
















MQTTClient_Interface

MQTTClient_Interface		MQTT	
MQTTClient_Interface1			
• SigCLib	InterfaceState	MCIS_Connected	  
• Stdlib	ClientID	"03934756"	 
• OpenSSL	ClientActive	1	 
• cUseXML	CleanSession	0	 
• cDataMapPubTimeout	SessionPresent	1	  
• cLogConfig	XMLFilename	"C:/MQTT_Client.xml"	 
• cDataMapSubTimeout	Reset	0	
• cDataMapResetTimeout	ComDiagnosis	0	
• cConnectTimeout			
• cAutoResetTime			
• cEventLog			
• cActivateTrace			

This class provides interfaces for the MQTT client. The client is configured through an “.xml” file stored in the control or through the MQTTClient_Interface_Config(Base) class. An example of such an “.xml” file can be found in the appendix at the end of this document.

Interfaces

Clients

SigCLib	Object channel to the SigCLib class. Does not have to be connected.	
	Data type	DINT
StdLib	Object channel to the StdLib class. Does not have to be connected.	
	Data type	DINT
OpenSSL	Object channel to the OpenSSL class. Does not have to be connected.	
	Data type	DINT
cUseXML	The user can specify whether an XML File should be read on this client: 1 – Use XML file 0 – Do not use XML file	
	Data type	DINT
cLogConfig	This client shows the current logging configuration. The logging information can be set by: 1. Initializing this client 2. Configuration via the .xml file	
	Data type	MQTTClient_Interface::t_bd_LogConfig
cDataMapPubTimeout	Timeout for the interface publish procedure. (ms)	
	Data type	UDINT
cDataMapSubTimeout	Timeout for the interface subscribe procedure. (ms)	
	Data type	UDINT
cConnectTimeout	Communication timeout. (ms)	
	Data type	UDINT

Servers

State	<p>This server can be used to call the global methods of the class. Shows the current status of the MQTT client interface.</p> <table><tr><td>Unit</td><td>-</td><td>Data type</td><td>MQTTClient_Interface::t_e_MQTTClientStates</td></tr><tr><td>Value range</td><td>0-6</td><td>Write Protected</td><td>TRUE</td></tr><tr><td>Default value.</td><td>-</td><td>Retentive</td><td>FALSE</td></tr></table>	Unit	-	Data type	MQTTClient_Interface::t_e_MQTTClientStates	Value range	0-6	Write Protected	TRUE	Default value.	-	Retentive	FALSE
Unit	-	Data type	MQTTClient_Interface::t_e_MQTTClientStates										
Value range	0-6	Write Protected	TRUE										
Default value.	-	Retentive	FALSE										
ClientID	<p>The MQTT client ID must be defined here. This ID must be unique in the system. Set from the .xml file.</p>												
	<table><tr><td>Unit</td><td>-</td><td>Data type</td><td>String</td></tr><tr><td>Value range</td><td>-</td><td>Write Protected</td><td>FALSE</td></tr><tr><td>Default value.</td><td>-</td><td>Retentive</td><td>FALSE</td></tr></table>	Unit	-	Data type	String	Value range	-	Write Protected	FALSE	Default value.	-	Retentive	FALSE
Unit	-	Data type	String										
Value range	-	Write Protected	FALSE										
Default value.	-	Retentive	FALSE										
ClientActive	<p>Via this server, the MQTT client can be activated and deactivated.</p> <p>0...deactivated 1...activated.</p> <table><tr><td>Unit</td><td>-</td><td>Data type</td><td>DINT</td></tr><tr><td>Value range</td><td>0-1</td><td>Write Protected</td><td>FALSE</td></tr><tr><td>Default value.</td><td>-</td><td>Retentive</td><td>FALSE</td></tr></table>	Unit	-	Data type	DINT	Value range	0-1	Write Protected	FALSE	Default value.	-	Retentive	FALSE
Unit	-	Data type	DINT										
Value range	0-1	Write Protected	FALSE										
Default value.	-	Retentive	FALSE										
CleanSession	<p>Here, whether the session from the MQTT client should be stored on the server can be set (e.g. subscribed topics)</p> <p>0...The MQTT server should save the current session from the client if the connection is lost. 1...The MQTT server discards the previous session if a new connection is made or the connection is lost.</p> <p>This setting is only used while creating the connection.</p> <table><tr><td>Unit</td><td>-</td><td>Data type</td><td>DINT</td></tr><tr><td>Value range</td><td>0-1</td><td>Write Protected</td><td>FALSE</td></tr><tr><td>Default value.</td><td>-</td><td>Retentive</td><td>FALSE</td></tr></table>	Unit	-	Data type	DINT	Value range	0-1	Write Protected	FALSE	Default value.	-	Retentive	FALSE
Unit	-	Data type	DINT										
Value range	0-1	Write Protected	FALSE										
Default value.	-	Retentive	FALSE										
SessionPresent	<p>Indicates whether an active session was stored on the MQTT server.</p>												

	0...no session from the client is stored 1...a session from the client is available on the server.			
	Unit	-	Data type	DINT
	Value range	0-1	Write Protected	TRUE
	Default value.	-	Retentive	FALSE
XMLFilename	Here, the path of the ".xml" file for configuring the MQTT client must be defined. (Including file name and ending.)			
	Unit	-	Data type	String
	Value range	-	Write Protected	FALSE
	Default value.	"C:/MQTT_Client.xml"	Retentive	FALSE
Reset	Via this server, the class can be reset from the error status. The MQTT client is thereby reinitialized. Must be set to 1.			
	Unit	-	Data type	DINT
	Value range	-	Write Protected	FALSE
	Default value.	-	Retentive	FALSE
ComDiagnosis	Server to connect the command channel of the ComDiagnosis_MQTT object to. This server contains the required NewInst() methods for the functioning of the object.			
	Unit	-	Data type	DINT
	Value range	-	Write Protected	FALSE
	Default value.	-	Retentive	FALSE

Global Methods

Init	Class initialization method. Allocate from memory.	
CyWork	Internal routine call	
MQTT_ReadXML	Can be called to reread the configuration of the “.xml” file. The MQTT client is thereby reinitialized.	
	◀ bReadStarted (BOOL)	True...reading of the configuration file was enabled False...reading the configuration file could not be enabled
MQTT_ReadLogConfig	With this method, the logging configuration can be reread from the cLogConfig client.	
	◀ bUpdated (BOOL)	True...logging configuration was updated False...logging configuration could not be updated
MQTT_Subscribe	With this method, the user subscribes to a specific topic.	
	▶ pTopic (^UINT)	Pointer to the topic name
	▶ iQoS (INT)	<p>Here, the “Quality of Service” for data transfer can be set. This is always between a client and server (Broker).</p> <p>0...Data are sent once, at most. The recipient does not confirm receiving the data. Provides the same guarantee as the underlying TCP protocol.</p> <p>1...Data are sent once, at least. The sender waits for confirmation from the recipient. If no confirmation is given within a specified time, the data are sent again. It is possible that the same data are frequently sent and received.</p> <p>2...Data are sent exactly once. Here, that the recipient receives the message exactly one time is ensured.</p> <p>Note: The higher the “Quality of Service” is set, the longer it takes to process the sending and receiving routines.</p>

