

Bergermeister Home Automation

Source Code Documentation (SCD)

Contents

1	Namespace Index	7
1.1	Namespace List	7
2	Hierarchical Index	8
2.1	Class Hierarchy	8
3	Class Index	9
3.1	Class List	9
4	File Index	10
4.1	File List	10
5	Namespace Documentation	11
5.1	GNCommon Namespace Reference	11
5.1.1	Detailed Description	11
5.1.2	Typedef Documentation	12
5.1.2.1	Tb8	12
5.1.2.2	Tc8	12
5.1.2.3	Tf32	12
5.1.2.4	Tf64	12
5.1.2.5	Ti16	13
5.1.2.6	Ti32	13
5.1.2.7	Ti64	13
5.1.2.8	Ti8	13
5.1.2.9	Tu16	13

5.1.2.10	Tu32	14
5.1.2.11	Tu64	14
5.1.2.12	Tu8	14
5.2	GNCommon::NDataAuthentication Namespace Reference	14
5.2.1	Detailed Description	14
5.3	GNCommon::NNotification Namespace Reference	14
5.3.1	Detailed Description	15
5.3.2	Enumeration Type Documentation	15
5.3.2.1	TcComponentId	15
5.3.2.2	TcCriticality	15
5.3.2.3	TcEventId	16
5.3.2.4	TcGroupId	16
6	Class Documentation	17
6.1	GNCommon::NComponent::GTcEvent< auIMaxListeners > Class Template Reference	17
6.1.1	Detailed Description	17
6.2	GNCommon::NContainers::GTcLinkedList< GTcType > Class Template Reference	17
6.2.1	Detailed Description	18
6.3	GNCommon::NContainers::GTcList< GTcType > Class Template Reference	18
6.3.1	Detailed Description	19
6.4	GNCommon::NComponent::GTcListener Class Reference	19
6.4.1	Detailed Description	19
6.5	GNCommon::NContainers::GTcListNode< GTcType > Class Template Reference	19
6.5.1	Detailed Description	20
6.6	GNCommon::NContainers::GTcQueue< GTcType > Class Template Reference	20
6.6.1	Detailed Description	20
6.7	GNCommon::GTcStopWatch Class Reference	20
6.7.1	Detailed Description	21
6.7.2	Member Data Documentation	21
6.7.2.1	xulTimeBase	21

6.8	GNCommon::NNotification::TcAlert Class Reference	21
6.8.1	Detailed Description	22
6.8.2	Constructor & Destructor Documentation	22
6.8.2.1	TcAlert() [1/2]	22
6.8.2.2	~TcAlert()	23
6.8.2.3	TcAlert() [2/2]	23
6.8.3	Member Function Documentation	23
6.8.3.1	MGetData()	23
6.8.3.2	MGetIdentifier()	24
6.8.3.3	MGetStatus()	24
6.8.3.4	MGetTimestamp()	24
6.8.3.5	MSetData()	24
6.8.3.6	MSetIdentifier()	25
6.8.3.7	MSetStatus()	25
6.8.3.8	MSetTimestamp()	25
6.8.3.9	operator=()	26
6.8.4	Member Data Documentation	26
6.8.4.1	voID	26
6.8.4.2	voStatus	26
6.8.4.3	vulData	27
6.8.4.4	vulTimestamp	27
6.8.4.5	XuiSizeOfIdentifier	27
6.8.4.6	XuiSizeOfStatus	27
6.9	GNCommon::NDataAuthentication::TcCRC32 Class Reference	27
6.9.1	Detailed Description	28
6.9.2	Member Data Documentation	28
6.9.2.1	xuiTable	28
6.10	GNCommon::NNotification::TcIdentifier Class Reference	28
6.10.1	Detailed Description	29
6.10.2	Constructor & Destructor Documentation	29

