

# Core JavaScript Classes

# Contents

## JavaScript Reference    [JavaScript Guide](#)

Scripting is one of the most powerful features in the CS3 suite. It can save lots of time: almost anything you can do with the user interface, you can do with scripts, and there are even a few things you can do in scripting that you cannot do from within the user interface. The contents on the right is categorized by function; in the index at the bottom, you can find all objects in alphabetical order. Each object description lists its elements, properties, and methods.

[Contents :: Index](#)

# Core JavaScript Classes

## Contents

## JavaScript

JavaScript	\$
	Array
	Boolean
	Date
	Dictionary
	Error
	File
	Folder
	Function
	Math
	Namespace
	Number
	Object
	QName
	Reflection
	ReflectionInfo
	RegExp
	Socket
	String
	UnitValue
	XML

[Contents :: Index](#)

Class

# \$

The \$ object provides a number of debugging facilities and informational methods.

**QuickLinks** [about](#), [bp](#), [clearbp](#), [colorPicker](#), [evalFile](#), [gc](#), [getenv](#), [setbp](#), [sleep](#), [toString](#), [write](#), [writeln](#)

## Properties

Property	Type	Access	Description
build	number	readonly	The ExtendScript build number.
buildDate	Date	readonly	The ExtendScript build date.
engineName	string	readonly	The name of the current ExtendScript engine, if set.
error	Error	r/w	<p>The most recent run-time error information.</p> <p>Assigning error text to this property generates a run-time error; however, the preferred way to generate a run-time error is to throw an <a href="#">Error</a> object.</p>
fileName	string	readonly	The file name of the current script.
flags	number	r/w	<p>Gets or sets low-level debug output flags.</p> <p>A logical AND of bit flag values: 0x0002 (2): Displays each line with its line number as it is executed. 0x0040 (64): Enables excessive garbage collection. Usually, garbage</p>

collection starts when the number of objects has increased by a certain amount since the last garbage collection. This flag causes ExtendScript to garbage collect after almost every statement. This impairs performance severely, but is useful when you suspect that an object gets released too soon.

0x0080 (128): Displays all calls with their arguments and the return value.

0x0100 (256): Enables extended error handling (see strict). 0x0200 (512): Enables the localization feature of the `toString` method. Equivalent to the `localize` property.

global	Object	readonly	A reference to the global object, which contains the JavaScript global namespace.
--------	--------	----------	---

includePath	string	readonly	The path for include files for the current script.
-------------	--------	----------	--

level	number	r/w	The current debugging level, which enables or disables the JavaScript debugger.  One of 0 (no debugging), 1 (break on runtime errors), or 2 (full debug mode).
-------	--------	-----	--

locale	string	r/w	<p>Gets or sets the current locale.</p> <p>The string contains five characters in the form LL_RR, where LL is an ISO 639 language specifier, and RR is an ISO 3166 region specifier. Initially, this is the value that the application or the platform returns for the current user. You can set it to temporarily change the locale for testing. To return to the application or platform setting, set to undefined, null, or the empty string.</p>
localize	bool	r/w	Set to true to enable the extended localization features of the built-in <code>toString()</code> method.
memCache	number	r/w	The ExtendScript memory cache size, in bytes.
os	string	readonly	The current operating system version information.
screens	Array of <a href="#">Object</a>	readonly	<p>An array of objects containing information about the display screens attached to your computer.</p> <p>Each object has the properties left, top, right, bottom, which contain the four corners of each screen in global coordinates. A property primary is true if that object describes the</p>

			primary display.						
stack	string	readonly	The current stack trace.						
strict	any	r/w	Sets or clears strict mode for object modification.  When true, any attempt to write to a read-only property causes a runtime error. Some objects do not permit the creation of new properties when true.						
version	string	readonly	The version number of the ExtendScript engine.  Formatted as a three-part number and description; for example: "3.7.50 (debug)".						
<b>Methods</b>		<b>string about ()</b> Shows an About box for the ExtendScript component, and returns the text for the box.							
<b>void bp (condition:any)</b> Breaks execution at the current position.		<table border="1"> <thead> <tr> <th>Parameter</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>condition</td><td>any</td><td>A string containing a JavaScript statement to be used as a condition. If the statement evaluates to true or nonzero when this point is reached, execution stops.</td></tr> </tbody> </table>		Parameter	Type	Description	condition	any	A string containing a JavaScript statement to be used as a condition. If the statement evaluates to true or nonzero when this point is reached, execution stops.
Parameter	Type	Description							
condition	any	A string containing a JavaScript statement to be used as a condition. If the statement evaluates to true or nonzero when this point is reached, execution stops.							
<b>void clearbp ( [line:number=0])</b> Clears a breakpoint.		<table border="1"> <thead> <tr> <th>Parameter</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>line</td><td>number</td><td>The line number. (default: 0)</td></tr> </tbody> </table>		Parameter	Type	Description	line	number	The line number. (default: 0)
Parameter	Type	Description							
line	number	The line number. (default: 0)							
<b>number colorPicker (color:number)</b> Invokes the platform-specific color selection dialog, and returns									

the selected color.

Parameter	Type	Description
color	number	The color to be preselected in the dialog, as 0xRRGGBB, or -1 for the platform default.

**any evalFile (file:File [, timeout:number=10000])**

Loads and evaluates a file.

Parameter	Type	Description
file	File	The file to load.
timeout	number	An optional timeout in milliseconds. (default: <b>10000</b> )

**void gc ()**

Initiates garbage collection in the ExtendScript engine.

**string getenv (name:string)**

Retrieves the value of an environment variable.

Parameter	Type	Description
name	string	The name of the variable.

**void setbp (line:number, condition:string)**

Sets a breakpoint in the current script.

Parameter	Type	Description
line	number	The line at which to stop execution.
condition	string	

**void sleep (msecs:number)**

Suspends the calling thread for a number of milliseconds.

Parameter	Type	Description
msecs	number	Number of milliseconds to sleep.

**string toString ()**

Converts this object to a string.

**void write (text:any)**

Prints text to the Console.

Parameter	Type	Description
-----------	------	-------------

text	any	The text to print. All arguments are concatenated.
------	-----	--

**void writeln (text:any)**

Prints text to the Console, and adds a newline character.

Parameter	Type	Description
-----------	------	-------------

text	any	The text to print. All arguments are concatenated.
------	-----	--

[Contents :: Index](#)

Class

# Array

The Array class is an array with integer indexes and a length property.

QuickLinks [Array](#), [concat](#), [join](#), [pop](#), [push](#), [reverse](#), [shift](#), [slice](#), [sort](#), [splice](#), [toLocaleString](#), [toSource](#), [toString](#), [unshift](#)

## Properties

Property	Type	Access	Description
length	number	r/w	The length of the array

## Methods

**Array** **Array** (**length:number**)

Creates and returns a new array.

Parameter	Type	Description
length	number	

**Array** **concat** (**value:any**)

Returns a new array created by concatenating the given values to the end of the original array.

Parameter	Type	Description
value	any	Any number of values to be added to the end of the array.

**string** **join** (**delimiter:string**)

Joins all elements of the array into a string; optionally, each element is separated by delimiter.

Parameter	Type	Description
delimiter	string	A string used to separate each element of the array.

**any** **pop** ()

Removes the last element from the array, decreases the length by 1, and returns the value of the element.

**number** **push** (**value:number**)

Places one or more values onto the end of the array and increases length by n.

Parameter	Type	Description
value	number	Any number of values to be pushed onto the end of the array.

### [Array reverse \(\)](#)

Reverses the order of the elements in the array.

### [any shift \(\)](#)

Removes the first element from the array, decreases the length by 1, and returns the value of the element.

### [Array slice \(\)](#)

Creates a new array, which contains a subset of the original array's elements.

### [void sort \(userFunction:Function\)](#)

Sorts the elements of the array in place, using the given function to compare to elements.

Parameter	Type	Description
userFunction	Function	A user-supplied function of the form userFunction(a, b) which returns less than 0 if a is greater than b, 0 if a and b are equal, and greater than 0 if b is greater than a.

### [Array splice \(start:number, num:number, value:any\)](#)

Removes num elements from the array beginning with index, start.

Parameter	Type	Description
start	number	The index of the first element to remove. Negative values are relative to the end of the array.
num	number	The number of array elements to remove, including start. If omitted, all elements from array index start to the end of the array are removed.
value	any	A list of one or more values to be added

to the array starting at index start.

**string `toLocaleString()`**

Converts an array to a string and returns the string (localized).

**string `toSource()`**

Creates a string representation of this object that can be fed back to **eval()** to re-create an object. Works only with built-in classes.

**string `toString()`**

Converts an array to a string and returns the string.

**number `unshift(value:any)`**

Adds one or more elements to the beginning of the array.

Parameter	Type	Description
value	any	The values of one or more elements to be added to the beginning of the array.

