issuer of the Action Site certificate. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1
seat count state of <license>	*Plain*	<string>	Returns one of "Unrestricted", "Grace" or "Restricted". Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
seat of <license>	*Plain*	<integer>	The license number assigned to the client. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site number of <license>	*Plain*	<integer>	A unique number assigned to the Action Site certificate. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1
start date of <license>	*Plain*	<time>	The starting date specified for the BigFix license. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1

Local Mssql Database

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

Creation Methods

Key Phrase	*Form*	Description
local mssql database	*PlainGlobal*	Creates the local MSSQL database objects. Win:1.2
local mssql database <string>	*NamedGlobal*	Creates the local Microsoft SQL (MSSQL) database object identified by the name provided. Win:1.2

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
audit level of <local mssql database>	*Plain*	<integer>	Returns the integer audit level of the MSSQL database. Win:1.2
has blank sa password of <local mssql database>	*Plain*	<boolean>	Returns TRUE if the MSSQL database sa account has a blank password. Win:1.2
instance name of <local mssql database>	*Plain*	<string>	Returns the name of the MSSQL database. Win:1.2
login mode of <local mssql database>	*Plain*	<integer>	Returns the login mode of the MSSQL database. Win:1.2
running of <local mssql database>	*Plain*	<boolean>	Returns a boolean indicating if the MSSQL database is running. Win:1.2

Service

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

Creation Methods

Key Phrase	*Form*	Description
main gather service	*PlainGlobal*	Returns FALSE. Included for compatibility with Windows Inspectors. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
relay service	*PlainGlobal*	Returns a service object for the relay component of BES. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
running service	*PlainGlobal*	Creates objects corresponding to all the running services. Win:1.2
running service <string>	*NamedGlobal*	Creates the running service object for the specified name. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
service	*PlainGlobal*	Creates objects for all the services. Win:1.2
service <string>	*NamedGlobal*	Creates the service object matching the specified name, regardless of its running state. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<service> as string	*Cast*	<string>	Returns a string containing the Service name, Display name, and State of the service. Win:1.2
can interact with desktop of <service>	*Plain*	<boolean>	Indicates the system is configured to allow the service to interact with the desktop. Win:1.2
checkpoint of <service>	*Plain*	<integer>	Service specific value indicating its checkpoint state. Win:1.2
display name of <service>	*Plain*	<string>	Returns the display name of the service. Win:1.2
file of <service>	*Plain*	<file>	Returns a file object corresponding to the specified <service>. Win:3.0
image path of <service>	*Plain*	<string>	Returns the full path to the service executable. Win:6.0
login account of <service>	*Plain*	<string>	Returns the login account under which the service is configured to run. Win:1.2
service name of <service>	*Plain*	<string>	Returns the name of the service. Win:1.2
service specific exit code of <service>	*Plain*	<integer>	Service specific exit code. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
start type of <service>	*Plain*	<string>	Returns a string that represents the service startup configuration. It describes when the driver is loaded, which can be one of: • boot: started by OS loader (usually these are needed to launch the OS). • system: started during OS initialization (used by PnP drivers that do device detection after the loader is done). • auto: started by the Service Control Manager (SCM). • demand: started on demand, either by PnP manager when a device is enumerated or by the SCM in response to user demand. • disabled: can't be started (used to temporarily disable driver services). Win:3.0
state of <service>	*Plain*	<string>	Returns one of Continuing, Pausing, Paused, Running, Starting, Stopping, Stopped, Unknown. Win:1.2, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
win32 exit code of <service>	*Plain*	<integer>	Service specific Win32 exit code. Win:1.2

Examples

- `running services`
- Returns a list of all the currently running services.

- `expand environment string of ( image path of service "AudioSrv”)`
- Returns a path name for the Windows Audio service, such as C:\WINDOWS\System32\svchost.exe -k netsvcs.

© 2007 by BigFix, Inc.

Language

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

Creation Methods

Key Phrase	*Form*	Description
system locale	*PlainGlobal*	Determines which bitmap fonts, and OEM, ANSI, and MAC code pages are defaults for the system. This only affects applications that are not fully Unicode. Win:4.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1
system ui language	*PlainGlobal*	Determines the default language of menus and dialogs, messages, INF files, and help files. Win:4.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1
user locale	*PlainGlobal*	Determines which settings are used for formatting dates, times, currency, and numbers as a default for each user. Also determines the sort order for sorting text. Win:4.1
user ui language	*PlainGlobal*	Non-MUI: Same as system UI Language. • MUI: Determines the language of menus and dialogs, messages, and help files. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<language> as string	*Cast*	<string>	Returns the language of the system locale. Win:4.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1
primary language of <language>	*Plain*	<primary language>	Extracts the primary language identifier from a language. Win:4.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1

Examples

- `system locale as string`
- Returns English (United States) if the system locale is US English.

© 2007 by BigFix, Inc.

Primary Language

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

Creation Methods

Key Phrase	*Form*	Description
primary language of <language>	*Plain*	Extracts the primary language identifier from a language. Win:4.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1

Properties

Key Phrase	*Form*	Return Type	Description
<primary language> as string	*Cast*	<string>	Returns the primary language. Win:4.1, Lin:5.1, Sol:5.1, HPUX:5.1, AIX:5.1

Examples

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

© 2007 by BigFix, Inc.

Firewall Objects

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

Firewall

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

Creation Methods

Key Phrase	*Form*	Description
firewall	*PlainGlobal*	Returns the global firewall object for this computer. Win:5.1

Properties

Key Phrase	*Form*	Return Type	Description
current profile type of <firewall>	*Plain*	<firewall profile type>	Returns the current profile type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_PROFILE_CURRENT. Win:5.1
local policy modify state of <firewall>	*Plain*	<firewall local policy modify state>	Determines if adding or setting a rule or group of rules will take effect in the specified firewall profile. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
local policy of <firewall>	*Plain*	<firewall policy>	Returns the local policy of the specified firewall. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
rule group currently enabled <string> of <firewall>	*Named*	<boolean>	Determines whether a specified group of firewall rules is enabled or disabled for the current profile, considering the firewall's state, BlockAllInboundTraffic state and group policy overrides state. The string is used to group rules together. It can be the group name or an indirect string to the group name in the form of "@yourresourcedll.dll,-23255." Rules belonging to this group will be queried. Win:7.0
rule of <firewall>	*Plain*	<firewall rule>	Retrieves the collection of rules for the specified firewall. For more information, see the MSDN Library article on INetFwPolicy2. Win:7.0
service restriction of <firewall>	*Plain*	<firewall service restriction>	Retrieves the access interface to manipulate the Windows Service Hardening store. On Windows operating systems earlier than Vista, this Inspector always returns no-such-object. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0

© 2007 by BigFix, Inc.

Firewall Authorized Application

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

Creation Methods

Key Phrase	*Form*	Description
authorized application of <firewall profile>	*Plain*	Provides access to the properties of any application that has been authorized to have firewall openings. There are several methods attached to this Inspector, including: • Name • ProcessImageFileName • IpVersion • Scope • RemoteAddress • Enabled. Win:5.1

Properties

Key Phrase	*Form*	Return Type	Description
enabled of <firewall authorized application>	*Plain*	<boolean>	Returns the contents of the Enabled property for the specified application. Returns TRUE if the settings for this application are currently enabled. Win:5.1
ip version of <firewall authorized application>	*Plain*	<ip version>	Returns the contents of the IpVersion property for the specified application. Win:5.1
name of <firewall authorized application>	*Plain*	<string>	Returns the contents of the Friendly Name property for the specified application. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
process image file name of <firewall authorized application>	*Plain*	<string>	Returns the contents of the ProcessImageFileName property for the specified application. Win:5.1
remote addresses of <firewall authorized application>	*Plain*	<string>	Returns the contents of the RemoteAddresses property for the specified application. This property accesses a set of remote addresses that an application can use to listen for traffic. Win:5.1
scope of <firewall authorized application>	*Plain*	<firewall scope>	Returns the contents of the Scope property for the specified application. This property controls the network scope that a port can listen to. Win:5.1

Firewall Profile

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

Creation Methods

Key Phrase	*Form*	Description
current profile of <firewall policy>	*Plain*	Returns the profile currently in effect for the specified firewall policy. Win:5.1
domain profile of <firewall policy>	*Plain*	Returns the domain profile of the specified firewall policy. The domain profile settings are used when a computer is connected to a network that contains the organization's domain controllers. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
private profile of <firewall policy>	*Plain*	Retrieves the private profile type from the specified firewall policy. This profile type is used for home and other private network types. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
public profile of <firewall policy>	*Plain*	Retrieves the public profile type from the specified firewall policy. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
standard profile of <firewall policy>	*Plain*	Returns the standard profile of the specified firewall policy. The standard profile settings are used when a computer is connected to a network that does not contain the organization's domain controllers. Win:5.1

Properties

Key Phrase	*Form*	Return Type	Description
authorized application of <firewall profile>	*Plain*	<firewall authorized application>	Access to the AuthorizedApplications collection for this profile. This Inspector can be iterated through all the authorized applications in each firewall profile. Win:5.1
exceptions allowed of <firewall profile>	*Plain*	<boolean>	Returns the property that indicates whether exceptions should be allowed by the firewall. Win:6.0
excluded interface of <firewall profile>	*Plain*	<string>	Gets the value of the ExcludedInterfaces property from the specified firewall profile. This property contains the list of interfaces excluded from a the profile's firewall rules. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
firewall enabled of <firewall profile>	*Plain*	<boolean>	Gets the value of the FirewallEnabled setting. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
globally open port of <firewall profile>	*Plain*	<firewall open port>	Provides access to the GloballyOpenPorts collection for this profile. This property can be iterated. Win:5.1
icmp settings of <firewall profile>	*Plain*	<firewall icmp settings>	Gets the object governing settings for ICMP packets. Win:5.1
inbound connections allowed of <firewall profile>	*Plain*	<boolean>	Determines whether the default action for inbound traffic for the specified firewall profile is NET_FW_ACTION_ALLOW. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
notifications disabled of <firewall profile>	*Plain*	<boolean>	Gets the value of the NotificationsDisabled setting, TRUE or FALSE. Win:5.1
outbound connections allowed of <firewall profile>	*Plain*	<boolean>	Determines whether the default action for outbound traffic for the specified firewall profile is NET_FW_ACTION_ALLOW. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
remote admin settings of <firewall profile>	*Plain*	<firewall remote admin settings>	Gets the object containing the remote administration settings. Win:5.1
rule group enabled <string> of <firewall profile>	*Named*	<boolean>	Determines whether a specified group of firewall rules are enabled or disabled. For more information, see the MSDN Library entry for INetFwPolicy2. Win:7.0
service of <firewall profile>	*Plain*	<firewall service>	Gets the collection containing the services for this profile. This Inspector can be iterated over all services. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
type of <firewall profile>	*Plain*	<firewall profile type>	Returns the type of the specified firewall profile: domain, standard or current. Win:5.1
unicast responses to multicast broadcast disabled of <firewall profile>	*Plain*	<boolean>	Gets the value of the UnicastResponsesToMulticastBroadcastDisabled setting. Win:5.1

Examples

- `firewall enabled of current profile of local policy of firewall`
  - Verify that the firewall is enabled.

Firewall Profile Type

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

Creation Methods

Key Phrase	*Form*	Description
current firewall profile type	*PlainGlobal*	Retrieves the type of firewall profile that is currently in effect. Win:5.1
current profile type of <firewall>	*Plain*	Creates a domain firewall profile type for comparison. Win:5.1
domain firewall profile type	*PlainGlobal*	Returns the current profile type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_PROFILE_DOMAIN. Win:5.1
firewall profile type <integer>	*NumberedGlobal*	Returns the firewall profile type corresponding to the given integer: • 0: Domain • 1: Standard • 2: Current. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
private firewall profile type	*PlainGlobal*	Retrieves the enumerated variable corresponding to private profile type. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
public firewall profile type	*PlainGlobal*	Retrieves the enumerated variable corresponding to public profile type. This profile type is used for public internet access points. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
standard firewall profile type	*PlainGlobal*	Returns the Standard firewall profile type. Win:5.1
type of <firewall profile>	*Plain*	Returns the type of the specified firewall profile, corresponding to the Microsoft Windows Firewall enumerated types: • NET_FW_PROFILE_DOMAIN • NET_FW_PROFILE_STANDARD • NET_FW_PROFILE_CURRENT. Win:5.1

Operators

Key phrase	Return Type	Description
<firewall profile type> = <firewall profile type>	*<boolean>*	Compares two firewall policies for equality only. Win:5.1

Examples

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

© 2007 by BigFix, Inc.

Firewall Policy

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

Creation Methods

Key Phrase	*Form*	Description
local policy of <firewall>	*Plain*	Returns the local policy of the specified firewall. Win:5.1

Properties

Key Phrase	*Form*	Return Type	Description
current profile of <firewall policy>	*Plain*	<firewall profile>	Returns the profile currently in effect for the specified firewall policy. Win:5.1
domain profile of <firewall policy>	*Plain*	<firewall profile>	Returns the domain profile of the specified firewall policy. The domain profile settings are used when a computer is connected to a network that contains the organization's domain controllers. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:5.1
private profile of <firewall policy>	*Plain*	<firewall profile>	Retrieves the private profile type from the specified firewall policy. This profile type is used for home and other private network types. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0
public profile of <firewall policy>	*Plain*	<firewall profile>	Retrieves the public profile type from the specified firewall policy. For more information, see the MSDN Library entry for NET_FW_PROFILE_TYPE2. Win:7.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
standard profile of <firewall policy>	*Plain*	<firewall profile>	Returns the standard profile of the specified firewall policy. The standard profile settings are used when a computer is connected to a network that does not contain the organization's domain controllers. This Inspector corresponds to the Microsoft Windows Firewall enumerated type NET_FW_PROFILE_STANDARD. Win:5.1

Firewall Scope

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

Creation Methods

Key Phrase	*Form*	Description
all firewall scope	*PlainGlobal*	Returns the scope of computers that allow ALL traffic through the firewall, corresponding to the Microsoft enumerated type NET_FW_SCOPE_ALL. Win:5.1
custom firewall scope	*PlainGlobal*	Returns the custom firewall scope, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SCOPE_CUSTOM. Win:5.1
firewall scope <integer>	*NumberedGlobal*	Returns the scope of addresses from which a port can listen, corresponding to the Microsoft enumerated types: • NET_FW_SCOPE_ALL • NET_FW_SCOPE_LOCAL_SUBNET • NET_FW_SCOPE_CUSTOM. Win:5.1
local subnet firewall scope	*PlainGlobal*	Returns the local subnet firewall scope, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SCOPE_LOCAL_SUBNET. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
scope of <firewall authorized application>	*Plain*	Retrieves the contents of the Scope property of the authorized application. Win:5.1
scope of <firewall open port>	*Plain*	Retrieves the contents of the Scope property of the open port. Win:5.1
scope of <firewall service>	*Plain*	Retrieves the contents of the Scope property of the firewall service. Win:5.1

Operators

Key phrase	Return Type	Description
<firewall scope> = <firewall scope>	*<boolean>*	Compares two firewall scopes for equality only. Win:5.1

Firewall Open Port

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

Creation Methods

Key Phrase	*Form*	Description
globally open port of <firewall profile>	*Plain*	Provides access to the GloballyOpenPorts collection for this profile. Win:5.1
globally open port of <firewall service>	*Plain*	Returns the collection of globally open ports associated with the firewall service. Win:5.1

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
built in of <firewall open port>	*Plain*	<boolean>	Returns the contents of the BuiltIn property of the firewall open port. Win:5.1
enabled of <firewall open port>	*Plain*	<boolean>	Returns the contents of the Enabled property of the firewall open port. Win:5.1
ip version of <firewall open port>	*Plain*	<ip version>	Returns the IpVersion property of the firewall open port. Win:5.1
name of <firewall open port>	*Plain*	<string>	Returns the Name property of the firewall open port. Win:5.1
port of <firewall open port>	*Plain*	<integer>	Returns the Port property of the firewall open port. Win:5.1
protocol of <firewall open port>	*Plain*	<internet protocol>	Returns the Protocol property of the firewall open port. Win:5.1
remote addresses of <firewall open port>	*Plain*	<string>	Returns the RemoteAddresses property of the firewall open port. Win:5.1
scope of <firewall open port>	*Plain*	<firewall scope>	Returns the Scope property of the firewall open port. Win:5.1

© 2007 by BigFix, Inc.

Firewall Service

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

Creation Methods

Key Phrase	*Form*	Description
service of <firewall profile>	*Plain*	Gets the collection containing the services for this profile. Win:5.1

Properties

Key Phrase	*Form*	Return Type	Description
customized of <firewall service>	*Plain*	<boolean>	Returns a flag that indicates whether at least one of the ports associated with the service has been customized. Either TRUE or FALSE. Win:5.1
enabled of <firewall service>	*Plain*	<boolean>	Returns the enabled flag for the specified firewall service. Win:5.1
globally open port of <firewall service>	*Plain*	<firewall open port>	Returns the collection of globally open ports associated with the firewall service. Win:5.1
ip version of <firewall service>	*Plain*	<ip version>	Returns the the IP version for the specified firewall service. Win:5.1
name of <firewall service>	*Plain*	<string>	Returns the friendly name of the firewall service. Win:5.1
remote addresses of <firewall service>	*Plain*	<string>	Returns the contents of the RemoteAddresses property for the specified firewall service. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
scope of <firewall service>	*Plain*	<firewall scope>	Retrieves the contents of the Scope property of the firewall service. Win:5.1
type of <firewall service>	*Plain*	<firewall service type>	Returns the type of the specified firewall service (file and print, upnp, remote desktop or none). Win:5.1

Examples

- `exists globally open port whose (port of it = 52311 and protocol of it = udp and enabled of it) of current profile of local policy of firewall`
  - Returns TRUE if the BES Client can receive pings.

Firewall Service Type

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

Creation Methods

Key Phrase	*Form*	Description
file_and_print firewall service type	*PlainGlobal*	Returns the global service type for file and print sharing, corresponding to the Microsoft enumerated type: NET_FW_SERVICE_FILE_AND_PRINT. Win:6.0
firewall service type <integer>	*NumberedGlobal*	Returns the firewall service type specified by <integer>, corresponding to the Microsoft Windows Firewall enumerated types: • NET_FW_SERVICE_FILE_AND_PRINT • NET_FW_SERVICE_UPNP • NET_FW_SERVICE_REMOTE_DESKTOP • NET_FW_SERVICE_NONE. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
none firewall service type	*PlainGlobal*	Returns the no firewall service type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SERVICE_NONE. Win:6.0
remote desktop firewall service type	*PlainGlobal*	Returns the remote desktop firewall service type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SERVICE_REMOTE_DESKTOP. Win:5.1
type of <firewall service>	*Plain*	Returns the type of the specified firewall service, corresponding to the Microsoft Windows Firewall enumerated types: • NET_FW_SERVICE_FILE_AND_PRINT • NET_FW_SERVICE_UPNP • NET_FW_SERVICE_REMOTE_DESKTOP • NET_FW_SERVICE_NONE. Win:5.1
upnp firewall service type	*PlainGlobal*	Returns the UPnP (Universal Plug and Play) firewall service type, corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_SERVICE_UPNP. • Note: UPnP is not the same as PnP. UPnP is used for network connectivity via TCP/IP to various devices (scanners, printers, etc.). Win:5.1

Operators

Key phrase	Return Type	Description
<firewall service type> = <firewall service type>	*<boolean>*	Compares two firewall service types for equality only. Win:5.1

Examples

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

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

© 2007 by BigFix, Inc.

Firewall Icmp Settings

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

Creation Methods

Key Phrase	*Form*	Description
icmp settings of <firewall profile>	*Plain*	The Microsoft Windows Firewall INetFwIcmpSettings interface provides access to the settings controlling Internet Control Message Protocol (ICMP) packets. Win:5.1

Properties

Key Phrase	*Form*	Return Type	Description
allow inbound echo request of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowInboundEchoRequest property. Type common to IPv4 and IPv6. Win:5.1
allow inbound mask request of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowInboundMaskRequest property. Type common to IPv4 only. Win:5.1
allow inbound router request of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowInboundRouterRequest property. Type common to IPv4 only. Win:5.1
allow inbound timestamp request of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowInboundTimestampRequest property. Type common to IPv4 only. Win:5.1
allow outbound destination unreachable of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundDestinationUnreachable property. Type common to IPv4 and IPv6. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
allow outbound packet too big of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundPacketTooBig property. Type common to IPv6 only. Win:5.1
allow outbound parameter problem of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundParameterProblem property. Type common to IPv4 and IPv6. Win:5.1
allow outbound source quench of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundSourceQuench property. Type common to IPv4 only. Win:5.1
allow outbound time exceeded of <firewall icmp settings>	*Plain*	<boolean>	Returns the value of the AllowOutboundTimeExceeded property. Type common to IPv4 and IPv6. Win:5.1
allow redirect of <firewall icmp settings>	*Plain*	<boolean>	Accesses the AllowRedirect property. Type common to IPv4 and IPv6. Win:5.1

Firewall Remote Admin Settings

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

Creation Methods

Key Phrase	*Form*	Description
remote admin settings of <firewall profile>	*Plain*	Gets the object containing the remote administration settings. These settings include the following properties: • Enabled • IpVersion • RemoteAddresses • Scope. Win:5.1

© 2007 by BigFix, Inc.

Firewall Action

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

Creation Methods

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

Operators

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

© 2007 by BigFix, Inc.

Firewall Local Policy Modify State

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

Creation Methods

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

© 2007 by BigFix, Inc.

Operators

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

Firewall Rule

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

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

© 2007 by BigFix, Inc.

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

© 2007 by BigFix, Inc.

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

© 2007 by BigFix, Inc.

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

Firewall Service Restriction

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
service restricted <(string, string)> of <firewall service restriction>	*Indexed*	<boolean>	Returns the service restriction state of a given service. There are two strings passed to define the restriction: the service name and the application name. For more information, see the MSDN Library section on INetFwServiceRestriction. Win:7.0

Internet Protocol

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

Creation Methods

Key Phrase	*Form*	Description
internet protocol <integer>	*NumberedGlobal*	Returns the firewall internet protocol specified, either tcp or udp. These correspond to the Microsoft Windows Firewall enumerated types: • NET_FW_IP_PROTOCOL_TCP • NET_FW_IP_PROTOCOL_UDP. Win:5.1
protocol of <firewall open port>	*Plain*	Returns the Protocol property of the firewall open port. Win:5.1
protocol of <firewall rule>	*Plain*	Retrieves the Protocol property for the specified firewall rule. The firewall rule Inspectors are wrappers around the Windows Vista Firewall API. For more information, see the MSDN library for INetFwRule. Win:7.0
tcp	*PlainGlobal*	Returns an internet protocol corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_IP_PROTOCOL_TCP. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
udp	*PlainGlobal*	Returns an internet protocol corresponding to the Microsoft Windows Firewall enumerated type: NET_FW_IP_PROTOCOL_UDP. Win:5.1

Operators

Key phrase	Return Type	Description
<internet protocol> = <internet protocol>	*<boolean>*	Compares two firewall internet protocols for equality only. Win:5.1

Examples

- `exists globally open port whose (port of it = 52311 and protocol of it = udp and enabled of it) of current profile of local policy of firewall`
  - Returns TRUE if the BES Client can receive pings.

Ip Version

Returns the firewall ip version information corresponding to the Microsoft enumerated types, either ipv4 or ipv6. The Windows Firewall is supported on Windows XP SP2. For more information, search for 'Windows Firewall API' at the msdn site (http://msdn.microsoft.com/library/).

Creation Methods

Key Phrase	*Form*	Description
any ip version	*PlainGlobal*	Returns a type corresponding to the Microsoft enumerated value NET_FW_IP_VERSION_ANY. Win:5.1
ip version <integer>	*NumberedGlobal*	Returns the the IP version for the <integer> port corresponding to the Microsoft enumerated types: • NET_FW_IP_VERSION_V4 • NET_FW_IP_VERSION_V6 • NET_FW_IP_VERSION_ANY • NET_FW_IP_VERSION_MAX. Win:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
ip version of <firewall authorized application>	*Plain*	Returns the the IP version for the specified firewall authorized application. Win:5.1
ip version of <firewall open port>	*Plain*	Returns the the IP version for the specified firewall open port. Win:5.1
ip version of <firewall service>	*Plain*	Returns the the IP version for the specified firewall service. Win:5.1
ipv4	*PlainGlobal*	Provides a comparison value for a firewall ip version inspector. Win:5.1
ipv6	*PlainGlobal*	Provides a comparison value for a firewall ip version inspector. Win:5.1

Operators

Key phrase	Return Type	Description
<ip version> = <ip version>	*<boolean>*	Compares two firewall ip versions for equality only. Win:5.1

© 2007 by BigFix, Inc.

WMI Objects

The Inspectors for Windows Management Instrumentation (WMI).

Wmi

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

Creation Methods

Key Phrase	*Form*	Description
full wmi <string>	*NamedGlobal*	Returns a wmi object which can retrieve all values, including system values. Win:2.0
rsop computer wmi	*PlainGlobal*	Provides access to the Resultant Set of Policy (RSoP) WMI classes via the RsopLoggingModeProvider. This is used to examine the state of the GPO (Group Policy object) security policies. Win:7.0
rsop user wmi <security identifier>	*IndexedGlobal*	Each user has its own RSoP (Resultant Set of Policy) namespace based on the user's Security Identifier (SID). This Inspector returns the namespace specified by the <security identifier>. Win:7.0
wmi	*PlainGlobal*	Returns the wmi object corresponding to the "root\cimv2" namespace. Win:2.0
wmi <string>	*NamedGlobal*	Returns the wmi object corresponding to the namespace string provided. Win:2.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
select <string> of <wmi>	*Named*	<wmi select>	Returns the wmi select whose name matches the string provided. Win:2.0
select object <string> of <wmi>	*Named*	<wmi object>	Returns the desired property (specified by <string>) from the given wmi object. Win:6.0

Wmi Select

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

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

Creation Methods

Key Phrase	*Form*	Description
property <string> of <wmi object>	*Named*	Returns the Inspector properties of the form <string> of <wmi object>. Win:6.0
property of <wmi object>	*Plain*	Returns the Inspector properties of the specified wmi object. Win:6.0
select <string> of <wmi>	*Named*	Returns the wmi select whose name matches the string provided. Win:2.0

© 2007 by BigFix, Inc.

Note

Here are a few other examples of using the wmi Inspectors. Each of the examples below hands back dozens of settings:
Q: selects "* from Win32_ComputerSystem" of wmi
Q: selects "* from win32_keyboard" of wmi
Q: selects "* from win32_CDROMDrive" of wmi
Q: selects "* from win32_DiskDrive" of wmi
Q: selects "* from win32_BIOS" of wmi
Q: selects "* from win32_CacheMemory" of wmi
Q: selects "* from win32_DMIChannel" of wmi
Q: selects "* from win32_FloppyController" of wmi
Q: selects "* from win32_IDEController" of wmi
Q: selects "* from win32_IRQResource" of wmi
Q: selects "* from win32_MemoryDevice" of wmi
Q: selects "* from win32_MotherboardDevice" of wmi
Q: selects "* from win32_ParallelPort" of wmi
Q: selects "* from Win32_PNPDevice" of wmi
Q: selects "* from win32_Processor" of wmi
Q: selects "* from win32_SerialPort" of wmi
Q: selects "* from win32_SoundDevice" of wmi
Q: selects "* from win32_NetworkAdapter" of wmi
Q: selects "* from win32_NetworkAdapterSetting" of wmi
Q: selects "* from win32_Battery" of wmi
Q: selects "* from win32_PrinterPrinterDriver" of wmi
Q: selects "* from win32_PrinterSetting" of wmi

Properties

Key Phrase	***Form***	**Return Type**	**Description**
<wmi select> as string	*Cast*	<string>	Returns a string formatted as <name>=<value> for the wmi select. Win:2.0
boolean value <integer> of <wmi select>	*Numbered*	<boolean>	Returns the WMI value cast to boolean. Win:4.1
boolean value of <wmi select>	*Plain*	<boolean>	Returns the boolean value of a <wmi select> object (exists only for boolean objects). Win:2.0
integer value <integer> of <wmi select>	*Numbered*	<integer>	Returns the WMI value cast to an integer. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
integer value of <wmi select>	*Plain*	<integer>	Returns the integer value of a <wmi select> object whose value is of type integer. Win:2.0
name of <wmi select>	*Plain*	<string>	Returns the name of the wmi select. Win:2.0
string value <integer> of <wmi select>	*Numbered*	<string>	Returns the numbered string of a WMI array value. Win:4.1
string value of <wmi select>	*Plain*	<string>	Returns the string value of the wmi select. Win:2.0
time value <integer> of <wmi select>	*Numbered*	<time>	Returns the numbered time of a WMI array value. Win:4.1
time value of <wmi select>	*Plain*	<time>	Returns the time value of the wmi select whose value is of type time. Win:2.0
type of <wmi select>	*Plain*	<integer>	Returns the type of data stored in the wmi select value. Possible values are: • 8 (VT_BSTR) • 11 (VT_BOOL) • 22 (VT_INT) • (See MSDN documentation for the meaning of additional values.). Win:2.0

© 2007 by BigFix, Inc.

Wmi Object

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

Creation Methods

Key Phrase	*Form*	Description
select object <string> of <wmi>	*Named*	Returns the desired property (specified by <string>) from the given wmi object. Win:6.0

Properties

Key Phrase	*Form*	Return Type	Description
<wmi object> as string	*Cast*	<string>	Casts the given wmi object as a string type. Win:6.0
property <string> of <wmi object>	*Named*	<wmi select>	Returns the Inspector properties of the form <string> of <wmi object>. Win:6.0
property of <wmi object>	*Plain*	<wmi select>	Returns the Inspector properties of the specified wmi object. Win:6.0

Examples

- `Property "SerialNumber" of select object "* from win32_operatingsystem" of wmi`
  - Returns serial number of the selected wmi object, in a form like SerialNumber=76487-OEM-0012903-00925.

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

© 2007 by BigFix, Inc.

Site Objects

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

Site

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

Creation Methods

Key Phrase	*Form*	Description
current site	*PlainGlobal*	Creates the site object corresponding to the site that provided the current Fixlet. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site	*PlainGlobal*	Iterates through all the sites. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site <string>	*NamedGlobal*	Creates the site object that corresponds to the name provided. The name is interpreted as a site locator and is therefore a URL. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Properties

Key Phrase	*Form*	Return Type	Description
client folder of <site>	*Plain*	<folder>	The folder containing the site content on the client machine. Site content is gathered into this location. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
fixlet of <site>	*Plain*	<fixlet>	Iterates through the Fixlet messages of the specified site. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
gather schedule authority of <site>	*Plain*	<string>	Returns a string corresponding to the authority of the site schedule, e.g.: Publisher, Custom, Manual or Disabled. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
gather schedule time interval of <site>	*Plain*	<time interval>	Returns the time interval between automatic gathering of site content. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
group <integer> of <site>	*Numbered*	<site group>	Returns an object corresponding to the numbered group of the specified site. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
last gather time of <site>	*Plain*	<time>	Returns the time of last successful gathering from the site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
masthead of <site>	*Plain*	<file>	Each site has a masthead, and the masthead is saved into the site data folder upon successful creation. This property returns a file object that corresponds to the copy in the site data folder. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
name of <site>	*Plain*	<string>	The name of the site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
relevant fixlet of <site>	*Plain*	<fixlet>	Iterates through the Relevant Fixlet messages for the specified site. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
setting <string> of <site>	*Named*	<setting>	Returns the setting whose name matches the string provided from the Fixlet site settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
setting of <site>	*Plain*	<setting>	Returns one or more settings from the site settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
site tag of <site>	*Plain*	<string>	Returns the last component of the specified site's url, eg. ‘actionsite’, ‘enteprisesecurity’, etc. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
site version list of <site>	*Plain*	<site version list>	Returns the last gathered site version list (manyversion) of the specified site. Win:7.0
subscribe time of <site>	*Plain*	<time>	Returns the time that the current machine began subscribing to the site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
type of <site>	*Plain*	<string>	Returns one of the following 4 literal strings: • Master Action Site • Operator Site • Custom Site • Fixlet Site. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
url of <site>	*Plain*	<string>	Returns the Locator found in the masthead. A site locator is used to synchronize with the site. It normally contains the URL of a remote file system folder, or the URL of a cgi-bin program that provides a remote directory listing of the site. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
version of <site>	*Plain*	<integer>	Returns the version number of the site content. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Note

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

Examples

- `exists site "actionsite"`
  - TRUE when the action site exists on the target machine.
- `exists file "siteicon.bmp" of client folder of current site`
  - TRUE if the specified file exists in the client folder.
- `last gather time of current site > now - 30 * day`
  - Return TRUE if it has been over 30 days since last gathering, or synchronizing, with the site.
- `last gather time of current site < time "4 Aug 1997 01:00 pdt"`
  - Returns TRUE if the site was last synchronized before the specified date.
- `modification time of masthead of current site < time "4 Aug 1997 01:00 pdt"`
  - Returns TRUE if the masthead of the current site is older than the specified date.

