System library

1.00

Generated on Fri Nov 29 2024 for System library by Doxygen 1.12.0

Fri Nov 29 2024 22:16:29

Chapter 1

System Library

namespace : `Lib::System`

Chapter 2

Topic Index

2.1 Topics

Here is a list of all topics with brief descriptions:

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 4

Class Index

4.1 Class List

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

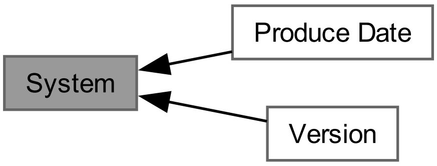
Chapter 5

Topic Documentation

5.1 System

5.1.1 Detailed Description

Collaboration diagram for System:



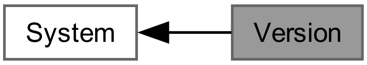
Topics

- Version
- Produce Date

5.1.2 Version

5.1.2.1 Detailed Description

Collaboration diagram for Version:



Classes

- class Lib::System::Version::Hardware

  Hardware Version.
- class Lib::System::Version::Firmware

  Firmware Version.

Variables

- Version::Firmware Lib::System::System::firmwareVersion

  Instance of the Firmware version.
- Version::Hardware Lib::System::System::hardwareVersion

  Instance of the Hardware version.

5.1.2.2 Variable Documentation

5.1.2.2.1 firmwareVersion

```
Version::Firmware Lib::System::System::firmwareVersion
```

Instance of the Firmware version.

5.1.2.2.2 hardwareVersion

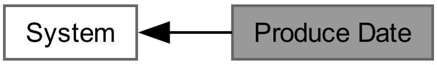
```
Version::Hardware Lib::System::System::hardwareVersion
```

Instance of the Hardware version.

5.1.3 Produce Date

5.1.3.1 Detailed Description

Collaboration diagram for Produce Date:



Classes

- class Lib::System::ProduceDate

  Produce Date class.

Variables

- ProduceDate Lib::System::System::produceDate

  Instance of the Produce date.
- ProduceDate Lib::System::System::saleDate

  Instance of the Sale date.

5.1.3.2 Variable Documentation

5.1.3.2.1 produceDate

```
ProduceDate Lib::System::System::produceDate
```

Instance of the Produce date.

5.1.3.2.2 saleDate

```
ProduceDate Lib::System::System::saleDate
```

Instance of the Sale date.

Chapter 6

Namespace Documentation

6.1 Lib Namespace Reference

Namespaces

- namespace System

  System library.

6.2 Lib::System Namespace Reference

6.2.1 Detailed Description

System library.

Classes

- class Configuration

  System configuration class.
- class Parameters

  System parameters class.
- class ProduceDate

  Produce Date class.
- class SerialNumber

  Serial Nubmer class.
- class System

  System class.
- class Version

  System Version.

Chapter 7

Class Documentation

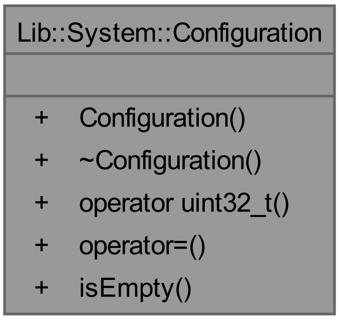
7.1 Lib::System::Configuration Class Reference

7.1.1 Detailed Description

System configuration class.

This class stores a configuration word for the system. It defines some constant parameters that the system maintains.

Collaboration diagram for Lib::System::Configuration:



Public Member Functions

- Configuration (uint32_t offset)

    Configuration ctor.
- virtual ∼Configuration ()=default

    Configuration dtor.
- operator uint32_t () const

    Reads the value from OTM Rom.
- void operator= (uint32_t value)

    Writes value to OTP Rom.
- virtual bool isEmpty () const

    Checks if the field in the OTP Flash memory where the date should be empty.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 Configuration()

```
Lib::System::Configuration::Configuration (
            uint32_t offset)  [inline]
```

Configuration ctor.

Parameters

offset	- offset in OTP flash memory

7.1.2.2 ∼Configuration()

```
virtual Lib::System::Configuration::∼Configuration ()  [virtual], [default]
```

Configuration dtor.

7.1.3 Member Function Documentation

7.1.3.1 isEmpty()

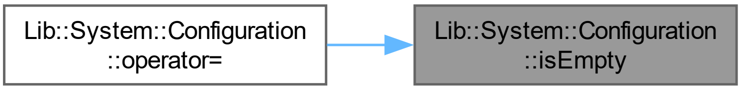
```
virtual bool Lib::System::Configuration::isEmpty () const  [inline], [virtual]
```

Checks if the field in the OTP Flash memory where the date should be empty.