Used in [Function.apply \(thisObj:Object, args:Array\)](#)

Return

**Array `Array.Array (length:number)`**

**Array `Array.concat (value:any)`**

**Array `Array.reverse ()`**

**Array `Array.slice ()`**

**Array `Array.splice (start:number, num:number, value:any)`**

**Array `Folder.getFiles (mask:any)`**

**Array `RegExp.exec (text:string)`**

**Array `String.match (regexp:RegExp)`**

**Array `XML.inScopeNamespaces ()`**

**Array `XML.namespaceDeclarations ()`**

[Contents :: Index](#)

Class

# Boolean

Wraps a boolean value.

QuickLinks [Boolean](#), [toSource](#), [toString](#), [valueOf](#)

## Methods

**bool Boolean (value:any)**

Creates and returns a new Boolean object set to the value of the argument converted to a boolean.

Parameter	Type	Description
value	any	The value to be converted to a Boolean.

**string toSource ()**

Creates a string representation of this object that can be fed back to **eval()** to re-create an object. Works only with built-in classes.

**string toString ()**

Returns the string representation of the value of bool.

**bool valueOf ()**

Returns the primitive value of bool.

[Contents :: Index](#)

Class

# Date

A date/time object.

## QuickLinks

[Date](#), [UTC](#), [getDate](#), [getDay](#), [getFullYear](#), [getHours](#), [getMilliseconds](#), [getMinutes](#), [getMonth](#), [getSeconds](#), [getTime](#), [getTimezoneOffset](#), [getUTCDate](#), [getUTCDay](#), [getUTCFullYear](#), [getUTCHours](#), [getUTCMilliseconds](#), [getUTCMinutes](#), [getUTCMonth](#), [getUTCSeconds](#), [getYear](#), [parse](#),  [setDate](#), [setFullYear](#), [setHours](#), [setMilliseconds](#), [setMinutes](#), [setMonth](#), [setSeconds](#), [setTime](#), [setUTCDate](#), [setUTCFullYear](#), [setUTCHours](#), [setUTCMilliseconds](#), [setUTCMinutes](#), [setUTCMonth](#), [setUTCSeconds](#), [setYear](#), [toDateString](#), [toGMTString](#), [toLocaleDateString](#), [toLocaleString](#), [toLocaleTimeString](#), [toSource](#), [toString](#), [toTimeString](#), [toUTCString](#), [valueOf](#)

## Methods

**Date** **Date** (**year:number** [, **month:number=0**] [, **day:number=0**] [, **hours:number=0**] [, **min:number=0**] [, **sec:number=0**] [, **ms:number=0**])

Returns a new Date object holding the current date and time.

Parameter	Type	Description
year	number	The year expressed in four digits.
month	number	An integer value from 0 (Jan) to 11 (Dec). (default: 0)
day	number	An integer value from 1 to 31, If this argument is not supplied, its value is set to 0. (default: 0)
hours	number	An integer value from 0 (midnight) to 23 (11 PM). If this argument is not supplied, its value is set to 0. (default: 0)
min	number	An integer value from 0 to 59. If this argument is not supplied, its value is set to 0. (default: 0)

sec	number	An Integer value from 0 to 59. If this argument is not supplied, its value is set to 0. (default: <b>0</b> )
ms	number	An integer value from 0 to 999. If this argument is not supplied, its value is set to 0. (default: <b>0</b> )

**Date UTC** (`year:number [, month:number=0] [, day:number=0] [, hours:number=0] [, min:number=0] [, sec:number=0] [, ms:number=0]`)

Returns the number of milliseconds between midnight January 1, 1970, UTC, and the specified time.

Parameter	Type	Description
year	number	The year expressed in four digits, for example, 2001.
month	number	An integer value from 0 (Jan) to 11 (Dec). (default: <b>0</b> )
day	number	An integer value from 1 to 31, If this argument is not supplied, its value is set to 0. (default: <b>0</b> )
hours	number	An integer value from 0 (midnight) to 23 (11 PM). If this argument is not supplied, its value is set to 0. (default: <b>0</b> )
min	number	An integer value from 0 to 59. If this argument is not supplied, its value is set to 0. (default: <b>0</b> )
sec	number	An Integer value from 0 to 59. If this argument is not supplied, its value is set to 0. (default: <b>0</b> )
ms	number	An integer value from 0 to 999. If this argument is not supplied, its value is set to 0. (default: <b>0</b> )

**number getDate ()**

Returns the day of the month of the specified Date object in local time.

**number getDay ()**

Returns the day of the week for the specified Date object in local time.

**number getFullYear ()**

Returns the four digit year of the specified Date object in local time.

**number getHours ()**

Returns the hour of the specified Date object in local time.

**number getMilliseconds ()**

Returns the milliseconds of the specified Date object in local time.

**number getMinutes ()**

Returns the minutes of the specified Date object in local time.

**number getMonth ()**

Returns the month of the specified Date object in local time.

**number getSeconds ()**

Returns the seconds of the specified Date object in local time.

**number getTime ()**

Returns the number of milliseconds since midnight January 1, 1970 UTC for the specified Date object.

**number getTimezoneOffset ()**

Returns the difference in minutes between the computer's local time and UTC.

**number getUTCDate ()**

Returns the day of the month of the specified Date object according to UTC.

**number getUTCDay ()**

Returns the day of the week for the specified Date object according to UTC.

**number getUTCFullYear ()**

Returns the four digit year of the specified Date object according to UTC.

**number `getUTCHours ()`**

Returns the hour of the specified Date object according to UTC.

**number `getUTCMilliseconds ()`**

Returns the milliseconds of the specified Date object according to UTC.

**number `getUTCMinutes ()`**

Returns the minutes of the specified Date object according to UTC.

**number `getUTCMonth ()`**

Returns the month of the specified Date object according to UTC.

**number `getUTCSeconds ()`**

Returns the seconds of the specified Date object according to UTC.

**number `getYear ()`**

Returns the year of the specified Date object, as a difference from 1900, in local time.

**Date `parse (text:string)`**

Parses a string, returning a new Date object. The string should be similar to the string returned by `toString()`.

Parameter	Type	Description
text	string	The string to parse.

**number  `setDate (date:number)`**

Sets the day of the month of a specified Date object according to local time.

Parameter	Type	Description
date	number	An integer from 1 to 31 indicating the day of the month.

**number `setFullYear (year:number)`**

Sets the year of a specified Date object according to local time.

Parameter	Type	Description
year	number	A four-digit integer value indicating the year to set.

**number `setHours` (`hour:number`)**

Sets the hours of a specified Date object according to local time.

Parameter	Type	Description
hour	number	An integer value from 0 (midnight) to 23 (11 PM).

**number `setMilliseconds` (`ms:number`)**

Sets the milliseconds of a specified Date object according to local time.

Parameter	Type	Description
ms	number	An integer value from 0 to 999.

**number `setMinutes` (`minutes:number`)**

Sets the minutes of a specified Date object according to local time.

Parameter	Type	Description
minutes	number	An integer value from 0 to 59.

**number `setMonth` (`month:number`)**

Sets the month of a specified Date object according to local time.

Parameter	Type	Description
month	number	An integer value from 0 (Jan) to 11 (Dec).

**number `setSeconds` (`seconds:number`)**

Sets the seconds of a specified Date object according to local time.

Parameter	Type	Description

seconds	number	An integer value from 0 to 59.
---------	--------	--------------------------------

**number setTime (`ms:number`)**

Sets the date of a specified Date object in milliseconds since midnight, January 1, 1970.

Parameter	Type	Description
ms	number	An integer indicating the number of milliseconds between the date set and midnight, January 1, 1970.

**number setDate (`date:number`)**

Sets the date of a specified Date object according to universal time.

Parameter	Type	Description
date	number	An integer from 1 to 31 indicating the day of the month.

**number setFullYear (`year:number`)**

Sets the year of a specified Date object according to UTC, can also set the month and date.

Parameter	Type	Description
year	number	The year expressed in four digits.

**number setUTCHours (`hours:number`)**

Sets the hours of a specified Date object according to UTC.

Parameter	Type	Description
hours	number	An integer value from 0 (midnight) to 23 (11 PM) indicating the hour to be set.

**number setUTCMilliseconds (`ms:number`)**

Sets the milliseconds of a specified Date object according to UTC.

Parameter	Type	Description
ms	number	An integer value in the range of 0 to 999 indicating the number of

milliseconds to set.

#### **number `setUTCMilliseconds` (`min:number`)**

Sets the milliseconds of a specified Date object according to UTC.

Parameter	Type	Description
min	number	An integer value in the range 0 to 59 indicating the number of minutes to be set.

#### **number `setUTCMonth` (`month:number`)**

Sets the month of a specified Date object according to UTC.

