

# UbiGEM API Guide

(C++)

V2.0



# 목차

Method List.....	9
Initialize, Start and Stop.....	9
Get State of Driver .....	9
Send Primary Message .....	9
Send Secondary Message .....	10
Create and Delete Instance .....	11
Set Variable and Equipment Constant Value .....	13
Subscribe Events .....	14
Event List.....	16
Connection and State Changed Events .....	16
Primary Message Received Events .....	16
Secondary Message Received Events.....	17
Variable Value Update Events .....	18
ETC Events .....	18
Invalid or Unknown Message Receive Events.....	18
Method .....	19
<i>Initialize, Start and Stop.</i> .....	19
void Terminate().....	20
GEMResult Start() .....	20
GEMResult Stop().....	21
Get State of Driver .....	22
bool GetConnected().....	22
GEMCommunicationState GetCommunicationState().....	22
GEMControlState GetControlState().....	23
Send Primary Message .....	24
GEMResult EstablishCommunication().....	24
GEMResult RequestAreYouThere() .....	24
GEMResult RequestOffline().....	25
GEMResult RequestOnlineLocal().....	25
GEMResult RequestOnlineRemote().....	26
GEMResult RequestHostOffline().....	26
GEMResult RequestLoopback(List<uint8_t> * abs) .....	27
GEMResult RequestDateTime().....	27

<i>GEMResult RequestPPLoadInquire(LPCTSTR strPPID, int length)</i> .....	28
<i>GEMResult RequestPPSend(LPCTSTR strPPID, LPCTSTR strFileName)</i> .....	28
<i>GEMResult RequestPPSend(LPCTSTR strPPID, List&lt;uint8_t&gt;* ppBody)</i> .....	29
<i>GEMResult RequestPPRequest(LPCTSTR strPPID) .....</i>	29
<i>GEMResult RequestPPChanged(uint64_t changeStatus, LPCTSTR strPPID)</i> .....	30
<i>GEMResult RequestFmtPPSend(FmtPPCollection* fmtPPCollection)</i> .....	30
<i>GEMResult RequestFmtPPSendWithoutValue(FmtPPCollection* fmtPPCollection)</i> .....	31
<i>GEMResult RequestFmtPPRequest(LPCTSTR strPPID)</i> .....	31
<i>GEMResult RequestFmtPPVerificationSend(FmtPPVerificationCollection* fmtPPVerificationCollection)</i> .....	32
<i>GEMResult RequestFmtPPChanged(uint64_t changeStatus, LPCTSTR strPPID)</i> .....	32
<i>GEMResult ReportEquipmentProcessingState(int equipmentProcessState)</i> .....	33
<i>GEMResult ReportCollectionEvent(LPCTSTR strCEID, bool bUseRaiseEvent = true)</i> .....	33
<i>GEMResult ReportCollectionEvent(CollectionEventInfo* collectionEventInfo)</i> .....	34
<i>GEMResult ReportAlarmSet(int64_t alarmID)</i> .....	34
<i>GEMResult ReportAlarmClear(int64_t alarmID)</i> .....	35
<i>GEMResult ReportTerminalMessage(int tid, LPCTSTR strText)</i> .....	35
<i>GEMResult ReportConversationTimeout(LPCTSTR strMEXP, LPCTSTR strEDID)</i> .....	36
<i>GEMResult SendSECSMessage(SECSMessage *message)</i> .....	36
<i>Send Secondary Message</i> .....	37
<i>GEMResult ReplyRemoteCommandAck(RemoteCommandInfo* rcmdInfo, RemoteCommandResult* rcmdResult)</i> .....	37
<i>GEMResult ReplyEnhancedRemoteCommandAck(EnhancedRemoteCommandInfo* ercCmdInfo, RemoteCommandResult* rcmdResult)</i> .....	37
<i>GEMResult ReplyRequestOfflineAck(uint32_t systemBytes, int ack)</i> .....	38
<i>GEMResult ReplyRequestOnlineAck(uint32_t systemBytes, int ack)</i> .....	38
<i>GEMResult ReplyNewEquipmentConstantSend(uint32_t systemBytes, VariableCollection* newVariableCollection, int ack)</i> .....	39
<i>GEMResult ReplyDateTimeRequest(uint32_t systemBytes, DateTime* timeData)</i> .....	39
<i>GEMResult ReplyDateTimeSetRequest(uint32_t systemBytes, int ack, DateTime* targetTime)</i> .....	40
<i>GEMResult ReplyPPLoadInquireAck(uint32_t systemBytes, int processProgramGrantStatus)</i> .....	40
<i>GEMResult ReplyPPSendAck(uint32_t systemBytes, int ack)</i> .....	41
<i>GEMResult ReplyPPRequestAck(uint32_t systemBytes, List&lt;uint8_t&gt;* ppbody, bool ack)</i> .....	41
<i>GEMResult ReplyPPDeleteAck(uint32_t systemBytes, int ack)</i> .....	42
<i>GEMResult ReplyCurrentEPPDRequestAck(uint32_t systemBytes, List&lt;LPCTSTR&gt;* strPPIDs, bool ack)</i> .....	42
<i>GEMResult ReplyFmtPSendAck(uint32_t systemBytes, int ack)</i> .....	43
<i>GEMResult ReplyFmtPPRequestAck(uint32_t systemBytes, LPCTSTR strPPID, FmtPPCollection* fmtPPCollection, bool ack)</i> .....	43
<i>GEMResult ReplyTerminalMessageAck(uint32_t systemBytes, int ack)</i> .....	44
<i>GEMResult ReplyTerminalMultiMessageAck(uint32_t systemBytes, int ack)</i> .....	44
<i>GEMResult ReplySECSMessage(uint32_t systemBytes, SECSMessage* primaryMessage, SECSMessage* secondaryMessage)</i> .....	45

<i>Create and Delete Instance</i> .....	46
<i>List&lt;bool&gt; * CreateBoolList()</i> .....	46
<i>void DeleteBoolList(List&lt;bool&gt; * p)</i> .....	46
<i>List&lt;int8_t&gt; * CreateI1List()</i> .....	47
<i>void DeleteI1List(List&lt;int8_t&gt; * p)</i> .....	47
<i>List&lt;int16_t&gt; * CreateI2List()</i> .....	48
<i>void DeleteI2List(List&lt;int16_t&gt; * p)</i> .....	48
<i>List&lt;int32_t&gt; * CreateI4List()</i> .....	49
<i>void DeleteI4List(List&lt;int32_t&gt; * p)</i> .....	49
<i>List&lt;int64_t&gt; * CreateI8List()</i> .....	50
<i>void DeleteI8List(List&lt;int64_t&gt; * p)</i> .....	50
<i>List&lt;uint8_t&gt; * CreateU1List()</i> .....	51
<i>void DeleteU1List(List&lt;uint8_t&gt; * p)</i> .....	51
<i>List&lt;uint16_t&gt; * CreateU2List()</i> .....	52
<i>void DeleteU2List(List&lt;uint16_t&gt; * p)</i> .....	52
<i>List&lt;uint32_t&gt; * CreateU4List()</i> .....	53
<i>void DeleteU4List(List&lt;uint32_t&gt; * p)</i> .....	53
<i>List&lt;uint64_t&gt; * CreateU8List()</i> .....	54
<i>void DeleteU8List(List&lt;uint64_t&gt; * p)</i> .....	54
<i>List&lt;float&gt; * CreateF4List()</i> .....	55
<i>void DeleteF4List(List&lt;float&gt; * p)</i> .....	55
<i>List&lt;double&gt; * CreateF8List()</i> .....	56
<i>void DeleteF8List(List&lt;double&gt; * p)</i> .....	56
<i>List&lt;LPCTSTR&gt; * CreateStringList()</i> .....	57
<i>void DeleteStringList(List&lt;LPCTSTR&gt; * p, bool bItemDelete = true)</i> .....	57
<i>VariableCollection* GetAllVariable(bool bRebuild = false)</i> .....	58
<i>VariableInfo* CreateVariable(LPCTSTR strVID, GEMSECSFormat format, LPCTSTR strName = nullptr, GEMVariableType vidType = UbiGEMWrapper::Structure::GEMVariable_SV, LPCTSTR strDescription = nullptr)</i> .....	58
<i>VariableInfo* GetVariable(LPCTSTR strVID)</i> .....	59
<i>void DeleteVariableInfo(VariableInfo *p)</i> .....	59
<i>CollectionEventCollection* GetAllCollectionEventInfo(bool bRebuild = false)</i> .....	60
<i>CollectionEventInfo* CreateCollectionEventInfo(LPCTSTR strCEID)</i> .....	60
<i>CollectionEventInfo* GetCollectionEventInfo(LPCTSTR strCEID)</i> .....	61
<i>void DeleteCollectionEventInfo(CollectionEventInfo *p)</i> .....	61
<i>ReportInfo* CreateReportInfo(LPCTSTR strRPTID)</i> .....	62
<i>void DeleteReportInfo(ReportInfo* p)</i> .....	62
<i>SECSMessageCollection* GetAllUserMessage(bool bRebuild = false)</i> .....	63

