# Bergermeister Home Automation

Source Code Documentation (SCD)

# Contents

# Chapter 1

# Namespace Index

## 1.1 Namespace List

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

# Chapter 2

# Hierarchical Index

## 2.1 Class Hierarchy

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

# Chapter 3

# Class Index

## 3.1 Class List

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

# Chapter 4

# File Index

## 4.1 File List

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

# Chapter 5

# Namespace Documentation

## 5.1 GNCommon Namespace Reference

Namespace containing Common components and infrastrucutre.

### Namespaces

- NDataAuthentication

  *Namespace containing Data Authentication and Validity Checking utilities.*
- NNotification

  *Namespace containing system Alerts.*

### Classes

- class GTcStopWatch

### Typedefs

- typedef bool Tb8
- typedef char Tc8
- typedef signed char Ti8
- typedef unsigned char Tu8
- typedef signed short Ti16
- typedef unsigned short Tu16
- typedef signed long Ti32
- typedef unsigned long Tu32
- typedef signed long long Ti64
- typedef unsigned long long Tu64
- typedef float Tf32
- typedef double Tf64

### 5.1.1 Detailed Description

Namespace containing Common components and infrastrucutre.

### 5.1.2 Typedef Documentation

#### 5.1.2.1 Tb8

```
typedef bool GNCommon::Tb8
```

Type definition for 8-bit boolean primitive

Definition at line 54 of file Types.h.

#### 5.1.2.2 Tc8

```
typedef char GNCommon::Tc8
```

Type definition for 8-bit character primitive

Definition at line 55 of file Types.h.

#### 5.1.2.3 Tf32

```
typedef float GNCommon::Tf32
```

Type definition for 32-bit single-precision floating point primitive

Definition at line 64 of file Types.h.

#### 5.1.2.4 Tf64

```
typedef double GNCommon::Tf64
```

Type definition for 64-bit double-precision floating point primitive

Definition at line 65 of file Types.h.

#### 5.1.2.5 Ti16

```
typedef signed short GNCommon::Ti16
```

Type definition for signed 16-bit integer primitive

Definition at line 58 of file Types.h.

#### 5.1.2.6 Ti32

```
typedef signed long GNCommon::Ti32
```

Type definition for signed 32-bit integer primitive

Definition at line 60 of file Types.h.

#### 5.1.2.7 Ti64

```
typedef signed long long GNCommon::Ti64
```

Type definition for signed 64-bit integer primitive

Definition at line 62 of file Types.h.

#### 5.1.2.8 Ti8

```
typedef signed char GNCommon::Ti8
```

Type definition for signed 8-bit integer primitive

Definition at line 56 of file Types.h.

#### 5.1.2.9 Tu16

```
typedef unsigned short GNCommon::Tu16
```

Type definition for unsigned 16-bit integer primitive

Definition at line 59 of file Types.h.

#### 5.1.2.10 Tu32

```
typedef unsigned long GNCommon::Tu32
```

Type definition for unsigned 32-bit primitive

Definition at line 61 of file Types.h.

#### 5.1.2.11 Tu64

```
typedef unsigned long long GNCommon::Tu64
```

Type definition for unsigned 64-bit integer primitive

Definition at line 63 of file Types.h.

#### 5.1.2.12 Tu8

```
typedef unsigned char GNCommon::Tu8
```

Type definition for unsigned 8-bit integer primitive

Definition at line 57 of file Types.h.

## 5.2 GNCommon::NDataAuthentication Namespace Reference

Namespace containing Data Authentication and Validity Checking utilities.

### Classes

- class TcCRC32

### 5.2.1 Detailed Description

Namespace containing Data Authentication and Validity Checking utilities.

## 5.3 GNCommon::NNotification Namespace Reference

Namespace containing system Alerts.

### Classes

- class TcAlert
- class TcIdentifier
- class TcStatus

### Enumerations

- enum TcComponentId : Tu8 { TcComponentId::XeNone = 0, TcComponentId::XeServer = 1, TcComponentId::XeSensor = 2 }

  *Enumeration of Alert Component Identifiers.*
- enum TcCriticality : Tu8 { TcCriticality::XeNone = 0, TcCriticality::XeNotice = 1, TcCriticality::XeWarning = 2, TcCriticality::XeAlarm = 3 }
- enum TcEventId : Tu32 { **XeIdentifier**, **XeStatus**, **XeTimestamp**, **XeData** }
- enum TcGroupId : Tu8 { TcGroupId::XeNone = 0, TcGroupId::XeNetwork = 1 }

## 5.3.1 Detailed Description

Namespace containing system Alerts.

## 5.3.2 Enumeration Type Documentation

### 5.3.2.1 TcComponentId

```
enum GNCommon::NNotification::TcComponentId :  Tu8  [strong]
```

Enumeration of Alert Component Identifiers.

**Enumerator**

| | |
|---|---|
| XeNone | Enumerated Component: None |
| XeServer | Enumerated Component: Server |
| XeSensor | Enumerated Component: Sensor |

Definition at line 12 of file ComponentId.h.

### 5.3.2.2 TcCriticality

```
enum GNCommon::NNotification::TcCriticality :  Tu8  [strong]
```

Enumeration of Alert Criticality Identifiers

**Enumerator**

| | |
|---|---|
| XeNone | Enumerated Alert Level: None |
| XeNotice | Enumerated Alert Level: Notice |
| XeWarning | Enumerated Alert Level: Warning |
| XeAlarm | Enumerated Alert Level: Alarm |

Definition at line 12 of file Criticality.h.

#### 5.3.2.3 TcEventId

```
enum GNCommon::NNotification::TcEventId :  Tu32  [strong]
```

Enumeration of Alert Event Identifiers

Definition at line 12 of file EventId.h.

#### 5.3.2.4 TcGroupId

```
enum GNCommon::NNotification::TcGroupId :  Tu8  [strong]
```

Enumeration of Group Identifiers

**Enumerator**

| | |
|---|---|
| XeNone | Enumerated Group: None |
| XeNetwork | Enumerated Group: Network |

Definition at line 12 of file GroupId.h.