Parameter	Type	Description
month	number	An integer value in the range 0 (Jan.) to 11 (Dec.) indicating the month to set.

#### **number `setUTCSeconds` (`sec:number`)**

Sets the seconds of a specified Date object according to UTC.

Parameter	Type	Description
sec	number	An integer value in the range 0 to 59 indicating the number of seconds to set.

#### **number `setYear` (`year:number`)**

Sets the year of a specified Date object according to local time, as a difference between the current year and 1900.

Parameter	Type	Description
year	number	An integer value indicating the year to set.

**string `toDateString` ()**

**string `toGMTString` ()**

**string `toLocaleDateString` ()**

**string `toLocaleString` ()**

**string `toLocaleTimeString` ()**

**string `toSource` ()**

Creates a string representation of this object that can be fed back to **eval()** to re-create an object. Works only with built-in classes.

**string `toString()`**

Returns a string value representing the date and time stored in the Date object in human readable format.

**string `toTimeString()`**

**string `toUTCString()`**

**number `valueOf()`**

The `valueOf()` method returns the number of milliseconds that have passed since midnight, Returns an integer.

Element of `$.buildDate`

`File.created`

`File.modified`

`Folder.created`

`Folder.modified`

Return **Date `Date.Date (year:number [, month:number=0] [, day:number=0] [, hours:number=0] [, min:number=0] [, sec:number=0] [, ms:number=0])`**

**Date `Date.UTC (year:number [, month:number=0] [, day:number=0] [, hours:number=0] [, min:number=0] [, sec:number=0] [, ms:number=0])`**

**Date `Date.parse (text:string)`**

[Contents](#) :: [Index](#)

Class

# Dictionary

A dictionary of class descriptions.

QuickLinks [getClass](#), [getClasses](#), [getGroups](#), [toXML](#)

## Methods

[Reflection](#) **Reflection getClass (name:string)**

Gets a class description.

Parameter	Type	Description
name	string	The class name.

Array of [string](#) **getClasses ( [name:string=All])**

Gets a list of classe names by group.

Parameter	Type	Description
name	string	The group name. If not supplied, all classes are returned. (default: All)

Array of [string](#) **getGroups ()**

Gets the list of groups.

[XML](#) **toXML (prefix:string)**

Returns a Table of Contents in OMV format.

Parameter	Type	Description
prefix	string	The prefix to be used in the href attributes. Optional.

[Contents](#) :: [Index](#)

Class

# Error

Wraps a runtime error.

QuickLinks [Error](#), [toSource](#), [toString](#)

## Properties

Property	Type	Access	Description
description	string	r/w	The error message.

## Methods

**Error** **Error** (`msg:string, file:string, line:number`)

Creates a new Error object.

Parameter	Type	Description
msg	string	The error message.
file	string	The name of the file.
line	number	The line number.

`string toSource ()`

Creates a string representation of this object that can be fed back to **eval()** to re-create an object. Works only with built-in classes.

`string toString ()`

Convert this object to a string.

Element of `$.error`

Return **Error** **Error** (`msg:string, file:string, line:number`)

[Contents :: Index](#)

Class

# File

Represents a file in the local file system in a platform-independent manner.

**QuickLinks** [File](#), [changePath](#), [close](#), [copy](#), [createAlias](#), [decode](#), [encode](#), [execute](#), [getRelativeURI](#), [isEncodingAvailable](#), [open](#), [openDialog](#), [openDlg](#), [read](#), [readch](#), [readln](#), [remove](#), [rename](#), [resolve](#), [saveDialog](#), [saveDlg](#), [seek](#), [tell](#), [toSource](#), [toString](#), [write](#), [writeln](#)

## Properties

Property	Type	Access	Description
absoluteURI	string	readonly	The full path name for the referenced file in URI notation.
alias	bool	readonly	If true, the object refers to a file system alias or shortcut.
created	Date	readonly	The creation date of the referenced file, or null if the object does not refer to a file on disk.
creator	string	readonly	In Mac OS, the file creator as a four-character string. In Windows or UNIX, value is "????".
displayName	string	readonly	The localized name of the referenced file, without the path specification.
encoding	string	r/w	Gets or sets the encoding for subsequent read/write operations. One of the encoding constants listed in

				the JavaScript Tools Guide. If the value is not recognized, uses the system default encoding. A special encoder, BINARY, is used to read binary files. It stores each byte of the file as one Unicode character regardless of any encoding. When writing, the lower byte of each Unicode character is treated as a single byte to write.
eof	bool	readonly		When true, a read attempt caused the current position to be at the end of the file, or the file is not open.
error	string	r/w		A string containing a message describing the most recent file system error.  Typically set by the file system, but a script can set it. Setting this value clears any error message and resets the error bit for opened files. Contains the empty string if there is no error.
exists	bool	readonly		If true, this object refers to a file or file-system alias that actually exists in the file system.
fs	string	readonly		The name of the file system.  This is a class property

			accessed through the File constructor. Valid values are "Windows", "Macintosh", and "Unix".
fsName	string	readonly	The platform-specific full path name for the referenced file.
fullName	string	readonly	The full path name for the referenced file in URI notation.
hidden	bool	r/w	When true, the file is not shown in the platform-specific file browser.  If the object references a file-system alias or shortcut, the flag is altered on the alias, not on the original file.
length	number	r/w	The size of the file in bytes.  Can be set only for a file that is not open, in which case it truncates or pads the file with 0-bytes to the new length.
lineFeed	string	r/w	How line feed characters are written in the file system.  One of the values "Windows", "Macintosh", or "Unix".
modified	Date	readonly	The date of the referenced file's last modification, or null if the object does not refer to a file on the disk.

name	string	readonly	The file name portion of the absolute URI for the referenced file, without the path specification.
parent	Folder	readonly	The Folder object for the folder that contains this file.
path	string	readonly	The path portion of the absolute URI for the referenced file, without the file name.
readonly	bool	r/w	<p>When true, prevents the file from being altered or deleted.</p> <p>If the referenced file is a file-system alias or shortcut, the flag is altered on the alias, not on the original file.</p>
relativeURI	string	readonly	The path name for the object in URI notation, relative to the current folder.
type	string	readonly	<p>The file type as a four-character string.</p> <p>In Mac OS, the Mac OS file type. In Windows, "appl" for .EXE files, "shlb" for .DLL files and "TEXT" for any other file.</p>
Methods	<b>File</b> <b>File</b> ( <b>path:string</b> )		
	Creates and returns a new File object referring to a given file system location.		

Parameter	Type	Description
path	string	The full or partial path name of the file, in platform-specific or URI format.

**bool changePath (path:string)**

Changes the path specification of the referenced file.

Parameter	Type	Description
path	string	A string containing the new path, absolute or relative to the current folder.

**bool close ()**

Closes this open file.

**bool copy (target:string File)**

Copies this object's referenced file to the specified target location.

Parameter	Type	Description
target	string File	A string with the URI path to the target location, or a File object that references the target location.

**void createAlias (path:string)**

Makes this file a file-system alias or shortcut to the specified file.

Parameter	Type	Description
path	string	A string containing the path of the target file.

**string decode (uri:string)**

Decodes a UTF-8 encoded string as required by RFC 2396, and returns the decoded string.

Parameter	Type	Description
uri	string	The UTF-8 encoded string to decode.

**string encode (name:string)**

Encodes a string as required by RFC 2396, and returns the

encoded string.

Parameter	Type	Description
name	string	The string to encode.

**bool execute ()**

Executes or opens this file using the appropriate application, as if it had been double-clicked in a file browser.

**string getRelativeURI (basePath:string)**

Retrieves and returns the path for this file, relative to the specified base path, in URI notation.

Parameter	Type	Description
basePath	string	A base path in URI notation.

**bool isEncodingAvailable (name:string)**

Reports whether a given encoding is available.

Parameter	Type	Description
name	string	The encoding name.

**bool open (mode:string, type:string, creator:string)**

Opens the referenced file for subsequent read/write operations. The method resolves any aliases to find the file.

Parameter	Type	Description
mode	string	The read-write mode, a single-character string.
type	string	In Mac OS, the type of a newly created file, a 4-character string. Ignored in Windows and UNIX.
creator	string	In Mac OS, the creator of a newly created file, a 4-character string. Ignored in Windows and UNIX.

**File openDialog (prompt:string [, filter:any=null] [, multiSelect:bool=false])**

Opens a dialog so the user can select one or more files to open.

Parameter	Type	Description
prompt	string	The prompt text, displayed if the dialog allows a prompt.
filter	any	A filter that limits the types of files displayed in the dialog. (default: <b>null</b> )
multiSelect	bool	When true, the user can select multiple files and the return value is an array. (default: <b>false</b> )

**File openDlg (prompt:string [, filter:any=null] [, multiSelect:bool=false])**

Opens the built-in platform-specific file-browsing dialog, in which the user can select an existing file or files, and creates new File objects to represent the selected files.

