

Lib act data

1.0

Generated by Doxygen 1.8.2

Sun Dec 30 2012 19:19:14

## Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>1</b>
2.1	Class List . . . . .	1
<b>3</b>	<b>File Index</b>	<b>2</b>
3.1	File List . . . . .	2
<b>4</b>	<b>Class Documentation</b>	<b>2</b>
4.1	CalcData Class Reference . . . . .	2
4.1.1	Detailed Description . . . . .	3
4.1.2	Constructor & Destructor Documentation . . . . .	3
4.1.3	Member Function Documentation . . . . .	3
4.2	Context Class Reference . . . . .	6
4.2.1	Detailed Description . . . . .	7
4.2.2	Constructor & Destructor Documentation . . . . .	7
4.2.3	Member Function Documentation . . . . .	7
4.3	LoadSave Class Reference . . . . .	11
4.3.1	Detailed Description . . . . .	12
4.3.2	Member Function Documentation . . . . .	12
4.4	LoadSaveFactory Class Reference . . . . .	14
4.4.1	Detailed Description . . . . .	14
4.4.2	Member Function Documentation . . . . .	14
4.5	LoadSaveXML Class Reference . . . . .	15
4.5.1	Detailed Description . . . . .	16
4.5.2	Member Function Documentation . . . . .	16
4.6	ReadWrite Class Reference . . . . .	18
4.6.1	Detailed Description . . . . .	18
4.6.2	Member Function Documentation . . . . .	19
4.7	ReadWriteFactory Class Reference . . . . .	21
4.7.1	Detailed Description . . . . .	22
4.7.2	Member Function Documentation . . . . .	22
4.8	ReadWriteModbus Class Reference . . . . .	22
4.8.1	Detailed Description . . . . .	24
4.8.2	Member Function Documentation . . . . .	24
4.9	ReadWriteXML Class Reference . . . . .	29
4.9.1	Detailed Description . . . . .	30
4.9.2	Member Function Documentation . . . . .	30

<b>5 File Documentation</b>	<b>33</b>
5.1 CalcData.h File Reference . . . . .	34
5.2 Context.h File Reference . . . . .	34
5.3 LoadSave.h File Reference . . . . .	35
5.4 LoadSaveFactory.h File Reference . . . . .	36
5.5 LoadSaveXML.h File Reference . . . . .	37
5.6 ReadWrite.h File Reference . . . . .	38
5.7 ReadWriteModbus.h File Reference . . . . .	39
5.7.1 Macro Definition Documentation . . . . .	40
5.8 ReadWriteXML.h File Reference . . . . .	41

<b>Index</b>	<b>42</b>
--------------	-----------

## 1 Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<b>CalcData</b>	<b>2</b>
<b>Context</b>	<b>6</b>
<b>LoadSave</b>	<b>11</b>
<b>LoadSaveXML</b>	<b>15</b>
<b>LoadSaveFactory</b>	<b>14</b>
<b>ReadWrite</b>	<b>18</b>
<b>ReadWriteModbus</b>	<b>22</b>
<b>ReadWriteXML</b>	<b>29</b>
<b>ReadWriteFactory</b>	<b>21</b>

## 2 Class Index

### 2.1 Class List

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

<b>CalcData</b>	<b>2</b>
<b>Context</b>	<b>6</b>
<b>LoadSave</b>	
Virtual class with API for Loas and Save XML files for configuration and data	<b>11</b>
<b>LoadSaveFactory</b>	<b>14</b>

<a href="#">LoadSaveXML</a>	15
<a href="#">ReadWrite</a>	18
<a href="#">ReadWriteFactory</a>	21
<a href="#">ReadWriteModbus</a>	22
<a href="#">ReadWriteXML</a>	29

## 3 File Index

### 3.1 File List

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

<b>CalcData.cpp</b>	??
<a href="#">CalcData.h</a>	34
<b>Context.cpp</b>	??
<a href="#">Context.h</a>	34
<a href="#">LoadSave.h</a>	35
<b>LoadSaveFactory.cpp</b>	??
<a href="#">LoadSaveFactory.h</a>	36
<b>LoadSaveXML.cpp</b>	??
<a href="#">LoadSaveXML.h</a>	37
<a href="#">ReadWrite.h</a>	38
<b>ReadWriteFactory.cpp</b>	??
<b>ReadWriteFactory.h</b>	??
<b>ReadWriteModbus.cpp</b>	??
<a href="#">ReadWriteModbus.h</a>	39
<b>ReadWriteXML.cpp</b>	??
<a href="#">ReadWriteXML.h</a>	41

## 4 Class Documentation

### 4.1 CalcData Class Reference

#### Public Member Functions

- [CalcData](#) (u16 mTN, u16 mTP)  
*Constructor.*
- float [u](#) (u16 u)

- Convert u values.*

  - float **i** (u16 i)

*Convert i values.*

  - float **angle** (s16 ang)

*Convert angle values.*

  - float **p** (float p)

*Convert power values.*

  - float **s** (u16 **u**, u16 **i**)

*Calculate apparent power values.*

  - float **d** (u16 **u**, u16 **i**, float **p**, float q)

*Calculate deformed power values.*

  - string **iO** (u16 num)

*Decode input and output signalization.*

  - string **underOverfl** (u8 num)

*Decode underflow and overflow signalization.*

  - float **h** (float h)

*Convert values.*

#### Private Attributes

- float **mTP**
- float **mTN**

#### 4.1.1 Detailed Description

Definition at line 9 of file CalcData.h.

#### 4.1.2 Constructor & Destructor Documentation

##### 4.1.2.1 CalcData::CalcData ( u16 mTN, u16 mTP )

Constructor.

Sets private variables *mTN* and *mTP*.

#### Parameters

<i>mTP</i>	unconverted mTP value.
<i>mTN</i>	unconverted mTN value.

Definition at line 12 of file CalcData.cpp.

#### 4.1.3 Member Function Documentation

##### 4.1.3.1 float CalcData::angle ( s16 ang )

Convert angle values.

#### Parameters

<i>ang</i>	unconverted angle value.
------------	--------------------------

**Returns**

converted value.

Definition at line 38 of file CalcData.cpp.

**4.1.3.2 float CalcData::d ( u16 *u*, u16 *i*, float *p*, float *q* )**

Calculate deformed power values.

**Parameters**

<i>u</i>	unconverted u value.
<i>i</i>	unconverted i value.
<i>p</i>	active power.
<i>q</i>	reactive power.

**Returns**

calculated apparent power.

Definition at line 50 of file CalcData.cpp.

**4.1.3.3 float CalcData::h ( float *h* )**

Convert values.

**Parameters**

<i>unconverted</i>	value.
--------------------	--------

**Returns**

value/100.

Definition at line 42 of file CalcData.cpp.

**4.1.3.4 float CalcData::i ( u16 *i* )**

Convert i values.

**Parameters**

<i>i</i>	unconverted i value.
----------	----------------------

**Returns**

converted value.

Definition at line 30 of file CalcData.cpp.

**4.1.3.5 string CalcData::iO ( u16 *num* )**

Decode input and output signalization.

**Parameters**

<i>encoded</i>	IO
----------------	----

**Returns**

decoded string.

Definition at line 119 of file CalcData.cpp.

**4.1.3.6 float CalcData::p ( float *p* )**

Convert power values.

**Parameters**

<i>power</i>	unconverted power value.
--------------	--------------------------

**Returns**

converted value.

Definition at line 34 of file CalcData.cpp.

**4.1.3.7 float CalcData::s ( u16 *u*, u16 *i* )**

Calculate apparent power values.

**Parameters**

<i>u</i>	unconverted u value.
<i>i</i>	unconverted i value.

**Returns**

calculated apparent power.

Definition at line 46 of file CalcData.cpp.

**4.1.3.8 float CalcData::u ( u16 *u* )**

Convert u values.

**Parameters**

<i>u</i>	unconverted u value.
----------	----------------------

**Returns**

converted value.

Definition at line 26 of file CalcData.cpp.

**4.1.3.9 string CalcData::underOverfl ( u8 *num* )**

Decode underflow and overflow signalization.

**Parameters**

<i>encoded</i>	value
----------------	-------

## Returns

decoded string.

Definition at line 54 of file CalcData.cpp.