© 2007 by BigFix, Inc.

Site Group

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

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
id of <site group>	*Plain*	<integer>	Returns the numeric ID of the specified site group. This is the number assigned to an automatic group when it is first defined. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
member of <site group>	*Plain*	<boolean>	Returns TRUE if the current computer is a member of the specified group. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

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
site version list <string>	*NamedGlobal*	Returns a textual representation of a site version list ("manyversion"). Win:7.0
site version list of <site>	*Plain*	Returns the last gathered site version list (manyversion) of the specified site. Win:7.0

© 2007 by BigFix, Inc.

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
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

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

Fixlet

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

Creation Methods

Key Phrase	*Form*	Description
fixlet of <site>	*Plain*	This Inspector iterates over all the Fixlet messages in the given site. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
relevant fixlet of <site>	*Plain*	Iterates over all the relevant Fixlet messages in the specified site. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
header <string> of <fixlet>	*Named*	<fixlet_header>	Returns the named header (case insensitive) of the specified Fixlet message. Fixlet headers are name:value pairs. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0

© 2007 by BigFix, Inc.

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

Fixlet_header

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

Creation Methods

Key Phrase	*Form*	Description
header <string> of <fixlet>	*Named*	Returns the named header (case insensitive) of the specified Fixlet message. Fixlet headers are name:value pairs. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
header of <fixlet>	*Plain*	Iterates over all the headers of the Fixlet message. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0

Properties

Key Phrase	*Form*	Return Type	Description
name of <fixlet_header>	*Plain*	<string>	Headers are name:value pairs, separated by a colon. This Inspector returns the name on the left hand side of the pair. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0
value of <fixlet_header>	*Plain*	<string>	Headers are name:value pairs, separated by a colon. This Inspector returns the value on the right hand side of the pair. Win:5.0, Lin:5.0, Sol:5.0, HPUX:5.0, AIX:5.0, Mac:6.0

© 2007 by BigFix, Inc.

Examples

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

© 2007 by BigFix, Inc.

Client Objects

These Inspectors retrieve information about the organization of the BES Client computers.

Client

The client object allows access to properties of the client application hosting the relevance evaluation, typically a BigFix program. In addition, the client maintains a collection of settings with both name and value properties that are inspectable using the client object. Client objects are derived from file objects, so they share all the file properties.

Creation Methods

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

Properties

Key Phrase	*Form*	Return Type	Description
administrator <string> of <client>	*Named*	<setting>	If the administrator named in the <string> is enabled on the given <client> computer, this property returns a setting with the given name and the value 'allow.' For instance, if the name of the administrator is joe_admin, then the client would return a setting object with the name 'joe_admin' and a value of 'allow'. Casting this as a string would return 'joe_admin=allow'. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
administrator of <client>	*Plain*	<setting>	Creates a setting object corresponding to the administrators of the given <client>. Win:3.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:6.0
setting <string> of <client>	*Named*	<setting>	Returns a client setting whose name matches the string provided from the client settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
setting of <client>	*Plain*	<setting>	Returns one or more settings from the client settings. Win:2.0, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

Setting

A setting is a simple object with name and value properties. It is a property of a client, or a property of a site. Settings of a site are assigned by site authors. Settings of the client are assigned by the BES console operator.

Creation Methods

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

© 2007 by BigFix, Inc.

Properties

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

Examples

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

Selected Server

The BES Server or BES Relay to which the BigFix agent or client reports.

Creation Methods

Key Phrase	*Form*	Description
selected server	*PlainGlobal*	The BES Server or BES Relay to which the agent reports. Returned as the "selected server" type. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
competition size of <selected server>	*Plain*	<integer>	The number of servers in the competition from which this server was selected. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
competition weight of <selected server>	*Plain*	<integer>	The total of the weights of the servers in the competition from which this server was selected. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
distance of <selected server>	*Plain*	<integer range>	The distance, in IP gateway hops, to the server. Among servers with the same priority, closer servers are preferred. Returns an integer range, since the exact distance may not be known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
gateway address <integer> of <selected server>	*Numbered*	<ipv4 address>	The ip address of a gateway between the agent and the selected server at the given distance from the agent, if known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
gateway address of <selected server>	*Plain*	<ipv4 address>	All known ip addresses of gateways between the agent and the selected server. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
ip address of <selected server>	*Plain*	<ipv4 address>	The ip address to which reports are sent. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
name of <selected server>	*Plain*	<string>	The DNS name of the server, if known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
port number of <selected server>	*Plain*	<integer>	The port number to which reports are sent. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
priority of <selected server>	*Plain*	<integer>	The priority assigned to the server by the BES console. Servers with low priorities are preferred to servers with high priority. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1
weight of <selected server>	*Plain*	<integer>	The weight assigned to the server by the BES console. Servers with the same priority and approximate distance compete to be chosen; servers with higher weights are more likely to be chosen. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Operating System Product Type

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

Creation Methods

Key Phrase	*Form*	Description
nt domain controller product type	*PlainGlobal*	Returns an object corresponding to OS product type of nt domain controller. Win:6.0
nt server product type	*PlainGlobal*	Returns an object corresponding to OS product type of nt server. Win:6.0
nt workstation product type	*PlainGlobal*	Returns an object corresponding to OS product type of nt workstation. Win:6.0
operating system product type <integer>	*NumberedGlobal*	Returns an object corresponding to the numbered OS product type. Win:6.0
product type of <operating system>	*Plain*	Returns the product type of the operating system, which includes Workstations, Domain Controllers and Servers. Win:6.0

Operators

Key phrase	Return Type	Description
<operating system product type> = <operating system product type>	*<boolean>*	Compare two operating system product types for equality. Win:6.0

© 2007 by BigFix, Inc.

Operating System Suite Mask

These Inspectors provide detailed information about the operating system version.

Creation Methods

Key Phrase	*Form*	Description
suite mask of <operating system>	*Plain*	Returns the bit-mapped suite mask for the operating system, which contains further fine-grain information about the version. Win:6.0

Examples

- `suite mask of operating system`
  - Returns the suite mask for the operating system.

Local Group

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

Creation Methods

Key Phrase	*Form*	Description
local group	*PlainGlobal*	Returns local groups defined on the local computer using the windows NetLocalGroupEnum API. Several local groups are defined simply by a default operating system install, and have names such as Administrators, Backup Operators, Guests, Network Configuration Operators, Power users, Users, etc. Some software applications also define local groups in order to help manage protections. Win:6.0
local group <string>	*NamedGlobal*	Returns a local group corresponding to the given name, such as Adminstrator, Guests, etc. Win:6.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
comment of <local group>	*Plain*	<string>	Returns a string containing a comment associated with the specfied local group (Administrator, Guest, Users). Win:6.0
member of <local group>	*Plain*	<local group member>	Returns a list of the members of the specified local group. Win:6.0
name of <local group>	*Plain*	<string>	Returns a list if the names of the local group. Win:6.0

Examples

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

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

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

Local Group Member

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

Creation Methods

Key Phrase	*Form*	Description
member of <local group>	*Plain*	Returns an object corresponding to a member of the specified local group. Win:6.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
<local group member> as string	*Cast*	<string>	Casts a local group member as a string. Win:6.0
sid of <local group member>	*Plain*	<security identifier>	Returns the security ID for the specified local group member. Win:6.0

Examples

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

Event Log

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

Creation Methods

Key Phrase	*Form*	Description
application event log	*PlainGlobal*	Returns an object corresponding to an application event log, one of the event logs created by most Windows systems. Win:6.0
event log <string>	*NamedGlobal*	Returns the event log object with the specified name. Win:6.0
security event log	*PlainGlobal*	Returns an event log object for the security event log. Win:6.0
system event log	*PlainGlobal*	Returns a system event log, which records OS or component events, such as the failure of a bootup service. Win:6.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
oldest record number of <event log>	*Plain*	<integer>	Returns an integer corresponding to the oldest record number on the Client computer's event log. Win:6.0
record <integer> of <event log>	*Numbered*	<event log record>	Returns the nth record corresponding to the specified event log. Win:6.0
record count of <event log>	*Plain*	<integer>	Returns the record count for the specified event log. Win:6.0
record of <event log>	*Plain*	<event log record>	Returns the record corresponding to the specified event log, for instance the application or system event log. Win:6.0

Examples

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

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

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

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

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

© 2007 by BigFix, Inc.

Event Log Record

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

Creation Methods

Key Phrase	*Form*	Description
record <integer> of <event log>	*Numbered*	Returns the nth record corresponding to the specified event log. Win:6.0
record of <event log>	*Plain*	Retrieves the record from the event log. Win:6.0

Properties

Key Phrase	*Form*	Return Type	Description
category of <event log record>	*Plain*	<integer>	Returns the category of the specified event log, which is sometimes used to further describe the related action. Win:6.0
computer of <event log record>	*Plain*	<string>	Returns the name of the computer that has entered a record in the specified log. Win:6.0
description of <event log record>	*Plain*	<string>	Returns a human-readable description of the specified event log record. Win:6.0
event id of <event log record>	*Plain*	<integer>	Returns an integer corresponding to the ID of the specified record in the Client computer's event log. Win:6.0
event type of <event log record>	*Plain*	<event log event type>	Returns the type of the specified event log record, such as error, warning, information, etc. Win:6.0
length of <event log record>	*Plain*	<integer>	Returns the length of the specified record. This is not the same as the length of the description. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
record number of <event log record>	*Plain*	<integer>	Returns the integer value of the record number corresponding to the specified event log record. Win:6.0
source of <event log record>	*Plain*	<string>	Returns the source name (from the application, service, or component that logged the event) of the specified event log record. Win:6.0
time generated of <event log record>	*Plain*	<time>	Returns the time that the specified event log record was generated. Win:6.0
time written of <event log record>	*Plain*	<time>	Returns the time that the specified event record was written to the log. Win:6.0
user sid of <event log record>	*Plain*	<security identifier>	Returns the user security ID for the specified record in the event log. Win:6.0

Examples

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

- `exists records of application event log`
  - Returns TRUE if the application event log contains any records.

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

- `computer of record (oldest record number of it) of application event log`
  - Returns the name of the computer that logged the last entry in the application event log.

- `description of record (oldest record number of it) of application event log`
  - Returns a description of the oldest record in the application event log. Typically includes a description of the programs and what happened for each entry in the event log.

© 2007 by BigFix, Inc.

- `event id of record (oldest record number of it) of application event log`
  - Returns the id number of the oldest record in the application event log.

- `length of record (oldest record number of it) of application event log`
  - Returns the length of the specified record in the application event log.

- `record number of record (oldest record number of it) of application event log`
  - Returns the integer record number corresponding to the oldest record in the application event log.

- `source of record (oldest record number of it) of application event log`
  - Returns name of the source of the oldest record in the application event log.

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

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

- `user sid of record (oldest record number of it) of application event log`
  - Returns the user security ID for the oldest record in the application event log, for instance NT AUTHORITY\SYSTEM.

© 2007 by BigFix, Inc.

Event Log Event Type

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

Creation Methods

Key Phrase	*Form*	Description
audit failure event log event type	*PlainGlobal*	Returns an object corresponding to an audit failure -- an event related to the failed execution of an action. Win:6.0
audit success event log event type	*PlainGlobal*	Returns an object corresponding to an audit success in an event log. Win:6.0
error event log event type	*PlainGlobal*	Returns an object corresponding to an error event in the log, such as the failure of a service to start. Win:6.0
event log event type <integer>	*NumberedGlobal*	Returns an event type object corresponding to the specified number. The enumerated types include: • 1: error event • 2: warning event • 4: information event • 8: audit success event • 16: audit failure event. Win:6.0
event type of <event log record>	*Plain*	Returns the event type of the specified record from the event log. Win:6.0
information event log event type	*PlainGlobal*	Returns an object corresponding to an information event -- An informational event which is generally related to a successful action. Win:6.0
warning event log event type	*PlainGlobal*	Returns an object corresponding to a warning in the event log. Warnings can be used to prevent future system problems. Win:6.0

© 2007 by BigFix, Inc.

Operators

Key phrase	Return Type	Description
<event log event type> = <event log event type>	*<boolean>*	Compare two event log event types for equality. Win:6.0

Examples

- `audit failure event log event type= event type of record (oldest record number of it) of application event log`
  - Returns TRUE if the oldest record of the application event log contains an audit failure.

- `error event log event type= event type of record (oldest record number of it) of application event log`
  - Returns TRUE if the oldest record of the application event log contains an error.

Current Relay

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

Creation Methods

Key Phrase	*Form*	Description
current relay	*PlainGlobal*	Returns an object corresponding to the server or relay that the client last registered with. This may be a BES Relay or the BES root server. Win:7.0

Properties

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

© 2007 by BigFix, Inc.

Root Server

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

Creation Methods

Key Phrase	*Form*	Description
root server	*PlainGlobal*	Returns an object representing the root BES Server to which the client last registered. Win:7.0

Properties

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

© 2007 by BigFix, Inc.

Environment Objects

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

Environment

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

Creation Methods

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

Properties

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

Examples

- `exists environment`
  - TRUE if the computer has an environment object.
- `value of variable "path" of environment contains "\extras\"`
  - TRUE if there is an environment variable named "path" and its value contains "\extras\".
- `number of variables of environment`
  - Returns the total number of variables in the environment space.

© 2007 by BigFix, Inc.

Environment Variable

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

Creation Methods

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

Properties

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

Examples

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

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

© 2007 by BigFix, Inc.

Authorization Objects

These inspectors retrieve security and access settings.

Access Control Entry

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

Creation Methods

Key Phrase	*Form*	Description
entry of <access control list>	*Plain*	Iterates the ACEs of an ACL. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
access mode of <access control entry>	*Plain*	<integer>	For a discretionary ACL (DACL), this flag indicates whether the ACL allows (1) or denies (3) the specified access rights. Win:4.1
append permission of <access control entry>	*Plain*	<boolean>	For a file ACE, returns TRUE if the ACE grants or denies append permissions. Win:4.1
change notification permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies change notification permissions. Win:4.1
create file permission of <access control entry>	*Plain*	<boolean>	For a folder ACE, returns TRUE if the ACE grants or denies create file permissions. Win:4.1
create folder permission of <access control entry>	*Plain*	<boolean>	For a folder ACE, returns TRUE if the ACE grants or denies create folder permissions. Win:4.1
create link permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies create key link permissions. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
create subkey permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies creation of subkey permissions. Win:4.1
delete child permission of <access control entry>	*Plain*	<boolean>	For a folder ACE, returns TRUE if the ACE grants or denies child deletion permissions. Win:4.1
delete permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or generic delete permissions. Win:4.1
enumerate subkeys permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or enumerate subkey permissions. Win:4.1
execute permission of <access control entry>	*Plain*	<boolean>	For a file ACE, returns TRUE if the ACE grants or denies execute permissions. Win:4.1
generic all permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies all generic permissions. Win:4.1
generic execute permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies generic execute permissions. Win:4.1
generic read permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies generic read permissions. Win:4.1
generic write permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies generic write permissions. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
inheritance of <access control entry>	*Plain*	<integer>	A set of bit flags that determines whether other containers or objects can inherit the ACE from the primary object to which the ACL is attached. The actual values of the constants are: • NO_INHERITANCE = 0 • SUB_OBJECTS_ONLY_INHERIT = 1 • SUB_CONTAINERS_ONLY_INHERIT = 2 • SUB_CONTAINERS_AND_OBJECTS_INHERIT = 3 • OJECT_INHERIT_ACE = 1 • CONTAINER_INHERIT_ACE = 2 • NO_PROPAGATE_INHERIT_ACE = 4 • INHERIT_ONLY_ACE = 8. Win:4.1
list permission of <access control entry>	*Plain*	<boolean>	For a folder ACE, returns TRUE if the ACE grants or denies list permissions. Win:4.1
maximum allowed permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies maximum allowed permissions. Win:4.1
query value permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies query value permissions. Win:4.1
read attributes permission of <access control entry>	*Plain*	<boolean>	For a file or folder ACE, returns TRUE if the ACE grants or denies read attributes permissions. Win:4.1
read control permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies reading access control permissions. Win:4.1
read extended attributes permission of <access control entry>	*Plain*	<boolean>	For a file or folder ACE, returns TRUE if the ACE grants or denies read extended attributes permissions. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
read permission of <access control entry>	*Plain*	<boolean>	For a file ACE, returns TRUE if the ACE grants or denies read permissions. Win:4.1
set value permission of <access control entry>	*Plain*	<boolean>	For a registry key ACE, returns TRUE if the ACE grants or denies set value permissions. Win:4.1
synchronize permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies synchronize permissions. Win:4.1
traverse permission of <access control entry>	*Plain*	<boolean>	For the specified folder ACE, returns TRUE if it grants or denies traverse • folder permission. Win:4.1
trustee of <access control entry>	*Plain*	<security identifier>	Returns the trustee to whom the specified ACE applies. Win:4.1
trustee type of <access control entry>	*Plain*	<integer>	Returns the type of trustee to whom the specified ACE applies. Win:4.1
write attributes permission of <access control entry>	*Plain*	<boolean>	For a file or folder ACE, returns TRUE if the ACE grants or denies write attribute permissions. Win:4.1
write dac permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies write DAC permissions. Win:4.1
write extended attributes permission of <access control entry>	*Plain*	<boolean>	For a file or folder ACE, returns TRUE if the ACE grants or denies write extended attribute permissions. Win:4.1
write owner permission of <access control entry>	*Plain*	<boolean>	For any ACE, returns TRUE if the ACE grants or denies write owner permissions. Win:4.1
write permission of <access control entry>	*Plain*	<boolean>	For a file ACE, returns TRUE if the ACE grants or denies write permissions. Win:4.1

© 2007 by BigFix, Inc.

Access Control List

An Access Control List, or ACL, is a list of security protections that applies to an object. An object can be a file, process, event, or anything else having a security descriptor. An entry in an access control list (ACL) is an access control entry (ACE). Access rights run up through the object hierarchy, so rights granted at a low level can be vetoed by rights higher up. Some of the following Inspectors determine the effective permissions for a given trustee by traversing the hierarchy. They work by exposing the GetEffectiveRightsFromAcl method, as explained here: http://tinyurl.com/dtmje.

Creation Methods

Key Phrase	*Form*	Description
dacl of <security descriptor>	*Plain*	Returns the discretionary access control list associated with the given security descriptor. Win:4.1

Note

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

Properties

Key Phrase	*Form*	Return Type	Description
effective access mode for <string> of <access control list>	*Named*	<integer>	Returns an integer corresponding to the access mode for the trustee specified by <string> of the given access control list. Win:6.0
effective access system security permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has access system security permissions on the given access control list. Win:6.0
effective append permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has append permissions on the given access control list. Win:6.0
effective change notification permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has change notification permissions on the given access control list. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
effective create file permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has file creation permissions on the given access control list. Win:6.0
effective create folder permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has folder creation permissions on the given access control list. Win:6.0
effective create link permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has link creation permissions on the given access control list. Win:6.0
effective create subkey permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has subkey creation permissions on the given access control list. Win:6.0
effective delete child permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has child deletion permissions on the given access control list. Win:6.0
effective delete permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has delete permissions on the given access control list. Win:6.0
effective enumerate subkeys permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has subkey enumeration permissions on the given access control list. Win:6.0
effective execute permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has execution permissions on the given access control list. Win:6.0
effective generic all permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has all generic permissions on the given access control list. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
effective generic execute permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has generic execution permissions on the given access control list. Win:6.0
effective generic read permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has generic read permissions on the given access control list. Win:6.0
effective generic write permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has generic write permissions on the given access control list. Win:6.0
effective list permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has list permissions on the given access control list. Win:6.0
effective maximum allowed permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has maximum allowed permissions on the given access control list. Win:6.0
effective query value permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has query value permissions on the given access control list. Win:6.0
effective read attributes permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has read attribute permissions on the given access control list. Win:6.0
effective read control permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has read control permissions on the given access control list. Win:6.0
effective read extended attributes permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has extended read attribute permissions on the given access control list. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
effective read permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has read permissions on the given access control list. Win:6.0
effective set value permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has value setting permissions on the given access control list. Win:6.0
effective synchronize permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has synchronization permissions on the given access control list. Win:6.0
effective traverse permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has traverse permissions on the given access control list. Win:6.0
effective write attributes permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has attribute writing permissions on the given access control list. Win:6.0
effective write dac permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has dac writing permissions on the given access control list. Win:6.0
effective write extended attributes permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has extended attribute writing permissions on the given access control list. Win:6.0
effective write owner permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has write owner permissions on the given access control list. Win:6.0
effective write permission for <string> of <access control list>	*Named*	<boolean>	Returns TRUE if the trustee specified by <string> has write permissions on the given access control list. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
entry of <access control list>	*Plain*	<access control entry>	Iterates the ACEs of a ACL. Win:4.1

Note

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

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

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

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

© 2007 by BigFix, Inc.

Examples

- `number of dacls of (security descriptor of file "mshtml.dll" of system folder) = 1`

  Returns TRUE if there is exactly one discretionary access control list associated with the security descriptor of the specified file.

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

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

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

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

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

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

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

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

Security Descriptor

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

Creation Methods

Key Phrase	*Form*	Description
security descriptor of <file>	*Plain*	Specifies the security descriptor associated with the specified file. Win:4.1
security descriptor of <folder>	*Plain*	Specifies the security descriptor associated with the specified folder. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
security descriptor of <network share>	*Plain*	Specifies the security descriptor associated with the specified network share. Win:4.1
security descriptor of <registry key>	*Plain*	Specifies the security descriptor associated with the specified registry key. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<security descriptor> as string	*Cast*	<string>	Returns the security descriptor in string format. Win:4.1
control of <security descriptor>	*Plain*	<integer>	Returns the integer property obtained by using the Microsoft Windows GetSecurityDescriptorControl API. This integer contains bits that indicate DACL behaviors as well as default behaviors. See the MSDN documentation of SECURITY_DESCRIPTOR_CONTROL for more information. Win:6.0
dacl of <security descriptor>	*Plain*	<access control list>	Returns the discretionary access control list associated with the given security descriptor. Win:4.1
group of <security descriptor>	*Plain*	<security identifier>	Returns the security identifier of the group of the specified security descriptor. Win:4.1
owner of <security descriptor>	*Plain*	<security identifier>	Returns the security identifier of the owner of the specified security descriptor. Win:4.1

© 2007 by BigFix, Inc.

Security Identifier

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

Creation Methods

Key Phrase	*Form*	Description
group of <security descriptor>	*Plain*	Returns the SID of the group of the specified security descriptor. Win:4.1
owner of <security descriptor>	*Plain*	Returns the security identifier of the owner of the specified security descriptor. Win:4.1
sid of <local group member>	*Plain*	Returns the security ID for the specified local group member. Win:6.0
sid of <logged on user>	*Plain*	Returns the Security ID (SID) of the user associated with the session's primary access token. With Windows 2003/XP/Vista, this is determined by WTSQueryUserToken. With NT4/2000 it is determined by the apparent shell process running in the given session. This Inspector may fail if run in a non-privileged context. The SID does not exist under Windows 9x. Win:7.0
trustee of <access control entry>	*Plain*	Returns the trustee to whom the specified ACE applies. Win:4.1
user sid of <event log record>	*Plain*	Returns the user security ID for the specified record in the event log. Win:6.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
<security identifier> as string	*Cast*	<string>	Returns the security identifier in string format. Win:4.1
account name of <security identifier>	*Plain*	<string>	Retrieves the name of the account for this SID and the name of the first domain on which this SID is found. Win:4.1
domain name of <security identifier>	*Plain*	<string>	Returns the domain name of the first domain on which the specified SID is found. Win:4.1

Operators

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

© 2007 by BigFix, Inc.

User Objects

Local User

A Local User object is provided to access the user data of the local machine. Note that domain users are not available through this Inspector.

Creation Methods

Key Phrase	*Form*	Description
domain user	*PlainGlobal*	Returns all of the users that are members of the domain for which the machine is a user. Win:4.1
domain user <string>	*NamedGlobal*	Returns the local user object corresponding to the specified name. Win:4.1
local user	*PlainGlobal*	Creates an object with all the local user accounts. Win:1.2
local user <string>	*NamedGlobal*	Creates an object with a named user account. Win:1.2

Properties

Key Phrase	*Form*	Return Type	Description
account disabled flag of <local user>	*Plain*	<boolean>	Indicates that this account is disabled. Win:1.2
account expiration of <local user>	*Plain*	<time>	Returns the time when this account is set to expire. Win:1.2
accounts operator flag of <local user>	*Plain*	<boolean>	This user has the accounts operator privilege. Win:1.2
admin privilege of <local user>	*Plain*	<boolean>	Indicates that the user has a privilege level of 'admin'. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
allowed workstations string of <local user>	*Plain*	<string>	Returns a list of workstations this user is allowed to login to. If this string is empty, no restrictions apply. Win:1.2
application parameter string of <local user>	*Plain*	<string>	Returns a string used by Microsoft products to store user configuration information. Win:1.2
bad password count of <local user>	*Plain*	<integer>	Returns the number of attempts to logon to this account with a bad password. Win:1.2
code page of <local user>	*Plain*	<integer>	Returns the code page for the user's preferred language. Win:1.2
comment of <local user>	*Plain*	<string>	Returns the comment associated with this user's account. Win:1.2
communications operator flag of <local user>	*Plain*	<boolean>	This user has the communications operator privilege. Win:1.2
country code of <local user>	*Plain*	<integer>	Returns the country code of the user's preferred language. Win:1.2
full name of <local user>	*Plain*	<string>	Returns the full name of the user. Win:1.2
guest privilege of <local user>	*Plain*	<boolean>	Indicates that the user has a privilege level of 'guest'. Win:1.2
home directory drive of <local user>	*Plain*	<string>	Returns the name of the drive assigned to the user's home directory. Win:1.2
home directory of <local user>	*Plain*	<string>	Returns the directory where the user files are stored for the particular user. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
home directory required flag of <local user>	*Plain*	<boolean>	Indicates that a home directory is required for the user. Win:1.2
interdomain trust account flag of <local user>	*Plain*	<boolean>	This is an account which specifies that a domain should trust other domains. Win:1.2
last logoff of <local user>	*Plain*	<time>	Returns the time when the user last logged off. Win:1.2
last logon of <local user>	*Plain*	<time>	Returns the time when the user last logged on. Win:1.2
locked out flag of <local user>	*Plain*	<boolean>	Indicates that this user is currently locked out. Win:1.2
logon count of <local user>	*Plain*	<integer>	Returns the number of times which this account has successfully logged on to the local machine. Win:1.2
logon script of <local user>	*Plain*	<string>	Returns the pathname of this user's logon script. Win:1.2
logon server of <local user>	*Plain*	<string>	Returns the name of the server to which logon requests are sent for this account. Win:1.2
maximum storage of <local user>	*Plain*	<integer>	Returns the user's disk quota. Will return FALSE if the user has no disk quota. Win:1.2
name of <local user>	*Plain*	<string>	Returns the name of the user. Win:1.2
no password required flag of <local user>	*Plain*	<boolean>	Indicates that no password is required for this user. Win:1.2
normal account flag of <local user>	*Plain*	<boolean>	Indicates that this account has a default account type that represents a typical user. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
password age of <local user>	*Plain*	<time interval>	Gives the time since the user's password was last changes. Win:1.2
password change disabled flag of <local user>	*Plain*	<boolean>	Indicates that this user is not allowed to change his password. Win:1.2
password expiration disabled flag of <local user>	*Plain*	<boolean>	Indicates that the password on this account will never expire. Win:1.2
password expired of <local user>	*Plain*	<boolean>	Indicates that the user's password has expired. Win:1.2
primary group id of <local user>	*Plain*	<integer>	Returns the RID of the user's primary group. Win:1.2
print operator flag of <local user>	*Plain*	<boolean>	This user has the print operator privilege. Win:1.2
profile folder of <local user>	*Plain*	<string>	Returns the pathname of the folder which contains the user's profile. Win:1.2
script flag of <local user>	*Plain*	<boolean>	Indicates that the logon script executed. Win:1.2
server operator flag of <local user>	*Plain*	<boolean>	This user has the server operator privilege. Win:1.2
server trust account flag of <local user>	*Plain*	<boolean>	This is an account for a backup domain controller. Win:1.2
temporary duplicate account flag of <local user>	*Plain*	<boolean>	Indicates that this is a temporary duplicate account. Win:1.2
user comment of <local user>	*Plain*	<string>	Returns the user comment of this user. Win:1.2
user id of <local user>	*Plain*	<integer>	Returns the user's RID number. Win:1.2

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
user privilege of <local user>	*Plain*	<boolean>	Indicates that the user has a privilege level of 'user'. Win:1.2
workstation trust account flag of <local user>	*Plain*	<boolean>	This account is for a workstation or server. Win:1.2

Examples

- `exists local user "Administrator"`
- TRUE if there exists a local user named Administrator.

- `exists local user whose (bad password count of it > 5)`
- TRUE if there have been more than 5 bad password attempts on this account.

- `password age of local user "Administrator" > 30 * day`
- TRUE if the Administrator's password is older than 30 days.

Current User

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

Examples

- `local user (name of current user)`
- Provides access to all the local user properties of the currently logged-in user.

Logged On User

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

Creation Methods

Key Phrase	*Form*	Description
current user	*PlainGlobal*	Returns the active, console (local) user, if logged on. Otherwise does not exist. Win:7.0
logged on user	*PlainGlobal*	Returns the user logged on to this BES Client. This Inspector iterates through all logged-on users, using Terminal Services, ACLs, and on Win 9x, the registry. Win:7.0

© 2007 by BigFix, Inc.

Properties

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

© 2007 by BigFix, Inc.

Action Objects

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

Action

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
group leader of <action>	*Plain*	<boolean>	Returns TRUE if the action is a group action and the action component is the group leader. When you deploy a mult-action from the BES Console, it constructs a group action with a group leader to control the overall behavior of the action. This inspector is used internally to manage the progress of the group action. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0
id of <action>	*Plain*	<integer>	Returns the action id of the action. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
last change time of <action>	*Plain*	<time>	Returns the time when the action state last changed. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
offer accepted of <action>	*Plain*	<boolean>	Returns TRUE when users indicated they want to run the action by accepting the offer presented by the BES Client UI. When an offer has been accepted, the Client evaluates its constraints and runs as soon as conditions allow. Win:7.0
offer of <action>	*Plain*	<boolean>	Returns TRUE when the Action is presented as an offer (as indicated by the header "x-offer: 1"). Win:7.0
origin fixlet id of <action>	*Plain*	<integer>	Returns the Fixlet id that contained the action. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
parameter <string> of <action>	*Named*	<string>	Returns the value of parameter <string> for the active action. Parameters only live as long as the action is active. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
pending login of <action>	*Plain*	<boolean>	Returns TRUE if the specified action included an 'action requires login' command, and a login has not yet occurred since the action has run. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
pending of <action>	*Plain*	<boolean>	Returns TRUE if action is available to run. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
pending restart of <action>	*Plain*	<boolean>	Returns TRUE if the specified action included an 'action requires restart' command and a restart has not occurred since the action has run. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
pending time of <action>	*Plain*	<time>	Returns the time the action became pending. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
status of <action>	*Plain*	<string>	Returns one of the following strings: • Running = when the action is currently active. • Executed = no longer relevant and action has completed. • Not Relevant = action was not relevant. • Waiting = action is relevant, but waiting to run. • Not Executed = action is relevant, unconstrained, but has not yet started. • Failed = action is relevant, unconstrained, has completed, but is still relevant. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1
waiting for download of <action>	*Plain*	<boolean>	Returns TRUE if client is waiting for mirroring server to have downloads required by the action. Win:1.2, Lin:3.1, Sol:3.1, HPUX:4.0, AIX:4.1, Mac:4.1

© 2007 by BigFix, Inc.

Networking Objects

This chapter includes the various networking Inspectors.

Network

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

Creation Methods

Key Phrase	*Form*	Description
network	*PlainGlobal*	Creates an object containing properties of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
adapter of <network>	*Plain*	<network adapter>	Returns the one or more network adapter objects of the network. Win:2.0
connection of <network>	*Plain*	<connection>	Returns a connection to the specified network. This Inspector requires Windows XP or better. Win:5.0
dns server of <network>	*Plain*	<network address list>	Returns a list of DNS servers used by the local computer. Win:4.1
interface <integer> of <network>	*Numbered*	<network interface>	Returns the particular interface of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
interface of <network>	*Plain*	<network interface>	Returns all the interfaces of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ip interface <integer> of <network>	*Numbered*	<network ip interface>	Returns the particular ip interface of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ip interface of <network>	*Plain*	<network ip interface>	Returns all the ip interfaces of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
winsock2 supported of <network>	*Plain*	<boolean>	Indicates that winsock2 is supported by the network. If this returns FALSE, many of the other properties of the interface are not available for inspection. Win:1.2

Network Interface

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

Creation Methods

Key Phrase	*Form*	Description
interface <integer> of <network>	*Numbered*	Creates an object with the specified network interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
interface of <network>	*Plain*	Creates an object with all the interfaces of the network. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

Properties

Key Phrase	*Form*	Return Type	Description
family of <network interface>	*Plain*	<integer>	Returns an family designator of the address family (i.e., 2=AFI_NET). Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

Examples

- `names of interfaces of network`
- Returns a list of the network interface names, e.g., lo0, gif0, stf0, en0.