Parameter	Type	Description
prompt	string	A string containing the prompt text, if the dialog allows a prompt.
filter	any	A filter that limits the types of files displayed in the dialog. (default: <b>null</b> )
multiSelect	bool	When true, the user can select multiple files and the return value is an array. (default: <b>false</b> )

**string read (chars:number)**

Reads the contents of the file, starting at the current position.

Parameter	Type	Description
chars	number	An integer specifying the number of characters to read.

**string readch ()**

Reads a single text character from the file at the current position.

**string readln ()**

Reads a single line of text from the file at the current position.

**bool remove ()**

Deletes the file associated with this object from disk immediately, without moving it to the system trash.

**bool rename (*newName:string*)**

Renames the associated file.

Parameter	Type	Description
<i>newName</i>	string	The new file name, with no path information.

**File resolve ()**

Attempts to resolve the file-system alias or shortcut that this object refers to.

**File saveDialog (*prompt:string* [, *filter:any=null*])**

Opens a dialog so the user can select a file name to save to.

Parameter	Type	Description
<i>prompt</i>	string	The prompt text, displayed if the dialog allows a prompt.
<i>filter</i>	any	A filter that limits the types of files displayed in the dialog. (default: <b>null</b> )

**File saveDlg (*prompt:string* [, *filter:any=null*])**

Opens the built-in platform-specific file-browsing dialog, in which the user can select an existing file location to which to save information, and creates a new File object to represent the selected file.

Parameter	Type	Description
<i>prompt</i>	string	A string containing the prompt text, if the dialog allows a prompt.
<i>filter</i>	any	A filter that limits the types of files displayed in the dialog. (default: <b>null</b> )

**bool seek (*pos:number* [, *mode:number=0*])**

Seeks to a given position in the file.

Parameter	Type	Description

pos	number	The new current position in the file as an offset in bytes from the start, current position, or end, depending on the mode.
mode	number	The seek mode. (default: <b>0</b> )

### **number tell ()**

Retrieves the current position as a byte offset from the start of the file.

### **string toSource ()**

Creates and returns a serialized string representation of this object.

### **string toString ()**

Converts this object to a string.

### **bool write (text:string)**

Writes the specified text to the file at the current position.

Parameter	Type	Description
text	string	A text string to be written.

### **bool writeln (text:string)**

Writes a string to the file at the current position and appends a line-feed sequence.

Parameter	Type	Description
text	string	A text string to be written.

Element of [Reflection.sampleFile](#)  
[ReflectionInfo.sampleFile](#)

Used in [\\$.evalFile \(file:File \[, timeout:number=10000\]\)](#)  
[File.copy \(target:string File\)](#)

Return [File File.File \(path:string\)](#)  
[File File.openDialog \(prompt:string \[, filter:any=null\] \[, multiSelect:bool=false\]\)](#)  
[File File.openDlg \(prompt:string \[, filter:any=null\] \[, multiSelect:bool=false\]\)](#)

**File** **File.resolve** ()  
**File** **File.saveDialog** (*prompt:string* [, *filter:any=null*])  
**File** **File.saveDlg** (*prompt:string* [, *filter:any=null*])

[Contents](#) :: [Index](#)

Class

# Folder

Represents a file-system folder or directory in a platform-independent manner.

## QuickLinks

[Folder](#), [changePath](#), [create](#), [decode](#), [encode](#), [execute](#), [getFiles](#), [getRelativeURI](#), [isEncodingAvailable](#), [remove](#), [rename](#), [resolve](#), [selectDialog](#), [selectDlg](#), [toSource](#), [toString](#)

## Properties

	Property	Type	Access	Description
	absoluteURI	string	readonly	The full path name for the referenced folder in URI notation.
	alias	bool	readonly	When true, the object refers to a file system alias or shortcut.
	appData	<a href="#">Folder</a>	readonly	<p>The folder containing the application data for all users.</p> <p>In Windows, the value of %APPDATA% (by default, C:\\Documents and Settings\\All Users\\Application Data) In Mac OS, /Library/Application Support</p>
	appPackage	<a href="#">Folder</a>	readonly	<p>In Mac OS, a Folder object for the folder containing the bundle of the running application.</p>
	commonFiles	<a href="#">Folder</a>	readonly	<p>A Folder object for the folder containing common files for programs installed by the user.</p> <p>In Windows, the value of %CommonProgramFiles% (by default, C:\\Program Files\\Common Files) In Mac OS, /Library/Application Support</p>

created	Date	readonly	The creation date of the referenced folder, or null if the object does not refer to a folder on disk.
current	Folder	r/w	A Folder object for the current folder.  Assign a Folder object or a string containing the new path name to set the current folder This is a class property accessed through the Folder constructor.
desktop	Folder	readonly	A Folder object for the folder that contains the user's desktop.  In Windows, C:\\Documents and Settings\\username\\Desktop Mac OS, ~/Desktop
displayName	string	readonly	The localized name portion of the absolute URI for the referenced folder, without the path specification.
error	string	r/w	A message describing the most recent file system error.  Typically set by the file system but a script can set it. Setting this value clears any error message and resets the error list for opened files. Contains the empty string if there is no error.
exists	bool	readonly	When true, this object refers to a folder that currently exists in the file system.

	fs	string	readonly	The name of the current file system.  One of "Windows", "Macintosh", or "Unix".
	fsName	string	readonly	The platform-specific name of the referenced folder as a full path name.
	fullName	string	readonly	The full path name for the referenced folder in URI notation..
	modified	Date	readonly	The date of the referenced folder's last modification, or null if the object does not refer to a folder on disk.
	myDocuments	Folder	readonly	A folder pointing to the user's My Documents folder.  In Windows, C:\\Documents and Settings\\username\\My Documents In Mac OS, ~/Documents
	name	string	readonly	The folder name portion of the absolute URI for the referenced file, without the path specification.
	parent	Folder	readonly	The Folder object for the folder that contains this folder or null if this object refers to the root folder of a volume.
	path	string	readonly	The path portion of the object's absolute URI for the referenced file, without the folder name.
	relativeURI	string	readonly	The path name for the referenced folder in URI

			notation, relative to the current folder.
startup	<a href="#">Folder</a>	readonly	A Folder object for the folder containing the executable image of the running application.
system	<a href="#">Folder</a>	readonly	A Folder object for the folder containing the operating system files.  In Windows, the value of %windir% (by default, C:\\Windows) In Mac OS, /System
temp	<a href="#">Folder</a>	readonly	A Folder object for the default folder for temporary files.
trash	<a href="#">Folder</a>	readonly	A Folder object for the folder containing deleted items.
userData	<a href="#">Folder</a>	readonly	A Folder object for the folder containing the user's application data.  In Windows, the value of %USERDATA% (by default, C:\\Documents and Settings\\username\\Application Data) In Mac OS, ~/Library/Application Support

**Methods** [\*\*Folder\*\*](#) [\*\*Folder \(path:string\)\*\*](#)  
Creates and returns a new Folder object referring to a given file-system location.

Parameter	Type	Description
path	string	The absolute or relative path to the folder associated with this object, specified in URL format.

**bool changePath (path:string)**

Changes the path specification of the referenced folder.

Parameter	Type	Description
-----------	------	-------------

path	string	A string containing the new path, absolute or relative to the current folder.
------	--------	---

**bool create ()**

Creates a folder at the location given by this object's path property.

**string decode (uri:string)**

Decodes a UTF-8 encoded string as required by RFC 2396, and returns the decoded string.

Parameter	Type	Description
-----------	------	-------------

uri	string	The UTF-8 string to decode.
-----	--------	-----------------------------

**string encode (name:string)**

Encodes a string as required by RFC 2396, and returns the encoded string.

Parameter	Type	Description
-----------	------	-------------

name	string	The string to encode.
------	--------	-----------------------

**bool execute ()**

Opens this folder in the platform-specific file browser (as if it had been double-clicked in the file browser).

**Array getFiles (mask:any)**

Retrieves the contents of this folder, filtered by the supplied mask.

Parameter	Type	Description
-----------	------	-------------

mask	any	A search mask for file names, specified as a string or a function.
------	-----	--

**string getRelativeURI ( [basePath:string=.] )**

Retrieves and returns the path for this file, relative to the specified base path, in URI notation.

Parameter	Type	Description
-----------	------	-------------

basePath	string	A base path in URI notation. (default: .)
----------	--------	---

**bool isEncodingAvailable (name:string)**

Reports whether a given encoding is available.

Parameter	Type	Description
name	string	The encoding name.

**bool remove ()**

Deletes the folder associated with this object from disk immediately, without moving it to the system trash.

**bool rename (newName:string)**

Renames the associated folder.

Parameter	Type	Description
newName	string	The new folder name, with no path information.