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

- [CalcData.h](#)
- [CalcData.cpp](#)

## 4.2 Context Class Reference

## Public Member Functions

- [Context](#) ()  
*Constructor.*
- int [getConnType](#) ()  
*Getter for variable connType.*
- void [setConnType](#) (int [connType](#))  
*Setter for variable connType.*
- bool [getXmlConnected](#) ()  
*Getter for variable xmlConnected.*
- void [setXmlConnected](#) (bool [xmlConnected](#))  
*Setter for variable xmlConnected.*
- bool [getModbusConnected](#) ()  
*Getter for variable modbusConnected.*
- void [setModbusConnected](#) (bool [modbusConnected](#))  
*Setter for variable modbusConnected.*
- int [getModbusPort](#) ()  
*Getter for variable modbusPort.*
- void [setModbusPort](#) (int [modbusPort](#))  
*Setter for variable modbusPort.*
- char \* [getModbusIP](#) ()  
*Getter for variable modbusIP.*
- void [setModbusIP](#) (char \*[modbusIP](#))  
*Setter for variable modbusIP.*
- modbus\_t \* [getModbusConnection](#) ()  
*Getter for identifier of modbus connection.*
- void [setModbusConnection](#) (modbus\_t \*[modbusConn](#))  
*Setter for identifier of modbus connection.*
- char \* [getSmpConfigFileLoc](#) ()  
*Getter for variable SmpConfigFileLoc.*
- char \* [getSmpInstallConfigFileLoc](#) ()  
*Getter for variable SmpInstallConfigFileLoc.*
- char \* [getSmpElectricityMeterConfigFileLoc](#) ()  
*Getter for variable SmpElectricityMeterConfigFileLoc.*
- char \* [getCSMPDataFileLoc](#) ()  
*Getter for variable CSMPDataFileLoc.*
- void [setSmpConfigFileLoc](#) (char \*file)  
*Setter for variable SmpConfigFileLoc.*
- void [setSmpInstallConfigFileLoc](#) (char \*file)  
*Setter for variable SmpInstallConfigFileLoc.*



- void [setSmpElectricityMeterConfigFileLoc](#) (char \*file)  
*Setter for variable SmpElectricityMeterConfigFileLoc.*
- void [setCSMPDataFileLoc](#) (char \*file)  
*Setter for variable SmpCSMPDataFileLoc.*

#### Private Attributes

- int [connType](#)  
*Connection type.*
- bool [xmlConnected](#)  
*State of XML connection.*
- bool [modbusConnected](#)  
*State of modbus connection.*
- int [modbusPort](#)  
*Modbus port.*
- char \* [modbusIP](#)  
*Modbus IP.*
- modbus\_t \* [modbusConn](#)  
*Identificator of modbus connection.*
- char \* [smpConfigFileLoc](#)  
*Path to file with SMP config.*
- char \* [smpInstallConfigFileLoc](#)  
*Path to file with SMP install config.*
- char \* [smpElectricityMeterConfigFileLoc](#)  
*Path to file with SMP electricity meter config.*
- char \* [cSMPDataFileLoc](#)  
*Path to file with CSMP data.*

#### 4.2.1 Detailed Description

Definition at line 7 of file Context.h.

#### 4.2.2 Constructor & Destructor Documentation

##### 4.2.2.1 Context::Context ( ) [inline]

Constructor.

Sets private variables to default value.

Definition at line 13 of file Context.h.

#### 4.2.3 Member Function Documentation

##### 4.2.3.1 int Context::getConnType ( )

Getter for variable connType.

Gets variable containing type of connection.

#### Returns

private variable variable connType.

Definition at line 12 of file Context.cpp.

#### 4.2.3.2 `char * Context::getCSMPDataFileLoc ( )`

Getter for variable CSMPDataFileLoc.

Gets variabe containing path to file with CSMPData.

##### Returns

private variable CSMPDataFileLoc.

Definition at line 72 of file Context.cpp.

#### 4.2.3.3 `bool Context::getModbusConnected ( )`

Getter for variable modbusConnected.

Gets variabe containing state of modbus connection.

##### Returns

private variable modbusConnected.

Definition at line 20 of file Context.cpp.

#### 4.2.3.4 `modbus_t * Context::getModbusConnection ( )`

Getter for identificator of modbus connection.

##### Returns

identificator for modbus connection.

Definition at line 52 of file Context.cpp.

#### 4.2.3.5 `char * Context::getModbusIP ( )`

Getter for variable modbusIP.

Gets variabe containing modbus IP.

##### Returns

private variable modbusIP.

Definition at line 44 of file Context.cpp.

#### 4.2.3.6 `int Context::getModbusPort ( )`

Getter for variable modbusPort.

Gets variabe containing modbus port.

##### Returns

private variable modbusPort.

Definition at line 36 of file Context.cpp.

#### 4.2.3.7 `char * Context::getSmpConfigFileLoc ( )`

Getter for variable SmpConfigFileLoc.

Gets variabe containing path to file with SMPConfig.

**Returns**

private variable SmpConfigFileLoc.

Definition at line 75 of file Context.cpp.

**4.2.3.8 char \* Context::getSmpElectricityMeterConfigFileLoc ( )**

Getter for variable SmpElectricityMeterConfigFileLoc.

Gets varialbe containing path to file with SMPElectricityMeterConfig.

**Returns**

private variable SmpElectricityMeterConfigFileLoc.

Definition at line 81 of file Context.cpp.

**4.2.3.9 char \* Context::getSmpInstallConfigFileLoc ( )**

Getter for variable SmpInstallConfigFileLoc.

Gets varialbe containing path to file with SMPInstallConfig.

**Returns**

private variable SmpInstallConfigFileLoc.

Definition at line 78 of file Context.cpp.

**4.2.3.10 bool Context::getXmlConnected ( )**

Getter for variable xmlConnected.

Gets varialbe containing state of xml connection.

**Returns**

private variable xmlConnected.

Definition at line 28 of file Context.cpp.

**4.2.3.11 void Context::setConnType ( int *connType* )**

Setter for variable connType.

Sets varialbe containing type of connection.

**Parameters**

<i>connType</i>	variable with type of connection.
-----------------	-----------------------------------

Definition at line 16 of file Context.cpp.

**4.2.3.12 void Context::setCSMPDataFileLoc ( char \* *file* )**

Setter for variable SmpCSMPDataFileLoc.

Sets varialbe containing path to file with CSMPData.

**Parameters**

<i>private</i>	variable CSMPDataFileLoc.
----------------	---------------------------

Definition at line 60 of file Context.cpp.

#### 4.2.3.13 void Context::setModbusConnected ( bool *modbusConnected* )

Setter for variable modbusConnected.

Sets variable containing state of modbus connection.

##### Parameters

<i>modbus-Connected</i>	modbus connection.
-------------------------	--------------------

Definition at line 24 of file Context.cpp.

#### 4.2.3.14 void Context::setModbusConnection ( modbus\_t \* *modbusConn* )

Setter for identifier of modbus connection.

##### Parameters

<i>modbusConn</i>	identifier of modbus connection.
-------------------	----------------------------------

Definition at line 56 of file Context.cpp.

#### 4.2.3.15 void Context::setModbusIP ( char \* *modbusIP* )

Setter for variable modbusIP.

Sets variable containing modbus IP.

##### Parameters

<i>modbusIP</i>	IP address for modbus connection.
-----------------	-----------------------------------

Definition at line 48 of file Context.cpp.

#### 4.2.3.16 void Context::setModbusPort ( int *modbusPort* )

Setter for variable modbusPort.

Sets variable containing modbus port.

##### Returns

private variable modbusPort.

Definition at line 40 of file Context.cpp.

#### 4.2.3.17 void Context::setSmpConfigFileLoc ( char \* *file* )

Setter for variable SmpConfigFileLoc.

Sets variable containing path to file with SMPConfig.

##### Parameters

<i>private</i>	variable SmpConfigFileLoc.
----------------	----------------------------

Definition at line 63 of file Context.cpp.

**4.2.3.18 void Context::setSmpElectricityMeterConfigFileLoc ( char \* *file* )**

Setter for variable SmpElectricityMeterConfigFileLoc.

Sets variabe containing path to file with SMPElectricityMeterConfig.

**Parameters**

<i>private</i>	variable SmpElectricityMeterConfigFileLoc.
----------------	--

Definition at line 69 of file Context.cpp.

**4.2.3.19 void Context::setSmpInstallConfigFileLoc ( char \* *file* )**

Setter for variable SmpInstallConfigFileLoc.