<i>SECSMessage* CreateUserMessage(LPCTSTR strName, int stream, int function, bool bWaitBit = true, bool bToHost = true)</i> .....	63
<i>SECSMessage* GetUserMessage(int stream, int function, bool bToHost = true) .....</i>	64
<i>void DeleteSECSMessage(SECSMessage* p) .....</i>	64
<i>FmtPPCollection* CreateFmtPPCollection(LPCTSTR strPPID, LPCTSTR strMDLN, LPCTSTR strSOFTREV).....</i>	65
<i>void DeleteFmtPPCollection(FmtPPCollection* p) .....</i>	65
<i>FmtPPCodeInfo* CreateFmtPPCodeInfo(LPCTSTR strCommandCode) .....</i>	66
<i>void DeleteFmtPPCodeInfo(FmtPPCodeInfo* p) .....</i>	66
<i>FmtPPItem* CreateFmtPPItem(LPCTSTR strPPName, LPCTSTR strPPValue, GEMSECSFormat format).....</i>	67
<i>void DeleteFmtPPItem(FmtPPItem *p) .....</i>	67
<i>FmtPPVerificationInfo* CreateFmtPPVerificationInfo(int nACK, int nSeqNum, LPCTSTR strErrW7) .....</i>	68
<i>void DeleteFmtPPVerificationInfo(FmtPPVerificationInfo* p) .....</i>	68
<i>FmtPPVerificationCollection* CreateFmtPPVerificationCollection(LPCTSTR strPPID) .....</i>	69
<i>void DeleteFmtPPVerificationCollection(FmtPPVerificationCollection *p) .....</i>	69
<i>RemoteCommandResult* CreateRemoteCommandResult(int nHostCommandAck = 0) .....</i>	70
<i>void DeleteRemoteCommandResult(RemoteCommandResult *p) .....</i>	70
<i>RemoteCommandParameterResult* CreateRemoteCommandParameterResult(LPCTSTR strCpName, int nParameterAck) .....</i>	71
<i>void DeleteRemoteCommandParameterResult(RemoteCommandParameterResult* p) .....</i>	71
<i>Set Variable and Equipment Constant Value.....</i>	72
<i>GEMResult SetVariable(LPCTSTR strVID, bool newValue) .....</i>	72
<i>GEMResult SetVariable(LPCTSTR strVID, uint8_t newValue) .....</i>	72
<i>GEMResult SetVariable(LPCTSTR strVID, uint16_t newValue) .....</i>	73
<i>GEMResult SetVariable(LPCTSTR strVID, uint32_t newValue) .....</i>	73
<i>GEMResult SetVariable(LPCTSTR strVID, uint64_t newValue) .....</i>	74
<i>GEMResult SetVariable(LPCTSTR strVID, int8_t newValue) .....</i>	74
<i>GEMResult SetVariable(LPCTSTR strVID, int16_t newValue) .....</i>	75
<i>GEMResult SetVariable(LPCTSTR strVID, int32_t newValue) .....</i>	75
<i>GEMResult SetVariable(LPCTSTR strVID, int64_t newValue) .....</i>	76
<i>GEMResult SetVariable(LPCTSTR strVID, float newValue) .....</i>	76
<i>GEMResult SetVariable(LPCTSTR strVID, double newValue) .....</i>	77
<i>GEMResult SetVariable(LPCTSTR strVID, LPCTSTR newValue) .....</i>	77
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;bool&gt;* arrayValue) .....</i>	78
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;uint8_t&gt;* arrayValue) .....</i>	78
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;uint16_t&gt;* arrayValue) .....</i>	79
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;uint32_t&gt;* arrayValue) .....</i>	79
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;uint64_t&gt;* arrayValue) .....</i>	80
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;int8_t&gt;* arrayValue) .....</i>	80

<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;int16_t&gt;* arrayValue)</i> .....	81
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;int32_t&gt;* arrayValue)</i> .....	81
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;int64_t&gt;* arrayValue)</i> .....	82
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;float&gt;* arrayValue)</i> .....	82
<i>GEMResult SetVariable(LPCTSTR strVID, List&lt;double&gt;* arrayValue)</i> .....	83
<i>GEMResult SetVariable(VariableInfo* variableInfo)</i> .....	83
<i>GEMResult SetVariable(VariableCollection* variableInfos)</i> .....	84
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, bool newValue)</i> .....	84
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, uint8_t newValue)</i> .....	85
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, uint16_t newValue)</i> .....	85
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, uint32_t newValue)</i> .....	86
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, uint64_t newValue)</i> .....	86
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, int8_t newValue)</i> .....	87
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, int16_t newValue)</i> .....	87
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, int32_t newValue)</i> .....	88
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, int64_t newValue)</i> .....	88
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, float newValue)</i> .....	89
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, double newValue)</i> .....	89
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, LPCTSTR newValue)</i> .....	90
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;bool&gt;* arrayValue)</i> .....	90
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;uint8_t&gt;* arrayValue)</i> .....	91
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;uint16_t&gt;* arrayValue)</i> .....	91
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;uint32_t&gt;* arrayValue)</i> .....	92
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;uint64_t&gt;* arrayValue)</i> .....	92
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;int8_t&gt;* arrayValue)</i> .....	93
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;int16_t&gt;* arrayValue)</i> .....	93
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;int32_t&gt;* arrayValue)</i> .....	94
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;int64_t&gt;* arrayValue)</i> .....	94
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;float&gt;* arrayValue)</i> .....	95
<i>GEMResult SetEquipmentConstant(LPCTSTR strECID, List&lt;double&gt;* arrayValue)</i> .....	95
<i>GEMResult SetEquipmentConstant(VariableInfo* variableInfo)</i> .....	96
<i>GEMResult SetEquipmentConstant(VariableCollection* variableInfos)</i> .....	96
<i>Subscribe Events</i> .....	97
<i>void SubscribeInitialize(IGEMDriverEvents* eventReceiver)</i> .....	97
<i>void SubscribeGEMConnected()</i> .....	97
<i>void SubscribeGEMDisconnected()</i> .....	98

<i>void SubscribeGEMSelected()</i> .....	98
<i>void SubscribeGEMDeselected()</i> .....	99
<i>void SubscribeCommunicationStateChanged()</i> .....	99
<i>void SubscribeControlStateChanged()</i> .....	100
<i>void SubscribeControlStateOnlineChangeFailed()</i> .....	100
<i>void SubscribeEquipmentProcessState()</i> .....	101
<i>void SubscribeSpoolStateChanged()</i> .....	101
<i>void SubscribeReceivedEstablishCommunicationsRequest()</i> .....	102
<i>void SubscribeReceivedRemoteCommand()</i> .....	102
<i>void SubscribeReceivedEnhancedRemoteCommand()</i> .....	103
<i>void SubscribeReceivedNewECVSend()</i> .....	103
<i>void SubscribeReceivedLoopback()</i> .....	104
<i>void SubscribeReceivedTerminalMessage()</i> .....	104
<i>void SubscribeReceivedTerminalMultiMessage()</i> .....	105
<i>void SubscribeReceivedRequestOffline()</i> .....	105
<i>void SubscribeReceivedRequestOnline()</i> .....	106
<i>void SubscribeReceivedDefineReport()</i> .....	106
<i>void SubscribeReceivedLinkEventReport()</i> .....	107
<i>void SubscribeReceivedEnableDisableEventReport()</i> .....	107
<i>void SubscribeReceivedEnableDisableAlarmSend()</i> .....	108
<i>void SubscribeReceivedPPLoadInquire()</i> .....	108
<i>void SubscribeReceivedPPSend()</i> .....	109
<i>void SubscribeReceivedFmtPPSend()</i> .....	109
<i>void SubscribeReceivedDateTimeRequest()</i> .....	110
<i>void SubscribeReceivedDateTimeSetRequest()</i> .....	110
<i>void SubscribeReceivedPPRequest()</i> .....	111
<i>void SubscribeReceivedFmtPPRequest()</i> .....	111
<i>void SubscribeReceivedDeletePPSend()</i> .....	112
<i>void SubscribeReceivedCurrentEPPDRequest()</i> .....	112
<i>void SubscribeUserPrimaryMessageReceived()</i> .....	113
<i>void SubscribeResponseDateTimeRequest()</i> .....	113
<i>void SubscribeResponseLoopback()</i> .....	114
<i>void SubscribeResponseEventReportAcknowledge()</i> .....	114
<i>void SubscribeResponsePPLoadInquire()</i> .....	115
<i>void SubscribeResponsePPSend()</i> .....	115
<i>void SubscribeResponsePPRequest()</i> .....	116

<i>void SubscribeResponseFmtPPSend()</i> .....	116
<i>void SubscribeResponseFmtPPRequest()</i> .....	117
<i>void SubscribeResponseFmtPPVerification()</i> .....	117
<i>void SubscribeResponseTerminalRequest()</i> .....	118
<i>void SubscribeUserSecondaryMessageReceived()</i> .....	118
<i>void SubscribeVariableUpdateRequest()</i> .....	119
<i>void SubscribeTraceDataUpdateRequest()</i> .....	119
<i>void SubscribeUserGEMMessageUpdateRequest()</i> .....	120
<i>void SubscribeInvalidMessageReceived()</i> .....	120
<i>void SubscribeReceivedUnknownMessage()</i> .....	121
<i>void SubscribeReceivedInvalidRemoteCommand()</i> .....	121
<i>void SubscribeReceivedInvalidEnhancedRemoteCommand()</i> .....	122
<b>Event.....</b>	123
<i>Connection and State Changed Events .....</i>	123
<i>void OnGEMConnected(LPCTSTR strIpAddress, int nPortNo)</i> .....	123
<i>void OnGEMDisconnected(LPCTSTR strIpAddress, int nPortNo)</i> .....	123
<i>void OnGEMSelected(LPCTSTR strIpAddress, int nPortNo)</i> .....	124
<i>void OnGEMDeselected(LPCTSTR strIpAddress, int nPortNo)</i> .....	124
<i>void OnCommunicationStateChanged(CommunicationState communicationState)</i> .....	125
<i>void OnControlStateChanged(ControlState controlState)</i> .....	125
<i>void OnControlStateOnlineChangeFailed()</i> .....	126
<i>void OnEquipmentProcessState(uint8_t equipmentProcessState)</i> .....	126
<i>void OnSpoolStateChanged(GEMSpoolState spoolState)</i> .....	127
<i>Primary Message Received Events .....</i>	128
<i>int OnReceivedEstablishCommunicationRequest(LPCTSTR strMDLN, LPCTSTR strSOFTREV)</i> .....	128
<i>void OnReceivedRemoteCommand(RemoteCommandInfo*remoteCommandInfo)</i> .....	128
<i>void OnReceivedEnhancedRemoteCommand(EnhancedRemoteCommandInfo* ercCmdInfo)</i> .....	129
<i>void OnReceivedNewECVSend(VariableCollection* newEcInfo)</i> .....	129
<i>void OnReceivedLoopback(List&lt;uint8_t&gt;* receiveData)</i> .....	130
<i>void OnReceivedTerminalMessage(uint32_t systemBytes, int tid, LPCTSTR strTerminalMessage)</i> .....	130
<i>void OnReceivedTerminalMultiMessage(uint32_t systemBytes, int tid, List&lt;LPCTSTR&gt;* strTerminalMessages)</i> .....	131
<i>void OnReceivedRequestOffline(uint32_t systemBytes)</i> .....	131
<i>void OnReceivedRequestOnline(uint32_t systemBytes)</i> .....	132
<i>void OnReceivedDefineReport()</i> .....	132
<i>void OnReceivedLinkEventReport()</i> .....	133
<i>void OnReceivedEnableDisableEventReport()</i> .....	133