Returns

bool - is empty

Here is the caller graph for this function:


Lib::System::Configuration ::operator=
Lib::System::Configuration ::isEmpty

7.1.3.2 operator uint32_t()

```
Lib::System::Configuration::operator uint32_t () const  [inline]
```

Reads the value from OTM Rom.

Returns

value

7.1.3.3 operator=()

```
void Lib::System::Configuration::operator= (
             uint32_t value)  [inline]
```

Writes value to OTP Rom.

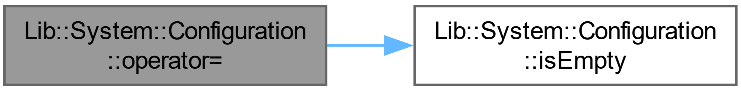
Parameters

value	-value

Returns

returns `true` on successful save

Here is the call graph for this function:

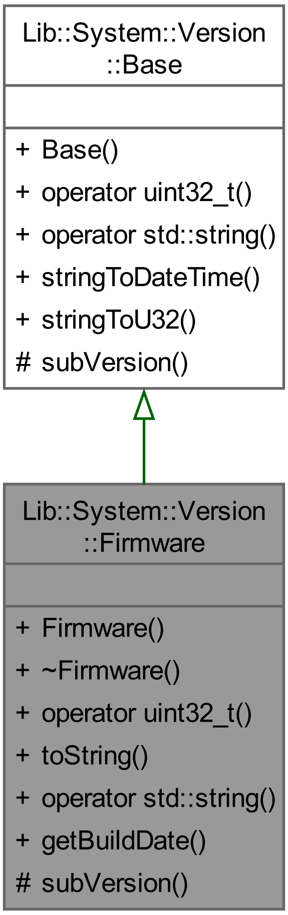

Lib::System::Configuration ::operator=
Lib::System::Configuration ::isEmpty

7.2 Lib::System::Version::Firmware Class Reference

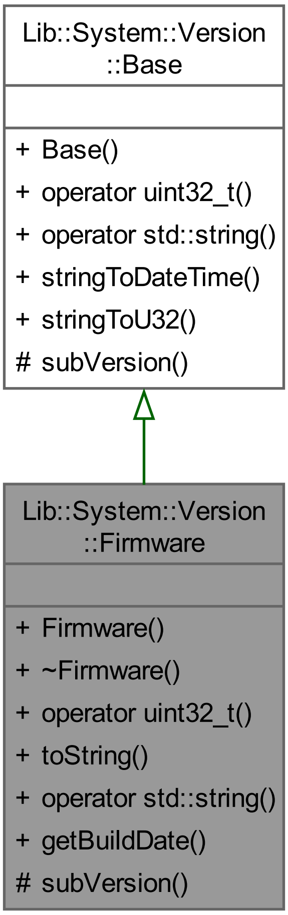
7.2.1 Detailed Description

Firmware Version.

Inheritance diagram for Lib::System::Version::Firmware:



Collaboration diagram for Lib::System::Version::Firmware:



Public Member Functions

- Firmware (const std::string &string, uint32_t subVersion)

  Firmware ctor.
- ∼Firmware ()=default

  Firmware dtor.
- virtual operator uint32_t () const

  Gets the firmware version.
- virtual std::string toString () const

  Converts version to string.
- virtual operator std::string () const

  Type convertion operator.
- virtual const std::string getBuildDate () const

  Gets the build date.

Protected Member Functions

- virtual uint32_t subVersion () const override
  Gets a sub version.

Static Protected Member Functions

- static const Lib::Helper::DateTime stringToDateTime (const char ∗timeStr, const char ∗dateStr)
  Converts the Date/Time Strings To DateTime.
- static uint32_t stringToU32 (const std::string &string)
  Converts the version string to uint32_t.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 Firmware()

```
Lib::System::Version::Firmware::Firmware (
            const std::string & string,
            uint32_t subVersion)  [inline]
```

Firmware ctor.

Parameters

string	

7.2.2.2 ∼Firmware()

```
Lib::System::Version::Firmware::∼Firmware ()  [default]
```

Firmware dtor.