**Folder resolve ()**

Attempts to resolve the file-system alias or shortcut that this object refers to.

**Folder selectDialog (prompt:string)**

Opens the built-in platform-specific file-browsing dialog, and creates a new File or Folder object for the selected file or folder.

Parameter	Type	Description
prompt	string	The prompt text, if the dialog allows a prompt.

**Folder selectDlg (prompt:string)**

Opens the built-in platform-specific file-browsing dialog, and creates a new File or Folder object for the selected file or folder.

Parameter	Type	Description
prompt	string	The prompt text, if the dialog allows a prompt.

**string toSource ()**

Creates and returns a serialized string representation of this object.

**string toString ()**

Converts this object to a string.

---

Element of `File.parent`  
`Folder.appData`  
`Folder.appPackage`  
`Folder.commonFiles`  
`Folder.current`  
`Folder.desktop`  
`Folder.myDocuments`  
`Folder.parent`  
`Folder.startup`  
`Folder.system`  
`Folder.temp`  
`Folder.trash`  
`Folder.userData`

---

Return `Folder Folder.Folder (path:string)`  
`Folder Folder.resolve ()`  
`Folder Folder.selectDialog (prompt:string)`  
`Folder Folder.selectDlg (prompt:string)`

---

[Contents](#) :: [Index](#)

Class

# Function

The Function class wraps a built-in or JavaScript function.

QuickLinks [Function](#), [apply](#), [call](#), [toSource](#), [toString](#)

## Properties

Property	Type	Access	Description
arguments	Object	r/w	The function arguments, packed into an array.  This property is deprecated; use the arguments property within the function body.
arity	number	readonly	The number of formal arguments.  This property is deprecated; use the length property instead.
length	number	readonly	The number of formal arguments.
name	string	readonly	The function name.

## Methods

**Function** **Function** (`arguments:string`, `body:string`)

Parameter	Type	Description
arguments	string	The list of formal arguments, separated by commas.
body	string	The body of the function to create.

**any** **apply** (`thisObj:Object`, `args:Array`)

Apply a this object and an argument list to a function.

Parameter	Type	Description
thisObj	Object	

args	Array	An array of arguments.
------	-------	------------------------

**any call (thisObj:Object, argument:any)**  
Apply a this object and arguments to a function.

Parameter	Type	Description
thisObj	Object	
argument	any	

**string toSource ()**

Creates a string representation of this object that can be fed back to **eval()** to re-create an object. Works only with JavaScript functions.

**string toString ()**

Returns the function definition as a string.

Element of **Object.constructor**

Used in **Array.sort (userFunction:Function)**  
**Object.watch (name:string, func:Function)**

Return **Function Function.Function (arguments:string, body:string)**

[Contents :: Index](#)

Class

# global

The global methods and properties.

## QuickLinks

[alert](#), [confirm](#), [decodeURI](#), [decodeURIComponent](#), [encodeURI](#), [encodeURIComponent](#), [escape](#), [eval](#), [isFinite](#), [isNaN](#), [isXMLName](#), [localize](#), [localize](#), [parseFloat](#), [parseInt](#), [prompt](#), [setDefaultXMLNamespace](#), [unescape](#), [uneval](#)

## Properties

Property	Type	Access	Description
Infinity	number	r/w	The Infinity global property is a predefined variable with the value for infinity.
NaN	number	r/w	The NaN global property is a predefined variable with the value NaN (Not-a-Number), as specified by the IEEE-754 standard.
undefined	Undefined	r/w	This global property is a predefined variable with the value for an undefined value.

## Methods

**void alert (message:string, title:string [, errorIcon:bool=false])**  
Displays an alert box

Parameter	Type	Description
message	string	The text to display
title	string	The title of the alert; ignored on the Macintosh
errorIcon	bool	Display an Error icon; ignored on the Macintosh (default: <b>false</b> )

**bool confirm (message:string [, noAsDefault:bool=false], title:string)**

Displays an alert box with Yes and No buttons; returns true for Yes

Parameter	Type	Description
message	string	The text to display
noAsDefault	bool	Set to true to set the No button as the default button (default: <b>false</b> )
title	string	The title of the alert; ignored on the Macintosh

**string decodeURI (uri:string)**

Decodes a string created with **encodeURI()**.

Parameter	Type	Description
uri	string	The text to decode.

**string decodeURIComponent (uri:string)**

Decodes a string created with **encodeURIComponent()**.

Parameter	Type	Description
uri	string	The text to decode.

**string encodeURI (text:string)**

Encodes a string after RFC2396.

Parameter	Type	Description
text	string	The text to encode.

**string encodeURIComponent (text:string)**

Encodes a string after RFC2396.

Parameter	Type	Description
text	string	The text to encode.

**string escape (aString:string)**

Creates a URL-encoded string from aString.

Parameter	Type	Description
aString	string	The string to be encoded.

**any eval (stringExpression:string)**

Evaluates its argument as a JavaScript script, and returns the

result of evaluation.

Parameter	Type	Description
stringExpression	string	The string to evaluate.

**bool isFinite (*expression:number*)**

Evaluates an expression and reports whether the result is a finite number.

Parameter	Type	Description
expression	number	Any valid JavaScript expression.

**bool isNaN (*expression:number*)**

Evaluates an expression and reports whether the result is "Not-a-Number" (NaN).

Parameter	Type	Description
expression	number	Any valid JavaScript expression.

**bool isXMLName (*name:string*)**

Returns true if the supplied string is a valid XML name.

Parameter	Type	Description
name	string	The XML name to test.

**string localize (*what:string, argument:any*)**

Localizes a ZString-encoded string and merges additional arguments into the string.

Parameter	Type	Description
what	string	The string to localize. A ZString-encoded string that can contain placeholder for additional arguments in the form %1 to %n.
argument	any	Optional argument(s) to be merged into the string.

**string localize (*what:string, argument:any*)**

Localizes a ZString-encoded string and merges additional arguments into the string.

Parameter	Type	Description
what	string	The string to localize. A ZString-encoded string that can contain placeholder for additional arguments in the form %1 to %n.
argument	any	Optional argument(s) to be merged into the string.

**number parseFloat (`text:string`)**

Extracts a floating-point number from a string.

Parameter	Type	Description
text	string	The string from which to extract a floating point number.

**number parseInt (`text:string, base:number`)**

Extracts an integer from a string.

Parameter	Type	Description
text	string	The string from which to extract an integer.
base	number	The base of the string to parse (from base 2 to base 36).

**string prompt (`prompt:string, default:string, title:string`)**

Displays a dialog allowing the user to enter text

Parameter	Type	Description
prompt	string	The text to display
default	string	The default text to preset the edit field with
title	string	The title of the dialog;

**void setDefaultXMLNamespace (`namespace:Namespace`)**

Defines the default XML namespace.

Parameter	Type	Description
namespace	Namespace	The namespace to use.

---

**string unescape (stringExpression:string)**

Translates URL-encoded string into a regular string, and returns that string.

Parameter	Type	Description
stringExpression	string	The URL-encoded string to convert.

**string uneval (what:any)**

Creates a source code representation of the supplied argument, and returns it as a string.

Parameter	Type	Description
what	any	The object to uneval.

[Contents :: Index](#)

Class

# Math

A global object containing a set of math functions and constants.

QuickLinks [abs](#), [acos](#), [asin](#), [atan](#), [atan2](#), [ceil](#), [cos](#), [exp](#), [floor](#), [log](#), [max](#), [min](#), [pow](#), [random](#), [round](#), [sin](#), [sqrt](#), [tan](#)

## Properties

Property	Type	Access	Description
E	number	readonly	Euler's constant and the base of natural logarithms. (default: <b>2.7182818284590452354</b> )
LN10	number	readonly	The natural logarithm of 10. (default: <b>2.302585092994046</b> )
LN2	number	readonly	The natural logarithm of 2. (default: <b>0.6931471805599453</b> )
LOG10E	number	readonly	The base 10 logarithm of e. (default: <b>0.4342944819032518</b> )
LOG2E	number	readonly	The base 2 logarithm of e. (default: <b>1.4426950408886934</b> )
PI	number	readonly	The ratio of the circumference of a circle to its diameter. (default: <b>3.14159265358979323846</b> )
SQRT1_2	number	readonly	The reciprocal of the square root of 1/2. (default: <b>0.7071067811865476</b> )
SQRT2	number	readonly	The square root of 2. (default: <b>1.4142135623730951</b> )

Methods number **abs** (*x:number*)

Returns the absolute value of a number.

Parameter	Type	Description
x	number	A number.

**number acos (x:number)**

Returns the arc cosine(in radians) of a number.

Parameter	Type	Description
x	number	A number.

**number asin (x:number)**

Returns the arc sin(in radians) of a number.

Parameter	Type	Description
x	number	A number.

**number atan (x:number)**

Returns the arc tangent(in radians) of a number.

