

# Interoperability API

## 3.0.5.0

Generated by Doxygen 1.7.4

Sat Aug 6 2011 16:12:06



# Contents

<b>1</b>	<b>Interoperability 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	InteropHostStruct Struct Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
<b>5</b>	<b>File Documentation</b>	<b>9</b>
5.1	interop.h File Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.1.2	Typedef Documentation . . . . .	10
5.1.2.1	AppExpandStringFunc . . . . .	10
5.1.2.2	AppGetConfigFunc . . . . .	10
5.1.2.3	AppGetConfigSizeFunc . . . . .	10
5.1.2.4	InteropExecuteCallback . . . . .	10
5.1.2.5	InteropInvokeInstanceCallback . . . . .	11
5.1.2.6	InteropProcessInstanceCallback . . . . .	11
5.1.2.7	InteropRemoveInstanceCallback . . . . .	11
5.2	interopstub.h File Reference . . . . .	12
5.2.1	Detailed Description . . . . .	12
5.2.2	Function Documentation . . . . .	12

5.2.2.1	<a href="#">Interop_CreateInstance</a>	12
5.2.2.2	<a href="#">Interop_Init</a>	13
5.2.2.3	<a href="#">Interop_Remove</a>	13
5.2.2.4	<a href="#">Interop_SetOption</a>	13

## Chapter 1

# Interoperability API

### 1.1 Overview

The interoperability API allows developers to integrate with the Solid State Networks host.

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:

[InteropHostStruct](#) . . . . . 7





# Chapter 3

## File Index

### 3.1 File List

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

<a href="#">interop.h</a> . . . . .	9
<a href="#">interopstub.h</a> . . . . .	12



## Chapter 4

# Class Documentation

### 4.1 InteropHostStruct Struct Reference

#### Public Attributes

- [AppExpandStringFunc](#) [AppExpandString](#)  
*String expansion function.*
- [AppGetConfigSizeFunc](#) [AppGetConfigSize](#)  
*Size of app.config.xml.*
- [AppGetConfigFunc](#) [AppGetConfig](#)  
*Retrieve app.config.xml.*

#### 4.1.1 Detailed Description

Structure used to store global information about the interop host

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

- [interop.h](#)



## Chapter 5

# File Documentation

### 5.1 interop.h File Reference

#### Classes

- struct [InteropHostStruct](#)

#### Defines

- #define [MAX\\_JSON\\_STRINGLENGTH](#) (1024 \* 8)  
*Max JSON string length supported.*

#### Typedefs

- typedef int32(\* [AppExpandStringFunc](#) )(char \*String, char \*ResultString, int32 ResultStringLength)
- typedef int32(\* [AppGetConfigSizeFunc](#) )(int32 \*BufferSize)
- typedef int32(\* [AppGetConfigFunc](#) )(uint8 \*BufferPtr, int32 BufferSize)
- typedef int32(\* [InteropExecuteCallback](#) )(void \*UserPtr, char \*InstanceId, char \*CallbackInfo, char \*ResultString, int32 ResultStringLength)
- typedef int32(\* [InteropRemoveInstanceCallback](#) )(void \*UserPtr, char \*InstanceId)
- typedef int32(\* [InteropInvokeInstanceCallback](#) )(void \*UserPtr, char \*InstanceId, char \*String, char \*ResultString, int32 ResultStringLength)
- typedef int32(\* [InteropProcessInstanceCallback](#) )(void \*UserPtr, char \*InstanceId, int32 \*Continue)

#### 5.1.1 Detailed Description

Interop functions and callbacks

## 5.1.2 Typedef Documentation

### 5.1.2.1 `typedef int32(* AppExpandStringFunc)(char *String, char *ResultString, int32 ResultStringLength)`

Function definition for string expansion

#### Parameters

<i>String</i>	original string requiring expansion
<i>ResultString</i>	string buffer with the expanded version
<i>Result-StringLength</i>	maximum length of result string

#### Returns

TRUE on success, FALSE otherwise

### 5.1.2.2 `typedef int32(* AppGetConfigFunc)(uint8 *BufferPtr, int32 BufferSize)`

Function definition for retrieving the app.config.xml file

#### Parameters

<i>BufferPtr</i>	string buffer containing the app.config.xml file
<i>BufferSize</i>	maximum size of the buffer contained in buffer ptr

#### Returns

TRUE on success, FALSE otherwise

### 5.1.2.3 `typedef int32(* AppGetConfigSizeFunc)(int32 *BufferSize)`

Function definition for retrieving the size of the app.config.xml file

#### Parameters

<i>BufferSize</i>	the size of the xml file
-------------------	--------------------------

#### Returns

TRUE on success, FALSE otherwise

### 5.1.2.4 `typedef int32(* InteropExecuteCallback)(void *UserPtr, char *Instanceld, char *CallbackInfo, char *ResultString, int32 ResultStringLength)`

Function definition for script execution

**Parameters**

<i>UserPtr</i>	pointer stored when callback is setup
<i>Instanceld</i>	id of the object to invoke
<i>CallbackInfo</i>	json string representing the method invocation
<i>ResultString</i>	json string representing the return value
<i>Result-StringLength</i>	maximum length of the ResultString