<i>void OnReceivedEnableDisableAlarmSend()</i> .....	134
<i>void OnReceivedPPLoadInquire(uint32_t systemBytes, LPCTSTR strPPID, int length)</i> .....	134
<i>void OnReceivedPPSend(uint32_t systemBytes, LPCTSTR strPPID, List&lt;uint8_t*&gt;* ppbody)</i> .....	135
<i>void OnReceivedFmtPPSend(uint32_t systemBytes, FmtPPCollection* fmtPPCollection)</i> .....	135
<i>void OnReceivedDateTimeRequest(uint32_t systemBytes, DateTime* timeData)</i> .....	136
<i>void OnReceivedDateTimeSetRequest(uint32_t systemBytes, DateTime* timeData)</i> .....	136
<i>void OnReceivedPPRequest(uint32_t systemBytes, LPCTSTR strPPID)</i> .....	137
<i>void OnReceivedFmtPPRequest(uint32_t systemBytes, LPCTSTR strPPID)</i> .....	137
<i>void OnReceivedDeletePPSend(uint32_t systemBytes, List&lt;LPCTSTR&gt;* ppids)</i> .....	138
<i>void OnReceivedCurrentEPPDRequest(uint32_t systemBytes)</i> .....	138
<i>void OnUserPrimaryMessageReceived(SECSMessage *message)</i> .....	139
<i>Secondary Message Received Events</i> .....	140
<i>bool OnResponseDateTimeRequest(DateTime* timeData)</i> .....	140
<i>void OnResponseLoopback(List&lt;uint8_t*&gt;* receiveData, List&lt;uint8_t*&gt;* sendData)</i> .....	140
<i>void OnResponseEventReportAcknowledge(LPCTSTR strCEID, int ack)</i> .....	141
<i>void OnResponsePPLoadInquire(int ppgnt, LPCTSTR strPPID)</i> .....	141
<i>void OnResponsePPSend(int ack, LPCTSTR strPPID)</i> .....	142
<i>void OnResponsePPRequest(LPCTSTR strPPID, List&lt;uint8_t*&gt;* ppbody)</i> .....	142
<i>void OnRequestFmtPPSend(int ack, FmtPPCollection* fmtPPCollection)</i> .....	143
<i>void OnResponseFmtPPRequest(FmtPPCollection *fmtPPCollection)</i> .....	143
<i>void OnResponseFmtPPVerification(FmtPPVerificationCollection *fmtPPVerificationCollection)</i> .....	144
<i>void OnResponseTerminalRequest(int ack)</i> .....	144
<i>void OnUserSecondaryMessageReceived(SECSMessage* primaryMessage, SECSMessage* secondaryMessage)</i> .....	145
<i>Variable Value Update Events</i> .....	146
<i>void OnVariableUpdateRequest(GEMVariableUpdateType updateType, List&lt;LPCTSTR&gt;* strVIDs)</i> .....	146
<i>void OnTraceDataUpdateRequest(List&lt;LPCTSTR&gt;* strVIDs)</i> .....	147
<i>ETC Events</i> .....	148
<i>void OnUserGEMMessageUpdateRequest(SECSMessage* message)</i> .....	148
<i>Invalid or Unknown Message Receive Events</i> .....	149
<i>void OnInvalidMessageReceived(GEMMessageValidationError error, SECSMessage* message)</i> .....	149
<i>void OnReceivedUnknownMessage(SECSMessage* message)</i> .....	149
<i>void OnReceivedInvalidRemoteCommand(RemoteCommandInfo* remoteCommandInfo)</i> .....	150
<i>void OnReceivedInvalidEnhancedRemoteCommand(EnhancedRemoteCommandInfo* ercmdInfo)</i> .....	150
<i>Error Code</i> .....	151
<i>GEMResult</i> .....	151

## Method List

### Initialize, Start and Stop

Return Value	Name	Parameter
GEMResult	Initialize	LPCTSTR strUgcFileName
LPCTSTR	GetInitializeError	
void	Terminate	
GEMResult	Start	
GEMResult	Stop	

### Get State of Driver

Return Value	Name	Parameter
bool	GetConnected	
GEMCommunicationState	GetCommunicationState	
GEMControlState	GetControlState	

### Send Primary Message

Return Value	Name	Parameter
GEMResult	EstablishCommunication	
GEMResult	RequestAreYouThere	
GEMResult	RequestOffline	
GEMResult	RequestOnlineLocal	
GEMResult	RequestOnlineRemote	
GEMResult	RequestHostOffline	
GEMResult	RequestLoopback	List<uint8_t>* abs
GEMResult	RequestDateTime	
GEMResult	RequestPPLoadInquire	LPCTSTR strPpid
		int length
GEMResult	RequestPPSend	LPCTSTR strPpid
		LPCTSTR strFileName
GEMResult	RequestPPSend	LPCTSTR strPpid
		List<uint8_t>* ppbody
GEMResult	RequestPPRequest	LPCTSTR strPpid
GEMResult	RequestPPChanged	int64_t changeStatus
		LPCTSTR strPpid
GEMResult	RequestFmtPPSend	FmtPPCollection* fmtPPCollection
GEMResult	RequestFmtPPSendWithoutValue	FmtPPCollection* fmtPPCollection

GEMResult	RequestFmtPPRequest	LPCTSTR strPpid
GEMResult	RequestFmtPPVerificationSend	FmtPPVerificationCollection* fmtPPVerificationCollection
GEMResult	RequestFmtPPChanged	int64_t changeStatus
		LPCTSTR strPpid
GEMResult	ReportEquipmentProcessingState	int equipmentProcessState
GEMResult	ReportCollectionEvent	LPCTSTR strCEID
		[bool bUseRaiseEvent]
GEMResult	ReportCollectionEvent	CollectionEventInfo* collectionEventInfo
GEMResult	ReportAlarmSet	int64_t alarmId
GEMResult	ReportAlarmClear	int64_t alarmId
GEMResult	ReportTerminalMessage	int tid
		LPCTSTR strText
GEMResult	ReportConversationTimeout	LPCTSTR strMexp
		LPCTSTR strEdid
GEMResult	SendSECSMessage	SECSMessage* message

### Send Secondary Message

Return Value	Name	Parameter
GEMResult	ReplyRemoteCommandAck	RemoteCommandInfo* remoteCommandInfo
		RemoteCommandResult* remoteCommandResult
GEMResult	ReplyEnhancedRemoteCommandAck	EnhancedRemoteCommandInfo* remoteCommandInfo
		RemoteCommandResult* remoteCommandResult
GEMResult	ReplyRequestOfflineAck	uint32_t systemBytes
		int ack
GEMResult	ReplyRequestOnlineAck	uint32_t systemBytes
		int ack
GEMResult	ReplyNewEquipmentConstantSend	uint32_t systemBytes
		VariableCollection* newVariableCollection
		int ack
GEMResult	ReplyDateTimeRequest	uint32_t systemBytes
		DateTime* timeData
GEMResult	ReplyDateTimeSetRequest	uint32_t systemBytes
		int ack
		DateTime* targetTime
GEMResult	ReplyPPLoadInquireAck	uint32_t systemBytes
		int processProgramGrantStatus
GEMResult	ReplyPPSendAck	uint32_t systemBytes
		int processProgramGrantStatus
GEMResult	ReplyPPRequestAck	uint32_t systemBytes

		LPCTSTR strPpid
		List<uint8_t>* ppbody,
		bool ack
GEMResult	ReplyPPDeleteAck	uint32_t systemBytes
		int ack
GEMResult	ReplyCurrentEPPDRequestAck	uint32_t systemBytes,
		List<LPCTSTR>* strPpids, bool ack
GEMResult	ReplyFmtPPSendAck	uint32_t systemBytes
		int ack
GEMResult	ReplyFmtPPRequestAck	uint32_t systemBytes
		LPCTSTR strPpid
		FmtPPCollection* fmtPPCollection
		bool ack
GEMResult	ReplyTerminalMessageAck	uint32_t systemBytes
		int ack
GEMResult	ReplyTerminalMultiMessageAck	uint32_t systemBytes
		int ack
GEMResult	ReplySECSMessage	SECSMessage* primaryMessage
		SECSMessage* secondaryMessage