	► pThis (^VOID)	The pointer from the object calling.
	► pCallback (^VOID)	Pointer to the callback method of the object calling. Information on the subscribed topic is sent via the callback. See the appendix for the callback structure.
	► pMid (^INT)	Pointer to a variable, in which the message ID should be stored.
	► pApplMsgID (^UDINT)	Pointer to a variable, in which the internal applications message ID should be stored. This ID can be used for the MQTT_Unsubscribe function.
	◄ retcode (INT):	0...successful 1...no memory available for further login. 3...an error has occurred.
MQTT_Unsubscribe		
With this method, the user can unsubscribe from a specific topic.		
	► pApplMsgID (^UDINT)	Pointer to a variable where the internal application message ID was stored. (MQTT_Subscribe) If the parameter pTopic is set, this ID is not required.
	► pTopic (^UINT)	Pointer to the topic name If the parameter pApplMsgID is set, the name is not required.
	► pMid (^INT)	Pointer to a variable, in which the message ID should be stored.
	► pThis (^VOID)	The pointer from the object calling.
	► pCallback (^VOID)	Pointer to the callback method of the object calling. Information on the cancellation is sent via the callback. See the appendix for the callback structure.
	◄ retcode (INT):	0...successful 1...no memory available in message buffer. 3...invalid input parameter 4...no connection to the MQTT server
MQTT_Publish		
With this method, data for a specific topic can be sent.		

	<p>► pTopic (^UINT)</p>	<p>Pointer to the topic name</p>
	<p>► iQoS (INT)</p>	<p>Here, the “Quality of Service” for data transfer can be set.</p> <p>This is always between a client and server (Broker).</p> <p>0...Data are sent once, at most. The recipient does not confirm receiving the data. Provides the same guarantee as the underlying TCP protocol.</p> <p>1...Data are sent once, at most. The sender waits for confirmation from the recipient. If no confirmation is given within a specified time, the data are sent again. It is possible that the same data are frequently sent and received.</p> <p>2...Data are sent exactly once. Here, that the recipient receives the message exactly one time is ensured.</p> <p>Note: The higher the “Quality of Service” is set, the longer it takes to process the sending and receiving routines.</p>
	<p>► pPayload (^VOID)</p>	<p>Pointer to the data that should be sent.</p>
	<p>► udPayloadLen (UDINT)</p>	<p>Data length in bytes</p>
	<p>► bRetain (BOOL)</p>	<p>Here, whether the data sent to the server for a specific topic (Publish) should be stored. If a new client logs into this topic, it is sent the most recently data stored.</p> <p>Otherwise, the new client only receives data as soon as something is set to this topic.</p> <p>False...data are not stored True...data are stored</p>
	<p>► pThis (^VOID)</p>	<p>The pointer from the object calling.</p>
	<p>► pCallback (^VOID)</p>	<p>Pointer to the callback method of the object calling.</p> <p>Via the callback, information is provided for sending.</p> <p>See the appendix for the callback structure.</p>
	<p>► pMid (^INT)</p>	<p>Pointer to a variable, in which the message ID should be stored.</p>
	<p>► pAppMsgID (^UDINT)</p>	<p>Pointer to a variable, in which the internal applications message ID should be stored.</p>

	<table> <tr> <td>◀ retcode (INT):</td><td> 0...successful 1...no memory available 3...input parameter invalid 4...no connection to the MQTT server 18...topic name is not formatted correctly. (Unicode) </td></tr> </table>	◀ retcode (INT):	0...successful 1...no memory available 3...input parameter invalid 4...no connection to the MQTT server 18...topic name is not formatted correctly. (Unicode)														
◀ retcode (INT):	0...successful 1...no memory available 3...input parameter invalid 4...no connection to the MQTT server 18...topic name is not formatted correctly. (Unicode)																
MQTT_SetCleanSession	<p>With this method, all open sessions can be deleted. All elements entered for the "Publish" & "Subscribe" processes are removed.</p> <table> <tr> <td>◀ bOk (BOOL)</td><td> True...all sessions were deleted False...deleting the sessions failed </td></tr> </table>	◀ bOk (BOOL)	True...all sessions were deleted False...deleting the sessions failed														
◀ bOk (BOOL)	True...all sessions were deleted False...deleting the sessions failed																
Logging_CreateLog	<p>With this method, log entries can be created.</p> <table> <tr> <td>▶ usLevel (USINT)</td><td> Log entry level: Info (2#00001) Warning (2#00100) Error (2#01000) Debug (2#10000) </td></tr> <tr> <td>▶ udMsgGroup (UDINT)</td><td>Message group number</td></tr> <tr> <td>▶ udMsgNbr (UDINT)</td><td>Message number</td></tr> <tr> <td>▶ pMsg (^CHAR)</td><td>Pointer to the message text</td></tr> <tr> <td>▶ pPara1 (^DINT)</td><td>Pointer to an optional parameter</td></tr> <tr> <td>▶ pPara2 (^DINT)</td><td>Pointer to an optional parameter</td></tr> <tr> <td>▶ pPara3 (^DINT)</td><td>Pointer to an optional parameter</td></tr> <tr> <td>▶ pPara4 (^DINT)</td><td>Pointer to an optional parameter</td></tr> </table>	▶ usLevel (USINT)	Log entry level: Info (2#00001) Warning (2#00100) Error (2#01000) Debug (2#10000)	▶ udMsgGroup (UDINT)	Message group number	▶ udMsgNbr (UDINT)	Message number	▶ pMsg (^CHAR)	Pointer to the message text	▶ pPara1 (^DINT)	Pointer to an optional parameter	▶ pPara2 (^DINT)	Pointer to an optional parameter	▶ pPara3 (^DINT)	Pointer to an optional parameter	▶ pPara4 (^DINT)	Pointer to an optional parameter
▶ usLevel (USINT)	Log entry level: Info (2#00001) Warning (2#00100) Error (2#01000) Debug (2#10000)																
▶ udMsgGroup (UDINT)	Message group number																
▶ udMsgNbr (UDINT)	Message number																
▶ pMsg (^CHAR)	Pointer to the message text																
▶ pPara1 (^DINT)	Pointer to an optional parameter																
▶ pPara2 (^DINT)	Pointer to an optional parameter																
▶ pPara3 (^DINT)	Pointer to an optional parameter																
▶ pPara4 (^DINT)	Pointer to an optional parameter																
Logging_GetLastClientEvent	<p>This method delivers the last Event that was logged by the MQTTClient object.</p> <table> <tr> <td>▶ pGroup (DINT)</td><td>Pointer to the DINT variable where the Group number of the last event must be written.</td></tr> <tr> <td>▶ pMessage (DINT)</td><td>Pointer to the DINT variable where the Message number of the last event must be written.</td></tr> <tr> <td>▶ Level (DINT)</td><td>Level of the last event of which the Group and Message numbers should be written.</td></tr> <tr> <td>◀ bOK (BOOL)</td><td>True...OK</td></tr> </table>	▶ pGroup (DINT)	Pointer to the DINT variable where the Group number of the last event must be written.	▶ pMessage (DINT)	Pointer to the DINT variable where the Message number of the last event must be written.	▶ Level (DINT)	Level of the last event of which the Group and Message numbers should be written.	◀ bOK (BOOL)	True...OK								
▶ pGroup (DINT)	Pointer to the DINT variable where the Group number of the last event must be written.																
▶ pMessage (DINT)	Pointer to the DINT variable where the Message number of the last event must be written.																
▶ Level (DINT)	Level of the last event of which the Group and Message numbers should be written.																
◀ bOK (BOOL)	True...OK																