6.10.2.1	TcIdentifier() [1/2]	29
6.10.2.2	TcIdentifier() [2/2]	30
6.10.2.3	~TcIdentifier()	30
6.10.3	Member Function Documentation	30
6.10.3.1	operator"!=()	31
6.10.3.2	operator=()	31
6.10.3.3	operator==()	32
6.10.4	Member Data Documentation	32
6.10.4.1	voCompDet	32
6.10.4.2	voCompGen	32
6.10.4.3	voGroup	33
6.10.4.4	vucIndex	33
6.11	GNCommon::NComponent::TcModel Class Reference	33
6.11.1	Detailed Description	33
6.12	GNCommon::NNotification::TcStatus Class Reference	34
6.12.1	Detailed Description	34
6.12.2	Constructor & Destructor Documentation	34
6.12.2.1	TcStatus() [1/2]	35
6.12.2.2	TcStatus() [2/2]	35
6.12.2.3	~TcStatus()	36
6.12.3	Member Function Documentation	36
6.12.3.1	operator"!=()	36
6.12.3.2	operator=()	37
6.12.3.3	operator==()	37
6.12.4	Member Data Documentation	37
6.12.4.1	voCriticality	38
6.12.4.2	vucAcknowledged	38
6.12.4.3	vucActive	38
6.12.4.4	vucChildren	38
6.12.4.5	vucCleared	38
6.12.4.6	vucSpare1	39
6.12.4.7	vucSpare2	39
6.12.4.8	vucTrigger	39

7	File Documentation	40
7.1	C:/Projects/BergermeisterHome/Software/Common/inc/DataAuthentication/CRC32.h File Reference	40
7.1.1	Detailed Description	40
7.2	CRC32.h	41
7.3	C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Alert.h File Reference	41
7.3.1	Detailed Description	42
7.4	Alert.h	42
7.5	C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ComponentId.h File Reference	42
7.6	ComponentId.h	43
7.7	C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Criticality.h File Reference	43
7.7.1	Detailed Description	43
7.8	Criticality.h	44
7.9	C:/Projects/BergermeisterHome/Software/Common/inc/Notification/EventId.h File Reference	44
7.9.1	Detailed Description	44
7.10	EventId.h	44
7.11	C:/Projects/BergermeisterHome/Software/Common/inc/Notification/GroupId.h File Reference	45
7.11.1	Detailed Description	45
7.12	GroupId.h	45
7.13	C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Identifier.h File Reference	45
7.13.1	Detailed Description	46
7.14	Identifier.h	46
7.15	C:/Projects/BergermeisterHome/Software/Common/inc/Notification/Status.h File Reference	46
7.15.1	Detailed Description	47
7.16	Status.h	47
7.17	C:/Projects/BergermeisterHome/Software/Common/inc/StopWatch.h File Reference	47
7.17.1	Detailed Description	48
7.18	StopWatch.h	48
7.19	C:/Projects/BergermeisterHome/Software/Common/inc/Types.h File Reference	48
7.19.1	Detailed Description	49
7.20	Types.h	50
7.21	C:/Projects/BergermeisterHome/Software/Common/src/Notification/Identifier.cpp File Reference	50
7.21.1	Detailed Description	50
7.22	Identifier.cpp	50
7.23	C:/Projects/BergermeisterHome/Software/Common/src/Notification/Status.cpp File Reference	51
7.23.1	Detailed Description	51
7.24	Status.cpp	51

Index	53
--------------	-----------

Chapter 1

Namespace Index

1.1 Namespace List

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

[GNCommon](#)
Namespace containing Common components and infrastrucutre 11

[GNCommon::NDataAuthentication](#)
Namespace containing Data Authentication and Validity Checking utilities 14

