

Launcher Javascript API

3.1.6.0

Generated by Doxygen 1.7.4

Fri Nov 4 2011 15:34:31

Contents

1	Launcher Javascript API	1
1.1	Overview	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	Launcher Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Function Documentation	8
4.1.2.1	fileExists	8
4.1.2.2	getRegistryInt32	8
4.1.2.3	getRegistryString	8
4.1.2.4	getVersion	8
4.1.2.5	reboot	9
4.1.2.6	registryKeyPathExists	9
4.1.2.7	release	9
4.1.2.8	setRegistryInt32	9
4.1.2.9	setRegistryString	10
4.1.2.10	shellOpen	10
4.1.3	Member Data Documentation	10
4.1.3.1	instanceId	10
4.2	Process Class Reference	10

4.2.1	Detailed Description	11
4.2.2	Member Function Documentation	11
4.2.2.1	getApplicationName	11
4.2.2.2	getArguments	11
4.2.2.3	getElevate	11
4.2.2.4	getSha1	12
4.2.2.5	getShowWindow	12
4.2.2.6	getWorkingDirectory	12
4.2.2.7	launch	12
4.2.2.8	release	12
4.2.2.9	setApplicationName	12
4.2.2.10	setArguments	13
4.2.2.11	setElevate	13
4.2.2.12	setSha1	13
4.2.2.13	setShowWindow	13
4.2.2.14	setWorkingDirectory	13
4.2.3	Member Data Documentation	13
4.2.3.1	instanceId	13
4.2.3.2	onComplete	14
4.2.3.3	onExit	14
4.3	TextFileReader Class Reference	14
4.3.1	Detailed Description	14
4.3.2	Member Function Documentation	15
4.3.2.1	getBufferSize	15
4.3.2.2	getFileName	15
4.3.2.3	release	15
4.3.2.4	setBufferSize	15
4.3.2.5	setFileName	15
4.3.2.6	start	15
4.3.3	Member Data Documentation	15
4.3.3.1	instanceId	16
4.3.3.2	onComplete	16
4.3.3.3	onProgress	16
4.3.3.4	onRead	16

4.3.3.5	onStart	16
4.4	TextFileWriter Class Reference	16
4.4.1	Detailed Description	17
4.4.2	Member Function Documentation	17
4.4.2.1	close	17
4.4.2.2	open	17
4.4.2.3	release	17
4.4.2.4	write	17
4.4.3	Member Data Documentation	18
4.4.3.1	instanceId	18
5	File Documentation	19
5.1	launcher.js File Reference	19
5.1.1	Detailed Description	19
5.1.2	Function Documentation	19
5.1.2.1	createLauncher	19
5.2	process.js File Reference	19
5.2.1	Detailed Description	20
5.2.2	Function Documentation	20
5.2.2.1	createProcess	20
5.3	serialize.js File Reference	20
5.3.1	Detailed Description	20
5.3.2	Function Documentation	20
5.3.2.1	loadObjectFromFile	20
5.3.2.2	saveObjectToFile	21
5.4	textfilereader.js File Reference	21
5.4.1	Detailed Description	21
5.4.2	Function Documentation	21
5.4.2.1	createTextFileReader	21
5.5	textfilewriter.js File Reference	21
5.5.1	Detailed Description	22
5.5.2	Function Documentation	22
5.5.2.1	createTextFileWriter	22

Chapter 1

Launcher Javascript API

1.1 Overview

The launcher javascript API allows users to integrate with objects implemented using the launcher API.

The usage of this API is restricted to entities which have signed a license agreement with Solid State Networks, Inc.

Licenses are valid for only one PRODUCT usage. Please see the license agreement for further details.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Launcher (General Launcher functionality)	7
Process (Process launcher)	10
TextFileReader (Text file reader)	14
TextFileWriter (Text file writer)	16

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

launcher.js (File containing Launcher class and creation function)	19
process.js (File containing Process class and creation function)	19
serialize.js (File containing javascript object extensions for serialization) . . .	20
textfilereader.js (File containing TextFileReader class and creation function) .	21
textfilewriter.js (File containing TextFileWriter class and creation function) . . .	21

Chapter 4

Class Documentation

4.1 Launcher Class Reference

General [Launcher](#) functionality.