7.2.3 Member Function Documentation

7.2.3.1 getBuildDate()

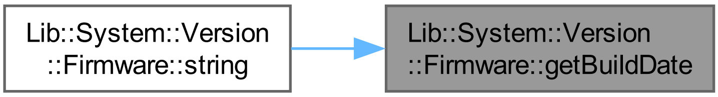
```
virtual const std::string Lib::System::Version::Firmware::getBuildDate () const  [inline],
[virtual]
```

Gets the build date.

Returns

returns a string of the compilation date

Here is the caller graph for this function:


Lib::System::Version ::Firmware::string
Lib::System::Version ::Firmware::getBuildDate

7.2.3.2 operator std::string()

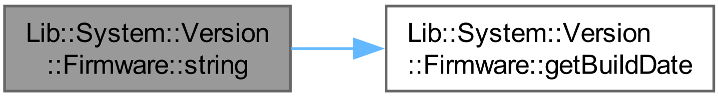
```
virtual Lib::System::Version::Firmware::operator std::string () const  [inline], [virtual]
```

Type convertion operator.

Returns

std::string

Here is the call graph for this function:


Lib::System::Version ::Firmware::string
Lib::System::Version ::Firmware::getBuildDate

7.2.3.3 operator uint32_t()

```
virtual Lib::System::Version::Firmware::operator uint32_t () const  [inline], [virtual]
```

Gets the firmware version.

Returns

uint32_t

7.2.3.4 stringToDateTime()

```
static const Lib::Helper::DateTime Lib::System::Version::Base::stringToDateTime (
             const char * timeStr,
             const char * dateStr)  [inline], [static], [inherited]
```

Converts the Date/Time Strings To DateTime.

Parameters

timeStr	
dateStr	

Returns

DateTime

7.2.3.5 stringToU32()

```
static uint32_t Lib::System::Version::Base::stringToU32 (
             const std::string & string)  [inline], [static], [inherited]
```

Converts the version string to uint32_t.

Returns

uint32_t

Note

format of version string must be `x.xx`

7.2.3.6 subVersion()

```
virtual uint32_t Lib::System::Version::Firmware::subVersion () const  [inline], [override],
[protected], [virtual]
```

Gets a sub version.

Returns

7.2.3.7 toString()

```
virtual std::string Lib::System::Version::Firmware::toString () const  [inline], [virtual]
```

Converts version to string.

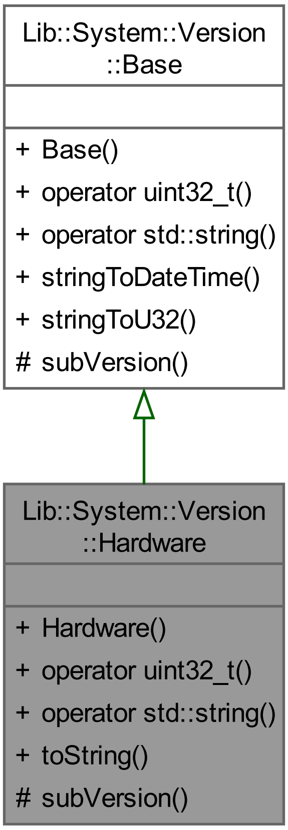
Returns

7.3 Lib::System::Version::Hardware Class Reference

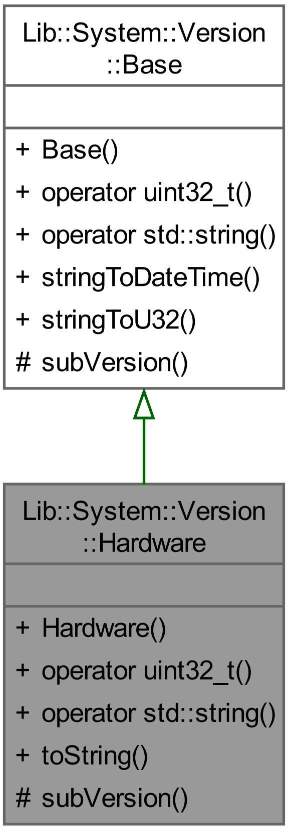
7.3.1 Detailed Description

Hardware Version.

Inheritance diagram for Lib::System::Version::Hardware:



Collaboration diagram for Lib::System::Version::Hardware:



Public Member Functions

- Hardware (const std::string &string, uint32_t subVersion)

  Hardware version ctor.
- virtual operator uint32_t () const override

  Get Hardware Version.
- virtual operator std::string () const override

  Get The Version String.
- virtual std::string toString () const

  Convert version to string.