		False...pGroup = NIL / pMessage = NIL.
Log- ging_GetLastInterfaceEvent	This method delivers the last Event that was logged by the MQTTClient_Interface object.	
	► pGroup (DINT)	Pointer to the DINT variable where the Group number of the last event must be written.
	► pMessage (DINT)	Pointer to the DINT variable where the Message number of the last event must be written.
	► Level (DINT)	Level of the last event of which the Group and Message numbers should be written.
	◄ bOK (BOOL)	True...OK False...pGroup = NIL / pMessage = NIL.
Log- ging_GetLastTopicEvent	This method delivers the last Event that was logged by an MQTTTopic object.	
	► pGroup (DINT)	Pointer to the DINT variable where the Group number of the last event must be written.
	► pMessage (DINT)	Pointer to the DINT variable where the Message number of the last event must be written.
	► Level (DINT)	Level of the last event of which the Group and Message numbers should be written.
	◄ bOK (BOOL)	True...OK False...pGroup = NIL / pMessage = NIL.
Log- ging_GetLastGlobalEvent	This method delivers the last Event that was logged by the any class connected to the MQTTClient_Interface object.	
	► pGroup (DINT)	Pointer to the DINT variable where the Group number of the last event must be written.
	► pMessage (DINT)	Pointer to the DINT variable where the Message number of the last event must be written.
	► Level (DINT)	Level of the last event of which the Group and Message numbers should be written.
	◄ bOK (BOOL)	True...OK False...pGroup = NIL / pMessage = NIL.

Appendix

Definition of the Callback for the MQTT_Subscribe(), MQTT_Unsubscribe() and MQTT_Publish() Methods:

```
FUNCTION GLOBAL MQTTTopic::PubSubData_Callback
VAR_INPUT
    pThis      : ^void;
    MsgType    : MQTTClient_Interface::t_e_PubSubRetCodes;
    iMid       : INT;
    iGrantedQoS : INT;
    pMessage    : ^MQTTClient::t_s_MQTTMessage;
END_VAR
```

pThis (^VOID)	Pointer to the object
MsgType (t_e_PubSubRetCodes)	Message type: PSRC_Subscribed...return message for Subscribe process. PSRC_Subscribed...return message for Unsubscribe process. PSRC_PublishReveived...retain data from "subscribed" topic. PSRC_Published...Return message for Publish process PSRC_Disconnected...MQTT Client was disconnected
iMid (INT)	Message ID
iGrantedQoS (INT)	Guaranteed Quality of Service from MQTT server. If this value is 16#80 the MsgType is PSRC_Subscribed, the "Subscribe" process has failed.
pMessage (^t_s_MQTTMessage)	Pointer to the message structure.

Sample for the configuration.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<MQTTClient>

  <Config>
    <ClientID>UniqueID</ClientID>
    <Connection>
      <hostname>10.10.55.50</hostname>
      <port>1883</port>
      <username>username</username>
      <password>sigmatek</password>
      <keepalive>60</keepalive>
      <max_inflight_messages>20</max_inflight_messages>
    </Connection>
    <Reconnection>
      <min_interval>3</min_interval>
      <max_interval>30</max_interval>
      <exponential_scale>1</exponential_scale>
    </Reconnection>
    <TLS>
      <cafile>ca.crt</cafile>
      <certfile>certfile.cert</certfile>
      <keyfile>keyfile.key</keyfile>
      <pw>password</pw>
    </TLS>
    <Logging>
      <info>1</info>
      <notice>1</notice>
      <warning>1</warning>
      <error>1</error>
      <debugging>1</debugging>
    </Logging>
    <Will>
      <topic>MyLastTopicWill</topic>
      <payloadtext>My belongings will be handed to...</payloadtext>
      <qos>2</qos>
      <retain>0</retain>
    </Will>
  </Config>
```

```
<DataMap>
<topicsection name="FirstTopicPart">
<topicsection name="SecondTopicPart">
<topicsection name="ThirdTopicPart">
<topicsection name="FourthTopicPart">
  <topicendpoint name="TopicNameDINT" element="Objectname.Servername1" operation="subscribe"
    datatype="dint" qos="2" retain="false"/>
  <topicendpoint name="TopicNameUDINT" element="Objectname.Servername2" operation="poll"
    interval="500" datatype="uint" qos="2" retain="true"/>
  <topicendpoint name="TopicNameREAL" element="Objectname.Servername3" operation="change"
    interval="500" datatype="real" qos="1" retain="true"/>
  <topicendpoint name="TopicNameString" element="Objectname.Servername4" operation="change"
    interval="500" datatype="string" qos="0" retain="false"/>
</topicsection>
</topicsection>
</topicsection>
</DataMap>