Public Member Functions

- bool [fileExists](#) (string fileName)
- int [getRegistryInt32](#) (string type, string keyPath, string name, int defaultValue)
- string [getRegistryString](#) (string type, string keyPath, string name, string defaultValue)
- string [getVersion](#) (string fileName, string defaultValue)
- void [reboot](#) ()
- bool [registryKeyPathExists](#) (string type, string keyPath)
- void [release](#) ()
- bool [setRegistryInt32](#) (string type, string keyPath, string name, int value)
- bool [setRegistryString](#) (string type, string keyPath, string name, string value)
- bool [shellOpen](#) (string path)

Public Attributes

- string [instanceId](#)

4.1.1 Detailed Description

General [Launcher](#) functionality.

4.1.2 Member Function Documentation

4.1.2.1 bool Launcher.fileExists (string *fileName*)

Check if a file exists

Parameters

<i>fileName</i>	filename to check.
-----------------	--------------------

Returns

true if successful, false otherwise

4.1.2.2 int Launcher.getRegistryInt32 (string *type*, string *keyPath*, string *name*, int *defaultValue*)

returns a registry value as an integer

Parameters

<i>type</i>	type of hive.
<i>keyPath</i>	path of the key.
<i>name</i>	name of value.
<i>defaultValue</i>	default value returned if unable to read key.

Returns

value of the key.

4.1.2.3 string Launcher.getRegistryString (string *type*, string *keyPath*, string *name*, string *defaultValue*)

returns a registry value as a string

Parameters

<i>type</i>	type of hive.
<i>keyPath</i>	path of the key.
<i>name</i>	name of value.
<i>defaultValue</i>	default value returned if unable to read key.

Returns

value of the key.

4.1.2.4 string Launcher.getVersion (string *fileName*, string *defaultValue*)

returns the version of the filename

Parameters

<i>fileName</i>	name of the file to get the version of.
<i>defaultValue</i>	default value returned if unable to read version.

Returns

value of the version.

4.1.2.5 void Launcher.reboot ()

Reboots the machine

4.1.2.6 bool Launcher.registryKeyPathExists (string type, string keyPath)

Check if a registry key path exists

Parameters

<i>type</i>	type of hive.
<i>keyPath</i>	path of the key.

Returns

true if exists, false otherwise

4.1.2.7 void Launcher.release ()

releases the object

4.1.2.8 bool Launcher.setRegistryInt32 (string type, string keyPath, string name, int value)

sets an integer in the registry

Parameters

<i>type</i>	type of hive.
<i>keyPath</i>	path of the key.
<i>name</i>	name of value.
<i>value</i>	value of key.

Returns

true if successful, false otherwise

4.1.2.9 `bool Launcher.setRegistryString (string type, string keyPath, string name, string value)`

sets a string in the registry

Parameters

<i>type</i>	type of hive.
<i>keyPath</i>	path of the key.
<i>name</i>	name of value.
<i>value</i>	value of key.

Returns

true if successful, false otherwise

4.1.2.10 `bool Launcher.shellOpen (string path)`

Launch the process

Parameters

<i>path</i>	string to url/filename to open.
-------------	---------------------------------

Returns

true if successful, false otherwise

4.1.3 Member Data Documentation

4.1.3.1 `string Launcher.instanceId`

Instance id used to bind proxy object to native object

The documentation for this class was generated from the following file:

- [launcher.js](#)

4.2 Process Class Reference

[Process](#) launcher.

Public Member Functions

- string [getApplicationName](#) ()
- string [getArguments](#) ()
- bool [getElevate](#) ()

- string [getSha1](#) ()
- bool [getShowWindow](#) ()
- string [getWorkingDirectory](#) ()
- bool [launch](#) ()
- void [release](#) ()
- void [setApplicationName](#) (string value)
- void [setArguments](#) (string value)
- void [setElevate](#) (bool value)
- void [setSha1](#) (string value)
- void [setShowWindow](#) (bool value)
- void [setWorkingDirectory](#) (string value)

Public Attributes

- string [instanceId](#)
- event [onComplete](#)
- event [onExit](#)

4.2.1 Detailed Description

[Process](#) launcher.

