Code_generator

Generated by Doxygen 1.8.15

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Namespace Documentation

3.1 Package common

Classes

- enum ErrorCode
- class Features

3.1.1 Detailed Description

Common information that needs to be accessed across all the project

Author

Miguel Diaz

Version

0.1

3.2 Package configurator

Classes

- class ConfigurationFile
- class PinConf

3.2.1 Detailed Description

Configuration classes

Author

Miguel Diaz

Version

0.1

3.3 Package framework

Classes

- class CodeGenerator
- class Common
- class **GpioGenerator**

3.3.1 Detailed Description

Framework information

Author

H112943

Version

0.1

3.4 Package gui

Classes

- class AboutWindow
- class GpioConfWindow
- class MainGui
- class MainWindow
- class Messages

3.4.1 Detailed Description

Author

Miguel Diaz

Version

0.1

3.5 Package microcontroller

Classes

- class Microcontroller
- class Pin

3.5.1 Detailed Description

Microcontroller related classes

Author

Miguel Diaz

Version

0.1

3.6 Package projectConfiguration

Classes

- class ProjectSettings

3.6.1 Detailed Description

Project settings and configuration files

Author

Miguel Diaz

Version

0.1

3.7 Package xmlCreator

Classes

- class ConfXmlWriter

3.7.1 Detailed Description

Create configuration XML

Author

Miguel Diaz

Version

0.1

3.8 Package xmlParser

Classes

- class TestMain
- class XmlOpener

3.8.1 Detailed Description

XML parser for microcontroller information and project settings

Author

Miguel Diaz

Version

0.1

Chapter 4

Class Documentation

4.1 gui.AboutWindow Class Reference

Public Member Functions

- AboutWindow ()

Static Public Member Functions

- static void main (String[ ] args)

4.1.1 Detailed Description

About Window, contains version and contact information

Author

ovd

4.1.2 Constructor & Destructor Documentation

4.1.2.1 AboutWindow()

```
gui.AboutWindow.AboutWindow ( )
```

Create the application.

4.1.3 Member Function Documentation

4.1.3.1 main()

```
static void gui.AboutWindow.main (
            String [] args )  [static]
```

About window main

Parameters

args	Init parameters

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

- src/gui/AboutWindow.java

4.2 framework.CodeGenerator Class Reference

Public Member Functions

- CodeGenerator (Microcontroller uC, ProjectSettings projectSettings)
- ErrorCode Generate ()

4.2.1 Detailed Description

Author

ovd

4.2.2 Constructor & Destructor Documentation

4.2.2.1 CodeGenerator()

```
framework.CodeGenerator.CodeGenerator (
            Microcontroller uC,
            ProjectSettings projectSettings )
```

Constructor

Parameters

uC	Project's microcontroller
projectSettings	Project's settings

4.2.3 Member Function Documentation

4.2.3.1 Generate()

```
ErrorCode framework.CodeGenerator.Generate ( )
```

Generate project's configuration files

Returns

Error code

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

- src/framework/CodeGenerator.java

4.3 configurator.GPIO.CodeName Enum Reference

Public Attributes

- CODE_NAME

Static Public Attributes

- static final String STR_NAME = "codeName"

4.3.1 Detailed Description

Author

Miguel Diaz

Version

0.1

4.3.2 Member Data Documentation

4.3.2.1 CODE_NAME

```
configurator.GPIO.CodeName.CODE_NAME
```

Code name for pin

4.3.2.2 STR_NAME

```
static final String configurator.GPIO.CodeName.STR_NAME = "codeName"  [static]
```

Name as String

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

- src/configurator/GPIO/CodeName.java

4.4 framework.Common Class Reference

Static Public Member Functions

- static String getInstallationFwkPath ()
- static void setInstallationFwkPath (String installationFwkPath)
- static String getProjectFwkPath ()
- static void setProjectFwkPath (String projectFwkPath)
- static String getCfgPath (String fwkPath, String cfgModule)
- static String getCfgFileCPath (String fwkPath, String cfgModule)
- static String getCfgFileHPath (String fwkPath, String cfgModule)
- static String getFrameworkCommonFilePath (String fwkPath)
- static String getFrameworkIncludesFilePath (String fwkPath)
- static String getCommonIncludes (Microcontroller uC)

Static Public Attributes

- static final String STR_MODULE_GPIO = "gpio"
- static final String NL = "\r\n"
- static final String STR_DEFINITION = "#define "
- static final String STR_INCLUDE = "#include "
- static final String STR_HEADER_EXT = ".h"

4.4.1 Detailed Description

Framework common fields and methods

Author

Miguel Diaz

Version

0.1

4.4.2 Member Function Documentation

4.4.2.1 getCfgFileCPath()

```
static String framework.Common.getCfgFileCPath (
             String fwkPath,
             String cfgModule )  [static]
```

Get GPIO configuration file path

Parameters

fwkPath	Framework folder path
cfgModule	Configuration module name

Returns

GPIO configuration file path

4.4.2.2 getCfgFileHPath()

```
static String framework.Common.getCfgFileHPath (
             String fwkPath,
             String cfgModule )  [static]
```

Get GPIO configuration header file path

Parameters

fwkPath	Framework folder path
cfgModule	Configuration module name

Returns

GPIO configuration header file path

4.4.2.3 getCfgPath()

```
static String framework.Common.getCfgPath (
             String fwkPath,
             String cfgModule )  [static]
```

Get configuration module files folder path

Parameters

fwkPath	Framework folder path
cfgModule	Configuration module name

Returns

Configuration files folder path

4.4.2.4 getCommonIncludes()

```
static String framework.Common.getCommonIncludes (
             Microcontroller uC )  [static]
```

Parameters

uC	Microcontroller used

Returns

Common headers needed for framework

4.4.2.5 getFrameworkCommonFilePath()

```
static String framework.Common.getFrameworkCommonFilePath (
             String fwkPath )  [static]
```

Get the framework common header path

Parameters

fwkPath	Framework folder path

Returns

Framework common header path

4.4.2.6 getFrameworkIncludesFilePath()

```
static String framework.Common.getFrameworkIncludesFilePath (
             String fwkPath )  [static]
```

Get the framework includes header path

Parameters

fwkPath	Framework folder path

Returns

Framework includes header path

4.4.2.7 getInstallationFwkPath()

```
static String framework.Common.getInstallationFwkPath ( )  [static]
```