Sets variabe containing path to file with SMPInstallConfig.

**Parameters**

<i>private</i>	variable SmpInstallConfigFileLoc.
----------------	-----------------------------------

Definition at line 66 of file Context.cpp.

**4.2.3.20 void Context::setXmlConnected ( bool *xmlConnected* )**

Setter for variable xmlConnected.

Sets variabe containing state of connection.

**Parameters**

<i>xmlConnected</i>	variable containing xmlConnected.
---------------------	-----------------------------------

Definition at line 32 of file Context.cpp.

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

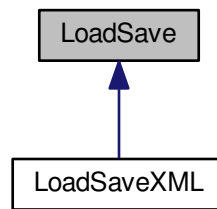
- [Context.h](#)
- Context.cpp

**4.3 LoadSave Class Reference**

Virtual class with API for Loas and Save XML files for configuration and data.

```
#include <LoadSave.h>
```

Inheritance diagram for LoadSave:



#### Public Member Functions

- virtual int [load](#) (SmpConfig \*smpConfig, const char \*file)=0  
*Loads XML file with SMP configuration.*
- virtual int [save](#) (SmpConfig \*smpConfig, const char \*file)=0  
*Saves XML file with SMP configuration.*
- virtual int [load](#) (SmpInstallConfig \*smpInstallConfig, const char \*file)=0  
*Loads XML file with SMP install configuration.*
- virtual int [save](#) (SmpInstallConfig \*smpInstallConfig, const char \*file)=0  
*Saves XML file with SMP install configuration.*
- virtual int [load](#) (SmpElectricityMeterConfig \*data, const char \*file)=0  
*Loads XML file with SMP electricity meter configuration.*
- virtual int [save](#) (SmpElectricityMeterConfig \*data, const char \*file)=0  
*Saves XML file with SMP electricity meter configuration.*

#### 4.3.1 Detailed Description

Virtual class with API for Load and Save XML files for configuration and data.

API consists of two overloaded functions for load and save XML files with configuration or data.

Definition at line 10 of file LoadSave.h.

#### 4.3.2 Member Function Documentation

**4.3.2.1** virtual int LoadSave::load ( SmpConfig \* *smpConfig*, const char \* *file* ) [pure virtual]

Loads XML file with SMP configuration.

##### Parameters

<i>smpConfig</i>	pointer to smpConfig structure
<i>path</i>	to file

##### Returns

0 if everything worked fine or error code

Implemented in [LoadSaveXML](#).

4.3.2.2 `virtual int LoadSave::load ( SmpInstallConfig * smpInstallConfig, const char * file )` [pure virtual]

Loads XML file with SMP install configuration.

#### Parameters

<i>smpInstallConfig</i>	pointer to smpInstallConfig structure
<i>path</i>	to file

#### Returns

0 if everything worked fine or error code

Implemented in [LoadSaveXML](#).

4.3.2.3 `virtual int LoadSave::load ( SmpElectricityMeterConfig * data, const char * file )` [pure virtual]

Loads XML file with SMP electricity meter configuration.

#### Parameters

<i>smpElectricity-MeterConfig</i>	pointer to smpElectricityMeterConfig structure
<i>path</i>	to file

#### Returns

0 if everything worked fine or error code

Implemented in [LoadSaveXML](#).

4.3.2.4 `virtual int LoadSave::save ( SmpConfig * smpConfig, const char * file )` [pure virtual]

Saves XML file with SMP configuration.

#### Parameters

<i>smpConfig</i>	pointer to smpConfig structure
<i>path</i>	to file

#### Returns

0 if everything worked fine or error code

Implemented in [LoadSaveXML](#).

4.3.2.5 `virtual int LoadSave::save ( SmpInstallConfig * smpInstallConfig, const char * file )` [pure virtual]

Saves XML file with SMP install configuration.

#### Parameters

<i>smpInstallConfig</i>	pointer to smpInstallConfig structure
<i>path</i>	to file

#### Returns

0 if everything worked fine or error code

Implemented in [LoadSaveXML](#).

4.3.2.6 `virtual int LoadSave::save ( SmpElectricityMeterConfig * data, const char * file )` [pure virtual]

Saves XML file with SMP electricity meter configuration.

#### Parameters

<i>smpElectricity-MeterConfig</i>	pointer to smpElectricityMeterConfig structure
<i>path</i>	to file

#### Returns

0 if everything worked fine or error code

Implemented in [LoadSaveXML](#).

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

- [LoadSave.h](#)

## 4.4 LoadSaveFactory Class Reference

### Public Member Functions

- [LoadSave](#) \* [chooseLSType](#) (Context \*cont)  
*Choose Load Save Type.*

#### 4.4.1 Detailed Description

Definition at line 8 of file LoadSaveFactory.h.

#### 4.4.2 Member Function Documentation

##### 4.4.2.1 [LoadSave](#) \* LoadSaveFactory::chooseLSType ( Context \* *cont* )

Choose Load Save Type.

\*According [Context](#) choose appropriate type of Load Save and returns.

#### Parameters

<i>cont</i>	class with data describing connection
-------------	---------------------------------------

Definition at line 10 of file LoadSaveFactory.cpp.

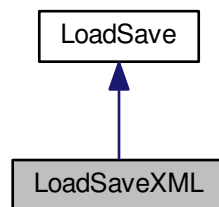
The documentation for this class was generated from the following files:

- [LoadSaveFactory.h](#)
- [LoadSaveFactory.cpp](#)

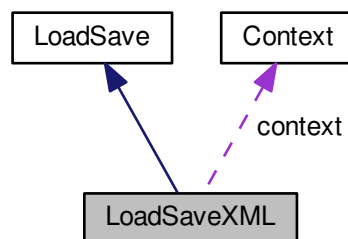


## 4.5 LoadSaveXML Class Reference

Inheritance diagram for LoadSaveXML:



Collaboration diagram for LoadSaveXML:



### Public Member Functions

- `int load (SmpConfig *smpConfig, const char *file)`  
*Loads XML file with SMP configuration.*
- `int save (SmpConfig *smpConfig, const char *file)`  
*Saves XML file with SMP configuration.*
- `int load (SmpInstallConfig *smpInstallConfig, const char *file)`  
*Loads XML file with SMP install configuration.*
- `int save (SmpInstallConfig *smpInstallConfig, const char *file)`  
*Saves XML file with SMP install configuration.*
- `int load (SmpElectricityMeterConfig *data, const char *file)`  
*Loads XML file with SMP electricity meter configuration.*
- `int save (SmpElectricityMeterConfig *data, const char *file)`  
*Saves XML file with SMP electricity meter configuration.*

### Private Attributes

- `Context * context`

## 4.5.1 Detailed Description

Definition at line 8 of file LoadSaveXML.h.

## 4.5.2 Member Function Documentation

4.5.2.1 `int LoadSaveXML::load ( SmpConfig * smpConfig, const char * file )` `[virtual]`

Loads XML file with SMP configuration.

## Parameters

<i>smpConfig</i>	pointer to smpConfig structure
<i>path</i>	to file

## Returns

0 if everything worked fine or error code

Implements [LoadSave](#).

Definition at line 25 of file LoadSaveXML.cpp.

4.5.2.2 `int LoadSaveXML::load ( SmpInstallConfig * smpInstallConfig, const char * file )` `[virtual]`

Loads XML file with SMP install configuration.

## Parameters

<i>smpInstallConfig</i>	pointer to smpInstallConfig structure
<i>path</i>	to file

## Returns

0 if everything worked fine or error code

Implements [LoadSave](#).

Definition at line 46 of file LoadSaveXML.cpp.

4.5.2.3 `int LoadSaveXML::load ( SmpElectricityMeterConfig * data, const char * file )` `[virtual]`

Loads XML file with SMP electricity meter configuration.

## Parameters

<i>smpElectricity-MeterConfig</i>	pointer to smpElectricityMeterConfig structure
<i>path</i>	to file

## Returns

0 if everything worked fine or error code

Implements [LoadSave](#).

Definition at line 10 of file LoadSaveXML.cpp.

4.5.2.4 `int LoadSaveXML::save ( SmpConfig * smpConfig, const char * file )` `[virtual]`

Saves XML file with SMP configuration.

## Parameters

<i>smpConfig</i>	pointer to smpConfig structure
<i>path</i>	to file

## Returns

0 if everything worked fine or error code

Implements [LoadSave](#).

