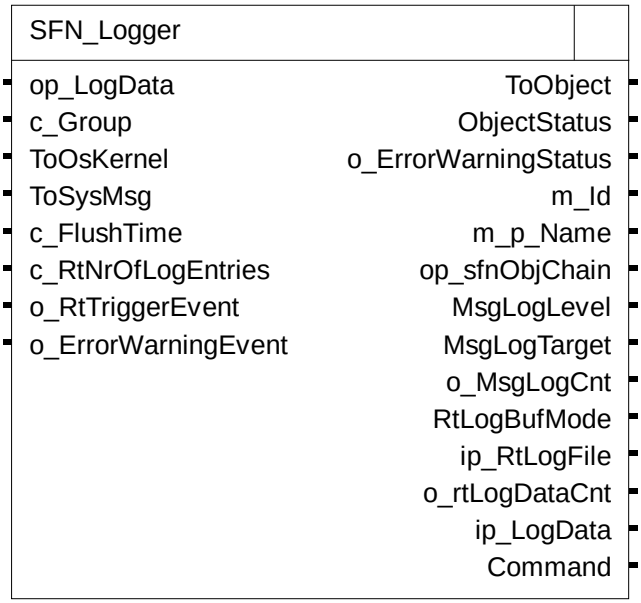
Name: SFN_Logger

Revision: 0.3

Ladder: false
Complex: true

Graphic Schema:



Comment: SF Logger
Central place to log messages and/or realtime data
received from other classes.
(see methodes SF_Base.WriteLogRtData() en SF_Base.WriteLogMessage())
This class supports
1) message logging
2) Realtime data logging
3) Create a chain of connected SFN_Base objects for logging
Both loggings can be activated simultanuously.
Message logging stores messages to either screen, file or Both screen and file.
Filename for message logging is fixed EVENT01.LOG (defined by Lasal OS).
Server p_sfnObjChain Points to start of chain of connected SFN_Base objects.
Server MsgLogLevel we can select the message priority levels which should be logged.
Server MsgLogTarget we can select the output to send the messages to.
Server o_MsgLogCnt Show nr of messages logged
Server RtLogMBufMode is used to select the buffer mode
- cyclic overwrite buffer if it is full
- once write until buffer is full
Server ip_RtLogFile a pointer to File name string for RealtIme data logging can be set.
Note: Filename for message logging is controled by Lasal OS.
Server o_RtLogDataCnt: Show nr of realtime data entries logged
Server ip_LogData is the interface to other class objects. Via this server this class receives the logdata from other classes.
Server command is used to send commands.
With methode SetTriggerRtLog() a trigger condition can be set (Stored in variable RtTriggerSetting).
Message logging logs messages in an OS buffer. If buffer is full the content is send to File.
Client flusTime is used to set the time to flush automtically messages to a file in case buffer is not full.
Realtime Data Logging logs data into an internal RAM buffer.
With Client c_RtNrOfLogEntries we can select the number of entries to be stored.
Note: Size of buffer may exceed the limit of 64K which is used by Lasal for class objects
With LogCommand LOGCMD_SENDFILE_RT_LOG the stored RtLogData can be stored in a File.

Clients: Name: op_LogData
Class: SFN_Base
DataType: gpt_LogData
Type: Data Channel
Required: false
Comment: Output for logging data to SF_Logger

Name: c_Group
Class: SFN_Base
DataType: UDINT
Type: Data Channel
Required: false
Comment: Group identification object is part of.

Name: ToOsKernel
Class: _OSKernel
DataType: DINT
Type: Object Channel
Required: false
Comment: To OS Kernel
(Automatically connected to OS)

Name: ToSysMsg
Class: _SysMsg
DataType: DINT
Type: Object Channel
Required: false
Comment: To OS System Message
(Automatically connected to OS)

Name: c_FlushTime
Class: SFN_Logger
DataType: UDINT
Type: Data Channel
Required: false
Comment: Time to flush logging messages (write buffer to file)

Unit: Miliseconds

Name: c_RtNrOfLogEntries
Class: SFN_Logger
DataType: UDINT
Type: Data Channel
Required: true
Comment: Nr of entries in Realtime LogData buffer

Name: o_RtTriggerEvent
Class: SFN_Logger
DataType: DINT
Type: Data Channel
Required: false
Comment: Trigger found for Realtime data logging
Generic interface is used
so if wanted each object can use
this event for debugging.