Protected Member Functions

- virtual uint32_t subVersion () const override

Static Protected Member Functions

- static const Lib::Helper::DateTime stringToDateTime (const char ∗timeStr, const char ∗dateStr)

  Converts the Date/Time Strings To DateTime.
- static uint32_t stringToU32 (const std::string &string)

  Converts the version string to uint32_t.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 Hardware()

```
Lib::System::Version::Hardware::Hardware (
            const std::string & string,
            uint32_t subVersion)  [inline]
```

Hardware version ctor.

Parameters

string	

7.3.3 Member Function Documentation

7.3.3.1 operator std::string()

```
virtual Lib::System::Version::Hardware::operator std::string () const  [inline], [override],
[virtual]
```

Get The Version String.

Returns

std::string

7.3.3.2 operator uint32_t()

```
virtual Lib::System::Version::Hardware::operator uint32_t () const  [inline], [override],
[virtual]
```

Get Hardware Version.

Returns

uint32_t

7.3.3.3 stringToDateTime()

```
static const Lib::Helper::DateTime Lib::System::Version::Base::stringToDateTime (
            const char * timeStr,
            const char * dateStr)  [inline], [static], [inherited]
```

Converts the Date/Time Strings To DateTime.

Parameters

timeStr	
dateStr	

Returns

DateTime

7.3.3.4 stringToU32()

```
static uint32_t Lib::System::Version::Base::stringToU32 (
             const std::string & string)  [inline], [static], [inherited]
```

Converts the version string to uint32_t.

Returns

uint32_t

Note

format of version string must be `x.xx`

7.3.3.5 subVersion()

```
virtual uint32_t Lib::System::Version::Hardware::subVersion () const  [inline], [override],
[protected], [virtual]
```

7.3.3.6 toString()

```
virtual std::string Lib::System::Version::Hardware::toString () const  [inline], [virtual]
```

Convert version to string.

Returns

7.4 Lib::System::Parameters< T > Class Template Reference

7.4.1 Detailed Description

template<class T>
class Lib::System::Parameters< T >

System parameters class.

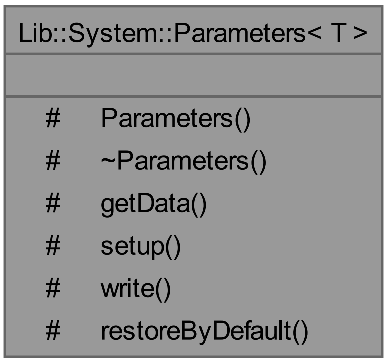
See also

Example:

`System::Parametes`

`System::ParametesData`

Collaboration diagram for Lib::System::Parameters< T >:



Classes

- class StorageBase

  StorageBase class.

Protected Member Functions

- Parameters (StorageBase ∗storage)

  Parameters ctor.
- ∼Parameters ()=default

  Parameters dtor.
- virtual T & getData ()

  Get the reference of data.
- virtual bool setup ()

  Setup the storage.
- virtual bool write ()

  Write the system parameters data.
- virtual void restoreByDefault ()=0

7.4.2 Constructor & Destructor Documentation

7.4.2.1 Parameters()

```
template<class T >
Lib::System::Parameters< T >::Parameters (
            StorageBase * storage)  [inline], [protected]
```

Parameters ctor.

Parameters

storage	- pointer to storage instance

7.4.2.2 ∼Parameters()

```
template<class T >
Lib::System::Parameters< T >::∼Parameters ()  [protected], [default]
```

Parameters dtor.

7.4.3 Member Function Documentation

7.4.3.1 getData()

```
template<class T >
virtual T & Lib::System::Parameters< T >::getData ()  [inline], [protected], [virtual]
```

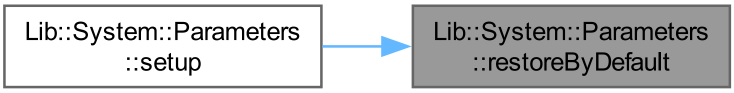
Get the reference of data.

Returns

7.4.3.2 restoreByDefault()

```
template<class T >
virtual void Lib::System::Parameters< T >::restoreByDefault ()  [protected], [pure virtual]
```

Here is the caller graph for this function:


Lib::System::Parameters ::setup
Lib::System::Parameters ::restoreByDefault

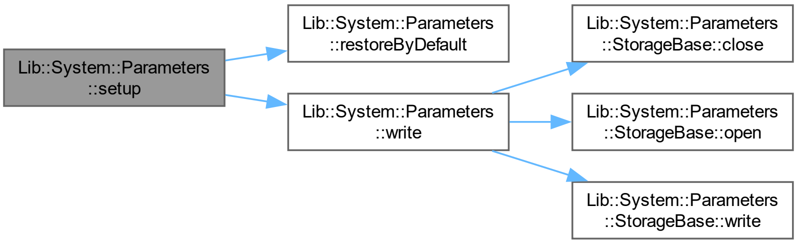
7.4.3.3 setup()

```
template<class T >
virtual bool Lib::System::Parameters< T >::setup ()  [inline], [protected], [virtual]
```

Setup the storage.