[GNCommon::NNotification](#)
Namespace containing system Alerts 14

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

GNCommon::NComponent::GTcEvent< auIMaxListeners > 17

GNCommon::NComponent::GTcEvent< xuiMaxListeners > 17

GNCommon::NContainers::GTcLinkedList< GTcType > 17

GNCommon::NContainers::GTcLinkedList< GNCommon::NComponent::GTcListener > 17

GNCommon::NContainers::GTcList< GTcType > 18

GNCommon::NComponent::GTcListener 19

GNCommon::NContainers::GTcListNode< GTcType > 19

GNCommon::NContainers::GTcListNode< GNCommon::NComponent::GTcListener > 19

GNCommon::NContainers::GTcQueue< GTcType > 20

GNCommon::GTcStopWatch 20

GNCommon::NDataAuthentication::TcCRC32 27

GNCommon::NNotification::TcIdentifier 28

GNCommon::NComponent::TcModel 33

 GNCommon::NNotification::TcAlert 21

GNCommon::NNotification::TcStatus 34

Chapter 3

Class Index

3.1 Class List

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

GNCommon::NComponent::GTcEvent< auimaxListeners > 17

GNCommon::NContainers::GTcLinkedList< GTcType > 17

GNCommon::NContainers::GTcList< GTcType > 18

GNCommon::NComponent::GTcListener 19

GNCommon::NContainers::GTcListNode< GTcType > 19

GNCommon::NContainers::GTcQueue< GTcType > 20

GNCommon::GTcStopWatch 20

GNCommon::NNotification::TcAlert 21

GNCommon::NDataAuthentication::TcCRC32 27

GNCommon::NNotification::TcIdentifier 28

GNCommon::NComponent::TcModel 33

GNCommon::NNotification::TcStatus 34

Chapter 4

File Index

4.1 File List

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

C:/Projects/BergermeisterHome/Software/Common/inc/ Constants.h	??
C:/Projects/BergermeisterHome/Software/Common/inc/ StopWatch.h	47
C:/Projects/BergermeisterHome/Software/Common/inc/ Types.h	
Common Framework namespace, type definitions, and coding style guide Defines the common namespace (GNCommon), common primitive types, and provides the style guide to be used	48
C:/Projects/BergermeisterHome/Software/Common/inc/Component/ Event.h	??
C:/Projects/BergermeisterHome/Software/Common/inc/Component/ Listener.h	??
C:/Projects/BergermeisterHome/Software/Common/inc/Component/ Model.h	??
C:/Projects/BergermeisterHome/Software/Common/inc/Containers/ LinkedList.h	??
C:/Projects/BergermeisterHome/Software/Common/inc/Containers/ List.h	??
C:/Projects/BergermeisterHome/Software/Common/inc/Containers/ ListNode.h	??
C:/Projects/BergermeisterHome/Software/Common/inc/Containers/ Queue.h	??
C:/Projects/BergermeisterHome/Software/Common/inc/DataAuthentication/ CRC32.h	
Package interface for the CRC32 class	40
C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ Alert.h	
Package interface for the Alert Class	41
C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ ComponentId.h	42
C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ Criticality.h	
Package interface for the Alert Criticality Enumeration	43
C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ EventId.h	
Package interface for the Event Identifier Enumeration	44
C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ GroupId.h	
Package interface for the Group Identifier Enumeration	45
C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ Identifier.h	
Package interface for the Alert Identifier	45
C:/Projects/BergermeisterHome/Software/Common/inc/Notification/ Status.h	
Package interface for the Alert Status	46
C:/Projects/BergermeisterHome/Software/Common/src/ StopWatch.cpp	??
C:/Projects/BergermeisterHome/Software/Common/src/Component/ Listener.cpp	??
C:/Projects/BergermeisterHome/Software/Common/src/Component/ Model.cpp	??
C:/Projects/BergermeisterHome/Software/Common/src/DataAuthentication/ CRC32.cpp	??
C:/Projects/BergermeisterHome/Software/Common/src/Notification/ Alert.cpp	??
C:/Projects/BergermeisterHome/Software/Common/src/Notification/ Identifier.cpp	50
C:/Projects/BergermeisterHome/Software/Common/src/Notification/ Status.cpp	
Package implementation for the Alert Status	51

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](#) auIdentifier)
- [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](#) auSize)
- [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](#) **MRemoveAt** (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](#).

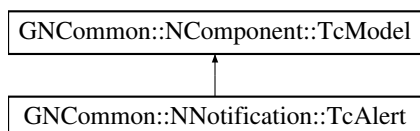
The documentation for this class was generated from the following files:

- 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:



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

<i>aorAlert</i>	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->vuTimestamp 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

<i>aulData</i>	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

<i>aorID</i>	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

<i>aorStatus</i>	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

<i>aulTimestamp</i>	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

<i>aorAlert</i>	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 vuclIndex](#)
- [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

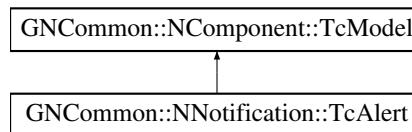
Definition at line 19 of file [Identifier.h](#).