# Chapter 6

# Class Documentation

## 6.1 GNCommon::NComponent::GTcEvent< auiMaxListeners > Class Template Reference

### Public Member Functions

- void **MNotify** (void ∗aopSender, const Tu32 auiIdentifier)
- GTcEvent & **operator+=** (const GTcListener &aorListener)
- GTcEvent & **operator-=** (const GTcListener &aorListener)

### Protected Attributes

- GNCommon::NContainers::GTcListNode< GTcListener > **voBuffer** [auiMaxListeners]
- GNCommon::NContainers::GTcLinkedList< GTcListener > **voListeners**

### 6.1.1 Detailed Description

**template< Tu32 auiMaxListeners >**
**class GNCommon::NComponent::GTcEvent< auiMaxListeners >**

Definition at line 14 of file Event.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Component/Event.h

## 6.2 GNCommon::NContainers::GTcLinkedList< GTcType > Class Template Reference

### Public Member Functions

- void **MInitialize** (GTcListNode< GTcType > ∗aopBuffer, Tu32 auiSize)
- Tb8 **MIsInitialized** (void) const
- Tb8 **MInsertAtHead** (GTcType &aorObject)
- Tb8 **MInsertAtTail** (GTcType &aorObject)
- Tb8 **MInsert** (GTcType &aorObject)
- Tb8 **MRemoveHead** (void)
- Tb8 **MRemoveTail** (void)
- Tb8 **MRemove** (GTcType &aorObject)

### Protected Attributes

- GTcListNode< GTcType > ∗ **vopBuff**
- GTcListNode< GTcType > ∗ **vopHead**
- GTcListNode< GTcType > ∗ **vopTail**
- Tu32 **vuiSize**
- Tu32 **vuiCount**
- Tb8 **vbInitialized**

### 6.2.1 Detailed Description

**template< class GTcType>**
**class GNCommon::NContainers::GTcLinkedList< GTcType >**

Definition at line 11 of file LinkedList.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Containers/LinkedList.h

## 6.3 GNCommon::NContainers::GTcList< GTcType > Class Template Reference

### Public Member Functions

- **GTcList** (GTcType ∗aopBuffer, const Tu32 auiCapacity)
- **GTcList** (const GTcList< GTcType > &aorList)
- virtual GTcList< GTcType > & **operator=** (const GTcList< GTcType > &aorList)
- virtual Tb8 **MAdd** (const GTcType &aorItem)
- virtual Tb8 **MInsert** (const GTcType &aorItem, const Tu32 auiIndex)
- virtual Tb8 **MRemove** (const GTcType &aorItem)
- virtual Tb8 **MRemvoeAt** (const Tu32 auiIndex)
- virtual void **MClear** (void)
- virtual Tb8 **MContains** (const GTcType &aorItem) const
- virtual Tu32 **MIndexOf** (const GTcType &aorItem) const
- Tu32 **MCapacity** (void) const
- Tu32 **MCount** (void) const

### Public Attributes

- GTcType ∗& **VorItem**
- const Tu32 & **VuirCapacity**
- const Tu32 & **VuirCount**

### Protected Attributes

- GTcType ∗ **vopBuffer**
- Tu32 **vuiCapacity**
- Tu32 **vuiCount**

### 6.3.1 Detailed Description

**template< class GTcType >**
**class GNCommon::NContainers::GTcList< GTcType >**

Definition at line 8 of file List.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Containers/List.h

## 6.4 GNCommon::NComponent::GTcListener Class Reference

### Public Types

- typedef void(∗ **TsHandle**) (void ∗aopListener, void ∗aopParameter)

### Public Member Functions

- **GTcListener** (void ∗aopInstance, const TsHandle aopHandle)
- void ∗ **MGetInstance** (void) const
- TsHandle **MGetHandle** (void) const
- Tb8 **operator==** (const GTcListener &aorListener)

### Protected Attributes

- void ∗ **vopInstance**
- TsHandle **vopHandle**

### 6.4.1 Detailed Description

Definition at line 7 of file Listener.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Component/Listener.h
- C:/Projects/BergermeisterHome/Software/Common/src/Component/Listener.cpp

## 6.5 GNCommon::NContainers::GTcListNode< GTcType > Class Template Reference

### Public Member Functions

- void **MSetObject** (GTcType &aorObject)
- GTcType ∗ **MGetObject** (void)
- void **MSetNext** (GTcListNode< GTcType > ∗aopNode)
- void **MSetPrev** (GTcListNode< GTcType > ∗aopNode)
- GTcListNode< GTcType > ∗ **MSGetNext** (void)
- GTcListNode< GTcType > ∗ **MSGetPrev** (void)
- Tb8 **MInsertAfter** (GTcListNode< GTcType > &aorNode)
- Tb8 **MInsertBefore** (GTcListNode< GTcType > &aorNode)
- Tb8 **MRemove** (void)

### Protected Attributes

- GTcListNode< GTcType > ∗ **vopNext**
- GTcListNode< GTcType > ∗ **vopPrev**
- GTcType **voObject**
- Tb8 **vbAvailable**

### 6.5.1 Detailed Description

**template< class GTcType >**
**class GNCommon::NContainers::GTcListNode< GTcType >**

Definition at line 11 of file ListNode.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Containers/ListNode.h

## 6.6 GNCommon::NContainers::GTcQueue< GTcType > Class Template Reference

### Public Member Functions

- **GTcQueue** (GTcType ∗aopBuffer, Tu32 auiSize)
- Tb8 **MEnqueue** (GTcType &aorElement)
- Tb8 **MDequeue** (GTcType &aorElement)
- Tb8 **MIsEmpty** (void) const
- Tb8 **MIsFull** (void) const
- Tu32 **MCount** (void) const

### 6.6.1 Detailed Description

**template< class GTcType >**
**class GNCommon::NContainers::GTcQueue< GTcType >**

Definition at line 8 of file Queue.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Containers/Queue.h

## 6.7 GNCommon::GTcStopWatch Class Reference

### Public Member Functions

- **GTcStopWatch** (const GTcStopWatch &aorStopWatch)
- GTcStopWatch & **operator=** (const GTcStopWatch &aorStopWatch)
- void **MStart** (void)
- Tu64 **MStop** (void)
- Tu64 **MElapsed** (void)

### Protected Attributes

- Ti64 **vlStart**
- Ti64 **vlEnd**
- Tb8 **vbRunning**

### Static Protected Attributes

- static const Ti64 **xlMaxQuadPart** = 9223372036854775807
- static const Tu64 xulTimeBase = 1000000LL
- static Tu64 **vulFrequency** = 0

### 6.7.1 Detailed Description

Definition at line 10 of file StopWatch.h.