Name: o_ErrorWarningEvent
Class: SFN_Logger
DataType: UDINT
Type: Data Channel
Required: false
Comment: Event to indicate an Error or Warning occured.
Only pass events for
- Warning
- application error
- internal error
- critical error
Note:
Event will always send even logging is disabled.

Server: Name: ToObject
GUID: {9EED831D-1495-49CA-AD11-0E8CDA8941D2}
Class: SFN_Base
DataType: pVoid

Type:	Object Channel
Initialize:	false
WriteProtected:	true
Retentive:	false
Comment:	ToObject: Object channel server. Channel to reach the methodes of this object. In case this object should be Accessible by a pointer the variable ToObject is used to handover the address of this object. This server is never used for anything else!!!

Name:	ObjectStatus
GUID:	{A29E2DB3-B7AD-4363-A726-DB93BF808383}
Visualized:	false
DataType:	gt_ObjectStatus
Type:	Data Channel
Initialize:	false
WriteProtected:	false
Retentive:	false
Comment:	Generic object status interface. Bit 0 - 15 is reserved for generic Safan framework Bit 16 -31 can be used by the application See also type gt_ObjectStatus.

Name:	o_ErrorWarningStatus
GUID:	{71D405E1-73D5-4283-95DF-BAB38DCB9D0B}
Visualized:	false
DataType:	UDINT
Type:	Data Channel
Initialize:	false
WriteProtected:	true
Retentive:	false
Comment:	Actual active Error or Warning status. A lower level message will not overwite an active higher level message. for example a warning will not overwrite this server if it has already an active error Note: see also internal variable FirstError to find the first error detected since last error handling.

Name:	m_Id
GUID:	{AAB7581E-3B92-42E8-9287-D9BCBCF7E8ED}
Visualized:	false
DataType:	UDINT
Type:	Data Channel
Initialize:	true
WriteProtected:	false
Retentive:	false
Comment:	An identification number of this object. Free to use/define by derived classes and application.

Name:	m_p_Name
GUID:	{33AD7665-E336-4A23-BACF-BD56E31FDC73}
Visualized:	false
DataType:	pChar
Type:	Data Channel
Initialize:	false
WriteProtected:	true
Retentive:	false
Comment:	Name of this object. String is closed by an end of sting character (value 0) String is read only!

Name:	op_sfnObjChain
GUID:	{38436872-8F2E-4408-B2CD-79AAD14AEEC0}
Visualized:	false
DataType:	pVoid
Initialize:	false

WriteProtected:	true
Retentive:	false
Comment:	Points to chain of all SFN_Base objects registrated to the Logger. Can be used to go through all objects.

Name:	MsgLogLevel
GUID:	{C19A5892-EF6B-4E5D-BAA4-4ACDBC2F24C5}
Visualized:	false
DataType:	gt_LogLevel
Type:	Data Channel
Initialize:	true
WriteProtected:	false
Retentive:	false
Comment:	Configuration of Message loglevel. Only log messages of selected loglevel and higher levels. Note: LL_DEBUG_RT is reserved for realtime datalogging. Setting LL_DEBUG_RT will be changed into loglevel LL_DEBUG. Tip: To select temporary individual log levels for debug purpose, modify ActiveLogFlags by hand.

Name:	MsgLogTarget
GUID:	{9C544215-D5CB-4C09-AB67-750061B2C276}
Visualized:	false
DataType:	gt_LogTarget
Type:	Data Channel
Initialize:	true
WriteProtected:	false
Retentive:	false
Comment:	Configuration of target where to send log messages.