Returns

Here is the call graph for this function:


Lib::System::Parameters ::setup
Lib::System::Parameters ::restoreByDefault
Lib::System::Parameters ::write
Lib::System::Parameters ::StorageBase::close
Lib::System::Parameters ::StorageBase::open
Lib::System::Parameters ::StorageBase::write

7.4.3.4 write()

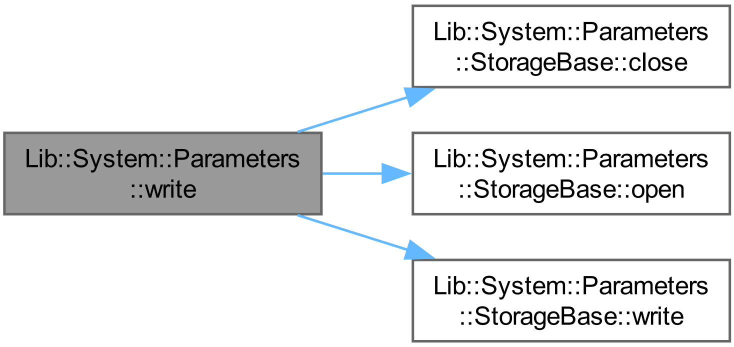
```
template<class T >
virtual bool Lib::System::Parameters< T >::write ()  [inline], [protected], [virtual]
```

Write the system parameters data.

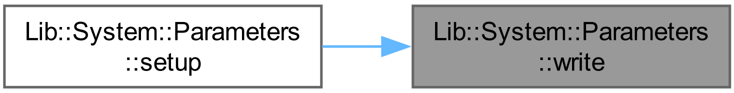
Returns

true if success

Here is the call graph for this function:


Lib::System::Parameters ::write
Lib::System::Parameters ::StorageBase::close
Lib::System::Parameters ::StorageBase::open
Lib::System::Parameters ::StorageBase::write

Here is the caller graph for this function:

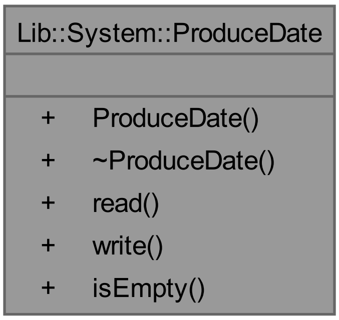

Lib::System::Parameters ::setup
Lib::System::Parameters ::write

7.5 Lib::System::ProduceDate Class Reference

7.5.1 Detailed Description

Produce Date class.

Collaboration diagram for Lib::System::ProduceDate:



Public Types

- using DateTime = Lib::Helper::DateTime

Public Member Functions

- ProduceDate (uint32_t offset)

  ProduceDate ctor.
- virtual ∼ProduceDate ()=default

  Produce Date dtor.
- virtual bool read (DateTime &result) const

  Read the Produce Date.
- virtual bool write (const DateTime &value)

  Write Produce Date in OTP ROM.
- virtual bool isEmpty () const

  Checks if the field in the OTP Flash memory where the date should be empty.

7.5.2 Member Typedef Documentation

7.5.2.1 DateTime

```
using Lib::System::ProduceDate::DateTime = Lib::Helper::DateTime
```

7.5.3 Constructor & Destructor Documentation

7.5.3.1 ProduceDate()

```
Lib::System::ProduceDate::ProduceDate (
            uint32_t offset)  [inline]
```

ProduceDate ctor.

Parameters

offset	- offset in OTP Flash Memory

7.5.3.2 ∼ProduceDate()

```
virtual Lib::System::ProduceDate::∼ProduceDate ()  [virtual], [default]
```

Produce Date dtor.

7.5.4 Member Function Documentation

7.5.4.1 isEmpty()

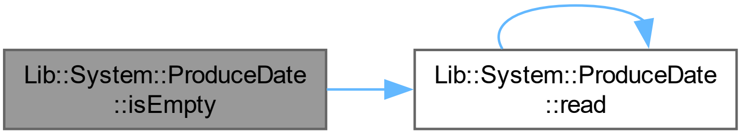
```
virtual bool Lib::System::ProduceDate::isEmpty () const  [inline], [virtual]
```

Checks if the field in the OTP Flash memory where the date should be empty.