### 6.7.2 Member Data Documentation

#### 6.7.2.1 xulTimeBase

```
const Tu64 GNCommon::GTcStopWatch::xulTimeBase = 1000000LL  [static], [protected]
```

Microseconds

Definition at line 14 of file StopWatch.h.

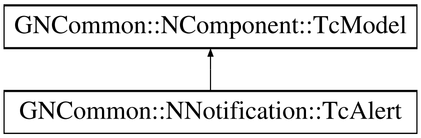
The documentation for this class was generated from the following files:

- C:/Projects/BergermeisterHome/Software/Common/inc/StopWatch.h
- C:/Projects/BergermeisterHome/Software/Common/src/StopWatch.cpp

## 6.8 GNCommon::NNotification::TcAlert Class Reference

```
#include <Alert.h>
```

Inheritance diagram for GNCommon::NNotification::TcAlert:


GNCommon::NComponent::TcModel
GNCommon::NNotification::TcAlert

### Public Member Functions

- TcAlert (void)
- ∼TcAlert (void)
- TcAlert (const TcAlert &aorAlert)
- TcAlert & operator= (const TcAlert &aorAlert)
- void MSetIdentifier (const TcIdentifier &aorID)
- void MSetStatus (const TcStatus &aorStatus)
- void MSetTimestamp (const Tu64 aulTimestamp)
- void MSetData (const Tu64 aulData)
- TcIdentifier MGetIdentifier (void) const
- TcStatus MGetStatus (void) const
- Tu64 MGetTimestamp (void) const
- Tu64 MGetData (void) const

### Static Public Attributes

- static const Tu32 XuiSizeOfIdentifier = sizeof( TcIdentifier )
- static const Tu32 XuiSizeOfStatus = sizeof( TcStatus )

### Protected Attributes

- TcIdentifier voID
- TcStatus voStatus
- Tu64 vulTimestamp
- Tu64 vulData

### Additional Inherited Members

### 6.8.1 Detailed Description

Alert Class

Definition at line 19 of file Alert.h.

### 6.8.2 Constructor & Destructor Documentation

#### 6.8.2.1 TcAlert() [1/2]

```
TcAlert::TcAlert (
             void  )
```

Default Constructor

Definition at line 13 of file Alert.cpp.

#### 6.8.2.2 ∼TcAlert()

```
TcAlert::∼TcAlert (
              void  )
```

Default Destructor

Definition at line 24 of file Alert.cpp.

#### 6.8.2.3 TcAlert() [2/2]

```
TcAlert::TcAlert (
              const TcAlert & aorAlert )
```

Copy Constructor

**Parameters**

| | |
|---|---|
| *aorAlert* | Alert Constant Reference to copy |

Definition at line 33 of file Alert.cpp.

### 6.8.3 Member Function Documentation

#### 6.8.3.1 MGetData()

```
Tu64 TcAlert::MGetData (
              void  ) const
```

MGetTrigger

**Returns**

this->vulData constant 64-bit Integer Data

Definition at line 136 of file Alert.cpp.

#### 6.8.3.2 MGetIdentifier()

```
TcIdentifier TcAlert::MGetIdentifier (
              void  ) const
```

MGetIdentifier

**Returns**

this->voID constant TsIdentifier

Definition at line 109 of file Alert.cpp.

#### 6.8.3.3 MGetStatus()

```
TcStatus TcAlert::MGetStatus (
              void  ) const
```

MGetStatus

**Returns**

this->voStatus constant TsStatus

Definition at line 118 of file Alert.cpp.

#### 6.8.3.4 MGetTimestamp()

```
Tu64 TcAlert::MGetTimestamp (
              void  ) const
```

MGetTimestamp

**Returns**

this->vulTimestamp constant 64-bit Integer Timestamp

Definition at line 127 of file Alert.cpp.

#### 6.8.3.5 MSetData()

```
void TcAlert::MSetData (
              const Tu64 aulData )
```

MSetData

**Parameters**

| | |
|---|---|
| *aulData* | constant 64-bit Integer Data |

Definition at line 96 of file Alert.cpp.

### 6.8.3.6 MSetIdentifier()

```
void TcAlert::MSetIdentifier (
             const TcIdentifier & aorID )
```

MSetIdentifier

**Parameters**

| | |
|---|---|
| *aorID* | Identifier Structure Constant Reference |

Definition at line 57 of file Alert.cpp.

### 6.8.3.7 MSetStatus()

```
void TcAlert::MSetStatus (
             const TcStatus & aorStatus )
```

MSetStatus

**Parameters**

| | |
|---|---|
| *aorStatus* | Status Structure Constant Reference |

Definition at line 70 of file Alert.cpp.

### 6.8.3.8 MSetTimestamp()

```
void TcAlert::MSetTimestamp (
             const Tu64 aulTimestamp )
```

MSetTimeStamp

**Parameters**

| | |
|---|---|
| *aulTimestamp* | constant 64-bit Integer Timestamp |

Definition at line 83 of file Alert.cpp.

#### 6.8.3.9 operator=()

```
TcAlert & TcAlert::operator= (
             const TcAlert & aorAlert )
```

operator= Override

**Parameters**

| | |
|---|---|
| *aorAlert* | Alert Constant Reference to copy |

**Returns**

∗this GTcAlert Reference

Definition at line 43 of file Alert.cpp.

### 6.8.4 Member Data Documentation

#### 6.8.4.1 voID

```
TcIdentifier GNCommon::NNotification::TcAlert::voID  [protected]
```

Encoded Identifier

Definition at line 26 of file Alert.h.

#### 6.8.4.2 voStatus

```
TcStatus GNCommon::NNotification::TcAlert::voStatus  [protected]
```

Encoded Status

Definition at line 27 of file Alert.h.

#### 6.8.4.3 vulData

```
Tu64 GNCommon::NNotification::TcAlert::vulData  [protected]
```

64-bit Additional Data

Definition at line 29 of file Alert.h.

#### 6.8.4.4 vulTimestamp

```
Tu64 GNCommon::NNotification::TcAlert::vulTimestamp  [protected]
```

64-bit Timestamp of Occurrence

Definition at line 28 of file Alert.h.

#### 6.8.4.5 XuiSizeOfIdentifier

```
const Tu32 GNCommon::NNotification::TcAlert::XuiSizeOfIdentifier = sizeof( TcIdentifier )  [static]
```

Size of Identifier

Definition at line 22 of file Alert.h.