Name:	o_MsgLogCnt
GUID:	{8136FAB9-0812-4C81-8B98-EEE901B9B6B7}
Visualized:	false
DataType:	UDINT
Type:	Data Channel
Initialize:	false
WriteProtected:	true
Retentive:	false
Comment:	Counter of logged messages since last startup or reset cmd. To be sure message is stored in the log file execute command LOGCMD_FLUSH_MSG_LOG.

Name:	RtLogBufMode
GUID:	{D673F089-0855-4304-860F-99C4DB063DA9}
Visualized:	false
DataType:	gt_FifoBufMode
Type:	Data Channel
Initialize:	true
WriteProtected:	false
Retentive:	false
Comment:	Buffer mode for Realtime data logging. - Log Cyclic Continuous logging - Log Once Until buffer is full

Name:	ip_RtLogFile
GUID:	{CF57D162-A946-416C-85A6-44EE74F7FE5D}
Visualized:	false
DataType:	pChar
Type:	Data Channel
Initialize:	false
WriteProtected:	false
Retentive:	false
Comment:	Pointer to filename string for real datalogging. (Including drive letter and path)

DataType: UDINT
Type: Data Channel
Initialize: false
WriteProtected: true
Retentive: false
Comment: Counter for number of entries actual logged in realtime data buffer

Name: ip_LogData
GUID: {88708DC1-95EC-4EBF-96ED-1F94A509DD92}
Visualized: false
DataType: gpt_LogData
Type: Data Channel
Initialize: false
WriteProtected: false
Retentive: false
Comment: Input Pointer to LogData
Interface for other objects which sends logdata (Log messages or Realtime Data)
to Safan logger

Name: Command
GUID: {9E8B59E0-2A63-40C7-9E3C-8741C29C73FF}
Visualized: false
DataType: gt_LogCmd
Type: Data Channel
Initialize: true
WriteProtected: false
Retentive: false
Comment: Log command
Send a log command.
(See gt_LogCmd for available commands)

Methods:

Name: Init
Virtual: true
Global access: true
AWL implementation: false
CDecl: false

Name: Background
Virtual: true
Global access: true
AWL implementation: false
CDecl: false
Input:: Name: EAX
Type: UDINT
Pointer: false
Register: EAX
Output:: Name: state
Type: UDINT
Pointer: false
Register: EAX

Name: SFN_Logger
Virtual: false
Global access: false
AWL implementation: false
CDecl: false
Comment: Constructor
Output:: Name: ret_code
Type: ConfStates
Pointer: false
Register: <undefined>

Name: HandleRtLogState
Virtual: false
Global access: false
AWL implementation: false
CDecl: false
Comment: Handle state for Realtime logging.

Global access: false
AWL implementation: false
CDecl: false
Comment: Handle received LogCommand

Name: SetLogLevel
Virtual: false
Global access: true
AWL implementation: false
CDecl: false
Comment: Set's the log level used to filter messages
This can be either of the following ;
- LL_NONE ; no logging
- LL_DEBUG ; log debug, info, warning and error messages
- LL_INFO ; log information, warning and error messages
- LL_WARNING ; log warning and error messages
- LL_ERROR APPLICATION : Log application error and other error messages
- LL_ERROR INTERNAL : log Internal and critical error messages
- LL_ERROR CRITICAL ; log only critical error messages

Note: LL_DEBUG_RT will be modified to LL_DEBUG

Input:: Name: NewLevel
Type: gt_LogLevel
Pointer: false
Register: <undefined>

Name: SetMsgLogTarget
Virtual: false
Global access: true
AWL implementation: false
CDecl: false
Comment: Set target where to log.
Input:: Name: NewTarget
Type: gt_LogTarget
Pointer: false
Register: <undefined>
Output:: Name: Result
Type: DINT
Pointer: false
Register: <undefined>

Name: SetMsgLogFlushTime
Virtual: false
Global access: true
AWL implementation: false
CDecl: false
Comment: Set flush time for message logging
Input:: Name: NewFlushTime
Type: UDINT
Pointer: false
Register: <undefined>