Get installation framework path

Returns

installation framework path

4.4.2.8 getProjectFwkPath()

```
static String framework.Common.getProjectFwkPath ( )  [static]
```

Get project's framework path

Returns

project's framework path

4.4.2.9 setInstallationFwkPath()

```
static void framework.Common.setInstallationFwkPath (
             String installationFwkPath )  [static]
```

Set installation framework path

Parameters

installationFwkPath	installation framework path

4.4.2.10 setProjectFwkPath()

```
static void framework.Common.setProjectFwkPath (
             String projectFwkPath )  [static]
```

Set project's framework path

Parameters

projectFwkPath	project's framework path

4.4.3 Member Data Documentation

4.4.3.1 NL

```
final String framework.Common.NL = "\r\n"  [static]
```

Common implementation of New Line

4.4.3.2 STR_DEFINITION

```
final String framework.Common.STR_DEFINITION = "#define "  [static]
```

Macro definition String

4.4.3.3 STR_HEADER_EXT

```
final String framework.Common.STR_HEADER_EXT = ".h"  [static]
```

Header file extension

4.4.3.4 STR_INCLUDE

```
final String framework.Common.STR_INCLUDE = "#include "  [static]
```

Include header file string

4.4.3.5 STR_MODULE_GPIO

```
final String framework.Common.STR_MODULE_GPIO = "gpio"  [static]
```

GPIO module name

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

- src/framework/Common.java

4.5 configurator.ConfigurationFile Class Reference

Static Public Attributes

- static final String STR_PROJ_CONF_FILE = "cgs"

4.5.1 Detailed Description

Configuration files properties

Author

Miguel Diaz

Version

0.1

4.5.2 Member Data Documentation

4.5.2.1 STR_PROJ_CONF_FILE

```
final String configurator.ConfigurationFile.STR_PROJ_CONF_FILE = "cgs"  [static]
```

Public configuration file extension

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

- src/configurator/ConfigurationFile.java

4.6 xmlCreator.ConfXmlWriter Class Reference

Public Member Functions

- ConfXmlWriter (Microcontroller uC)
- void addPin (PinConf pin, int pinNum)
- ErrorCode writeXml (String fileName)

4.6.1 Detailed Description

Write a XML file

Author

Miguel Diaz

Version

0.1

4.6.2 Constructor & Destructor Documentation

4.6.2.1 ConfXmlWriter()

```
xmlCreator.ConfXmlWriter.ConfXmlWriter (
            Microcontroller uC )
```

Constructor

Parameters

uC	Microcontroller configuration

4.6.3 Member Function Documentation

4.6.3.1 addPin()

```
void xmlCreator.ConfXmlWriter.addPin (
             PinConf pin,
             int pinNum )
```

Add a pin configuration to the file

Parameters

pin	Pin configuration
pinNum	Number of GPIO pin

4.6.3.2 writeXml()

```
ErrorCode xmlCreator.ConfXmlWriter.writeXml (
             String fileName )
```

Write the XMl file

Parameters

fileName	Name of XML configuration file

Returns

Error status

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

- src/xmlCreator/ConfXmlWriter.java

4.7 common.ErrorCode Enum Reference

Public Attributes

- NO_ERROR
- EX_ERROR
- FILE_READ_ERROR
- FILE_WRITE_ERROR
- FILE_CONF_ERROR

Static Public Attributes

- static final String STR_INVALID = "STR_INVALID"

4.7.1 Detailed Description

Error codes enum

Author

Miguel Diaz

Version

0.1

4.7.2 Member Data Documentation

4.7.2.1 EX_ERROR

```
common.ErrorCode.EX_ERROR
```

Error during execution

4.7.2.2 FILE_CONF_ERROR

```
common.ErrorCode.FILE_CONF_ERROR
```

File configuration error

4.7.2.3 FILE_READ_ERROR

```
common.ErrorCode.FILE_READ_ERROR
```

File reading error

4.7.2.4 FILE_WRITE_ERROR

```
common.ErrorCode.FILE_WRITE_ERROR
```

File writing error

4.7.2.5 NO_ERROR

```
common.ErrorCode.NO_ERROR
```

No error message

4.7.2.6 STR_INVALID

```
static final String common.ErrorCode.STR_INVALID = "STR_INVALID"  [static]
```

Error string

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

- src/common/ErrorCode.java

4.8 common.Features Class Reference

Static Public Member Functions

- static void verbosePrint (String verboseMessage)
- static void debugPrint (String debugMessage)

Static Public Attributes

- static final boolean DEBUG = true
- static final boolean VERBOSE = true
- static final String VERBOSE_STR = "# "
- static final String DEBUG_STR = "#$ "
- static final String SW_VERSION = VERSION_MAJOR + "." + VERSION_MINOR + "." + VERSION_PATCH
- static final String VERSION_STATUS = "Alpha"
- static final String VERSION_NAME = "Bespin"

4.8.1 Detailed Description

Class that includes all project features

Author

Miguel Diaz

Version

0.1

4.8.2 Member Function Documentation

4.8.2.1 debugPrint()

```
static void common.Features.debugPrint (
             String debugMessage )  [static]
```

Print Debug message to console

Parameters

debugMessage	Message to display

4.8.2.2 verbosePrint()

```
static void common.Features.verbosePrint (
             String verboseMessage )  [static]
```

Print Verbose message to console

Parameters

verboseMessage	Message to display

4.8.3 Member Data Documentation

4.8.3.1 DEBUG

```
final boolean common.Features.DEBUG = true  [static]
```

Enables debug functions

4.8.3.2 DEBUG_STR

```
final String common.Features.DEBUG_STR = "#$ "  [static]
```

Debug messages indicator on system console

4.8.3.3 SW_VERSION

```
final String common.Features.SW_VERSION = VERSION_MAJOR + "." + VERSION_MINOR + "." + VERSION_PA↩
TCH  [static]
```

Complete Software version

4.8.3.4 VERBOSE

```
final boolean common.Features.VERBOSE = true  [static]
```

Enables console messages

4.8.3.5 VERBOSE_STR

```
final String common.Features.VERBOSE_STR = "# "  [static]
```

Verbose messages indicator on system console

4.8.3.6 VERSION_NAME

```
final String common.Features.VERSION_NAME = "Bespin"  [static]
```

Code name of the software version

4.8.3.7 VERSION_STATUS

```
final String common.Features.VERSION_STATUS = "Alpha"  [static]
```

Status of the software version

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

- src/common/Features.java

4.9 gui.GpioConfWindow Class Reference

Public Member Functions

- GpioConfWindow (Microcontroller uCtrl)