Parameter	Type	Description
x	number	A number.

**number atan2 (y:number, x:number)**

Returns the arc tangent of the quotient of its arguments (y/x).

Parameter	Type	Description
y	number	A number.
x	number	A number.

**number ceil (x:number)**

Rounds the number up to the nearest integer.

Parameter	Type	Description
x	number	A number.

**number cos (x:number)**

Returns the cosine of an angle provided in radians.

Parameter	Type	Description
x	number	An angle, in radians.

**number exp (x:number)**

Returns Math.E raised to the power of a number.

Parameter	Type	Description
x	number	A number.

**number floor (x:number)**

Rounds a number down to the nearest integer.

Parameter	Type	Description
x	number	A number.

**number log (x:number)**

Returns the natural logarithm of a number.

Parameter	Type	Description
x	number	A number.

**number max (x:number, y:number)**

Returns the maximum of two numbers.

Parameter	Type	Description
x	number	
y	number	

**number min (x:number, y:number)**

Returns the minimum of two numbers.

Parameter	Type	Description
x	number	Numbers.
y	number	

**number pow (x:number, y:number)**

Returns x raised to the power of y.

Parameter	Type	Description
x	number	Numbers.
y	number	

**number random ()**

Returns a pseudo-random number from 0.0 up to but not including 1.0.

**number round (*x:number*)**

Rounds a number to the nearest integer.

Parameter	Type	Description
x	number	A number.

**number sin (*x:number*)**

Returns the sine of an angle provided in radians.

Parameter	Type	Description
x	number	An angle, in radians.

**number sqrt (*x:number*)**

Returns the square root of a number.

Parameter	Type	Description
x	number	A number.

**number tan (*x:number*)**

Returns the tangent of an angle provided in radians.

Parameter	Type	Description
x	number	An angle, in radians.

[Contents](#) :: [Index](#)

Class

# Namespace

Properties	Property	Type	Access	Description
	prefix	string	readonly	
	uri	string	readonly	
Used in	<code>XML.addNamespace (namespace:Namespace)</code> <code>XML.removeNamespace (namespace:Namespace)</code> <code>XML.setNamespace (namespace:Namespace)</code> <code>global.setDefaultXMLNamespace (namespace:Namespace)</code>			
Return	<code>Namespace XML.namespace ()</code>			

[Contents :: Index](#)

Class

# Number

A wrapper class for numbers.

QuickLinks [Number](#), [toExponential](#), [toFixed](#), [toLocaleString](#), [toPrecision](#), [toString](#), [valueOf](#)

## Properties

Property	Type	Access	Description
MAX_VALUE	number	readonly	A constant representing the largest representable number. (default: <b>1.7976931348e+308</b> )
MIN_VALUE	number	readonly	A constant representing the smallest representable number. (default: <b>2.2250738585e-308</b> )
NEGATIVE_INFINITY	number	readonly	A constant representing negative infinity.
Nan	number	readonly	A constant representing the special "Not a value. (default)
POSITIVE_INFINITY	number	readonly	A constant representing positive infinity.

## Methods

**Number Number (value:any)**

Returns a new Number object set to the value of the argument.

Parameter	Type	Description
value	any	The value of the object being created.

**number toExponential (decimals:number)**

Converts the Number object to a string in scientific notation.

Parameter	Type	Description
decimals	number	The number of decimals.

**number toFixed (*decimals:number*)**

Converts the Number object to a string with fixed decimals.

Parameter	Type	Description
decimals	number	The number of decimals.

**number toLocaleString ()**

Returns the value of a Number object converted to a string, using conventions.

**number toPrecision (*decimals:number*)**

Converts the Number object to a string in either scientific or fixed notation depending on its value.

Parameter	Type	Description
decimals	number	The number of decimals.

**number toSource ()**

Creates a string representation of this object that can be fed back into **eval()** to re-create an object. Works only with built-in classes.

**number toString (*radix:number*)**

Returns the value of a Number object converted to a string.

Parameter	Type	Description
radix	number	The optional conversion radix.

**number valueOf ()**

Returns the value of a Number object as a primitive number.

Return **Number** **Number.Number** (*value:any*)

[Contents :: Index](#)

Class

# Object

The base class of all JavaScript objects.

**QuickLinks** [Object](#), [hasOwnProperty](#), [isPrototypeOf](#), [isValid](#), [propertyIsEnumerable](#), [toLocaleString](#), [toSource](#), [toString](#), [unwatch](#), [valueOf](#), [watch](#)

Properties	Property	Type	Access	Description
	constructor	Function	readonly	Points to the constructor function that created this object.  Note that this property is treated as an XML element in the XML class.
	prototype	Object	readonly	Points to the prototype object for this object.  Note that this property is treated as an XML element in the XML class.
	reflect	Reflection	readonly	Retrieves and returns the Reflection object associated with this method or a property.  Note that this property is treated as an XML element in the XML class.

## Methods

**Object Object (what:any)**

Creates and returns a new object of a given type.

Parameter	Type	Description
what	any	The object type.

**bool hasOwnProperty (name:string)**

Reports whether a given property is defined with an instance or

within the prototype chain.

Parameter	Type	Description
name	string	The name of the property to check.

**bool isPrototypeOf (what:Object)**

Checks whether the given object is a prototype of this object.

Parameter	Type	Description
what	Object	The object to check.

**bool isValid (what:Object)**

Reports whether an object is still valid.

Parameter	Type	Description
what	Object	The object to check.

**bool propertyIsEnumerable (name:string)**

Reports whether a given property is enumerable.

Parameter	Type	Description
name	string	The name of the property to check.

**string toLocaleString ()**

Creates and returns a string representing this object, localized for the current locale. See **toString()**.

**string toSource ()**

Creates and returns a string representation of this object.

**string toString ()**

Creates and returns a string representing this object.

**void unwatch (name:string)**

Removes the watch function of a property.

Parameter	Type	Description
name	string	The name of the property to unwatch.

**Object valueOf ()**

Retrieves and returns the primitive value of this object.

**void watch (name:string, func:Function)**

Adds a watch function to a property, which is called when the value changes.

Parameter	Type	Description
name	string	The name of the property to watch.
func	Function	The function to be called when the value of this property changes.

Element of [\\$.global](#)  
[\\$.screens](#)  
[Function.arguments](#)  
[Object.prototype](#)

Used in [Function.apply \(thisObj:Object, args:Array\)](#)  
[Function.call \(thisObj:Object, argument:any\)](#)  
[Object.isValid \(what:Object\)](#)  
[Object.isPrototypeOf \(what:Object\)](#)

Return [Object Object.Object \(what:any\)](#)  
[Object Object.valueOf \(\)](#)  
[Object XML.defaultSettings \(\)](#)  
[Object XML.settings \(\)](#)

[Contents :: Index](#)

Class

# QName

## Properties

Property	Type	Access	Description
localName	string	readonly	
uri	string	readonly	

Used in [XML.setName \(name:QName\)](#)

Return [QName XML.name \(\)](#)

[Contents :: Index](#)

Class

# Reflection

Provides information about a class.

QuickLinks [find](#), [toXML](#)

Properties	Property	Type	Access	Description
	description	string	readonly	The long description text.
	help	string	readonly	The short description text.
	methods	Array of <a href="#">ReflectionInfo</a>	readonly	An array of method descriptions.
	name	string	readonly	The class name.
	properties	Array of <a href="#">ReflectionInfo</a>	readonly	An array of property descriptions.
	sampleCode	string	readonly	Sample code, if present.
	sampleFile	File	readonly	A file containing sample code. May be null.
	staticMethods	Array of <a href="#">ReflectionInfo</a>	readonly	An array of class method descriptions.

staticProperties	Array of <a href="#">ReflectionInfo</a>	readonly	An array of class property descriptions.
------------------	--	----------	--

---

Methods [ReflectionInfo find \(name:string\)](#)  
Find an element description by name.

Parameter	Type	Description
name	string	The name of the element to find.

[XML toXML \(\)](#)  
Return this class information as XML in OMV format.

---

Element of [Object.reflect](#)  
[ReflectionInfo.parent](#)

---

Return [Reflection Dictionary.getClass \(name:string\)](#)

---

[Contents :: Index](#)

Class

# ReflectionInfo

Provides information about a method, a property or method parameters.

## Properties

Property	Type	Access	Description
arguments	Array of ReflectionInfo	readonly	The description of method or function arguments.
dataType	string	readonly	The data type.
defaultValue	any	readonly	The default value.
description	string	readonly	The long description text.
help	string	readonly	The short description text.
isCollection	bool	readonly	Contains true if class description collects class.
max	number	readonly	The maximum value.
min	number	readonly	The minimum value.