© 2007 by BigFix, Inc.

Network Ip Interface

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
point to point of <network ip interface>	*Plain*	<boolean>	Indicates that the interface is a point-to-point interface. Usually TRUE for dialup connections. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
subnet address of <network ip interface>	*Plain*	<ipv4 address>	The subnet to which the interface belongs. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
subnet mask of <network ip interface>	*Plain*	<ipv4 address>	The subnet mask of the interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1

Examples

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

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

- `address of ip interface whose (loopback of it = false) of network = "192.168.127.127"`
- Returns TRUE if the given IP address doesn't have loopback.

Network Share

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

Creation Methods

Key Phrase	*Form*	Description
network share	*PlainGlobal*	Creates a network shared object. Win:4.1
network share <string>	*NamedGlobal*	Creates a named network shared object. Win:4.1

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
attribute permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to modify the attributes of the shared resource (such as the date and time when a file was last modified). Win:4.1
comment of <network share>	*Plain*	<string>	Returns a string specifying an optional comment about the shared resource. Win:4.1
create permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to create an instance of a shared resource (such as a file). Win:4.1
delete permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to delete the resource. Win:4.1
execute permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to execute the resource. Win:4.1
name of <network share>	*Plain*	<string>	Returns a string specifying the name of the specified shared resource. Win:4.1
password of <network share>	*Plain*	<string>	A string that specifies the share's password (when the server is running with share-level security). Note that the Windows Server 2003 family, Windows XP, Windows 2000, and Windows NT do not support share-level security. Win:4.1
path of <network share>	*Plain*	<string>	A string containing the local path for the shared resource. Win:4.1
permission permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to modify the permissions (read, write, create, execute, and delete) for the specified network share. Win:4.1
read permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to read data from a resource and, by default, to execute

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
			the specified network share. Win:4.1
security descriptor of <network share>	*Plain*	<security descriptor>	Specifies the security descriptor associated with the specified network share. Win:4.1
type of <network share>	*Plain*	<integer>	Specifies an integer value that indicates the type of share. (See the Microsoft document on SHARE_INFO_502). Win:4.1
use count of <network share>	*Plain*	<integer>	Specifies an integer value that indicates the number of current connections to the specified network share. Win:4.1
use limit of <network share>	*Plain*	<integer>	Specifies an integer value indicating the maximum number of concurrent connections that the shared resource can accommodate. Win:4.1
write permission of <network share>	*Plain*	<boolean>	Returns TRUE if permission is granted to write data to the specified network share. Win:4.1

Network Adapter

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

Creation Methods

Key Phrase	*Form*	Description
adapter of <network>	*Plain*	Returns one or more adapters of the network. Win:2.0

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
address list of <network adapter>	*Plain*	<network address list>	Returns the address list of the network adapter. Win:2.0
address of <network adapter>	*Plain*	<ipv4 address>	Returns the ip address of the network adapter (returns the first address if it is a list). Win:2.0
description of <network adapter>	*Plain*	<string>	Returns the description of the network adapter. Win:2.0
dhcp enabled of <network adapter>	*Plain*	<boolean>	Returns TRUE if dhcp is enabled on the network adapter. Win:2.0
dhcp server of <network adapter>	*Plain*	<ipv4 address>	Returns the ip address of the dhcp server of the network adapter. Win:2.0
dns server of <network adapter>	*Plain*	<network address list>	Returns a list of DNS servers used by the specified adapter. Win:4.1
dns suffix of <network adapter>	*Plain*	<string>	Returns the Domain Name System (DNS) suffix associated with the specified adapter. Win:7.0
friendly name of <network adapter>	*Plain*	<string>	Returns a user-friendly name for the adapter, for example "Local Area Connection 1". Win:7.0
gateway list of <network adapter>	*Plain*	<network address list>	Returns the gateway network address list of the network adapter. Win:2.0
gateway of <network adapter>	*Plain*	<ipv4 address>	Returns the ip address of the gateway of the network adapter. Win:2.0
internet connection firewall of <network adapter>	*Plain*	<internet connection firewall>	Creates a Windows XP firewall object. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
ipv6 address of <network adapter>	*Plain*	<ipv6 address>	Returns the local IPv6 address of the adapter. Only for XP/Server 2003 and later. Win:7.0
ipv6 dns server of <network adapter>	*Plain*	<ipv6 address>	Returns the DNS server IPv6 address of the adapter. Only for XP/Server 2003 and later. Win:7.0
lease expires of <network adapter>	*Plain*	<time>	Returns the time that the dhcp lease will expire of the network adapter. Win:2.0
lease obtained of <network adapter>	*Plain*	<time>	Returns the time that the dhcp lease was obtained of the network adapter. Win:2.0
link speed of <network adapter>	*Plain*	<integer>	This is a property of a network adapter. It returns the maximum speed of the NIC card in bits per second. Win:6.0
mac address of <network adapter>	*Plain*	<string>	Returns the mac address of the network adapter. Win:2.0
maximum transmission unit of <network adapter>	*Plain*	<integer>	The maximum transmission unit (MTU) size, in bytes, of the specified adapter. Win:7.0
name of <network adapter>	*Plain*	<string>	Returns the name of the network adapter. Win:2.0
primary wins server of <network adapter>	*Plain*	<ipv4 address>	Returns the ip address of the primary wins server of the network adapter. Win:2.0
secondary wins server of <network adapter>	*Plain*	<ipv4 address>	Returns the ip address of the secondary wins server of the network adapter. Win:2.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
status of <network adapter>	*Plain*	<integer>	The operational status for the interface as defined in RFC 2863. It can be one of the values from the IF_OPER_STATUS enumeration type defined in the Iftypes.h header file. On Windows Vista and later, the header files were reorganized and this enumeration is defined in the Ifdef.h header file. Win:7.0
subnet address of <network adapter>	*Plain*	<ipv4 address>	Returns the subnet address of the network adapter. Win:2.0
subnet mask of <network adapter>	*Plain*	<ipv4 address>	Returns the subnet mask of the network adapter. Win:2.0
type of <network adapter>	*Plain*	<integer>	Returns the interface type of the specified adapter as defined by the Internet Assigned Names Authority (IANA). Possible values for the interface type are listed in the Ipifcons.h header file. Win:7.0
wakeonlan enabled of <network adapter>	*Plain*	<boolean>	Returns TRUE if the specified network adapter is configured to react to Wake-On-Lan requests. Wake-On-Lan is a mechanism used to trigger a boot of a machine in standby mode by sending a special packet. • Note: Wake-On-Lan is only supported for Windows 2000 and XP machines. Win:5.1
wins enabled of <network adapter>	*Plain*	<boolean>	Returns TRUE if WINS is enabled on the network adapter. Win:2.0

© 2007 by BigFix, Inc.

Network Address List

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

Creation Methods

Key Phrase	*Form*	Description
address list of <network adapter>	*Plain*	Returns the address list of the network adapter. Win:2.0
dns server of <network adapter>	*Plain*	Returns a list of DNS servers used by the specified adapter. Win:4.1
dns server of <network>	*Plain*	Returns a list of DNS servers used by the local computer. Win:4.1
gateway list of <network adapter>	*Plain*	Returns the gateway network address list of the network adapter. Win:2.0

Properties

Key Phrase	*Form*	Return Type	Description
address of <network address list>	*Plain*	<ipv4 address>	Returns the address of the address list. Win:2.0
subnet address of <network address list>	*Plain*	<ipv4 address>	Returns the subnet address of the network address list. Win:2.0
subnet mask of <network address list>	*Plain*	<ipv4 address>	Returns the subnet mask of the network address list. Win:2.0

© 2007 by BigFix, Inc.

Internet Connection Firewall

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

Creation Methods

Key Phrase	*Form*	Description
firewall of <connection>	*Plain*	Returns the internet connection firewall object corresponding to the specified connection. Win:5.0
internet connection firewall of <network adapter>	*Plain*	Creates a Windows XP firewall object. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
enabled of <internet connection firewall>	*Plain*	<boolean>	Returns TRUE if the local computer has the Windows XP built-in firewall enabled. Win:4.1
port mapping of <internet connection firewall>	*Plain*	<port mapping>	Creates a port mapping object for the built-in firewall. Win:4.1

Port Mapping

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

Creation Methods

Key Phrase	*Form*	Description
port mapping of <internet connection firewall>	*Plain*	Creates a port mapping object for the built-in firewall. Win:4.1

© 2007 by BigFix, Inc.

Properties

Key Phrase	*Form*	Return Type	Description
enabled of <port mapping>	*Plain*	<boolean>	A boolean indicating whether or not the port mapping is enabled. Win:4.1
external port of <port mapping>	*Plain*	<integer>	Returns the external port number of the specified ICF port mapping. Win:4.1
internal port of <port mapping>	*Plain*	<integer>	Returns the internal port number of the specified ICF port mapping. Win:4.1
name of <port mapping>	*Plain*	<string>	Returns the name of the specified port mapping. Win:4.1
options of <port mapping>	*Plain*	<integer>	See port mapping at MSDN. Win:4.1
protocol of <port mapping>	*Plain*	<string>	Returns a string like "tcp" or "udp", corresponding to the protocol of the specified port mapping. Win:4.1
target ip address of <port mapping>	*Plain*	<ipv4 address>	Returns the target IP address of the specified port mapping. Win:4.1
target name of <port mapping>	*Plain*	<string>	Returns the target name of the specified port mapping. Win:4.1

© 2007 by BigFix, Inc.

Connection

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

Creation Methods

Key Phrase	*Form*	Description
connection of <network>	*Plain*	Returns a connection to the specified network. This Inspector requires Windows XP or better. Win:5.0

Properties

Key Phrase	*Form*	Return Type	Description
device name of <connection>	*Plain*	<string>	Returns the name of the device associated with the specified connection. Win:5.0
firewall of <connection>	*Plain*	<internet connection firewall>	Returns the internet connection firewall object corresponding to the specified connection. Win:5.0
guid of <connection>	*Plain*	<string>	Returns the globally-unique identifier (GUID) for the specified connection. Win:5.0
media type of <connection>	*Plain*	<media type>	Returns the media type of for the specified connection. Win:5.0
name of <connection>	*Plain*	<string>	Returns the name of the specified connection. Win:5.0
status of <connection>	*Plain*	<connection status>	Returns the status of the specified connection. Win:5.0

© 2007 by BigFix, Inc.

Connection Status

This object returns information about the status of your connections.

Creation Methods

Key Phrase	*Form*	Description
connection status <integer>	*NumberedGlobal*	Returns the connection status based on its integer value. This Inspector is included to take advantage of new (or undocumented) additions to the status values. Win:5.0
connection status authenticating	*PlainGlobal*	Returns the value NCS_AUTHENTICATING: The connection is waiting for authentication to occur. Win:5.0
connection status authentication failed	*PlainGlobal*	Returns the value NCS_AUTHENTICATION_FAILED: Authentication has failed on this connection. Win:5.0
connection status authentication succeeded	*PlainGlobal*	Returns the value NCS_AUTHENTICATION_SUCCEEDED: Authentication has succeeded on this connection. Win:5.0
connection status connected	*PlainGlobal*	Returns the value NCS_CONNECTED: The connection is in a connected state. Win:5.0
connection status connecting	*PlainGlobal*	Returns the value NCS_CONNECTING: The connection is in the process of connecting. Win:5.0
connection status disconnected	*PlainGlobal*	Returns the value NCS_DISCONNECTED: The connection is disconnected. Win:5.0
connection status disconnecting	*PlainGlobal*	Returns the value NCS_DISCONNECTING: The connection is in the process of disconnecting. Win:5.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
connection status hardware disabled	*PlainGlobal*	Returns the value NCS_HARDWARE_DISABLED: The hardware for the connection is present, but is not enabled. Win:5.0
connection status hardware malfunction	*PlainGlobal*	Returns the value NCS_HARDWARE_MALFUNCTION: A malfunction has occurred in the hardware for the connection. Win:5.0
connection status media disconnected	*PlainGlobal*	Returns the value NCS_MEDIA_DISCONNECTED: The media, for example the network cable, is disconnected. Win:5.0
connection status no hardware present	*PlainGlobal*	Returns the value NCS_NO_HARDWARE_PRESENT: The hardware for the connection, for example network interface card (NIC), is not present. Win:6.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
status of <connection>	*Plain*	Returns the current status of the connection. You can compare this status to the 'connection status' global objects, which act as constants. Some of the possible values include: • NCS_DISCONNECTED: The connection is disconnected. • NCS_CONNECTING: The connection is in the process of connecting. • NCS_CONNECTED: The connection is in a connected state. • NCS_DISCONNECTING: The connection is in the process of disconnecting. • NCS_HARDWARE_NOT_PRESENT: The hardware for the connection, for example network interface card (NIC), is not present. • NCS_HARDWARE_DISABLED: The hardware for the connection is present, but is not enabled. • NCS_HARDWARE_MALFUNCTION: A malfunction has occurred in the hardware for the connection. • NCS_MEDIA_DISCONNECTED: The media, for example the network cable, is disconnected. • NCS_AUTHENTICATING: The connection is waiting for authentication to occur. • NCS_AUTHENTICATION_SUCCEEDED: Authentication has succeeded on this connection. • NCS_AUTHENTICATION_FAILED: Authentication has failed on this connection. • NCS_INVALID_ADDRESS: The address is invalid. • NCS_CREDENTIALS_REQUIRED: Security credentials are required. Win:5.0

Operators

Key phrase	Return Type	Description
<connection status> = <connection status>	*<boolean>*	Compare the statuses of two connections. Win:5.0

© 2007 by BigFix, Inc.

Media Type

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

Creation Methods

Key Phrase	*Form*	Description
media type <integer>	*NumberedGlobal*	Returns the media type based on its integer value. This Inspector is included to take advantage of new (or undocumented) additions to the media types. Win:5.0
media type bridge	*PlainGlobal*	Returns the value NCM_BRIDGE: Bridged connection. Win:5.0
media type direct	*PlainGlobal*	Returns the value NCM_DIRECT: Direct serial connection through a serial port. Win:5.0
media type isdn	*PlainGlobal*	Returns the value NCM_ISDN: Connection is through an integrated services digital network (ISDN) line. Win:5.0
media type lan	*PlainGlobal*	Returns the value NCM_LAN: Connection is to a local area network (LAN). Win:5.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
media type of <connection>	*Plain*	Returns the media type of the specified connection: • NCM_NONE: No media is present. • NCM_DIRECT: Direct serial connection through a serial port. • NCM_ISDN: Connection is through an integrated services digital network (ISDN) line. • NCM_LAN: Connection is to a local area network (LAN). • NCM_PHONE: Dial-up connection over a conventional phone line. • NCM_TUNNEL: Virtual private network (VPN) connection. • NCM_PPPOE: Point-to-Point protocol (PPP) over Ethernet. • NCM_BRIDGE: Bridged connection. • NCM_SHAREDACCESSHOST_LAN: Shared connection to a LAN. • NCM_SHAREDACCESSHOST_RAS: Shared connection to a remote or wide area network (WAN). Win:5.0
media type phone	*PlainGlobal*	Returns the value NCM_PHONE: Dial-up connection over a conventional phone line. Win:5.0
media type pppoe	*PlainGlobal*	Returns the value NCM_PPPOE: Point-to-Point protocol (PPP) over Ethernet. Win:5.0
media type shared access host lan	*PlainGlobal*	Returns the value NCM_SHAREDACCESSHOST_LAN: Shared connection to a LAN. Win:5.0
media type shared access host ras	*PlainGlobal*	Returns the value NCM_SHAREDACCESSHOST_RAS: Shared connection to a remote or wide area network (WAN). Win:5.0
media type tunnel	*PlainGlobal*	Returns the value NCM_TUNNEL: Virtual private network (VPN) connection. Win:5.0

© 2007 by BigFix, Inc.

Operators

Key phrase	Return Type	Description
<media type> = <media type>	*<boolean>*	Compares two media types. Win:5.0

Ipv4 Address

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

Creation Methods

Key Phrase	*Form*	Description
address of <network adapter>	*Plain*	Returns the ip address of the network adapter. Win:2.0
address of <network address list>	*Plain*	Returns the ip address of the network adapter list. Win:2.0
address of <network ip interface>	*Plain*	Creates an object with the ip address of the interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
broadcast address of <network ip interface>	*Plain*	Creates an object with the broadcast address of the interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
dhcp server of <network adapter>	*Plain*	Returns the ip address of the dhcp server of the network adapter. Win:2.0
gateway address <integer> of <selected server>	*Numbered*	The ip address of a gateway between the agent and the selected server at the given distance from the agent, if known. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
gateway address of <selected server>	*Plain*	All known ip addresses of gateways between the agent and the selected server. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
gateway of <network adapter>	*Plain*	Returns the ip address of the gateway of the network adapter. Win:2.0

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
ip address of <selected server>	*Plain*	The ip address to which reports are sent. Win:4.1, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ipv4 address <string>	*NamedGlobal*	Creates an object with an ip address for the string provided. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
ipv4 part of <ipv6 address>	*Plain*	Returns the lowest 32-bits of the IPv6 address as an IPv4 address. Win:7.0
primary wins server of <network adapter>	*Plain*	Returns the ip address of the primary wins server of the network adapter. Win:2.0
secondary wins server of <network adapter>	*Plain*	Returns the ip address of the secondary wins server of the network adapter. Win:2.0
subnet address of <network adapter>	*Plain*	Returns the subnet address of the network adapter. Win:2.0
subnet address of <network address list>	*Plain*	Returns the subnet address of the network address list. Win:2.0
subnet address of <network ip interface>	*Plain*	Creates an object with the subnet address of the network interface. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
subnet mask of <network adapter>	*Plain*	Returns the subnet mask of the network adapter. Win:2.0
subnet mask of <network address list>	*Plain*	Returns the subnet mask of the network adapter list. Win:2.0
subnet mask of <network ip interface>	*Plain*	Creates an object with the address bitwise ANDed with the subnet mask. Win:1.2, Lin:4.1, Sol:4.1, HPUX:4.1, AIX:4.1, Mac:5.1
target ip address of <port mapping>	*Plain*	Returns the target IP address of the specified port mapping. Win:4.1

© 2007 by BigFix, Inc.

Properties

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

Operators

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

Examples

- `exists ip interface whose (address of it = "127.0.0.1" and loopback of it) of network`
  - Returns TRUE if the specified ip interface (with loopback) exists on this computer.
- `addresses of ip interfaces of network`
  - Returns a list of IP addresses configured on the machine.
- `ipv4 address "192.168.100.1"`
  - Returns the four-byte ip address 192.168.100.1.

© 2007 by BigFix, Inc.

Ipv6 Address

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

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
<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
ipv6 address <string>	*NamedGlobal*	Converts a string representations of an IPv6 address (with colons and/or dots) as an IPv6 address type. Win:7.0
ipv6 address of <network adapter>	*Plain*	Returns the local IPv6 address of the adapter. Only for XP/Server 2003 and later. Win:7.0
ipv6 dns server of <network adapter>	*Plain*	Returns the DNS server IPv6 address of the adapter. Only for XP/Server 2003 and later. Win:7.0

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
<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

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
<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
<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
<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
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
ipv4 part of <ipv6 address>	*Plain*	<ipv4 address>	Returns the lowest 32-bits of the IPv6 address as an IPv4 address. Win:7.0

Operators

Key phrase	Return Type	Description
<ipv6 address> {cmp} <ipv6 address>	*<boolean>*	Where {cmp} is one of: <, <=, =. Win:7.0

© 2007 by BigFix, Inc.

Active Directory Local Computer

These are the Active Directory Inspectors for the local computer.

Creation Methods

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

Properties

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

Examples

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

© 2007 by BigFix, Inc.

Active Directory Server

These are the Active Directory Server Inspectors.

Creation Methods

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

Properties

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

Distinguished Name

These Inspectors refer to the distinguished name (DN) as defined by the Microsoft Active Directory service.

Creation Methods

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

© 2007 by BigFix, Inc.

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.

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
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

© 2007 by BigFix, Inc.

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".

© 2007 by BigFix, Inc.

Microsoft IIS Metabase Objects

The Microsoft IIS Metabase is a repository for most IIS configuration values. The following inspectors retrieve information about the Microsoft IIS Metabase.

Metabase

The IIS metabase is similar in structure to the Windows Registry, providing hierarchal storage of IIS configuration properties for Web sites, virtual directories, FTP, etc.

Creation Methods

Key Phrase	*Form*	Description
metabase	*PlainGlobal*	Returns the IIS metabase object. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
key <string> of <metabase>	*Named*	<metabase key>	Returns the named key of the metabase. Win:4.1
key of <metabase>	*Plain*	<metabase key>	Returns the root key of the IIS metabase. Win:4.1

© 2007 by BigFix, Inc.

Metabase Identifier

A metabase identifier is a unique numerical identifier which IIS recognizes internally.

Creation Methods

Key Phrase	*Form*	Description
identifier of <metabase value>	*Plain*	Returns the metabase identifier of the specified value. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<metabase identifier> as integer	*Cast*	<integer>	The integer value of the specified identifier. Use the Microsoft MetaEdit utility to find the integer value of an identifier. Win:4.1

Operators

Key phrase	Return Type	Description
<metabase identifier> = <metabase identifier>	*<boolean>*	Compares two metabase identifiers for equality. Win:4.1

© 2007 by BigFix, Inc.

Metabase Key

Like a registry key, a metabase key contains named properties about the IIS metabase.

Creation Methods

Key Phrase	*Form*	Description
key <string> of <metabase key>	*Named*	Returns the named subkey of the specified metabase key. Win:4.1
key <string> of <metabase>	*Named*	Returns the named key of the metabase. Win:4.1
key of <metabase key>	*Plain*	Iterates the subkeys of the specified metabase key. Win:4.1
key of <metabase>	*Plain*	Returns the root key of the IIS metabase. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
key <string> of <metabase key>	*Named*	<metabase key>	Returns the named subkey of the specified metabase key. Win:4.1
key of <metabase key>	*Plain*	<metabase key>	Iterates the subkeys of the specified metabase key. Win:4.1
name of <metabase key>	*Plain*	<string>	Returns the name of the specified metabase key. Win:4.1
value of <metabase key>	*Plain*	<metabase value>	Returns the value of the specified metabase key. Win:4.1

Examples

- `names of keys of metabase`
- Depending on the metabase configuration, returns the names of the metabase keys, such as:

• LM
• Schema.

© 2007 by BigFix, Inc.

- `names of keys of key "/LM" of metabase`

Depending on the metabase configuration, returns the key names in the metabase "/LM" key, such as IISADMIN, W3SVC or MimeMap.

Metabase Type

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

Creation Methods

Key Phrase	*Form*	Description
type of <metabase value>	*Plain*	Returns the type of the specified metabase value. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<metabase type> as integer	*Cast*	<integer>	Meanings same as registry types: • 1 = DWORD • 2 = STRING • 3 = BINARY • 4 = EXPANDSZ • 5 = MULTISZ. Win:4.1
<metabase type> as string	*Cast*	<string>	Returns metabase types as strings: • "DWord" • "String" • "Binary" • "ExpandSz" • "MultiSz". Win:4.1

Operators

Key phrase	Return Type	Description
<metabase type> = <metabase type>	*<boolean>*	Compares two metabase types for equality. Win:4.1

© 2007 by BigFix, Inc.

Metabase User Type

The user type is a DWORD that specifies how the property value is used. User types enable IIS to classify properties by application.

Creation Methods

Key Phrase	*Form*	Description
user type of <metabase value>	*Plain*	Returns the user type of the specified metabase value. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<metabase user type> as integer	*Cast*	<integer>	Returns the metabase user type as an integer: • 2 = Server • 2 = File • 100 = WAM • 200 = ASP App. Win:4.1
<metabase user type> as string	*Cast*	<string>	Returns the metabase user type as a string: • "Server" • "File" • "WAM" • "ASP App" • "(Other)". Win:4.1

Operators

Key phrase	Return Type	Description
<metabase user type> = <metabase user type>	*<boolean>*	Compares two metabase user types for equality. Win:4.1

© 2007 by BigFix, Inc.

Metabase Value

This Inspector is used to access values stored in an IIS metabase key. The type of the data stored in the value determines what casting operations are allowed. There are several casting Inspectors that you can use to extract values from the registry.

Creation Methods

Key Phrase	*Form*	Description
value of <metabase key>	*Plain*	Returns the value of the specified metabase key. Win:4.1

Properties

Key Phrase	*Form*	Return Type	Description
<metabase value> as integer	*Cast*	<integer>	Returns the integer value of the metabase value. Win:4.1
<metabase value> as string	*Cast*	<string>	Returns the string value of the metabase value. Win:4.1
identifier of <metabase value>	*Plain*	<metabase identifier>	Returns the metabase identifier of the specified value. Win:4.1
inherit attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the inherit attribute set. Win:4.1
insert path attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the insert path attribute set. Win:4.1
reference attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the reference attribute set. Win:4.1
secure attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the secure attribute set. Win:4.1
type of <metabase value>	*Plain*	<metabase type>	Returns the type of the specified metabase value. Win:4.1

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Return Type	Description
user type of <metabase value>	*Plain*	<metabase user type>	Returns the user type of the specified metabase value. Win:4.1
volatile attribute of <metabase value>	*Plain*	<boolean>	Returns TRUE if the specified metabase value has the volatile attribute set. Win:4.1

Examples

- `inherit attributes of values of key "/Schema" of metabase`

Returns a boolean True or False depending on the inherit attributes of each sub-key in the specified key of the metabase.

- `volatile attributes of values of key "/LM" of metabase`

Returns a boolean True or False depending on the volatile attributes of each sub-key in the specified key of the metabase.

© 2007 by BigFix, Inc.

Introspectors

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

Type

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

Creation Methods

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

© 2007 by BigFix, Inc.

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

Properties

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

© 2007 by BigFix, Inc.

Operators

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

Examples

- `properties "lines" of type "file"`
- Returns the various line properties of the "file" type.

- `properties returning (type "file line") of type "file"`
- Returns the list of properties that return the <type> specified given an object of type <type>.

Property

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

Creation Methods

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

© 2007 by BigFix, Inc.

Key Phrase	*Form*	Description
property returning <type>	*IndexedGlobal*	Produces a list of the Inspector properties that return the "file" type. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0
property returning <type> of <type>	*Indexed*	Returns Inspectors of the form <type> of <type>. Typically there is more than one property, so this is often used in the plural. Win:6.0, Lin:6.0, Sol:6.0, HPUX:6.0, AIX:6.0, Mac:6.0

Properties

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

© 2007 by BigFix, Inc.

Examples

- `property "first matches"`
  - Returns the list of properties accessed by the string provided.
- `usual name of property "booleans"`
  - Returns "boolean".

Binary Operator

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

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

Examples

- `binary operators "&"`
- Returns a list of all the concatentation inspectors available.

Unary Operator

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

Creation Methods

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

© 2007 by BigFix, Inc.

Properties

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

Examples

- `unary operators "-"`
  - Returns a list of the objects that can be made negative, such as integers, floating point numbers, etc.

- `unary operators returning (type "hertz")`
  - Returns - <hertz>: hertz, the only unary operator that returns a hertz object.

© 2007 by BigFix, Inc.

Cast

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

Creation Methods

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

Properties

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

© 2007 by BigFix, Inc.

Examples

- `casts "integer"`

Returns a list of the objects that can be cast as integers, eg., <string> as integer, <integer> as integer, etc.

- `casts returning (type "integer")`

Returns.

© 2007 by BigFix, Inc.

Key Phrases (Inspectors)