</MQTTClient>
```

Elements	Description
<MQTTClient>	First element
<Config>	The configuration area begins here
<ClientID>	Here, the unique ID for the MQTT client must be set
<Connection>	The area for the connection parameters begins here
<hostname>	The MQTT server address must be entered here.
<port>	Here, the MQTT server port must be entered
<username>	If required, a user name can be entered here
<password>	If required, a password can be entered here
<keepalive>	Specifies in seconds, how often a Keep Alive packet is sent
<max_inflight_messages>	Here, how many messages can be processed at one time is defined
<Reconnection>	The area for the reconnection parameters begins here
<min_interval>	Specifies the minimum connection attempt intervals in seconds
<max_interval>	Specifies the maximum connection attempt intervals in seconds
<exponential_scale>	Specifies whether the connection attempt should increase exponentially.
<TLS>	The area for TLS encryption parameters begins here. (optional)
<cafile>	Path of the PEM encoded CA certificate file, which contains the trusted "Root Server". If not path is specified, the default certificate under the path C:\LSLSYS\SSL\ is used.
<certfile>	Path to an optional PEM encoded file, which contains the client certificate (certificate chain). This file must be stored in the directory C:\LSLSYS\SSL\ . The path cannot exceed a length of 255 characters. If a path is sent, a valid "key" file must be included (<keyfile>)
<keyfile>	Path of a PEM encoded file, which contains the "private keys" for the client. This file must be stored in the directory C:\LSLSYS\SSL\ . The path cannot exceed a length of 255 characters.
<pw>	Optional password used for encrypting the "key" file. The password cannot exceed a length of 1023 characters.
<Logging>	The area for the logging parameters begins here
<info>	Indicates whether messages from the Info level should be logged.
<notice>	Indicates whether messages from the Notice level should be logged.
<Warning>	Indicates whether messages from the Warning level should be logged.
<Error>	Indicates whether messages from the Error level should be logged.
<debugging>	Indicates whether messages from the Debug level should be logged.
<Will>	Here, the area for "Will" parameters begins. (optional) If the MQTT server loses the connection to the MQTT client, the server writes the "Will" messages of the client to the defined topic.

<topic>	Topic name for the last will
<payloadtext>	Last will message
<qos>	Quality of Service for the last will
<retain>	Indicates whether the last will should be stored in the MQTT server.
<DataMap>	Here, the area for data mapping begins. (optional)
<Topicsection>	Topic name. If multiple <TopicSection> are nested, the topic name is always expanded until the next <topicendpoint>
<Topicendpoint>	<p>The element to map</p> <p>name ... last part of the topic name. Attached to the nesting of <TopicSection>.</p> <p>element ... reference to the element (objectname.servername)</p> <p>operation ... defines the operation for the element. (subscribe, poll (element sent in interval), change (element sent when changed))</p> <p>interval ... interval time for the operation poll and change. [ms]</p> <p>datatype ... dint, uint, real, string</p> <p>qos ... Quality of Service</p> <p>retain ... indicates whether the message should be stored in the MQTT server.</p>

Definition of the logging entries that can be created in the MQTT Package:

MQTTTopic

Group	Number	Type	Text
30	0	ERROR	Subscribe: Executing the Subscribe method in the MQTTClient_Interface failed. Para1=Mid, Para2 = MQTTClient_interface::MQTT_Subscribe() Retcode.
30	1	ERROR	Subscribe: The MQTTTopic instance subscription has failed. Para1=Last ApplMsgID
30	2	ERROR	Subscribe: Waiting for the Subscription confirmation timed out. The MQTTTopic instance is now in error state. Para1 = Last ApplMsgID
30	3	ERROR	Unsubscribe: Executing the Unsubscribe method in the MQTTClient_Interface failed. The MQTTTopic instance is now in error state. Para1 = RetCode. Para2 = Last ApplMsgID
30	4	ERROR	Unsubscribe: Waiting for the Unsubscribe Confirmation timed out. the MQTTTopic instance is now in error state. Para1 = Last ApplMsgID
30	5	ERROR	Publish: The User passed NIL data to the User_GetPublishData() method, the publish cannot continue, the MQTTTopic instance is now in Error State.
30	6	ERROR	Publish: Executing the Publish method in the MQTTClient_Interface failed. Para1=Retcode. Para2=ApplMsgID
30	7	ERROR	Publish: Waiting for the Publish confirmation timed out. The MQTTTopic instance is now in Error State. Para1=Last ApplMsgID. Para2=mid
32	2000	INFO	Subscribe: The MQTTTopic instance is now Subscribed. Para1=Mid, Para2=Granted QoS
32	2001	INFO	Unsubscribe: The MQTTTopic instance is now Unsubscribed. Para1=Mid
32	2002	INFO	Subscribe: Data has been received on the subscribed topic. Para1=Mid
32	2003	INFO	Publish: The Publish by the MQTTTopic instance has completed successfully. Para1=Mid
32	2004	INFO	Connection: The MQTT Client has been disconnected, the MQTTTopic instance is now waiting for a new connection. Para1= Last ApplMsgID
32	2005	INFO	Publish: The Publish interval has passed, the MQTTTopic instance will now attempt to publish.
32	2006	INFO	Publish: The Publish Data has chnaged, the MQTTTopic instance will now attempt to publish.
32	2007	INFO	Publish: The Manual Publish method has been called, the MQTTTopic instance will now attempt to publish.
32	2008	INFO	FSM: The MQTTTopic instance Config has changed, the instance will no re-initialize. Para1 = Last ApplMsgID
33	3000	DEBUG	FSM: The MQTTTopic instance error has been resetted. Para1= Last ApplMsgID

MQTTClient

Group	Number	Type	Text
20	0	ERROR	TCP: Receiving MQTT Packet timed out, packet dropped. Para1=PacketID
20	1	ERROR	Publish: QoS 2 Outgoing Publish, PUBCOMP Max Retries reached, Packet Dropped. Para1=PacketID
20	2	ERROR	Publish: QoS 2 Outgoing Publish, PUBREL Max Retries reached, Packet Dropped. Para1=PacketID
20	3	ERROR	Publish: QoS 2 Outgoing Publish, PUBREL Insert into Packet Queue Failed, Dropped. Para1=PacketID
20	4	ERROR	Publish: QoS 2 Outgoing Publish, PUBREC Max Retries reached, Packet Dropped. Para1=PacketID
20	5	ERROR	Publish: QoS 2 Outgoing Publish, PUBLISH Max Retries reached, Packet Dropped. Para1=PacketID
20	6	ERROR	Publish: QoS 2 Outgoing Publish, PUBLISH Insert into Packet Queue Failed, Dropped. Para1=PacketID
20	7	ERROR	Publish: QoS 1 Outgoing Publish, PUBACK Max Retries reached, Packet Dropped. Para1=PacketID
20	8	ERROR	Publish: QoS 1 Outgoing Publish, PUBLISH Max Retries Reached, Packet Dropped. Para1=PacketID