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`:



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

```

00001
00005 #pragma once
00006
00007 #include <Types.h>
00008
00009 namespace GNCommon
00010 {
00012     namespace NDataAuthentication
00013     {
00014         class TcCRC32
00015         {
00016             protected:    // Protected Attributes
00017                 static const Tu32 xuiDefaultSeed = 0xFFFFFFFF;
00018                 static const Tu32 xuiTableCRC    = 0x6FCF9E13;
00019                 static const Tu32 xuiShift8      = 8;
00020                 static const Tu32 xuiMaskByte    = 0x000000FF;
00021                 static const Tu32 xuiMaskAlign   = 0x00000003;
00022                 static const Tu32 xuiTableSize   = 256;
00023                 static const Tu32 xuiTable[ xuiTableSize ];
00024
00025             public:        // Public Methods
00026                 TcCRC32( void );
00027
00028                 Tu32 MGet( const Tu8* aucpBuffer, const Tu32 auiBytes, const
00029 Tu32 auiSeed = xuiDefaultSeed ) const;
00030
00031                 Tb8 MVerify( void ) const;
00032
00033             protected:    // Protected Methods
00034                 Tu32 mGetLE( const Tu8* aucpBuffer, const Tu32 auiBytes, const
00035 Tu32 auiSeed = xuiDefaultSeed ) const;
00036                 Tu32 mGetBE( const Tu8* aucpBuffer, const Tu32 auiBytes, const
00037 Tu32 auiSeed = xuiDefaultSeed ) const;
00038             };
00039         };
00040     };
00041 }

```

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

```

00001
00005 #pragma once
00006
00007 #include <Types.h>
00008 #include <Component\Model.h>
00009 #include <Notification\ComponentId.h>
00010 #include <Notification\Identifier.h>
00011 #include <Notification\Status.h>
00012
00013 namespace GNCommon
00014 {
00016     namespace NNotification
00017     {
00019         class TcAlert : public GNCommon::NComponent::TcModel
00020         {
00021             public:          // Public Type Definitions
00022                 static const Tu32 XuiSizeOfIdentifier = sizeof(
00023 TcIdentifier );
00024                 static const Tu32 XuiSizeOfStatus    = sizeof(
00025 TcStatus
00026 );
00027             protected:     // Protected Attributes
00028                 TcIdentifier voID;
00029                 TcStatus    voStatus;
00030                 Tu64        vulTimestamp;
00031                 Tu64        vulData;
00032             public:         // Public Methods
00033                 TcAlert ( void );
00034                 ~TcAlert( void );
00035                 TcAlert ( const TcAlert& aorAlert );
00036                 TcAlert& operator=( const TcAlert& aorAlert );
00037
00038                 void MSetIdentifier( const TcIdentifier& aorID
00039 );
00040                 void MSetStatus    ( const TcStatus&      aorStatus
00041 );
00042                 void MSetTimestamp ( const Tu64          aulTimestamp );
00043                 void MSetData      ( const Tu64          aulData
00044 );
00045
00046                 TcIdentifier MGetIdentifier( void ) const;
00047                 TcStatus     MGetStatus    ( void ) const;
00048                 Tu64         MGetTimestamp ( void ) const;
00049                 Tu64         MGetData      ( void ) const;
00050         };
00051     }
00052 }
00053

```

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

```

00001
00005 #pragma once
00006
00007 namespace GNCommon
00008 {
00009     namespace NNotification
00010     {
00012         enum class TcComponentId : Tu8
00013         {
00014             XeNone    = 0,
00015             XeServer  = 1,
00016             XeSensor  = 2
00017         };
00018     }
00019 }
00020

```

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::TcCriticality::XeWarning](#) = 1, [GNCommon::NNotification::TcCriticality::XeAlarm](#) = 2, [GNCommon::NNotification::TcCriticality::XeCritical](#) = 3 }

7.7.1 Detailed Description

Package interface for the Alert Criticality Enumeration.

Definition in file [Criticality.h](#).

7.8 Criticality.h

```

00001
00005 #pragma once
00006
00007 namespace GNCommon
00008 {
00009     namespace NNotification
00010     {
00012         enum class TcCriticality : Tu8
00013         {
00014             XeNone      = 0,
00015             XeNotice    = 1,
00016             XeWarning   = 2,
00017             XeAlarm     = 3
00018         };
00019     }
00020 }
00021

```

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

```

00001
00005 #pragma once
00006
00007 namespace GNCommon
00008 {
00009     namespace NNotification
00010     {
00012         enum class TcEventId : Tu32
00013         {
00014             XeIdentifier,
00015             XeStatus,
00016             XeTimestamp,
00017             XeData
00018         };
00019     }
00020 }
00021

```

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::TcGroupId::XeNetwork](#) = 1 }

7.11.1 Detailed Description

Package interface for the Group Identifier Enumeration.