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

Key phrases

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

Key Phrase	Plural	Creates a	From a	*Form*
absolute value of <hertz>	absolute values	<hertz>	<hertz>	*Plain*
absolute value of <integer>	absolute values	<integer>	<integer>	*Plain*
absolute value of <time interval>	absolute values	<time interval>	<time interval>	*Plain*
access mode of <access control entry>	access modes	<integer>	<access control entry>	*Plain*
accessed time of <filesystem object>	accessed times	<time>	<filesystem object>	*Plain*
account disabled flag of <local user>	account disabled flags	<boolean>	<local user>	*Plain*
account expiration of <local user>	account expirations	<time>	<local user>	*Plain*
account name of <security identifier>	account names	<string>	<security identifier>	*Plain*
accounts operator flag of <local user>	accounts operator flags	<boolean>	<local user>	*Plain*
action	actions	<action>	<world>	*PlainGlobal*
action <integer>	actions	<action>	<world>	*NumberedGlobal*
action lock state	action lock states	<action lock state>	<world>	*PlainGlobal*
action of <firewall rule>	actions	<firewall action>	<firewall rule>	*Plain*
active action	active actions	<action>	<world>	*PlainGlobal*
active device	active devices	<active device>	<world>	*PlainGlobal*
active device file	active device files	<file>	<world>	*PlainGlobal*
active device file <string>	active device files	<file>	<world>	*NamedGlobal*
active of <action>	actives	<boolean>	<action>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
active of <logged on user>	actives	<boolean>	<logged on user>	*Plain*
active start time of <action>	active start times	<time>	<action>	*Plain*
adapter of <network>	adapters	<network adapter>	<network>	*Plain*
address list of <network adapter>	address lists	<network address list>	<network adapter>	*Plain*
address of <network adapter>	addresses	<ipv4 address>	<network adapter>	*Plain*
address of <network address list>	addresses	<ipv4 address>	<network address list>	*Plain*
address of <network ip interface>	addresses	<ipv4 address>	<network ip interface>	*Plain*
admin privilege of <local user>	admin privileges	<boolean>	<local user>	*Plain*
administrator <string> of <client>	administrators	<setting>	<client>	*Named*
administrator of <client>	administrators	<setting>	<client>	*Plain*
all firewall scope	all firewall scopes	<firewall scope>	<world>	*PlainGlobal*
allow firewall action	allow firewall actions	<firewall action>	<world>	*PlainGlobal*
allow inbound echo request of <firewall icmp settings>	allow inbound echo requests	<boolean>	<firewall icmp settings>	*Plain*
allow inbound mask request of <firewall icmp settings>	allow inbound mask requests	<boolean>	<firewall icmp settings>	*Plain*
allow inbound router request of <firewall icmp settings>	allow inbound router requests	<boolean>	<firewall icmp settings>	*Plain*
allow inbound timestamp request of <firewall icmp settings>	allow inbound timestamp requests	<boolean>	<firewall icmp settings>	*Plain*
allow outbound destination unreachable of <firewall icmp settings>	allow outbound destination unreachables	<boolean>	<firewall icmp settings>	*Plain*
allow outbound packet too big of <firewall icmp settings>	allow outbound packet too bigs	<boolean>	<firewall icmp settings>	*Plain*
allow outbound parameter problem of <firewall icmp settings>	allow outbound parameter problems	<boolean>	<firewall icmp settings>	*Plain*
allow outbound source quench of <firewall icmp settings>	allow outbound source quenches	<boolean>	<firewall icmp settings>	*Plain*
allow outbound time exceeded of <firewall icmp settings>	allow outbound time exceededs	<boolean>	<firewall icmp settings>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
allow redirect of <firewall icmp settings>	allow redirects	<boolean>	<firewall icmp settings>	*Plain*
allowed workstations string of <local user>	allowed workstations strings	<string>	<local user>	*Plain*
ancestor of <filesystem object>	ancestors	<folder>	<filesystem object>	*Plain*
ansi code page	ansi code pages	<integer>	<world>	*PlainGlobal*
any ip version	any ip versions	<ip version>	<world>	*PlainGlobal*
apparent registration server time	apparent registration server times	<time>	<world>	*PlainGlobal*
append permission of <access control entry>	append permissions	<boolean>	<access control entry>	*Plain*
application <string>	applications	<application>	<world>	*NamedGlobal*
application <string> of <folder>	applications	<application>	<folder>	*Named*
application <string> of <registry key>	applications	<application>	<registry key>	*Named*
application <string> of <registry>	applications	<application>	<registry>	*Named*
application event log	application event logs	<event log>	<world>	*PlainGlobal*
application folder <string> of <registry key>	application folders	<folder>	<registry key>	*Named*
application folder <string> of <registry>	application folders	<folder>	<registry>	*Named*
application folder of <registry key>	application folders	<folder>	<registry key>	*Plain*
application name of <firewall rule>	application names	<string>	<firewall rule>	*Plain*
application of <registry key>	applications	<application>	<registry key>	*Plain*
application of <registry>	applications	<application>	<registry>	*Plain*
application parameter string of <local user>	application parameter strings	<string>	<local user>	*Plain*
application usage summary	application usage summaries	<application usage summary>	<world>	*PlainGlobal*
application usage summary <string>	application usage summaries	<application usage summary>	<world>	*NamedGlobal*
april	aprils	<month>	<world>	*PlainGlobal*
april <integer>	aprils	<day of year>	<world>	*NumberedGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
april <integer> of <integer>	aprils	<date>	<integer>	*Numbered*
april of <integer>	aprils	<month and year>	<integer>	*Plain*
archive of <filesystem object>	archives	<boolean>	<filesystem object>	*Plain*
argument string of <file shortcut>	argument strings	<string>	<file shortcut>	*Plain*
attribute <integer> of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Numbered*
attribute <string> of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Named*
attribute of <xml dom node>	attributes	<xml dom node>	<xml dom node>	*Plain*
attribute permission of <network share>	attribute permissions	<boolean>	<network share>	*Plain*
audit failure event log event type	audit failure event log event types	<event log event type>	<world>	*PlainGlobal*
audit level of <local mssql database>	audit levels	<integer>	<local mssql database>	*Plain*
audit success event log event type	audit success event log event types	<event log event type>	<world>	*PlainGlobal*
august	augusts	<month>	<world>	*PlainGlobal*
august <integer>	augusts	<day of year>	<world>	*NumberedGlobal*
august <integer> of <integer>	augusts	<date>	<integer>	*Numbered*
august of <integer>	augusts	<month and year>	<integer>	*Plain*
authorized application of <firewall profile>	authorized applications	<firewall authorized application>	<firewall profile>	*Plain*
backoffice bit <operating system suite mask>	backoffice bits	<boolean>	<world>	*IndexedGlobal*
bad password count of <local user>	bad password counts	<integer>	<local user>	*Plain*
bes license	bes licenses	<license>	<world>	*PlainGlobal*
binary operator <string>	binary operators	<binary operator>	<world>	*NamedGlobal*
binary operator returning <type>	binary operators returning	<binary operator>	<world>	*IndexedGlobal*
bit <integer>	bits	<bit set>	<world>	*NumberedGlobal*
bit <integer> of <bit set>	bits	<boolean>	<bit set>	*Numbered*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
bit <integer> of <integer>	bits	<boolean>	<integer>	*Numbered*
bit set <string>	bit sets	<bit set>	<world>	*NamedGlobal*
blade bit <operating system suite mask>	blade bits	<boolean>	<world>	*IndexedGlobal*
block firewall action	block firewall actions	<firewall action>	<world>	*PlainGlobal*
boolean <string>	booleans	<boolean>	<world>	*NamedGlobal*
boolean value <integer> of <wmi select>	boolean values	<boolean>	<wmi select>	*Numbered*
boolean value of <wmi select>	boolean values	<boolean>	<wmi select>	*Plain*
boot time of <operating system>	boot times	<time>	<operating system>	*Plain*
brand id of <processor>	brand ids	<integer>	<processor>	*Plain*
brand string of <processor>	brand strings	<string>	<processor>	*Plain*
broadcast address of <network ip interface>	broadcast addresses	<ipv4 address>	<network ip interface>	*Plain*
broadcast support of <network ip interface>	broadcast supports	<boolean>	<network ip interface>	*Plain*
build number high of <operating system>	build number highs	<integer>	<operating system>	*Plain*
build number low of <operating system>	build number lows	<integer>	<operating system>	*Plain*
build number of <operating system>	build numbers	<integer>	<operating system>	*Plain*
built in of <firewall open port>	built ins	<boolean>	<firewall open port>	*Plain*
byte <integer> of <file>	bytes	<integer>	<file>	*Numbered*
can interact with desktop of <service>	can interact with desktops	<boolean>	<service>	*Plain*
case insensitive regex <string>	case insensitive regexes	<regular expression>	<world>	*NamedGlobal*
case insensitive regular expression <string>	case insensitive regular expressions	<regular expression>	<world>	*NamedGlobal*
cast <string>	casts	<cast>	<world>	*NamedGlobal*
cast from of <type>	casts from	<cast>	<type>	*Plain*
cast returning <type>	casts returning	<cast>	<world>	*IndexedGlobal*
category of <event log record>	categories	<integer>	<event log record>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
change notification permission of <access control entry>	change notification permissions	<boolean>	<access control entry>	*Plain*
character <integer>	characters	<string>	<world>	*NumberedGlobal*
character <integer> of <string>	characters	<substring>	<string>	*Numbered*
character of <string>	characters	<substring>	<string>	*Plain*
checkpoint of <service>	checkpoints	<integer>	<service>	*Plain*
child node <integer> of <xml dom node>	child nodes	<xml dom node>	<xml dom node>	*Numbered*
child node of <xml dom node>	child nodes	<xml dom node>	<xml dom node>	*Plain*
class of <active device>	classes	<string>	<active device>	*Plain*
client	clients	<client>	<world>	*PlainGlobal*
client folder of <site>	client folders	<folder>	<site>	*Plain*
client license	client licenses	<license>	<world>	*PlainGlobal*
code page of <local user>	code pages	<integer>	<local user>	*Plain*
codepage of <file version block>	codepages	<string>	<file version block>	*Plain*
comment of <local group>	comments	<string>	<local group>	*Plain*
comment of <local user>	comments	<string>	<local user>	*Plain*
comment of <network share>	comments	<string>	<network share>	*Plain*
common name of <license>	common names	<string>	<license>	*Plain*
communications bit <operating system suite mask>	communications bits	<boolean>	<world>	*IndexedGlobal*
communications operator flag of <local user>	communications operator flags	<boolean>	<local user>	*Plain*
competition size of <selected server>	competition sizes	<integer>	<selected server>	*Plain*
competition weight of <selected server>	competition weights	<integer>	<selected server>	*Plain*
complete time of <action>	complete times	<time>	<action>	*Plain*
component <integer> of <distinguished name>	components	<distinguished name component>	<distinguished name>	*Numbered*
component <integer> of <site version list>	components	<integer>	<site version list>	*Numbered*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
component of <distinguished name>	components	<distinguished name component>	<distinguished name>	*Plain*
compressed of <filesystem object>	compresseds	<boolean>	<filesystem object>	*Plain*
computer id	computer ids	<integer>	<world>	*PlainGlobal*
computer name	computer names	<string>	<world>	*PlainGlobal*
computer of <event log record>	computers	<string>	<event log record>	*Plain*
concatenation <html> of <html>	concatenations	<html>	<html>	*Indexed*
concatenation <html> of <string>	concatenations	<html>	<string>	*Indexed*
concatenation <string> of <string>	concatenations	<string>	<string>	*Named*
concatenation of <string>	concatenations	<string>	<string>	*Plain*
conjunction of <boolean>	conjunctions	<boolean>	<boolean>	*Plain*
connection of <network>	connections	<connection>	<network>	*Plain*
connection status <integer>	connection statuses	<connection status>	<world>	*NumberedGlobal*
connection status authenticating	connection statuses authenticating	<connection status>	<world>	*PlainGlobal*
connection status authentication failed	connection statuses authentication failed	<connection status>	<world>	*PlainGlobal*
connection status authentication succeeded	connection statuses authentication succeeded	<connection status>	<world>	*PlainGlobal*
connection status connected	connection statuses connected	<connection status>	<world>	*PlainGlobal*
connection status connecting	connection statuses connecting	<connection status>	<world>	*PlainGlobal*
connection status disconnected	connection statuses disconnected	<connection status>	<world>	*PlainGlobal*
connection status disconnecting	connection statuses disconnecting	<connection status>	<world>	*PlainGlobal*
connection status hardware disabled	connection statuses hardware disabled	<connection status>	<world>	*PlainGlobal*
connection status hardware malfunction	connection statuses hardware malfunction	<connection status>	<world>	*PlainGlobal*
connection status media disconnected	connection statuses media disconnected	<connection status>	<world>	*PlainGlobal*
connection status no hardware present	connection statuses no hardware present	<connection status>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
constrained of <action>	constraineds	<boolean>	<action>	*Plain*
content of <file>	contents	<file content>	<file>	*Plain*
control of <security descriptor>	controls	<integer>	<security descriptor>	*Plain*
controller of <action lock state>	controllers	<string>	<action lock state>	*Plain*
country code of <local user>	country codes	<integer>	<local user>	*Plain*
create file permission of <access control entry>	create file permissions	<boolean>	<access control entry>	*Plain*
create folder permission of <access control entry>	create folder permissions	<boolean>	<access control entry>	*Plain*
create link permission of <access control entry>	create link permissions	<boolean>	<access control entry>	*Plain*
create permission of <network share>	create permissions	<boolean>	<network share>	*Plain*
create subkey permission of <access control entry>	create subkey permissions	<boolean>	<access control entry>	*Plain*
creation time of <filesystem object>	creation times	<time>	<filesystem object>	*Plain*
csd version of <operating system>	csd versions	<string>	<operating system>	*Plain*
csidl folder <integer>	csidl folders	<folder>	<world>	*NumberedGlobal*
current date	current dates	<date>	<world>	*PlainGlobal*
current day_of_month	current days_of_month	<day of month>	<world>	*PlainGlobal*
current day_of_week	current days_of_week	<day of week>	<world>	*PlainGlobal*
current day_of_year	current days_of_year	<day of year>	<world>	*PlainGlobal*
current firewall profile type	current firewall profile types	<firewall profile type>	<world>	*PlainGlobal*
current month	current months	<month>	<world>	*PlainGlobal*
current month_and_year	current months_and_years	<month and year>	<world>	*PlainGlobal*
current profile of <firewall policy>	current profiles	<firewall profile>	<firewall policy>	*Plain*
current profile type of <firewall>	current profile types	<firewall profile type>	<firewall>	*Plain*
current relay	current relays	<current relay>	<world>	*PlainGlobal*
current site	current sites	<site>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
current time_of_day	current times_of_day	<time of day with time zone>	<world>	*PlainGlobal*
current time_of_day <time zone>	current times_of_day	<time of day with time zone>	<world>	*IndexedGlobal*
current user	current users	<current user>	<world>	*PlainGlobal*
current user	current users	<logged on user>	<world>	*PlainGlobal*
current user key <logged on user> of <registry>	current user keys	<registry key>	<registry>	*Indexed*
current year	current years	<year>	<world>	*PlainGlobal*
currently active of <firewall rule>	currently actives	<boolean>	<firewall rule>	*Plain*
custom firewall scope	custom firewall scopes	<firewall scope>	<world>	*PlainGlobal*
custom site subscription effective date <string>	custom site subscription effective dates	<time>	<world>	*NamedGlobal*
customized of <firewall service>	customizeds	<boolean>	<firewall service>	*Plain*
dacl of <security descriptor>	dacls	<access control list>	<security descriptor>	*Plain*
datacenter bit <operating system suite mask>	datacenter bits	<boolean>	<world>	*IndexedGlobal*
date <string>	dates	<date>	<world>	*NamedGlobal*
date <time zone> of <time>	dates	<date>	<time>	*Indexed*
date of <bios>	dates	<string>	<bios>	*Plain*
day	days	<time interval>	<world>	*PlainGlobal*
day of <day of year>	days	<day of month>	<day of year>	*Plain*
day_of_month <integer>	days_of_month	<day of month>	<world>	*NumberedGlobal*
day_of_month <string>	days_of_month	<day of month>	<world>	*NamedGlobal*
day_of_month of <date>	days_of_month	<day of month>	<date>	*Plain*
day_of_week <string>	days_of_week	<day of week>	<world>	*NamedGlobal*
day_of_week of <date>	days_of_week	<day of week>	<date>	*Plain*
day_of_year of <date>	days_of_year	<day of year>	<date>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
december	decembers	<month>	<world>	*PlainGlobal*
december <integer>	decembers	<day of year>	<world>	*NumberedGlobal*
december <integer> of <integer>	decembers	<date>	<integer>	*Numbered*
december of <integer>	decembers	<month and year>	<integer>	*Plain*
default value of <registry key>	default values	<registry key value>	<registry key>	*Plain*
default web browser	default web browsers	<application>	<world>	*PlainGlobal*
delete child permission of <access control entry>	delete child permissions	<boolean>	<access control entry>	*Plain*
delete permission of <access control entry>	delete permissions	<boolean>	<access control entry>	*Plain*
delete permission of <network share>	delete permissions	<boolean>	<network share>	*Plain*
descendant folder of <folder>	descendant folders	<folder>	<folder>	*Plain*
descendant of <folder>	descendants	<file>	<folder>	*Plain*
description of <active device>	descriptions	<string>	<active device>	*Plain*
description of <event log record>	descriptions	<string>	<event log record>	*Plain*
description of <firewall rule>	descriptions	<string>	<firewall rule>	*Plain*
description of <network adapter>	descriptions	<string>	<network adapter>	*Plain*
device key <string> of <registry>	device keys	<registry key>	<registry>	*Named*
device key of <registry>	device keys	<registry key>	<registry>	*Plain*
device name of <connection>	device names	<string>	<connection>	*Plain*
dhcp enabled of <network adapter>	dhcp enableds	<boolean>	<network adapter>	*Plain*
dhcp server of <network adapter>	dhcp servers	<ipv4 address>	<network adapter>	*Plain*
direct object type of <property>	direct object types	<type>	<property>	*Plain*
disjunction of <boolean>	disjunctions	<boolean>	<boolean>	*Plain*
display name of <service>	display names	<string>	<service>	*Plain*
distance of <selected server>	distances	<integer range>	<selected server>	*Plain*
distinguished name <string>	distinguished names	<distinguished name>	<world>	*NamedGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
distinguished name error message of <active directory local computer>	distinguished name error messages	<string>	<active directory local computer>	*Plain*
distinguished name of <active directory local computer>	distinguished names	<string>	<active directory local computer>	*Plain*
divided by zero of <floating point>	divided by zeroes	<boolean>	<floating point>	*Plain*
dmi	dmis	<dmi>	<world>	*PlainGlobal*
dns name	dns names	<string>	<world>	*PlainGlobal*
dns server of <network adapter>	dns servers	<network address list>	<network adapter>	*Plain*
dns server of <network>	dns servers	<network address list>	<network>	*Plain*
dns suffix of <network adapter>	dns suffixes	<string>	<network adapter>	*Plain*
domain firewall profile type	domain firewall profile types	<firewall profile type>	<world>	*PlainGlobal*
domain name of <security identifier>	domain names	<string>	<security identifier>	*Plain*
domain profile of <firewall policy>	domain profiles	<firewall profile>	<firewall policy>	*Plain*
domain user	domain users	<local user>	<world>	*PlainGlobal*
domain user <string>	domain users	<local user>	<world>	*NamedGlobal*
drive	drives	<drive>	<world>	*PlainGlobal*
drive <string>	drives	<drive>	<world>	*NamedGlobal*
drive of <filesystem object>	drives	<drive>	<filesystem object>	*Plain*
driver key of <active device>	driver keys	<registry key>	<active device>	*Plain*
driver key of <registry key>	driver keys	<registry key>	<registry key>	*Plain*
driver key value name of <active device>	driver key value names	<string>	<active device>	*Plain*
edge traversal allowed of <firewall rule>	edge traversal alloweds	<boolean>	<firewall rule>	*Plain*
effective access mode for <string> of <access control list>	effective access modes for	<integer>	<access control list>	*Named*
effective access system security permission for <string> of <access control list>	effective access system security permissions for	<boolean>	<access control list>	*Named*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
effective append permission for <string> of <access control list>	effective append permissions for	<boolean>	<access control list>	*Named*
effective change notification permission for <string> of <access control list>	effective change notification permissions for	<boolean>	<access control list>	*Named*
effective create file permission for <string> of <access control list>	effective create file permissions for	<boolean>	<access control list>	*Named*
effective create folder permission for <string> of <access control list>	effective create folder permissions for	<boolean>	<access control list>	*Named*
effective create link permission for <string> of <access control list>	effective create link permissions for	<boolean>	<access control list>	*Named*
effective create subkey permission for <string> of <access control list>	effective create subkey permissions for	<boolean>	<access control list>	*Named*
effective date of <action lock state>	effective dates	<time>	<action lock state>	*Plain*
effective date of <setting>	effective dates	<time>	<setting>	*Plain*
effective delete child permission for <string> of <access control list>	effective delete child permissions for	<boolean>	<access control list>	*Named*
effective delete permission for <string> of <access control list>	effective delete permissions for	<boolean>	<access control list>	*Named*
effective enumerate subkeys permission for <string> of <access control list>	effective enumerate subkeys permissions for	<boolean>	<access control list>	*Named*
effective execute permission for <string> of <access control list>	effective execute permissions for	<boolean>	<access control list>	*Named*
effective generic all permission for <string> of <access control list>	effective generic all permissions for	<boolean>	<access control list>	*Named*
effective generic execute permission for <string> of <access control list>	effective generic execute permissions for	<boolean>	<access control list>	*Named*
effective generic read permission for <string> of <access control list>	effective generic read permissions for	<boolean>	<access control list>	*Named*
effective generic write permission for <string> of <access control list>	effective generic write permissions for	<boolean>	<access control list>	*Named*
effective list permission for <string> of <access control list>	effective list permissions for	<boolean>	<access control list>	*Named*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
effective maximum allowed permission for <string> of <access control list>	effective maximum allowed permissions for	<boolean>	<access control list>	*Named*
effective query value permission for <string> of <access control list>	effective query value permissions for	<boolean>	<access control list>	*Named*
effective read attributes permission for <string> of <access control list>	effective read attributes permissions for	<boolean>	<access control list>	*Named*
effective read control permission for <string> of <access control list>	effective read control permissions for	<boolean>	<access control list>	*Named*
effective read extended attributes permission for <string> of <access control list>	effective read extended attributes permissions for	<boolean>	<access control list>	*Named*
effective read permission for <string> of <access control list>	effective read permissions for	<boolean>	<access control list>	*Named*
effective set value permission for <string> of <access control list>	effective set value permissions for	<boolean>	<access control list>	*Named*
effective synchronize permission for <string> of <access control list>	effective synchronize permissions for	<boolean>	<access control list>	*Named*
effective traverse permission for <string> of <access control list>	effective traverse permissions for	<boolean>	<access control list>	*Named*
effective write attributes permission for <string> of <access control list>	effective write attributes permissions for	<boolean>	<access control list>	*Named*
effective write dac permission for <string> of <access control list>	effective write dac permissions for	<boolean>	<access control list>	*Named*
effective write extended attributes permission for <string> of <access control list>	effective write extended attributes permissions for	<boolean>	<access control list>	*Named*
effective write owner permission for <string> of <access control list>	effective write owner permissions for	<boolean>	<access control list>	*Named*
effective write permission for <string> of <access control list>	effective write permissions for	<boolean>	<access control list>	*Named*
element of <integer set>	elements	<integer>	<integer set>	*Plain*
element of <string set>	elements	<string>	<string set>	*Plain*
email address of <license>	email addresses	<string>	<license>	*Plain*
embedded nt bit <operating system suite mask>	embedded nt bits	<boolean>	<world>	*IndexedGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
embedded restricted bit <operating system suite mask>	embedded restricted bits	<boolean>	<world>	*IndexedGlobal*
enabled of <firewall authorized application>	enableds	<boolean>	<firewall authorized application>	*Plain*
enabled of <firewall open port>	enableds	<boolean>	<firewall open port>	*Plain*
enabled of <firewall rule>	enableds	<boolean>	<firewall rule>	*Plain*
enabled of <firewall service>	enableds	<boolean>	<firewall service>	*Plain*
enabled of <internet connection firewall>	enableds	<boolean>	<internet connection firewall>	*Plain*
enabled of <port mapping>	enableds	<boolean>	<port mapping>	*Plain*
end of <substring>	ends	<string position>	<substring>	*Plain*
end of <time range>	ends	<time>	<time range>	*Plain*
enterprise bit <operating system suite mask>	enterprise bits	<boolean>	<world>	*IndexedGlobal*
entry of <access control list>	entries	<access control entry>	<access control list>	*Plain*
enumerate subkeys permission of <access control entry>	enumerate subkeys permissions	<boolean>	<access control entry>	*Plain*
environment	environments	<environment>	<world>	*PlainGlobal*
error <string>	errors	<undefined>	<world>	*NamedGlobal*
error event log event type	error event log event types	<event log event type>	<world>	*PlainGlobal*
escape of <string>	escapes	<string>	<string>	*Plain*
evaluation of <license>	evaluations	<boolean>	<license>	*Plain*
event id of <event log record>	event ids	<integer>	<event log record>	*Plain*
event log <string>	event logs	<event log>	<world>	*NamedGlobal*
event log event type <integer>	event log event types	<event log event type>	<world>	*NumberedGlobal*
event type of <event log record>	event types	<event log event type>	<event log record>	*Plain*
exceptions allowed of <firewall profile>	exceptions alloweds	<boolean>	<firewall profile>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
excluded interface of <firewall profile>	excluded interfaces	<string>	<firewall profile>	*Plain*
executable file format of <file>	executable file formats	<string>	<file>	*Plain*
execute permission of <access control entry>	execute permissions	<boolean>	<access control entry>	*Plain*
execute permission of <network share>	execute permissions	<boolean>	<network share>	*Plain*
expand environment string of <string>	expand environment strings	<string>	<string>	*Plain*
expiration date of <action lock state>	expiration dates	<time>	<action lock state>	*Plain*
expiration date of <license>	expiration dates	<time>	<license>	*Plain*
expiration state of <license>	expiration states	<string>	<license>	*Plain*
extended family of <processor>	extended families	<integer>	<processor>	*Plain*
extended model of <processor>	extended models	<integer>	<processor>	*Plain*
external port of <port mapping>	external ports	<integer>	<port mapping>	*Plain*
false	falses	<boolean>	<world>	*PlainGlobal*
family name of <processor>	family names	<string>	<processor>	*Plain*
family of <network interface>	families	<integer>	<network interface>	*Plain*
family of <processor>	families	<integer>	<processor>	*Plain*
feature mask of <processor>	feature masks	<integer>	<processor>	*Plain*
february	februarys	<month>	<world>	*PlainGlobal*
february <integer>	februarys	<day of year>	<world>	*NumberedGlobal*
february <integer> of <integer>	februarys	<date>	<integer>	*Numbered*
february of <integer>	februarys	<month and year>	<integer>	*Plain*
file <string>	files	<file>	<world>	*NamedGlobal*
file <string> of <folder>	files	<file>	<folder>	*Named*
file extension <string> of <registry>	file extensions	<registry key>	<registry>	*Named*
file of <folder>	files	<file>	<folder>	*Plain*
file of <service>	files	<file>	<service>	*Plain*
file system type of <drive>	file system types	<string>	<drive>	*Plain*
file type <string> of <registry>	file types	<registry key>	<registry>	*Named*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
file version of <file>	file versions	<version>	<file>	*Plain*
file_and_print firewall service type	file_and_print firewall service types	<firewall service type>	<world>	*PlainGlobal*
file_supports_encryption of <drive>	file_supports_encryptions	<boolean>	<drive>	*Plain*
file_supports_object_ids of <drive>	file_supports_object_idss	<boolean>	<drive>	*Plain*
file_supports_reparse_points of <drive>	file_supports_reparse_pointss	<boolean>	<drive>	*Plain*
file_supports_sparse_files of <drive>	file_supports_sparse_filess	<boolean>	<drive>	*Plain*
file_volume_quotas of <drive>	file_volume_quotass	<boolean>	<drive>	*Plain*
final part <time interval> of <time range>	final parts	<time range>	<time range>	*Indexed*
find file <string> of <folder>	find files	<file>	<folder>	*Named*
finite of <floating point>	finites	<boolean>	<floating point>	*Plain*
firewall	firewalls	<firewall>	<world>	*PlainGlobal*
firewall action <integer>	firewall actions	<firewall action>	<world>	*NumberedGlobal*
firewall enabled of <firewall profile>	firewalls enabled	<boolean>	<firewall profile>	*Plain*
firewall local policy modify state <integer>	firewall local policy modify states	<firewall local policy modify state>	<world>	*NumberedGlobal*
firewall of <connection>	firewalls	<internet connection firewall>	<connection>	*Plain*
firewall profile type <integer>	firewall profile types	<firewall profile type>	<world>	*NumberedGlobal*
firewall scope <integer>	firewall scopes	<firewall scope>	<world>	*NumberedGlobal*
firewall service type <integer>	firewall service types	<firewall service type>	<world>	*NumberedGlobal*
first <day of week> of <month and year>	firsts	<date>	<month and year>	*Indexed*
first <integer> of <string>	firsts	<substring>	<string>	*Numbered*
first <string> of <string>	firsts	<substring>	<string>	*Named*
first child of <xml dom node>	first children	<xml dom node>	<xml dom node>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
first friday of <month and year>	first fridays	<date>	<month and year>	*Plain*
first match <regular expression> of <string>	first matches	<regular expression match>	<string>	*Indexed*
first monday of <month and year>	first mondays	<date>	<month and year>	*Plain*
first raw version block of <file>	first raw version blocks	<file version block>	<file>	*Plain*
first saturday of <month and year>	first saturdays	<date>	<month and year>	*Plain*
first start time of <application usage summary>	first start times	<time>	<application usage summary>	*Plain*
first sunday of <month and year>	first sundays	<date>	<month and year>	*Plain*
first thursday of <month and year>	first thursdays	<date>	<month and year>	*Plain*
first tuesday of <month and year>	first tuesdays	<date>	<month and year>	*Plain*
first wednesday of <month and year>	first wednesdays	<date>	<month and year>	*Plain*
fixlet of <site>	fixlets	<fixlet>	<site>	*Plain*
floating point <string>	floating points	<floating point>	<world>	*NamedGlobal*
folder <string>	folders	<folder>	<world>	*NamedGlobal*
folder <string> of <drive>	folders	<folder>	<drive>	*Named*
folder <string> of <folder>	folders	<folder>	<folder>	*Named*
folder of <folder>	folders	<folder>	<folder>	*Plain*
following text of <string position>	following texts	<substring>	<string position>	*Plain*
following text of <substring>	following texts	<substring>	<substring>	*Plain*
free space of <drive>	free spaces	<integer>	<drive>	*Plain*
friday	fridays	<day of week>	<world>	*PlainGlobal*
friendly name of <active device>	friendly names	<string>	<active device>	*Plain*
friendly name of <network adapter>	friendly names	<string>	<network adapter>	*Plain*
fs_case_is_preserved of <drive>	fs_case_is_preserveds	<boolean>	<drive>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
fs_case_sensitive of <drive>	fs_case_sensitives	<boolean>	<drive>	*Plain*
fs_file_compression of <drive>	fs_file_compressions	<boolean>	<drive>	*Plain*
fs_persistent_acls of <drive>	fs_persistent_aclss	<boolean>	<drive>	*Plain*
fs_unicode_stored_on_disk of <drive>	fs_unicode_stored_on_disks	<boolean>	<drive>	*Plain*
fs_vol_is_compressed of <drive>	fs_vol_is_compresseds	<boolean>	<drive>	*Plain*
full name of <local user>	full names	<string>	<local user>	*Plain*
full wmi <string>	full wmis	<wmi>	<world>	*NamedGlobal*
gateway address <integer> of <selected server>	gateway addresses	<ipv4 address>	<selected server>	*Numbered*
gateway address of <selected server>	gateway addresses	<ipv4 address>	<selected server>	*Plain*
gateway list of <network adapter>	gateway lists	<network address list>	<network adapter>	*Plain*
gateway of <network adapter>	gateways	<ipv4 address>	<network adapter>	*Plain*
gather schedule authority of <site>	gather schedule authoritys	<string>	<site>	*Plain*
gather schedule time interval of <site>	gather schedule time intervals	<time interval>	<site>	*Plain*
gather url of <license>	gather urls	<string>	<license>	*Plain*
generic all permission of <access control entry>	generic all permissions	<boolean>	<access control entry>	*Plain*
generic execute permission of <access control entry>	generic execute permissions	<boolean>	<access control entry>	*Plain*
generic read permission of <access control entry>	generic read permissions	<boolean>	<access control entry>	*Plain*
generic write permission of <access control entry>	generic write permissions	<boolean>	<access control entry>	*Plain*
ghz	ghzs	<hertz>	<world>	*PlainGlobal*
globally open port of <firewall profile>	globally open ports	<firewall open port>	<firewall profile>	*Plain*
globally open port of <firewall service>	globally open ports	<firewall open port>	<firewall service>	*Plain*
gp override firewall local policy modify state	gp override firewall local policy modify states	<firewall local policy modify state>	<world>	*PlainGlobal*
greatest hz	greatest hzs	<hertz>	<world>	*PlainGlobal*
greatest integer	greatest integers	<integer>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
greatest time interval	greatest time intervals	<time interval>	<world>	*PlainGlobal*
group <integer> of <site>	groups	<site group>	<site>	*Numbered*
group leader of <action>	group leaders	<boolean>	<action>	*Plain*
group of <security descriptor>	groups	<security identifier>	<security descriptor>	*Plain*
grouping of <firewall rule>	groupings	<string>	<firewall rule>	*Plain*
guest privilege of <local user>	guest privileges	<boolean>	<local user>	*Plain*
guid of <connection>	guids	<string>	<connection>	*Plain*
hardware id of <active device>	hardware ids	<string>	<active device>	*Plain*
has blank sa password of <local mssql database>	has blank sa passwords	<boolean>	<local mssql database>	*Plain*
header <string> of <fixlet>	headers	<fixlet_header>	<fixlet>	*Named*
header of <fixlet>	headers	<fixlet_header>	<fixlet>	*Plain*
hexadecet <integer> of <ipv6 address>	hexadecets	<integer>	<ipv6 address>	*Numbered*
hexadecimal integer <string>	hexadecimal integers	<integer>	<world>	*NamedGlobal*
hexadecimal string <string>	hexadecimal strings	<string>	<world>	*NamedGlobal*
hidden of <filesystem object>	hiddens	<boolean>	<filesystem object>	*Plain*
home directory drive of <local user>	home directory drives	<string>	<local user>	*Plain*
home directory of <local user>	home directories	<string>	<local user>	*Plain*
home directory required flag of <local user>	home directory required flags	<boolean>	<local user>	*Plain*
host name of <root server>	host names	<string>	<root server>	*Plain*
hostname	hostnames	<string>	<world>	*PlainGlobal*
hour	hours	<time interval>	<world>	*PlainGlobal*
hour_of_day of <time of day with time zone>	hours_of_day	<integer>	<time of day with time zone>	*Plain*
hour_of_day of <time of day>	hours_of_day	<integer>	<time of day>	*Plain*
hyperthreading capable	hyperthreading capables	<boolean>	<world>	*PlainGlobal*
hyperthreading enabled	hyperthreading enableds	<boolean>	<world>	*PlainGlobal*
hz	hzs	<hertz>	<world>	*PlainGlobal*
ia64 of <operating system>	ia64s	<boolean>	<operating system>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
icmp settings of <firewall profile>	icmp settingses	<firewall icmp settings>	<firewall profile>	*Plain*
icmp types_and_codes string of <firewall rule>	icmp types_and_codes strings	<string>	<firewall rule>	*Plain*
icon index of <file shortcut>	icon indexes	<integer>	<file shortcut>	*Plain*
icon pathname of <file shortcut>	icon pathnames	<string>	<file shortcut>	*Plain*
id of <action>	ids	<integer>	<action>	*Plain*
id of <file version block>	ids	<string>	<file version block>	*Plain*
id of <fixlet>	ids	<integer>	<fixlet>	*Plain*
id of <root server>	ids	<integer>	<root server>	*Plain*
id of <site group>	ids	<integer>	<site group>	*Plain*
identifier of <metabase value>	identifiers	<metabase identifier>	<metabase value>	*Plain*
image path of <service>	image paths	<string>	<service>	*Plain*
inbound blocked firewall local policy modify state	inbound blocked firewall local policy modify states	<firewall local policy modify state>	<world>	*PlainGlobal*
inbound connections allowed of <firewall profile>	inbound connections alloweds	<boolean>	<firewall profile>	*Plain*
inbound of <firewall rule>	inbounds	<boolean>	<firewall rule>	*Plain*
index type of <property>	index types	<type>	<property>	*Plain*
inexact of <floating point>	inexacts	<boolean>	<floating point>	*Plain*
infinite of <floating point>	infinites	<boolean>	<floating point>	*Plain*
information event log event type	information event log event types	<event log event type>	<world>	*PlainGlobal*
inherit attribute of <metabase value>	inherit attributes	<boolean>	<metabase value>	*Plain*
inheritance of <access control entry>	inheritances	<integer>	<access control entry>	*Plain*
initial part <time interval> of <time range>	initial parts	<time range>	<time range>	*Indexed*
insert path attribute of <metabase value>	insert path attributes	<boolean>	<metabase value>	*Plain*
install folder <integer>	install folders	<folder>	<world>	*NumberedGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
instance name of <local mssql database>	instance names	<string>	<local mssql database>	*Plain*
integer <integer>	integers	<integer>	<world>	*NumberedGlobal*
integer <string>	integers	<integer>	<world>	*NamedGlobal*
integer ceiling of <floating point>	integer ceilings	<integer>	<floating point>	*Plain*
integer floor of <floating point>	integer floors	<integer>	<floating point>	*Plain*
integer value <integer> of <wmi select>	integer values	<integer>	<wmi select>	*Numbered*
integer value of <wmi select>	integer values	<integer>	<wmi select>	*Plain*
interdomain trust account flag of <local user>	interdomain trust account flags	<boolean>	<local user>	*Plain*
interface <integer> of <network>	interfaces	<network interface>	<network>	*Numbered*
interface of <firewall rule>	interfaces	<string>	<firewall rule>	*Plain*
interface of <network>	interfaces	<network interface>	<network>	*Plain*
interface types string of <firewall rule>	interface types strings	<string>	<firewall rule>	*Plain*
internal port of <port mapping>	internal ports	<integer>	<port mapping>	*Plain*
internet connection firewall of <network adapter>	internet connection firewalls	<internet connection firewall>	<network adapter>	*Plain*
internet protocol <integer>	internet protocols	<internet protocol>	<world>	*NumberedGlobal*
intersection of <integer set>	intersections	<integer set>	<integer set>	*Plain*
intersection of <string set>	intersections	<string set>	<string set>	*Plain*
invalid of <floating point>	invalids	<boolean>	<floating point>	*Plain*
ip address of <selected server>	ip addresses	<ipv4 address>	<selected server>	*Plain*
ip interface <integer> of <network>	ip interfaces	<network ip interface>	<network>	*Numbered*
ip interface of <network>	ip interfaces	<network ip interface>	<network>	*Plain*
ip version <integer>	ip versions	<ip version>	<world>	*NumberedGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
ip version of <firewall authorized application>	ip versions	<ip version>	<firewall authorized application>	*Plain*
ip version of <firewall open port>	ip versions	<ip version>	<firewall open port>	*Plain*
ip version of <firewall service>	ip versions	<ip version>	<firewall service>	*Plain*
ipv4	ipv4s	<ip version>	<world>	*PlainGlobal*
ipv4 address <string>	ipv4 addresses	<ipv4 address>	<world>	*NamedGlobal*
ipv4 part of <ipv6 address>	ipv4 parts	<ipv4 address>	<ipv6 address>	*Plain*
ipv6	ipv6s	<ip version>	<world>	*PlainGlobal*
ipv6 address <string>	ipv6 addresses	<ipv6 address>	<world>	*NamedGlobal*
ipv6 address of <network adapter>	ipv6 addresses	<ipv6 address>	<network adapter>	*Plain*
ipv6 dns server of <network adapter>	ipv6 dns servers	<ipv6 address>	<network adapter>	*Plain*
january	januarys	<month>	<world>	*PlainGlobal*
january <integer>	januarys	<day of year>	<world>	*NumberedGlobal*
january <integer> of <integer>	januarys	<date>	<integer>	*Numbered*
january of <integer>	januarys	<month and year>	<integer>	*Plain*
july	julys	<month>	<world>	*PlainGlobal*
july <integer>	julys	<day of year>	<world>	*NumberedGlobal*
july <integer> of <integer>	julys	<date>	<integer>	*Numbered*
july of <integer>	julys	<month and year>	<integer>	*Plain*
june	junes	<month>	<world>	*PlainGlobal*
june <integer>	junes	<day of year>	<world>	*NumberedGlobal*
june <integer> of <integer>	junes	<date>	<integer>	*Numbered*
june of <integer>	junes	<month and year>	<integer>	*Plain*
key <string> of <file section>	keys	<string>	<file section>	*Named*
key <string> of <file>	keys	<string>	<file>	*Named*
key <string> of <metabase key>	keys	<metabase key>	<metabase key>	*Named*
key <string> of <metabase>	keys	<metabase key>	<metabase>	*Named*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
key <string> of <registry key>	keys	<registry key>	<registry key>	*Named*
key <string> of <registry>	keys	<registry key>	<registry>	*Named*
key of <metabase key>	keys	<metabase key>	<metabase key>	*Plain*
key of <metabase>	keys	<metabase key>	<metabase>	*Plain*
key of <registry key>	keys	<registry key>	<registry key>	*Plain*
khz	khzs	<hertz>	<world>	*PlainGlobal*
language of <file version block>	languages	<string>	<file version block>	*Plain*
last <integer> of <string>	lasts	<substring>	<string>	*Numbered*