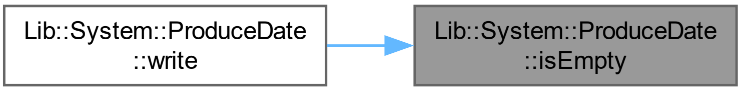
Returns

bool - is empty

Here is the call graph for this function:


Lib::System::ProduceDate ::isEmpty
Lib::System::ProduceDate ::read

Here is the caller graph for this function:



7.5.4.2 read()

```
virtual bool Lib::System::ProduceDate::read (
             DateTime & result) const  [inline], [virtual]
```

Read the Produce Date.

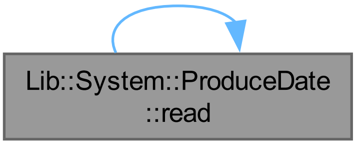
Parameters

result	reference to result

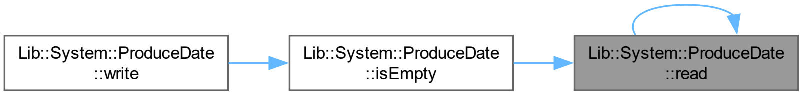
Returns

bool

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.4.3 write()

```
virtual bool Lib::System::ProduceDate::write (
             const DateTime & value) [inline], [virtual]
```

Write Produce Date in OTP ROM.

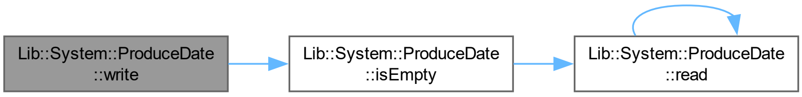
Parameters

value	- reference to DateTime instance

Returns

bool - truth in a successful record

Here is the call graph for this function:

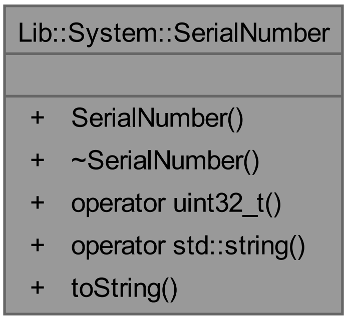

Lib::System::ProduceDate ::write
Lib::System::ProduceDate ::isEmpty
Lib::System::ProduceDate ::read

7.6 Lib::System::SerialNumber Class Reference

7.6.1 Detailed Description

Serial Nubmer class.

Collaboration diagram for Lib::System::SerialNumber:



Public Member Functions

- SerialNumber ()

  Serial Number ctor.
- virtual ∼SerialNumber ()=default

  Serial Number dtor.
- virtual operator uint32_t () const

  Type Convertion operator.
- virtual operator std::string () const

  Type Convertion operator.
- virtual std::string toString () const

  get Serial Number as string

7.6.2 Constructor & Destructor Documentation

7.6.2.1 SerialNumber()

```
Lib::System::SerialNumber::SerialNumber ()  [inline]
```

Serial Number ctor.

7.6.2.2 ∼SerialNumber()

```
virtual Lib::System::SerialNumber::∼SerialNumber ()  [virtual], [default]
```

Serial Number dtor.

7.6.3 Member Function Documentation

7.6.3.1 operator std::string()

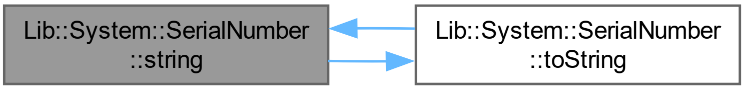
```
virtual Lib::System::SerialNumber::operator std::string () const  [inline], [virtual]
```

Type Convertion operator.

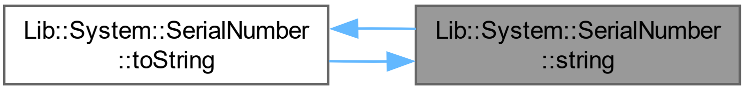
Returns

uint32_t

Here is the call graph for this function:


Lib::System::SerialNumber ::string
Lib::System::SerialNumber ::toString

Here is the caller graph for this function:


Lib::System::SerialNumber ::toString
Lib::System::SerialNumber ::string

7.6.3.2 operator uint32_t()

```
virtual Lib::System::SerialNumber::operator uint32_t () const  [inline], [virtual]
```

Type Convertion operator.