Definition in file [GroupId.h](#).

7.12 GroupId.h

```

00001
00005 #pragma once
00006
00007 namespace GNCommon
00008 {
00009     namespace NNotification
00010     {
00012         enum class TcGroupId : Tu8
00013         {
00014             XeNone      = 0,
00015             XeNetwork    = 1
00016         };
00017     }
00018 }
```

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

```

00001
00005 #pragma once
00006
00007 #include <Types.h>
00008 #include <Notification\GroupId.h>
00009 #include <Notification\ComponentId.h>
00010
00011 namespace GNCommon
00012 {
00013     namespace NNotification
00014     {
00016         class TcIdentifier
00017         {
00018             public:          // Public Attributes
00019                 Tu8         vucIndex;
00020                 TcGroupId   voGroup;
00021                 TcComponentId voCompDet;
00022                 TcComponentId voCompGen;
00024             public:        // Public Methods
00025                 TcIdentifier ( void );
00026                 TcIdentifier ( const TcIdentifier& aorIdentifier );
00027                 ~TcIdentifier( void );
00028                 TcIdentifier& operator=( const TcIdentifier& aorIdentifier );
00029
00030                 Tb8 operator==( const TcIdentifier& aorIdentifier );
00031                 Tb8 operator!=( const TcIdentifier& aorIdentifier );
00032         };
00033     }
00034 }
```

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

```

00001
00005 #pragma once
00006
00007 #include <Types.h>
00008 #include <Notification\Criticality.h>
00009
00010 namespace GNCommon
00011 {
00012     namespace NNotification
00013     {
00014         class TcStatus
00015         {
00016         public:          // Public Attributes
00017             Tu8          vucActive          : 1;
00018             Tu8          vucAcknowledged    : 1;
00019             Tu8          vucCleared        : 1;
00020             Tu8          vucTrigger        : 1;
00021             Tu8          vucSpare1        : 4;
00022             TcCriticality voCriticality;
00023             Tu8          vucSpare2;
00024             Tu8          vucChildren;
00025         public:          // Public Methods
00026             TcStatus ( void );
00027             TcStatus ( const TcStatus& aorStatus );
00028             ~TcStatus( void );
00029             TcStatus& operator=( const TcStatus& aorStatus );
00030
00031             Tb8 operator==( const TcStatus& aorStatus );
00032             Tb8 operator!=( const TcStatus& aorStatus );
00033         };
00034     }
00035 }
00036
00037 
```

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

```

00001
00006 #pragma once
00007
00008 namespace GNCommon
00009 {
00010     class GTcStopWatch
00011     {
00012     protected:    // Protected Attributes
00013         static const Ti64 xlMaxQuadPart = 9223372036854775807;
00014         static const Tu64 xulTimeBase  = 1000000LL;
00015         static Tu64 vulFrequency;
00016
00017         Ti64 vlStart;
00018         Ti64 vlEnd;
00019         Tb8 vbRunning;
00020
00021     public:
00022         GTcStopWatch( void );
00023         GTcStopWatch( const GTcStopWatch& aorStopWatch );
00024         ~GTcStopWatch( void );
00025
00026         GTcStopWatch& operator=( const GTcStopWatch& aorStopWatch );
00027
00028         void MStart ( void );
00029         Tu64 MStop ( void );
00030         Tu64 MElapsed( void );
00031     };
00032
00033 }
00034

```

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

Common 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

Common 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

```

00001
00048 #pragma once
00049
00051 namespace GNCommon
00052 {
00053
00054     typedef          bool   Tb8;
00055     typedef          char   Tc8;
00056     typedef signed    char   Ti8;
00057     typedef unsigned  char   Tu8;
00058     typedef signed    short  Ti16;
00059     typedef unsigned  short  Tu16;
00060     typedef signed    long   Ti32;
00061     typedef unsigned  long   Tu32;
00062     typedef signed    long long Ti64;
00063     typedef unsigned  long long Tu64;
00064     typedef          float  Tf32;
00065     typedef          double Tf64;
00067     static const Tu32 XuiSizeOfTb8  = sizeof( Tb8 );
00068     static const Tu32 XuiSizeOfTi8   = sizeof( Ti8 );
00069     static const Tu32 XuiSizeOfTu8   = sizeof( Tu8 );
00070     static const Tu32 XuiSizeOfTi16  = sizeof( Ti16 );
00071     static const Tu32 XuiSizeOfTu16  = sizeof( Tu16 );
00072     static const Tu32 XuiSizeOfTi32  = sizeof( Ti32 );
00073     static const Tu32 XuiSizeOfTu32  = sizeof( Tu32 );
00074     static const Tu32 XuiSizeOfTi64  = sizeof( Ti64 );
00075     static const Tu32 XuiSizeOfTu64  = sizeof( Tu64 );
00076     static const Tu32 XuiSizeOfTf32  = sizeof( Tu64 );
00077     static const Tu32 XuiSizeOfTf64  = sizeof( Tu64 );
00078 }

