

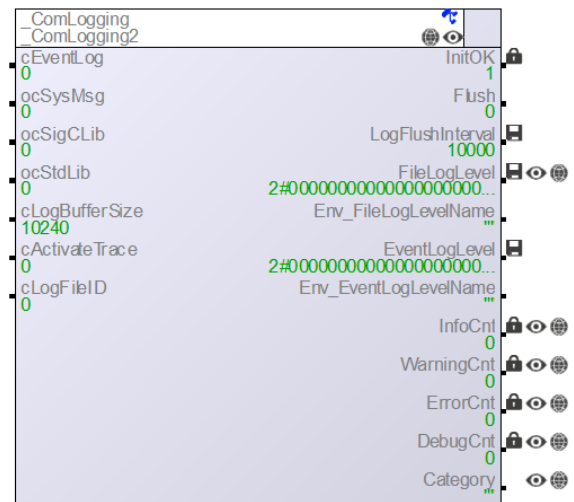
_ComLogging

Diese Klasse ist eine Ableitung der _UserLogging Klasse.

Sie erlaubt es durch Aufrufen von Methoden Meldungen in Log-Dateien, das Eventjournal und den Debug Trace zu schreiben.

Dabei werden der entsprechenden Methode Informationen wie z.B. die Art der Meldung, der Meldungstext und Parameter übergeben, welche dann verarbeitet werden und in einem Text ausgegeben bzw. gespeichert werden.

Des Weiteren kann der Nutzer bestimmen welche Art von Meldungen verarbeitet werden sollen und wo sie gespeichert werden.



Inbetriebnahme der Klasse

Zur Inbetriebnahme der Klasse muss diese entweder in ein normales Netzwerk oder in ein komplexes Netzwerk (als Unterklasse einer anderen) platziert werden.

Möchte man, dass die Log Meldungen auch in EventJournal geschrieben werden sollen, muss der Client „cEventLog“ mit dem ClassSvr der Klasse EventQueue des EventJournal AddOns als CommandChannel verbunden werden.

Im Inneren der _ComLogging Klasse liegt die _UserLogging Klasse, auf deren Clients standardmäßig folgende Parameter eingestellt sind. Der Client cLogBufferSize, der die Größe des Log Speichers bestimmt, ist auf 10240 bytes (10 kilobytes) eingestellt. Der cActivateTrace Client, der eingeschaltet wird um Debug-Trace Meldungen auszugeben, ist standardmäßig auf „0“, also deaktiviert. Der cLogFileID Client, der die Log Dateinummer bestimmt, ist standardmäßig auf 6 eingestellt, damit „C:/SYSMSG/Event16.log“ zum schreiben der Log Datei verwendet wird.

Der Server FileLogLevel bekommt seinen Wert von einer Environment Variable und ist standardmäßig auf 2#01100 eingestellt. Der EventLogLevel bekommt seinen Wert auch von einer Environment Variable und ist standardmäßig auf 10#0“ (entspricht 2#0) eingestellt. Die beiden Server dienen als Filterfunktion um zu bestimmen, welche Arten von Meldungen verarbeitet werden und welche nicht.

Dabei stehen folgende Bit-Werte für folgende Optionen:

2#00001 = Info
 2#00100 = Warning
 2#01000 = Error
 2#10000 = Debug

Steht der Server FileLogLevel beispielsweise auf 2#00101, so werden nur eingehende Meldungen welche Info-Meldungen (2#00001) oder Warning-Meldungen (2#00100) sind, in der Log Datei gespeichert. Da auf dem Server FileLogLevel standardmäßig der Wert 2#01100 initialisiert wird, werden nur Warnungen und Error-Meldungen in der Log Datei gespeichert. Möchte man beispielsweise, dass auch Info-Meldungen gespeichert werden, muss der Server auf 2#01101 geändert werden.

Der Server Env_FileLogLevelName zeigt den Namen der Environment Variable für den FileLogLevel und ist standardmäßig „LOGLEVEL_FILE_XXXX“. Der Server Env_EventLogLevelName zeigt den Namen der Environment Variable für den EventLogLevel und ist standardmäßig „LOGLEVEL_EVENTLOG_XXXX“.