20	9	ERROR	Publish: QoS 1 Outgoing Publish, PUBLISH Insert Packet into Packet Queue failed, Dropped. Para1=PacketID
20	10	ERROR	Publish: QoS 0 Outgoing Publish, PUBLISH Max Retries Reached, Packet Dropped. Para1=PacketID
20	11	ERROR	Publish: QoS 0 Outgoing Publish, Message Insert Packet into Packet Queue Failed, Dropped. Para1=PacketID
20	12	ERROR	Publish: QoS 2 Incoming Publish, sending PUBCOMP max retries reached, dropped. Para1=PacketID.
20	13	ERROR	Publish: QoS 2 Incoming Publish, inserting PUBCOMP into Packet Queue failed, dropped. Para1=PacketID
20	14	ERROR	Publish: QoS 2 Incoming Publish, waiting for PUBREL maximum retries reached, dropped. Para1=PacketID.
20	15	ERROR	Publish: QoS 2 Incoming Publish, sending PUBREC maximum retries reached, dropped. Para1=PacketID.
20	16	ERROR	Publish: QoS 2 Incoming Publish, failed to insert PUBREC response message into Message Queue. Para1=PacketID
20	17	ERROR	Publish: QoS 1 Incoming Publish, sending PUBACK max retries reached, dropped. Para1=PacketID.
20	18	ERROR	Publish: QoS 1 Incoming Publish, could not insert the PUBACK into the Packet Queue, dropped. Para1=PacketID.
20	19	ERROR	Unsubscribe: Outgoing Unsubscribe, sending UNSUBSCRIBE Packet max retries reached, dropped. Para1=PacketID.
20	20	ERROR	Subscribe: Outgoing Subscribe, sending SUBSCRIBE Packet max retries reached, dropped. Para1=PacketID.
20	21	ERROR	Publish: Incoming Publish Message, input values not valid, packet dropped. Para1=QoS.
20	22	ERROR	Publish: Incoming Publish Message, adding inflight message failed, Packet Dropped. Para1=PacketID.

20	23	ERROR	Publish: Incoming PUBCOMP Message, incorrect size message received, message not interpreted, dropped.
20	24	ERROR	Publish: Incoming PUBCOMP Message, corresponding PUBLISH message not found, dropped. Para1=PacketID.
20	25	ERROR	Publish: Incoming PUBCOMP Message, corresponding message state not OK. Para1=PacketID.
20	26	ERROR	Publish: Incoming PUBREL Message, incorrect size message received, message not interpreted, dropped.
20	27	ERROR	Publish: Incoming PUBREL Message, corresponding PUBLISH not found, PUBCOMP dropped. Para1=PacketID.
20	28	ERROR	Publish: Incoming PUBREL Message, corresponding message state not OK. Para1=PacketID.
20	29	ERROR	Publish: Incoming PUBREC Message, incorrect size message received, message not interpreted, dropped.
20	30	ERROR	Publish: Incoming PUBREC Message, corresponding PUBLISH not found, PUBCOMP dropped. Para1=PacketID.
20	31	ERROR	Publish: Incoming PUBREC Message, corresponding message state not OK. Para1=PacketID.
20	32	ERROR	Publish: Incoming PUBACK Message, incorrect size message received, message not interpreted, dropped.
20	33	ERROR	Publish: Incoming PUBACK Message, corresponding PUBLISH not found, dropped. Para1=PacketID.
20	34	ERROR	Publish: Incoming PUBACK Message, corresponding message state not OK. Para1=PacketID.
20	35	ERROR	Unsubscribe: Incoming UNSUBACK Message, incorrect size message received, message not interpreted, dropped.
20	36	ERROR	Unsubscribe: Incoming PUBACK Message, corresponding PUBLISH not found, dropped. Para1=PacketID.
20	37	ERROR	Subscribe: Incoming SUBACK Message, incorrect size message received, message not interpreted, dropped.
20	38	ERROR	Subscribe: Incoming SUBACK Message, corresponding SUBSCRIBE not found, dropped. Para1=PacketID.
20	39	ERROR	Subscribe: Incoming SUBACK, 0 granted QoS Pairs received, Dropped. Para1=PacketID.
20	40	ERROR	Keepalive: Incoming PINGREQ Message, incorrect data length, message dropped.
20	41	ERROR	Connection: Incoming CONNACK Message, Incorrect state - CONNACK Unexpected, message has been dropped. Para1=Retcode
20	42	ERROR	Connection: Incoming CONNACK Message, Unexpected RetCode, Connection Failed. Para1=Retcode.
20	43	ERROR	KeepAlive: Outgoing PINGRESP Message, Failed to add to Inflight Messages, Packet Dropped.
20	44	ERROR	KeepAlive: Outgoing PINGREQ Message, Failed to add to Inflight Messages, Packet Dropped.
20	45	ERROR	TCP: TCP Connection Dropped. Para1=TCP ErrorGroup, Para2=TCP ErrorCode. Para3=_FSM_TCP_USER.
20	46	ERROR	TCP: TCP Packet append failed, packet dropped. Para1=Incoming PacketQueueID. Para2=AvailableBytes.

20	47	ERROR	TCP: TCP Communication, Socket Error. Para1=PacketQueue Return. Para2=Packet Len.
20	48	ERROR	MQTT: MQTT Incoming Packet Queue, client received unknown packet, Protocol Error. Dropped.
20	49	ERROR	MQTT: MQTT Incoming Packet Queue, client received DISCONNECT, Protocol Error. Dropped.
20	50	ERROR	MQTT: MQTT Incoming Packet Queue, client received UNSUBSCRIBE, Protocol Error. Dropped.
20	51	ERROR	MQTT: MQTT Incoming Packet Queue, client received SUBSCRIBE, Protocol Error. Dropped.
20	52	ERROR	MQTT: MQTT Incoming Packet Queue, client received CONNECT, Protocol Error. Dropped.
20	53	ERROR	MQTT: MQTT Incoming Packet Queue, Decoding packet length failed, packet dropped.
20	54	ERROR	TCP: TCP Communication, failed to send packet, data dropped. Para1=MessageID. Para2=TCP SendData Retcode.
20	55	ERROR	Connection: Disconnect procedure, sending DISCONNECT timed out, Disconnected.
20	56	ERROR	Connection: Connect procedure, adding CONNECT Packet to Packet Queue Failed, reconnecting.
20	57	ERROR	TCP: Connecting procedure, creating TCP connection timed out, reconnecting.
20	58	ERROR	TCP: Connecting procedure, adding TCP connection failed, reconnecting. Para1=TCP AddConnection Retcode.
20	59	ERROR	Connection: Connecting procedure, failed to get IP address for hostname, reconnecting.
20	60	ERROR	Configuration: Setting the amount of Inflight Messages failed not initialized. Para1=max_inflight_messages.
20	61	ERROR	Configuration: Setting Inflight Messages failed, No mem. Para1=num_msgs, Para2=Mem Avail, Para3=Mem Req
20	62	ERROR	Configuration: Setting Inflight Messages, ReAlloc Failed. Para1=num_msgs, Para2=Mem Avail, Para3=Mem Req.
20	63	ERROR	Subscribe: Subscribe request failed, input not OK.
20	64	ERROR	Publish: Could not accept publish request, invalid input values.
20	65	ERROR	Configuration: Starting the Multitask Thread Failed. Para1=_MultiTask::RESUME() RetCode
20	66	ERROR	Connection: Connect Async Call Failed, the Input to the method is invalid.
20	67	ERROR	MultiTask: The MQTTClient failed to suspend the Multitasking Thread. Para1=_MultiTask::SUSPEND() RetCode
20	68	ERROR	MultiTask: Creating the Multitask Thread Failed.
20	69	ERROR	TCP: The OS TCP User Interface is not available, the connection cannot be made
21	1000	WARNING	Publish: QoS 2 Outgoing Publish, PUBREL wait timed out, Retrying. Para1=PacketID
21	1001	WARNING	Publish: QoS 2 Outgoing Publish, PUBCOMP wait timed out, Retrying. Para1=PacketID