```

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

```

00001
00006 #include <Types.h>
00007 #include <Notification\GroupId.h>
00008 #include <Notification\ComponentId.h>
00009 #include <Notification\Identifier.h>
00010
00011 using namespace GNCommon;
00012 using namespace GNCommon::NNotification;
00013
00027 TcIdentifier::TcIdentifier( void )
00028 {
00029     this->vucIndex = 0;
00030     this->voGroup   = TcGroupId::XeNone;
00031     this->voCompDet = TcComponentId::XeNone;

```

```

00032     this->voCompGen = TcComponentId::XeNone;
00033 }
00034
00048 TcIdentifier::TcIdentifier( const TcIdentifier& aorIdentifier )
00049 {
00050     *this = aorIdentifier;
00051 }
00052
00066 TcIdentifier::~TcIdentifier( void )
00067 {
00068     // Nothing to Destruct
00069 }
00070
00084 TcIdentifier& TcIdentifier::operator=( const
TcIdentifier& aorIdentifier )
00085 {
00086     this->vucIndex = aorIdentifier.vucIndex;
00087     this->voGroup = aorIdentifier.voGroup;
00088     this->voCompDet = aorIdentifier.voCompDet;
00089     this->voCompGen = aorIdentifier.voCompGen;
00090
00091     return( *this );
00092 }
00093
00107 Tb8 TcIdentifier::operator==( const TcIdentifier& aorIdentifier )
00108 {
00109     Tb8 kbEqual = true; // Assume Equal
00110
00111     if ( ( this->vucIndex != aorIdentifier.vucIndex ) ||
00112         ( this->voGroup != aorIdentifier.voGroup ) ||
00113         ( this->voCompDet != aorIdentifier.voCompDet ) ||
00114         ( this->voCompGen != aorIdentifier.voCompGen ) )
00115     {
00116         kbEqual = false;
00117     }
00118
00119     return( kbEqual );
00120 }
00121
00135 Tb8 TcIdentifier::operator!=( const TcIdentifier& aorIdentifier )
00136 {
00137     return( !( *this == aorIdentifier ) );
00138 }
00139

```

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

```

00001
00005 #include <Types.h>
00006 #include <Notification\Criticality.h>
00007 #include <Notification\Status.h>

```

```

00008
00009 using namespace GNCommon;
00010 using namespace GNCommon::NNotification;
00011
00025 TcStatus::TcStatus( void )
00026 {
00027     this->vucActive           = false;
00028     this->vucAcknowledged     = false;
00029     this->vucCleared          = false;
00030     this->vucTrigger           = false;
00031     this->vucSpare1            = 0;
00032     this->voCriticality        = TcCriticality::XeNone;
00033     this->vucSpare2            = 0;
00034     this->vucChildren          = 0;
00035 }
00036
00050 TcStatus::TcStatus( const TcStatus& aorStatus )
00051 {
00052     *this = aorStatus;
00053 }
00054
00068 TcStatus::~~TcStatus( void )
00069 {
00070     // Nothing to destruct
00071 }
00072
00073
00087 TcStatus& TcStatus::operator=( const TcStatus& aorStatus )
00088 {
00089     this->vucActive           = aorStatus.vucActive;
00090     this->vucAcknowledged     = aorStatus.vucAcknowledged;
00091     this->vucCleared          = aorStatus.vucCleared;
00092     this->vucTrigger           = aorStatus.vucTrigger;
00093     this->vucSpare1            = aorStatus.vucSpare1;
00094     this->voCriticality        = aorStatus.voCriticality;
00095     this->vucSpare2            = aorStatus.vucSpare2;
00096     this->vucChildren          = aorStatus.vucChildren;
00097
00098     return( *this );
00099 }
00100
00114 Tb8 TcStatus::operator==( const TcStatus& aorStatus )
00115 {
00116     Tb8 kbEqual = true; // Assume Equal
00117
00118     if ( ( this->vucActive           != aorStatus.vucActive           ) ||
00119         ( this->vucAcknowledged     != aorStatus.vucAcknowledged     ) ||
00120         ( this->vucCleared          != aorStatus.vucCleared          ) ||
00121         ( this->vucTrigger           != aorStatus.vucTrigger           ) ||
00122         ( this->vucSpare1            != aorStatus.vucSpare1            ) ||
00123         ( this->voCriticality        != aorStatus.voCriticality        ) ||
00124         ( this->vucSpare2            != aorStatus.vucSpare2            ) ||
00125         ( this->vucChildren          != aorStatus.vucChildren          ) )
00126     {
00127         kbEqual = false;
00128     }
00129
00130     return( kbEqual );
00131 }
00132
00146 Tb8 TcStatus::operator!=( const TcStatus& aorStatus )
00147 {
00148     return( !( *this == aorStatus ) );
00149 }
00150