### Create and Delete Instance

Return Value	Name	Parameter
List<bool>*	CreateBoolList	
void	DeleteBoolList	List<bool>* p
List<int8_t>*	CreateI1List	
void	DeleteI1List	List<int8_t>* p
List<int16_t>*	CreateI2List	
void	DeleteI2List	List<int16_t>* p
List<int32_t>*	CreateI4List	
void	DeleteI4List	List<int32_t>* p
List<int64_t>*	CreateI8List	
void	DeleteI8List	List<int64_t>* p
List<uint8_t>*	CreateU1List	
void	DeleteU1List	List<uint8_t>* p
List<uint16_t>*	CreateU2List	
void	DeleteU2List	List<uint16_t>* p
List<uint32_t>*	CreateU4List	
void	DeleteU4List	List<uint32_t>* p
List<uint64_t>*	CreateU8List	

void	DeleteU8List	List<uint64_t>* p
List<float>*	CreateF4List	
void	DeleteF4List	List<float>* p
List<double>*	CreateF8List	
void	DeleteF8List	List<double>* p
List<LPCTSTR>*	CreateStringList	
void	DeleteStringList	List<LPCTSTR>* p
VariableCollection*	GetAllVariable	Optinal bool bRebuild
VariableInfo*	CreateVariable	LPCTSTR strVID
		GEMSECSFormat format
		Optinal LPCTSTR strName
		Optinal GEMVariableType vidType
		Optinal LPCTSTR strDescription
VariableInfo*	GetVariable	LPCTSTR strVid
void	DeleteVariableInfo	VariableInfo* p
CollectionEventCollection*	GetAllCollectionEventInfo	Optinal bool bRebuild
CollectionEventInfo*	CreateCollectionEventInfo	LPCTSTR strCEID
CollectionEventInfo*	GetCollectionEventInfo	LPCTSTR strCEID
ReportInfo*	CreateReportInfo	LPCTSTR strRptid
void	DeleteReportInfo	ReportInfo* p
SECSMessageCollection*	GetAllUserMessage	Optinal bool bRebuild
SECSMessage*	CreateUserMessage	LPCTSTR strName
		int stream
		int function
		Optinal bool bWaitBit
		Optinal bool bToHost
SECSMessage*	GetUserMessage	int stream
		int function
		Optinal bool bToHost
void	DeleteSECSMessage	SECSMessage* p
FmtPPCollection*	CreateFmtPPCollection	LPCTSTR strPPID,
		Optinal LPCTSTR strMDLN
		Optinal LPCTSTR strSOFTREV
void	DeleteFmtPPCollection	FmtPPCollection* p
FmtPPCCodeInfo*	CreateFmtPPCCodeInfo	LPCTSTR strCommandCode
void	DeleteFmtPPCCodeInfo	FmtPPCCodeInfo* p
FmtPPIItem*	CreateFmtPPIItem	LPCTSTR strPPName
		LPCTSTR strPPValue
		GEMSECSFormat eFormat
void	DeleteFmtPPIItem	FmtPPIItem* p

FmtPPVerificationInfo*	CreateFmtPPVerificationInfo	int nACK int nSeqNum LPCTSTR strErrW7
void	DeleteFmtPPVerificationInfo	FmtPPVerificationInfo* p
FmtPPVerificationCollection*	CreateFmtPPVerificationCollection	LPCTSTR strPPID
void	DeleteFmtPPVerificationCollection	FmtPPVerificationCollection* p
RemoteCommandResult*	CreateRemoteCommandResult	Optional int nHostCommandAck
void	DeleteRemoteCommandResult	RemoteCommandResult* p
RemoteCommandParameterResult*	CreateRemoteCommandParameterResult	LPCTSTR strCpName
		int bParameterAck
void	DeleteRemoteCommandParameterResult	RemoteCommandParameterResult* p

### Set Variable and Equipment Constant Value

Return Value	Name	Parameter
GEMResult	SetVariable	LPCTSTR strVid PrimitiveType newValue
		PrimitiveType: bool, uint8_t, uint16_t, uint32_t, uint64_t, int8_t, int16_t, int32_t, int64_t, float, double
GEMResult	SetVariable	LPCTSTR strVid LPCTSTR newValue
GEMResult	SetVariable	LPCTSTR strVid List<PrimitiveType>* arrayValue
		PrimitiveType: bool, uint8_t, uint16_t, uint32_t, uint64_t, int8_t, int16_t, int32_t, int64_t, float, double
GEMResult	SetVariable	VariableInfo* variableInfo
GEMResult	SetVariable	VariableCollection* variableInfos
GEMResult	SetEquipmentConstant	LPCTSTR strEcid PrimitiveType newValue
		PrimitiveType: bool, uint8_t, uint16_t, uint32_t, uint64_t, int8_t, int16_t, int32_t, int64_t, float, double
GEMResult	SetEquipmentConstant	LPCTSTR strEcid LPCTSTR newValue
GEMResult	SetEquipmentConstant	LPCTSTR strEcid List<PrimitiveType>* arrayValue
		PrimitiveType: bool, uint8_t, uint16_t, uint32_t, uint64_t, int8_t, int16_t, int32_t, int64_t, float, double
GEMResult	SetEquipmentConstant	VariableInfo* variableInfo
GEMResult	SetEquipmentConstant	VariableCollection* variableInfos

## Subscribe Events

Return Value	Name	Parameter
void	SubscribeInitilaize	IGEMDriverEvents* eventReceiver
void	SubscribeGEMConnected	
void	SubscribeGEMDisconnected	
void	SubscribeGEMSelected	
void	SubscribeGEMDeselected	
void	SubscribeCommunicationStateChanged	
void	SubscribeControlStateChanged	
void	SubscribeControlStateOnlineChangeFailed	
void	SubscribeEquipmentProcessState	
void	SubscribeSpoolStateChanged	
void	SubscribeReceivedEstablishCommunicationsRequest	
void	SubscribeReceivedRemoteCommand	
void	SubscribeReceivedEnhancedRemoteCommand	
void	SubscribeReceivedNewECVSend	
void	SubscribeReceivedLoopback	
void	SubscribeReceivedTerminalMessage	
void	SubscribeReceivedTerminalMultiMessage	
void	SubscribeReceivedRequestOffline	
void	SubscribeReceivedRequestOnline	
void	SubscribeReceivedDefineReport	
void	SubscribeReceivedLinkEventReport	
void	SubscribeReceivedEnableDisableEventReport	
void	SubscribeReceivedEnableDisableAlarmSend	
void	SubscribeReceivedPPLoadInquire	
void	SubscribeReceivedPPSend	
void	SubscribeReceivedFmtPPSend	
void	SubscribeReceivedDateTimeRequest	
void	SubscribeReceivedDateTimeSetRequest	
void	SubscribeReceivedPPRequest	
void	SubscribeReceivedFmtPPRequest	
void	SubscribeReceivedDeletePPSend	
void	SubscribeReceivedCurrentEPPDRequest	
void	SubscribeUserPrimaryMessageReceived	
void	SubscribeResponseDateTimeRequest	
void	SubscribeResponseLoopback	
void	SubscribeResponseEventReportAcknowledge	
void	SubscribeResponsePPLoadInquire	
void	SubscribeResponsePPSend	

void	SubscribeResponsePPRequest	
void	SubscribeResponseFmtPPSend	
void	SubscribeResponseFmtPPRequest	
void	SubscribeResponseFmtPPVerification	
void	SubscribeResponseTerminalRequest	
void	SubscribeUserSecondaryMessageReceived	
void	SubscribeVariableUpdateRequest	
void	SubscribeTraceDataUpdateRequest	
void	SubscribeUserGEMMessageUpdateRequest	
void	SubscribeInvalidMessageReceived	
void	SubscribeReceivedUnknownMessage	
void	SubscribeReceivedInvalidRemoteCommand	
void	SubscribeReceivedInvalidEnhancedRemoteCommand	

## Event List

### Connection and State Changed Events

Return Value	Name	Parameter
void	OnGEMConnected	LPCTSTR strIpAddress
		int nPortNo
void	OnGEMDisconnected	LPCTSTR strIpAddress
		int nPortNo
void	OnGEMSelected	LPCTSTR strIpAddress
		int nPortNo
void	OnGEMDeselected	LPCTSTR strIpAddress
		int nPortNo
void	OnCommunicationStateChanged	GEMCommunicationState communicationState
void	OnControlStateChanged	GEMControlState controlState
void	OnControlStateOnlineChangeFailed	
void	OnEquipmentProcessState	equipmentProcessState
void	OnSpoolStateChanged	GEMSpoolState spoolState

### Primary Message Received Events

Return Value	Name	Parameter
int	OnReceivedEstablishCommunicationsRequest	LPCTSTR strMDLN
		LPCTSTR strSOFTREV
void	OnReceivedRemoteCommand	RemoteCommandInfo* remoteCommandInfo
void	OnReceivedEnhancedRemoteCommand	EnhancedRemoteCommandInfo* erCommandInfo
void	OnReceivedNewECVSend	VariableCollection* newEcInfo
void	OnReceivedLoopback	List<uint8_t>* receiveData
void	OnReceivedTerminalMessage	uint32_t systemBytes
		int tid
		LPCTSTR strTerminalMessage
void	OnReceivedTerminalMultiMessage	uint32_t systemBytes
		int tid
		List<LPCTSTR>* strTerminalMessages
void	OnReceivedRequestOffline	uint32_t systemBytes
void	OnReceivedRequestOnline	uint32_t systemBytes
void	OnReceivedDefineReport	
void	OnReceivedLinkEventReport	
void	OnReceivedEnableDisableEventReport	
void	OnReceivedEnableDisableAlarmSend	

void	OnReceivedPPLoadInquire	uint32_t systemBytes LPCTSTR strPPID, int length
void	OnReceivedPPSend	uint32_t systemBytes LPCTSTR strPPID List<uint8_t>* ppbody
void	OnReceivedFmtPPSend	uint32_t systemBytes FmtPPCollection* fmtPPCollection
void	OnReceivedDateTimeRequest	uint32_t systemBytes DateTime* timeData
void	OnReceivedDateTimeSetRequest	uint32_t systemBytes DateTime* timeData
void	OnReceivedPPRequest	uint32_t systemBytes LPCTSTR strPPID
void	OnReceivedFmtPPRequest	uint32_t systemBytes LPCTSTR strPPID
void	OnReceivedDeletePPSend	uint32_t systemBytes List<LPCTSTR>* ppids
void	OnReceivedCurrentEPPDRequest	uint32_t systemBytes
void	OnUserPrimaryMessageReceived	SECSMessage* message