Static Public Member Functions

- static void main (String[ ] args)

4.9.1 Detailed Description

Window for configuring GPIO pins

Author

Miguel Diaz

Version

0.1

4.9.2 Constructor & Destructor Documentation

4.9.2.1 GpioConfWindow()

```
gui.GpioConfWindow.GpioConfWindow (
            Microcontroller uCtrl )
```

Create the GPIO configuration window and show it

Parameters

uCtrl	Microcontroller object containing all pin's information

4.9.3 Member Function Documentation

4.9.3.1 main()

```
static void gui.GpioConfWindow.main (
             String [] args )  [static]
```

Gpio configuration window main

Parameters

args	Init parameters

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

- src/gui/GpioConfWindow.java

4.10 gui.MainGui Class Reference

Static Public Member Functions

- static void main (String[ ] args)
- static ErrorCode loadProjectFile (File inFile)
- static void showErrorDialog (String message)
- static void showAboutWindow ()
- static void showGpioConfWindow ()
- static void setNewUC (Microcontroller uC)
- static void saveUc ()
- static void generateCode ()

Static Public Attributes

- static File ProjectFile
- static String ProjectPath

4.10.1 Detailed Description

Main GUI state machine

Author

Miguel Diaz

Version

0.1

4.10.2 Member Function Documentation

4.10.2.1 generateCode()

```
static void gui.MainGui.generateCode ( )  [static]
```

Generate source code files

4.10.2.2 loadProjectFile()

```
static ErrorCode gui.MainGui.loadProjectFile (
             File inFile )  [static]
```

Load the project settings file

Parameters

inFile	Settings file

Returns

Error status

4.10.2.3 main()

```
static void gui.MainGui.main (
             String [] args )  [static]
```

Parameters

args	TBD

4.10.2.4 saveUc()

```
static void gui.MainGui.saveUc ( )  [static]
```

Save the microcontroller's configuration to disk

4.10.2.5 setNewUC()

```
static void gui.MainGui.setNewUC (
             Microcontroller uC )  [static]
```

Set the project's microcontroller configuration

Parameters

uC	Microcontroller configuration

4.10.2.6 showAboutWindow()

```
static void gui.MainGui.showAboutWindow ( )  [static]
```

Show about information window

4.10.2.7 showErrorDialog()

```
static void gui.MainGui.showErrorDialog (
             String message )  [static]
```

Show an error dialog

Parameters

message	Message to display

4.10.2.8 showGpioConfWindow()

```
static void gui.MainGui.showGpioConfWindow ( )  [static]
```

Show the GPIOs configuration window

4.10.3 Member Data Documentation

4.10.3.1 ProjectFile

```
File gui.MainGui.ProjectFile  [static]
```

Project configuration file

4.10.3.2 ProjectPath

```
String gui.MainGui.ProjectPath  [static]
```

Project's location

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

- src/gui/MainGui.java

4.11 gui.MainWindow Class Reference

Public Member Functions

- MainWindow ()
- void setVisible (boolean status)
- File OpenFileChooser (String initialPath, String title, FileNameExtensionFilter fileFilter)
- ErrorCode setProjectInformation (String projectName, String ucManufacturer, String ucName)

Static Public Member Functions

- static void main (String[ ] args)

Public Attributes

- JFrame FrmCodeGenerator

4.11.1 Detailed Description

Main application window

Author

Miguel Diaz

Version

0.1

4.11.2 Constructor & Destructor Documentation

4.11.2.1 MainWindow()

```
gui.MainWindow.MainWindow ( )
```

Create the application.

4.11.3 Member Function Documentation

4.11.3.1 main()

```
static void gui.MainWindow.main (
             String [] args )  [static]
```

Open main window

Parameters

args	To be determined

4.11.3.2 OpenFileChooser()

```
File gui.MainWindow.OpenFileChooser (
             String initialPath,
             String title,
             FileNameExtensionFilter fileFilter )
```

Open file chooser dialog and get the selected file

Parameters

initialPath	Path to search the file in
title	Dialog title
fileFilter	Extension filter

Returns

Selected file

4.11.3.3 setProjectInformation()

```
ErrorCode gui.MainWindow.setProjectInformation (
             String projectName,
             String ucManufacturer,
             String ucName )
```

Set Project's name in its label

Parameters

projectName	Project's name
ucManufacturer	Microcontroller's manufacturer
ucName	Microcontroller's model

Returns

Error status

4.11.3.4 setVisible()

```
void gui.MainWindow.setVisible (
             boolean status )
```

Set visibility of About window

Parameters

status	true if visible

4.11.4 Member Data Documentation

4.11.4.1 FrmCodeGenerator

```
JFrame gui.MainWindow.FrmCodeGenerator
```

Frame for the main Window

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

- src/gui/MainWindow.java

4.12 gui.Messages Class Reference

Static Public Member Functions

- static String getString (String key)

4.12.1 Detailed Description

Messages window

Author

ovd

4.12.2 Member Function Documentation

4.12.2.1 getString()

```
static String gui.Messages.getString (
             String key )  [static]
```

Get String

Parameters

key	Key

Returns

String

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

- src/gui/Messages.java

4.13 microcontroller.Microcontroller Class Reference

Public Member Functions

- Microcontroller (Document ucDoc)
- ErrorCode processDocument ()
- ErrorCode loadPinsConf (Document confDoc)
- Pin getPin (int pinNum)
- String getUc_model ()
- String getUc_manufacturer ()
- int getUc_pinNum ()
- int getUc_gpioNum ()
- int getUc_portNum ()
- PinConf getConfiguredPin (String gpioName)
- boolean isValid ()

Public Attributes

- String [ ] Ports
- String [ ] Includes_Common
- String [ ] Includes_Gpio
- PinConf [ ] GpioCfgPin