Definition at line 32 of file LoadSaveXML.cpp.

4.5.2.5 `int LoadSaveXML::save ( SmpInstallConfig * smpInstallConfig, const char * file )` [virtual]

Saves XML file with SMP install configuration.

## Parameters

<i>smpInstallConfig</i>	pointer to smpInstallConfig structure
<i>path</i>	to file

## Returns

0 if everything worked fine or error code

Implements [LoadSave](#).

Definition at line 39 of file LoadSaveXML.cpp.

4.5.2.6 `int LoadSaveXML::save ( SmpElectricityMeterConfig * data, const char * file )` [virtual]

Saves XML file with SMP electricity meter configuration.

## Parameters

<i>smpElectricity-MeterConfig</i>	pointer to smpElectricityMeterConfig structure
<i>path</i>	to file

## Returns

0 if everything worked fine or error code

Implements [LoadSave](#).

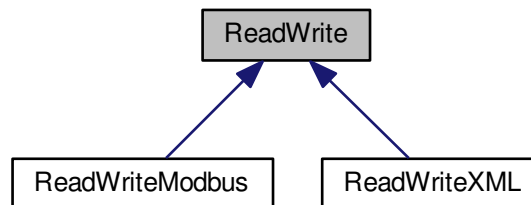
Definition at line 18 of file LoadSaveXML.cpp.

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

- [LoadSaveXML.h](#)
- [LoadSaveXML.cpp](#)

## 4.6 ReadWrite Class Reference

Inheritance diagram for ReadWrite:



### Public Member Functions

- virtual void `setContext` (`Context` \*context)=0  
*Set Context.*
- virtual bool `isConnected` ()=0  
*Is Connected.*
- virtual int `connect` ()=0  
*Connect.*
- virtual int `disconnect` ()=0  
*Disconnect.*
- virtual int `read` (`SmpConfig` \*data)=0  
*Read SMP Config.*
- virtual int `write` (`SmpConfig` \*data)=0  
*Write SMP Config.*
- virtual int `read` (`SmpInstallConfig` \*data)=0  
*Read SMP Install Config.*
- virtual int `write` (`SmpInstallConfig` \*data)=0  
*Write SMP Install Config.*
- virtual int `read` (`CSMPDataCalc` \*pCSMPDataCalc)=0  
*Read CSMP Data.*
- virtual int `read` (`ElmerData` \*pData)=0  
*Read Electricity Meter Data.*
- virtual int `read` (`SmpElectricityMeterConfig` \*data)=0  
*Read SMP Electricity Meter Config.*
- virtual int `write` (`SmpElectricityMeterConfig` \*data)=0  
*Write SMP Electiricty Meter Config.*

### 4.6.1 Detailed Description

Definition at line 8 of file `ReadWrite.h`.

## 4.6.2 Member Function Documentation

### 4.6.2.1 `virtual int ReadWrite::connect ( ) [pure virtual]`

Connect.

\*Connects to desired connection according to [Context](#)

#### Returns

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

### 4.6.2.2 `virtual int ReadWrite::disconnect ( ) [pure virtual]`

Disconnect.

Disconnects from current connection

#### Returns

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

### 4.6.2.3 `virtual bool ReadWrite::isConnected ( ) [pure virtual]`

Is Connected.

#### Returns

connection state

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

### 4.6.2.4 `virtual int ReadWrite::read ( SmpConfig * data ) [pure virtual]`

Read SMP Config.

\*Reads SMP Config from opened connection.

#### Parameters

<i>data</i>	pointer to allocated SmpConfig structure
-------------	--

#### Returns

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

### 4.6.2.5 `virtual int ReadWrite::read ( SmpInstallConfig * data ) [pure virtual]`

Read SMP Install Config.

\*Reads SMP Install Config from opened connection.

#### Parameters

<i>data</i>	pointer to allocated SmpInstallConfig structure
-------------	---

**Returns**

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

**4.6.2.6** `virtual int ReadWrite::read ( CSMPDataCalc * pCSMPDataCalc ) [pure virtual]`

Read CSMP Data.

\*Reads CSMP Data from opened connection.

**Parameters**

<i>data</i>	pointer to allocated CSMPData structure
-------------	---

**Returns**

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

**4.6.2.7** `virtual int ReadWrite::read ( ElmerData * pData ) [pure virtual]`

Read Electricity Meter Data.

\*Reads ElmerData from opened connection.

**Parameters**

<i>data</i>	pointer to allocated ElmerData structure
-------------	--

**Returns**

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

**4.6.2.8** `virtual int ReadWrite::read ( SmpElectricityMeterConfig * data ) [pure virtual]`

Read SMP Electricity Meter Config.

\*Reads SmpElectricityMeterConfig from opened connection.

**Parameters**

<i>data</i>	pointer to allocated SmpElectricityMeterConfig structure
-------------	--

**Returns**

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

**4.6.2.9** `virtual void ReadWrite::setContext ( Context * context ) [pure virtual]`

Set [Context](#).

**Parameters**

<i>context</i>	class with data describing connection
----------------	---------------------------------------

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

4.6.2.10 `virtual int ReadWrite::write ( SmpConfig * data ) [pure virtual]`

Write SMP Config.

\*Writes SMP Config to opened connection.

#### Parameters

<i>data</i>	pointer SmpConfig structure with data
-------------	---------------------------------------

#### Returns

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

4.6.2.11 `virtual int ReadWrite::write ( SmpInstallConfig * data ) [pure virtual]`

Write SMP Install Config.

\*Writes SMP Install Config to opened connection.

#### Parameters

<i>data</i>	pointer SmpInstallConfig structure with data
-------------	--

#### Returns

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

4.6.2.12 `virtual int ReadWrite::write ( SmpElectricityMeterConfig * data ) [pure virtual]`

Write SMP Electricity Meter Config.

\*Writes SMP Electricity Meter Config to opened connection.

#### Parameters

<i>data</i>	pointer SmpElectricityMeterConfig structure with data
-------------	---

#### Returns

0 if everything worked fine or error code

Implemented in [ReadWriteModbus](#), and [ReadWriteXML](#).

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

- [ReadWrite.h](#)

## 4.7 ReadWriteFactory Class Reference

### Public Member Functions

- [ReadWrite \\* chooseRWType \(Context \\*cont\)](#)  
*Choose Read Write Type.*

#### 4.7.1 Detailed Description

Definition at line 9 of file ReadWriteFactory.h.

#### 4.7.2 Member Function Documentation

##### 4.7.2.1 ReadWrite \* ReadWriteFactory::chooseRWType ( Context \* *cont* )

Choose Read Write Type.

\*According *Context* choose appropriate type of Read Write and returns it.

##### Parameters

<i>cont</i>	class with data describing connection
-------------	---------------------------------------

##### Returns

Desired connection

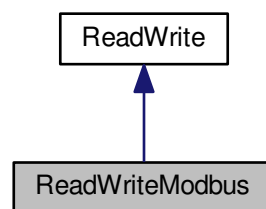
Definition at line 9 of file ReadWriteFactory.cpp.

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

- ReadWriteFactory.h
- ReadWriteFactory.cpp

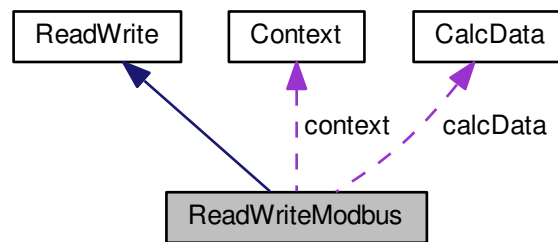
## 4.8 ReadWriteModbus Class Reference

Inheritance diagram for ReadWriteModbus:





Collaboration diagram for ReadWriteModbus:



#### Public Member Functions

- bool **isConnected** ()  
*Is Connected.*
- void **setContext** (Context \*context)  
*Set Context.*
- int **connect** ()  
*Connect.*
- int **disconnect** ()  
*Disconnect.*
- int **read** (CSMPDataCalc \*pCSMPDataCalc)  
*Read CSMP Data.*
- int **read** (ElmerData \*pData)  
*Read Electricity Meter Data.*
- int **read** (SmpIdent \*smpIdent)
- int **read** (SmpConfig \*smpConfig)  
*Read SMP Config.*
- int **write** (SmpConfig \*smpConfig)  
*Write SMP Config.*
- int **read** (SmpInstallConfig \*smpInstallConfig)  
*Read SMP Install Config.*
- int **write** (SmpInstallConfig \*smpInstallConfig)  
*Write SMP Install Config.*
- int **read** (SmpElectricityMeterConfig \*data)  
*Read SMP Electricity Meter Config.*
- int **write** (SmpElectricityMeterConfig \*data)  
*Write SMP Electiricty Meter Config.*

#### Private Member Functions

- float **toFloat** (u16 first, u16 second)  
*To float.*
- u64 **toU64** (u16 first, u16 second, u16 third, u16 fourth)  
*To Unsigned 64.*
- u16 \* **fromFloat** (float num)

- From float.
- `tm * initDate` (int timezone, int summertime)  
Initialize data.
- `int convertDate` (struct tm \*timeinfo, int TimeZone, int SummerTime, u64 time)  
Convert Date.