#### 6.8.4.6 XuiSizeOfStatus

```
const Tu32 GNCommon::NNotification::TcAlert::XuiSizeOfStatus = sizeof( TcStatus )  [static]
```

Size of Status

Definition at line 23 of file Alert.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Alert.h
- C:/Projects/BergermeisterHome/Software/Common/src/Notification/Alert.cpp

## 6.9 GNCommon::NDataAuthentication::TcCRC32 Class Reference

### Public Member Functions

- Tu32 **MGet** (const Tu8 ∗aucpBuffer, const Tu32 auiBytes, const Tu32 auiSeed=xuiDefaultSeed) const
- Tb8 **MVerify** (void) const

### Protected Member Functions

- Tu32 **mGetLE** (const Tu8 ∗aucpBuffer, const Tu32 auiBytes, const Tu32 auiSeed=xuiDefaultSeed) const
- Tu32 **mGetBE** (const Tu8 ∗aucpBuffer, const Tu32 auiBytes, const Tu32 auiSeed=xuiDefaultSeed) const

### Static Protected Attributes

- static const Tu32 **xuiDefaultSeed** = 0xFFFFFFFF
- static const Tu32 **xuiTableCRC** = 0x6FCF9E13
- static const Tu32 **xuiShift8** = 8
- static const Tu32 **xuiMaskByte** = 0x000000FF
- static const Tu32 **xuiMaskAlign** = 0x00000003
- static const Tu32 **xuiTableSize** = 256
- static const Tu32 xuiTable [xuiTableSize]

### 6.9.1 Detailed Description

Definition at line 14 of file CRC32.h.

### 6.9.2 Member Data Documentation

#### 6.9.2.1 xuiTable

```
const Tu32 TcCRC32::xuiTable  [static], [protected]
```

Table obtained from `http://www.efg2.com/Lab/Mathematics/CRC.htm`

Definition at line 23 of file CRC32.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/DataAuthentication/CRC32.h
- C:/Projects/BergermeisterHome/Software/Common/src/DataAuthentication/CRC32.cpp

## 6.10 GNCommon::NNotification::TcIdentifier Class Reference

```
#include <Identifier.h>
```

### Public Member Functions

- TcIdentifier (void)
- TcIdentifier (const TcIdentifier &aorIdentifier)
- ∼TcIdentifier (void)
- TcIdentifier & operator= (const TcIdentifier &aorIdentifier)
- Tb8 operator== (const TcIdentifier &aorIdentifier)
- Tb8 operator!= (const TcIdentifier &aorIdentifier)

### Public Attributes

- Tu8 vucIndex
- TcGroupId voGroup
- TcComponentId voCompDet
- TcComponentId voCompGen

## 6.10.1 Detailed Description

Alert Identifier

Definition at line 16 of file Identifier.h.

## 6.10.2 Constructor & Destructor Documentation

### 6.10.2.1 TcIdentifier() [1/2]

```
TcIdentifier::TcIdentifier (
            void  )
```

Default Constructor

Initializes the internal members to default values.

**Returns**

None

**Formal Parameters**

```
None
```

**Local Symbols**

```
None
```

Definition at line 27 of file Identifier.cpp.

#### 6.10.2.2 TcIdentifier() [2/2]

```
TcIdentifier::TcIdentifier (
            const TcIdentifier & aorIdentifier )
```

Copy Constructor

Copies the internal members of the given Identifier to this Identifier via the assignment operator.

**Returns**

None

**Formal Parameters**

```
[ in ]  aorIdentifier    Identifier object reference to be copied
```

**Local Symbols**

```
None
```

Definition at line 48 of file Identifier.cpp.

#### 6.10.2.3 ∼TcIdentifier()

```
TcIdentifier::∼TcIdentifier (
            void  )
```

Destructor

Nothing to destruct.

**Returns**

None

**Formal Parameters**

```
None
```

**Local Symbols**

```
None
```

Definition at line 66 of file Identifier.cpp.

### 6.10.3 Member Function Documentation

#### 6.10.3.1 operator"!=()

```
Tb8 TcIdentifier::operator!= (
            const TcIdentifier & aorIdentifier )
```

Inequality Operator

Compares this Identifier to the given Identifier and returns if they are not equal.

**Returns**

True if any of this Identifier's internal members are not equal to the given Identifier's internal members, False otherwise

**Formal Parameters**

```
[ in ]  aorIdentifier    Identifier object reference to be compared
```

**Local Symbols**

```
None
```

Definition at line 135 of file Identifier.cpp.

#### 6.10.3.2 operator=()

```
TcIdentifier & TcIdentifier::operator= (
            const TcIdentifier & aorIdentifier )
```

Assignment Operator

Copies the internal members of the given Identifier

**Returns**

Identifier object reference to this Identifier

**Formal Parameters**

```
[ in ]  aorIdentifier    Identifier object reference to be copied
```

**Local Symbols**

```
None
```

Definition at line 84 of file Identifier.cpp.

#### 6.10.3.3 operator==()

```
Tb8 TcIdentifier::operator== (
              const TcIdentifier & aorIdentifier )
```

Equality Operator

Compares this Identifier to the given Identifier and returns if they are equal.

**Returns**

True if all of this Identifier's internal members equal the given Identifier's internal members, False otherwise

**Formal Parameters**

```
[ in ]  aorIdentifier    Identifier object reference to be compared
```

**Local Symbols**

```
kbEqual    Flag indicating if the Identifiers are equal
```

Definition at line 107 of file Identifier.cpp.

### 6.10.4 Member Data Documentation

#### 6.10.4.1 voCompDet

```
TcComponentId GNCommon::NNotification::TcIdentifier::voCompDet
```

Bits 16 - 23 : Detecting Component

Definition at line 21 of file Identifier.h.

#### 6.10.4.2 voCompGen

```
TcComponentId GNCommon::NNotification::TcIdentifier::voCompGen
```

Bits 24 - 31 : Generating Component

Definition at line 22 of file Identifier.h.

#### 6.10.4.3 voGroup

```
TcGroupId GNCommon::NNotification::TcIdentifier::voGroup
```

Bits 8 - 15 : Group

Definition at line 20 of file Identifier.h.