21	1002	WARNING	Publish: QoS 2 Outgoing Publish, PUBREC wait timed out,

			Retrying. Para1=PacketID
21	1003	WARNING	Publish: QoS 2 Outgoing Publish, PUBLISH wait timed out, Retrying. Para1=PacketID
21	1004	WARNING	Publish: QoS 1 Outgoing Publish, PUBACK wait timed out, Retrying. Para1=PacketID.
21	1005	WARNING	Publish: QoS 1 Outgoing Publish, PUBLISH wait timed out, Retrying. Para1=PacketID
21	1006	WARNING	Publish: QoS 0 Outgoing Publish, PUBLISH wait timed out, Retrying. Para1=PacketID
21	1007	WARNING	Publish: QoS 2 Incoming Publish, could not deliver to application as the OnMessage Callback is not set.
21	1008	WARNING	Publish: QoS 2 Incoming Publish, waiting for PUBCOMP send timed out, retrying. Para1=PacketID.
21	1009	WARNING	Publish: QoS 2 Incoming Publish, waiting for PUBREL message timed out, retry sending PUBREC. Para1=PacketID.
21	1010	WARNING	Publish: QoS 2 Incoming Publish, sending PUBREC response timed out, retrying. Para1=PacketID
21	1011	WARNING	Publish: QoS 1 Incoming Publish, could not deliver to application as the OnMessage Callback is not set.
21	1012	WARNING	Publish: QoS 1 Incoming Publish, waiting for PUBACK send timed out, retrying. Para1=PacketID.
21	1013	WARNING	Publish: QoS 0 Incoming Publish, could not deliver to application as the OnMessage Callback is not set.
21	1014	WARNING	Unsubscribe: Outgoing Unsubscribe, waiting for UNSUBSCRIBE Packet send timed out, retrying. Para1=PacketID.
21	1015	WARNING	Subscribe: Outgoing Subscribe, waiting for SUBSCRIBE Packet send timed out, retrying. Para1=PacketID.
21	1016	WARNING	Subscribe: Incoming SUBACK, too many granted QoS Pairs, Truncated. Para1=PacketID.
21	1017	WARNING	Keepalive: MQTT Keepalive, PINGREQ <-> PINGRESP timed out, attempting reconnect.
21	1018	WARNING	Connection: Connect procedure, awaiting CONNACK timed out, attempting reconnect.
21	1019	WARNING	Connection: Reconnect procedure, reconnect wait timed out, retrying.
21	1020	WARNING	Configuration: Setting the Inflight Messages failed, supplied value too low, setting ignored. Para1=Min Value, Para2=Set Value
21	1021	WARNING	Configuration: Set Inf Msgs failed, incorrect MQTTClient state. Para1=MQTTClientState
21	1022	WARNING	Subscribe: Subscription ignored, the MQTTClient is in the incorrect state. Para1=MQTTClientState
21	1023	WARNING	Publish: Publish ignored, the MQTTClient is in the incorrect state. Para1=MQTTClientState
21	1024	WARNING	Connection: A call to mqtt_loop_stop() was made whilst the connection is still active, the call was ignored.
22	2000	INFO	Publish: QoS 1 Outgoing Publish, Message successfully sent. Para1=PacketID
22	2001	INFO	Publish: QoS 2 Outgoing Publish, Message successfully sent.

			Para1=PacketID
22	2002	INFO	Publish: QoS 0 Outgoing Publish, Message successfully send. Para1=PacketID
22	2003	INFO	Publish: QoS 2 Incoming Publish Delivered to Application. Para1=mid
22	2004	INFO	Publish: QoS 1 Incoming Publish Delivered to Application. Para1=mid
22	2005	INFO	Publish: QoS 0 Incoming Publish Delivered to Application. Para1=mid
22	2006	INFO	Publish: Incoming PUBCOMP successfully Processed. Para1=mid
22	2007	INFO	Publish: Incoming PUBREL successfully Processed. Para1=mid
22	2008	INFO	Publish: Incoming PUBREC successfully Processed. Para1=mid
22	2009	INFO	Publish: Incoming PUBACK successfully Processed. Para1=mid
22	2010	INFO	Unsubscribe: Incoming UNSUBACK successfully Processed. Para1=mid
22	2011	INFO	Subscribe: Incoming SUBACK successfully Processed. Para1=mid
22	2012	INFO	Connection: Incoming CONNACK Message, Connection Rejected, Not Authorized.
22	2013	INFO	Connection: Incoming CONNACK Message, Connection Rejected, Username/Password Incorrect.
22	2014	INFO	Connection: Incoming CONNACK Message, Connection Rejected, Server Unavailable.
22	2015	INFO	Connection: Incoming CONNACK Message, Connection Rejected, ID Rejected.
22	2016	INFO	Connection: Incoming CONNACK Message, Connection Rejected, Protocol Version Mismatch.
22	2017	INFO	Connection: Incoming CONNACK Message, Connection Accepted.
22	2018	INFO	Configuration: Set Inflight Messages success. Para1=num_msgs, Para2=Mem Avail, Para3=Mem Req.
22	2019	INFO	Subscribe: Successfully added an Inflight Message for a Subscription. Para1=mid, Para2=QoS
22	2020	INFO	Publish: Successfully added inflight message for publish. Para1=mid
22	2021	INFO	MultiTask: The MultiTask Thread was stopped.

MQTTClient Interface

Group	Number	Type	Text
10	0	ERROR	Init: Failed to assign memory for DataMap Pub & Sub Entries
10	1	ERROR	Init: Failed to assign memory for DataMap Entries
10	2	ERROR	Init: Failed to assign Memory for PubSubList Main Structure
10	3	ERROR	Init: Failed to assign Memory for PubSubList Entries
10	4	ERROR	FSM: Creating Connection the the MQTT Broker Failed, Para1 = MQTTClient Retcode
10	5	ERROR	FSM Initialize: Failed to read the specified XML File.
10	6	ERROR	FSM Initialize: Preparing DataMap Failed - Pointer to DataMap Entries is NIL.
10	7	ERROR	FSM Prepare: Setting client ID failed, not enough memory.
10	8	ERROR	FSM Prepare: Error during setting MQTTClient Parameters from XML File, Setting ClientID Failed, Input Value / Client State Not OK.
10	9	ERROR	FSM Prepare: Error during setting MQTTClient Parameters from XML File, Setting Username & Password Failed, Not Enough memory
10	10	ERROR	FSM Prepare: Error during setting MQTTClient Parameters from XML File, Setting Username & Password Failed, Input Value / Client State Not OK
10	11	ERROR	FSM Prepare: Error during setting MQTTClient Parameters from XML File, Setting Max Inflight Messages Failed, Input Value / Client State Not OK
10	12	ERROR	FSM Prepare: Setting will failed. Not enough memory
10	13	ERROR	FSM Prepare: Setting will failed. Invalid input parameters or wrong client state
10	14	ERROR	FSM Prepare: Setting will failed. Invalid Payload size. Para1 = Payload size
10	15	ERROR	FSM Prepare: Setting will failed. Malformed UTF8
10	16	ERROR	FSM Prepare: Setting reconnect failed. Invalid input values or state. Para1 = T Min. Para2 = T Max. Para3 = Exp
10	17	ERROR	FSM Prepare: Setting TLS Data failed. Not enough Memory
10	18	ERROR	FSM Prepare: Setting TLS Data failed. Invalid input parameters or invalid client state
10	19	ERROR	Subscribe: Subscribe failed, Incorrect Input Para. Para1= pTopic. Para2=Qos. Para3=pThis. Para4 = pCallback
10	20	ERROR	Subscribe: Subscribe Failed, incorrect state. Para1 = InterfaceState
10	21	ERROR	Subscribe: Subscribe failed, No Mem in PubSub list. Para1=NumEntries. Para2=Max Num Entries

10	22	ERROR	Subscribe: Subscribe failed, Para1=ReturnCode from client
10	23	ERROR	Publish: Publish failed, invalid input para. Para1=pTopic. Para2=QoS. Para3=Payload Len. Para4=pPayload