#### Private Attributes

- `CalcData * calcData`  
Class `CalcData`.
- `Context * context`  
Class `Context` with data describing connection.

#### 4.8.1 Detailed Description

Definition at line 24 of file ReadWriteModbus.h.

#### 4.8.2 Member Function Documentation

##### 4.8.2.1 `int ReadWriteModbus::connect ( )` [virtual]

Connect.

\*Connects to desired connection according to `Context`

#### Returns

0 if everything worked fine or error code

Implements `ReadWrite`.

Definition at line 46 of file ReadWriteModbus.cpp.

##### 4.8.2.2 `int ReadWriteModbus::convertDate ( struct tm * timeinfo, int TimeZone, int SummerTime, u64 time )` [private]

Convert Date.

\*Convert date from unsigned 64b integer in millisecond from 1.1.2000 to standard tm date

#### Parameters

<i>tm</i>	pointer to tm
<i>TimeZone</i>	integer with timezone
<i>SummerTime</i>	integer with SummerTime - 0 if normal time, 1 if summertime
<i>time</i>	time in milliseconds from 1.1.2000

#### Returns

0 if everything worked fine or error code

Definition at line 28 of file ReadWriteModbus.cpp.

##### 4.8.2.3 `int ReadWriteModbus::disconnect ( )` [virtual]

Disconnect.

Disconnects from current connection

**Returns**

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 105 of file ReadWriteModbus.cpp.

**4.8.2.4** `u16 * ReadWriteModbus::fromFloat ( float num ) [private]`

From float.

\*Splits float to two unsigned 16 bites integers

**Parameters**

<i>num</i>	float which should be split
------------	-----------------------------

**Returns**

pointer to array with two splitted u16 integers

Definition at line 135 of file ReadWriteModbus.cpp.

**4.8.2.5** `tm * ReadWriteModbus::initDate ( int timezone, int summertime ) [private]`

Initialize data.

\*Initializes data with default values

**Parameters**

<i>timezone</i>	desired timezone
<i>summertime</i>	0 if normal time and 1 if summer time

**Returns**

pointer tm

Definition at line 11 of file ReadWriteModbus.cpp.

**4.8.2.6** `bool ReadWriteModbus::isConnected ( ) [virtual]`

Is Connected.

**Returns**

connection state

Implements [ReadWrite](#).

Definition at line 38 of file ReadWriteModbus.cpp.

**4.8.2.7** `int ReadWriteModbus::read ( CSMPDataCalc * pCSMPDataCalc ) [virtual]`

Read CSMP Data.

\*Reads CSMP Data from opened connection.

**Parameters**

<i>data</i>	pointer to allocated CSMPData structure
-------------	---

**Returns**

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 467 of file ReadWriteModbus.cpp.

**4.8.2.8** `int ReadWriteModbus::read ( ElmerData * pData ) [virtual]`

Read Electricity Meter Data.

\*Reads ElmerData from opened connection.

**Parameters**

<i>data</i>	pointer to allocated ElmerData structure
-------------	--

**Returns**

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 324 of file ReadWriteModbus.cpp.

**4.8.2.9** `int ReadWriteModbus::read ( SmpConfig * data ) [virtual]`

Read SMP Config.

\*Reads SMP Config from opened connection.

**Parameters**

<i>data</i>	pointer to allocated SmpConfig structure
-------------	--

**Returns**

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 316 of file ReadWriteModbus.cpp.

**4.8.2.10** `int ReadWriteModbus::read ( SmpInstallConfig * data ) [virtual]`

Read SMP Install Config.

\*Reads SMP Install Config from opened connection.

**Parameters**

<i>data</i>	pointer to allocated SmpInstallConfig structure
-------------	---

**Returns**

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 192 of file ReadWriteModbus.cpp.

**4.8.2.11** `int ReadWriteModbus::read ( SmpElectricityMeterConfig * data )` `[virtual]`

Read SMP Electricity Meter Config.

\*Reads SmpElectricityMeterConfig from opened connection.

**Parameters**

<i>data</i>	pointer to allocated SmpElectricityMeterConfig structure
-------------	--

**Returns**

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 640 of file ReadWriteModbus.cpp.

**4.8.2.12** `void ReadWriteModbus::setContext ( Context * context )` `[virtual]`

Set [Context](#).

**Parameters**

<i>context</i>	class with data describing connection
----------------	---------------------------------------

Implements [ReadWrite](#).

Definition at line 42 of file ReadWriteModbus.cpp.

**4.8.2.13** `float ReadWriteModbus::toFloat ( u16 first, u16 second )` `[private]`

To float.

\*Merge two u16 variables to one float

**Parameters**

<i>first</i>	variable to merge
<i>second</i>	variable to merge

**Returns**

merged float

Definition at line 117 of file ReadWriteModbus.cpp.

**4.8.2.14** `u64 ReadWriteModbus::toU64 ( u16 first, u16 second, u16 third, u16 fourth )` `[private]`

To Unsigned 64.

\*Merge four u16 variables to one u64, used with u64 containing time

## Parameters

<i>first</i>	variable to merge
<i>second</i>	variable to merge
<i>third</i>	variable to merge
<i>fourth</i>	variable to merge

## Returns

merged u64

Definition at line 124 of file ReadWriteModbus.cpp.

**4.8.2.15** `int ReadWriteModbus::write ( SmpConfig * data ) [virtual]`

Write SMP Config.

\*Writes SMP Config to opened connection.

## Parameters

<i>data</i>	pointer SmpConfig structure with data
-------------	---------------------------------------

## Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 320 of file ReadWriteModbus.cpp.

**4.8.2.16** `int ReadWriteModbus::write ( SmpInstallConfig * data ) [virtual]`

Write SMP Install Config.

\*Writes SMP Install Config to opened connection.

## Parameters

<i>data</i>	pointer SmpInstallConfig structure with data
-------------	--

## Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 246 of file ReadWriteModbus.cpp.

**4.8.2.17** `int ReadWriteModbus::write ( SmpElectricityMeterConfig * data ) [virtual]`

Write SMP Electricity Meter Config.

\*Writes SMP Electricity Meter Config to opened connection.

## Parameters

<i>data</i>	pointer SmpElectricityMeterConfig structure with data
-------------	---

**Returns**

0 if everything worked fine or error code

Implements [ReadWrite](#).

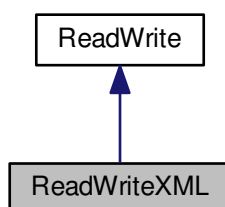
Definition at line 644 of file ReadWriteModbus.cpp.