Static Public Attributes

- static final int MAX_NUMBER_OF_PINS_PER_PORT = 16

4.13.1 Detailed Description

Microcontroller related methods

Author

Miguel Diaz

Version

0.1

4.13.2 Constructor & Destructor Documentation

4.13.2.1 Microcontroller()

```
microcontroller.Microcontroller.Microcontroller (
            Document ucDoc )
```

Constructor

Parameters

ucDoc	Document obtained from XML file

4.13.3 Member Function Documentation

4.13.3.1 getConfiguredPin()

```
PinConf microcontroller.Microcontroller.getConfiguredPin (
             String gpioName )
```

Get the configuration of a pin

Parameters

gpioName	Name of the pin

Returns

Pin configuration

4.13.3.2 getPin()

```
Pin microcontroller.Microcontroller.getPin (
             int pinNum )
```

Get a pin's characteristics

Parameters

pinNum	Number of pin

Returns

Pin's characteristics

4.13.3.3 getUc_gpioNum()

```
int microcontroller.Microcontroller.getUc_gpioNum ( )
```

Get the number of GPIOs in the microcontroller

Returns

Number of GPIOs

4.13.3.4 getUc_manufacturer()

```
String microcontroller.Microcontroller.getUc_manufacturer ( )
```

Get the microcontroller's manufacturer

Returns

Microcontroller's manufacturer

4.13.3.5 getUc_model()

```
String microcontroller.Microcontroller.getUc_model ( )
```

Get the microcontroller's model

Returns

Microcontroller's model

4.13.3.6 getUc_pinNum()

```
int microcontroller.Microcontroller.getUc_pinNum ( )
```

Get the microcontroller's pins number

Returns

Number of pins

4.13.3.7 getUc_portNum()

```
int microcontroller.Microcontroller.getUc_portNum ( )
```

Get the number of ports in the microcontroller

Returns

Number of ports

4.13.3.8 isValid()

```
boolean microcontroller.Microcontroller.isValid ( )
```

Check if the microcontroller configuration is valid

Returns

true if valid

4.13.3.9 loadPinsConf()

```
ErrorCode microcontroller.Microcontroller.loadPinsConf (
             Document confDoc )
```

Load pins' configuration

Parameters

confDoc	Document with pins

Returns

Error Code

4.13.3.10 processDocument()

```
ErrorCode microcontroller.Microcontroller.processDocument ( )
```

Process the document obtained from XML file

Returns

Error status

4.13.4 Member Data Documentation

4.13.4.1 GpioCfgPin

```
PinConf [] microcontroller.Microcontroller.GpioCfgPin
```

Configured pins list

4.13.4.2 Includes_Common

```
String [] microcontroller.Microcontroller.Includes_Common
```

List of public includes

4.13.4.3 Includes_Gpio

```
String [] microcontroller.Microcontroller.Includes_Gpio
```

List of Includes for GPIO module

4.13.4.4 MAX_NUMBER_OF_PINS_PER_PORT

```
final int microcontroller.Microcontroller.MAX_NUMBER_OF_PINS_PER_PORT = 16  [static]
```

Maximum number of pins allowed in a single port

4.13.4.5 Ports

```
String [] microcontroller.Microcontroller.Ports
```

Ports name list

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

- src/microcontroller/Microcontroller.java

4.14 configurator.GPIO.Mode Enum Reference

Static Public Member Functions

- static Mode getConfFromString (String conf)

Public Attributes

- MODE_INPUT
- MODE_OUTPUT
- MODE_ALTERNATE_FUNCTION
- MODE_ANALOG
- MODE_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "Mode"

4.14.1 Detailed Description

GPIO modes

Author

Miguel Diaz

Version

0.1

4.14.2 Member Function Documentation

4.14.2.1 getConfFromString()

```
static Mode configurator.GPIO.Mode.getConfFromString (
              String conf )  [static]
```

Get the corresponding mode from its name as String

Parameters

conf	Configuration name

Returns

Mode

4.14.3 Member Data Documentation

4.14.3.1 MODE_ALTERNATE_FUNCTION

```
configurator.GPIO.Mode.MODE_ALTERNATE_FUNCTION
```

Alternate function

4.14.3.2 MODE_ANALOG

```
configurator.GPIO.Mode.MODE_ANALOG
```

Analog

4.14.3.3 MODE_INPUT

```
configurator.GPIO.Mode.MODE_INPUT
```

Input

4.14.3.4 MODE_MAX_VALUE

```
configurator.GPIO.Mode.MODE_MAX_VALUE
```

Maximum value for Mode enum

4.14.3.5 MODE_OUTPUT

```
configurator.GPIO.Mode.MODE_OUTPUT
```

Output

4.14.3.6 STR_NAME

```
static final String configurator.GPIO.Mode.STR_NAME = "Mode"  [static]
```

Name as String

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

- src/configurator/GPIO/Mode.java

4.15 configurator.GPIO.OutLevel Enum Reference

Static Public Member Functions

- static OutLevel getConfFromString (String conf)

Public Attributes

- LOW
- HIGH
- MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "OutLevel"

4.15.1 Detailed Description

Pin's output/input level

Author

Miguel Diaz

Version

0.1

4.15.2 Member Function Documentation

4.15.2.1 getConfFromString()

```
static OutLevel configurator.GPIO.OutLevel.getConfFromString (
             String conf )  [static]
```

Get the corresponding mode from its name as String

Parameters

conf	Configuration name

Returns

level

4.15.3 Member Data Documentation

4.15.3.1 HIGH

```
configurator.GPIO.OutLevel.HIGH
```

High, logical 1, Vcc

4.15.3.2 LOW

```
configurator.GPIO.OutLevel.LOW
```

Low, logical 0, Ground

4.15.3.3 MAX_VALUE

```
configurator.GPIO.OutLevel.MAX_VALUE
```

Maximum value for OutLevel enum

4.15.3.4 STR_NAME

```
static final String configurator.GPIO.OutLevel.STR_NAME = "OutLevel"  [static]
```

Name as String

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

- src/configurator/GPIO/OutLevel.java

4.16 configurator.GPIO.OutType Enum Reference

Static Public Member Functions

- static OutType getConfFromString (String conf)

Public Attributes

- OTYPE_PUSH_PULL
- OTYPE_OPEN_DRAIN
- OTYPE_NOT_AVAILABLE
- OTYPE_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "OutType"

4.16.1 Detailed Description

Pin's output type

Author

Miguel Diaz

Version

0.1

4.16.2 Member Function Documentation

4.16.2.1 getConfFromString()

```
static OutType configurator.GPIO.OutType.getConfFromString (
             String conf )  [static]
```