10	24	ERROR	Publish: Publish failed, No Broker connection or not initialized. Para1=InterfaceState
10	25	ERROR	Publish: Publish failed, No Mem in PubSub list. Para1=NumEntries. Para2=Max Num Entries
10	26	ERROR	Publish: Publish failed, Para1=ReturnCode from client
10	27	ERROR	Unsubscribe: Unsubscribe failed, Could not find topic in PubSub list by name
10	28	ERROR	Unsubscribe: Unsubscribe failed, Para1=ReturnCode from client
10	29	ERROR	Unsubscribe: Unsubscribe client request failed, incorrect input parameter. Para1 = Appl Msg Id. Para2 = mid
10	30	ERROR	Unsubscribe: Unsubscribe client request failed, MQTT Client is not connected to the broker. Para1 = Appl Msg Id. Para2 = mid
10	31	ERROR	Unsubscribe: Unsubscribe client request failed, No free memory in message buffer. Para1 = Appl Msg Id. Para2 = Mid
10	32	ERROR	Unsubscribe: Unsubscribe failed, Subscription could not be found in the PubSub List. Para1=ApplMsgID, Para2=mid
10	33	ERROR	Unsubscribe: Unsubscribe failed, incorrect input para
10	34	ERROR	Set Clean Session: Resetting the session failed, PubSub list is not Initialised
10	35	ERROR	Set Clean Session: Resetting the Session failed, MQTT Interface is not in the correct state. Para1 = InterfaceState
10	36	ERROR	Process XML: Root element in XML File is invalid. Cannot process
10	37	ERROR	Process XML Config: Processing Client ID Section of configuration XML file failed - Missing Entries.
10	38	ERROR	Process XML Config: Processing Connection Section of configuration XML file failed - Missing Entries.
10	39	ERROR	Process XML Config: Processing Will section of configuration XML file failed - Missing Entries.
10	40	ERROR	Process XML Config: Processing reconnection section of configuration XML file failed - Missing Entries.
10	41	ERROR	Process XML Config: Processing TLS section of configuratoin XML file failed - Missing Entries
10	42	ERROR	Process XML Config: Processing Logging section of configuration XML file failed - Missing Entries
10	43	ERROR	Process XML DataMap: Unexpected input in DataMap Section
10	44	ERROR	Process XML DataMap: Failed processing a Topic Section in the DataMap section.
10	45	ERROR	Process XML DataMap: Failed processing a Topic Endpoint in the DataMap section
10	46	ERROR	Process XML DataMap Topic Section: Processing Entry Failed - Section title too long.
10	47	ERROR	Process XML DataMap Topic Section: Failed Processing Entry - Section title is empty

10	48	ERROR	Process XML DataMap, Max topic depth (Nested Topic Sections) exceeded. Para1=Topic depth. Para2 = Max topic depth
10	49	ERROR	Process XML DataMap Topic Endpoint: Element name is too long or Zero, Para1 = Len
10	50	ERROR	Process XML DataMap Topic Endpoint: Element text is too long
10	51	ERROR	Process XML DataMap Topic Endpoint: Unknown operation type
10	52	ERROR	Process XML DataMap Topic Endpoint: Interval string too long, Para1 = Len
10	53	ERROR	Process XML DataMap Topic Endpoint: Unknown datatype
10	54	ERROR	Process XML DataMap Topic Endpoint: QoS string too long, Para1 = Len
10	55	ERROR	Process XML DataMap Topic Endpoint: Qos invalid. Para1 = Qos
10	56	ERROR	Process XML DataMap Topic Endpoint: Retain string too long. Para1 = Len
10	57	ERROR	Process XML DataMap Topic Endpoint: Retain value not known. Para1 = Retain value
10	58	ERROR	Process XML DataMap Topic Endpoint: Topic too long, Para1 = Len
10	59	ERROR	Process XML DataMap Topic Endpoint: Topic name too long, Para1 = Len
10	60	ERROR	Process XML DataMap Topic Endpoint: Adding Element to DataMap failed. Para 1 = DataMap RetCode.
10	61	ERROR	Process XML DataMap, Failed processing topic endpoint. All required attributes have not been set
10	62	ERROR	DataMap: Adding DataMap entry failed, Invalid input parameters received.
10	63	ERROR	DataMap: Adding DataMap entry failed, Topic name too long.
10	64	ERROR	DataMap: Adding Element failed - Element name too long. Para 1 = Len
10	65	ERROR	DataMap: Adding Element failed - Element not found.
10	66	ERROR	DataMap: Adding Element Failed - Type Mismatch.
10	67	ERROR	DataMap: Adding Publish Element Failed - Too Many Entries, see client DataMapEntries.
10	68	ERROR	DataMap: Adding Subscribe Element Failed - Too Many Entries, see client DataMapEntries.
10	69	ERROR	DataMap: Failed to allocate memory to send the string for the entry. Para1 = Mid. Para2 = String Object pThis
10	70	ERROR	DataMap: Publish Request Failed. Para1 = Mid, Para2 = MQTTClient RetCode, Para3 = pData
10	71	ERROR	DataMap: Publish Entry timed out whilst waiting for confirmation. Para1 = Mid, Para2 = Timeout, Para3 = pData
10	72	ERROR	DataMap: Request to subscribe failed. Para1 = Mid. Para2 = Retcode, Para3 = pData
10	73	ERROR	DataMap: Subscribe Entry timed out whilst waiting for confirmation. Para1 = Mid, Para2 = Timeout, Para3 = pData

10	74	ERROR	DataMap: Request to unsubscribe failed. Para1 = Mid, Para2 = Retcode, Para3 = pData
10	75	ERROR	DataMap: Unsubscribe Entry timed out whilst waiting for confirmation. Para1 = Mid, Para2 = Timeout, Para3 = pData
10	76	ERROR	DataMap: Failed to remove unsubscribe entry. Para1=Mid, Para2=Index
10	77	ERROR	DataMap: Removing Element Failed, Invalid index supplied. Para1 = Index
10	78	ERROR	DataMap: Received data, there is not enough memory available to process the message. Para1=iMid. Para2 = Message Mid
10	79	ERROR	DataMap: Subscription confirms with invalid QoS. Para1=iMid. Para2=Message Mid.Para3=Granted QoS.Para4=Requested Qos
10	80	ERROR	Connection: The connection was refused, the Protocol Versions mismatch. Para1 = CONNACK RetCode
10	81	ERROR	Connection: Connection failed. Client ID Rejected. Para1 = retcode
10	82	ERROR	Connection: Connection Rejected. Server Unavailable. Para1 = retcode
10	83	ERROR	Connection: Connection Rejected. Username and or Password incorrect. Para1 = retcode
10	84	ERROR	Connection: Connection Rejected. Not Authorized. Para1 = retcode
10	85	ERROR	Connection: Unexpectedly disconnected, KeepAlive Ping timeout or TCP error occurred. Para1 = RetCode
10	86	ERROR	Connection: Unexpectedly disconnected, Retcode unknown. Para1 = RetCode
10	87	ERROR	Publish: Received unexpected Publish Confirm from the MQTTClient, PubSub List not Initied. Para1 = mid
10	88	ERROR	Publish: Received unexpected Message (PUBLISH) from the MQTTClient, PubSub List not Initied. Para1 = mid
10	89	ERROR	Logging: Received a log request with no text. Para1 = Log Level
10	90	ERROR	Logging: Received a log request with incorrectly specified level. Para1 = Log Level
10	91	ERROR	Config Object: Setting the specified Client ID Failed, the string is not set or is too long
10	92	ERROR	Config Object: Setting the specified Hostname Failed, the string is not set or is too long