last <string> of <string>	lasts	<substring>	<string>	*Named*
last change time of <action>	last change times	<time>	<action>	*Plain*
last child of <xml dom node>	last children	<xml dom node>	<xml dom node>	*Plain*
last gather time of <site>	last gather times	<time>	<site>	*Plain*
last logoff of <local user>	last logoffs	<time>	<local user>	*Plain*
last logon of <local user>	last logons	<time>	<local user>	*Plain*
last start time of <application usage summary>	last start times	<time>	<application usage summary>	*Plain*
last time seen of <application usage summary>	last times seen	<time>	<application usage summary>	*Plain*
leap of <year>	leaps	<boolean>	<year>	*Plain*
lease expires of <network adapter>	leases expire	<time>	<network adapter>	*Plain*
lease obtained of <network adapter>	leases obtained	<time>	<network adapter>	*Plain*
least hz	least hzs	<hertz>	<world>	*PlainGlobal*
least integer	least integers	<integer>	<world>	*PlainGlobal*
least significant one bit of <bit set>	least significant one bits	<integer>	<bit set>	*Plain*
least time interval	least time intervals	<time interval>	<world>	*PlainGlobal*
left operand type of <binary operator>	left operand types	<type>	<binary operator>	*Plain*
left shift <integer> of <bit set>	left shifts	<bit set>	<bit set>	*Numbered*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
length of <event log record>	lengths	<integer>	<event log record>	*Plain*
length of <month and year>	lengths	<time interval>	<month and year>	*Plain*
length of <rope>	lengths	<integer>	<rope>	*Plain*
length of <string>	lengths	<integer>	<string>	*Plain*
length of <time range>	lengths	<time interval>	<time range>	*Plain*
length of <year>	lengths	<time interval>	<year>	*Plain*
less significance <integer> of <floating point>	less significances	<floating point>	<floating point>	*Numbered*
line <integer> of <file>	lines	<file line>	<file>	*Numbered*
line containing <string> of <file>	lines containing	<file line>	<file>	*Named*
line number of <file line>	line numbers	<integer>	<file line>	*Plain*
line of <file>	lines	<file line>	<file>	*Plain*
line starting with <string> of <file>	lines starting with	<file line>	<file>	*Named*
link speed of <network adapter>	link speeds	<integer>	<network adapter>	*Plain*
list permission of <access control entry>	list permissions	<boolean>	<access control entry>	*Plain*
local addresses string of <firewall rule>	local addresses strings	<string>	<firewall rule>	*Plain*
local administrator	local administrators	<boolean>	<world>	*PlainGlobal*
local computer of <active directory server>	local computers	<active directory local computer>	<active directory server>	*Plain*
local group	local groups	<local group>	<world>	*PlainGlobal*
local group <string>	local groups	<local group>	<world>	*NamedGlobal*
local mssql database	local mssql databases	<local mssql database>	<world>	*PlainGlobal*
local mssql database <string>	local mssql databases	<local mssql database>	<world>	*NamedGlobal*
local policy modify state of <firewall>	local policy modify states	<firewall local policy modify state>	<firewall>	*Plain*
local policy of <firewall>	local policies	<firewall policy>	<firewall>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
local ports string of <firewall rule>	local ports strings	<string>	<firewall rule>	*Plain*
local subnet firewall scope	local subnet firewall scopes	<firewall scope>	<world>	*PlainGlobal*
local time <string>	local times	<time>	<world>	*NamedGlobal*
local time zone	local time zones	<time zone>	<world>	*PlainGlobal*
local user	local users	<local user>	<world>	*PlainGlobal*
local user <string>	local users	<local user>	<world>	*NamedGlobal*
location information of <active device>	location informations	<string>	<active device>	*Plain*
location of <filesystem object>	locations	<string>	<filesystem object>	*Plain*
lock string of <action lock state>	lock strings	<string>	<action lock state>	*Plain*
locked of <action lock state>	lockeds	<boolean>	<action lock state>	*Plain*
locked out flag of <local user>	locked out flags	<boolean>	<local user>	*Plain*
logged on user	logged on users	<logged on user>	<world>	*PlainGlobal*
logical processor count	logical processor counts	<integer>	<world>	*PlainGlobal*
login account of <service>	login accounts	<string>	<service>	*Plain*
login mode of <local mssql database>	login modes	<integer>	<local mssql database>	*Plain*
logon count of <local user>	logon counts	<integer>	<local user>	*Plain*
logon script of <local user>	logon scripts	<string>	<local user>	*Plain*
logon server of <local user>	logon servers	<string>	<local user>	*Plain*
loopback of <network ip interface>	loopbacks	<boolean>	<network ip interface>	*Plain*
lower bound of <integer range>	lower bounds	<integer>	<integer range>	*Plain*
mac address of <network adapter>	mac addresses	<string>	<network adapter>	*Plain*
main gather service	main gather services	<service>	<world>	*PlainGlobal*
main processor	main processors	<processor>	<world>	*PlainGlobal*
major version of <operating system>	major versions	<integer>	<operating system>	*Plain*
manufacturer of <active device>	manufacturers	<string>	<active device>	*Plain*
march	marchs	<month>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
march <integer>	marchs	<day of year>	<world>	*NumberedGlobal*
march <integer> of <integer>	marchs	<date>	<integer>	*Numbered*
march of <integer>	marchs	<month and year>	<integer>	*Plain*
masthead of <site>	mastheads	<file>	<site>	*Plain*
match <regular expression> of <string>	matches	<regular expression match>	<string>	*Indexed*
maximum allowed permission of <access control entry>	maximum allowed permissions	<boolean>	<access control entry>	*Plain*
maximum of <integer>	maxima	<integer>	<integer>	*Plain*
maximum of <time interval>	maxima	<time interval>	<time interval>	*Plain*
maximum of <time>	maxima	<time>	<time>	*Plain*
maximum seat count of <license>	maximum seat counts	<integer>	<license>	*Plain*
maximum storage of <local user>	maximum storages	<integer>	<local user>	*Plain*
maximum transmission unit of <network adapter>	maximum transmission units	<integer>	<network adapter>	*Plain*
may	mays	<month>	<world>	*PlainGlobal*
may <integer>	mays	<day of year>	<world>	*NumberedGlobal*
may <integer> of <integer>	mays	<date>	<integer>	*Numbered*
may of <integer>	mays	<month and year>	<integer>	*Plain*
mean of <floating point>	means	<floating point>	<floating point>	*Plain*
mean of <integer>	means	<floating point>	<integer>	*Plain*
media type <integer>	media types	<media type>	<world>	*NumberedGlobal*
media type bridge	media types bridge	<media type>	<world>	*PlainGlobal*
media type direct	media types direct	<media type>	<world>	*PlainGlobal*
media type isdn	media types isdn	<media type>	<world>	*PlainGlobal*
media type lan	media types lans	<media type>	<world>	*PlainGlobal*
media type of <connection>	media types	<media type>	<connection>	*Plain*
media type phone	media types phone	<media type>	<world>	*PlainGlobal*
media type pppoe	media types pppoe	<media type>	<world>	*PlainGlobal*
media type shared access host lan	media types shared access host lan	<media type>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
media type shared access host ras	media types shared access host ras	<media type>	<world>	*PlainGlobal*
media type tunnel	media types tunnel	<media type>	<world>	*PlainGlobal*
member of <local group>	members	<local group member>	<local group>	*Plain*
member of <site group>	members	<boolean>	<site group>	*Plain*
metabase	metabases	<metabase>	<world>	*PlainGlobal*
metric <integer> of <operating system>	metrics	<integer>	<operating system>	*Numbered*
mhz	mhzs	<hertz>	<world>	*PlainGlobal*
microsecond	microseconds	<time interval>	<world>	*PlainGlobal*
midnight	midnights	<time of day>	<world>	*PlainGlobal*
millisecond	milliseconds	<time interval>	<world>	*PlainGlobal*
minimum of <integer>	minima	<integer>	<integer>	*Plain*
minimum of <time interval>	minima	<time interval>	<time interval>	*Plain*
minimum of <time>	minima	<time>	<time>	*Plain*
minor version of <operating system>	minor versions	<integer>	<operating system>	*Plain*
minute	minutes	<time interval>	<world>	*PlainGlobal*
minute_of_hour of <time of day with time zone>	minutes_of_hour	<integer>	<time of day with time zone>	*Plain*
minute_of_hour of <time of day>	minutes_of_hour	<integer>	<time of day>	*Plain*
model of <processor>	models	<integer>	<processor>	*Plain*
modification time of <filesystem object>	modification times	<time>	<filesystem object>	*Plain*
module <string>	modules	<module>	<world>	*NamedGlobal*
monday	mondays	<day of week>	<world>	*PlainGlobal*
month	months	<number of months>	<world>	*PlainGlobal*
month <integer>	months	<month>	<world>	*NumberedGlobal*
month <string>	months	<month>	<world>	*NamedGlobal*
month of <date>	months	<month>	<date>	*Plain*
month of <day of year>	months	<month>	<day of year>	*Plain*
month of <month and year>	months	<month>	<month and year>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
month_and_year of <date>	months_and_years	<month and year>	<date>	*Plain*
more significance <integer> of <floating point>	more significances	<floating point>	<floating point>	*Numbered*
most significant one bit of <bit set>	most significant one bits	<integer>	<bit set>	*Plain*
multicast support of <network ip interface>	multicast supports	<boolean>	<network ip interface>	*Plain*
multiplicity of <integer with multiplicity>	multiplicities	<integer>	<integer with multiplicity>	*Plain*
multiplicity of <string with multiplicity>	multiplicities	<integer>	<string with multiplicity>	*Plain*
multivalued of <property>	multivalueds	<boolean>	<property>	*Plain*
name of <application usage summary>	names	<string>	<application usage summary>	*Plain*
name of <binary operator>	names	<string>	<binary operator>	*Plain*
name of <cast>	names	<string>	<cast>	*Plain*
name of <connection>	names	<string>	<connection>	*Plain*
name of <current user>	names	<string>	<current user>	*Plain*
name of <drive>	names	<string>	<drive>	*Plain*
name of <environment variable>	names	<string>	<environment variable>	*Plain*
name of <filesystem object>	names	<string>	<filesystem object>	*Plain*
name of <firewall authorized application>	names	<string>	<firewall authorized application>	*Plain*
name of <firewall open port>	names	<string>	<firewall open port>	*Plain*
name of <firewall rule>	names	<string>	<firewall rule>	*Plain*
name of <firewall service>	names	<string>	<firewall service>	*Plain*
name of <fixlet_header>	names	<string>	<fixlet_header>	*Plain*
name of <local group>	names	<string>	<local group>	*Plain*
name of <local user>	names	<string>	<local user>	*Plain*
name of <logged on user>	names	<string>	<logged on user>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
name of <metabase key>	names	<string>	<metabase key>	*Plain*
name of <network adapter>	names	<string>	<network adapter>	*Plain*
name of <network share>	names	<string>	<network share>	*Plain*
name of <operating system>	names	<string>	<operating system>	*Plain*
name of <port mapping>	names	<string>	<port mapping>	*Plain*
name of <registry key value>	names	<string>	<registry key value>	*Plain*
name of <registry key>	names	<string>	<registry key>	*Plain*
name of <selected server>	names	<string>	<selected server>	*Plain*
name of <setting>	names	<string>	<setting>	*Plain*
name of <site>	names	<string>	<site>	*Plain*
name of <type>	names	<string>	<type>	*Plain*
name of <unary operator>	names	<string>	<unary operator>	*Plain*
name of <wmi select>	names	<string>	<wmi select>	*Plain*
nan of <floating point>	nans	<boolean>	<floating point>	*Plain*
native registry	native registries	<registry>	<world>	*PlainGlobal*
network	networks	<network>	<world>	*PlainGlobal*
network share	network shares	<network share>	<world>	*PlainGlobal*
network share <string>	network shares	<network share>	<world>	*NamedGlobal*
next line of <file line>	next lines	<file line>	<file line>	*Plain*
next sibling of <xml dom node>	next siblings	<xml dom node>	<xml dom node>	*Plain*
no password required flag of <local user>	no password required flags	<boolean>	<local user>	*Plain*
node name of <xml dom node>	node names	<string>	<xml dom node>	*Plain*
node type of <xml dom node>	node types	<integer>	<xml dom node>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
node value of <xml dom node>	node values	<string>	<xml dom node>	*Plain*
none firewall service type	none firewall service types	<firewall service type>	<world>	*PlainGlobal*
noon	noons	<time of day>	<world>	*PlainGlobal*
normal account flag of <local user>	normal account flags	<boolean>	<local user>	*Plain*
normal of <filesystem object>	normals	<boolean>	<filesystem object>	*Plain*
normal of <floating point>	normals	<boolean>	<floating point>	*Plain*
notifications disabled of <firewall profile>	notifications disableds	<boolean>	<firewall profile>	*Plain*
november	novembers	<month>	<world>	*PlainGlobal*
november <integer>	novembers	<day of year>	<world>	*NumberedGlobal*
november <integer> of <integer>	novembers	<date>	<integer>	*Numbered*
november of <integer>	novembers	<month and year>	<integer>	*Plain*
now	nows	<time>	<world>	*PlainGlobal*
nt domain controller product type	nt domain controller product types	<operating system product type>	<world>	*PlainGlobal*
nt server product type	nt server product types	<operating system product type>	<world>	*PlainGlobal*
nt workstation product type	nt workstation product types	<operating system product type>	<world>	*PlainGlobal*
numeric type of <drive>	numeric types	<integer>	<drive>	*Plain*
numeric value of <string>	numeric values	<integer>	<string>	*Plain*
october	octobers	<month>	<world>	*PlainGlobal*
october <integer>	octobers	<day of year>	<world>	*NumberedGlobal*
october <integer> of <integer>	octobers	<date>	<integer>	*Numbered*
october of <integer>	octobers	<month and year>	<integer>	*Plain*
oem code page	oem code pages	<integer>	<world>	*PlainGlobal*
offer accepted of <action>	offer accepteds	<boolean>	<action>	*Plain*
offer of <action>	offers	<boolean>	<action>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
offline of <filesystem object>	offlines	<boolean>	<filesystem object>	*Plain*
ok firewall local policy modify state	ok firewall local policy modify states	<firewall local policy modify state>	<world>	*PlainGlobal*
oldest record number of <event log>	oldest record numbers	<integer>	<event log>	*Plain*
one bit of <bit set>	one bits	<integer>	<bit set>	*Plain*
only raw version block of <file>	only raw version blocks	<file version block>	<file>	*Plain*
only version block of <file>	only version blocks	<file version block>	<file>	*Plain*
operand type of <cast>	operand types	<type>	<cast>	*Plain*
operand type of <unary operator>	operand types	<type>	<unary operator>	*Plain*
operating system	operating systems	<operating system>	<world>	*PlainGlobal*
operating system product type <integer>	operating system product types	<operating system product type>	<world>	*NumberedGlobal*
options of <port mapping>	optionss	<integer>	<port mapping>	*Plain*
organization of <license>	organizations	<string>	<license>	*Plain*
origin fixlet id of <action>	origin fixlet ids	<integer>	<action>	*Plain*
outbound connections allowed of <firewall profile>	outbound connections alloweds	<boolean>	<firewall profile>	*Plain*
outbound of <firewall rule>	outbounds	<boolean>	<firewall rule>	*Plain*
overflow of <floating point>	overflows	<boolean>	<floating point>	*Plain*
owner document of <xml dom node>	owner documents	<xml dom document>	<xml dom node>	*Plain*
owner of <security descriptor>	owners	<security identifier>	<security descriptor>	*Plain*
pad of <version>	pads	<version>	<version>	*Plain*
parameter <string>	parameters	<string>	<world>	*NamedGlobal*
parameter <string> of <action>	parameters	<string>	<action>	*Named*
parent folder of <filesystem object>	parent folders	<folder>	<filesystem object>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
parent node of <xml dom node>	parent nodes	<xml dom node>	<xml dom node>	*Plain*
parent of <type>	parents	<type>	<type>	*Plain*
parenthesized part <integer> of <regular expression match>	parenthesized parts	<substring>	<regular expression match>	*Numbered*
parenthesized part of <regular expression match>	parenthesized parts	<substring>	<regular expression match>	*Plain*
password age of <local user>	password ages	<time interval>	<local user>	*Plain*
password change disabled flag of <local user>	password change disabled flags	<boolean>	<local user>	*Plain*
password expiration disabled flag of <local user>	password expiration disabled flags	<boolean>	<local user>	*Plain*
password expired of <local user>	passwords expired	<boolean>	<local user>	*Plain*
password of <network share>	passwords	<string>	<network share>	*Plain*
path of <network share>	paths	<string>	<network share>	*Plain*
pathname of <file shortcut>	pathnames	<string>	<file shortcut>	*Plain*
pathname of <filesystem object>	pathnames	<string>	<filesystem object>	*Plain*
pending login	pending logins	<boolean>	<world>	*PlainGlobal*
pending login of <action>	pending logins	<boolean>	<action>	*Plain*
pending of <action>	pendings	<boolean>	<action>	*Plain*
pending restart	pending restarts	<boolean>	<world>	*PlainGlobal*
pending restart <string>	pending restarts	<boolean>	<world>	*NamedGlobal*
pending restart of <action>	pending restarts	<boolean>	<action>	*Plain*
pending time of <action>	pending times	<time>	<action>	*Plain*
performance counter frequency of <operating system>	performance counter frequencies	<hertz>	<operating system>	*Plain*
performance counter of <operating system>	performance counters	<integer>	<operating system>	*Plain*
permission permission of <network share>	permission permissions	<boolean>	<network share>	*Plain*
personal bit <operating system suite mask>	personal bits	<boolean>	<world>	*IndexedGlobal*
physical processor count	physical processor counts	<integer>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
platform id of <operating system>	platform ids	<integer>	<operating system>	*Plain*
plural name of <property>	plural names	<string>	<property>	*Plain*
point to point of <network ip interface>	point to points	<boolean>	<network ip interface>	*Plain*
port mapping of <internet connection firewall>	port mappings	<port mapping>	<internet connection firewall>	*Plain*
port number of <selected server>	port numbers	<integer>	<selected server>	*Plain*
port of <firewall open port>	ports	<integer>	<firewall open port>	*Plain*
position <integer> of <string>	positions	<string position>	<string>	*Numbered*
position of <string>	positions	<string position>	<string>	*Plain*
preceding text of <string position>	preceding texts	<substring>	<string position>	*Plain*
preceding text of <substring>	preceding texts	<substring>	<substring>	*Plain*
previous line of <file line>	previous lines	<file line>	<file line>	*Plain*
previous sibling of <xml dom node>	previous siblings	<xml dom node>	<xml dom node>	*Plain*
primary group id of <local user>	primary group ids	<integer>	<local user>	*Plain*
primary language of <language>	primary languages	<primary language>	<language>	*Plain*
primary wins server of <network adapter>	primary wins servers	<ipv4 address>	<network adapter>	*Plain*
print operator flag of <local user>	print operator flags	<boolean>	<local user>	*Plain*
priority of <selected server>	priorities	<integer>	<selected server>	*Plain*
private firewall profile type	private firewall profile types	<firewall profile type>	<world>	*PlainGlobal*
private profile of <firewall policy>	private profiles	<firewall profile>	<firewall policy>	*Plain*
problem id of <active device>	problem ids	<integer>	<active device>	*Plain*
process image file name of <firewall authorized application>	process image file names	<string>	<firewall authorized application>	*Plain*
processor	processors	<processor>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
processor <integer>	processors	<processor>	<world>	*NumberedGlobal*
product of <integer>	products	<integer>	<integer>	*Plain*
product type of <operating system>	product types	<operating system product type>	<operating system>	*Plain*
product version of <file>	product versions	<version>	<file>	*Plain*
profile <firewall profile type> of <firewall rule>	profiles	<boolean>	<firewall rule>	*Indexed*
profile folder of <local user>	profile folders	<string>	<local user>	*Plain*
property <string>	properties	<property>	<world>	*NamedGlobal*
property <string> of <type>	properties	<property>	<type>	*Named*
property <string> of <wmi object>	properties	<wmi select>	<wmi object>	*Named*
property of <type>	properties	<property>	<type>	*Plain*
property of <wmi object>	properties	<wmi select>	<wmi object>	*Plain*
property returning <type>	properties returning	<property>	<world>	*IndexedGlobal*
property returning <type> of <type>	properties returning	<property>	<type>	*Indexed*
protocol of <firewall open port>	protocols	<internet protocol>	<firewall open port>	*Plain*
protocol of <firewall rule>	protocols	<internet protocol>	<firewall rule>	*Plain*
protocol of <port mapping>	protocols	<string>	<port mapping>	*Plain*
public firewall profile type	public firewall profile types	<firewall profile type>	<world>	*PlainGlobal*
public profile of <firewall policy>	public profiles	<firewall profile>	<firewall policy>	*Plain*
query value permission of <access control entry>	query value permissions	<boolean>	<access control entry>	*Plain*
ram	rams	<ram>	<world>	*PlainGlobal*
random access memory	random access memories	<ram>	<world>	*PlainGlobal*
range after <time> of <time range>	ranges after	<time range>	<time range>	*Indexed*
range before <time> of <time range>	ranges before	<time range>	<time range>	*Indexed*
raw file version of <file>	raw file versions	<version>	<file>	*Plain*
raw product version of <file>	raw product versions	<version>	<file>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
raw version block <integer> of <file>	raw version blocks	<file version block>	<file>	*Numbered*
raw version block <string> of <file>	raw version blocks	<file version block>	<file>	*Named*
raw version block of <file>	raw version blocks	<file version block>	<file>	*Plain*
raw version of <file>	raw versions	<version>	<file>	*Plain*
read attributes permission of <access control entry>	read attributes permissions	<boolean>	<access control entry>	*Plain*
read control permission of <access control entry>	read control permissions	<boolean>	<access control entry>	*Plain*
read extended attributes permission of <access control entry>	read extended attributes permissions	<boolean>	<access control entry>	*Plain*
read permission of <access control entry>	read permissions	<boolean>	<access control entry>	*Plain*
read permission of <network share>	read permissions	<boolean>	<network share>	*Plain*
readonly of <filesystem object>	readonlys	<boolean>	<filesystem object>	*Plain*
recent application	recent applications	<application>	<world>	*PlainGlobal*
recent application <string>	recent applications	<application>	<world>	*NamedGlobal*
record <integer> of <event log>	records	<event log record>	<event log>	*Numbered*
record count of <event log>	record counts	<integer>	<event log>	*Plain*
record number of <event log record>	record numbers	<integer>	<event log record>	*Plain*
record of <event log>	records	<event log record>	<event log>	*Plain*
reference attribute of <metabase value>	reference attributes	<boolean>	<metabase value>	*Plain*
regapp	regapps	<application>	<world>	*PlainGlobal*
regapp <string>	regapps	<application>	<world>	*NamedGlobal*
regex <string>	regexes	<regular expression>	<world>	*NamedGlobal*
registrar number of <license>	registrar numbers	<integer>	<license>	*Plain*
registry	registries	<registry>	<world>	*PlainGlobal*
regular expression <string>	regular expressions	<regular expression>	<world>	*NamedGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
relative significance place <integer> of <floating point>	relative significance places	<floating point>	<floating point>	*Numbered*
relative significance place of <floating point>	relative significance places	<floating point>	<floating point>	*Plain*
relay service	relay services	<service>	<world>	*PlainGlobal*
release of <operating system>	releases	<string>	<operating system>	*Plain*
relevance of <fixlet>	relevances	<boolean>	<fixlet>	*Plain*
relevant fixlet of <site>	relevant fixlets	<fixlet>	<site>	*Plain*
remote addresses of <firewall authorized application>	remote addresseses	<string>	<firewall authorized application>	*Plain*
remote addresses of <firewall open port>	remote addresseses	<string>	<firewall open port>	*Plain*
remote addresses of <firewall service>	remote addresseses	<string>	<firewall service>	*Plain*
remote addresses string of <firewall rule>	remote addresses strings	<string>	<firewall rule>	*Plain*
remote admin settings of <firewall profile>	remote admin settingses	<firewall remote admin settings>	<firewall profile>	*Plain*
remote desktop firewall service type	remote desktop firewall service types	<firewall service type>	<world>	*PlainGlobal*
remote of <logged on user>	remotes	<boolean>	<logged on user>	*Plain*
remote ports string of <firewall rule>	remote ports strings	<string>	<firewall rule>	*Plain*
result type of <binary operator>	result types	<type>	<binary operator>	*Plain*
result type of <property>	result types	<type>	<property>	*Plain*
result type of <unary operator>	result types	<type>	<unary operator>	*Plain*
right operand type of <binary operator>	right operand types	<type>	<binary operator>	*Plain*
right shift <integer> of <bit set>	right shifts	<bit set>	<bit set>	*Numbered*
root folder of <drive>	root folders	<folder>	<drive>	*Plain*
root server	root servers	<root server>	<world>	*PlainGlobal*
rope <string>	ropes	<rope>	<world>	*NamedGlobal*
rsop computer wmi	rsop computer wmis	<wmi>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
rsop user wmi <security identifier>	rsop user wmis	<wmi>	<world>	*IndexedGlobal*
rule group currently enabled <string> of <firewall>	rule group currently enableds	<boolean>	<firewall>	*Named*
rule group enabled <string> of <firewall profile>	rule group enableds	<boolean>	<firewall profile>	*Named*
rule of <firewall service restriction>	rules	<firewall rule>	<firewall service restriction>	*Plain*
rule of <firewall>	rules	<firewall rule>	<firewall>	*Plain*
running application	running applications	<application>	<world>	*PlainGlobal*
running application <string>	running applications	<application>	<world>	*NamedGlobal*
running of <application usage summary>	runnings	<boolean>	<application usage summary>	*Plain*
running of <local mssql database>	runnings	<boolean>	<local mssql database>	*Plain*
running service	running services	<service>	<world>	*PlainGlobal*
running service <string>	running services	<service>	<world>	*NamedGlobal*
saturday	saturdays	<day of week>	<world>	*PlainGlobal*
scope of <firewall authorized application>	scopes	<firewall scope>	<firewall authorized application>	*Plain*
scope of <firewall open port>	scopes	<firewall scope>	<firewall open port>	*Plain*
scope of <firewall service>	scopes	<firewall scope>	<firewall service>	*Plain*
script flag of <local user>	script flags	<boolean>	<local user>	*Plain*
seat count state of <license>	seat count states	<string>	<license>	*Plain*
seat of <license>	seats	<integer>	<license>	*Plain*
second	seconds	<time interval>	<world>	*PlainGlobal*
second_of_minute of <time of day with time zone>	seconds_of_minute	<integer>	<time of day with time zone>	*Plain*
second_of_minute of <time of day>	seconds_of_minute	<integer>	<time of day>	*Plain*
secondary wins server of <network adapter>	secondary wins servers	<ipv4 address>	<network adapter>	*Plain*
section <string> of <file>	sections	<file section>	<file>	*Named*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
secure attribute of <metabase value>	secure attributes	<boolean>	<metabase value>	*Plain*
security descriptor of <file>	security descriptors	<security descriptor>	<file>	*Plain*
security descriptor of <folder>	security descriptors	<security descriptor>	<folder>	*Plain*
security descriptor of <network share>	security descriptors	<security descriptor>	<network share>	*Plain*
security descriptor of <registry key>	security descriptors	<security descriptor>	<registry key>	*Plain*
security event log	security event logs	<event log>	<world>	*PlainGlobal*
select <string> of <wmi>	selects	<wmi select>	<wmi>	*Named*
select <string> of <xml dom node>	selects	<xml dom node>	<xml dom node>	*Named*
select object <string> of <wmi>	select objects	<wmi object>	<wmi>	*Named*
selected server	selected servers	<selected server>	<world>	*PlainGlobal*
september	septembers	<month>	<world>	*PlainGlobal*
september <integer>	septembers	<day of year>	<world>	*NumberedGlobal*
september <integer> of <integer>	septembers	<date>	<integer>	*Numbered*
september of <integer>	septembers	<month and year>	<integer>	*Plain*
server operator flag of <local user>	server operator flags	<boolean>	<local user>	*Plain*
server trust account flag of <local user>	server trust account flags	<boolean>	<local user>	*Plain*
service	services	<service>	<world>	*PlainGlobal*
service <string>	services	<service>	<world>	*NamedGlobal*
service key value name of <active device>	service key value names	<string>	<active device>	*Plain*
service name of <firewall rule>	service names	<string>	<firewall rule>	*Plain*
service name of <service>	service names	<string>	<service>	*Plain*
service of <firewall profile>	services	<firewall service>	<firewall profile>	*Plain*
service pack major version of <operating system>	service pack major versions	<integer>	<operating system>	*Plain*
service pack minor version of <operating system>	service pack minor versions	<integer>	<operating system>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
service restricted <(string, string)> of <firewall service restriction>	service restricteds	<boolean>	<firewall service restriction>	*Indexed*
service restriction of <firewall>	service restrictions	<firewall service restriction>	<firewall>	*Plain*
service specific exit code of <service>	service specific exit codes	<integer>	<service>	*Plain*
set of <integer>	sets	<integer set>	<integer>	*Plain*
set of <string>	sets	<string set>	<string>	*Plain*
set value permission of <access control entry>	set value permissions	<boolean>	<access control entry>	*Plain*
setting <string> of <client>	settings	<setting>	<client>	*Named*
setting <string> of <site>	settings	<setting>	<site>	*Named*
setting of <client>	settings	<setting>	<client>	*Plain*
setting of <site>	settings	<setting>	<site>	*Plain*
sha1 of <file>	sha1s	<string>	<file>	*Plain*
shortcut of <file>	shortcuts	<file shortcut>	<file>	*Plain*
sid of <local group member>	sids	<security identifier>	<local group member>	*Plain*
sid of <logged on user>	sids	<security identifier>	<logged on user>	*Plain*
significance place <integer> of <floating point>	significance places	<floating point>	<floating point>	*Numbered*
significance place of <floating point>	significance places	<floating point>	<floating point>	*Plain*
significance threshold of <floating point>	significance thresholds	<floating point>	<floating point>	*Plain*
significant digits <integer> of <hertz>	significant digitss	<hertz>	<hertz>	*Numbered*
significant digits <integer> of <integer>	significant digitss	<integer>	<integer>	*Numbered*
single user ts bit <operating system suite mask>	single user ts bits	<boolean>	<world>	*IndexedGlobal*
singular name of <property>	singular names	<string>	<property>	*Plain*
site	sites	<site>	<world>	*PlainGlobal*
site <string>	sites	<site>	<world>	*NamedGlobal*
site number of <license>	site numbers	<integer>	<license>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
site tag of <site>	site tags	<string>	<site>	*Plain*
site version list <string>	site version lists	<site version list>	<world>	*NamedGlobal*
site version list of <site>	site version lists	<site version list>	<site>	*Plain*
size of <file>	sizes	<integer>	<file>	*Plain*
size of <integer set>	sizes	<integer>	<integer set>	*Plain*
size of <ram>	sizes	<integer>	<ram>	*Plain*
size of <registry key value>	sizes	<integer>	<registry key value>	*Plain*
size of <string set>	sizes	<integer>	<string set>	*Plain*
size of <type>	sizes	<integer>	<type>	*Plain*
small business bit <operating system suite mask>	small business bits	<boolean>	<world>	*IndexedGlobal*
small business restricted bit <operating system suite mask>	small business restricted bits	<boolean>	<world>	*IndexedGlobal*
source of <event log record>	sources	<string>	<event log record>	*Plain*
speed of <processor>	speeds	<hertz>	<processor>	*Plain*
standard deviation of <floating point>	standard deviations	<floating point>	<floating point>	*Plain*
standard deviation of <integer>	standard deviations	<floating point>	<integer>	*Plain*
standard firewall profile type	standard firewall profile types	<firewall profile type>	<world>	*PlainGlobal*
standard profile of <firewall policy>	standard profiles	<firewall profile>	<firewall policy>	*Plain*
start date of <license>	start dates	<time>	<license>	*Plain*
start in pathname of <file shortcut>	start in pathnames	<string>	<file shortcut>	*Plain*
start of <substring>	starts	<string position>	<substring>	*Plain*
start of <time range>	starts	<time>	<time range>	*Plain*
start type of <service>	start types	<string>	<service>	*Plain*
state of <service>	states	<string>	<service>	*Plain*
status of <action>	statuss	<string>	<action>	*Plain*
status of <active device>	statuss	<integer>	<active device>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
status of <connection>	statuses	<connection status>	<connection>	*Plain*
status of <network adapter>	statuses	<integer>	<network adapter>	*Plain*
stepping of <processor>	steppings	<integer>	<processor>	*Plain*
string <string>	strings	<string>	<world>	*NamedGlobal*
string value <integer> of <wmi select>	string values	<string>	<wmi select>	*Numbered*
string value of <wmi select>	string values	<string>	<wmi select>	*Plain*
subnet address of <network adapter>	subnet addresses	<ipv4 address>	<network adapter>	*Plain*
subnet address of <network address list>	subnet addresses	<ipv4 address>	<network address list>	*Plain*
subnet address of <network ip interface>	subnet addresses	<ipv4 address>	<network ip interface>	*Plain*
subnet mask of <network adapter>	subnet masks	<ipv4 address>	<network adapter>	*Plain*
subnet mask of <network address list>	subnet masks	<ipv4 address>	<network address list>	*Plain*
subnet mask of <network ip interface>	subnet masks	<ipv4 address>	<network ip interface>	*Plain*
subscribe time of <site>	subscribe times	<time>	<site>	*Plain*
substring <string> of <string>	substrings	<substring>	<string>	*Named*
substring after <string> of <string>	substrings after	<substring>	<string>	*Named*
substring before <string> of <string>	substrings before	<substring>	<string>	*Named*
substring between <string> of <string>	substrings between	<substring>	<string>	*Named*
substring separated by <string> of <string>	substrings separated by	<substring>	<string>	*Named*
suite mask of <operating system>	suite masks	<operating system suite mask>	<operating system>	*Plain*
sum of <integer>	sums	<integer>	<integer>	*Plain*
sunday	sundays	<day of week>	<world>	*PlainGlobal*
symbol of <binary operator>	symbols	<string>	<binary operator>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
symbol of <unary operator>	symbols	<string>	<unary operator>	*Plain*
synchronize permission of <access control entry>	synchronize permissions	<boolean>	<access control entry>	*Plain*
system event log	system event logs	<event log>	<world>	*PlainGlobal*
system file <string>	system files	<file>	<world>	*NamedGlobal*
system ini device file	system ini device files	<file>	<world>	*PlainGlobal*
system ini device file <string>	system ini device files	<file>	<world>	*NamedGlobal*
system language	system languages	<string>	<world>	*PlainGlobal*
system locale	system locales	<language>	<world>	*PlainGlobal*
system of <filesystem object>	systems	<boolean>	<filesystem object>	*Plain*
system ui language	system ui languages	<language>	<world>	*PlainGlobal*
system wow64 folder	system wow64 folders	<folder>	<world>	*PlainGlobal*
system x32 folder	system x32 folders	<folder>	<world>	*PlainGlobal*
system x64 folder	system x64 folders	<folder>	<world>	*PlainGlobal*
target ip address of <port mapping>	target ip addresses	<ipv4 address>	<port mapping>	*Plain*
target name of <port mapping>	target names	<string>	<port mapping>	*Plain*
tcp	tcps	<internet protocol>	<world>	*PlainGlobal*
temporary duplicate account flag of <local user>	temporary duplicate account flags	<boolean>	<local user>	*Plain*
temporary of <filesystem object>	temporarys	<boolean>	<filesystem object>	*Plain*
terminal bit <operating system suite mask>	terminal bits	<boolean>	<world>	*IndexedGlobal*
thursday	thursdays	<day of week>	<world>	*PlainGlobal*
time <string>	times	<time>	<world>	*NamedGlobal*
time <time zone> of <time>	times	<time of day with time zone>	<time>	*Indexed*
time generated of <event log record>	times generated	<time>	<event log record>	*Plain*
time interval <string>	time intervals	<time interval>	<world>	*NamedGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
time of <time of day with time zone>	times	<time of day>	<time of day with time zone>	*Plain*
time value <integer> of <wmi select>	time values	<time>	<wmi select>	*Numbered*
time value of <wmi select>	time values	<time>	<wmi select>	*Plain*
time written of <event log record>	times written	<time>	<event log record>	*Plain*
time zone <string>	time zones	<time zone>	<world>	*NamedGlobal*
time_of_day <string>	times_of_day	<time of day>	<world>	*NamedGlobal*
total duration of <application usage summary>	total durations	<time interval>	<application usage summary>	*Plain*
total processor core count	total processor core counts	<integer>	<world>	*PlainGlobal*
total run count of <application usage summary>	total run counts	<integer>	<application usage summary>	*Plain*
total space of <drive>	total spaces	<integer>	<drive>	*Plain*
traverse permission of <access control entry>	traverse permissions	<boolean>	<access control entry>	*Plain*
true	trues	<boolean>	<world>	*PlainGlobal*
trustee of <access control entry>	trustees	<security identifier>	<access control entry>	*Plain*
trustee type of <access control entry>	trustee types	<integer>	<access control entry>	*Plain*
tuesday	tuesdays	<day of week>	<world>	*PlainGlobal*
two digit hour of <time of day with time zone>	two digit hours	<string>	<time of day with time zone>	*Plain*
two digit hour of <time of day>	two digit hours	<string>	<time of day>	*Plain*
two digit minute of <time of day with time zone>	two digit minutes	<string>	<time of day with time zone>	*Plain*
two digit minute of <time of day>	two digit minutes	<string>	<time of day>	*Plain*
two digit second of <time of day with time zone>	two digit seconds	<string>	<time of day with time zone>	*Plain*
two digit second of <time of day>	two digit seconds	<string>	<time of day>	*Plain*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
type of <distinguished name component>	types	<string>	<distinguished name component>	*Plain*
type of <drive>	types	<string>	<drive>	*Plain*
type of <firewall profile>	types	<firewall profile type>	<firewall profile>	*Plain*
type of <firewall service>	types	<firewall service type>	<firewall service>	*Plain*
type of <metabase value>	types	<metabase type>	<metabase value>	*Plain*
type of <network adapter>	types	<integer>	<network adapter>	*Plain*
type of <network share>	types	<integer>	<network share>	*Plain*
type of <processor>	types	<integer>	<processor>	*Plain*
type of <registry key value>	types	<registry key value type>	<registry key value>	*Plain*
type of <site>	types	<string>	<site>	*Plain*
type of <wmi select>	types	<integer>	<wmi select>	*Plain*
udp	udps	<internet protocol>	<world>	*PlainGlobal*
unary operator <string>	unary operators	<unary operator>	<world>	*NamedGlobal*
unary operator returning <type>	unary operators returning	<unary operator>	<world>	*IndexedGlobal*
underflow of <floating point>	underflows	<boolean>	<floating point>	*Plain*
unicast responses to multicast broadcast disabled of <firewall profile>	unicast responses to multicast broadcast disableds	<boolean>	<firewall profile>	*Plain*
union of <integer set>	unions	<integer set>	<integer set>	*Plain*
union of <string set>	unions	<string set>	<string set>	*Plain*
unique value of <integer>	unique values	<integer with multiplicity>	<integer>	*Plain*
unique value of <string>	unique values	<string with multiplicity>	<string>	*Plain*
universal time <string>	universal times	<time>	<world>	*NamedGlobal*
universal time zone	universal time zones	<time zone>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
upnp firewall service type	upnp firewall service types	<firewall service type>	<world>	*PlainGlobal*
upper bound of <integer range>	upper bounds	<integer>	<integer range>	*Plain*
uptime of <operating system>	uptimes	<time interval>	<operating system>	*Plain*
url of <site>	urls	<string>	<site>	*Plain*
use count of <network share>	use counts	<integer>	<network share>	*Plain*
use limit of <network share>	use limits	<integer>	<network share>	*Plain*
user comment of <local user>	user comments	<string>	<local user>	*Plain*
user id of <local user>	user ids	<integer>	<local user>	*Plain*
user language	user languages	<string>	<world>	*PlainGlobal*
user locale	user locales	<language>	<world>	*PlainGlobal*
user privilege of <local user>	user privileges	<boolean>	<local user>	*Plain*
user sid of <event log record>	user sids	<security identifier>	<event log record>	*Plain*
user type of <metabase value>	user types	<metabase user type>	<metabase value>	*Plain*
user ui language	user ui languages	<language>	<world>	*PlainGlobal*
usual name of <property>	usual names	<string>	<property>	*Plain*
value <string> of <file version block>	values	<string>	<file version block>	*Named*
value <string> of <registry key>	values	<registry key value>	<registry key>	*Named*
value of <distinguished name component>	values	<string>	<distinguished name component>	*Plain*
value of <environment variable>	values	<string>	<environment variable>	*Plain*
value of <fixlet_header>	values	<string>	<fixlet_header>	*Plain*
value of <metabase key>	values	<metabase value>	<metabase key>	*Plain*
value of <registry key>	values	<registry key value>	<registry key>	*Plain*
value of <setting>	values	<string>	<setting>	*Plain*
variable <string> of <environment>	variables	<environment variable>	<environment>	*Named*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
variable of <environment>	variables	<environment variable>	<environment>	*Plain*
variable of <file>	variables	<string>	<file>	*Plain*
vendor name of <processor>	vendor names	<string>	<processor>	*Plain*
version <string>	versions	<version>	<world>	*NamedGlobal*
version block <integer> of <file>	version blocks	<file version block>	<file>	*Numbered*
version block <string> of <file>	version blocks	<file version block>	<file>	*Named*
version block of <file>	version blocks	<file version block>	<file>	*Plain*
version of <bios>	versions	<string>	<bios>	*Plain*
version of <current relay>	versions	<version>	<current relay>	*Plain*
version of <file>	versions	<version>	<file>	*Plain*
version of <site>	versions	<integer>	<site>	*Plain*
version string <string> of <module>	version strings	<string>	<module>	*Named*
volatile attribute of <metabase value>	volatile attributes	<boolean>	<metabase value>	*Plain*
waiting for download of <action>	waiting for downloads	<boolean>	<action>	*Plain*
wakeonlan enabled of <network adapter>	wakeonlan enableds	<boolean>	<network adapter>	*Plain*
warning event log event type	warning event log event types	<event log event type>	<world>	*PlainGlobal*
wednesday	wednesdays	<day of week>	<world>	*PlainGlobal*
week	weeks	<time interval>	<world>	*PlainGlobal*
weight of <selected server>	weights	<integer>	<selected server>	*Plain*
win32 exit code of <service>	win32 exit codes	<integer>	<service>	*Plain*
windows display time <string>	windows display times	<time>	<world>	*NamedGlobal*
windows file <string>	windows files	<file>	<world>	*NamedGlobal*
windows folder	windows folders	<folder>	<world>	*PlainGlobal*
wins enabled of <network adapter>	wins enableds	<boolean>	<network adapter>	*Plain*
winsock2 supported of <network>	winsock2 supporteds	<boolean>	<network>	*Plain*
wmi	wmis	<wmi>	<world>	*PlainGlobal*