Get the corresponding output type from its name as String

Parameters

conf	Configuration name

Returns

Output type

4.16.3 Member Data Documentation

4.16.3.1 OTYPE_MAX_VALUE

```
configurator.GPIO.OutType.OTYPE_MAX_VALUE
```

Maximum value for OutType enum

4.16.3.2 OTYPE_NOT_AVAILABLE

```
configurator.GPIO.OutType.OTYPE_NOT_AVAILABLE
```

If the pin is configured as input

4.16.3.3 OTYPE_OPEN_DRAIN

```
configurator.GPIO.OutType.OTYPE_OPEN_DRAIN
```

Open Drain

4.16.3.4 OTYPE_PUSH_PULL

```
configurator.GPIO.OutType.OTYPE_PUSH_PULL
```

Push Pull, totem

4.16.3.5 STR_NAME

```
static final String configurator.GPIO.OutType.STR_NAME = "OutType"  [static]
```

Name as String

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

- src/configurator/GPIO/OutType.java

4.17 microcontroller.Pin Class Reference

Public Member Functions

- Pin ()
- void setFunc_vcc (boolean funcState)
- boolean getFunc_vcc ()
- void setFunc_gnd (boolean funcState)
- boolean getFunc_gnd ()
- void setFunc_gpio (boolean funcState)
- boolean getFunc_gpio ()
- void setFunc_reset (boolean funcState)
- boolean getFunc_reset ()
- void setFunc_misc (boolean funcState)
- boolean getFunc_misc ()
- void setFeat_int (boolean featState)
- boolean getFeat_int ()
- void setFeat_adc (boolean featState)
- boolean getFeat_adc ()
- void setFeat_uart (boolean featState)
- boolean getFeat_uart ()
- void setFeat_i2c (boolean featState)
- boolean getFeat_i2c ()
- void setFeat_spi (boolean featState)
- boolean getFeat_spi ()
- void setFeat_clock (boolean featState)
- boolean getFeat_clock ()
- void setFeat_timer (boolean featState)
- boolean getFeat_timer ()
- void setFeat_reset (boolean featState)
- boolean getFeat_reset ()
- void setInt (String feature)
- String getInt ()
- void setAdc (String feature)
- String getAdc ()
- void setUart (String feature)
- String getUart ()
- void setI2c (String feature)
- String getI2c ()
- void setSpi (String feature)
- String getSpi ()
- void setClock (String feature)
- String getClock ()
- void setReset (String feature)
- String getReset ()
- void setTimer (String feature)
- String getTimer ()
- void setName (String pinName)
- String getName ()
- void setNumber (int pinNum)
- int getNumber ()
- void setPort (String pinPort)
- String getPort ()
- boolean isValid ()

Static Public Attributes

- static final boolean ENABLE = true
- static final boolean DISABLE = false
- static final boolean DEF_FUNCTION = DEF_BOOLEAN
- static final boolean DEF_FEATURE_AV = DEF_BOOLEAN
- static final String DEF_FEATURE = DEF_STRING
- static final String DEF_NAME = DEF_STRING
- static final int DEF_NUMBER = DEF_INT
- static final String DEF_PORT = DEF_STRING

4.17.1 Detailed Description

Basic pin object.

- Pin necessary characteristics:
  - Name
  - Number
- Pin optional characteristics:
  - Port
- Pin main functions:
  - VCC
  - GND
  - GPIO
  - RESET
  - MISC
- Pin features:
  - Interruption
  - ADC
  - UART
  - I2C
  - SPI
  - Clock
  - Reset