### Secondary Message Received Events

Return Value	Name	Parameter
bool	OnResponseDateTimeRequest	DateTime* timeData
void	OnResponseLoopback	List<uint8_t>* receiveData List<uint8_t>* sendData
void	OnResponseEventReportAcknowledge	LPCTSTR strCEID int ack
void	OnResponsePPLoadInquire	int ppgnt LPCTSTR strPPID
void	OnResponsePPSend	int ack LPCTSTR strPPID
void	OnResponsePPRequest	LPCTSTR strPPID List<uint8_t>* ppbody
void	OnResponseFmtPPSend	int ack FmtPPCollection* fmtPPCollection
void	OnResponseFmtPPRequest	FmtPPCollection* fmtPPCollection
void	OnResponseFmtPPVerification	FmtPPVerificationCollection* fmtPPVerificationCollection
void	OnResponseTerminalRequest	int ack

void	OnUserSecondaryMessageReceived	SECSMessage* primaryMessage
		SECSMessage* secondaryMessage

#### Variable Value Update Events

Return Value	Name	Parameter
void	OnVariableUpdateRequest	GEMVariableUpdateType updateType
		List<LPCTSTR>* variables
void	OnTraceDataUpdateRequest	List<LPCTSTR>* variables

#### ETC Events

Return Value	Name	Parameter
void	OnUserGEMMessageUpdateRequest	SECSMessage* message

#### Invalid or Unknown Message Receive Events

Return Value	Name	Parameter
void	OnInvalidMessageReceived	GEMMessageValidationError error
		SECSMessage* message
void	OnReceivedUnknownMessage	SECSMessage* message
void	OnReceivedInvalidRemoteCommand	RemoteCommandInfo* remoteCommandInfo
void	OnReceivedInvalidEnhancedRemoteCommand	EnhancedRemoteCommandInfo* erCommandInfo

## Method

### *Initialize, Start and Stop*

**GEMResult Initialize(LPCTSTR strUgcFileName)**

#### *Description*

GEM Driver를 초기화합니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strUgcFileName	GEM Driver 환경 설정 파일입니다.

#### *Return Value*

Type	Description
GEMResult	GEM Driver 초기화 결과입니다.

**LPCTSTR GetInitializeError()**

#### *Description*

Initialize가 OK가 아닐 경우 상세한 사유를 가져옵니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
LPCTSTR	GEM Driver 초기화 실패 사유입니다.

***void Terminate()***

*Description*

GEM Driver를 종료합니다.

*Parameters*

(none)

*Return Value*

(none)

***GEMResult Start()***

*Description*

GEM Driver를 시작합니다.

*Parameters*

(none)

*Return Value*

Type	Description
GEMResult	GEM Driver 시작 결과입니다.

## **GEMResult Stop()**

### *Description*

GEM Driver를 중지합니다.

### *Parameters*

(none)

### *Return Value*

Type	Description
GEMResult	GEM Driver 중지 결과입니다.

### ***Get State of Driver***

***bool GetConnected()***

#### *Description*

GEM Driver의 통신 연결 상태를 가져옵니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
bool	현재 통신 연결 상태입니다.

### ***GEMCommunicationState GetCommunicationState()***

#### *Description*

GEM Driver의 Communication 상태를 가져옵니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
GEMCommunicationState	현재 Communication 상태입니다.

***GEMControlState GetControlState()***

*Description*

GEM Driver의 Control 상태를 가져옵니다.

*Parameters*

(none)

*Return Value*

Type	Description
GEMControlState	현재 Control 상태입니다.

## ***Send Primary Message***

### ***GEMResult EstablishCommunication()***

#### *Description*

S1F13(Establish Communication Request)를 송신하여 Communication 상태를 갱신합니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
GEMResult	Establish Communication 송신 결과입니다.

### ***GEMResult RequestAreYouThere()***

#### *Description*

S1F1(Are You There Request)를 송신합니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
GEMResult	Are You There 송신 결과입니다.

### **GEMResult RequestOffline()**

#### *Description*

Control 상태를 Offline으로 변경합니다. 현재 상태에 따라 S6F11(Event Report Send)를 송신합니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
GEMResult	Control 상태 변경 결과입니다.

### **GEMResult RequestOnlineLocal()**

#### *Description*

Control 상태를 Online – Local로 변경합니다. 현재 상태에 따라 S6F11(Event Report Send)를 송신합니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
GEMResult	Control 상태 변경 결과입니다.

### **GEMResult RequestOnlineRemote()**

#### *Description*

Control 상태를 Online – Remote로 변경합니다. 현재 상태에 따라 S6F11(Event Report Send)를 송신합니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
GEMResult	Control 상태 변경 결과입니다.

### **GEMResult RequestHostOffline()**

#### *Description*

Control 상태를 HostOffline으로 변경합니다. 현재 상태에 따라 S6F11(Event Report Send)를 송신합니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
GEMResult	Control 상태 변경 결과입니다.

**GEMResult RequestLoopback(List<uint8\_t>\* abs)**

*Description*

S2F25(Loopback Diagnostic Request)를 송신합니다.

*Parameters*

Type	Name	Description
List<uint8_t>*	abs	ABS 항목에 사용할 데이터입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult RequestDateTime()**

*Description*

S2F17(Date and Time Request)를 송신합니다.

*Parameters*

(none)

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult RequestPPLoadInquire(LPCTSTR strPPID, int length)**

#### *Description*

S7F1(Process Program Load Inquire)를 송신합니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strPPID	Host 로 송신할 PPID 입니다.
int	length	PPBody 의 length 입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult RequestPPSend(LPCTSTR strPPID, LPCTSTR strFileName)**

#### *Description*

strFileName 파일의 내용으로 S7F3(Process Program Send)를 송신합니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strPPID	Host 로 송신할 PPID 입니다.
LPCTSTR	strFileName	파일 경로입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult RequestPPSend(LPCTSTR strPPID, List<uint8\_t> \* ppBody)**

#### *Description*

ppBody의 내용으로 S7F3(Process Program Send)를 송신합니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strPPID	Host로 송신할 PPID입니다.
List<uint8_t> *	ppBody	ppBody입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult RequestPPRequest(LPCTSTR strPPID)**

#### *Description*

S7F5(Process Program Request)를 송신합니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strPPID	Host로 송신할 PPID입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult RequestPPChanged(uint64\_t changeStatus, LPCTSTR strPPID)**

*Description*

S6F11(Event Report Send) – Process Program Changed 이벤트를 보고합니다.

*Parameters*

Type	Name	Description
uint64_t	changeStatus	변경 상태입니다.
LPCTSTR	strPPID	Host로 송신할 PPID입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult RequestFmtPPSend(FmtPPCollection\* fmtPPCollection)**

*Description*

S7F23(Formatted Process Program Send)를 송신합니다.

*Parameters*

Type	Name	Description
FmtPPCollection*	fmtPPCollection	Host로 송신할 Formatted Process Program입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult RequestFmtPPSendWithoutValue(FmtPPCollection\* fmtPPCollection)**

#### *Description*

S7F23(Formatted Process Program Send)를 Name Only로 송신합니다.

#### *Parameters*

Type	Name	Description
FmtPPCollection*	fmtPPCollection	Host로 송신할 Formatted Process Program입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult RequestFmtPPRequest(LPCTSTR strPPID)**

#### *Description*

S7F25(Formatted Process Program Request)를 송신합니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strPPID	Host로 송신할 PPID입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult RequestFmtPPVerificationSend(FmtPPVerificationCollection\***  
**fmtPPVerificationCollection)**

*Description*

S7F27(Process Program Verification Send) 를 송신합니다.

*Parameters*

Type	Name	Description
FmtPPVerificationCollection*	fmtPPVerificationCollection	Host 로 송신할 Formatted Process Program Verification 데이터입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult RequestFmtPPChanged(uint64\_t changeStatus, LPCTSTR strPPID)**

*Description*

S6F11(Event Report Send) – Process Program Changed 이벤트를 보고합니다.

*Parameters*

Type	Name	Description
uint64_t	changeStatus	변경 상태입니다.
LPCTSTR	strPPID	Host 로 송신할 PPID 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReportEquipmentProcessingState(int equipmentProcessState)**

#### *Description*

S6F11(Event Report Send) – Process State Changed 이벤트를 보고합니다.

#### *Parameters*

Type	Name	Description
int	equipmentProcessState	설비 Process 상태입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReportCollectionEvent(LPCTSTR strCEID, bool bUseRaiseEvent = true)**

#### *Description*

S6F11(Event Report Send) – strCEID에 해당하는 이벤트를 보고합니다.

bUseRaiseEvent가 true인 경우 OnVariableUpdateRequest Callback을 발생시키며, Callback Handler에서 Variable의 Value를 Update할 수 있습니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strCEID	보고할 Collection Event ID입니다.
[Optional] bool	bUseRaiseEvent	OnVariableUpdateRequest Callback 발생 여부입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReportCollectionEvent(CollectionEventInfo\* collectionEventInfo)**

#### *Description*

S6F11(Event Report Send) – collectionEventInfo에 해당하는 이벤트를 보고합니다.