4.2.2 Member Function Documentation

4.2.2.1 string [Process.getApplicationName](#) ()

gets the application to launch

Returns

Application to launch

4.2.2.2 string [Process.getArguments](#) ()

gets the arguments for the process

Returns

Arguments for the process

4.2.2.3 bool [Process.getElevate](#) ()

gets the true if process requires elevatation, false otherwise

Returns

true if process requires elevatation, false otherwise

4.2.2.4 string Process.getSha1 ()

gets the sha1 value of application, blank to skip check

Returns

sha1 value of application, blank to skip check

4.2.2.5 bool Process.getShowWindow ()

gets the true if process is shown, false otherwise

Returns

true if process is shown, false otherwise

4.2.2.6 string Process.getWorkingDirectory ()

gets the the full path to the current directory for the process

Returns

The full path to the current directory for the process

4.2.2.7 bool Process.launch ()

Launch the process

Returns

true if successful, false otherwise

4.2.2.8 void Process.release ()

releases the object

4.2.2.9 void Process.setApplicationName (string value)

sets the application to launch

Parameters

<i>value</i>	Application to launch
--------------	-----------------------

4.2.2.10 void Process.setArguments (string *value*)

sets the arguments for the process

Parameters

<i>value</i>	Arguments for the process
--------------	---------------------------

4.2.2.11 void Process.setElevate (bool *value*)

sets the true if process requires elevatation, false otherwise

Parameters

<i>value</i>	true if process requires elevatation, false otherwise
--------------	---

4.2.2.12 void Process.setSha1 (string *value*)

sets the sha1 value of application, blank to skip check

Parameters

<i>value</i>	sha1 value of application, blank to skip check
--------------	--

4.2.2.13 void Process.setShowWindow (bool *value*)

sets the true if process is shown, false otherwise

Parameters

<i>value</i>	true if process is shown, false otherwise
--------------	---

4.2.2.14 void Process.setWorkingDirectory (string *value*)

sets the the full path to the current directory for the process

Parameters

<i>value</i>	The full path to the current directory for the process
--------------	--

4.2.3 Member Data Documentation**4.2.3.1 string Process.instanceId**

Instance id used to bind proxy object to native object

4.2.3.2 event `Process.onComplete`

Triggered when the process is complete.

Parameters

<i>successful</i>	true if the process was successful, false otherwise.
-------------------	--

4.2.3.3 event `Process.onExit`

Triggered when the process is exited.

Parameters

<i>exitCode</i>	Application's exit code
-----------------	-------------------------

The documentation for this class was generated from the following file:

- [process.js](#)

4.3 TextFileReader Class Reference

Text file reader.

Public Member Functions

- int [getBufferSize](#) ()
- string [getFileName](#) ()
- void [release](#) ()
- void [setBufferSize](#) (int value)
- void [setFileName](#) (string value)
- void [start](#) ()

Public Attributes

- string [instanceId](#)
- event [onComplete](#)
- event [onProgress](#)
- event [onRead](#)
- event [onStart](#)

4.3.1 Detailed Description

Text file reader.

4.3.2 Member Function Documentation

4.3.2.1 int TextFileReader.getBufferSize ()

gets the determines the max size of the buffer being read into

Returns

Determines the max size of the buffer being read into

4.3.2.2 string TextFileReader.getFileName ()

gets the file name where url is saved

Returns

File name where url is saved

4.3.2.3 void TextFileReader.release ()

releases the object

4.3.2.4 void TextFileReader.setBufferSize (int *value*)

sets the determines the max size of the buffer being read into

Parameters

<i>value</i>	Determines the max size of the buffer being read into
--------------	---

4.3.2.5 void TextFileReader.setFileName (string *value*)

sets the file name where url is saved

Parameters

<i>value</i>	File name where url is saved
--------------	------------------------------

4.3.2.6 void TextFileReader.start ()

Start file reading

4.3.3 Member Data Documentation

4.3.3.1 string `TextFileReader.instanceId`

Instance id used to bind proxy object to native object

4.3.3.2 event `TextFileReader.onComplete`

Triggered when file reader is complete.