Author

Miguel Diaz

Version

0.1

4.17.2 Constructor & Destructor Documentation

4.17.2.1 Pin()

```
microcontroller.Pin.Pin ( )
```

Initialize all pin's characteristics and features to their default values

4.17.3 Member Function Documentation

4.17.3.1 getAdc()

```
String microcontroller.Pin.getAdc ( )
```

Get the pin's ADC name

Returns

Pin's ADC

4.17.3.2 getClock()

```
String microcontroller.Pin.getClock ( )
```

Get the pin's clock name

Returns

Pin's clock

4.17.3.3 getFeat_adc()

```
boolean microcontroller.Pin.getFeat_adc ( )
```

See if the pin has an ADC

Returns

Feature availability

4.17.3.4 getFeat_clock()

```
boolean microcontroller.Pin.getFeat_clock ( )
```

See if the pin supports a clock

Returns

Feature availability

4.17.3.5 getFeat_i2c()

```
boolean microcontroller.Pin.getFeat_i2c ( )
```

See if the pin has I2C

Returns

Feature availability

4.17.3.6 getFeat_int()

```
boolean microcontroller.Pin.getFeat_int ( )
```

See if the pin has an interruption

Returns

Feature availability

4.17.3.7 getFeat_reset()

```
boolean microcontroller.Pin.getFeat_reset ( )
```

See if the pin has a reset feature

Returns

Feature availability

4.17.3.8 getFeat_spi()

```
boolean microcontroller.Pin.getFeat_spi ( )
```

See if the pin has SPI

Returns

Feature availability

4.17.3.9 getFeat_timer()

```
boolean microcontroller.Pin.getFeat_timer ( )
```

See if the pin supports a timer

Returns

Feature availability

4.17.3.10 getFeat_uart()

```
boolean microcontroller.Pin.getFeat_uart ( )
```

See if the pin has a UART

Returns

Feature availability

4.17.3.11 getFunc_gnd()

```
boolean microcontroller.Pin.getFunc_gnd ( )
```

See if the pin is GND

Returns

Function availability

4.17.3.12 getFunc_gpio()

```
boolean microcontroller.Pin.getFunc_gpio ( )
```

See if the pin is GPIO

Returns

Function availability

4.17.3.13 getFunc_misc()

```
boolean microcontroller.Pin.getFunc_misc ( )
```

See if the pin is MISC

Returns

Function availability

4.17.3.14 getFunc_reset()

```
boolean microcontroller.Pin.getFunc_reset ( )
```

See if the pin is RESET

Returns

Function availability

4.17.3.15 getFunc_vcc()

```
boolean microcontroller.Pin.getFunc_vcc ( )
```

See if the pin is Vcc

Returns

Function availability

4.17.3.16 getI2c()

```
String microcontroller.Pin.getI2c ( )
```

Get the pin's I2C name

Returns

Pin's I2C

4.17.3.17 getInt()

```
String microcontroller.Pin.getInt ( )
```

Get the pin's interruption name

Returns

Pin's interruption

4.17.3.18 getName()

```
String microcontroller.Pin.getName ( )
```

Get the pin's name

Returns

Pin's name

4.17.3.19 getNumber()

```
int microcontroller.Pin.getNumber ( )
```

Get the pin's number

Returns

Pin's number

4.17.3.20 getPort()

```
String microcontroller.Pin.getPort ( )
```

Get the pin's port

Returns

Pin's port

4.17.3.21 getReset()

```
String microcontroller.Pin.getReset ( )
```

Get the pin's reset name

Returns

Pin's reset

4.17.3.22 getSpi()

```
String microcontroller.Pin.getSpi ( )
```

Get the pin's SPI name

Returns

Pin's SPI

4.17.3.23 getTimer()

```
String microcontroller.Pin.getTimer ( )
```

Get the pin's timer name

Returns

Pin's timer

4.17.3.24 getUart()

```
String microcontroller.Pin.getUart ( )
```

Get the pin's UART name

Returns

Pin's UART

4.17.3.25 isValid()

```
boolean microcontroller.Pin.isValid ( )
```

Check if the pin is correctly initialized

Returns

True if the pin is correctly initialized

4.17.3.26 setAdc()

```
void microcontroller.Pin.setAdc (
             String feature )
```

Set the pin's ADC

Parameters

feature	Pin's ADC

4.17.3.27 setClock()

```
void microcontroller.Pin.setClock (
             String feature )
```

Set the pin's clock

Parameters

feature	Pin's clock

4.17.3.28 setFeat_adc()

```
void microcontroller.Pin.setFeat_adc (
             boolean featState )
```

Set the pin's ADC feature

Parameters

featState	Feature availability

4.17.3.29 setFeat_clock()

```
void microcontroller.Pin.setFeat_clock (
             boolean featState )
```

Set the pin's Clock feature

Parameters

featState	Feature availability

4.17.3.30 setFeat_i2c()

```
void microcontroller.Pin.setFeat_i2c (
             boolean featState )
```

Set the pin's I2C feature

Parameters

featState	Feature availability

4.17.3.31 setFeat_int()

```
void microcontroller.Pin.setFeat_int (
             boolean featState )
```

Set the pin's interruption feature

Parameters

featState	Feature availability

4.17.3.32 setFeat_reset()

```
void microcontroller.Pin.setFeat_reset (
             boolean featState )
```

Set the pin's reset feature

Parameters

featState	Feature availability

4.17.3.33 setFeat_spi()

```
void microcontroller.Pin.setFeat_spi (
             boolean featState )
```

Set the pin's SPI feature

Parameters

featState	Feature availability

4.17.3.34 setFeat_timer()

```
void microcontroller.Pin.setFeat_timer (
             boolean featState )
```

Set the pin's timer feature

Parameters

featState	Feature availability

4.17.3.35 setFeat_uart()

```
void microcontroller.Pin.setFeat_uart (
             boolean featState )
```

Set the pin's UART feature

Parameters

featState	Feature availability

4.17.3.36 setFunc_gnd()

```
void microcontroller.Pin.setFunc_gnd (
             boolean funcState )
```

Set the pin to GND status

Parameters

funcState	Function availability

4.17.3.37 setFunc_gpio()

```
void microcontroller.Pin.setFunc_gpio (
             boolean funcState )
```

Set the pin to GPIO status

Parameters

funcState	Function availability

4.17.3.38 setFunc_misc()

```
void microcontroller.Pin.setFunc_misc (
             boolean funcState )
```

Set the pin to MISC status

Parameters

funcState	Function availability

4.17.3.39 setFunc_reset()

```
void microcontroller.Pin.setFunc_reset (
             boolean funcState )
```

Set the pin to RESET status

Parameters

funcState	Function availability

4.17.3.40 setFunc_vcc()

```
void microcontroller.Pin.setFunc_vcc (
             boolean funcState )
```

Set the pin to Vcc status

Parameters

funcState	Function availability

4.17.3.41 setI2c()

```
void microcontroller.Pin.setI2c (
             String feature )
```

Set the pin's I2C

Parameters

feature	Pin's I2C

4.17.3.42 setInt()

```
void microcontroller.Pin.setInt (
             String feature )
```

Set the pin's interruption

Parameters

feature	Pin's interruption

4.17.3.43 setName()

```
void microcontroller.Pin.setName (
              String pinName )
```

Set the pin's name

Parameters

pinName	Pin's name

4.17.3.44 setNumber()

```
void microcontroller.Pin.setNumber (
              int pinNum )
```

Set the pin's number

Parameters

pinNum	Pin's number

4.17.3.45 setPort()

```
void microcontroller.Pin.setPort (
              String pinPort )
```

Set the pin's port

Parameters

pinPort	Pin's port

4.17.3.46 setReset()

```
void microcontroller.Pin.setReset (
              String feature )
```

Set the pin's reset

Parameters

feature	Pin's reset

4.17.3.47 setSpi()

```
void microcontroller.Pin.setSpi (
             String feature )
```

Set the pin's SPI

Parameters

feature	Pin's SPI

4.17.3.48 setTimer()

```
void microcontroller.Pin.setTimer (
             String feature )
```

Set the pin's timer

Parameters

feature	Pin's timer

4.17.3.49 setUart()

```
void microcontroller.Pin.setUart (
             String feature )
```