#### 6.10.4.4 vucIndex

```
Tu8 GNCommon::NNotification::TcIdentifier::vucIndex
```

Bits 0 - 7 : Index

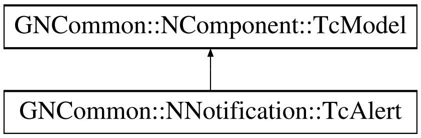
Definition at line 19 of file Identifier.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Identifier.h
- C:/Projects/BergermeisterHome/Software/Common/src/Notification/Identifier.cpp

## 6.11 GNCommon::NComponent::TcModel Class Reference

Inheritance diagram for GNCommon::NComponent::TcModel:


GNCommon::NComponent::TcModel
GNCommon::NNotification::TcAlert

### Public Attributes

- GTcEvent< xuiMaxListeners > **VoEvent**

### Static Protected Attributes

- static const Tu32 **xuiMaxListeners** = 16

### 6.11.1 Detailed Description

Definition at line 10 of file Model.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Component/Model.h
- C:/Projects/BergermeisterHome/Software/Common/src/Component/Model.cpp

## 6.12 GNCommon::NNotification::TcStatus Class Reference

```
#include <Status.h>
```

### Public Member Functions

- TcStatus (void)

  *Default Constructor.*
- TcStatus (const TcStatus &aorStatus)

  *Copy Constructor.*
- ∼TcStatus (void)

  *Destructor.*
- TcStatus & operator= (const TcStatus &aorStatus)

  *Assignment Operator.*
- Tb8 operator== (const TcStatus &aorStatus)

  *Equality Operator.*
- Tb8 operator!= (const TcStatus &aorStatus)

  *Inequality Operator.*

### Public Attributes

- Tu8 vucActive: 1
- Tu8 vucAcknowledged: 1
- Tu8 vucCleared: 1
- Tu8 vucTrigger: 1
- Tu8 vucSpare1: 4
- TcCriticality voCriticality
- Tu8 vucSpare2
- Tu8 vucChildren

### 6.12.1 Detailed Description

Alert Status

Definition at line 15 of file Status.h.

### 6.12.2 Constructor & Destructor Documentation

#### 6.12.2.1 TcStatus() [1/2]

```
TcStatus::TcStatus (
             void  )
```

Default Constructor.

Initializes the internal members to default values.

**Returns**

None

**Formal Parameters**

```
None
```

**Local Symbols**

```
None
```

Definition at line 25 of file Status.cpp.

#### 6.12.2.2 TcStatus() [2/2]

```
TcStatus::TcStatus (
             const TcStatus & aorStatus )
```

Copy Constructor.

Copies the internal members of the given Status to this Status via the assignment operator.

**Returns**

None

**Formal Parameters**

```
[ in ]  aorStatus    Status object reference to be copied
```

**Local Symbols**

```
None
```

Definition at line 50 of file Status.cpp.

#### 6.12.2.3 ∼TcStatus()

```
TcStatus::∼TcStatus (
             void  )
```

Destructor.

Nothing to destruct

**Returns**

None

**Formal Parameters**

```
None
```

**Local Symbols**

```
None
```

Definition at line 68 of file Status.cpp.

### 6.12.3 Member Function Documentation

#### 6.12.3.1 operator"!=()

```
Tb8 TcStatus::operator!= (
             const TcStatus & aorStatus )
```

Inequality Operator.

Compares this Status to the given Status and returns if they are not equal.

**Returns**

True if any of this Status's internal members are not equal to the given Status's internal members

**Formal Parameters**

```
[ in ]  aorStatus    Status object reference to be compared
```

**Local Symbols**

```
None
```

Definition at line 146 of file Status.cpp.

#### 6.12.3.2 operator=()

```
TcStatus & TcStatus::operator= (
              const TcStatus & aorStatus )
```

Assignment Operator.

Copies the internal members of the given Status

**Returns**

Status object reference to this Status

**Formal Parameters**

```
[ in ]  aorStatus    Status object reference to be copied
```

**Local Symbols**

```
None
```

Definition at line 87 of file Status.cpp.

#### 6.12.3.3 operator==()

```
Tb8 TcStatus::operator== (
              const TcStatus & aorStatus )
```

Equality Operator.

Compares this Status to the given Status and returns if they are equal.

**Returns**

True if all of this Status's internal members equal the given Status's internal members

**Formal Parameters**

```
[ in ]  aorStatus    Status object reference to be compared
```

**Local Symbols**

```
kbEqual    Flag indicating if the Statuses are equal
```

Definition at line 114 of file Status.cpp.

### 6.12.4 Member Data Documentation

#### 6.12.4.1 voCriticality

`TcCriticality GNCommon::NNotification::TcStatus::voCriticality`

Bits 8 - 15 : Alert Level

Definition at line 23 of file Status.h.

#### 6.12.4.2 vucAcknowledged

`Tu8 GNCommon::NNotification::TcStatus::vucAcknowledged`

Bit 1 : Acknowledged Flag

Definition at line 19 of file Status.h.

#### 6.12.4.3 vucActive

`Tu8 GNCommon::NNotification::TcStatus::vucActive`

Bit 0 : Active Flag

Definition at line 18 of file Status.h.

#### 6.12.4.4 vucChildren

`Tu8 GNCommon::NNotification::TcStatus::vucChildren`

Bits 24 - 31 : Number of Child Alerts

Definition at line 25 of file Status.h.

#### 6.12.4.5 vucCleared

`Tu8 GNCommon::NNotification::TcStatus::vucCleared`

Bit 2 : Cleared Flag

Definition at line 20 of file Status.h.

#### 6.12.4.6 vucSpare1

```
Tu8 GNCommon::NNotification::TcStatus::vucSpare1
```

Bits 4 - 7 : Spare Flags

Definition at line 22 of file Status.h.

#### 6.12.4.7 vucSpare2

```
Tu8 GNCommon::NNotification::TcStatus::vucSpare2
```

Bits 16 - 23 : Spare

Definition at line 24 of file Status.h.

#### 6.12.4.8 vucTrigger

```
Tu8 GNCommon::NNotification::TcStatus::vucTrigger
```

Bit 3 : Trigger Available Flag

Definition at line 21 of file Status.h.

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

- C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Status.h
- C:/Projects/BergermeisterHome/Software/Common/src/Notification/Status.cpp

# Chapter 7

# File Documentation

## 7.1 C:/Projects/BergermeisterHome/Software/Common/inc/DataAuthentication/CRC32.h File Reference

Package interface for the CRC32 class.

```
#include <Types.h>
```

### Classes

- class GNCommon::NDataAuthentication::TcCRC32

### Namespaces

- GNCommon

  *Namespace containing Common components and infrastrucutre.*
- GNCommon::NDataAuthentication

  *Namespace containing Data Authentication and Validity Checking utilities.*

### 7.1.1 Detailed Description

Package interface for the CRC32 class.