Parameters

<i>successful</i>	true if the file reader was successful, false otherwise.
-------------------	--

4.3.3.3 event `TextFileReader.onProgress`

Triggered when file reader has made progress.

Parameters

<i>percent</i>	(-1.0: Still calculating) (0.0 to 1.0: Percent completed)
----------------	---

4.3.3.4 event `TextFileReader.onRead`

Triggered when file reader reads new bytes.

Parameters

<i>text</i>	Bytes read from file.
-------------	-----------------------

4.3.3.5 event `TextFileReader.onStart`

Triggered when file reader is started.

The documentation for this class was generated from the following file:

- [textfilereader.js](#)

4.4 TextFileWriter Class Reference

Text file writer.

Public Member Functions

- bool [close](#) ()
- bool [open](#) (string fileName)

- void [release](#) ()
- int [write](#) (string value)

Public Attributes

- string [instanceld](#)

4.4.1 Detailed Description

Text file writer.

4.4.2 Member Function Documentation

4.4.2.1 bool TextFileWriter.close ()

Close file

Returns

true if successful, false otherwise.

4.4.2.2 bool TextFileWriter.open (string *fileName*)

Open file for writing

Parameters

<i>fileName</i>	name of the file to close.
-----------------	----------------------------

Returns

true if successful, false otherwise.

4.4.2.3 void TextFileWriter.release ()

releases the object

4.4.2.4 int TextFileWriter.write (string *value*)

Write string to file

Parameters

<i>value</i>	string to write to file.
--------------	--------------------------

Returns

number of bytes written to file.

4.4.3 Member Data Documentation**4.4.3.1 string `TextFileWriter.instanceId`**

Instance id used to bind proxy object to native object

The documentation for this class was generated from the following file:

- [textfilewriter.js](#)

Chapter 5

File Documentation

5.1 launcher.js File Reference

File containing [Launcher](#) class and creation function.

Classes

- class [Launcher](#)
General [Launcher](#) functionality.

Functions

- void [createLauncher](#) ()

5.1.1 Detailed Description

File containing [Launcher](#) class and creation function.

5.1.2 Function Documentation

5.1.2.1 void createLauncher ()

Create instance of launcher

5.2 process.js File Reference

File containing [Process](#) class and creation function.

Classes

- class [Process](#)
Process launcher.

Functions

- void [createProcess](#) ()

5.2.1 Detailed Description

File containing [Process](#) class and creation function.

5.2.2 Function Documentation

5.2.2.1 void [createProcess](#) ()

Create instance of process

5.3 [serialize.js](#) File Reference

File containing javascript object extensions for serialization.

Functions

- void [loadObjectFromFile](#) (string fileName, event onLoadComplete)
- bool [saveObjectToFile](#) (object obj, string fileName)

5.3.1 Detailed Description

File containing javascript object extensions for serialization.

5.3.2 Function Documentation

5.3.2.1 void [loadObjectFromFile](#) (string *fileName*, event *onLoadComplete*)

load object from a file

Parameters

<i>fileName</i>	location for load
<i>onLoad-Complete</i>	function called when load completed

5.3.2.2 bool saveObjectToFile (object *obj*, string *fileName*)

save an object to a file

Parameters

<i>obj</i>	object to save
<i>fileName</i>	location for save

Returns

true if the object is saved or false otherwise

5.4 textfilereader.js File Reference

File containing [TextFileReader](#) class and creation function.

Classes

- class [TextFileReader](#)

Text file reader.

Functions

- void [createTextFileReader](#) ()

5.4.1 Detailed Description

File containing [TextFileReader](#) class and creation function.

5.4.2 Function Documentation

5.4.2.1 void createTextFileReader ()

Create instance of textFileReader

5.5 textfilewriter.js File Reference

File containing [TextFileWriter](#) class and creation function.

Classes

- class [TextFileWriter](#)
Text file writer.

Functions

- void [createTextFileWriter](#) ()

5.5.1 Detailed Description

File containing [TextFileWriter](#) class and creation function.