© 2007 by BigFix, Inc.

Key Phrase	Plural	Creates a	From a	*Form*
wmi <string>	wmis	<wmi>	<world>	*NamedGlobal*
workstation trust account flag of <local user>	workstation trust account flags	<boolean>	<local user>	*Plain*
write attributes permission of <access control entry>	write attributes permissions	<boolean>	<access control entry>	*Plain*
write dac permission of <access control entry>	write dac permissions	<boolean>	<access control entry>	*Plain*
write extended attributes permission of <access control entry>	write extended attributes permissions	<boolean>	<access control entry>	*Plain*
write owner permission of <access control entry>	write owner permissions	<boolean>	<access control entry>	*Plain*
write permission of <access control entry>	write permissions	<boolean>	<access control entry>	*Plain*
write permission of <network share>	write permissions	<boolean>	<network share>	*Plain*
x32 application <string>	x32 applications	<application>	<world>	*NamedGlobal*
x32 file <string>	x32 files	<file>	<world>	*NamedGlobal*
x32 folder <string>	x32 folders	<folder>	<world>	*NamedGlobal*
x32 registry	x32 registries	<registry>	<world>	*PlainGlobal*
x64 application <string>	x64 applications	<application>	<world>	*NamedGlobal*
x64 file <string>	x64 files	<file>	<world>	*NamedGlobal*
x64 folder <string>	x64 folders	<folder>	<world>	*NamedGlobal*
x64 of <operating system>	x64s	<boolean>	<operating system>	*Plain*
x64 registry	x64 registries	<registry>	<world>	*PlainGlobal*
xml document of <file>	xml documents	<xml dom document>	<file>	*Plain*
xml document of <string>	xml documents	<xml dom document>	<string>	*Plain*
xpath <(string, string)> of <xml dom node>	xpaths	<xml dom node>	<xml dom node>	*Indexed*
xpath <string> of <xml dom node>	xpaths	<xml dom node>	<xml dom node>	*Named*
year	years	<number of months>	<world>	*PlainGlobal*
year <integer>	years	<year>	<world>	*NumberedGlobal*
year <string>	years	<year>	<world>	*NamedGlobal*

© 2007 by BigFix, Inc.

Key Phrase	**Plural**	**Creates a**	**From a**	***Form***
year of <date>	years	<year>	<date>	*Plain*
year of <month and year>	years	<year>	<month and year>	*Plain*
zone of <time of day with time zone>	zones	<time zone>	<time of day with time zone>	*Plain*
zoned time_of_day <string>	zoned times_of_day	<time of day with time zone>	<world>	*NamedGlobal*

© 2007 by BigFix, Inc.

Casting Operators

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

Key Phrase	Creates a	From a
<action lock state> as string	<string>	<action lock state>
<binary operator> as string	<string>	<binary operator>
<bios> as string	<string>	<bios>
<bit set> as integer	<integer>	<bit set>
<bit set> as string	<string>	<bit set>
<boolean> as boolean	<boolean>	<boolean>
<boolean> as string	<string>	<boolean>
<cast> as string	<string>	<cast>
<date> as string	<string>	<date>
<day of month> as integer	<integer>	<day of month>
<day of month> as string	<string>	<day of month>
<day of month> as two digits	<string>	<day of month>
<day of week> as string	<string>	<day of week>
<day of week> as three letters	<string>	<day of week>
<day of year> as string	<string>	<day of year>
<environment variable> as string	<string>	<environment variable>
<file content> as lowercase	<file content>	<file content>
<file content> as uppercase	<file content>	<file content>
<file> as string	<string>	<file>
<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>

© 2007 by BigFix, Inc.

Key Phrase	Creates a	From a
<hertz> as string	<string>	<hertz>
<html> as html	<html>	<html>
<html> as string	<string>	<html>
<integer> as bit set	<bit set>	<integer>
<integer> as bits	<bit set>	<integer>
<integer> as day_of_month	<day of month>	<integer>
<integer> as floating point	<floating point>	<integer>
<integer> as hexadecimal	<string>	<integer>
<integer> as integer	<integer>	<integer>
<integer> as month	<month>	<integer>
<integer> as string	<string>	<integer>
<integer> as year	<year>	<integer>
<ipv4 address> as ipv6 address	<ipv6 address>	<ipv4 address>
<ipv4 address> as string	<string>	<ipv4 address>
<ipv6 address> as compressed string	<string>	<ipv6 address>
<ipv6 address> as compressed string with ipv4	<string>	<ipv6 address>
<ipv6 address> as string	<string>	<ipv6 address>
<ipv6 address> as string with ipv4	<string>	<ipv6 address>
<ipv6 address> as string with leading zeros	<string>	<ipv6 address>
<language> as string	<string>	<language>
<local group member> as string	<string>	<local group member>
<metabase identifier> as integer	<integer>	<metabase identifier>
<metabase identifier> as string	<string>	<metabase identifier>
<metabase type> as integer	<integer>	<metabase type>
<metabase type> as string	<string>	<metabase type>

© 2007 by BigFix, Inc.

Key Phrase	Creates a	From a
<metabase user type> as integer	<integer>	<metabase user type>
<metabase user type> as string	<string>	<metabase user type>
<metabase value> as integer	<integer>	<metabase value>
<metabase value> as string	<string>	<metabase value>
<month and year> as string	<string>	<month and year>
<month> as integer	<integer>	<month>
<month> as string	<string>	<month>
<month> as three letters	<string>	<month>
<month> as two digits	<string>	<month>
<number of months> as string	<string>	<number of months>
<operating system> as string	<string>	<operating system>
<primary language> as string	<string>	<primary language>
<property> as string	<string>	<property>
<registry key value type> as string	<string>	<registry key value type>
<registry key value> as application	<application>	<registry key value>
<registry key value> as file	<file>	<registry key value>
<registry key value> as folder	<folder>	<registry key value>
<registry key value> as integer	<integer>	<registry key value>
<registry key value> as string	<string>	<registry key value>
<registry key value> as system file	<file>	<registry key value>

© 2007 by BigFix, Inc.

Key Phrase	Creates a	From a
<registry key value> as time	<time>	<registry key value>
<rope> as string	<string>	<rope>
<security descriptor> as string	<string>	<security descriptor>
<security identifier> as string	<string>	<security identifier>
<service> as string	<string>	<service>
<setting> as string	<string>	<setting>
<site version list> as string	<string>	<site version list>
<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 ipv6 address	<ipv6 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>

© 2007 by BigFix, Inc.

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>
<version> as string	<string>	<version>
<version> as version	<version>	<version>
<wmi object> as string	<string>	<wmi object>
<wmi select> as string	<string>	<wmi select>
<xml dom node> as text	<string>	<xml dom node>
<xml dom node> as xml	<string>	<xml dom node>
<year> as integer	<integer>	<year>

© 2007 by BigFix, Inc.

Key Phrase	Creates a	From a
<year> as string	<string>	<year>

© 2007 by BigFix, Inc.

Appendix

Folders on Windows Machines

On Windows machines, numeric identifiers can be used to locate many system folders. INF files are used to install system software components and device drives. INF files contain sections entitled DestinationDirs. This section is used with the corresponding CopyFiles section to specify destination locations for files placed on the system during the install. The Number identifies the directory. The numbers are sometimes called LDIDs and sometimes called DIRIDs. We call them install folders. Below is a table of install folders and the method BigFix uses to calculate the location.

Install folder#	Name	Calculated using
10	LDID_WIN	GetWindowsDirectory()
11	LDID_SYS	GetSystemDirectory()
12	LDID_IOS	GetSystemDirectory() + "\IOSUBSYS"
13	LDID_CMD	GetWindowsDirectory() + "\COMMAND"
14	LDID_CPL	GetPathFromCSIDL(CSIDL_CONTROLS)
15	LDID_PRINT	GetPathFromCSIDL(CSIDL_PRINTERS)
17	LDID_INF	GetWindowsDirectory() + "\INF"
18	LDID_HELP	GetWindowsDirectory() + "\HELP"
19	LDID_WINADMIN	*Registered Setup folder "WinAdminDir"
20	LDID_FONTS	GetPathFromCSIDL(CSIDL_ CSIDL_FONTS)
21	LDID_VIEWERS	GetSystemDirectory() + "\VIEWERS"
22	LDID_VMM32	GetSystemDirectory() + "\VMM32"
23	LDID_COLOR	*Registered Setup folder "ICMPath"
24	LDID_APPS	*Registered Setup folder "AppsDir"
25	LDID_SHARED	*Registered Setup folder "SharedDir"
26	LDID_WINBOOT	*Registered Setup folder "WinBootDir"
27	LDID_MACHINE	*Registered Setup folder "MachineDir"
28	LDID_HOST_WINBOOT	*Registered Setup folder "HostWinBootDir"
29	LDID_BOOT	*Registered Setup folder "BootDir"
30	LDID_BOOT_HOST	*Registered Setup folder "BootHost"
31	LDID_OLD_WINBOOT	*Registered Setup folder "OldWinBootDir"
32	LDID_OLD_WIN	*Registered Setup folder "OldWinDir"
33	LDID_OLD_DOS	*Registered Setup folder "OldDosDir"

*Registered Setup folders are stored in the Windows registry under the key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SETUP
An adjustable set of target locations has been added to the Windows Registry under the key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SETUP\VarLDID
Each value stored under this key is a string whose name is the VarLDID and whose value contains a path to a folder. For example, if the value named 28701 contains C:\Program Files, then
install folder "28701"
would return a folder corresponding to that location.

© 2007 by BigFix, Inc.

Processors

On Windows machines, the Vendor Name, Family, Type, Model, Extended Family, Extended Model and stepping are calculated using the CPUID instruction. The results depend upon the processor and the vendor of the processor. The inspectors return values based upon the Intel specification for the CPUID instruction. Other vendors or older processors may behave differently. An attempt is made to identify the Family and Family name for processors that do not support the CPUID instruction. You can depend upon the vendor name to distinguish the different vendors except that early versions of the 80486 from AMD are completely indistinguishable from an Intel processor. In this case "AmbiguousCPU" is returned for the vendor name. A complete list of bit values returned by the feature masks property is available in the Intel documentation. These can be found online at the Intel web site.
The speed is measured using a timed sequence of instructions. The speed returned may differ from the expected amount by a couple of MHz.
The CPUID instruction is executed with 1 in the EAX register to compute:

Stepping	Bits 0-3
Model	Bits 4-7
Family	Bits 8-11
Type	Bits 12-13
Extended Model	Bits 16-19
Extended Family	Bits 20-23

© 2007 by BigFix, Inc.

Processors (continued)

Numeric values returned for family of processor and string values returned by family name of processor are computed using the table below. For an unidentified family name, the “brand string” is returned, if available.

Vendor Name	Family Name	Family	Model	Extended Family
GenuineIntel	8086	0		
	80286	2		
	80386	3		
	80486	4		
	Pentium	5		
	Pentium Pro	6	0-2	
	Pentium II	6	38418	
	Pentium III	6	7 or greater	
	Pentium 4	15	0	0
AuthenticAMD	486	4		
	K5	5	0-5	
	K6	5	6 or greater	
	Athlon	6	1,2,4	
	Duron	6	3	
CyrixInstead	MediaGX	4		
	6x86	5	2	
	GXm	5	Not 2	
	6x86MX	6		
CentaurHauls	C6	5		

© 2007 by BigFix, Inc.

Index

A

absolute value of <hertz> · 50, 51, 316
absolute value of <integer> · 3, 10, 12, 316
absolute value of <time interval> · 64, 66, 316
access control entry · 254, 258, 316, 318, 321, 323, 325, 329, 330, 333, 335, 339, 341, 349, 350, 354, 357, 358, 362
access control list · 250, 254, 255, 256, 257, 258, 259, 260, 324, 326, 327, 328, 329
access mode of <access control entry> · 250, 316
accessed time of <filesystem object> · 135, 316
account disabled flag of <local user> · 263, 316
account expiration of <local user> · 263, 316
account name of <security identifier> · 262, 316
accounts operator flag of <local user> · 263, 316
action · 5, 29, 32, 91, 92, 93, 94, 103, 110, 119, 150, 187, 188, 199, 211, 213, 228, 242, 245, 270, 271, 272, 316, 317, 321, 323, 327, 330, 334, 335, 338, 340, 345, 346, 347, 355, 361, 364
action <integer> · 91, 270, 316
action lock state · 91, 316, 323, 327, 330, 340, 364
Action Objects · iv, 2, 270
action of <firewall rule> · 211, 213, 316
active action · 91, 270, 271, 316
active device · 91, 123, 125, 128, 137, 185, 186, 316, 321, 325, 326, 332, 334, 340, 348, 353, 355
active device file · 91, 137, 186, 316
active device file <string> · 91, 137, 316
active directory · 296, 297, 326, 339
active directory local computer · 297, 326, 339
active directory server · 339
active of <action> · 270, 316
active of <logged on user> · 269, 317
active start time of <action> · 270, 317
adapter of <network> · 273, 278, 317
address list of <network adapter> · 279, 282, 317
address of <network adapter> · 279, 291, 317
address of <network address list> · 282, 291, 317
address of <network ip interface> · 275, 291, 317
admin privilege of <local user> · 263, 317
administrator <string> of <client> · 233, 234, 317
administrator of <client> · 233, 234, 317
all firewall scope · 91, 203, 317
allow firewall action · 92, 211, 317
allow inbound echo request of <firewall icmp settings> · 209, 317
allow inbound mask request of <firewall icmp settings> · 209, 317
allow inbound router request of <firewall icmp settings> · 209, 317
allow inbound timestamp request of <firewall icmp settings> · 209, 317
allow outbound destination unreachable of <firewall icmp settings> · 209, 317
allow outbound packet too big of <firewall icmp settings> · 210, 317
allow outbound parameter problem of <firewall icmp settings> · 210, 317
allow outbound source quench of <firewall icmp settings> · 210, 317
allow outbound time exceeded of <firewall icmp settings> · 210, 317
allow redirect of <firewall icmp settings> · 210, 318
allowed workstations string of <local user> · 264, 318
ancestor of <filesystem object> · 135, 147, 318
ansi code page · 92, 120, 318
any ip version · 92, 219, 318

© 2007 by BigFix, Inc.

apparent registration server time · 92, 318
append permission of <access control entry> · 250, 318
application · 2, 92, 94, 99, 100, 112, 113, 115, 117, 119, 120, 121, 123, 124, 126, 128, 129, 130, 137, 143, 144, 145, 146, 147, 148, 150, 151, 154, 155, 160, 161, 163, 174, 175, 196, 197, 204, 214, 218, 233, 240, 241, 243, 244, 246, 248, 264, 304, 318, 325, 332, 338, 343, 350, 352, 358, 362, 366
application <string> · 92, 123, 126, 144, 150, 318
application <string> of <folder> · 144, 150, 318
application <string> of <registry key> · 126, 144, 318
application <string> of <registry> · 123, 144, 318
application event log · 92, 120, 240, 241, 243, 244, 246, 318
application folder <string> of <registry key> · 126, 147, 318
application folder <string> of <registry> · 123, 147, 318
application folder of <registry key> · 126, 148, 318
application name of <firewall rule> · 214, 318
application of <registry key> · 126, 144, 318
application of <registry> · 123, 145, 318
application parameter string of <local user> · 264, 318
application usage summary · 92, 174, 175, 318, 332, 338, 343, 352, 358
application usage summary <string> · 92, 174, 318
april · 68, 71, 76, 78, 79, 82, 83, 86, 92, 318, 319
april <integer> · 68, 76, 92, 318, 319
april <integer> of <integer> · 68, 319
april of <integer> · 83, 319
archive of <filesystem object> · 135, 319
argument string of <file shortcut> · 155, 319
attribute <integer> of <xml dom node> · 169, 170, 319
attribute <string> of <xml dom node> · 169, 170, 319
attribute of <xml dom node> · 169, 170, 319
attribute permission of <network share> · 277, 319
audit failure event log event type · 93, 245, 246, 319
audit level of <local mssql database> · 189, 319
audit success event log event type · 93, 245, 319
august · 68, 76, 79, 83, 93, 319
august <integer> · 68, 76, 93, 319
august <integer> of <integer> · 68, 319
august of <integer> · 83, 319
Authorization Objects · iv, 2, 250
authorized application of <firewall profile> · 196, 198, 319

B

backoffice bit <operating system suite mask> · 93, 319
bad password count of <local user> · 264, 319
bes license · 93, 187, 319
binary operator · 93, 120, 311, 312, 319, 338, 343, 351, 356, 364
binary operator <string> · 93, 311, 319
binary operator returning <type> · 93, 311, 319
bios · 2, 176, 177, 324, 361, 364
bit <integer> · 6, 12, 45, 46, 93, 319, 320
bit <integer> of <bit set> · 46, 319
bit <integer> of <integer> · 6, 12, 320
bit set · 12, 45, 46, 47, 93, 121, 319, 320, 338, 343, 346, 351, 364, 365
bit set <string> · 45, 93, 320
blade bit <operating system suite mask> · 93, 320
block firewall action · 94, 211, 320
boolean · 5, 6, 7, 8, 9, 12, 13, 14, 19, 22, 23, 24, 27, 29, 34, 35, 43, 44, 46, 47, 48, 51, 54, 57, 61, 73, 74, 78, 81, 85, 87, 89, 93, 94, 98, 99, 100, 103, 105, 110, 111, 114, 116, 117, 131, 132, 134, 135, 136, 152,

© 2007 by BigFix, Inc.