OnVariableUpdateRequest Callback이 발생하지 않습니다.

#### *Parameters*

Type	Name	Description
CollectionEventInfo*	collectionEventInfo	보고할 Collection Event 개체입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReportAlarmSet(int64\_t alarmID)**

#### *Description*

alarmID에 해당하는 알람이 발생했음을 보고합니다.

UGC 설정에 따라 S5F1(Alarm Report Send) 또는 S6F11(Event Report Send)를 송신합니다.

#### *Parameters*

Type	Name	Description
int64_t	alarmID	보고할 Alarm ID입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReportAlarmClear(int64\_t alarmID)**

#### *Description*

alarmID에 해당하는 알람이 해제되었음을 보고합니다.

UGC 설정에 따라 S5F1(Alarm Report Send) 또는 S6F11(Event Report Send)를 송신합니다.

#### *Parameters*

Type	Name	Description
int64_t	alarmID	보고할 Alarm ID입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReportTerminalMessage(int tid, LPCTSTR strText)**

#### *Description*

S10F1(Terminal Request)를 송신합니다.

#### *Parameters*

Type	Name	Description
int	tid	Terminal ID입니다.
LPCTSTR	strText	Terminal Text입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReportConversationTimeout(LPCTSTR strMEXP, LPCTSTR strEDID)**

*Description*

S9F13(Conversation Timeout)를 송신합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strMEXP	Conversation Timeout 이 발생한 SxFy입니다.
LPCTSTR	strEDID	Conversation Timeout 이 발생한 데이터의 ID입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SendSECSMessage(SECSMessage \*message)**

*Description*

사용자 정의 메시지를 전송합니다.

*Parameters*

Type	Name	Description
SECSMessage*	message	전송할 메시지입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### ***Send Secondary Message***

***GEMResult ReplyRemoteCommandAck(RemoteCommandInfo\* rcmdInfo, RemoteCommandResult\* rcmdResult)***

#### *Description*

S2F41(Host Command Send) 수신 후 S2F42(Host Command Acknowledge)를 송신합니다.

#### *Parameters*

Type	Name	Description
RemoteCommandInfo*	rcmdInfo	수신한 Host Command입니다.
RemoteCommandResult*	rcmdResult	응답할 결과입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

***GEMResult ReplyEnhancedRemoteCommandAck(EnhancedRemoteCommandInfo\* ercmdInfo, RemoteCommandResult\* rcmdResult)***

#### *Description*

S2F49(Enhanced Remote Command) 수신 후 S2F50(Enhanced Remote Command Acknowledge)를 송신합니다.

#### *Parameters*

Type	Name	Description
EnhancedRemoteCommandInfo*	ercmdInfo	수신한 Enhanced Remote Command입니다.
RemoteCommandResult*	rcmdResult	응답할 결과입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReplyRequestOfflineAck(uint32\_t systemBytes, int ack)**

#### *Description*

S1F15(Request OFF-LINE) 수신 후 S1F16(OFF-LINE Acknowledge)를 송신합니다.

ack가 0인 경우 Control 상태가 HostOffline로 변경됩니다.

#### *Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	ack	응답할 ack 입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReplyRequestOnlineAck(uint32\_t systemBytes, int ack)**

#### *Description*

S1F17(Request ON-LINE) 수신 후 S1F18(OFF-LINE Acknowledge)를 송신합니다.

ack가 0인 경우 Control 상태가 OnlineRemote로 변경됩니다.

#### *Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	ack	응답할 ack 입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyNewEquipmentConstantSend(uint32\_t systemBytes, VariableCollection\* newVariableCollection, int ack)**

*Description*

S2F15(New Equipment Constant Send) 수신 후 S2F16(New Equipment Constant Acknowledge)를 송신합니다.

ack가 0인 경우 newVariableCollection의 내용을 토대로 S6F11(Event Report Send) – Equipment Constant Changed를 보고합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
VariableCollection*	newVariableCollection	EquipmentConstantChanged 로 보고할 데이터입니다.
int	ack	응답할 ack 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyDateTimeRequest(uint32\_t systemBytes, DateTime\* timeData)**

*Description*

S2F17(Date and Time Request) 수신 후 S2F18(Date and Time Data)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
DateTime*	timeData	현재 시각입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyDateTimeSetRequest(uint32\_t systemBytes, int ack, DateTime\* targetTime)**

*Description*

S2F31(Date and Time Set Request) 수신 후 S2F32(Date and Time Set Acknowledge)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	ack	응답할 ack 입니다.
DateTime*	targetTime	Host 에서 수신한 DateTime 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyPPLoadInquireAck(uint32\_t systemBytes, int processProgramGrantStatus)**

*Description*

S7F1(Process Program Load Inquire) 수신 후 S7F2(Process Program Load Grant)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	processProgramGrantStatus	응답할 상태입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyPPSendAck(uint32\_t systemBytes, int ack)**

*Description*

S7F3(Process Program Send) 수신 후 S7F4(Process Program Acknowledge)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	ack	응답할 ack 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyPPRequestAck(uint32\_t systemBytes, List<uint8\_t>\* ppbody, bool ack)**

*Description*

S7F5(Process Program Request) 수신 후 S7F6(Process Program Data)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
List<uint8_t>*	ppbody	응답할 ppbody 입니다.
bool	ack	응답할 ack 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyPPDeleteAck(uint32\_t systemBytes, int ack)**

*Description*

S7F17(Delete Process Program Send) 수신 후 S7F18(Delete Process Program Acknowledge)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	ack	응답할 ack 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyCurrentEPPDRequestAck(uint32\_t systemBytes, List<LPCTSTR>\* strPPIDs, bool ack)**

*Description*

S7F19(Current EPPD Request) 수신 후 S7F20(Current EPPD Data)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
List<LPCTSTR>*	strPPIDs	응답할 PPID 목록입니다.
bool	ack	응답할 ack 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyFmtPSendAck(uint32\_t systemBytes, int ack)**

*Description*

S7F23(Formatted Process Program Send) 수신 후 S7F24(Formatted Process Program Acknowledge)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	ack	응답할 ack 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplyFmtPPRequestAck(uint32\_t systemBytes, LPCTSTR strPPID, FmtPPCollection\* fmtPPCollection, bool ack)**

*Description*

S7F25(Formatted Process Program Request) 수신 후 S7F26(Formatted Process Program Data)를 송신합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
LPCTSTR	strPPID	응답할 PPID 입니다.
FmtPPCollection*	fmtPPCollection	응답할 Formatted Process Program 입니다.
bool	ack	응답할 ack 입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReplyTerminalMessageAck(uint32\_t systemBytes, int ack)**

#### *Description*

S10F3(Terminal Display, Single) 수신 후 S10F4(Terminal Display, Single Acknowledge)를 송신합니다.

#### *Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	ack	응답할 ack 입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult ReplyTerminalMultiMessageAck(uint32\_t systemBytes, int ack)**

#### *Description*

S10F5(Terminal Display, Multi-Block) 수신 후 S10F6(Terminal Display, Multi-Block Acknowledge)를 송신합니다.

#### *Parameters*

Type	Name	Description
uint32_t	systemBytes	수신한 Message 의 SystemBytes 입니다.
int	ack	응답할 ack 입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult ReplySECSMessage(uint32\_t systemBytes, SECSCMessage\* primaryMessage, SECSCMessage\* secondaryMessage)**

*Description*

User Primary Message 수신 후 Secondary Message를 송신합니다.

*Parameters*

Type	Name	Description
SECSCMessage*	primaryMessage	수신한 Primary Message입니다.
SECSCMessage*	secondaryMessage	응답할 Secondary Message입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **Create and Delete Instance**

***List<bool>\* CreateBoolList()***

#### *Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

#### *Parameters*

(none)

#### *Return Value*

Type	Description
List<bool>	생성된 List입니다.

***void DeleteBoolList(List<bool>\* p)***

#### *Description*

List 객체를 삭제합니다.

#### *Parameters*

Type	Name	Description
List<bool>*	p	삭제할 객체의 주소입니다.

#### *Return Value*

(none)

*List<int8\_t>\* CreateI1List()*

*Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<int8_t>	생성된 List입니다.

*void DeleteI1List(List<int8\_t>\* p)*

*Description*

List 객체를 삭제합니다.

*Parameters*

Type	Name	Description
List<int8_t>*	p	삭제할 객체의 주소입니다.

*Return Value*

(none)

*List<int16\_t>\* CreateI2List()*

*Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<int16_t>	생성된 List입니다.

*void DeleteI2List(List<int16\_t>\* p)*

*Description*

List 객체를 삭제합니다.

*Parameters*

Type	Name	Description
List<int16_t>*	p	삭제할 객체의 주소입니다.

*Return Value*

(none)

*List<int32\_t>\* CreateI4List()*

*Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<int32_t>	생성된 List입니다.

*void DeleteI4List(List<int32\_t>\* p)*

*Description*

List 객체를 삭제합니다.

*Parameters*

Type	Name	Description
List<int32_t>*	p	삭제할 객체의 주소입니다.

*Return Value*

(none)

*List<int64\_t>\* CreateI8List()*

*Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<int64_t>	생성된 List입니다.

*void DeleteI8List(List<int64\_t>\* p)*

*Description*

List 객체를 삭제합니다.

*Parameters*

Type	Name	Description
List<int64_t>*	p	삭제할 객체의 주소입니다.