5.5.2 Function Documentation

5.5.2.1 void [createTextFileWriter](#) ()

Create instance of textFileWriter

Index

- close
 - TextFileWriter, 17
- createLauncher
 - launcher.js, 19
- createProcess
 - process.js, 20
- createTextFileReader
 - textfilereader.js, 21
- createTextFileWriter
 - textfilewriter.js, 22
- fileExists
 - Launcher, 8
- getApplicationName
 - Process, 11
- getArguments
 - Process, 11
- getBufferSize
 - TextFileReader, 15
- getElevate
 - Process, 11
- getFileName
 - TextFileReader, 15
- getRegistryInt32
 - Launcher, 8
- getRegistryString
 - Launcher, 8
- getSha1
 - Process, 11
- getShowWindow
 - Process, 12
- getVersion
 - Launcher, 8
- getWorkingDirectory
 - Process, 12
- instanceId
 - Launcher, 10
 - Process, 13
 - TextFileReader, 15
- TextFileWriter, 18
- launch
 - Process, 12
- Launcher, 7
 - fileExists, 8
 - getRegistryInt32, 8
 - getRegistryString, 8
 - getVersion, 8
 - instanceId, 10
 - reboot, 9
 - registryKeyPathExists, 9
 - release, 9
 - setRegistryInt32, 9
 - setRegistryString, 9
 - shellOpen, 10
- launcher.js, 19
 - createLauncher, 19
- loadObjectFromFile
 - serialize.js, 20
- onComplete
 - Process, 13
 - TextFileReader, 16
- onExit
 - Process, 14
- onProgress
 - TextFileReader, 16
- onRead
 - TextFileReader, 16
- onStart
 - TextFileReader, 16
- open
 - TextFileWriter, 17
- Process, 10
 - getApplicationName, 11
 - getArguments, 11
 - getElevate, 11
 - getSha1, 11
 - getShowWindow, 12

- getWorkingDirectory, [12](#)
- instanceId, [13](#)
- launch, [12](#)
- onComplete, [13](#)
- onExit, [14](#)
- release, [12](#)
- setApplicationName, [12](#)
- setArguments, [12](#)
- setElevate, [13](#)
- setSha1, [13](#)
- setShowWindow, [13](#)
- setWorkingDirectory, [13](#)
- process.js, [19](#)
 - createProcess, [20](#)
- reboot
 - Launcher, [9](#)
- registryKeyPathExists
 - Launcher, [9](#)
- release
 - Launcher, [9](#)
 - Process, [12](#)
 - TextFileReader, [15](#)
 - TextFileWriter, [17](#)
- saveObjectToFile
 - serialize.js, [21](#)
- serialize.js, [20](#)
 - loadObjectFromFile, [20](#)
 - saveObjectToFile, [21](#)
- setApplicationName
 - Process, [12](#)
- setArguments
 - Process, [12](#)
- setBufferSize
 - TextFileReader, [15](#)
- setElevate
 - Process, [13](#)
- setFileName
 - TextFileReader, [15](#)
- setRegistryInt32
 - Launcher, [9](#)
- setRegistryString
 - Launcher, [9](#)
- setSha1
 - Process, [13](#)
- setShowWindow
 - Process, [13](#)
- setWorkingDirectory
 - Process, [13](#)
- shellOpen
 - Launcher, [10](#)
- start
 - TextFileReader, [15](#)
- TextFileReader, [14](#)
 - getBufferSize, [15](#)
 - getFileName, [15](#)
 - instanceId, [15](#)
 - onComplete, [16](#)
 - onProgress, [16](#)
 - onRead, [16](#)
 - onStart, [16](#)
 - release, [15](#)
 - setBufferSize, [15](#)
 - setFileName, [15](#)
 - start, [15](#)
- textfilereader.js, [21](#)
 - createTextFileReader, [21](#)
- TextFileWriter, [16](#)
 - close, [17](#)
 - instanceId, [18](#)
 - open, [17](#)
 - release, [17](#)
 - write, [17](#)
- textfilewriter.js, [21](#)
 - createTextFileWriter, [22](#)
- write
 - TextFileWriter, [17](#)