```

Index

- ~TcAlert
 - GNCommon::NNotification::TcAlert, [22](#)
- ~TcIdentifier
 - GNCommon::NNotification::TcIdentifier, [30](#)
- ~TcStatus
 - GNCommon::NNotification::TcStatus, [35](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - DataAuthentication/CRC32.h, [40](#), [41](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - Notification/Alert.h, [41](#), [42](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - Notification/ComponentId.h, [42](#), [43](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - Notification/Criticality.h, [43](#), [44](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - Notification/EventId.h, [44](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - Notification/GroupId.h, [45](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - Notification/Identifier.h, [45](#), [46](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - Notification/Status.h, [46](#), [47](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - StopWatch.h, [47](#), [48](#)
- C:/Projects/BergermeisterHome/Software/Common/inc/↵
 - Types.h, [48](#), [50](#)
- C:/Projects/BergermeisterHome/Software/Common/src/↵
 - Notification/Identifier.cpp, [50](#)
- C:/Projects/BergermeisterHome/Software/Common/src/↵
 - Notification/Status.cpp, [51](#)
- GNCommon, [11](#)
 - Tb8, [12](#)
 - Tc8, [12](#)
 - Tf32, [12](#)
 - Tf64, [12](#)
 - Ti16, [12](#)
 - Ti32, [13](#)
 - Ti64, [13](#)
 - Ti8, [13](#)
 - Tu16, [13](#)
 - Tu32, [13](#)
 - Tu64, [14](#)
 - Tu8, [14](#)
- GNCommon::GTcStopWatch, [20](#)
 - xulTimeBase, [21](#)
- GNCommon::NComponent::GTcEvent< auiMaxListeners >, [17](#)
- GNCommon::NComponent::GTcListener, [19](#)
- GNCommon::NComponent::TcModel, [33](#)
- GNCommon::NContainers::GTcLinkedList< GTcType >, [17](#)
- GNCommon::NContainers::GTcList< GTcType >, [18](#)
- GNCommon::NContainers::GTcListNode< GTcType >, [19](#)
- GNCommon::NContainers::GTcQueue< GTcType >, [20](#)
- GNCommon::NDataAuthentication, [14](#)
- GNCommon::NDataAuthentication::TcCRC32, [27](#)
 - xuiTable, [28](#)
- GNCommon::NNotification, [14](#)
 - TcComponentId, [15](#)
 - TcCriticality, [15](#)
 - TcEventId, [16](#)
 - TcGroupId, [16](#)
- GNCommon::NNotification::TcAlert, [21](#)
 - ~TcAlert, [22](#)
 - MGetData, [23](#)
 - MGetIdentifier, [23](#)
 - MGetStatus, [24](#)
 - MGetTimestamp, [24](#)
 - MSetData, [24](#)
 - MSetIdentifier, [25](#)
 - MSetStatus, [25](#)
 - MSetTimestamp, [25](#)
 - operator=, [26](#)
 - TcAlert, [22](#), [23](#)
 - voId, [26](#)
 - voStatus, [26](#)
 - vulData, [26](#)
 - vulTimestamp, [27](#)
 - XuiSizeOfIdentifier, [27](#)
 - XuiSizeOfStatus, [27](#)
- GNCommon::NNotification::TcIdentifier, [28](#)
 - ~TcIdentifier, [30](#)
 - operator!=, [30](#)
 - operator=, [31](#)
 - operator==, [31](#)
 - TcIdentifier, [29](#)
 - voCompDet, [32](#)
 - voCompGen, [32](#)
 - voGroup, [32](#)
 - vucIndex, [33](#)
- GNCommon::NNotification::TcStatus, [34](#)
 - ~TcStatus, [35](#)
 - operator!=, [36](#)
 - operator=, [36](#)
 - operator==, [37](#)
 - TcStatus, [34](#), [35](#)
 - voCriticality, [37](#)