Returns

uint32_t

7.6.3.3 toString()

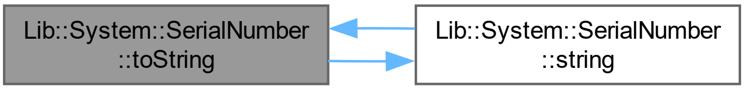
```
virtual std::string Lib::System::SerialNumber::toString () const  [inline], [virtual]
```

get Serial Number as string

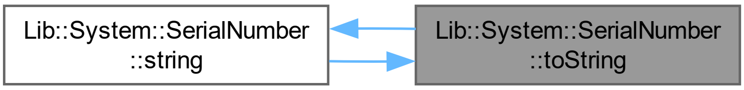
Returns

const std::string

Here is the call graph for this function:


Lib::System::SerialNumber ::toString
Lib::System::SerialNumber ::string

Here is the caller graph for this function:


Lib::System::SerialNumber ::string
Lib::System::SerialNumber ::toString

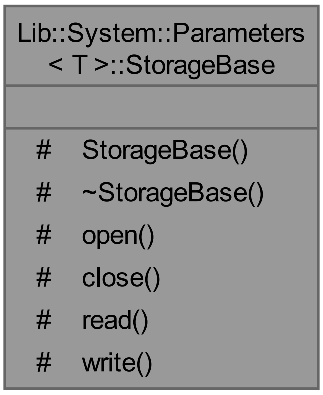
7.7 Lib::System::Parameters< T >::StorageBase Class Reference

7.7.1 Detailed Description

template< class T >
class Lib::System::Parameters< T >::StorageBase

StorageBase class.

Collaboration diagram for Lib::System::Parameters< T >::StorageBase:



Protected Member Functions

- StorageBase ()=default

    Storage Base ctor.
- virtual ∼StorageBase ()=default
- virtual bool open ()=0

    Open the _storage.
- virtual bool close ()=0

    Close the _storage.
- virtual bool read (void ∗dst, size_t len)=0

    Read the parameters data.
- virtual bool write (void ∗src, size_t len)=0

    Write the parameters data.

Friends

- class Parameters

7.7.2 Constructor & Destructor Documentation

7.7.2.1 StorageBase()

```
template<class T >
Lib::System::Parameters< T >::StorageBase::StorageBase ()  [protected], [default]
```

Storage Base ctor.

7.7.2.2 ∼StorageBase()

```
template<class T >
virtual Lib::System::Parameters< T >::StorageBase::∼StorageBase () [protected], [virtual],
[default]
```

7.7.3 Member Function Documentation

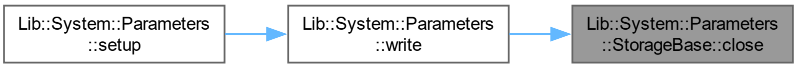
7.7.3.1 close()

```
template<class T >
virtual bool Lib::System::Parameters< T >::StorageBase::close () [protected], [pure virtual]
```

Close the _storage.

Returns

Here is the caller graph for this function:


Lib::System::Parameters ::setup
Lib::System::Parameters ::write
Lib::System::Parameters ::StorageBase::close

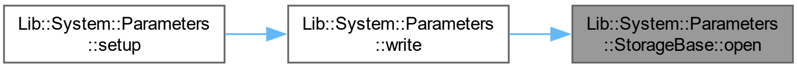
7.7.3.2 open()

```
template<class T >
virtual bool Lib::System::Parameters< T >::StorageBase::open () [protected], [pure virtual]
```

Open the _storage.

Returns

Here is the caller graph for this function:


Lib::System::Parameters ::setup
Lib::System::Parameters ::write
Lib::System::Parameters ::StorageBase::open

7.7.3.3 read()

```
template<class T >
virtual bool Lib::System::Parameters< T >::StorageBase::read (
             void * dst,
             size_t len)  [protected], [pure virtual]
```

Read the parameters data.

Returns

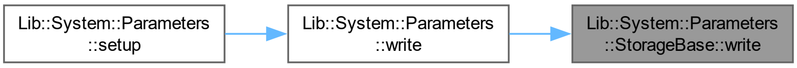
7.7.3.4 write()

```
template<class T >
virtual bool Lib::System::Parameters< T >::StorageBase::write (
             void * src,
             size_t len)  [protected], [pure virtual]
```

Write the parameters data.