153, 158, 160, 175, 178, 180, 187, 189, 190,聊195, 196, 198, 199, 200, 201, 204, 205, 206, 208, 209, 210, 211, 213, 214, 215, 216, 218, 219, 220, 223, 229, 230, 231, 237, 246, 250, 251, 252, 253, 254, 255, 256, 257, 262, 263, 264, 265, 266, 267, 269, 270, 271, 272, 274, 275, 276, 277, 278, 279, 281, 283, 284, 288, 291, 293, 295, 301, 303, 304, 305, 306, 309, 310, 311, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 357, 358, 359, 360, 361, 362, 364, 367
boolean <string> · 6, 94, 320
boolean value <integer> of <wmi select> · 223, 320
boolean value of <wmi select> · 223, 320
boot time of <operating system> · 177, 320
brand id of <processor> · 181, 320
brand string of <processor> · 181, 320
broadcast address of <network ip interface> · 275, 291, 320
broadcast support of <network ip interface> · 275, 320
build number high of <operating system> · 177, 320
build number low of <operating system> · 178, 320
build number of <operating system> · 178, 320
built in of <firewall open port> · 205, 320
byte <integer> of <file> · 139, 320

C

can interact with desktop of <service> · 190, 320
case insensitive regex <string> · 47, 94, 320
case insensitive regular expression <string> · 47, 94, 320
cast · 12, 28, 94, 121, 158, 159, 160, 223, 307, 308, 314, 315, 320, 343, 346, 364
cast <string> · 94, 314, 320
cast from of <type> · 308, 314, 320
cast returning <type> · 94, 314, 320
casts · 2, 10, 21, 121, 307, 308, 309, 311, 312, 314, 315, 320
category of <event log record> · 242, 320
change notification permission of <access control entry> · 250, 321
character <integer> · 29, 32, 38, 94, 321
character <integer> of <string> · 32, 38, 321
character of <string> · 32, 38, 321
checkpoint of <service> · 190, 321
child node <integer> of <xml dom node> · 169, 171, 321
child node of <xml dom node> · 169, 171, 321
class of <active device> · 185, 321
client · 2, 35, 91, 92, 93, 94, 97, 113, 143, 148, 151, 159, 174, 175, 181, 187, 188, 221, 226, 228, 233, 234, 235, 246, 247, 272, 317, 321, 354
client folder of <site> · 148, 226, 321
client license · 93, 94, 187, 321
Client Objects · iv, 2, 233
code page of <local user> · 264, 321
codepage of <file version block> · 162, 321
comment of <local group> · 239, 321
comment of <local user> · 264, 321
comment of <network share> · 277, 321
common name of <license> · 187, 321
communications bit <operating system suite mask> · 94, 321
communications operator flag of <local user> · 264, 321
competition size of <selected server> · 236, 321
competition weight of <selected server> · 236, 321
complete time of <action> · 270, 321
component <integer> of <distinguished name> · 298, 321
component <integer> of <site version list> · 230, 321
component of <distinguished name> · 298, 322
compressed of <filesystem object> · 135, 322
computer id · 95, 322
computer name · 95, 322

© 2007 by BigFix, Inc.

computer of <event log record> · 242, 322
concatenation <html> of <html> · 322
concatenation <html> of <string> · 32, 322
concatenation <string> of <string> · 29, 32, 322
concatenation of <string> · 29, 32, 322
conjunction of <boolean> · 6, 8, 322
connection · 95, 96, 107, 108, 121, 269, 273, 283, 285, 286, 287, 288, 289, 290, 322, 325, 331, 334, 341, 343, 356
connection of <network> · 273, 285, 322
connection status · 95, 96, 121, 285, 286, 287, 288, 322, 356
connection status <integer> · 95, 286, 322
connection status authenticating · 95, 286, 322
connection status authentication failed · 95, 286, 322
connection status authentication succeeded · 95, 286, 322
connection status connected · 95, 121, 286, 322
connection status connecting · 95, 286, 322
connection status disconnected · 95, 286, 322
connection status disconnecting · 95, 286, 322
connection status hardware disabled · 96, 287, 322
connection status hardware malfunction · 96, 287, 322
connection status media disconnected · 96, 287, 322
connection status no hardware present · 96, 287, 322
constrained of <action> · 270, 323
content of <file> · 139, 157, 323
control of <security descriptor> · 260, 323
controller of <action lock state> · 323
Conventions Used in this manual · 3
country code of <local user> · 264, 323
create file permission of <access control entry> · 250, 323
create folder permission of <access control entry> · 250, 323
create link permission of <access control entry> · 250, 323
create permission of <network share> · 277, 323
create subkey permission of <access control entry> · 251, 323
creation time of <filesystem object> · 135, 323
csd version of <operating system> · 178, 323
csidl folder <integer> · 96, 148, 323
current date · 9, 16, 68, 71, 75, 82, 83, 86, 96, 97, 323
current day_of_month · 73, 75, 96, 323
current day_of_week · 9, 71, 96, 323
current day_of_year · 76, 97, 323
current firewall profile type · 97, 121, 200, 323
current month · 9, 16, 79, 81, 82, 83, 86, 97, 323
current month_and_year · 83, 86, 97, 323
current profile of <firewall policy> · 197, 202, 323
current profile type of <firewall> · 194, 200, 323
current relay · 97, 246, 323, 361
current site · 97, 143, 226, 228, 323
current time_of_day · 59, 97, 324
current time_of_day <time zone> · 59, 97, 324
current user · 2, 97, 123, 125, 267, 268, 324, 343
current user key <logged on user> of <registry> · 123, 125, 324
current year · 86, 88, 90, 97, 324
currently active of <firewall rule> · 214, 324
custom firewall scope · 98, 203, 324
custom site subscription effective date <string> · 98, 324
customized of <firewall service> · 206, 324

D

dacl of <security descriptor> · 254, 260, 324
datacenter bit <operating system suite mask> · 98, 324
date · 1, 9, 15, 27, 30, 35, 53, 54, 61, 62, 63, 68, 69, 70, 71, 73, 74, 75, 76, 77, 78, 80, 81, 83, 84, 85, 86, 88, 89, 96, 98, 131, 135,

© 2007 by BigFix, Inc.

136, 137, 143, 175, 176, 177, 187, 188, 228, 235, 277, 319, 323, 324, 325, 330, 331, 332, 337, 341, 342, 343, 345, 353, 363, 364, 367
date <string> · 68, 98, 324
date <time zone> of <time> · 53, 68, 324
date of <bios> · 176, 324
day · 9, 12, 30, 31, 32, 35, 55, 56, 57, 59, 61, 64, 65, 67, 68, 69, 71, 72, 73, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85, 89, 92, 93, 96, 97, 98, 100, 102, 104, 107, 108, 109, 110, 113, 114, 115, 116, 117, 118, 177, 228, 244, 267, 318, 319, 323, 324, 325, 330,驷332, 334, 337, 341, 342, 345, 352, 353, 356, 357, 358, 361, 363, 364, 365, 367, 368
day of <day of year> · 73, 77, 324
day of month · 12, 30, 69, 73, 74, 75, 77, 81, 85, 96, 98, 323, 324, 364, 365, 367
day of week · 30, 68, 69, 71, 72, 73, 96, 98, 102, 108, 113, 115, 116, 117, 118, 323, 324, 332, 342, 352, 356, 357, 358, 361, 364, 367
day of year · 69, 74, 75, 76, 77, 78, 81, 85, 89, 92, 93, 97, 98, 100, 104, 107, 109, 110, 114, 318, 319, 323, 324, 325, 330, 337, 341, 342, 345, 353, 364
day_of_month <integer> · 74, 98, 324
day_of_month <string> · 74, 98, 324
day_of_month of <date> · 69, 74, 324
day_of_week <string> · 71, 98, 324
day_of_week of <date> · 69, 71, 324
day_of_year of <date> · 69, 76, 324
december · 68, 76, 79, 82, 83, 98, 325
december <integer> · 68, 76, 98, 325
december <integer> of <integer> · 68, 325
december of <integer> · 83, 325
default value of <registry key> · 126, 130, 325
default web browser · 99, 145, 325
delete child permission of <access control entry> · 251, 325
delete permission of <access control entry> · 251, 325
delete permission of <network share> · 277, 325
descendant folder of <folder> · 148, 150, 325
descendant of <folder> · 137, 150, 325
description of <active device> · 185, 325
description of <event log record> · 242, 325
description of <firewall rule> · 214, 325
description of <network adapter> · 279, 325
device key <string> of <registry> · 123, 125, 325
device key of <registry> · 123, 125, 325
device name of <connection> · 285, 325
dhcp enabled of <network adapter> · 279, 325
dhcp server of <network adapter> · 279, 291, 325
direct object type of <property> · 307, 310, 325
disjunction of <boolean> · 6, 8, 325
display name of <service> · 190, 325
distance of <selected server> · 16, 236, 325
distinguished name · 296, 297, 298, 299, 321, 322, 325, 326, 359, 360
distinguished name <string> · 297, 325
distinguished name component · 298, 321, 322, 359, 360
distinguished name error message of <active directory local computer> · 296, 326
distinguished name of <active directory local computer> · 296, 326
divided by zero of <floating point> · 22, 326
dmi · 2, 99, 326
DMI Objects · 2
dns name · 99, 326
dns server of <network adapter> · 279, 282, 326
dns server of <network> · 273, 282, 326
dns suffix of <network adapter> · 279, 326
domain firewall profile type · 99, 200, 201, 326
domain name of <security identifier> · 262, 326
domain profile of <firewall policy> · 197, 202, 326
domain user · 2, 99, 263, 326
domain user <string> · 99, 263, 326
drive · 36, 40, 99, 102, 135, 148, 149, 152, 153, 154, 155, 259, 264, 326, 330, 331,

© 2007 by BigFix, Inc.

332, 333, 343, 345, 351, 358, 359
drive <string> · 99, 152, 326
drive of <filesystem object> · 135, 152, 326
driver key of <active device> · 125, 185, 326
driver key of <registry key> · 125, 126, 326
driver key value name of <active device> · 185, 326

E

edge traversal allowed of <firewall rule> · 214, 326
effective access mode for <string> of <access control list> · 254, 326
effective access system security permission for <string> of <access control list> · 254, 326
effective append permission for <string> of <access control list> · 254, 327
effective change notification permission for <string> of <access control list> · 254, 327
effective create file permission for <string> of <access control list> · 255, 327
effective create folder permission for <string> of <access control list> · 255, 327
effective create link permission for <string> of <access control list> · 255, 327
effective create subkey permission for <string> of <access control list> · 255, 327
effective date of <action lock state> · 327
effective date of <setting> · 235, 327
effective delete child permission for <string> of <access control list> · 255, 327
effective delete permission for <string> of <access control list> · 255, 327
effective enumerate subkeys permission for <string> of <access control list> · 255, 327
effective execute permission for <string> of <access control list> · 255, 327
effective generic all permission for <string> of <access control list> · 255, 327
effective generic execute permission for <string> of <access control list> · 256, 327
effective generic read permission for <string> of <access control list> · 256, 327
effective generic write permission for <string> of <access control list> · 256, 327
effective list permission for <string> of <access control list> · 256, 327
effective maximum allowed permission for <string> of <access control list> · 256, 328
effective query value permission for <string> of <access control list> · 256, 328
effective read attributes permission for <string> of <access control list> · 256, 328
effective read control permission for <string> of <access control list> · 256, 328
effective read extended attributes permission for <string> of <access control list> · 256, 328
effective read permission for <string> of <access control list> · 257, 328
effective set value permission for <string> of <access control list> · 257, 328
effective synchronize permission for <string> of <access control list> · 257, 328
effective traverse permission for <string> of <access control list> · 257, 328
effective write attributes permission for <string> of <access control list> · 257, 328
effective write dac permission for <string> of <access control list> · 257, 328
effective write extended attributes permission for <string> of <access control list> · 257, 328
effective write owner permission for <string> of <access control list> · 257, 328
effective write permission for <string> of <access control list> · 257, 328
element of <integer set> · 18, 328
element of <string set> · 42, 328
email address of <license> · 187, 328
embedded nt bit <operating system suite mask> · 99, 328
embedded restricted bit <operating system suite mask> · 99, 329
enabled of <firewall authorized application> · 196, 329
enabled of <firewall open port> · 205, 329
enabled of <firewall rule> · 214, 329
enabled of <firewall service> · 206, 329
enabled of <internet connection firewall> ·

© 2007 by BigFix, Inc.

283, 329
enabled of <port mapping> · 284, 329
end of <substring> · 37, 40, 329
end of <time range> · 62, 329
enterprise bit <operating system suite mask> · 100, 329
entry of <access control list> · 250, 258, 329
enumerate subkeys permission of <access control entry> · 251, 329
environment · 2, 33, 100, 191, 248, 249, 329, 330, 343, 360, 361, 364
Environment Objects · iv, 2, 248
environment variable · 2, 33, 100, 248, 249, 343, 360, 361, 364
error <string> · 49, 100, 329
error event log event type · 100, 245, 246, 329
escape of <string> · 32, 131, 329
evaluation of <license> · 187, 329
event id of <event log record> · 242, 329
event log · 92, 93, 100, 103, 113, 115, 118, 120, 121, 240, 241, 242, 243, 244, 245, 246, 261, 318, 319, 320, 322, 325, 329, 335, 339, 346, 350, 353, 355, 357, 358, 360, 361
event log <string> · 100, 240, 329
event log event type · 93, 100, 103, 118, 242, 245, 246, 319, 329, 335, 361
event log event type <integer> · 100, 245, 329
event log record · 241, 242, 243, 320, 322, 325, 329, 339, 350, 355, 357, 358, 360
event type of <event log record> · 242, 245, 329
exceptions allowed of <firewall profile> · 198, 329
excluded interface of <firewall profile> · 198, 330
executable file format of <file> · 139, 330
execute permission of <access control entry> · 251, 330
execute permission of <network share> · 277, 330
execution · 2, 93, 245, 255, 256, 270
expand environment string of <string> · 33, 330
expiration date of <action lock state> · 330
expiration date of <license> · 187, 330
expiration state of <license> · 187, 330
extended family of <processor> · 181, 330
extended model of <processor> · 181, 330
external port of <port mapping> · 284, 330

F

false · 6, 8, 46, 47, 100, 276, 330
family name of <processor> · 181, 330
family of <network interface> · 274, 330
family of <processor> · 182, 330
feature mask of <processor> · 182, 330
february · 68, 76, 79, 83, 100, 330
february <integer> · 68, 76, 100, 330
february <integer> of <integer> · 68, 330
february of <integer> · 83, 330
file · 1, 3, 15, 16, 35, 36, 48, 52, 53, 91, 96, 101, 112, 113, 115, 116, 118, 119, 122, 123, 124, 125, 126, 128, 129, 130, 131, 132, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 165, 166, 167, 168, 169, 171, 172, 173, 174, 190, 207, 227, 228, 233, 250, 251, 252, 253, 254, 255, 259, 277, 281, 309, 310, 316, 319, 320, 321, 323, 325, 327, 330, 331, 332, 333, 335, 337, 338, 339, 341, 344, 346, 347, 348, 349, 350, 352, 353, 354, 355, 357, 360, 361, 362, 364, 366
file <string> · 101, 138, 150, 330
file <string> of <folder> · 138, 150, 330
file content · 139, 157, 158, 323, 364
file extension <string> of <registry> · 123, 125, 330
file line · 140, 166, 167, 309, 339, 344, 348
file of <folder> · 138, 150, 330
file of <service> · 138, 190, 330
file section · 141, 156, 158, 337, 352
file shortcut · 141, 319, 335, 347, 354, 355
File System Objects · 1
file system type of <drive> · 152, 330
file type <string> of <registry> · 123, 125, 330

© 2007 by BigFix, Inc.

file version block · 140, 141, 142, 163, 321, 332, 335, 338, 346, 350, 360, 361
file version of <file> · 139, 142, 158, 159, 331
file_and_print firewall service type · 101, 207, 331
file_supports_encryption of <drive> · 152, 331
file_supports_object_ids of <drive> · 152, 331
file_supports_reparse_points of <drive> · 152, 331
file_supports_sparse_files of <drive> · 152, 331
file_volume_quotas of <drive> · 152, 331
filesystem object · 101, 116, 119, 135, 136, 137, 138, 147, 149, 150, 152, 316, 318, 319, 322, 323, 326, 334, 340, 342, 343, 345, 346, 347, 350, 357
Filesystem Objects · iii, 135
final part <time interval> of <time range> · 62, 331
find file <string> of <folder> · 138, 150, 331
finite of <floating point> · 22, 331
firewall · 2, 91, 92, 94, 97, 98, 99, 101, 102, 103, 104, 105, 109, 110, 111, 112, 115, 117, 121, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 279, 283, 285, 316, 317, 318, 319, 320, 323, 324, 325, 326, 329, 330, 331, 333, 334, 335, 336, 337, 339, 340, 343, 345, 346, 348, 349, 351, 352, 353, 354, 355, 359, 360
firewall action · 92, 94, 101, 211, 213, 316, 317, 320, 331
firewall action <integer> · 101, 211, 331
firewall authorized application · 198, 220, 319, 329, 337, 343, 348, 351, 352
firewall enabled of <firewall profile> · 198, 331
firewall icmp settings · 199, 317, 318, 335
firewall local policy modify state · 101, 102, 103, 110, 194, 212, 213, 331, 333, 335, 339, 346
firewall local policy modify state <integer> · 101, 212, 331
Firewall Objects · iii, 2, 194
firewall of <connection> · 283, 285, 331
firewall open port · 199, 205, 206, 218, 220, 320, 329, 333, 337, 343, 348, 349, 351, 352
firewall policy · 194, 197, 198, 202, 203, 213, 323, 326, 339, 348, 349, 355
firewall profile · 97, 99, 101, 111, 115, 121, 194, 197, 198, 199, 200, 201, 202, 203, 212, 319, 323, 326, 329, 330, 331, 333, 335, 345, 346, 348, 349, 351, 352, 353, 355, 359
firewall profile type · 97, 99, 101, 111, 115, 121, 194, 200, 201, 323, 326, 331, 348, 349, 355, 359
firewall profile type <integer> · 101, 200, 331
firewall remote admin settings · 199, 351
firewall rule · 195, 198, 199, 211, 213, 214, 215, 216, 217, 218, 316, 318, 324, 325, 326, 329, 334, 335, 336, 339, 340, 343, 346, 349, 351, 352, 353
firewall scope · 91, 98, 101, 105, 197, 203, 204, 205, 207, 317, 324, 331, 340, 352
firewall scope <integer> · 101, 203, 331
firewall service · 101, 109, 112, 117, 195, 199, 204, 206, 207, 208, 213, 217, 220, 324, 329, 331, 333, 337, 343, 345, 351, 352, 353, 354, 359, 360
firewall service restriction · 195, 213, 217, 352, 354
firewall service type · 101, 109, 112, 117, 207, 208, 331, 345, 351, 359, 360
firewall service type <integer> · 101, 207, 331
first <day of week> of <month and year> · 84, 331
first <integer> of <string> · 33, 38, 331
first <string> of <string> · 33, 38, 331
first child of <xml dom node> · 169, 171, 331
first friday of <month and year> · 84, 332
first match <regular expression> of <string> · 48, 332
first monday of <month and year> · 84, 332
first raw version block of <file> · 140, 161, 332
first saturday of <month and year> · 84, 332

© 2007 by BigFix, Inc.

first start time of <application usage summary> · 174, 332
first sunday of <month and year> · 84, 332
first thursday of <month and year> · 85, 332
first tuesday of <month and year> · 85, 332
first wednesday of <month and year> · 85, 332
fixlet · 226, 227, 230, 231, 232, 332, 334, 335, 343, 346, 351, 360
fixlet of <site> · 226, 230, 332
fixlet_header · 230, 231, 334, 343, 360
floating point · 7, 8, 9, 10, 11, 12, 13, 14, 15, 20, 21, 22, 23, 24, 25, 26, 27, 30, 35, 52, 102, 313, 326, 331, 332, 335, 336, 339, 341, 343, 344, 345, 346, 351, 354, 355, 359, 364, 365, 367
floating point <string> · 20, 102, 332
folder · 15, 16, 35, 36, 56, 96, 102, 103, 116, 118, 119, 121, 123, 126, 130, 131, 135, 136, 137, 138, 143, 144, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 160, 167, 168, 173, 174, 186, 226, 227, 228, 250, 251, 252, 253, 255, 259, 266, 318,駉321, 323, 325, 327, 330, 331, 332, 335, 346, 351, 353, 357, 361, 362, 366, 370
folder <string> · 102, 148, 151, 153, 332
folder <string> of <drive> · 148, 153, 332
folder <string> of <folder> · 148, 151, 332
folder of <folder> · 148, 151, 332
following text of <string position> · 37, 38, 332
following text of <substring> · 39, 40, 332
free space of <drive> · 153, 332
friday · 72, 102, 332
friendly name of <active device> · 185, 332
friendly name of <network adapter> · 279, 332
fs_case_is_preserved of <drive> · 153, 332
fs_case_sensitive of <drive> · 153, 333
fs_file_compression of <drive> · 153, 333
fs_persistent_acls of <drive> · 153, 333
fs_unicode_stored_on_disk of <drive> · 153, 333
fs_vol_is_compressed of <drive> · 153, 333
full name of <local user> · 264, 333
full wmi <string> · 102, 221, 333

G

gateway address <integer> of <selected server> · 236, 291, 333
gateway address of <selected server> · 236, 291, 333
gateway list of <network adapter> · 279, 282, 333
gateway of <network adapter> · 279, 291, 333
gather schedule authority of <site> · 226, 333
gather schedule time interval of <site> · 226, 333
gather url of <license> · 187, 333
generic all permission of <access control entry> · 251, 333
generic execute permission of <access control entry> · 251, 333
generic read permission of <access control entry> · 251, 333
generic write permission of <access control entry> · 251, 333
ghz · 50, 51, 52, 102, 333
globally open port of <firewall profile> · 199, 204, 333
globally open port of <firewall service> · 204, 206, 333
gp override firewall local policy modify state · 102, 212, 333
greatest hz · 50, 51, 102, 333
greatest integer · 10, 102, 333
greatest time interval · 64, 102, 334
group <integer> of <site> · 227, 229, 334
group leader of <action> · 271, 334
group of <security descriptor> · 260, 261, 334
grouping of <firewall rule> · 215, 334
guest privilege of <local user> · 264, 334
guid of <connection> · 285, 334

H

hardware id of <active device> · 185, 334
has blank sa password of <local mssql

© 2007 by BigFix, Inc.

database> · 189, 334
header <string> of <fixlet> · 230, 231, 334
header of <fixlet> · 231, 334
hertz · 14, 27, 50, 51, 52, 102, 103, 105, 108, 179, 182, 313, 316, 333, 334, 338, 342, 347, 354, 355, 365
hexadecet <integer> of <ipv6 address> · 295, 334
hexadecimal integer <string> · 10, 102, 334
hexadecimal string <string> · 29, 102, 334
hidden of <filesystem object> · 135, 334
home directory drive of <local user> · 264, 334
home directory of <local user> · 264, 334
home directory required flag of <local user> · 265, 334
host name of <root server> · 247, 334
hostname · 103, 334
hour · 29, 55, 57, 59, 60, 61, 64, 65, 67, 103, 334, 342
hour_of_day of <time of day with time zone> · 60, 334
hour_of_day of <time of day> · 57, 334
html · 32, 47, 48, 128, 322, 365, 367
hyperthreading capable · 103, 334
hyperthreading enabled · 103, 334
hz · 27, 50, 52, 103, 334

I

ia64 of <operating system> · 178, 334
icmp settings of <firewall profile> · 199, 209, 335
icmp types_and_codes string of <firewall rule> · 215, 335
icon index of <file shortcut> · 155, 335
icon pathname of <file shortcut> · 155, 335
id of <action> · 271, 335
id of <file version block> · 162, 335
id of <fixlet> · 231, 335
id of <root server> · 247, 335
id of <site group> · 229, 335
identifier of <metabase value> · 301, 305, 335
image path of <service> · 190, 335
inbound blocked firewall local policy modify state · 103, 212, 335
inbound connections allowed of <firewall profile> · 199, 335
inbound of <firewall rule> · 215, 335
index type of <property> · 307, 310, 335
inexact of <floating point> · 7, 22, 335
infinite of <floating point> · 7, 22, 335
information event log event type · 103, 245, 335
inherit attribute of <metabase value> · 305, 335
inheritance of <access control entry> · 252, 335
initial part <time interval> of <time range> · 62, 63, 335
insert path attribute of <metabase value> · 305, 335
Inspector List · 2
install folder <integer> · 103, 148, 335
instance name of <local mssql database> · 189, 336
integer · 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 28, 30, 32, 33, 37, 41, 42, 44, 45, 46, 49, 50, 51, 57, 60, 66, 73, 74, 75, 79, 80, 83, 84, 87, 88, 89, 91, 92, 93, 95, 96, 98, 101, 102, 103, 104, 105, 106, 107, 108, 110, 111, 117, 120, 121, 130, 131, 133, 134, 139, 140, 141, 148, 153, 154, 155, 167, 171, 175, 177, 178, 179, 181, 182, 184, 186, 188, 189, 190, 191, 200, 205, 207, 212, 219, 223, 224, 228, 229, 230, 231, 236, 241, 242, 243, 244, 247, 250, 252, 253, 254, 260, 264, 265, 266, 270, 271, 274, 278, 280, 281, 284, 286, 289, 295, 298, 301, 303, 304, 305, 308, 315, 316, 318, 319, 320, 321, 322, 323, 325, 326, 328, 329, 330, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 352, 353, 354, 355, 356, 358, 359, 360, 361, 364, 365, 366, 367, 368
integer <integer> · 10, 103, 336
integer <string> · 10, 103, 336
integer ceiling of <floating point> · 10, 22, 336

© 2007 by BigFix, Inc.

integer floor of <floating point> · 11, 22, 336
integer range · 11, 16, 236, 325, 340, 360
integer set · 13, 14, 18, 19, 328, 336, 354, 355, 359
integer value <integer> of <wmi select> · 223, 336
integer value of <wmi select> · 224, 336
integer with multiplicity · 13, 343, 359
interdomain trust account flag of <local user> · 265, 336
interface <integer> of <network> · 273, 274, 336
interface of <firewall rule> · 215, 336
interface of <network> · 273, 274, 336
interface types string of <firewall rule> · 215, 336
internal port of <port mapping> · 284, 336
internet connection firewall · 279, 283, 285, 329, 331, 336, 348
internet connection firewall of <network adapter> · 279, 283, 336
internet protocol · 104, 116, 117, 205, 216, 218, 219, 336, 349, 357, 359
internet protocol <integer> · 104, 218, 336
intersection of <integer set> · 18, 336
intersection of <string set> · 42, 336
Introspectors · v, 2, 307
invalid of <floating point> · 7, 22, 336
ip address of <selected server> · 236, 292, 336
ip interface <integer> of <network> · 273, 275, 336
ip interface of <network> · 273, 275, 336
ip version · 92, 104, 196, 205, 206, 219, 220, 318, 336, 337
ip version <integer> · 104, 219, 336
ip version of <firewall authorized application> · 196, 220, 337
ip version of <firewall open port> · 205, 220, 337
ip version of <firewall service> · 206, 220, 337
ipv4 · 104, 219, 220, 236, 275, 276, 279, 280, 281, 282, 284, 292, 293, 294, 295, 317, 320, 325, 333, 336, 337, 348, 352, 356, 357, 365
ipv4 address · 104, 236, 275, 276, 279, 280, 281, 282, 284, 292, 293, 294, 295, 317, 320, 325, 333, 336, 337, 348, 352, 356, 357, 365
ipv4 address <string> · 104, 292, 337
ipv4 part of <ipv6 address> · 292, 295, 337
ipv6 · 30, 104, 219, 220, 280, 293, 294, 295, 334, 337, 365, 367
ipv6 address · 30, 104, 280, 293, 294, 295, 334, 337, 365, 367
ipv6 address <string> · 104, 294, 337
ipv6 address of <network adapter> · 280, 294, 337
ipv6 dns server of <network adapter> · 280, 294, 337

J

january · 68, 76, 79, 81, 82, 83, 86, 104, 337
january <integer> · 68, 76, 104, 337
january <integer> of <integer> · 68, 337
january of <integer> · 83, 337
july · 68, 76, 79, 83, 104, 337
july <integer> · 68, 76, 104, 337
july <integer> of <integer> · 68, 337
july of <integer> · 83, 337
june · 69, 75, 76, 79, 83, 104, 337
june <integer> · 69, 76, 104, 337
june <integer> of <integer> · 69, 337
june of <integer> · 83, 337

K

key <string> of <file section> · 156, 337
key <string> of <file> · 140, 337
key <string> of <metabase key> · 302, 337
key <string> of <metabase> · 300, 302, 337
key <string> of <registry key> · 126, 127, 338
key <string> of <registry> · 123, 126, 338
key of <metabase key> · 302, 338
key of <metabase> · 300, 302, 338
key of <registry key> · 126, 127, 338
Key Phrases (Inspectors) · v, 316
keywords · 1, 2, 3, 5, 122, 128, 135, 176, 226, 270, 273, 316

© 2007 by BigFix, Inc.

khz · 50, 105, 338

L

language · 1, 5, 6, 44, 91, 115, 117, 118, 140, 159, 161, 162, 163, 164, 165, 181, 192, 193, 264, 307, 309, 311, 312, 314, 338, 348, 357, 360, 365
language of <file version block> · 162, 338
last <integer> of <string> · 33, 39, 338
last <string> of <string> · 33, 39, 338
last change time of <action> · 271, 338
last child of <xml dom node> · 169, 171, 338
last gather time of <site> · 227, 338
last logoff of <local user> · 265, 338
last logon of <local user> · 265, 338
last start time of <application usage summary> · 175, 338
last time seen of <application usage summary> · 175, 338
leap of <year> · 89, 338
lease expires of <network adapter> · 280, 338
lease obtained of <network adapter> · 280, 338
least hz · 50, 52, 105, 338
least integer · 9, 11, 105, 338
least significant one bit of <bit set> · 46, 338
least time interval · 65, 105, 338
left operand type of <binary operator> · 307, 311, 338
left shift <integer> of <bit set> · 45, 46, 338
length of <event log record> · 242, 339
length of <month and year> · 85, 339
length of <rope> · 11, 44, 339
length of <string> · 11, 33, 339
length of <time range> · 63, 339
length of <year> · 89, 339
less significance <integer> of <floating point> · 20, 22, 339
license · 93, 94, 187, 188, 319, 321, 328, 329, 330, 333, 341, 346, 350, 352, 354, 355
line <integer> of <file> · 140, 166, 339
line containing <string> of <file> · 140, 166, 339
line number of <file line> · 167, 339
line of <file> · 140, 166, 339
line starting with <string> of <file> · 140, 166, 339
link speed of <network adapter> · 280, 339
list permission of <access control entry> · 252, 339
local addresses string of <firewall rule> · 215, 339
local administrator · 105, 239, 339
local computer of <active directory server> · 296, 297, 339
local group · 105, 113, 238, 239, 240, 261, 321, 339, 342, 343, 354, 365
local group <string> · 105, 238, 339
local group member · 239, 240, 261, 342, 354, 365
local mssql database · 105, 188, 319, 334, 336, 339, 340, 352
local mssql database <string> · 105, 188, 339
local policy modify state of <firewall> · 194, 212, 339
local policy of <firewall> · 194, 202, 339
local ports string of <firewall rule> · 216, 340
local subnet firewall scope · 105, 203, 340
local time <string> · 106, 340
local time zone · 30, 52, 53, 58, 59, 61, 97, 106, 340
local user · 99, 106, 263, 267, 268, 316, 317, 318, 319, 321, 323, 326, 333, 334, 336, 338, 340, 341, 343, 344, 345, 347, 348, 349, 352, 353, 357, 360, 362
local user <string> · 106, 263, 340
location information of <active device> · 185, 340
location of <filesystem object> · 136, 340
lock string of <action lock state> · 340
locked of <action lock state> · 340
locked out flag of <local user> · 265, 340
logged on user · 97, 106, 268, 317, 324, 340, 343, 351, 354
logical processor count · 106, 121, 340
login account of <service> · 190, 340
login mode of <local mssql database> · 189, 340
logon count of <local user> · 265, 340
logon script of <local user> · 265, 340
logon server of <local user> · 265, 340

© 2007 by BigFix, Inc.

loopback of <network ip interface> · 275, 340
lower bound of <integer range> · 11, 16, 340

M

mac address of <network adapter> · 280, 340
main gather service · 106, 189, 340
main processor · 106, 181, 183, 340
major version of <operating system> · 178, 340
manufacturer of <active device> · 186, 340
march · 69, 76, 79, 83, 107, 340, 341
march <integer> · 69, 76, 107, 341
march <integer> of <integer> · 69, 341
march of <integer> · 83, 341
masthead of <site> · 138, 227, 341
match <regular expression> of <string> · 48, 341
maximum allowed permission of <access control entry> · 252, 341
maximum of <integer> · 11, 12, 341
maximum of <time interval> · 65, 66, 341
maximum of <time> · 52, 53, 341
maximum seat count of <license> · 188, 341
maximum storage of <local user> · 265, 341
maximum transmission unit of <network adapter> · 280, 341
may · 2, 16, 52, 53, 69, 77, 79, 83, 97, 107, 110, 118, 123, 125, 126, 128, 130, 132, 142, 146, 151, 154, 156, 159, 161, 166, 167, 176, 179, 182, 206, 236, 246, 248, 261, 269, 278, 282, 309, 341, 371
may <integer> · 69, 77, 107, 341
may <integer> of <integer> · 69, 341
may of <integer> · 83, 341
mean of <floating point> · 20, 23, 341
mean of <integer> · 13, 20, 341
media type · 107, 108, 285, 289, 290, 291, 341, 342
media type <integer> · 107, 289, 341
media type bridge · 107, 289, 341
media type direct · 107, 289, 341
media type isdn · 107, 289, 341
media type lan · 107, 289, 341
media type of <connection> · 285, 290, 341
media type phone · 107, 290, 341
media type pppoe · 107, 290, 341
media type shared access host lan · 108, 290, 341
media type shared access host ras · 108, 290, 342
media type tunnel · 108, 290, 342
member of <local group> · 239, 342
member of <site group> · 229, 342
metabase · 108, 300, 301, 302, 303, 304, 305, 306, 335, 337, 338, 342, 344, 350, 353, 359, 360, 361, 365, 366
metabase identifier · 301, 305, 335, 365
metabase key · 300, 302, 303, 305, 337, 338, 344, 360
metabase type · 303, 305, 359, 365
metabase user type · 304, 306, 360, 366
metabase value · 302, 303, 304, 305, 306, 335, 350, 353, 359, 360, 361, 366
metric <integer> of <operating system> · 178, 342
mhz · 50, 52, 108, 342
microsecond · 64, 65, 108, 342
Microsoft IIS Metabase Objects · v, 2, 300
midnight · 55, 56, 108, 342
millisecond · 28, 65, 66, 108, 342
minimum of <integer> · 11, 13, 342
minimum of <time interval> · 65, 66, 342
minimum of <time> · 53, 54, 342
minor version of <operating system> · 178, 342
minute · 29, 55, 57, 58, 60, 61, 65, 108, 342, 352
minute_of_hour of <time of day with time zone> · 60, 342
minute_of_hour of <time of day> · 57, 342
model of <processor> · 182, 342
modification time of <filesystem object> · 136, 342
module · 108, 170, 342, 361
module <string> · 108, 342
monday · 72, 86, 108, 342
month · 12, 16, 30, 55, 68, 69, 70, 71, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 92, 93, 96, 97, 98, 100, 104, 107, 108, 109, 110, 114, 177, 244, 318,

© 2007 by BigFix, Inc.

