

# Launcher Javascript API

## 3.0.5.0

Generated by Doxygen 1.7.4

Sat Aug 6 2011 16:09:40



# Contents

<b>1</b>	<b>Launcher Javascript API</b>	<b>1</b>
1.1	Overview . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	Launcher Class Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Member Function Documentation . . . . .	7
4.1.2.1	fileExists . . . . .	7
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	shellOpen . . . . .	9
4.1.3	Member Data Documentation . . . . .	9
4.1.3.1	instanceId . . . . .	9
4.2	Process Class Reference . . . . .	10
4.2.1	Detailed Description . . . . .	10
4.2.2	Member Function Documentation . . . . .	10

4.2.2.1	<a href="#">getApplicationName</a>	10
4.2.2.2	<a href="#">getArguments</a>	11
4.2.2.3	<a href="#">getElevate</a>	11
4.2.2.4	<a href="#">getSha1</a>	11
4.2.2.5	<a href="#">getShowWindow</a>	11
4.2.2.6	<a href="#">getWorkingDirectory</a>	11
4.2.2.7	<a href="#">launch</a>	11
4.2.2.8	<a href="#">release</a>	12
4.2.2.9	<a href="#">setApplicationName</a>	12
4.2.2.10	<a href="#">setArguments</a>	12
4.2.2.11	<a href="#">setElevate</a>	12
4.2.2.12	<a href="#">setSha1</a>	12
4.2.2.13	<a href="#">setShowWindow</a>	12
4.2.2.14	<a href="#">setWorkingDirectory</a>	13
4.2.3	<a href="#">Member Data Documentation</a>	13
4.2.3.1	<a href="#">instanceId</a>	13
4.2.3.2	<a href="#">onComplete</a>	13
4.2.3.3	<a href="#">onExit</a>	13
4.3	<a href="#">TextFileReader Class Reference</a>	13
4.3.1	<a href="#">Detailed Description</a>	14
4.3.2	<a href="#">Member Function Documentation</a>	14
4.3.2.1	<a href="#">getBufferSize</a>	14
4.3.2.2	<a href="#">getFileName</a>	14
4.3.2.3	<a href="#">release</a>	14
4.3.2.4	<a href="#">setBufferSize</a>	14
4.3.2.5	<a href="#">setFileName</a>	15
4.3.2.6	<a href="#">start</a>	15
4.3.3	<a href="#">Member Data Documentation</a>	15
4.3.3.1	<a href="#">instanceId</a>	15
4.3.3.2	<a href="#">onComplete</a>	15
4.3.3.3	<a href="#">onProgress</a>	15
4.3.3.4	<a href="#">onRead</a>	15
4.3.3.5	<a href="#">onStart</a>	15
4.4	<a href="#">TextFileWriter Class Reference</a>	16

4.4.1	Detailed Description	16
4.4.2	Member Function Documentation	16
4.4.2.1	close	16
4.4.2.2	open	16
4.4.2.3	release	17
4.4.2.4	write	17
4.4.3	Member Data Documentation	17
4.4.3.1	instanceId	17
<b>5</b>	<b>File Documentation</b>	<b>19</b>
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:

<a href="#">Launcher</a> (General <a href="#">Launcher</a> functionality ) . . . . .	7
<a href="#">Process</a> ( <a href="#">Process</a> launcher ) . . . . .	10
<a href="#">TextFileReader</a> (Text file reader ) . . . . .	13
<a href="#">TextFileWriter</a> (Text file writer ) . . . . .	16



## Chapter 3

# File Index

### 3.1 File List

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

<a href="#">launcher.js</a> (File containing <a href="#">Launcher</a> class and creation function ) . . . . .	19
<a href="#">process.js</a> (File containing <a href="#">Process</a> class and creation function ) . . . . .	19
<a href="#">serialize.js</a> (File containing javascript object extensions for serialization ) . . .	20
<a href="#">textfilereader.js</a> (File containing <a href="#">TextFileReader</a> class and creation function ) .	21
<a href="#">textfilewriter.js</a> (File containing <a href="#">TextFileWriter</a> 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 [shellOpen](#) (string path)

#### Public Attributes

- string [instanceld](#)

#### 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.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 [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

<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, 16

createLauncher  
    launcher.js, 19

createProcess  
    process.js, 20

createTextFileReader  
    textfilereader.js, 21

createTextFileWriter  
    textfilewriter.js, 22

fileExists  
    Launcher, 7

getApplicationName  
    Process, 10

getArguments  
    Process, 10

getBufferSize  
    TextFileReader, 14

getElevate  
    Process, 11

getFileName  
    TextFileReader, 14

getRegistryInt32  
    Launcher, 8

getRegistryString  
    Launcher, 8

getSha1  
    Process, 11

getShowWindow  
    Process, 11

getVersion  
    Launcher, 8

getWorkingDirectory  
    Process, 11

instanceId  
    Launcher, 9  
    Process, 13  
    TextFileReader, 15

TextFileWriter, 17

launch  
    Process, 11

Launcher, 7  
    fileExists, 7  
    getRegistryInt32, 8  
    getRegistryString, 8  
    getVersion, 8  
    instanceId, 9  
    reboot, 9  
    registryKeyPathExists, 9  
    release, 9  
    shellOpen, 9

launcher.js, 19  
    createLauncher, 19

loadObjectFromFile  
    serialize.js, 20

onComplete  
    Process, 13  
    TextFileReader, 15

onExit  
    Process, 13

onProgress  
    TextFileReader, 15

onRead  
    TextFileReader, 15

onStart  
    TextFileReader, 15

open  
    TextFileWriter, 16

Process, 10  
    getApplicationName, 10  
    getArguments, 10  
    getElevate, 11  
    getSha1, 11  
    getShowWindow, 11  
    getWorkingDirectory, 11  
    instanceId, 13

- launch, 11
- onComplete, 13
- onExit, 13
- release, 11
- setApplicationName, 12
- setArguments, 12
- setElevate, 12
- setSha1, 12
- setShowWindow, 12
- setWorkingDirectory, 12
- process.js, 19
  - createProcess, 20
- reboot
  - Launcher, 9
- registryKeyPathExists
  - Launcher, 9
- release
  - Launcher, 9
  - Process, 11
  - TextFileReader, 14
  - TextFileWriter, 16
- saveObjectToFile
  - serialize.js, 21
- serialize.js, 20
  - loadObjectFromFile, 20
  - saveObjectToFile, 21
- setApplicationName
  - Process, 12
- setArguments
  - Process, 12
- setBufferSize
  - TextFileReader, 14
- setElevate
  - Process, 12
- setFileName
  - TextFileReader, 14
- setSha1
  - Process, 12
- setShowWindow
  - Process, 12
- setWorkingDirectory
  - Process, 12
- shellOpen
  - Launcher, 9
- start
  - TextFileReader, 15
- TextFileReader, 13
  - getBufferSize, 14
  - getFileName, 14
  - instanceId, 15
  - onComplete, 15
  - onProgress, 15
  - onRead, 15
  - onStart, 15
  - release, 14
  - setBufferSize, 14
  - setFileName, 14
  - start, 15
- textfilereader.js, 21
  - createTextFileReader, 21
- TextFileWriter, 16
  - close, 16
  - instanceId, 17
  - open, 16
  - release, 16
  - write, 17
- textfilewriter.js, 21
  - createTextFileWriter, 22
- write
  - TextFileWriter, 17