Returns

Here is the caller graph for this function:


Lib::System::Parameters ::setup
Lib::System::Parameters ::write
Lib::System::Parameters ::StorageBase::write

7.7.4 Friends And Related Symbol Documentation

7.7.4.1 Parameters

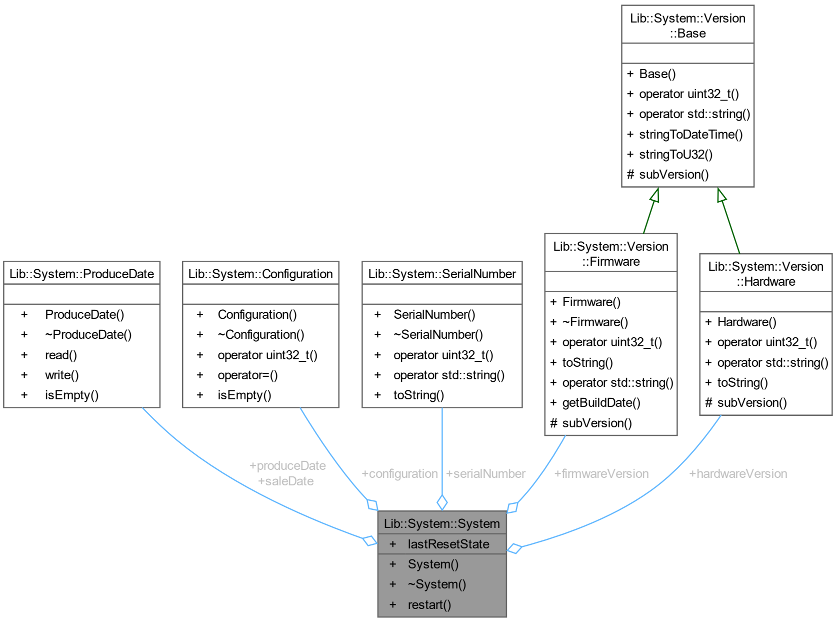
```
template<class T >
friend class Parameters  [friend]
```

7.8 Lib::System::System Class Reference

7.8.1 Detailed Description

System class.

Collaboration diagram for Lib::System::System:



Public Types

- using LastResetState = Lib::HAL::LastResetState::State

Public Member Functions

- System (const std::string &fwVersionString, uint32_t fwSubversion, const std::string &hwVersionString, uint32_t hwSubversion)

  System ctor.
- ∼System ()=default

  System dtor.
- void restart ()

  Restarts the system.

Public Attributes

- LastResetState lastResetState

  Instance of the Last reset state.
- ProduceDate produceDate

  Instance of the Produce date.
- ProduceDate saleDate

  Instance of the Sale date.
- Configuration configuration

  Instance of the System configuration word.
- SerialNumber serialNumber

  Instance of the system serial number.
- Version::Firmware firmwareVersion

  Instance of the Firmware version.
- Version::Hardware hardwareVersion

  Instance of the Hardware version.

7.8.2 Member Typedef Documentation

7.8.2.1 LastResetState

```
using Lib::System::System::LastResetState = Lib::HAL::LastResetState::State
```

7.8.3 Constructor & Destructor Documentation

7.8.3.1 System()

```
Lib::System::System::System (
            const std::string & fwVersionString,
            uint32_t fwSubversion,
            const std::string & hwVersionString,
            uint32_t hwSubversion) [inline]
```

System ctor.

Parameters

fwVersionString	Firmware version string
fwSubversion	Firmware sub-version
hwVersionString	Hardware version string
hwSubversion	Hardware sub-version

7.8.3.2 ∼System()

```
Lib::System::System::∼System () [default]
```

System dtor.

7.8.4 Member Function Documentation

7.8.4.1 restart()

```
void Lib::System::System::restart ()  [inline]
```

Restarts the system.

7.8.5 Member Data Documentation

7.8.5.1 configuration

```
Configuration Lib::System::System::configuration
```

Instance of the System configuration word.

7.8.5.2 lastResetState

```
LastResetState Lib::System::System::lastResetState
```

Instance of the Last reset state.

7.8.5.3 serialNumber

```
SerialNumber Lib::System::System::serialNumber
```

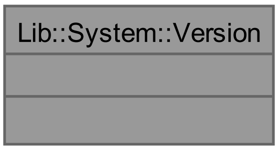
Instance of the system serial number.

7.9 Lib::System::Version Class Reference

7.9.1 Detailed Description

System Version.

Collaboration diagram for Lib::System::Version:



Classes

- class Firmware

  Firmware Version.
- class Hardware

  Hardware Version.

Index