Launcher Javascript API

3.0.5.0

Generated by Doxygen 1.7.4

Sat Aug 6 2011 16:09:40

Contents

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:

Chapter 3

File Index

3.1 File List

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

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 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

fileName	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

type	type of hive.
keyPath	path of the key.
name	name of value.
defaultValue	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

type	type of hive.
keyPath	path of the key.
name	name of value.
defaultValue	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

fileName	name of the file to get the version of.
defaultValue	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

type	type of hive.
keyPath	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.shellOpen (string *path*)

Launch the process

Parameters

path	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

value	Application to launch

4.2.2.10 void Process.setArguments (string *value*)

sets the arguments for the process

Parameters

value	Arguments for the process

4.2.2.11 void Process.setElevate (bool *value*)

sets the true if process requires elevatation, false otherwise

Parameters

value	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

value	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

value	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

value	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

successful	true if the process was successful, false otherwise.

4.2.3.3 event Process.onExit

Triggered when the process is exited.

Parameters

exitCode	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

value	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

value	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

successful	true if the file reader was successful, false otherwise.

4.3.3.3 event TextFileReader.onProgress

Triggered when file reader has made progress.

Parameters

percent	(-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

text	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 instanceId

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

fileName	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

value	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

fileName	location for load
onLoadComplete	function called when load completed

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

save an object to a file

Parameters

obj	object to save
fileName	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