Set the pin's UART

Parameters

feature	Pin's UART

4.17.4 Member Data Documentation

4.17.4.1 DEF_FEATURE

```
final String microcontroller.Pin.DEF_FEATURE = DEF_STRING  [static]
```

Default value for pin's feature as not available

4.17.4.2 DEF_FEATURE_AV

```
final boolean microcontroller.Pin.DEF_FEATURE_AV = DEF_BOOLEAN  [static]
```

Default value for pin's feature availability as not available

4.17.4.3 DEF_FUNCTION

```
final boolean microcontroller.Pin.DEF_FUNCTION = DEF_BOOLEAN  [static]
```

Default value for pin's function as not enabled

4.17.4.4 DEF_NAME

```
final String microcontroller.Pin.DEF_NAME = DEF_STRING  [static]
```

Default value for pin's name

4.17.4.5 DEF_NUMBER

```
final int microcontroller.Pin.DEF_NUMBER = DEF_INT  [static]
```

Default value for pin's number

4.17.4.6 DEF_PORT

```
final String microcontroller.Pin.DEF_PORT = DEF_STRING  [static]
```

Default value for pin's port

4.17.4.7 DISABLE

```
final boolean microcontroller.Pin.DISABLE = false  [static]
```

Disable value for features and functions

4.17.4.8 ENABLE

```
final boolean microcontroller.Pin.ENABLE = true  [static]
```

Enable value for features and functions

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

- src/microcontroller/Pin.java

4.18 configurator.PinConf Class Reference

Public Member Functions

- PinConf (Pin gpioPin)
- boolean isValid ()
- String getPort ()
- String getPinName ()
- String getCodeName ()
- void setCodeName (String name)
- Mode getMode ()
- void setMode (Mode mode)
- OutType getOutType ()
- void setOutType (OutType outType)
- OutLevel getOutLevel ()
- void setOutLevel (OutLevel level)
- Speed getSpeed ()
- void setSpeed (Speed speed)
- Pull getPull ()
- void setPull (Pull pull)
- boolean isAv_Adc ()
- boolean isAv_altFunc ()

Static Public Attributes

- static final Mode DF_MODE = Mode.MODE_INPUT
- static final Speed DF_SPEED = Speed.SPEED_FAST
- static final OutType DF_OUTTYPE = OutType.OTYPE_PUSH_PULL
- static final OutLevel DF_OUT_LEVEL = OutLevel.LOW
- static final Pull DF_PULL = Pull.PULL_NOT_AVAILABLE
- static final String DF_CODE_NAME = ""

4.18.1 Detailed Description

GPIO pin configuration

Author

Miguel Diaz

Version

0.1

4.18.2 Constructor & Destructor Documentation

4.18.2.1 PinConf()

```
configurator.PinConf.PinConf (
              Pin gpioPin )
```

Constructor

Parameters

gpioPin	Pin information

4.18.3 Member Function Documentation

4.18.3.1 getCodeName()

```
String configurator.PinConf.getCodeName ( )
```

Get the pin's user selected name

Returns

pin's name

4.18.3.2 getMode()

```
Mode configurator.PinConf.getMode ( )
```

Get the pin's mode configuration

Returns

Mode

4.18.3.3 getOutLevel()

```
OutLevel configurator.PinConf.getOutLevel ( )
```

Get the pin's output level

Returns

Pin's output level

4.18.3.4 getOutType()

```
OutType configurator.PinConf.getOutType ( )
```

Get the pin's output configuration

Returns

Output configuration

4.18.3.5 getPinName()

```
String configurator.PinConf.getPinName ( )
```

Get the pin's number

Returns

Pin's number

4.18.3.6 getPort()

```
String configurator.PinConf.getPort ( )
```

Get the pin's port

Returns

Port

4.18.3.7 getPull()

```
Pull configurator.PinConf.getPull ( )
```

Get the pin's pull resistor configuration

Returns

Pull Resistor configuration

4.18.3.8 getSpeed()

```
Speed configurator.PinConf.getSpeed ( )
```

Get the pin's speed

Returns

Speed

4.18.3.9 isAv_Adc()

```
boolean configurator.PinConf.isAv_Adc ( )
```

Check availability of ADC

Returns

True if ADC is available

4.18.3.10 isAv_altFunc()

```
boolean configurator.PinConf.isAv_altFunc ( )
```

Check the availability of alternate function

Returns

True if alternate function is available

4.18.3.11 isValid()

```
boolean configurator.PinConf.isValid ( )
```

Check if the GPIO pin is valid

Returns

True if valid

4.18.3.12 setCodeName()

```
void configurator.PinConf.setCodeName (
            String name )
```

Set the pin's user selected name

Parameters

name	Pin's name

4.18.3.13 setMode()

```
void configurator.PinConf.setMode (
            Mode mode )
```

Set the pin's mode configuration

Parameters

mode	Mode

4.18.3.14 setOutLevel()

```
void configurator.PinConf.setOutLevel (
             OutLevel level )
```

Set the pin's output level

Parameters

level	Pin's output level

4.18.3.15 setOutType()

```
void configurator.PinConf.setOutType (
             OutType outType )
```

Set the pin's output configuration

Parameters

outType	Output configuration

4.18.3.16 setPull()

```
void configurator.PinConf.setPull (
             Pull pull )
```

Set the pull resistor configuration

Parameters

pull	Resistor configuration

4.18.3.17 setSpeed()

```
void configurator.PinConf.setSpeed (
            Speed speed )
```

Set the pin's speed

Parameters

speed	Speed

4.18.4 Member Data Documentation

4.18.4.1 DF_CODE_NAME

```
final String configurator.PinConf.DF_CODE_NAME = ""  [static]
```

Default pin's code name

4.18.4.2 DF_MODE

```
final Mode configurator.PinConf.DF_MODE = Mode.MODE_INPUT  [static]
```

Default Pin mode

4.18.4.3 DF_OUT_LEVEL

```
final OutLevel configurator.PinConf.DF_OUT_LEVEL = OutLevel.LOW  [static]
```

Default pin's output level

4.18.4.4 DF_OUTTYPE

```
final OutType configurator.PinConf.DF_OUTTYPE = OutType.OTYPE_PUSH_PULL  [static]
```

Default pin's output type

4.18.4.5 DF_PULL

```
final Pull configurator.PinConf.DF_PULL = Pull.PULL_NOT_AVAILABLE  [static]
```

Default pin's pull resistor

4.18.4.6 DF_SPEED

```
final Speed configurator.PinConf.DF_SPEED = Speed.SPEED_FAST  [static]
```

Default pin's speed

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

- src/configurator/PinConf.java

4.19 projectConfiguration.ProjectSettings Class Reference

Public Member Functions

- ProjectSettings ()
- ErrorCode processDocument ()
- ErrorCode openProjectFile (File inFile)
- File getConfFile ()
- File getUcFile ()
- String getProjectName ()
- String getFrameworkPath ()

4.19.1 Detailed Description

Project settings class

Author

Miguel Diaz

Version

0.2

4.19.2 Constructor & Destructor Documentation

4.19.2.1 ProjectSettings()

```
projectConfiguration.ProjectSettings.ProjectSettings ( )
```

Constructor

4.19.3 Member Function Documentation

4.19.3.1 getConfFile()

```
File projectConfiguration.ProjectSettings.getConfFile ( )
```