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

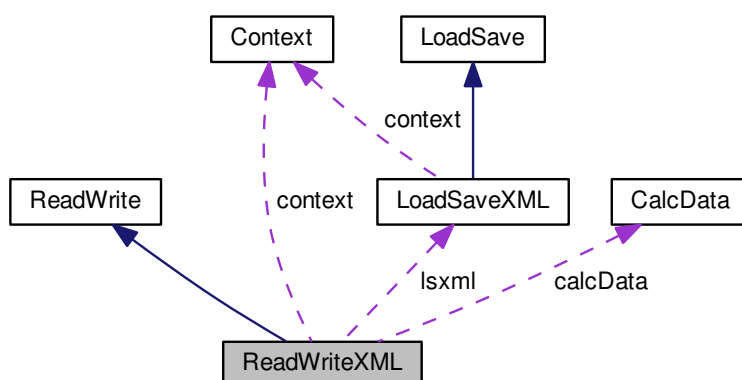
- [ReadWriteModbus.h](#)
- ReadWriteModbus.cpp

## 4.9 ReadWriteXML Class Reference

Inheritance diagram for ReadWriteXML:



Collaboration diagram for ReadWriteXML:

**Public Member Functions**

- bool [isConnected](#) ()  
*Is Connected.*

- void [setContext](#) ([Context](#) \*context)  
*Set Context.*
- int [connect](#) ()  
*Connect.*
- int [disconnect](#) ()  
*Disconnect.*
- int [load](#) (CSMPDataCalc \*pCSMPDataCalc, char \*file)
- int [read](#) (CSMPDataCalc \*pCSMPDataCalc)  
*Read CSMP Data.*
- int [read](#) (SmpConfig \*smpConfig)  
*Read SMP Config.*
- int [write](#) (SmpConfig \*smpConfig)  
*Write SMP Config.*
- int [read](#) (SmpInstallConfig \*smpInstallConfig)  
*Read SMP Install Config.*
- int [write](#) (SmpInstallConfig \*smpInstallConfig)  
*Write SMP Install Config.*
- int [read](#) (SmpElectricityMeterConfig \*data)  
*Read SMP Electricity Meter Config.*
- int [write](#) (SmpElectricityMeterConfig \*data)  
*Write SMP Electiricty Meter Config.*
- int [read](#) (ElmerData \*pData)  
*Read Electricity Meter Data.*

#### Private Attributes

- [CalcData](#) \* [calcData](#)  
*Class [CalcData](#).*
- [Context](#) \* [context](#)  
*Class [Context](#) with data describing connection.*
- [LoadSaveXML](#) \* [lsxml](#)  
*Class [LoadSaveXML](#).*

#### 4.9.1 Detailed Description

Definition at line 11 of file ReadWriteXML.h.

#### 4.9.2 Member Function Documentation

##### 4.9.2.1 int ReadWriteXML::connect ( ) [virtual]

Connect.

\*Connects to desired connection according to [Context](#)

#### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 22 of file ReadWriteXML.cpp.



#### 4.9.2.2 int ReadWriteXML::disconnect ( ) [virtual]

Disconnect.

Disconnects from current connection

##### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 27 of file ReadWriteXML.cpp.

#### 4.9.2.3 bool ReadWriteXML::isConnected ( ) [virtual]

Is Connected.

##### Returns

connection state

Implements [ReadWrite](#).

Definition at line 14 of file ReadWriteXML.cpp.

#### 4.9.2.4 int ReadWriteXML::read ( CSMPDataCalc \* pCSMPDataCalc ) [virtual]

Read CSMP Data.

\*Reads CSMP Data from opened connection.

##### Parameters

<i>data</i>	pointer to allocated CSMPData structure
-------------	---

##### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 161 of file ReadWriteXML.cpp.

#### 4.9.2.5 int ReadWriteXML::read ( SmpConfig \* data ) [virtual]

Read SMP Config.

\*Reads SMP Config from opened connection.

##### Parameters

<i>data</i>	pointer to allocated SmpConfig structure
-------------	--

##### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 173 of file ReadWriteXML.cpp.

#### 4.9.2.6 int ReadWriteXML::read ( SmpInstallConfig \* data ) [virtual]

Read SMP Install Config.

\*Reads SMP Install Config from opened connection.

#### Parameters

<i>data</i>	pointer to allocated SmpInstallConfig structure
-------------	---

#### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 165 of file ReadWriteXML.cpp.

**4.9.2.7** `int ReadWriteXML::read ( SmpElectricityMeterConfig * data ) [virtual]`

Read SMP Electricity Meter Config.

\*Reads SmpElectricityMeterConfig from opened connection.

#### Parameters

<i>data</i>	pointer to allocated SmpElectricityMeterConfig structure
-------------	--

#### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 181 of file ReadWriteXML.cpp.

**4.9.2.8** `int ReadWriteXML::read ( ElmerData * pData ) [virtual]`

Read Electricity Meter Data.

\*Reads ElmerData from opened connection.

#### Parameters

<i>data</i>	pointer to allocated ElmerData structure
-------------	--

#### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 188 of file ReadWriteXML.cpp.

**4.9.2.9** `void ReadWriteXML::setContext ( Context * context ) [virtual]`

Set [Context](#).

#### Parameters

<i>context</i>	class with data describing connection
----------------	---------------------------------------

Implements [ReadWrite](#).

Definition at line 18 of file ReadWriteXML.cpp.

#### 4.9.2.10 `int ReadWriteXML::write ( SmpConfig * data ) [virtual]`

Write SMP Config.

\*Writes SMP Config to opened connection.

##### Parameters

<i>data</i>	pointer SmpConfig structure with data
-------------	---------------------------------------

##### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 177 of file ReadWriteXML.cpp.

#### 4.9.2.11 `int ReadWriteXML::write ( SmpInstallConfig * data ) [virtual]`

Write SMP Install Config.

\*Writes SMP Install Config to opened connection.

##### Parameters

<i>data</i>	pointer SmpInstallConfig structure with data
-------------	--

##### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 169 of file ReadWriteXML.cpp.

#### 4.9.2.12 `int ReadWriteXML::write ( SmpElectricityMeterConfig * data ) [virtual]`

Write SMP Electiricty Meter Config.

\*Writes SMP Electiricty Meter Config to opened connection.

##### Parameters

<i>data</i>	pointer SmpElectricityMeterConfig structure with data
-------------	---

##### Returns

0 if everything worked fine or error code

Implements [ReadWrite](#).

Definition at line 185 of file ReadWriteXML.cpp.

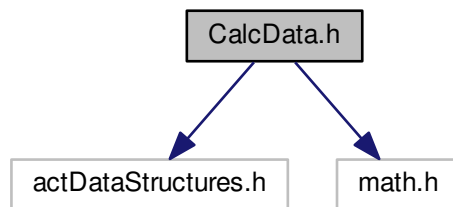
The documentation for this class was generated from the following files:

- [ReadWriteXML.h](#)
- ReadWriteXML.cpp

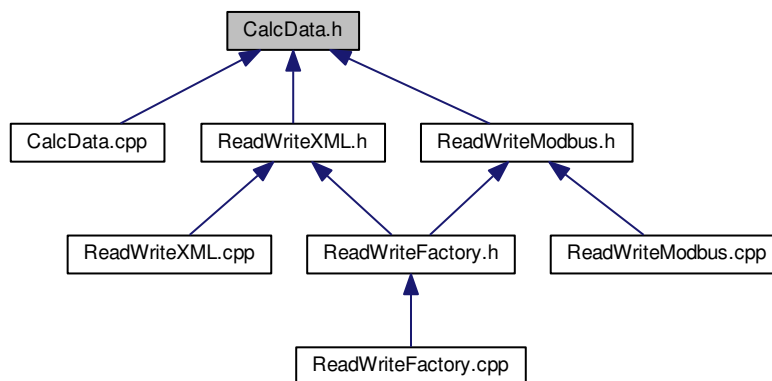
## 5 File Documentation

## 5.1 CalcData.h File Reference

```
#include "actDataStructures.h"  
#include "math.h"  
Include dependency graph for CalcData.h:
```



This graph shows which files directly or indirectly include this file:



### Classes

- class [CalcData](#)

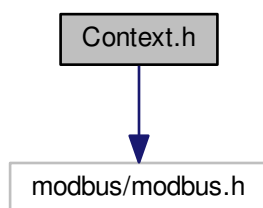
### Macros

- `#define PI 3.14159265358979323846264338327950288419716939937510`

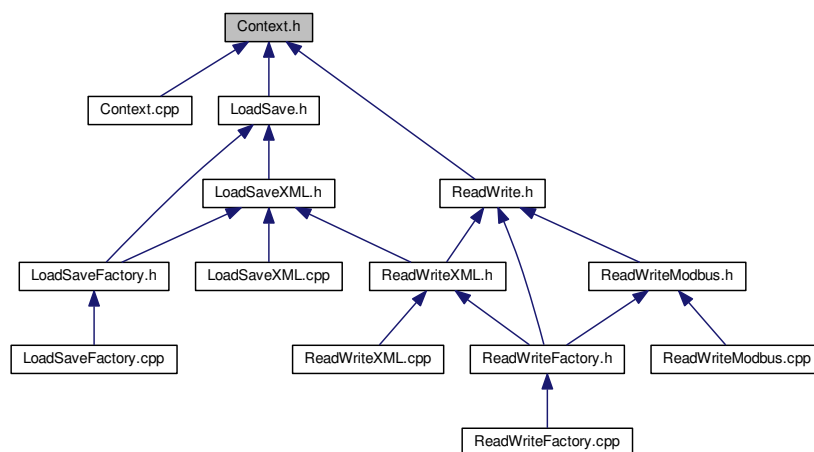
## 5.2 Context.h File Reference

```
#include "modbus/modbus.h"
```