*Return Value*

(none)

***List<uint8\_t>\* CreateU1List()***

*Description*

GEM Driver API의 인자로 사용할 List 개체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<uint8_t>	생성된 List입니다.

***void DeleteU1List(List<uint8\_t>\* p)***

*Description*

List 개체를 삭제합니다.

*Parameters*

Type	Name	Description
List<uint8_t>*	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

*List<uint16\_t>\* CreateU2List()*

*Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<uint16_t>	생성된 List입니다.

*void DeleteU2List(List<uint16\_t>\* p)*

*Description*

List 객체를 삭제합니다.

*Parameters*

Type	Name	Description
List<uint16_t>*	p	삭제할 객체의 주소입니다.

*Return Value*

(none)

*List<uint32\_t>\* CreateU4List()*

*Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<uint32_t>	생성된 List입니다.

*void DeleteU4List(List<uint32\_t>\* p)*

*Description*

List 객체를 삭제합니다.

*Parameters*

Type	Name	Description
List<uint32_t>*	p	삭제할 객체의 주소입니다.

*Return Value*

(none)

*List<uint64\_t>\* CreateU8List()*

*Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<uint64_t>	생성된 List입니다.

*void DeleteU8List(List<uint64\_t>\* p)*

*Description*

List 객체를 삭제합니다.

*Parameters*

Type	Name	Description
List<uint64_t>*	p	삭제할 객체의 주소입니다.

*Return Value*

(none)

***List<float> \* CreateF4List()***

*Description*

GEM Driver API의 인자로 사용할 List 개체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<float>	생성된 List입니다.

***void DeleteF4List(List<float> \* p)***

*Description*

List 개체를 삭제합니다.

*Parameters*

Type	Name	Description
List<float>*	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

***List<double>\* CreateF8List()***

*Description*

GEM Driver API의 인자로 사용할 List 개체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<double>	생성된 List입니다.

***void DeleteF8List(List<double>\* p)***

*Description*

List 개체를 삭제합니다.

*Parameters*

Type	Name	Description
List<double>*	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

*List<LPCTSTR>\* CreateStringList()*

*Description*

GEM Driver API의 인자로 사용할 List 객체를 생성합니다.

*Parameters*

(none)

*Return Value*

Type	Description
List<LPCTSTR>	생성된 List입니다.

*void DeleteStringList(List<LPCTSTR>\* p, bool bItemDelete = true)*

*Description*

List 객체를 삭제합니다.

bItemDelete: true: 내부 Item을 순회하며, 개별 삭제합니다.

false: 내부 Item을 삭제하지 않습니다.

*Parameters*

Type	Name	Description
List<LPCTSTR>*	p	삭제할 객체의 주소입니다.
[Optional] bool	bItemDelete	내부 Item 삭제 여부입니다.

*Return Value*

(none)

**VariableCollection\* GetAllVariable(bool bRebuild = false)**

*Description*

GEM Driver에 등록된 모든 Variable 개체를 가져옵니다.

bRebuild: true: UGC 데이터 다시 가져옴

false: 이전에 가져왔던 데이터 가져옴

GetAllVariable로 가져온 개체는 Wrapper 클래스 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
[Optional] bool	bRebuild	UGC 데이터 사용 여부

*Return Value*

Type	Description
VariableCollection*	Driver 에 등록된 모든 Variable 입니다.

**VariableInfo\* CreateVariable(LPCTSTR strVID, GEMSECSFormat format, LPCTSTR strName = nullptr, GEMVariableType vidType = UbiGEMWrapper::Structure::GEMVariable\_SV, LPCTSTR strDescription = nullptr)**

*Description*

VariableInfo 개체를 생성합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	생성할 VID입니다.
GEMSECSFormat	format	생성할 Variable의 format입니다.
[Optional] LPCTSTR	strName	생성할 Variable의 이름입니다.
[Optional] GEMVariableType	vidType	생성할 Variable의 타입입니다.
[Optional] LPCTSTR	strDescription	생성할 Variable의 Description입니다.

*Return Value*

Type	Description
VariableInfo*	생성된 Variable 개체입니다.

***VariableInfo\* GetVariable(LPCTSTR strVID)****Description*

GEM Driver에 등록된 VariableInfo 객체를 가져옵니다.

GEM Driver에 등록되지 않은 VID인 경우 null이 return됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	가져올 VID입니다.

*Return Value*

Type	Description
VariableInfo*	검색된 Variable 객체입니다.

***void DeleteVariableInfo(VariableInfo \*p)****Description*

VariableInfo 객체를 삭제합니다.

VariableCollection에 추가한 VariableInfo는 삭제할 필요 없습니다.

VariableCollection 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
VariableInfo *	p	삭제할 객체의 주소입니다.

*Return Value*

(none)

***CollectionEventCollection\* GetAllCollectionEventInfo(bool bRebuild = false)***

*Description*

GEM Driver에 등록된 모든 CollectionEvent 개체를 가져옵니다.

bRebuild: true: UGC 데이터 다시 가져옴

false: 이전에 가져왔던 데이터 가져옴

GetAllCollectionEventInfo로 가져온 개체는 [Wrapper 클래스](#) 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
[Optional] bool	bRebuild	UGC 데이터 사용 여부

*Return Value*

Type	Description
CollectionEventCollection*	Driver 에 등록된 모든 CollectionEvent 입니다.

***CollectionEventInfo\* CreateCollectionEventInfo(LPCTSTR strCEID)***

*Description*

CollectionEvent 개체를 생성합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strCEID	생성할 CEID입니다.

*Return Value*

Type	Description
CollectionEventInfo*	생성된 CollectionEventInfo 개체입니다.

***CollectionEventInfo\* GetCollectionEventInfo(LPCTSTR strCEID)***

*Description*

Driver에 등록된 CollectionEvent 개체를 가져옵니다.

*Parameters*

Type	Name	Description
LPCTSTR	strCEID	가져올 CEID입니다.

*Return Value*

Type	Description
CollectionEventInfo*	검색된 CollectionEventInfo 개체입니다.

***void DeleteCollectionEventInfo(CollectionEventInfo \*p)***

*Description*

CollectionEventInfo 개체를 삭제합니다.

*Parameters*

Type	Name	Description
CollectionEventInfo *	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

***ReportInfo\* CreateReportInfo(LPCTSTR strRPTID)***

*Description*

ReportInfo 개체를 생성합니다.

CollectionEventInfo에 추가한 ReportInfo는 삭제할 필요 없습니다.

CollectionEventInfo 개체 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strRPTID	생성할 RPTID입니다.

*Return Value*

Type	Description
ReportInfo*	생성된 ReportInfo 개체입니다.

***void DeleteReportInfo(ReportInfo\* p)***

*Description*

ReportInfo 개체를 삭제합니다.

CollectionEventInfo에 추가한 ReportInfo는 삭제할 필요 없습니다.

CollectionEventInfo 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
ReportInfo *	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

***SECSMessageCollection\* GetAllUserMessage(bool bRebuild = false)***

*Description*

GEM Driver에 등록된 모든 UserMessage 개체를 가져옵니다.

bRebuild: true: UGC 데이터 다시 가져옴

false: 이전에 가져왔던 데이터 가져옴

GetAllUserMessage로 가져온 개체는 Wrapper 클래스 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
[Optional] bool	bRebuild	UGC 데이터 사용 여부

*Return Value*

Type	Description
SECSMessageCollection*	Driver에 등록된 모든 UserMessage입니다.

***SECSMessage\* CreateUserMessage(LPCTSTR strName, int stream, int function, bool bWaitBit = true, bool bToHost = true)***

*Description*

UserMessage 개체를 생성합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strName	생성할 Message의 이름입니다.
int	stream	생성할 Message의 Stream입니다.
int	function	생성할 Message의 Function입니다.
[Optional] bool	bWaitBit	생성할 Message의 WaitBit 활성화 여부입니다.
[Optional] bool	bToHost	생성할 Message의 방향입니다.

*Return Value*

Type	Description
SECSMessage*	생성된 SECSMessage 개체입니다.

***SECSMessage\* GetUserMessage(int stream, int function, bool bToHost = true)***

*Description*

Message의 stream, function, 방향으로 GEM Driver에 등록된 UserMessage 개체를 가져옵니다.

GEM Driver에 등록되지 않은 경우 null이 return됩니다.

*Parameters*

Type	Name	Description
int	stream	가져올 Message 의 Stream입니다.
int	function	가져올 Message 의 Function입니다.
[Optional] bool	bToHost	가져올 Message 의 방향입니다.

*Return Value*

Type	Description
SECSMessage*	검색된 SECSMessage 개체입니다.

***void DeleteSECSMessage(SECSMessage\* p)***

*Description*

SECSMessage 개체를 삭제합니다.

*Parameters*

Type	Name	Description
SECSMessage*	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

***FmtPPCollection\* CreateFmtPPCollection(LPCTSTR strPPID, LPCTSTR strMDLN, LPCTSTR strSOFTREV)***

*Description*

*FmtPPCollection* 개체를 생성합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strPPID	생성할 PPID입니다.
[Optional] LPCTSTR	strMDLN	생성할 MDLN입니다.
[Optional] LPCTSTR	strSOFTREV	생성할 SOFTREV입니다.

*Return Value*

Type	Description
FmtPPCollection*	생성된 <i>FmtPPCollection</i> 개체입니다.

***void DeleteFmtPPCollection(FmtPPCollection\* p)***

*Description*

*FmtPPCollection* 개체를 삭제합니다.

*Parameters*

Type	Name	Description
FmtPPCollection*	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

*FmtPPCCodeInfo\* CreateFmtPPCCodeInfo(LPCTSTR strCommandCode)*

*Description*

*FmtPPCCodeInfo* 개체를 생성합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strCommandCode	생성할 CommandCode 입니다.