Definition in file CRC32.h.

## 7.2 CRC32.h

```

#pragma once

#include <Types.h>

namespace GNCommon
{
   namespace NDataAuthentication
   {
      class TcCRC32
      {
      protected:     // Protected Attributes
         static const Tu32 xuiDefaultSeed = 0xFFFFFFFF;
         static const Tu32 xuiTableCRC    = 0x6FCF9E13;
         static const Tu32 xuiShift8      = 8;
         static const Tu32 xuiMaskByte    = 0x000000FF;
         static const Tu32 xuiMaskAlign   = 0x00000003;
         static const Tu32 xuiTableSize   = 256;
         static const Tu32 xuiTable[ xuiTableSize ];

      public:        // Public Methods
         TcCRC32( void );

         Tu32 MGet( const Tu8* aucpBuffer, const Tu32 auiBytes, const
Tu32 auiSeed = xuiDefaultSeed ) const;

         Tb8 MVerify( void ) const;

      protected:     // Protected Methods
         Tu32 mGetLE( const Tu8* aucpBuffer, const Tu32 auiBytes, const
Tu32 auiSeed = xuiDefaultSeed ) const;
         Tu32 mGetBE( const Tu8* aucpBuffer, const Tu32 auiBytes, const
Tu32 auiSeed = xuiDefaultSeed ) const;
      };
   }
}
```

## 7.3 C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Alert.h File Reference

Package interface for the Alert Class.

```
#include <Types.h>
#include <Component\Model.h>
#include <Notification\ComponentId.h>
#include <Notification\Identifier.h>
#include <Notification\Status.h>
```

### Classes

- class GNCommon::NNotification::TcAlert

### Namespaces

- GNCommon

  *Namespace containing Common components and infrastrucutre.*
- GNCommon::NNotification

  *Namespace containing system Alerts.*

### 7.3.1 Detailed Description

Package interface for the Alert Class.

Definition in file Alert.h.

## 7.4 Alert.h

```

#pragma once

#include <Types.h>
#include <Component\Model.h>
#include <Notification\ComponentId.h>
#include <Notification\Identifier.h>
#include <Notification\Status.h>

namespace GNCommon
{
   namespace NNotification
   {
      class TcAlert : public GNCommon::NComponent::TcModel
      {
      public:        // Public Type Definitions
         static const Tu32 XuiSizeOfIdentifier = sizeof(
TcIdentifier );
         static const Tu32 XuiSizeOfStatus     = sizeof(
TcStatus     );
      protected:     // Protected Attributes
         TcIdentifier voID;
         TcStatus     voStatus;
         Tu64         vulTimestamp;
         Tu64         vulData;
      public:        // Public Methods
         TcAlert ( void );
         ~TcAlert( void );
         TcAlert ( const TcAlert& aorAlert );
         TcAlert& operator=( const TcAlert& aorAlert );

         void MSetIdentifier( const TcIdentifier& aorID        );
         void MSetStatus    ( const TcStatus&     aorStatus    );
         void MSetTimestamp ( const Tu64          aulTimestamp );
         void MSetData      ( const Tu64          aulData      );

         TcIdentifier MGetIdentifier( void ) const;
         TcStatus     MGetStatus    ( void ) const;
         Tu64         MGetTimestamp ( void ) const;
         Tu64         MGetData      ( void ) const;
      };
   }
}

```

## 7.5 C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ComponentId.h File Reference

### Namespaces

- GNCommon

  *Namespace containing Common components and infrastrucutre.*
- GNCommon::NNotification

  *Namespace containing system Alerts.*

### Enumerations

- enum GNCommon::NNotification::TcComponentId : Tu8 { GNCommon::NNotification::TcComponentId::XeNone = 0, GNCommon::NNotification::TcComponentId::XeServer = 1, GNCommon::NNotification::TcComponentId::XeSensor = 2 }

  *Enumeration of Alert Component Identifiers.*

## 7.6 ComponentId.h

```

#pragma once

namespace GNCommon
{
   namespace NNotification
   {
      enum class TcComponentId : Tu8
      {
         XeNone   = 0,
         XeServer = 1,
         XeSensor = 2
      };
   }
}

```

## 7.7 C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Criticality.h File Reference

Package interface for the Alert Criticality Enumeration.

### Namespaces

- GNCommon

  *Namespace containing Common components and infrastrucutre.*
- GNCommon::NNotification

  *Namespace containing system Alerts.*

### Enumerations

- enum GNCommon::NNotification::TcCriticality : Tu8 { GNCommon::NNotification::TcCriticality::XeNone = 0, GNCommon::NNotification:: = 1, GNCommon::NNotification::TcCriticality::XeWarning = 2, GNCommon::NNotification::TcCriticality::XeAlarm = 3 }

### 7.7.1 Detailed Description

Package interface for the Alert Criticality Enumeration.

Definition in file Criticality.h.

## 7.8 Criticality.h

```
#pragma once

namespace GNCommon
{
   namespace NNotification
   {
      enum class TcCriticality : Tu8
      {
         XeNone    = 0,
         XeNotice  = 1,
         XeWarning = 2,
         XeAlarm   = 3
      };
   }
}
```

## 7.9 C:/Projects/BergermeisterHome/Software/Common/inc/Notification/EventId.h File Reference

Package interface for the Event Identifier Enumeration.

### Namespaces

- GNCommon

  *Namespace containing Common components and infrastrucutre.*
- GNCommon::NNotification

  *Namespace containing system Alerts.*

### Enumerations

- enum GNCommon::NNotification::TcEventId : Tu32 { **XeIdentifier**, **XeStatus**, **XeTimestamp**, **XeData** }

### 7.9.1 Detailed Description

Package interface for the Event Identifier Enumeration.

Definition in file EventId.h.

## 7.10 EventId.h

```
#pragma once

namespace GNCommon
{
   namespace NNotification
   {
      enum class TcEventId : Tu32
      {
         XeIdentifier,
         XeStatus,
         XeTimestamp,
         XeData
      };
   }
}
```

## 7.11 C:/Projects/BergermeisterHome/Software/Common/inc/Notification/GroupId.h File Reference

Package interface for the Group Identifier Enumeration.