Include dependency graph for Context.h:



This graph shows which files directly or indirectly include this file:



#### Classes

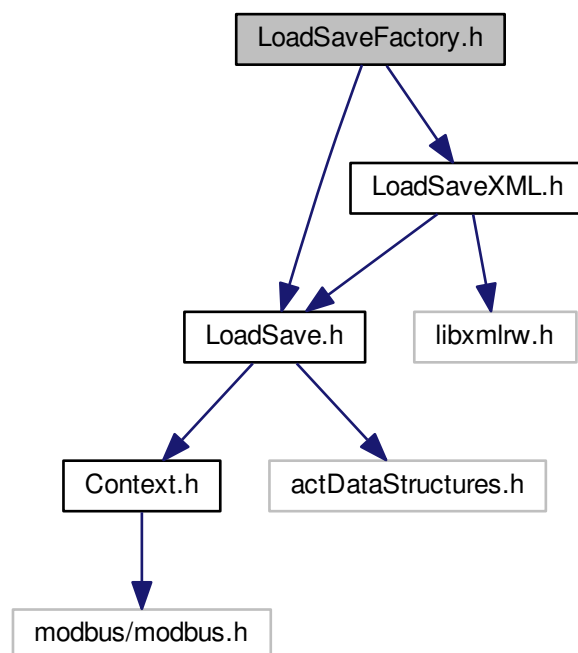
- class [Context](#)

### 5.3 LoadSave.h File Reference

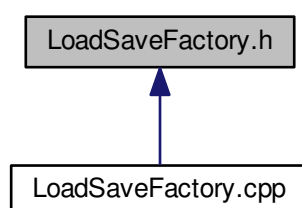
```
#include "Context.h"  
#include "actDataStructures.h"
```



Include dependency graph for LoadSaveFactory.h:



This graph shows which files directly or indirectly include this file:



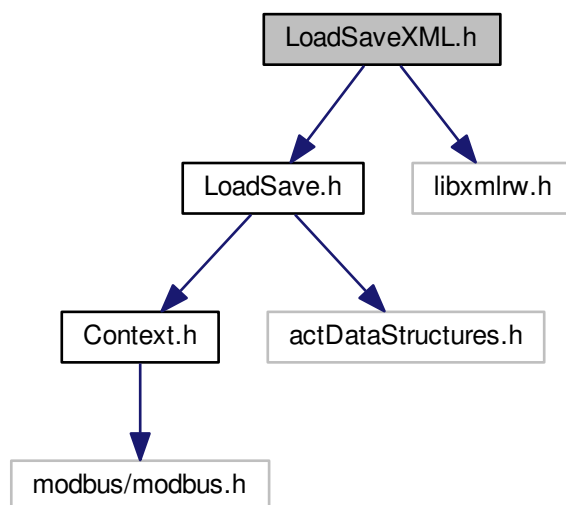
#### Classes

- class [LoadSaveFactory](#)

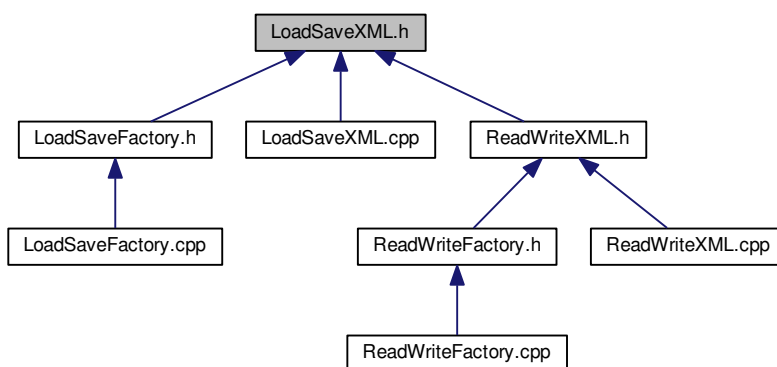
## 5.5 LoadSaveXML.h File Reference

```
#include "LoadSave.h"  
#include "libxmlrw.h"
```

Include dependency graph for LoadSaveXML.h:



This graph shows which files directly or indirectly include this file:



#### Classes

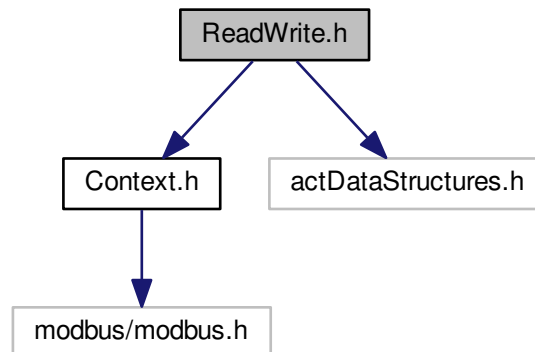
- class [LoadSaveXML](#)

## 5.6 ReadWrite.h File Reference

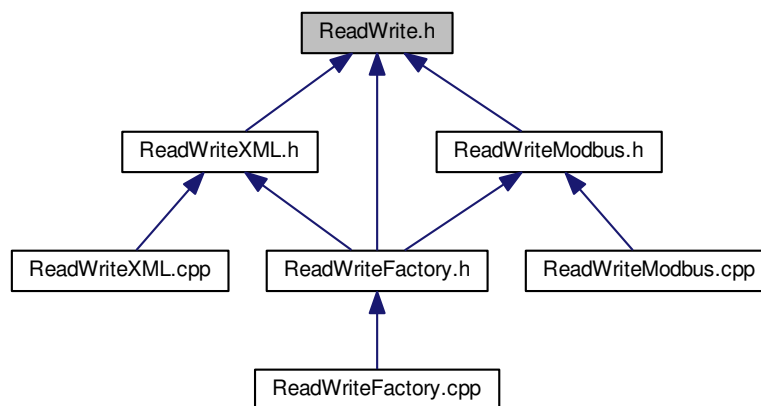
```
#include "Context.h"
#include "actDataStructures.h"
```



Include dependency graph for ReadWrite.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [ReadWrite](#)

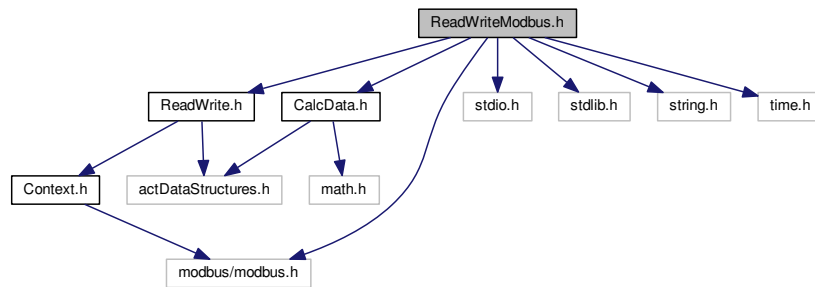
## 5.7 ReadWriteModbus.h File Reference

```

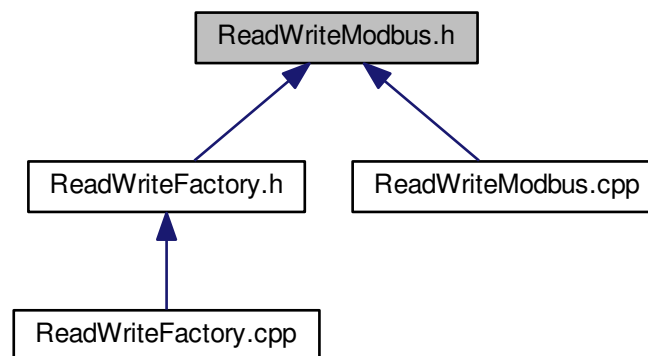
#include "ReadWrite.h"
#include "CalcData.h"
#include "modbus/modbus.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

```

Include dependency graph for ReadWriteModbus.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ReadWriteModbus](#)

## Macros

- `#define bswap64(x)`
- `#define SERVER_ID 17`

### 5.7.1 Macro Definition Documentation

#### 5.7.1.1 `#define bswap64( x )`

**Value:**

```

( (x << 56) & 0xff00000000000000UL ) | \
  ( (x << 40) & 0x00ff000000000000UL ) | \
  ( (x << 24) & 0x0000ff0000000000UL ) | \
  ( (x << 8) & 0x000000ff00000000UL ) | \

```

```

( (x >> 8) & 0x00000000ff000000UL ) | \
( (x >> 24) & 0x0000000000ff0000UL ) | \
( (x >> 40) & 0x000000000000ff00UL ) | \
( (x >> 56) & 0x000000000000ffUL )

```

Definition at line 13 of file ReadWriteModbus.h.