	<code>name</code>	<code>string</code>	<code>readonly</code>	The element's name.
	<code>parent</code>	<code>Reflection</code>	<code>readonly</code>	The class object this element belongs to.
	<code>sampleCode</code>	<code>string</code>	<code>readonly</code>	Sample code, if present.
	<code>sampleFile</code>	<code>File</code>	<code>readonly</code>	A file containing sample code. May be null.
	<code>type</code>	<code>string</code>	<code>readonly</code>	The element's type.  One of: unknown readonly readwrite create method parameter
Element of <code>Reflection.methods</code> <code>Reflection.properties</code> <code>Reflection.staticMethods</code> <code>Reflection.staticProperties</code> <code>ReflectionInfo.arguments</code>				
Return <b>ReflectionInfo</b> <code>Reflection.find (name:string)</code>				

[Contents :: Index](#)

Class

# RegExp

Wraps regular expressions.

QuickLinks [RegExp](#), [compile](#), [exec](#), [test](#), [toString](#)

## Properties

Property	Type	Access	Description
\$1	string	readonly	The matched subexpression #1.
\$2	string	readonly	The matched subexpression #2.
\$3	string	readonly	The matched subexpression #3.
\$4	string	readonly	The matched subexpression #4.
\$5	string	readonly	The matched subexpression #5.
\$6	string	readonly	The matched subexpression #6.
\$7	string	readonly	The matched subexpression #7.
\$8	string	readonly	The matched subexpression #8.
\$9	string	readonly	The matched subexpression #9.
global	bool	r/w	Indicates whether the match is a global match.
ignoreCase	bool	r/w	Indicates whether the match is not case sensitive.
input	string	r/w	The original input string.

lastMatch	string	readonly	The last match.
lastParen	string	readonly	The value of the last matched subexpression.
leftContext	string	readonly	The string before the match.
multiline	bool	r/w	Indicates whether the match matches multiple lines.
rightContext	string	readonly	The string after the match.

## Methods

### **RegExp** **RegExp** (**pattern:string**)

Creates and returns a new RegExp object set to the value of the argument converted to a regular expression.

Parameter	Type	Description
pattern	string	The pattern to convert.

### **bool** **compile** (**pattern:string**)

Compiles a string to a regular expression. Returns true if the compilation was successful.

Parameter	Type	Description
pattern	string	The pattern to compile.

### **Array** **exec** (**text:string**)

Execute a regular expression.

Parameter	Type	Description
text	string	The string to match.

### **bool** **test** (**text:string**)

Execute a regular expression, and return true if there is a match.

Parameter	Type	Description
text	string	The string to match.

### **string** **toString** ()

Converts this RegExp object to a string.

Used in [String.match \(regexp:RegExp\)](#)  
[String.search \(search:RegExp\)](#)

Return **RegExp** [RegExp.RegExp \(pattern:string\)](#)

[Contents :: Index](#)

Class

# Socket

Creates a TCP/IP connection, or establishes a TCP/IP server.

QuickLinks [Socket](#), [close](#), [listen](#), [open](#), [poll](#), [read](#), [readln](#), [write](#), [writeln](#)

## Properties

	Property	Type	Access	Description
	connected	bool	readonly	When true, the connection is active.
	encoding	string	r/w	Sets or retrieves the name of the encoding used to transmit data. Typical values are "ASCII", "BINARY", or "UTF-8".
	eof	bool	readonly	When true, the receive buffer is empty.
	error	string	r/w	A message describing the most recent error. Setting this value clears any error message.
	host	string	readonly	The name of the remote computer when a connection is established. If the connection is shut down or does not exist, the property contains the empty string.
	timeout	number	r/w	The timeout in seconds to be applied to read or write operations. (default: <b>10</b> )

## Methods

**Socket Socket ()**

Creates a new Socket object.

**bool close ()**

Terminates the open connection.

**bool listen (port:number [, encoding:string=ASCII])**

Instructs the object to start listening for an incoming connection.

Parameter	Type	Description
port	number	The TCP/IP port number to listen on.
encoding	string	The encoding to be used for the connection (default: <b>ASCII</b> )

**bool open (host:string [, encoding:string=ASCII])**

Opens the connection for subsequent read/write operations.

Parameter	Type	Description
host	string	The server to connect to.
encoding	string	The encoding to be used for the connection (default: <b>ASCII</b> )

**Socket poll ()**

Checks a listening object for a new incoming connection.

**string read (count:number)**

Reads up to the specified number of characters from the connection. CR characters are ignored unless the encoding is set to "BINARY".

Parameter	Type	Description
count	number	The number of characters to read.

**string readln ()**

Reads one line of text up to the next line feed.

**bool write (text:string)**

Concatenates all arguments into a single string and writes that string to the connection.

Parameter	Type	Description
text	string	Any number of string values. All arguments are concatenated to form the string to be written. CRLF sequences are

converted to LFs unless the encoding is set to "BINARY".

**bool writeln (*text:string*)**

Concatenates all arguments into a single string, appends a LF character, and writes that string to the connection.

Parameter	Type	Description
text	string	Any number of string values. All arguments are concatenated to form the string to be written. CRLF sequences are converted to LFs unless the encoding is set to "BINARY".

Return **Socket** **Socket.Socket ()**  
**Socket** **Socket.poll ()**

[Contents](#) :: [Index](#)

Class

# String

Character strings. Each character is addressable by index.

## QuickLinks

[String](#), [anchor](#), [big](#), [blink](#), [bold](#), [charAt](#), [charCodeAt](#), [concat](#), [fixed](#), [fontcolor](#), [fontsize](#), [fromCharCode](#), [indexOf](#), [italics](#), [lastIndexOf](#), [link](#), [localeCompare](#), [match](#), [replace](#), [search](#), [slice](#), [small](#), [split](#), [strike](#), [sub](#), [substr](#), [substring](#), [sup](#), [toLocaleLowerCase](#), [toLocaleUpperCase](#), [toLowerCase](#), [toSource](#), [toString](#), [toUpperCase](#), [valueOf](#)

## Properties

Property	Type	Access	Description
length	number	readonly	The length of the string.

## Methods

### `String String (value:any)`

Returns a string representation of the value given as an argument.

Parameter	Type	Description
value	any	A number, variable, or object to convert to a string.

### `string anchor (name:string)`

Returns a string consisting of this string enclosed in a <a> tag.

Parameter	Type	Description
name	string	The text to be stored in the anchors' name attribute.

### `string big ()`

Returns a string consisting of this string enclosed in a <big> tag.

### `string blink ()`

Returns a string consisting of this string enclosed in a <blink> tag.

### `string bold ()`

Returns a string consisting of this string enclosed in a <b> tag.

**string charAt (index:number)**

Returns the character at the specified index.

Parameter	Type	Description
index	number	An integer between 0 and string.length -1, specifying the character to return.

**number charCodeAt (index:number)**

Returns the Unicode value of the character at the given index.

Parameter	Type	Description
index	number	An integer between 0 and string.length -1, specifying the character.

**string concat (value:string)**

If necessary, converts the one or more given values to strings.

Parameter	Type	Description
value	string	The values to be concatenated with the given string.

**string fixed ()**

Returns a string consisting of this string enclosed in a <tt> tag.

**string fontcolor (color:string)**

Returns a string consisting of this string enclosed in a <font> tag.

Parameter	Type	Description
color	string	The value to be stored in the tag's color attribute.

**string fontsize (size:number)**

Returns a string consisting of this string enclosed in a <font> tag.

Parameter	Type	Description
size	number	The value to be stored in the tag's size attribute.

**String fromCharCode (value1:number)**

Returns a string created by concatenation one or more characters specified as ASCII values.

Parameter	Type	Description
value1	number	One or more ASCII values.

**number indexOf (searchValue:string, offset:number)**

Returns the index within the string of the first occurrence of the specified string, starting the search at fromIndex if provided.

Parameter	Type	Description
searchValue	string	The string for which to search.
offset	number	The starting offset of the search.

**string italics ()**

Returns a string consisting of this string enclosed in a <i> tag.

**number lastIndexOf (searchValue:string, offset:number)**

Returns the index within the string of the last occurrence of the specified value.

Parameter	Type	Description
searchValue	string	The string for which to search.
offset	number	The starting offset of the search.

**string link (href:string)**

Returns a string consisting of this string enclosed in a <a> tag.

Parameter	Type	Description
href	string	The value to be stored in the tag's href attribute.

**number localeCompare (what:string)**

Performs a localized comparison of two strings.

Parameter	Type	Description
what	string	The string to compare with.

**Array match (regexp:RegExp)**

Matches a string against a regular expression.

Parameter	Type	Description
regexp	RegExp	The regular expression to use.

**string replace (what:any, with:string)**

Parameter	Type	Description
what	any	
with	string	

**number search (search:RegExp)**

Parameter	Type	Description
search	RegExp	

**string slice (startSlice:number, endSlice:number)**

Extracts a substring of the given string and returns it as a new string.