### Namespaces

- GNCommon

  *Namespace containing Common components and infrastrucutre.*
- GNCommon::NNotification

  *Namespace containing system Alerts.*

### Enumerations

- enum GNCommon::NNotification::TcGroupId : Tu8 { GNCommon::NNotification::TcGroupId::XeNone = 0, GNCommon::NNotification::Tc
  = 1 }

### 7.11.1 Detailed Description

Package interface for the Group Identifier Enumeration.

Definition in file GroupId.h.

## 7.12 GroupId.h

```

#pragma once

namespace GNCommon
{
   namespace NNotification
   {
      enum class TcGroupId : Tu8
      {
         XeNone    = 0,
         XeNetwork = 1
      };
   }
}
```

## 7.13 C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Identifier.h File Reference

Package interface for the Alert Identifier.

```
#include <Types.h>
#include <Notification\GroupId.h>
#include <Notification\ComponentId.h>
```

### Classes

- class GNCommon::NNotification::TcIdentifier

### Namespaces

- GNCommon

  *Namespace containing Common components and infrastrucutre.*
- GNCommon::NNotification

  *Namespace containing system Alerts.*

### 7.13.1 Detailed Description

Package interface for the Alert Identifier.

Definition in file Identifier.h.

## 7.14 Identifier.h

```

#pragma once

#include <Types.h>
#include <Notification\GroupId.h>
#include <Notification\ComponentId.h>

namespace GNCommon
{
   namespace NNotification
   {
      class TcIdentifier
      {
      public:        // Public Attributes
         Tu8           vucIndex;
         TcGroupId     voGroup;
         TcComponentId voCompDet;
         TcComponentId voCompGen;
      public:        // Public Methods
         TcIdentifier ( void );
         TcIdentifier ( const TcIdentifier& aorIdentifier );
         ~TcIdentifier( void );
         TcIdentifier& operator=( const TcIdentifier& aorIdentifier );

         Tb8 operator==( const TcIdentifier& aorIdentifier );
         Tb8 operator!=( const TcIdentifier& aorIdentifier );
      };
   }
}
```

## 7.15 C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Status.h File Reference

Package interface for the Alert Status.

```
#include <Types.h>
#include <Notification\Criticality.h>
```

### Classes

- class GNCommon::NNotification::TcStatus

### Namespaces

- GNCommon

  *Namespace containing Common components and infrastrucutre.*
- GNCommon::NNotification

  *Namespace containing system Alerts.*

### 7.15.1 Detailed Description

Package interface for the Alert Status.

Definition in file Status.h.

## 7.16 Status.h

```

#pragma once

#include <Types.h>
#include <Notification\Criticality.h>

namespace GNCommon
{
   namespace NNotification
   {
      class TcStatus
      {
      public:        // Public Attributes
         Tu8           vucActive       : 1;
         Tu8           vucAcknowledged : 1;
         Tu8           vucCleared      : 1;
         Tu8           vucTrigger      : 1;
         Tu8           vucSpare1       : 4;
         TcCriticality voCriticality;
         Tu8           vucSpare2;
         Tu8           vucChildren;
      public:        // Public Methods
         TcStatus ( void );
         TcStatus ( const TcStatus& aorStatus );
         ~TcStatus( void );
         TcStatus& operator=( const TcStatus& aorStatus );

         Tb8 operator==( const TcStatus& aorStatus );
         Tb8 operator!=( const TcStatus& aorStatus );
      };
   }
}
```

## 7.17 C:/Projects/BergermeisterHome/Software/Common/inc/StopWatch.h File Reference

### Classes

- class GNCommon::GTcStopWatch

### Namespaces

- GNCommon

    *Namespace containing Common components and infrastrucutre.*

### 7.17.1 Detailed Description

Package interface for the Stop Watch class interface

Definition in file StopWatch.h.

## 7.18 StopWatch.h

```

#pragma once

namespace GNCommon
{
   class GTcStopWatch
   {
   protected:     // Protected Attributes
      static const Ti64 xlMaxQuadPart = 9223372036854775807;
      static const Tu64 xulTimeBase   = 1000000LL;
      static Tu64 vulFrequency;

      Ti64 vlStart;
      Ti64 vlEnd;
      Tb8  vbRunning;

   public:
      GTcStopWatch( void );
      GTcStopWatch( const GTcStopWatch& aorStopWatch );
      ~GTcStopWatch( void );

      GTcStopWatch& operator=( const GTcStopWatch& aorStopWatch );

      void  MStart  ( void );
      Tu64 MStop   ( void );
      Tu64 MElapsed( void );
   };

}

```

## 7.19 C:/Projects/BergermeisterHome/Software/Common/inc/Types.h File Reference

Commnon Framework namespace, type definitions, and coding style guide Defines the common namespace (GNCommon), common primitive types, and provides the style guide to be used.

### Namespaces

- GNCommon

    *Namespace containing Common components and infrastrucutre.*

### Typedefs

- typedef bool GNCommon::Tb8
- typedef char GNCommon::Tc8
- typedef signed char GNCommon::Ti8
- typedef unsigned char GNCommon::Tu8
- typedef signed short GNCommon::Ti16
- typedef unsigned short GNCommon::Tu16
- typedef signed long GNCommon::Ti32
- typedef unsigned long GNCommon::Tu32
- typedef signed long long GNCommon::Ti64
- typedef unsigned long long GNCommon::Tu64
- typedef float GNCommon::Tf32
- typedef double GNCommon::Tf64

### 7.19.1 Detailed Description

Commnon Framework namespace, type definitions, and coding style guide Defines the common namespace (GNCommon), common primitive types, and provides the style guide to be used.

```
* Style Guide
*
*  (S)cope:
*                 | Priv  |  Pub  | Global |
*                 |-------|-------|--------|
* Class Variable |   v   |   V   |  ----  |
* Stack Variable |   k   |  ---  |  ----  |
*       Argument |   a   |  ---  |  ----  |
*        Typedef |   t   |   T   |   GT   |
*   Constant ROM |   x   |   X   |   GX   |
*         Method |   m   |   M   |   GM   |
*
*  (T)ype:
*           | Prefix |
*           |--------|
*      Tb8 |    b   |
*      Tc8 |    c   |
*      Ti8 |    c   |
*      Tu8 |   uc   |
*     Ti16 |    s   |
*     Tu16 |   us   |
*     Ti32 |    i   |
*     Tu32 |   ui   |
*     Ti64 |    l   |
*     Tu64 |   ul   |
*     Tf32 |    f   |
*     Tf64 |    d   |
*
*  (O)perator:
*               | Prefix |
*               |--------|
*     pointer   |    p   |
*     reference |    r   |
*
*  Naming Convention:
*     STOCamelCaseName
*     GMFunctionGlobal
*
```

Definition in file Types.h.