vucAcknowledged, 38	
vucActive, 38	
vucChildren, 38	
vucCleared, 38	
vucSpare1, 38	
vucSpare2, 39	
vucTrigger, 39	
MGetData	
GNCommon::NNotification::TcAlert, 23	
MGetIdentifier	
GNCommon::NNotification::TcAlert, 23	
MGetStatus	
GNCommon::NNotification::TcAlert, 24	
MGetTimestamp	
GNCommon::NNotification::TcAlert, 24	
MSetData	
GNCommon::NNotification::TcAlert, 24	
MSetIdentifier	
GNCommon::NNotification::TcAlert, 25	
MSetStatus	
GNCommon::NNotification::TcAlert, 25	
MSetTimestamp	
GNCommon::NNotification::TcAlert, 25	
operator!=	
GNCommon::NNotification::TcIdentifier, 30	
GNCommon::NNotification::TcStatus, 36	
operator=	
GNCommon::NNotification::TcAlert, 26	
GNCommon::NNotification::TcIdentifier, 31	
GNCommon::NNotification::TcStatus, 36	
operator==	
GNCommon::NNotification::TcIdentifier, 31	
GNCommon::NNotification::TcStatus, 37	
Tb8	
GNCommon, 12	
Tc8	
GNCommon, 12	
TcAlert	
GNCommon::NNotification::TcAlert, 22, 23	
TcComponentId	
GNCommon::NNotification, 15	
TcCriticality	
GNCommon::NNotification, 15	
TcEventId	
GNCommon::NNotification, 16	
TcGroupId	
GNCommon::NNotification, 16	
TcIdentifier	
GNCommon::NNotification::TcIdentifier, 29	
TcStatus	
GNCommon::NNotification::TcStatus, 34, 35	
Tf32	
GNCommon, 12	
Tf64	
GNCommon, 12	
Ti16	
GNCommon, 12	
Ti32	
GNCommon, 13	
Ti64	
GNCommon, 13	
Ti8	
GNCommon, 13	
Tu16	
GNCommon, 13	
Tu32	
GNCommon, 13	
Tu64	
GNCommon, 14	
Tu8	
GNCommon, 14	
voCompDet	
GNCommon::NNotification::TcIdentifier, 32	
voCompGen	
GNCommon::NNotification::TcIdentifier, 32	
voCriticality	
GNCommon::NNotification::TcStatus, 37	
voGroup	
GNCommon::NNotification::TcIdentifier, 32	
void	
GNCommon::NNotification::TcAlert, 26	
voStatus	
GNCommon::NNotification::TcAlert, 26	
vucAcknowledged	
GNCommon::NNotification::TcStatus, 38	
vucActive	
GNCommon::NNotification::TcStatus, 38	
vucChildren	
GNCommon::NNotification::TcStatus, 38	
vucCleared	
GNCommon::NNotification::TcStatus, 38	
vucIndex	
GNCommon::NNotification::TcIdentifier, 33	
vucSpare1	
GNCommon::NNotification::TcStatus, 38	
vucSpare2	
GNCommon::NNotification::TcStatus, 39	
vucTrigger	
GNCommon::NNotification::TcStatus, 39	
vulData	
GNCommon::NNotification::TcAlert, 26	
vulTimestamp	
GNCommon::NNotification::TcAlert, 27	
XuiSizeOfIdentifier	
GNCommon::NNotification::TcAlert, 27	
XuiSizeOfStatus	
GNCommon::NNotification::TcAlert, 27	
xuiTable	
GNCommon::NDataAuthentication::TcCRC32, 28	
xulTimeBase	
GNCommon::GTcStopWatch, 21	