Parameter	Type	Description
startSlice	number	The index at which to begin extraction.
endSlice	number	The index at which to end extraction.

**string small ()**

Returns a string consisting of this string enclosed in a <small> tag.

**string split (delimiter:string, limit:number)**

Splits a string into a group of substrings, places those strings in an array, and returns the array.

Parameter	Type	Description
delimiter	string	Specifies the string to use for delimiting.
limit	number	

**string strike ()**

Returns a string consisting of this string enclosed in a <strike> tag.

**string sub ()**

Returns a string consisting of this string enclosed in a <sub> tag.

**string substr (start:number, length:number)**

Returns a string containing the characters beginning at the specified index, start, through the specified number of characters.

Parameter	Type	Description
start	number	Location at which to begin extracting characters.
length	number	(Optional) The number of characters to extract.

**string substring (indexA:number [, indexB:number])**

Returns a substring of the given string by extracting characters from indexA up to but not including indexB.

Parameter	Type	Description
indexA	number	The index to begin extracting.
indexB	number	(Optional) The index at which to end extraction.

**string sup ()**

Returns a string consisting of this string enclosed in a <sup> tag.

**string toLocaleLowerCase ()**

Returns a new string which contains all the characters of the original string converted to lowercase (localized).

**string toLocaleUpperCase ()**

Returns a new string which contains all the characters of the original string converted to uppercase (localized).

**string toLowerCase ()**

Returns a new string which contains all the characters of the original string converted to lowercase.

**string toSource ()**

Creates a string representation of this object that can be fed back to **eval()** to re-create an object. Works only with built-in classes.

**string** **toString** ()

Returns itself.

**string** **toUpperCase** ()

Returns a new string which contains all the characters of the original string converted to uppercase.

**string** **valueOf** ()

Returns itself.

Return **String** **String**.**String** (**value**:any)

**String** **String**.**fromCharCode** (**value1**:number)

[Contents](#) :: [Index](#)

Class

# UnitValue

QuickLinks [as](#), [convert](#)

## Properties

Property	Type	Access	Description
baseUnit	any	r/w	
baseUnit	<a href="#">UnitValue</a>	r/w	The base unit.
type	string	readonly	The unit name.
value	number	r/w	The numeric value.

## Methods

[UnitValue as \(unitName:string\)](#)

Return this instance as a different unit.

Parameter	Type	Description
unitName	string	The unit name.

[any convert \(unitName:string\)](#)

Convert this instance to a different unit.

Parameter	Type	Description
unitName	string	The unit name.

Element of [UnitValue.baseUnit](#)

Return **UnitValue UnitValue.as (unitName:string)**

[Contents :: Index](#)

Class

# XML

## QuickLinks

[addNamespace](#), [appendChild](#), [attribute](#), [attributes](#), [child](#), [childIndex](#), [children](#), [comments](#), [contains](#), [copy](#), [defaultSettings](#), [descendants](#), [elements](#), [hasComplexContent](#), [hasOwnProperty](#), [hasSimpleContent](#), [inScopeNamespaces](#), [insertChildAfter](#), [insertChildBefore](#), [length](#), [localName](#), [name](#), [namespace](#), [namespaceDeclarations](#), [nodeKind](#), [normalize](#), [parent](#), [prependChild](#), [processingInstructions](#), [propertyIsEnumerable](#), [removeNamespace](#), [replace](#), [setChildren](#), [setLocalName](#), [setName](#), [setNamespace](#), [setSettings](#), [settings](#), [text](#), [toString](#), [toXMLString](#), [xpath](#)

## Properties

Property	Type	Access	Description
ignoreComments	bool	r/w	
ignoreProcessingInstructions	bool	r/w	
ignoreWhitespace	bool	r/w	
prettyIndent	number	r/w	
prettyPrinting	bool	r/w	

## Methods

**void addNamespace (namespace:Namespace)**

Parameter	Type	Description
namespace	Namespace	

**XML appendChild (child:XML)**

Parameter	Type	Description
child	XML	

**XML attribute (name:string)**

Parameter	Type	Description
name	string	

**XML attributes ()**

**XML child** (*name:string*)

Parameter	Type	Description
name	string	

**number childIndex ()**

**XML children ()**

**XML comments ()**

**bool contains** (*xml:XML*)

Parameter	Type	Description
xml	XML	

**XML copy ()**

**Object defaultSettings ()**

**XML descendants** (*name:string*)

Parameter	Type	Description
name	string	

**XML elements** (*name:string*)

Parameter	Type	Description
name	string	

**bool hasComplexContent** (*name:string*)

Parameter	Type	Description
name	string	

**XML hasOwnProperty** (*name:string*)

Parameter	Type	Description
name	string	

**bool hasSimpleContent ()**

**Array inScopeNamespaces ()**

**void insertChildAfter** (*child1:XML, child2:XML*)

Parameter	Type	Description
child1	XML	
child2	XML	

**void insertChildBefore (child1:XML, child2:XML)**

Parameter	Type	Description
child1	XML	
child2	XML	

**number length ()**

**string localName ()**

**QName name ()**

**Namespace namespace ()**

**Array namespaceDeclarations ()**

**string nodeKind ()**

**XML normalize ()**

**XML parent ()**

**XML prependChild (value:XML)**

Parameter	Type	Description
value	XML	

**XML processingInstructions (name:string)**

Parameter	Type	Description
name	string	

**bool propertyIsEnumerable (name:string)**

Parameter	Type	Description
name	string	

**XML removeNamespace (namespace:Namespace)**

Parameter	Type	Description

namespace	Namespace
-----------	-----------

**XML replace (name:string, value:XML)**

Parameter	Type	Description
name	string	
value	XML	

**XML setChildren (children:XML)**

Parameter	Type	Description
children	XML	

**void setLocalName (name:string)**

Parameter	Type	Description
name	string	

**void setName (name:QName)**

Parameter	Type	Description
name	QName	

**void setNamespace (namespace:Namespace)**

Parameter	Type	Description
namespace	Namespace	

**any setSettings ()**

**Object settings ()**

**XML text ()**

**string toString ()**

**string toXMLString ()**

**XML xpath (expr:string)**

Parameter	Type	Description
expr	string	

Used in [XML.appendChild \(child:XML\)](#)  
[XML.contains \(xml:XML\)](#)  
[XML.insertChildAfter \(child1:XML, child2:XML\)](#)  
[XML.insertChildAfter \(child1:XML, child2:XML\)](#)  
[XML.insertChildBefore \(child1:XML, child2:XML\)](#)  
[XML.insertChildBefore \(child1:XML, child2:XML\)](#)  
[XML.prependChild \(value:XML\)](#)  
[XML.replace \(name:string, value:XML\)](#)  
[XML.setChildren \(children:XML\)](#)

Return [XML Dictionary.toXML \(prefix:string\)](#)  
[XML Reflection.toXML \(\)](#)  
[XML XML.appendChild \(child:XML\)](#)  
[XML XML.attribute \(name:string\)](#)  
[XML XML.attributes \(\)](#)  
[XML XML.child \(name:string\)](#)  
[XML XML.children \(\)](#)  
[XML XML.comments \(\)](#)  
[XML XML.copy \(\)](#)  
[XML XML.descendants \(name:string\)](#)  
[XML XML.elements \(name:string\)](#)  
[XML XML.hasOwnProperty \(name:string\)](#)  
[XML XML.normalize \(\)](#)  
[XML XML.parent \(\)](#)  
[XML XML.prependChild \(value:XML\)](#)  
[XML XML.processingInstructions \(name:string\)](#)  
[XML XML.removeNamespace \(namespace:Namespace\)](#)  
[XML XML.replace \(name:string, value:XML\)](#)  
[XML XML.setChildren \(children:XML\)](#)  
[XML XML.text \(\)](#)  
[XML XML.xpath \(expr:string\)](#)

[Contents :: Index](#)

# Index

A B C D E F G H I J K L M N O P Q R S T

Array

A B C D E F G H I J K L M N O P Q R S T

Boolean

A B C D E F G H I J K L M N O P Q R S T

Date

Dictionary

A B C D E F G H I J K L M N O P Q R S T

Error

A B C D E F G H I J K L M N O P Q R S T

File

Folder

Function

A B C D E F G H I J K L M N O P Q R S T

Math

A B C D E F G H I J K L M N O P Q R S T

Namespace

Number

A B C D E F G H I J K L M N O P Q R S T

Object

A B C D E F G H I J K L M N O P Q R S T

QName

A B C D E F G H I J K L M N O P Q R S T

Reflection

ReflectionInfo

RegExp

A B C D E F G H I J K L M N O P Q R S T

Socket

String

A B C D E F G H I J K L M N O P Q R S T

[UnitValue](#)

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [GHIJKL](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#)

[XML](#)

---

[Contents](#) :: [Index](#)