**Returns**

TRUE on success, FALSE otherwise

**5.1.2.5** `typedef int32(* InteropInvokeInstanceCallback)(void *UserPtr, char *Instanceld, char *String, char *ResultString, int32 ResultStringLength)`

Function definition for native execution

**Parameters**

<i>UserPtr</i>	pointer stored when callback is setup
<i>Instanceld</i>	id of the object to invoke
<i>String</i>	json string representing the method invocation
<i>ResultString</i>	json string representing the return value
<i>Result-StringLength</i>	maximum length of the ResultString

**Returns**

TRUE on success, FALSE otherwise

**5.1.2.6** `typedef int32(* InteropProcessInstanceCallback)(void *UserPtr, char *Instanceld, int32 *Continue)`

Function definition for instance processing

**Parameters**

<i>UserPtr</i>	pointer stored when callback is setup
<i>Instanceld</i>	id of the object to process
<i>Continue</i>	object still has data waiting to be processed

**Returns**

TRUE on success, FALSE otherwise

**5.1.2.7** `typedef int32(* InteropRemoveInstanceCallback)(void *UserPtr, char *Instanceld)`

Function definition for instance removal

**Parameters**

<i>UserPtr</i>	pointer stored when callback is setup
<i>Instanceld</i>	id of the object to remove

**Returns**

TRUE on success, FALSE otherwise

**5.2 interopstub.h File Reference****Functions**

- int32 [Interop\\_CreateInstance](#) (char \**TypeName*, char \**Instanceld*, int32 *InstanceldLength*, void \**ExecuteUserPtr*, [InteropExecuteCallback](#) *Execute*, [InteropInvokeInstanceCallback](#) \**InvokeInstance*, [InteropRemoveInstanceCallback](#) \**RemoveInstance*, [InteropProcessInstanceCallback](#) \**ProcessInstance*, void \*\**UserPtr*)
- int32 [Interop\\_SetOption](#) (char \**Key*, char \**Value*)
- int32 [Interop\\_Init](#) ([InteropHostStruct](#) \**InteropHost*)
- int32 [Interop\\_Remove](#) ()

**5.2.1 Detailed Description**

Interop implementation functions

**5.2.2 Function Documentation**

**5.2.2.1** int32 [Interop\\_CreateInstance](#) ( char \* *TypeName*, char \* *Instanceld*, int32 *InstanceldLength*, void \* *ExecuteUserPtr*, [InteropExecuteCallback](#) *Execute*, [InteropInvokeInstanceCallback](#) \* *InvokeInstance*, [InteropRemoveInstanceCallback](#) \* *RemoveInstance*, [InteropProcessInstanceCallback](#) \* *ProcessInstance*, void \*\* *UserPtr* )

Called when an instance is created

**Parameters**

<i>TypeName</i>	type of object to be created
<i>Instanceld</i>	id of the object to be created
<i>Instancel- dLength</i>	maximum length of the <i>Instanceld</i>
<i>Execu- teUserPtr</i>	user pointer stored with execute callback
<i>Execute</i>	execute callback
<i>InvokeIn- stance</i>	pointer to the instance invocation callback function
<i>RemoveIn- stance</i>	pointer to the instance remove callback function



<i>ProcessInstance</i>	pointer to the instance process callback function
<i>UserPtr</i>	pointer associated with instance id

**Returns**

TRUE on success, FALSE otherwise

**5.2.2.2 int32 Interop\_Init ( InteropHostStruct \* InteropHost )**

Initializes interop functions

**Parameters**

<i>InteropHost</i>	information about the interop host
--------------------	------------------------------------

**Returns**

TRUE on success, FALSE otherwise

**5.2.2.3 int32 Interop\_Remove ( )**

Uninitialize interop functions

**Returns**

TRUE on success, FALSE otherwise

**5.2.2.4 int32 Interop\_SetOption ( char \* Key, char \* Value )**

Set an interop option by key

**Parameters**

<i>Key</i>	identifier of option
<i>Value</i>	value of option

**Returns**

TRUE on success, FALSE otherwise

# Index

AppExpandStringFunc  
    interop.h, [10](#)

AppGetConfigFunc  
    interop.h, [10](#)

AppGetConfigSizeFunc  
    interop.h, [10](#)

interop.h, [9](#)  
    AppExpandStringFunc, [10](#)  
    AppGetConfigFunc, [10](#)  
    AppGetConfigSizeFunc, [10](#)  
    InteropExecuteCallback, [10](#)  
    InteropInvokeInstanceCallback, [11](#)  
    InteropProcessInstanceCallback, [11](#)  
    InteropRemoveInstanceCallback, [11](#)

Interop\_CreateInstance  
    interopstub.h, [12](#)

Interop\_Init  
    interopstub.h, [13](#)

Interop\_Remove  
    interopstub.h, [13](#)

Interop\_SetOption  
    interopstub.h, [13](#)

InteropExecuteCallback  
    interop.h, [10](#)

InteropHostStruct, [7](#)

InteropInvokeInstanceCallback  
    interop.h, [11](#)

InteropProcessInstanceCallback  
    interop.h, [11](#)

InteropRemoveInstanceCallback  
    interop.h, [11](#)

interopstub.h, [12](#)  
    Interop\_CreateInstance, [12](#)  
    Interop\_Init, [13](#)  
    Interop\_Remove, [13](#)  
    Interop\_SetOption, [13](#)