Name: SetRtLogFile
Virtual: false
Global access: true
AWL implementation: false
CDecl: false
Comment: Set Current File Path for Real time data logging
Input:: Name: NewLogFile
Type: ^CHAR
Pointer: true
Register: <undefined>

Name: ResetLogMsgData
Virtual: true
Global access: true
AWL implementation: false
CDecl: false
Comment: Reset logging message data

Name: ResetLogRtData
Virtual: true
Global access: true
AWL implementation: false
CDecl: false
Comment: Reset Realtime data logging,
Delete file and set all varibales
and parameters to default

Name: InstallRealtimeLogBuffer
Virtual: false
Global access: false
AWL implementation: false
CDecl: false
Comment: Install Log buffer for realtime logging

Name: SetTriggerRtLog
Virtual: false
Global access: true
AWL implementation: false
CDecl: false
Comment: Set trigger condition for realtime data logging
Input::
Name: p_SetTrigger
Type: ^gt_RtLogTrigger
Pointer: true
Register: <undefined>

Name: GetTriggerRtLog
Virtual: false
Global access: true
AWL implementation: false
CDecl: false
Comment: Get actual Realtime log trigger settings
Input::
Name: p_GetTrigger
Type: ^gt_RtLogTrigger
Pointer: true
Register: <undefined>

Name: IncRtLogIdx
Virtual: false
Global access: false
AWL implementation: false
CDecl: false
Comment: Increment log index for realtime LogDataBuffer and handle wrap around
Input::
Name: p_Idx
Type: ^UDINT
Pointer: true
Register: <undefined>

Name: GetFirstStoredRtLogEntry
Virtual: false
Global access: false
AWL implementation: false
CDecl: false
Comment: Calculate first Index to read from realtime LogDataBuffer
Output::
Name: FirstIdx
Type: UDINT
Pointer: false
Register: <undefined>

Name: WriteHeaderRtLog
Virtual: false
Global access: false
AWL implementation: false
CDecl: false
Comment: Write header of Realtime log file

Name: _WriteMessageData
Virtual: false
Global access: false
CDecl: false

Input:: Name: p_LogData
Type: gpt_LogData
Pointer: false
Register: <undefined>

Name: _WriteRtData
Virtual: false
Global access: false
AWL implementation: false
CDecl: false
Comment: Log Realtime data to internal RAM buffer
If logging is not active yet, trigger must be found first

Input:: Name: p_LogData
Type: gpt_LogData
Pointer: false
Register: <undefined>

Name: RegistrateSfnObject
Virtual: false
Global access: false
AWL implementation: false
CDecl: false
Comment: Registrate SFN_Objects.
- Add objects in chain
Input:: Name: p_RegObj
Type: ^SFN_Base
Pointer: true
Register: <undefined>
Comment: object to register

Types: Name: t_LogFlags
Type: BSINT
Size: 1
Public: false
Comment: Internal log flags

To be able to make an own filter selection of data to be logged.

Name: t_RtLogData
Type: STRUCT
Alignment: 1 Byte
Size: 12
Public: false
Comment: RealTime log data structure
Element: Name: RelLogTime
Type: UDINT
Pointer: false
Size: 4
Public: false
Element: Name: Id
Type: UDINT
Pointer: false
Size: 4
Public: false
Element: Name: Value
Type: DINT
Pointer: false
Size: 4
Public: false

Name: t_RtLogDataTable
Type: ARRAY OF t_RtLogData
ElementType: t_RtLogData
Pointer: false
Size: 60012
Public: false

Comment: Define a table layout so we can view memory
Be aware of table size in this structure.
This is not the real used buffersize but just
for debugging to view memory content
The real size is allocated via RtNrOfLogEntries.
We use "extended allocated" memory (Dynamic buffer) to be able using large memory (> 64K) in a class

Element:
Limit Low: 0
Limit Low value: 0
Limit High: 5000
Limit High value: 5000

Name: t_RtLogState
Type: ENUM
Size: UDINT
Public: false
Comment: States used for realtime datalogging
Elements: RT_LOG_IDLE, RT_LOG_ERROR, RT_LOG_ACTIVE, RT_LOG_WAIT_FOR_STARTTRIGGER, RT_LOG_WAIT_FOR_STOPTRIGGER, RT_LOG_SAVE_TO_FILE

