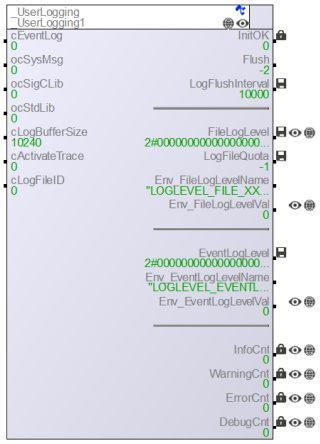
_UserLogging

Diese Klasse 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 (also Unterklasse einer anderen) platziert werden.

Auf dem Client cLogBufferSize wird die Größe des zu reservierenden Speichers für die LogMeldungen definiert. Er ist standardmäßig auf 10240 bytes (10 kilobytes) eingestellt, kann vom Benutzer jedoch beliebig verändert werden. Möchte man, dass die LogMeldungen auch in EventJournal geschrieben werden sollen, muss der Client „cEventLog“ mit dem ClassSvr der Klasse EventQueue des EventJournal AddOns als CommandChannel verbunden werden.

Auf dem Server FileLogLevel wird der Filter Wert ~~a, der sich auf einer EnvironmentVariable, welche sich im inneren der _UserLogging Klasse befindet, a~~ls BDINT dargestellt. Darauf kann definiert werden, welche Art von Meldung (als „Level“ definiert) in der Log Datei gespeichert werden soll und welche nicht. Das selbe gilt für den Server „EventLogLevel“, welcher definiert welche Art von Meldungen ans EventJournal weitergegeben werden sollen.

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 Selver LoggingLevel 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 FileLogLevel und ist standardmäßig „LOGLEVEL_EVENTLOG_XXXX“.

Wird in der autoexec.lsl der die Environment Variable gesetzt, wird diese beim Startup geladen und ist der neue Standardwert. Wird kein Wert in der autoexec.lsl 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.lsl sieht folgendermaßen aus:

SETENV LOGLEVEL_FILE_XXXX ~~XXXXX~~YYYYY
beziehungsweise:
SETENV LOGLEVEL_EVENTLOG_XXXX ~~XXXXX~~YYYYY

hat formatiert: Deutsch (Deutschland)

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

AddEntry	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 Methode ist außerdem threadsafe. => pCategory: ^CHAR: Zeiger auf einen String zur Bezeichnung der Kategorie => 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
Initialise	Methode zur Neuinitialisierung des Objektes. Diese Methode wird in der Init Phase automatisch aufgerufen und muss aufgerufen werden, wenn zur Laufzeit der Wert der ~~cLogFileID, cLogBufferSize~~Clients „cLogFileID und cLogBufferSize“ oder ~~oder~~ der Name der Environment Variable geändert wird, damit auf die Änderungen reagier~~en~~ werden kann. <= Retcodet_e_UserLogInitRet: Rückgabewert zum Erfolg der Methode

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 Log Meldungen auch 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

LogFileQuota

Auf diesem Server kann die maximale Größe für ein Logfile definiert werden. Wird diese überschritten, wird die aktuelle Datei umbenannt und eine neue Datei angelegt. Dabei werden nur zwei Dateien (die aktuelle und die vorige) aufbehalten.

Einheit	-	Datentyp	UDINT
Wertebereich	-	Write Protected	FALSE
Defaultwert	-	Retentive	FILE

Env_File_LogLevelName

Auf diesem Server erscheint der Name der Environment Variable für den FileLogLevel. Dieser kann hier auch umbenannt werden.

Einheit	-	Datentyp	UDINT
Wertebereich	-	Write Protected	FALSE
Defaultwert	-	Retentive	FALSE

Env_FileLogLevelVal

Auf diesem Server wird der FileLogLevel, welcher auf der Environment Variable gespeichert ist, als Dezimalzahl angezeigt.

Einheit	-	Datentyp	DINT
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 EventLogLevel. Dieser kann hier auch umbenannt werden.

Einheit	-	Datentyp	UDINT
Wertebereich	-	Write Protected	FALSE
Defaultwert	-	Retentive	FALSE

Env_EventLogLevelVal

Auf diesem Server wird der EventLogLevel, welcher auf der Environment Variable gespeichert ist, als Dezimalzahl angezeigt.

Einheit	-	Datentyp	DINT
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