Wird in der autoexec.isl der die Environment Variable gesetzt, wird diese beim Startup geladen und ist der neue Standardwert. Wird kein Wert in der autoexec.isl bzw. der Environment Variable gesetzt, wird nach dem Startup der auf dem Server gespeicherte Wert weiterverwendet.

Der Befehl für das Setzen der Environment Variable in der autoexec.isl sieht folgendermaßen aus:

```
SETENV LOGLEVEL_FILE_XXXX YYYY
```

beziehungsweise:

```
SETENV LOGLEVEL_EVENTLOG_XXXX YYYY
```

Beispiel eines Log Eintrages:

Hier ein Beispiel zur Anschauung des Aufbaus von Log Einträgen:

EMAP77;_COMLOGGING1;Error;44;1800;Text der Meldung;2222;7777;1111;4444

EMAP77: Dies ist die Kategorie, welche vom String Objekt gelesen wird.

COMLOGGING1: Hierbei handelt es sich um den Namen des Objektes in welchem die _UserLogging oder die _ComLogging Klasse platziert wurde.

Error: Dies ist die durch den Level definierte Art der Meldung. Sie kann, Info, Error, Warning oder Debug sein.

44: Diese Zahl stellt die Message Group dar.

1800: Diese Zahl stellt die Message Number dar.

Text der Meldung: An dieser Stelle steht der eingegebene Text der Meldung.

2222: Parameter 1

7777: Parameter 2

1111: Parameter 3

4444: Parameter 4

Globale Methoden

AddUserEntry	<p>Methode zum Übergeben von Daten, die dann entweder in einer Log Datei oder dem EventJournal gespeichert werden und als Debug-Trace Meldung ausgegeben werden können. Die Eingabewerte werden dann an die _UserLogging Klasse übergeben, wo die AddEntry Methode aufgerufen und mit den Eingabewerten der AddUserEntry gespeist wird. Anders als bei einem Aufruf der AddEntry Methode, müssen die Kategorie (pCategory) und der Objektname (pObjectname) bei dieser Methode nicht übergeben werden, da bei einem direkten Aufruf der AddUserEntry Methode der Objektname der Klasse, in welcher sich die _ComLogging Klasse befindet, automatisch ermittelt wird und die Kategorie von einem String Objekt im Inneren der _ComLogging Klasse abgelesen wird.</p> <p>=> Level: UDINT: Wert aus dem sich die Art der Meldung erschließen lässt => MsgGroup: DINT: Nummer der Gruppe der Meldung => MsgNbr: DINT: Nummer einer Nachricht => pMsg: ^CHAR: Zeiger auf den String des Meldungstextes => pPara1: ^DINT: Zeiger auf einen Parameter der übergeben wird => pPara2: ^DINT: Zeiger auf einen Parameter der übergeben wird => pPara3: ^DINT: Zeiger auf einen Parameter der übergeben wird => pPara4: ^DINT: Zeiger auf einen Parameter der übergeben wird <= Retcode: t_e_AddEntryRet: Rückgabewert zum Erfolg der Methode</p>
---------------------	---

Schnittstellen

Clients

cEventLog	Command Channel zum ClassSvr der Klasse EventQueue des EventJournal AddOns	
	Datentyp	DINT
ocSysMsg	Object Channel zur _SysMsg Klasse.	
	Datentyp	DINT
ocSigCLib	Object Channel zur SigCLib Klasse.	
	Datentyp	
ocStdLib	Object Channel zur _StdLib Klasse.	
	Datentyp	DINT
cLogBufferSize	Auf diesem Client wird die Grösse des Speichers für Log-Meldungen bestimmt	
	Datentyp	UDINT
cActivateTrace	Wird dieser Client auf „1“ gesetzt, werden alle Log Meldungen als Debug-Trace Meldungen ausgegeben.	
	Datentyp	UDINT
cLogFileID	Auf diesem Client wird die Dateinummer, mit welcher die Log Datei auf der Steuerung gespeichert werden soll, eingestellt. Dieser Wert sollte zwischen 1 und 9 liegen.	
	Datentyp	UDINT