Get the project configuration file

Returns

Project configuration file

4.19.3.2 getFrameworkPath()

```
String projectConfiguration.ProjectSettings.getFrameworkPath ( )
```

Get the framework folder

Returns

framework folder

4.19.3.3 getProjectName()

```
String projectConfiguration.ProjectSettings.getProjectName ( )
```

Get the project's name

Returns

Project's name

4.19.3.4 getUcFile()

```
File projectConfiguration.ProjectSettings.getUcFile ( )
```

Get the project microcontroller file

Returns

Project microcontroller file

4.19.3.5 openProjectFile()

```
ErrorCode projectConfiguration.ProjectSettings.openProjectFile (
            File inFile )
```

Open the project settings file

Parameters

inFile	Project file

Returns

Error Status

4.19.3.6 processDocument()

```
ErrorCode projectConfiguration.ProjectSettings.processDocument ( )
```

Process the document obtained from the XML file

Returns

Error Status

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

- src/projectConfiguration/ProjectSettings.java

4.20 configurator.GPIO.Pull Enum Reference

Static Public Member Functions

- static Pull getConfFromString (String conf)

Public Attributes

- PULL_UP
- PULL_DOWN
- PULL_NOT_AVAILABLE
- PULL_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "Pull"

4.20.1 Detailed Description

Pin's pull resistor

Author

Miguel Diaz

Version

0.1

4.20.2 Member Function Documentation

4.20.2.1 getConfFromString()

```
static Pull configurator.GPIO.Pull.getConfFromString (
              String conf )  [static]
```

Get the corresponding Pull configuration from its name as String

Parameters

conf	Configuration name

Returns

Pull configuration

4.20.3 Member Data Documentation

4.20.3.1 PULL_DOWN

```
configurator.GPIO.Pull.PULL_DOWN
```

Pull Down

4.20.3.2 PULL_MAX_VALUE

```
configurator.GPIO.Pull.PULL_MAX_VALUE
```

Maximum value for Pull enum

4.20.3.3 PULL_NOT_AVAILABLE

```
configurator.GPIO.Pull.PULL_NOT_AVAILABLE
```

If the pin is configured as output, or there is no resistor available

4.20.3.4 PULL_UP

```
configurator.GPIO.Pull.PULL_UP
```

Pull Up

4.20.3.5 STR_NAME

```
static final String configurator.GPIO.Pull.STR_NAME = "Pull"  [static]
```

Name as String

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

- src/configurator/GPIO/Pull.java

4.21 configurator.GPIO.Speed Enum Reference

Static Public Member Functions

- static Speed getConfFromString (String conf)

Public Attributes

- SPEED_FAST
- SPEED_MEDIUM
- SPEED_HIGH
- SPEED_NOT_AVAILABLE
- SPEED_MAX_VALUE

Static Public Attributes

- static final String STR_NAME = "Speed"

4.21.1 Detailed Description

Pin's speed

Author

Miguel Diaz

Version

0.1

4.21.2 Member Function Documentation

4.21.2.1 getConfFromString()

```
static Speed configurator.GPIO.Speed.getConfFromString (
             String conf )  [static]
```

Get the corresponding Speed configuration from its name as String

Parameters

conf	Configuration name

Returns

Speed

4.21.3 Member Data Documentation

4.21.3.1 SPEED_FAST

```
configurator.GPIO.Speed.SPEED_FAST
```

Fast

4.21.3.2 SPEED_HIGH

```
configurator.GPIO.Speed.SPEED_HIGH
```

High

4.21.3.3 SPEED_MAX_VALUE

```
configurator.GPIO.Speed.SPEED_MAX_VALUE
```

Maximum value for Speed enum

4.21.3.4 SPEED_MEDIUM

```
configurator.GPIO.Speed.SPEED_MEDIUM
```

Medium

4.21.3.5 SPEED_NOT_AVAILABLE

```
configurator.GPIO.Speed.SPEED_NOT_AVAILABLE
```

Not all MCUs will have this setting

4.21.3.6 STR_NAME

```
static final String configurator.GPIO.Speed.STR_NAME = "Speed"  [static]
```

Name as String

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

- src/configurator/GPIO/Speed.java

4.22 xmlParser.TestMain Class Reference

Static Public Member Functions

- static void main (String[ ] openOption)

4.22.1 Detailed Description

Dummy main class for testing the other classes

Author

Miguel Diaz

4.22.2 Member Function Documentation

4.22.2.1 main()

```
static void xmlParser.TestMain.main (
             String [] openOption )  [static]
```

Main without GUI

Parameters

openOption	Options include:

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

- src/xmlParser/TestMain.java

4.23 xmlParser.XmlOpener Class Reference

Public Member Functions

- XmlOpener ()
- ErrorCode OpenFile (File inFile)
- Document getParsedDoc ()

Static Public Member Functions

- static String getElementInfoFromDoc (Document doc, String elementName)
- static String getElementInfo (Element element, String elementName)

4.23.1 Detailed Description

Open and process XML files

Author

H112943

Version

0.1

4.23.2 Constructor & Destructor Documentation

4.23.2.1 XmlOpener()

```
xmlParser.XmlOpener.XmlOpener ( )
```

Constructor

4.23.3 Member Function Documentation

4.23.3.1 getElementInfo()

```
static String xmlParser.XmlOpener.getElementInfo (
             Element element,
             String elementName )  [static]
```

Get an XML sub element information

Parameters

element	XML main element
elementName	Sub element's name

Returns

Sub elemen't information

4.23.3.2 getElementInfoFromDoc()

```
static String xmlParser.XmlOpener.getElementInfoFromDoc (
             Document doc,
             String elementName )  [static]
```

Get an XML element information

Parameters

doc	Document from XML file
elementName	Element's name

Returns

Element's information

4.23.3.3 getParsedDoc()

```
Document xmlParser.XmlOpener.getParsedDoc ( )
```

Get the parsed document AFTER opening the file

Returns

Parsed document

4.23.3.4 OpenFile()

```
ErrorCode xmlParser.XmlOpener.OpenFile (
             File inFile )
```

Open the XML file

Parameters

inFile	XML file

Returns

Error code

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

- src/xmlParser/XmlOpener.java

Index