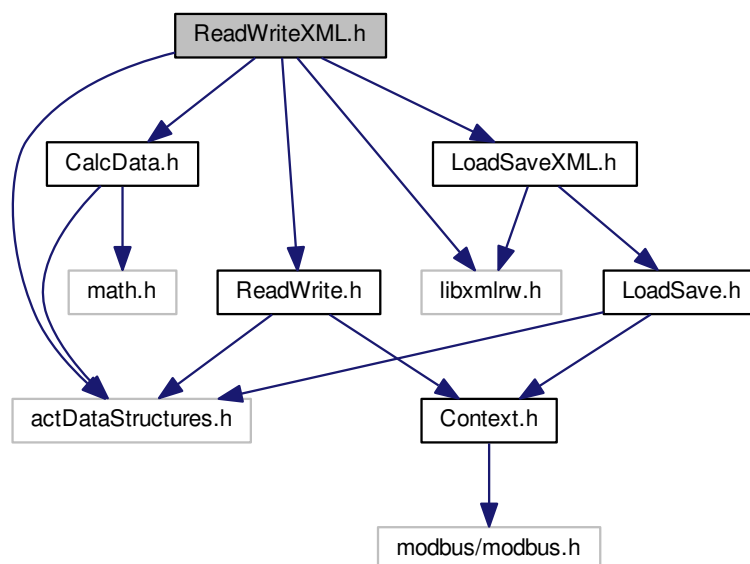
## 5.8 ReadWriteXML.h File Reference

```

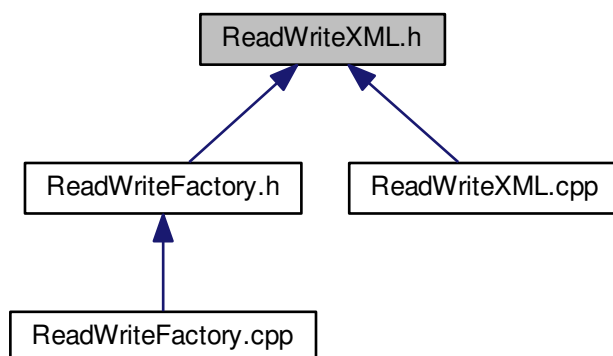
#include "ReadWrite.h"
#include "actDataStructures.h"
#include "CalcData.h"
#include "libxmlrw.h"
#include "LoadSaveXML.h"

```

Include dependency graph for ReadWriteXML.h:



This graph shows which files directly or indirectly include this file:



#### Classes

- class [ReadWriteXML](#)

## Index

- angle
  - CalcData, [4](#)
- bswap64
  - ReadWriteModbus.h, [41](#)
- CalcData, [3](#)
  - angle, [4](#)
  - CalcData, [4](#)
  - CalcData, [4](#)
  - d, [4](#)
  - h, [4](#)
  - i, [5](#)
  - iO, [5](#)
  - p, [5](#)
  - s, [5](#)
  - u, [6](#)
  - underOverfl, [6](#)
- CalcData.h, [34](#)
- chooseLSType
  - LoadSaveFactory, [15](#)
- chooseRWType
  - ReadWriteFactory, [22](#)
- connect
  - ReadWrite, [19](#)
  - ReadWriteModbus, [24](#)
  - ReadWriteXML, [31](#)
- Context, [6](#)
  - Context, [8](#)
  - getCSMPDataFileLoc, [8](#)
  - getConnType, [8](#)
  - getModbusConnected, [8](#)
  - getModbusConnection, [8](#)
  - getModbusIP, [9](#)
  - getModbusPort, [9](#)
  - getSmpConfigFileLoc, [9](#)
  - getSmpElectricityMeterConfigFileLoc, [9](#)
  - getSmpInstallConfigFileLoc, [9](#)
  - getXmlConnected, [10](#)
  - setCSMPDataFileLoc, [10](#)
  - setConnType, [10](#)
  - setModbusConnected, [10](#)
  - setModbusConnection, [10](#)
  - setModbusIP, [11](#)
  - setModbusPort, [11](#)
  - setSmpConfigFileLoc, [11](#)
  - setSmpElectricityMeterConfigFileLoc, [11](#)
  - setSmpInstallConfigFileLoc, [11](#)
  - setXmlConnected, [11](#)
- Context.h, [35](#)
- convertDate
  - ReadWriteModbus, [24](#)
- d
  - CalcData, [4](#)
- disconnect
  - ReadWrite, [19](#)
  - ReadWriteModbus, [25](#)
  - ReadWriteXML, [31](#)
- fromFloat
  - ReadWriteModbus, [25](#)
- getCSMPDataFileLoc
  - Context, [8](#)
- getConnType
  - Context, [8](#)
- getModbusConnected
  - Context, [8](#)
- getModbusConnection
  - Context, [8](#)
- getModbusIP
  - Context, [9](#)
- getModbusPort
  - Context, [9](#)
- getSmpConfigFileLoc
  - Context, [9](#)
- getSmpElectricityMeterConfigFileLoc
  - Context, [9](#)
- getSmpInstallConfigFileLoc
  - Context, [9](#)
- getXmlConnected
  - Context, [10](#)
- h
  - CalcData, [4](#)
- i
  - CalcData, [5](#)
- iO
  - CalcData, [5](#)
- initDate
  - ReadWriteModbus, [25](#)
- isConnected
  - ReadWrite, [19](#)
  - ReadWriteModbus, [26](#)
  - ReadWriteXML, [31](#)
- load
  - LoadSave, [13](#)
  - LoadSaveXML, [16](#)
- LoadSave, [12](#)
  - load, [13](#)
  - save, [13, 14](#)
- LoadSave.h, [36](#)
- LoadSaveFactory, [14](#)
  - chooseLSType, [15](#)
- LoadSaveFactory.h, [37](#)
- LoadSaveXML, [15](#)
  - load, [16](#)
  - save, [17](#)
- LoadSaveXML.h, [38](#)

## p

CalcData, 5

## read

ReadWrite, 19, 20

ReadWriteModbus, 26, 27

ReadWriteXML, 31–33

## ReadWrite, 18

connect, 19

disconnect, 19

isConnected, 19

read, 19, 20

setContext, 21

write, 21

## ReadWrite.h, 39

## ReadWriteFactory, 22

chooseRWType, 22

## ReadWriteModbus, 23

connect, 24

convertDate, 24

disconnect, 25

fromFloat, 25

initDate, 25

isConnected, 26

read, 26, 27

setContext, 27

toFloat, 27

toU64, 28

write, 28, 29

## ReadWriteModbus.h, 40

bswap64, 41

## ReadWriteXML, 29

connect, 31

disconnect, 31

isConnected, 31

read, 31–33

setContext, 33

write, 33, 34

## ReadWriteXML.h, 42

## s

CalcData, 5

## save

LoadSave, 13, 14

LoadSaveXML, 17

## setCSMPDataFileLoc

Context, 10

## setConnType

Context, 10

## setContext

ReadWrite, 21

ReadWriteModbus, 27

ReadWriteXML, 33

## setModbusConnected

Context, 10

## setModbusConnection

Context, 10

## setModbusIP

Context, 11

## setModbusPort

Context, 11

## setSmpConfigFileLoc

Context, 11

## setSmpElectricityMeterConfigFileLoc

Context, 11

## setSmpInstallConfigFileLoc

Context, 11

## setXmlConnected

Context, 11

## toFloat

ReadWriteModbus, 27

## toU64

ReadWriteModbus, 28

## u

CalcData, 6

## underOverfl

CalcData, 6

## write

ReadWrite, 21

ReadWriteModbus, 28, 29

ReadWriteXML, 33, 34