319, 323, 324, 325, 330, 331, 332, 337, 339, 340, 341, 342, 343, 345, 353, 363, 365, 366, 367
month <integer> · 79, 108, 342
month <string> · 80, 109, 342
month and year · 70, 74, 75, 77, 78, 81, 84, 85, 86, 89, 97, 319, 323, 325, 330, 331, 332, 337, 339, 341, 342, 343, 345, 353, 363, 366
month of <date> · 69, 80, 342
month of <day of year> · 77, 80, 342
month of <month and year> · 80, 85, 342
month_and_year of <date> · 70, 84, 343
more significance <integer> of <floating point> · 20, 23, 343
most significant one bit of <bit set> · 46, 343
multicast support of <network ip interface> · 275, 343
multiplicity of <integer with multiplicity> · 17, 343
multiplicity of <string with multiplicity> · 41, 343
multivalued of <property> · 310, 343

N

name of <application usage summary> · 175, 343
name of <binary operator> · 311, 343
name of <cast> · 314, 343
name of <connection> · 285, 343
name of <current user> · 267, 343
name of <drive> · 153, 343
name of <environment variable> · 249, 343
name of <filesystem object> · 136, 343
name of <firewall authorized application> · 196, 343
name of <firewall open port> · 205, 343
name of <firewall rule> · 216, 343
name of <firewall service> · 206, 343
name of <fixlet_header> · 231, 343
name of <local group> · 239, 343
name of <local user> · 265, 343
name of <logged on user> · 269, 343
name of <metabase key> · 302, 344
name of <network adapter> · 280, 344
name of <network share> · 277, 344
name of <operating system> · 179, 344
name of <port mapping> · 284, 344
name of <registry key value> · 131, 344
name of <registry key> · 127, 344
name of <selected server> · 236, 344
name of <setting> · 235, 344
name of <site> · 227, 344
name of <type> · 308, 344
name of <unary operator> · 313, 344
name of <wmi select> · 224, 344
nan of <floating point> · 7, 23, 344
native registry · 109, 120, 122, 344
network · 1, 2, 95, 96, 103, 107, 108, 109, 117, 121, 194, 197, 198, 202, 203, 208, 213, 217, 260, 261, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 285, 287, 288, 289, 290, 291, 292, 293, 295, 317, 319, 320, 321, 322, 323, 325, 326, 330, 332, 333, 336,聊337, 338, 339, 340, 341, 343, 344, 347, 348, 350, 352, 353, 356, 359, 360, 361, 362
network adapter · 273, 278, 279, 280, 281, 282, 291, 292, 317, 325, 326, 332, 333, 336, 337, 338, 339, 340, 341, 344, 348, 352, 356, 359, 361
network address list · 273, 279, 282, 292, 317, 326, 333, 356
network interface · 96, 273, 274, 287, 288, 292, 330, 336
network ip interface · 273, 275, 317, 320, 336, 340, 343, 348, 356
network share · 109, 121, 260, 276, 277, 278, 319, 321, 323, 325, 330, 344, 347, 350, 353, 359, 360, 362
network share <string> · 109, 276, 344
Networking Objects · iv, 273
next line of <file line> · 166, 167, 344
next sibling of <xml dom node> · 169, 171, 344
no password required flag of <local user> · 265, 344
node name of <xml dom node> · 171, 344
node type of <xml dom node> · 171, 344
node value of <xml dom node> · 171, 345
none firewall service type · 109, 208, 345

© 2007 by BigFix, Inc.

noon · 56, 109, 345
normal account flag of <local user> · 265, 345
normal of <filesystem object> · 136, 345
normal of <floating point> · 7, 23, 345
notifications disabled of <firewall profile> · 199, 345
november · 69, 77, 80, 84, 109, 345
november <integer> · 69, 77, 109, 345
november <integer> of <integer> · 69, 345
november of <integer> · 84, 345
now · 53, 56, 61, 64, 67, 71, 82, 92, 109, 180, 228, 345
nt domain controller product type · 109, 237, 345
nt server product type · 109, 237, 345
nt workstation product type · 110, 237, 345
number of months · 14, 15, 70, 78, 81, 82, 85, 86, 87, 88, 89, 108, 120, 342, 362, 366
numeric type of <drive> · 153, 345
numeric value of <string> · 11, 33, 345

O

october · 69, 77, 80, 84, 110, 345
october <integer> · 69, 77, 110, 345
october <integer> of <integer> · 69, 345
october of <integer> · 84, 345
oem code page · 110, 345
offer accepted of <action> · 271, 345
offer of <action> · 271, 345
offline of <filesystem object> · 136, 346
ok firewall local policy modify state · 110, 212, 346
oldest record number of <event log> · 241, 346
one bit of <bit set> · 46, 346
only raw version block of <file> · 140, 161, 346
only version block of <file> · 140, 161, 346
operand type of <cast> · 307, 314, 346
operand type of <unary operator> · 307, 313, 346
operating system · 3, 4, 5, 35, 93, 99, 103, 105, 109, 110, 118, 127, 128, 138, 146, 149, 176, 177, 178, 179, 180, 181, 195, 217, 237, 238, 242, 245, 268, 320, 323, 334, 340, 342, 344, 345, 346, 347, 348, 349, 351, 353, 356, 360, 362, 366
operating system product type · 109, 110, 179, 237, 345, 346, 349
operating system product type <integer> · 110, 237, 346
operating system suite mask · 93, 179, 356
options of <port mapping> · 284, 346
organization of <license> · 188, 346
origin fixlet id of <action> · 271, 346
outbound connections allowed of <firewall profile> · 199, 346
outbound of <firewall rule> · 216, 346
overflow of <floating point> · 7, 23, 346
owner document of <xml dom node> · 168, 171, 346
owner of <security descriptor> · 260, 261, 346

P

pad of <version> · 158, 159, 346
parameter <string> · 29, 110, 271, 346
parameter <string> of <action> · 29, 110, 271, 346
parent folder of <filesystem object> · 136, 149, 346
parent node of <xml dom node> · 169, 171, 347
parent of <type> · 307, 308, 347
parenthesized part <integer> of <regular expression match> · 49, 347
parenthesized part of <regular expression match> · 49, 347
password age of <local user> · 266, 347
password change disabled flag of <local user> · 266, 347
password expiration disabled flag of <local user> · 266, 347
password expired of <local user> · 266, 347
password of <network share> · 277, 347
path of <network share> · 277, 347
pathname of <file shortcut> · 155, 347
pathname of <filesystem object> · 136, 347
pending login · 110, 271, 347

© 2007 by BigFix, Inc.

pending login of <action> · 271, 347
pending of <action> · 271, 347
pending restart · 110, 111, 272, 347
pending restart <string> · 111, 347
pending restart of <action> · 272, 347
pending time of <action> · 272, 347
performance counter frequency of <operating system> · 179, 347
performance counter of <operating system> · 179, 347
permission permission of <network share> · 277, 347
personal bit <operating system suite mask> · 111, 347
physical processor count · 106, 111, 121, 347
platform id of <operating system> · 179, 348
plural name of <property> · 310, 348
point to point of <network ip interface> · 276, 348
port mapping · 283, 284, 292, 329, 330, 336, 344, 346, 348, 349, 357
port mapping of <internet connection firewall> · 283, 348
port number of <selected server> · 236, 348
port of <firewall open port> · 205, 348
position <integer> of <string> · 33, 37, 348
position of <string> · 33, 37, 348
preceding text of <string position> · 37, 39, 348
preceding text of <substring> · 39, 40, 348
previous line of <file line> · 166, 167, 348
previous sibling of <xml dom node> · 170, 171, 348
primary group id of <local user> · 266, 348
primary language · 192, 193, 348, 366
primary language of <language> · 192, 193, 348
primary wins server of <network adapter> · 280, 292, 348
Primitive Objects · ii, 1, 6
print operator flag of <local user> · 266, 348
priority of <selected server> · 236, 348
private firewall profile type · 111, 201, 348
private profile of <firewall policy> · 198, 202, 348
problem id of <active device> · 186, 348
process image file name of <firewall authorized application> · 197, 348
processor · 50, 51, 52, 103, 106, 111, 117, 181, 182, 183, 320, 330, 340, 342, 348, 349, 355, 356, 359, 361, 371, 372
processor <integer> · 111, 181, 349
product of <integer> · 11, 13, 349
product type of <operating system> · 179, 237, 349
product version of <file> · 140, 159, 349
profile <firewall profile type> of <firewall rule> · 216, 349
profile folder of <local user> · 266, 349
property · 111, 115, 126, 130, 131, 133, 135, 140, 148, 155, 158, 159, 163, 170, 172, 196, 197, 198, 199, 204, 205, 206, 207, 209, 210, 211, 213, 214, 215, 216, 217, 218, 222, 225, 227, 228, 230, 231, 233, 234, 260, 276, 278, 280, 304, 307, 308, 309, 310, 311, 325, 335, 343, 348, 349, 351, 354, 360, 366, 371
property <string> · 111, 222, 225, 308, 309, 349
property <string> of <type> · 308, 309, 349
property <string> of <wmi object> · 222, 225, 349
property of <type> · 308, 309, 349
property of <wmi object> · 222, 225, 349
property returning <type> · 111, 308, 310, 349
property returning <type> of <type> · 308, 310, 349
protocol of <firewall open port> · 205, 218, 349
protocol of <firewall rule> · 216, 218, 349
protocol of <port mapping> · 284, 349
public firewall profile type · 111, 201, 349
public profile of <firewall policy> · 198, 202, 349

Q

query value permission of <access control entry> · 252, 349

© 2007 by BigFix, Inc.

R

ram · 112, 184, 349, 355
random access memory · 112, 184, 349
range after <time> of <time range> · 62, 63, 349
range before <time> of <time range> · 62, 63, 349
rate · 179
raw file version of <file> · 140, 159, 349
raw product version of <file> · 141, 159, 349
raw version block <integer> of <file> · 141, 161, 350
raw version block <string> of <file> · 141, 161, 350
raw version block of <file> · 141, 161, 350
raw version of <file> · 141, 159, 350
read attributes permission of <access control entry> · 252, 350
read control permission of <access control entry> · 252, 350
read extended attributes permission of <access control entry> · 252, 350
read permission of <access control entry> · 253, 350
read permission of <network share> · 277, 350
readonly of <filesystem object> · 136, 350
recent application · 112, 145, 350
recent application <string> · 112, 145, 350
record <integer> of <event log> · 241, 242, 350
record count of <event log> · 241, 350
record number of <event log record> · 243, 350
record of <event log> · 241, 242, 350
reference attribute of <metabase value> · 305, 350
regapp · 36, 40, 112, 143, 145, 146, 147, 154, 160, 161, 165, 350
regapp <string> · 112, 145, 147, 350
regex <string> · 47, 112, 350
registrar number of <license> · 188, 350
registration server · 92
registry · 1, 14, 32, 106, 109, 110, 112, 119, 120, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 137, 143, 144, 145, 146, 147, 148, 156, 185, 250, 251, 252, 253, 260, 268, 302, 303, 305, 318, 324, 325, 326, 330, 338, 344, 350, 353, 355, 359, 360, 362, 366, 367, 370
registry key · 14, 32, 123, 125, 126, 127, 130, 131, 132, 133, 134, 137, 144, 145, 147, 148, 185, 250, 251, 252, 253, 260, 302, 318, 324, 325, 326, 330, 338, 344, 353, 355, 359, 360, 366, 367
registry key value · 14, 32, 126, 127, 130, 131, 132, 133, 134, 137, 144, 147, 325, 344, 355, 359, 360, 366, 367
registry key value type · 14, 131, 133, 134, 359, 366
Registry Objects · ii, 1, 122
regular expression · 47, 48, 49, 94, 112, 320, 332, 341, 347, 350
regular expression <string> · 47, 112, 350
regular expression match · 49, 332, 341, 347
relative significance place <integer> of <floating point> · 20, 23, 351
relative significance place of <floating point> · 21, 23, 351
relay service · 112, 189, 351
release of <operating system> · 179, 351
Relevance Language · 3
relevance of <fixlet> · 231, 351
relevant fixlet of <site> · 227, 230, 351
remote addresses of <firewall authorized application> · 197, 351
remote addresses of <firewall open port> · 205, 351
remote addresses of <firewall service> · 206, 351
remote addresses string of <firewall rule> · 216, 351
remote admin settings of <firewall profile> · 199, 210, 351
remote desktop firewall service type · 112, 208, 351
remote of <logged on user> · 269, 351
remote ports string of <firewall rule> · 217, 351
result type of <binary operator> · 307, 312, 351

© 2007 by BigFix, Inc.

result type of <property> · 307, 310, 351
result type of <unary operator> · 307, 313, 351
right operand type of <binary operator> · 308, 312, 351
right shift <integer> of <bit set> · 45, 46, 351
root folder of <drive> · 149, 153, 351
root server · 97, 113, 159, 246, 247, 334, 335, 351
rope · 11, 34, 44, 113, 339, 351, 367
rope <string> · 44, 113, 351
rsop computer wmi · 113, 221, 351
rsop user wmi <security identifier> · 113, 221, 352
rule group currently enabled <string> of <firewall> · 195, 352
rule group enabled <string> of <firewall profile> · 199, 352
rule of <firewall service restriction> · 213, 217, 352
rule of <firewall> · 195, 213, 352
running application · 113, 145, 147, 352
running application <string> · 113, 145, 352
running of <application usage summary> · 175, 352
running of <local mssql database> · 189, 352
running service · 113, 189, 191, 352
running service <string> · 113, 189, 352

S

saturday · 72, 113, 352
scope of <firewall authorized application> · 197, 204, 352
scope of <firewall open port> · 204, 205, 352
scope of <firewall service> · 204, 207, 352
script flag of <local user> · 266, 352
seat count state of <license> · 188, 352
seat of <license> · 188, 352
second · 19, 29, 34, 39, 42, 43, 57, 60, 65, 81, 82, 96, 113, 148, 167, 173, 179, 280, 352
second_of_minute of <time of day with time zone> · 60, 352
second_of_minute of <time of day> · 57, 352
secondary wins server of <network adapter> · 280, 292, 352
section <string> of <file> · 141, 156, 352
secure attribute of <metabase value> · 305, 353
security descriptor · 127, 141, 151, 254, 258, 259, 260, 261, 268, 269, 276, 278, 323, 324, 334, 346, 353, 367
security descriptor of <file> · 141, 259, 353
security descriptor of <folder> · 151, 259, 353
security descriptor of <network share> · 260, 278, 353
security descriptor of <registry key> · 127, 260, 353
security event log · 113, 240, 353
security identifier · 113, 221, 240, 243, 250, 253, 260, 261, 262, 269, 316, 326, 334, 346, 354, 358, 360, 367
select <string> of <wmi> · 222, 353
select <string> of <xml dom node> · 170, 172, 353
select object <string> of <wmi> · 222, 225, 353
selected server · 114, 235, 236, 291, 321, 325, 333, 336, 344, 348, 353, 361
september · 69, 77, 80, 84, 114, 353
september <integer> · 69, 77, 114, 353
september <integer> of <integer> · 69, 353
september of <integer> · 84, 353
server operator flag of <local user> · 266, 353
server trust account flag of <local user> · 266, 353
service · 100, 101, 106, 112, 113, 114, 115, 138, 178, 179, 186, 189, 190, 191, 195, 199, 206, 207, 208, 217, 218, 240, 243, 245, 297, 320, 321, 325, 330, 335, 340, 351, 352, 353, 354, 355, 361, 367
service <string> · 114, 190, 353
service key value name of <active device> · 186, 353
service name of <firewall rule> · 217, 353
service name of <service> · 190, 353
service of <firewall profile> · 199, 206, 353
service pack major version of <operating system> · 179, 353
service pack minor version of <operating system> · 179, 353

© 2007 by BigFix, Inc.

service restricted <(string, string)> of <firewall service restriction> · 218, 354
service restriction of <firewall> · 195, 217, 354
service specific exit code of <service> · 190, 354
set of <integer> · 13, 18, 354
set of <string> · 33, 354
set value permission of <access control entry> · 253, 354
setting · 32, 92, 94, 174, 194, 198, 199, 200, 211, 212, 227, 233, 234, 235, 257, 317, 327, 344, 354, 360, 367
setting <string> of <client> · 233, 234, 354
setting <string> of <site> · 227, 234, 354
setting of <client> · 234, 354
setting of <site> · 227, 234, 354
sha1 of <file> · 141, 354
shortcut of <file> · 141, 155, 354
sid of <local group member> · 240, 261, 354
sid of <logged on user> · 261, 269, 354
significance place <integer> of <floating point> · 21, 23, 354
significance place of <floating point> · 21, 23, 354
significance threshold of <floating point> · 21, 24, 354
significant digits <integer> of <hertz> · 50, 51, 354
significant digits <integer> of <integer> · 11, 13, 354
single user ts bit <operating system suite mask> · 114, 354
singular name of <property> · 310, 354
site · 31, 97, 98, 114, 116, 138, 148, 149, 187, 188, 194, 196, 197, 200, 202, 203, 204, 206, 207, 209, 210, 218, 219, 226, 227, 228, 229, 230, 231, 232, 234, 235, 321, 323, 324, 332, 333, 334, 335, 338, 341, 342, 344, 351, 354, 355, 356, 359, 360, 361, 367, 371
site <string> · 114, 226, 354
site group · 227, 229, 334, 335, 342
site number of <license> · 188, 354
Site Objects · iv, 2, 226
site tag of <site> · 227, 355
site version list · 31, 114, 227, 229, 230, 321, 355, 367
site version list <string> · 114, 229, 355
site version list of <site> · 227, 229, 355
size of <file> · 141, 355
size of <integer set> · 18, 355
size of <ram> · 184, 355
size of <registry key value> · 131, 355
size of <string set> · 42, 355
size of <type> · 308, 355
small business bit <operating system suite mask> · 114, 355
small business restricted bit <operating system suite mask> · 114, 355
source of <event log record> · 243, 355
speed of <processor> · 182, 355
standard deviation of <floating point> · 21, 24, 355
standard deviation of <integer> · 13, 21, 355
standard firewall profile type · 115, 201, 355
standard profile of <firewall policy> · 198, 203, 355
start date of <license> · 188, 355
start in pathname of <file shortcut> · 155, 355
start of <substring> · 37, 40, 355
start of <time range> · 63, 355
start type of <service> · 191, 355
state of <service> · 191, 355
status of <action> · 272, 355
status of <active device> · 186, 355
status of <connection> · 285, 288, 356
status of <network adapter> · 281, 356
stepping of <processor> · 182, 356
string · 6, 7, 10, 11, 12, 15, 20, 21, 22, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 47, 48, 49, 51, 52, 53, 55, 56, 57, 58, 59, 60, 62, 64, 65, 66, 68, 69, 71, 72, 73, 74, 77, 79, 80, 84, 87, 88, 89, 92, 93, 94, 95, 98, 99, 100, 101, 102, 103, 104, 109, 110, 112, 113, 115, 116, 117, 118, 119, 120, 121, 124, 126, 127, 129, 130, 131, 132, 133, 134, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 150, 151, 152, 153, 154, 155, 156, 158, 159, 160, 161, 162, 163, 165, 166, 167, 168, 170, 171, 172, 174,

© 2007 by BigFix, Inc.

175, 176, 177, 178, 179, 181, 182, 183, 185, 186, 187, 188, 189, 190, 191, 192, 193, 195, 196, 197, 198, 205, 206, 214, 215, 216, 217, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 233, 234, 235, 236, 239, 240, 242, 243, 247, 249, 254, 255, 256, 257, 260, 262, 264, 265, 266, 267, 269, 271, 272, 277, 279, 280, 284, 285, 292, 293, 294, 295, 296, 298, 302, 303, 304, 305, 308, 309, 310, 311, 312, 313, 314, 315, 316, 318, 319, 320, 321, 322, 323, 324, 325, 326, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 343, 344, 345, 346, 347, 348, 349, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 364, 365, 366, 367, 368, 369, 370, 372
string <string> · 29, 115, 356
string position · 27, 33, 37, 40, 329, 332, 348, 355
string set · 33, 34, 42, 43, 328, 336, 354, 355, 359
string value <integer> of <wmi select> · 224, 356
string value of <wmi select> · 224, 356
string with multiplicity · 34, 343, 359
subnet address of <network adapter> · 281, 292, 356
subnet address of <network address list> · 282, 292, 356
subnet address of <network ip interface> · 276, 292, 356
subnet mask of <network adapter> · 281, 292, 356
subnet mask of <network address list> · 282, 292, 356
subnet mask of <network ip interface> · 276, 292, 356
subscribe time of <site> · 227, 356
substring · 27, 32, 33, 34, 37, 38, 39, 40, 48, 49, 321, 329, 331, 332, 338, 347, 348, 355, 356
substring <string> of <string> · 34, 39, 356
substring after <string> of <string> · 34, 39, 356
substring before <string> of <string> · 34, 39, 356
substring between <string> of <string> · 34, 39, 356
substring separated by <string> of <string> · 34, 39, 356
suite mask of <operating system> · 179, 238, 356
sum of <integer> · 11, 13, 356
sunday · 72, 115, 356
symbol of <binary operator> · 312, 356
symbol of <unary operator> · 313, 357
synchronize permission of <access control entry> · 253, 357
system event log · 115, 240, 241, 357
system file <string> · 115, 138, 357
system folder · 370
system ini device file · 115, 138, 357
system ini device file <string> · 115, 138, 357
system language · 115, 357
system locale · 115, 192, 193, 357
System Objects · iii, 1, 2, 176
system of <filesystem object> · 136, 357
system ui language · 115, 192, 357
system wow64 folder · 116, 149, 357
system x32 folder · 116, 149, 357
system x64 folder · 116, 149, 357

T

target ip address of <port mapping> · 284, 292, 357
target name of <port mapping> · 284, 357
tcp · 116, 218, 284, 357
temporary duplicate account flag of <local user> · 266, 357
temporary of <filesystem object> · 136, 357
terminal bit <operating system suite mask> · 116, 357
thursday · 72, 116, 357
time · 8, 9, 14, 15, 28, 29, 30, 31, 32, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 70, 71, 72, 73, 74, 75, 77, 78, 85, 86, 89, 92, 97, 98, 102, 103, 105, 106, 108, 109, 113, 116, 117, 118, 120, 131, 135, 136, 137, 139, 142, 143, 144,

© 2007 by BigFix, Inc.

174, 175, 177, 179, 180, 187, 188, 224, 226, 227, 228, 235, 243, 244, 263, 265, 266, 270, 271, 272, 277, 280, 316, 317, 318, 320, 321, 323, 324, 327, 329, 330, 331, 332, 333, 334, 335, 338, 339, 340, 341, 342, 345, 347, 349, 352, 355, 356, 357, 358, 359, 360, 361, 363, 367, 368
time <string> · 53, 116, 357
time <time zone> of <time> · 54, 59, 357
time generated of <event log record> · 243, 357
time interval · 8, 9, 14, 15, 28, 31, 52, 54, 57, 59, 61, 63, 64, 65, 66, 67, 70, 71, 72, 73, 74, 75, 77, 78, 85, 86, 89, 98, 102, 103, 105, 108, 113, 116, 118, 175, 179, 226, 266, 316, 324, 333, 334, 338, 339, 341, 342, 347, 352, 357, 358, 360, 361, 367, 368
time interval <string> · 65, 116, 357
time of <time of day with time zone> · 56, 60, 358
time of day · 29, 30, 31, 32, 54, 56, 57, 58, 59, 60, 61, 67, 70, 97, 108, 109, 116, 120, 324, 334, 342, 345, 352, 357, 358, 363, 367, 368
time of day with time zone · 30, 31, 32, 54, 57, 58, 59, 60, 61, 67, 70, 97, 120, 324, 334, 342, 352, 357, 358, 363, 367, 368
time range · 8, 9, 14, 15, 54, 56, 62, 63, 64, 66, 67, 329, 331, 335, 339, 349, 355, 368
time value <integer> of <wmi select> · 224, 358
time value of <wmi select> · 224, 358
time written of <event log record> · 243, 358
time zone · 28, 31, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 66, 67, 68, 70, 97, 106, 116, 117, 340, 358, 359, 363, 368
time zone <string> · 58, 116, 358
time_of_day <string> · 56, 116, 358
total duration of <application usage summary> · 175, 358
total processor core count · 117, 358
total run count of <application usage summary> · 175, 358
total space of <drive> · 154, 358
traverse permission of <access control entry> · 253, 358
true · 6, 7, 8, 36, 40, 46, 47, 82, 111, 117, 214, 358
trustee of <access control entry> · 253, 261, 358
trustee type of <access control entry> · 253, 358
tuesday · 72, 117, 358
two digit hour of <time of day with time zone> · 29, 60, 358
two digit hour of <time of day> · 57, 358
two digit minute of <time of day with time zone> · 29, 60, 358
two digit minute of <time of day> · 57, 358
two digit second of <time of day with time zone> · 29, 60, 358
two digit second of <time of day> · 57, 358
type · 2, 3, 5, 6, 10, 12, 14, 20, 27, 30, 31, 49, 56, 57, 58, 59, 68, 69, 71, 73, 74, 76, 77, 78, 79, 80, 81, 85, 87, 88, 89, 91, 92, 93, 94, 97, 98, 100, 101, 102, 103, 104, 105, 107, 108, 109, 110, 111, 112, 114, 116, 117, 120, 123, 124, 125, 126, 128, 129,聊130, 131, 132, 133, 134, 144, 153, 154, 155, 159, 171, 179, 181, 182, 194, 198, 200, 201, 202, 203, 207, 208, 211, 212, 214, 218, 219, 224, 225, 228, 235, 237, 242, 245, 246, 253, 258, 265, 274, 275, 278, 281, 282, 289, 291, 294, 298, 299, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 320, 325, 335, 338, 344, 346, 347, 349, 351, 355, 359, 364, 368
type <string> · 308
type of <distinguished name component> · 298, 359
type of <drive> · 154, 359
type of <firewall profile> · 200, 201, 359
type of <firewall service> · 207, 208, 359
type of <metabase value> · 303, 305, 359
type of <network adapter> · 281, 359
type of <network share> · 278, 359
type of <processor> · 182, 359
type of <registry key value> · 131, 133, 359
type of <site> · 228, 359
type of <wmi select> · 224, 359

© 2007 by BigFix, Inc.

U

udp · 117, 207, 218, 219, 284, 359
unary operator · 117, 312, 313, 344, 346, 351, 357, 359, 368
unary operator <string> · 117, 312, 359
unary operator returning <type> · 117, 312, 359
undefined · 49, 100, 329
underflow of <floating point> · 7, 24, 359
unicast responses to multicast broadcast disabled of <firewall profile> · 200, 359
union of <integer set> · 18, 359
union of <string set> · 42, 359
unique value of <integer> · 13, 17, 359
unique value of <string> · 34, 41, 359
universal time <string> · 53, 117, 359
universal time zone · 52, 53, 58, 61, 117, 359
upnp firewall service type · 117, 208, 360
upper bound of <integer range> · 11, 16, 360
uptime of <operating system> · 179, 360
url of <site> · 228, 360
use count of <network share> · 278, 360
use limit of <network share> · 278, 360
user comment of <local user> · 266, 360
user id of <local user> · 266, 360
user language · 117, 360
user locale · 118, 192, 360
User Objects · iv, 2, 263
user privilege of <local user> · 267, 360
user sid of <event log record> · 243, 261, 360
user type of <metabase value> · 304, 306, 360
user ui language · 118, 192, 360
usual name of <property> · 310, 360

V

value <string> of <file version block> · 162, 360
value <string> of <registry key> · 127, 130, 360
value of <distinguished name component> · 298, 360
value of <environment variable> · 249, 360
value of <fixlet_header> · 231, 360
value of <metabase key> · 302, 305, 360
value of <registry key> · 127, 130, 360
value of <setting> · 235, 360
variable <string> of <environment> · 248, 249, 360
variable of <environment> · 248, 249, 361
variable of <file> · 142, 361
vendor name of <processor> · 183, 361
version · 1, 2, 3, 4, 28, 30, 31, 32, 93, 94, 97, 98, 99, 100, 104, 111, 114, 116, 118, 120, 122, 135, 137, 139, 140, 141, 142, 143, 144, 157, 158, 159, 160, 161, 162, 163, 165, 168, 172, 174, 176, 177, 178, 179, 180, 182, 206, 219, 220, 228, 229, 230, 238, 246, 291, 294, 331, 332, 346, 349, 350, 361, 368
version <string> · 118, 159, 361
version block <integer> of <file> · 142, 162, 361
version block <string> of <file> · 142, 162, 361
version block of <file> · 142, 162, 361
version of <bios> · 176, 361
version of <current relay> · 159, 246, 361
version of <file> · 142, 159, 361
version of <site> · 228, 361
version string <string> of <module> · 361
volatile attribute of <metabase value> · 306, 361

W

waiting for download of <action> · 272, 361
wakeonlan enabled of <network adapter> · 281, 361
warning event log event type · 118, 245, 361
wednesday · 72, 118, 361
week · 9, 30, 35, 55, 64, 65, 68, 69, 71, 72, 73, 84, 96, 98, 118, 180, 323, 324, 361, 367
weight of <selected server> · 236, 361
win32 exit code of <service> · 191, 361
windows display time <string> · 53, 118, 361
windows file <string> · 118, 138, 361
windows folder · 56, 118, 143, 146, 149, 151, 156, 160, 161, 259, 361
wins enabled of <network adapter> · 281,

© 2007 by BigFix, Inc.

361
winsock2 supported of <network> · 274, 361
wmi · 102, 113, 118, 221, 222, 223, 224, 225, 320, 333, 336, 344, 349, 351, 352, 353, 356, 358, 359, 361, 362, 368
wmi <string> · 118, 221, 362
wmi object · 102, 118, 221, 222, 225, 349, 353, 368
WMI Objects · iv, 2, 221
wmi select · 222, 223, 224, 225, 320, 336, 344, 349, 353, 356, 358, 359, 368
workstation trust account flag of <local user> · 267, 362
world · 1, 91, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363
World Objects · ii, 1, 91
write attributes permission of <access control entry> · 253, 362
write dac permission of <access control entry> · 253, 362
write extended attributes permission of <access control entry> · 253, 362
write owner permission of <access control entry> · 253, 362
write permission of <access control entry> · 253, 362
write permission of <network share> · 278, 362

X

x32 application <string> · 119, 145, 362
x32 file <string> · 119, 139, 362
x32 folder <string> · 119, 149, 362
x32 registry · 119, 122, 362
x64 application <string> · 119, 146, 362
x64 file <string> · 119, 139, 362
x64 folder <string> · 119, 150, 362
x64 of <operating system> · 180, 362
x64 registry · 120, 122, 362
xml document of <file> · 142, 168, 362
xml document of <string> · 168, 362
xml dom document · 142, 169, 170, 171, 172, 346, 362
xml dom node · 170, 171, 172, 173, 319, 321, 331, 338, 344, 345, 346, 347, 348, 353, 362, 368
xpath <(string, string)> of <xml dom node> · 170, 172, 362
xpath <string> of <xml dom node> · 170, 172, 362

Y

year · 3, 9, 12, 32, 35, 55, 68, 69, 70, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 97, 120, 177, 244, 323, 324, 338, 339, 362, 363, 365, 368, 369
year <integer> · 88, 120, 362
year <string> · 88, 120, 362
year of <date> · 70, 88, 363
year of <month and year> · 85, 88, 363

Z

zone of <time of day with time zone> · 60, 363
zoned time_of_day <string> · 60, 120, 363

© 2007 by BigFix, Inc.