Variables:

Name: ActiveLogFlags
Type: t_LogFlags
ElementType: t_LogFlags
Pointer: false
Comment: Internal flags to select what should be logged.
For debugging purpose we can change log bits also runtime to filter the specific messages

Name: timeLastNewMsg
Type: UDINT
ElementType: UDINT
Pointer: false
Comment: Time of last received new message

Name: MsgInBuffer
Type: BOOL
ElementType: BOOL
Pointer: false
Comment: Log message in message buffer

Name: rtFlushDataCnt
Type: UDINT
ElementType: UDINT
Pointer: false
Comment: Number of entries left to flush Realtime data logging to file

Name: p_RtLogTable
Type: ^t_RtLogDataTable
ElementType: t_RtLogDataTable
Pointer: true
Comment: Pointer to extended memory for storage of Realtime logdata.
Be aware of table size in this structure.
This is not the real used buffersize but just
for debugging to view memory content
The real size is allocated via RtNrOfLogEntries.
We use "extended allocated" memory (Dynamic buffer) to be able using large memory (> 64K) in a class

Name: WrRtDataIdx
Type: UDINT
ElementType: UDINT
Pointer: false
Comment: Actual entry index to write Realtime log data to buffer

Name: RdRtDataIdx
Type: UDINT
ElementType: UDINT
Comment: Actual entry index to read Realtime log data from buffer

Name: RtLogState
Type: t_RtLogState
ElementType: t_RtLogState
Pointer: false
Comment: State of Real time data logging

Name: RtTriggerSetting
Type: gt_RtLogTrigger
ElementType: gt_RtLogTrigger
Pointer: false
Comment: Actual trigger setting used to determine a trigger event.

Name: CmdRequest
Type: gt_LogCmd
ElementType: gt_LogCmd
Pointer: false
Comment: Actual command/status request

Name: StartDateRtLog
Type: SYSDATE
ElementType: SYSDATE
Pointer: false

Name: StartTimeRtLog
Type: SYSTIME
ElementType: SYSTIME
Pointer: false

Name: StartMsTimeRtLog
Type: UDINT
ElementType: UDINT
Pointer: false
Comment: Abs time of first Rt message

Defines:
Dependencies:
Class: SFN_Base
Class: SFN_DynamicBuffer
Class: SFN_FileBase
Class: SFN_LinkedListBase
Class: SFN_StringUtils
Types: ChDsc
Types: ChDscEntry
Types: ChMeth
Types: ChMode
Types: ClsHdr
Types: ClsHdrConst
Types: CltCh
Types: ConfStates
Types: gpt_LogData
Types: gt_FifoBufMode
Types: gt_LogCmd
Types: gt_LogData
Types: gt_LogLevel
Types: gt_LogTarget
Types: gt_ObjectStatus
Types: gt_RtLogTrigger
Types: Obj
Types: ObjDsc
Types: pChar
Types: pClsHdr
Types: pFct
Types: pVoid
Types: Revision
Types: SvrCh
Types: SvrDsc
Types: SYSDATE
Types: SYSTIME

File: C:\DevEnv\Sigmatek\Projects\SafanStdLib\SafanStdLib\Class\SFN_Base\SFN_Base.pdf
File: C:\DevEnv\Sigmatek\Projects\SafanStdLib\SafanStdLib\Class\SFN_LinkedListBase\SFN_Linked
File: C:\DevEnv\Sigmatek\Projects\SafanStdLib\SafanStdLib\Class\SFN_Logger\SFN_Logger.pdf
File: C:\DevEnv\Sigmatek\Projects\SafanStdLib\SafanStdLib\Class\SFN_DynamicBuffer\SFN_Dynam
File: C:\DevEnv\Sigmatek\Projects\SafanStdLib\SafanStdLib\Class\SFN_FileBase\SFN_FileBase.pdf