Server

InitOK	Der ClassSvr kann als Object Channel verwendet werden, um die AddUserEntry Methode von außen aufrufen zu können. Des Weiteren zeigt er den Status nach dem Aufruf der Initialisierungs-Methode.			
	Einheit	-	Datentyp	DINT
	Wertebereich	-	Write Protected	TRUE
	Defaultwert	-	Retentive	FALSE
Flush	Dieser Server hat eine Write Methode, welche beim Aktivieren einen Flush auslöst. Dadurch werden die im Speicher befindlichen Log Meldungen in einer Log Datei auf der Steuerung abgespeichert. Bei Erfolg wird „0“ auf den Server geschrieben, bei Misserfolg „-2“.			
	Einheit	-	Datentyp	DINT
	Wertebereich	-	Write Protected	FALSE
	Defaultwert	-	Retentive	FALSE
LogFlushInterval	Auf diesem Server kann der zeitliche Zyklus des Flushes bestimmt werden. Hierbei wird die Zeitspanne in Millisekunden auf den Server geschrieben, nach welcher ein Flush zyklisch aktiviert werden soll.			
	Einheit	-	Datentyp	UDINT
	Wertebereich	-	Write Protected	FALSE
	Defaultwert	10000	Retentive	File
FileLogLevel	Auf diesen Server wird der FileLoggingLevel, welcher auf einer Environment Variable, die im Inneren der Klasse steht, als BDINT dargestellt und kann auch geändert werden.			
	Einheit	-	Datentyp	BDINT
	Wertebereich	-	Write Protected	FALSE
	Defaultwert	2#1100	Retentive	FILE
Env_File_LogLe	Auf diesem Server erscheint der Name der Environment Variable für den FileLogLe-			

velName	vel. Dieser kann hier auch umbenannt werden.		
	Einheit	-	Datentyp UDINT
	Wertebereich	-	Write Protected FALSE
	Defaultwert	-	Retentive FALSE
EventLogLevel	Auf diesen Server wird der FileLoggingLevel, welcher auf der Environment Variable, die im Inneren der Klasse steht, als BDINT dargestellt und kann auch geändert werden.		
	Einheit	-	Datentyp BDINT
	Wertebereich	-	Write Protected FALSE
	Defaultwert	2#0	Retentive FILE
Env_EventLogLevelName	Auf diesem Server erscheint der Name der Environment Variable für den Event-LogLevel. Dieser kann hier auch umbenannt werden.		
	Einheit	-	Datentyp UDINT
	Wertebereich	-	Write Protected FALSE
	Defaultwert	-	Retentive FALSE
InfoCnt	Auf diesem Server wird die Anzahl an eingegangenen Meldungen der Kategorie „Info“ angezeigt. Bei jeder verarbeiteten Info Meldung wird dieser inkrementiert.		
	Einheit	-	Datentyp DINT
	Wertebereich	-	Write Protected TRUE
	Defaultwert	-	Retentive FALSE
WarningCnt	Auf diesem Server wird die Anzahl an eingegangenen Meldungen der Kategorie „Warning“ angezeigt. Bei jeder verarbeiteten Warning Meldung wird dieser inkrementiert.		
	Einheit	-	Datentyp DINT
	Wertebereich	-	Write Protected TRUE
	Defaultwert	-	Retentive FALSE
ErrorCnt	Auf diesem Server wird die Anzahl an eingegangenen Meldungen der Kategorie „Error“ angezeigt. Bei jeder verarbeiteten Error Meldung wird dieser inkrementiert.		
	Einheit	-	Datentyp DINT
	Wertebereich	-	Write Protected TRUE
	Defaultwert	-	Retentive FALSE
DebugCnt	Auf diesem Server wird die Anzahl an eingegangenen Meldungen der Kategorie „Debug“ angezeigt. Bei jeder verarbeiteten Debug Meldung wird dieser inkrementiert.		
	Einheit	-	Datentyp DINT
	Wertebereich	-	Write Protected TRUE
	Defaultwert	-	Retentive FALSE
Category	Auf diesem Server wird der Name der Kategorie des Events angezeigt. Dieser kommt von einem String Objekt, welches sich im Inneren der _ComLogging Klasse befindet und kann hier auch beschrieben werden.		
	Einheit	-	Datentyp UDINT
	Wertebereich	-	Write Protected FALSE
	Defaultwert	-	Retentive FALSE