## 7.20 Types.h

```

#pragma once

namespace GNCommon
{

   typedef                bool  Tb8;
   typedef                char  Tc8;
   typedef signed         char  Ti8;
   typedef unsigned       char  Tu8;
   typedef signed        short Ti16;
   typedef unsigned      short Tu16;
   typedef signed         long Ti32;
   typedef unsigned       long Tu32;
   typedef signed    long long Ti64;
   typedef unsigned  long long Tu64;
   typedef               float Tf32;
   typedef              double Tf64;
   static const Tu32 XuiSizeOfTb8  = sizeof( Tb8 );
   static const Tu32 XuiSizeOfTi8  = sizeof( Ti8 );
   static const Tu32 XuiSizeOfTu8  = sizeof( Tu8 );
   static const Tu32 XuiSizeOfTi16 = sizeof( Ti16 );
   static const Tu32 XuiSizeOfTu16 = sizeof( Tu16 );
   static const Tu32 XuiSizeOfTi32 = sizeof( Ti32 );
   static const Tu32 XuiSizeOfTu32 = sizeof( Tu32 );
   static const Tu32 XuiSizeOfTi64 = sizeof( Ti64 );
   static const Tu32 XuiSizeOfTu64 = sizeof( Tu64 );
   static const Tu32 XuiSizeOfTf32 = sizeof( Tu64 );
   static const Tu32 XuiSizeOfTf64 = sizeof( Tu64 );
}
```

## 7.21 C:/Projects/BergermeisterHome/Software/Common/src/Notification/Identifier.cpp File Reference

```
#include <Types.h>
#include <Notification\GroupId.h>
#include <Notification\ComponentId.h>
#include <Notification\Identifier.h>
```

### 7.21.1 Detailed Description

Package implementation for the Alert Identifier

Definition in file Identifier.cpp.

## 7.22 Identifier.cpp

```

#include <Types.h>
#include <Notification\GroupId.h>
#include <Notification\ComponentId.h>
#include <Notification\Identifier.h>

using namespace GNCommon;
using namespace GNCommon::NNotification;

TcIdentifier::TcIdentifier( void )
{
   this->vucIndex   = 0;
   this->voGroup    = TcGroupId::XeNone;
   this->voCompDet  = TcComponentId::XeNone;
   this->voCompGen = TcComponentId::XeNone;
}

TcIdentifier::TcIdentifier( const TcIdentifier& aorIdentifier )
{
   *this = aorIdentifier;
}

TcIdentifier::~TcIdentifier( void )
{
   // Nothing to Destruct
}

TcIdentifier& TcIdentifier::operator=( const TcIdentifier& aorIdentifier )
{
   this->vucIndex  = aorIdentifier.vucIndex;
   this->voGroup   = aorIdentifier.voGroup;
   this->voCompDet = aorIdentifier.voCompDet;
   this->voCompGen = aorIdentifier.voCompGen;

   return( *this );
}

Tb8 TcIdentifier::operator==( const TcIdentifier& aorIdentifier )
{
   Tb8 kbEqual = true; // Assume Equal

   if ( ( this->vucIndex  != aorIdentifier.vucIndex  ) ||
        ( this->voGroup   != aorIdentifier.voGroup   ) ||
        ( this->voCompDet != aorIdentifier.voCompDet ) ||
        ( this->voCompGen != aorIdentifier.voCompGen ) )
   {
      kbEqual = false;
   }

   return( kbEqual );
}

Tb8 TcIdentifier::operator!=( const TcIdentifier& aorIdentifier )
{
   return( !( *this == aorIdentifier ) );
}

```

## 7.23 C:/Projects/BergermeisterHome/Software/Common/src/Notification/Status.cpp File Reference

Package implementation for the Alert Status.

```
#include <Types.h>
#include <Notification\Criticality.h>
#include <Notification\Status.h>
```

### 7.23.1 Detailed Description

Package implementation for the Alert Status.

Definition in file Status.cpp.

## 7.24 Status.cpp

```

#include <Types.h>
#include <Notification\Criticality.h>
#include <Notification\Status.h>
```

```
using namespace GNCommon;
using namespace GNCommon::NNotification;

TcStatus::TcStatus( void )
{
   this->vucActive       = false;
   this->vucAcknowledged = false;
   this->vucCleared      = false;
   this->vucTrigger      = false;
   this->vucSpare1       = 0;
   this->voCriticality   = TcCriticality::XeNone;
   this->vucSpare2       = 0;
   this->vucChildren     = 0;
}

TcStatus::TcStatus( const TcStatus& aorStatus )
{
   *this = aorStatus;
}

TcStatus::~TcStatus( void )
{
   // Nothing to destruct
}


TcStatus& TcStatus::operator=( const TcStatus& aorStatus )
{
   this->vucActive       = aorStatus.vucActive;
   this->vucAcknowledged = aorStatus.vucAcknowledged;
   this->vucCleared      = aorStatus.vucCleared;
   this->vucTrigger      = aorStatus.vucTrigger;
   this->vucSpare1       = aorStatus.vucSpare1;
   this->voCriticality   = aorStatus.voCriticality;
   this->vucSpare2       = aorStatus.vucSpare2;
   this->vucChildren     = aorStatus.vucChildren;

   return( *this );
}

Tb8 TcStatus::operator==( const TcStatus& aorStatus )
{
   Tb8 kbEqual = true; // Assume Equal

   if ( ( this->vucActive       != aorStatus.vucActive       ) ||
        ( this->vucAcknowledged != aorStatus.vucAcknowledged ) ||
        ( this->vucCleared      != aorStatus.vucCleared      ) ||
        ( this->vucTrigger      != aorStatus.vucTrigger      ) ||
        ( this->vucSpare1       != aorStatus.vucSpare1       ) ||
        ( this->voCriticality   != aorStatus.voCriticality   ) ||
        ( this->vucSpare2       != aorStatus.vucSpare2       ) ||
        ( this->vucChildren     != aorStatus.vucChildren     ) )
   {
      kbEqual = false;
   }

   return( kbEqual );
}

Tb8 TcStatus::operator!=( const TcStatus& aorStatus )
{
   return( !( *this == aorStatus ) );
}

```

# Index