10	93	ERROR	Config Object: Setting the Port number failed, the specified port is outside of the the valid range. Para1 = Specified Port
10	94	ERROR	Config Object: Setting the KeepAlive Time failed, the specified value is outside of the the valid range. Para1 = Specified Time
10	95	ERROR	Config Object: Setting the Max Inflight Messages failed, the specified value is too low. Para1 = Min Value
10	96	ERROR	Config Object: Setting the specified Username Failed, the string is not set or is too long. Para1 = Max Len
10	97	ERROR	Config Object: Setting the specified Password Failed, the string is not set or is too long. Para1 = Max Len
10	98	ERROR	Config Object: Setting the specified Will Topic Failed, the string is not set or is too long. Para1 = Max Len

10	99	ERROR	Config Object: Setting the specified Will Payload Failed, the string is not set or is too long. Para1 = Max Len
10	100	ERROR	Config Object: Setting the Will QoS failed, the specified value is outside of the the valid range. Para1 = Specified QoS
10	101	ERROR	Config Object: Setting the Min Reconnect Time failed, the value is negative or >= Max Reconnect time.
10	102	ERROR	Config Object: Setting the Max Reconnect Time failed, the value is negative or <= Min Reconnect time.
10	103	ERROR	Config Object: Setting the specified TLS CAFile failed, the string is not set or is too long. Para1 = Max Len
10	104	ERROR	Config Object: Setting the specified TLS Cert File failed, the string is not set or is too long. Para1 = Max Len
10	105	ERROR	Config Object: Setting the specified TLS Key File failed, the string is not set or is too long. Para1 = Max Len
10	106	ERROR	Config Object: Setting the specified TLS Password failed, the string is not set or is too long. Para1 = Max Len
11	1000	WARNING	ClientActive: Disconnect request made although no connection is active
11	1001	WARNING	ClientActive: Unexpected retcode received whilst requesting MQTT Client Disconnect.
11	1002	WARNING	DataMap: Max String Len Exceeded, String Truncated. Para1=Max Payload Len, Para2=Received String Len.
11	1003	WARNING	DataMap: Publish Entry Temp Memory pointer was not NIL, this should however be the case, attempting free again. Para1 = Mid
11	1004	WARNING	DataMap: After publishing a String entry, the Temp Memory Pointer is NIL, this should not be the case, not freed. Para1 = mid, Para2 = String Object pThis
11	1005	WARNING	DataMap: Removing Element failed, DataMap not initialized. Para1 = Index
11	1006	WARNING	DataMap: Received callback whilst DataMap is not initialized. Para1=mid. Para2 = Message Mid
11	1007	WARNING	DataMap Received Data but did not find corresponding entry. Para1 = iMid. Para2 = Message Mid
11	1008	WARNING	DataMap: Subscription confirmed, however the Entry was in the wrong state. Para1 = mid, Para2 = Message mid, Para3 = Entry internal State
11	1009	WARNING	DataMap: Received data too long, truncated. (UDINT). Para1 = iMid. Para2 = Message Mid. Para3 = Received Len
11	1010	WARNING	DataMap: Received data too long, truncated. (DINT). Para1 = iMid. Para2 = Message Mid. Para3 = Received Len
11	1011	WARNING	DataMap: Received data too long, truncated. (REAL). Para1 = iMid. Para2 = Message Mid. Para3 = Received Len
11	1012	WARNING	DataMap: Received empty message. Para1 = iMid. Para2 = Message Mid
11	1013	WARNING	Connection: Received an unexpected Connect Callback. Para1 = Retcode
11	1014	WARNING	Connection: Received an unexpected Disconnected Callback. Para1 = Retcode
11	1015	WARNING	Publish: Received unexpected Publish Confirm from the MQTTClient, item not in the PubSub List. Para1 = mid

11	1016	WARNING	Publish: Received a Publish Confirm from the MQTTClient that could not be delivered to the application, Callback details incorrect. Para1 = mid
11	1017	WARNING	Message: Received a Message from the MQTTClient, the PubSubList is empty. Para1 = mid
11	1018	WARNING	Message: Received a Message from the MQTTClient that could not be delivered to the application, Callback details incorrect. Para1 = mid
11	1019	WARNING	Subscribe: Received an unexpected Subscribe Confirm from the MQTTClient. Para1 = mid
11	1020	WARNING	Subscribe: Received an unexpected Subscribe Confirm from the MQTTClient, the entry is a Publish Entry. Para1 = mid
11	1021	WARNING	Subscribe: Received a Subscribe confirmation from the MQTTClient that could not be delivered to the application, Callback details incorrect. Para1 = mid
11	1022	WARNING	Subscribe: Received unexpected Subscribe Confirm from the MQTTClient, item not in the PubSub List. Para1 = mid
11	1023	WARNING	Unsubscribe: Received an Unsubscribe confirmation from the MQTTClient, the PubSubList is empty. Para1 = mid
11	1024	WARNING	Unsubscribe: Received an unexpected Unsubscribe Confirm from the MQTTClient, the matching entry in the PubSub List is not an Unsubscribe request. Para1 = mid
11	1025	WARNING	Unsubscribe: Received an Unsubscribe confirmation from the MQTTClient that could not be delivered to the application, Callback details incorrect. Para1 = mid
12	2000	INFO	FSM: MQTTClient_Interface has been activated
12	2001	INFO	Subscribe: Element successfully added to subscribe list. Para1= Mid
12	2002	INFO	Unsubscribe: Unsubscribe Request Successfully passed to MQTT Client. Para1 = Appl Msg Id. Para2 = mid
12	2003	INFO	DataMap: Requested Publish for an Entry. Para1 = Mid, Para2 = pData
12	2004	INFO	DataMap: Publish complete. Para1 = Mid, Para2 = pData
12	2005	INFO	DataMap: Requested subscription for an Entry. Para1 = Mid, Para2 = pData
12	2006	INFO	DataMap: Requested Unsubscribe for DataMap Entry. Para1 = Mid, Para2 = pData
12	2007	INFO	DataMap: Subscription Confirmed, Granted QoS differs from requested. Para1 = mid, Para2 = Message mid, Para3 = Granted QoS, Para4 = Requested QoS
12	2008	INFO	DataMap: Unsubscribe for Entry Received. Para1 = iMid. Para2 = Message Mid.
12	2009	INFO	Connection: Connected. Para1 = Retcode, Para2 = Flags
12	2010	INFO	Connection: Disconnected. Para1 = RetCode
12	2011	INFO	Publish: Received Publish confirmation from the MQTTClient. Para1 = mid
12	2012	INFO	Unsubscribe: Received Unsubscribe confirmation from the

			MQTTClient. Para1 = mid
13	3000	DEBUG	ClientActive: Disconnect request Initiated
13	3001	DEBUG	Unsubscribe: Unsubscribe requested, however no callback to the Application is set. Para1=udApplMsgID, Para2=mid
13	3002	DEBUG	Unsubscribe: Unsubscribe requested for a topic where atleast one other party in the application is still subscribed, the Unsubscribe request will not be sent. Para1 = Requested Unsub ApplID, Para2 = ApplID of first subscriber that is still active.