*Return Value*

Type	Description
<i>FmtPPCCodeInfo*</i>	생성된 <i>FmtPPCCodeInfo</i> 개체입니다.

*void DeleteFmtPPCCodeInfo(FmtPPCCodeInfo\* p)*

*Description*

*FmtPPCCodeInfo* 개체를 삭제합니다.

*FmtPPCollection*에 추가한 *FmtPPCCodeInfo*는 삭제할 필요 없습니다.

*FmtPPCollection* 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
<i>FmtPPCCodeInfo*</i>	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

***FmtPPItem\* CreateFmtPPItem(LPCTSTR strPPName, LPCTSTR strPPValue, GEMSECSFormat format)***

*Description*

*FmtPPItem* 개체를 생성합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strPPName	생성할 PPName 입니다.
LPCTSTR	strPPValue	생성할 PPValue 입니다.
GEMSECSFormat	format	생성할 Format 입니다.

*Return Value*

Type	Description
FmtPPItem*	생성된 <i>FmtPPItem</i> 개체입니다.

***void DeleteFmtPPItem(FmtPPItem \*p)***

*Description*

*FmtPPItem* 개체를 삭제합니다.

*FmtPPCCodeInfo*에 추가한 *FmtPPItem*은 삭제할 필요 없습니다.

*FmtPPCCodeInfo* 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
FmtPPItem*	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

```
FmtPPVerificationInfo* CreateFmtPPVerificationInfo(int nACK, int nSeqNum, LPCTSTR strErrW7)
```

*Description*

FmtPPVerificationInfo 개체를 생성합니다.

*Parameters*

Type	Name	Description
int	nACK	생성할 ACK입니다.
int	nSeqNum	생성할 SeqNum입니다.
LPCTSTR	strErrW7	생성할 ErrW7입니다.

*Return Value*

Type	Description
FmtPPVerificationInfo*	생성된 FmtPPVerificationInfo 개체입니다.

```
void DeleteFmtPPVerificationInfo(FmtPPVerificationInfo* p)
```

*Description*

FmtPPVerificationInfo 개체를 삭제합니다.

FmtPPVerificationCollection에 추가한 FmtPPVerificationInfo는 삭제할 필요 없습니다.

FmtPPVerificationCollection 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
FmtPPVerificationInfo*	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

***FmtPPVerificationCollection\* CreateFmtPPVerificationCollection(LPCTSTR strPPID)***

*Description*

*FmtPPVerificationCollection* 개체를 생성합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strPPID	생성할 PPID입니다.

*Return Value*

Type	Description
<i>FmtPPVerificationCollection*</i>	생성된 <i>FmtPPVerificationCollection</i> 개체입니다.

***void DeleteFmtPPVerificationCollection(FmtPPVerificationCollection \*p)***

*Description*

*FmtPPVerificationCollection* 개체를 삭제합니다.

*Parameters*

Type	Name	Description
<i>FmtPPVerificationCollection *</i>	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

***RemoteCommandResult\* CreateRemoteCommandResult(int nHostCommandAck = 0)***

*Description*

RemoteCommandResult 개체를 생성합니다.

*Parameters*

Type	Name	Description
[Optional] int	nHostCommandAck	생성할 HostCommandAck입니다.

*Return Value*

Type	Description
RemoteCommandResult*	생성된 RemoteCommandResult 개체입니다.

***void DeleteRemoteCommandResult(RemoteCommandResult \*p)***

*Description*

RemoteCommandResult 개체를 삭제합니다.

*Parameters*

Type	Name	Description
RemoteCommandResult	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

**RemoteCommandParameterResult\***                    *CreateRemoteCommandParameterResult(LPCTSTR strCpName, int nParameterAck)*

*Description*

RemoteCommandParameterResult 개체를 생성합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strCpName	생성할 CpName 입니다.
int	nParameterAck	생성할 ParameterAck 입니다.

*Return Value*

Type	Description
RemoteCommandParameterResult*	생성된 RemoteCommandParameterResult 개체입니다.

**void DeleteRemoteCommandParameterResult(RemoteCommandParameterResult\* p)**

*Description*

RemoteCommandParameterResult 개체를 삭제합니다.

RemoteCommandResult에 추가한 RemoteCommandParameterResult는 삭제할 필요 없습니다.

RemoteCommandResult 삭제 시 자동 삭제됩니다.

*Parameters*

Type	Name	Description
RemoteCommandParameterResult*	p	삭제할 개체의 주소입니다.

*Return Value*

(none)

## ***Set Variable and Equipment Constant Value***

**GEMResult SetVariable(LPCTSTR strVID, bool newValue)**

### *Description*

Variable에 Value를 설정합니다.

### *Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
bool	newValue	설정할 값입니다.

### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, uint8\_t newValue)**

### *Description*

Variable에 Value를 설정합니다.

### *Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
uint8_t	newValue	설정할 값입니다.

### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, uint16\_t newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
uint16_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, uint32\_t newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
uint32_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, uint64\_t newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
uint64_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, int8\_t newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
int8_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, int16\_t newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
int16_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, int32\_t newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
int32_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, int64\_t newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
int64_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, float newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
float	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, double newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
double	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, LPCTSTR newValue)**

*Description*

Variable에 Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
LPCTSTR	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<bool>\* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<bool>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<uint8\_t>\* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<uint8_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<uint16\_t> \* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<uint16_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<uint32\_t> \* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<uint32_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<uint64\_t> \* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<uint64_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<int8\_t> \* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<int8_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<int16\_t>\* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<int16_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<int32\_t>\* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<int32_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<int64\_t> \* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<int64_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<float> \* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<float >*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(LPCTSTR strVID, List<double>\* arrayValue)**

*Description*

Variable에 Array Value를 설정합니다.

*Parameters*

Type	Name	Description
LPCTSTR	strVID	값을 설정할 VID입니다.
List<double>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(VariableInfo\* variableInfo)**

*Description*

Driver에 variableInfo를 설정합니다.

*Parameters*

Type	Name	Description
VariableInfo*	variableInfo	설정할 VariableInfo입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetVariable(VariableCollection\* variableInfos)**

*Description*

Driver에 variableCollection(복수개의 VariableInfo)를 설정합니다.

*Parameters*

Type	Name	Description
VariableCollection*	variableCollection	설정할 VariableCollection입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, bool newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
bool	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, uint8\_t newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
uint8_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, uint16\_t newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
uint16_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult SetEquipmentConstant(LPCTSTR strECID, uint32\_t newValue)**

#### *Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
uint32_t	newValue	설정할 값입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult SetEquipmentConstant(LPCTSTR strECID, uint64\_t newValue)**

#### *Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
uint64_t	newValue	설정할 값입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, int8\_t newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
int8_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, int16\_t newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
int16_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, int32\_t newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
int32_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, int64\_t newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
int64_t	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, float newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
float	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, double newValue)**

*Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
double	newValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult SetEquipmentConstant(LPCTSTR strECID, LPCTSTR newValue)**

#### *Description*

EC에 Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
LPCTSTR	newValue	설정할 값입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### **GEMResult SetEquipmentConstant(LPCTSTR strECID, List<bool> \* arrayValue)**

#### *Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<bool> *	arrayValue	설정할 값입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<uint8\_t>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<uint8_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<uint16\_t>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<uint16_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<uint32\_t>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<uint32_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<uint64\_t>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<uint64_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<int8\_t>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<int8_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<int16\_t>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<int16_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<int32\_t>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<int32_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<int64\_t>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<int64_t>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<float>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<float>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

**GEMResult SetEquipmentConstant(LPCTSTR strECID, List<double>\* arrayValue)**

*Description*

EC에 Array Value를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

*Parameters*

Type	Name	Description
LPCTSTR	strECID	값을 설정할 ECID입니다.
List<double>*	arrayValue	설정할 값입니다.

*Return Value*

Type	Description
GEMResult	수행 결과입니다.

### ***GEMResult SetEquipmentConstant(VariableInfo\* variableInfo)***

#### *Description*

Driver에 variableInfo를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

#### *Parameters*

Type	Name	Description
VariableInfo*	variableInfo	설정할 VariableInfo입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

### ***GEMResult SetEquipmentConstant(VariableCollection\* variableInfos)***

#### *Description*

Driver에 variableCollection(복수개의 VariableInfo)를 설정합니다.

S6F11(Event Report Send) – Equipment Constant Changed가 보고됩니다.

#### *Parameters*

Type	Name	Description
VariableCollection*	variableCollection	설정할 VariableCollection입니다.

#### *Return Value*

Type	Description
GEMResult	수행 결과입니다.

## **Subscribe Events**

**void SubscribeInitialize(IGEMDriverEvents\* eventReceiver)**

### *Description*

Driver에 Event Receiver를 등록(초기화) 합니다.

### *Parameters*

Type	Name	Description
IGEMDriverEvents*	eventReceiver	이벤트를 받을 개체입니다.

### *Return Value*

(none)

**void SubscribeGEMConnected()**

### *Description*

GEMConnected 이벤트를 활성화합니다.

### *Parameters*

(none)

### *Return Value*

(none)

**void SubscribeGEMDisconnected()**

*Description*

GEMDisconnected 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeGEMSelected()**

*Description*

GEMSelected 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeGEMDeselected()**

*Description*

GEMDeselected 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeCommunicationStateChanged()**

*Description*

CommunicationStateChanged 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeControlStateChanged()***

*Description*

ControlStateChanged 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeControlStateOnlineChangeFailed()***

*Description*

ControlStateOnlineChangeFailed 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeEquipmentProcessState()**

*Description*

EquipmentProcessState 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeSpoolStateChanged()**

*Description*

SpoolStateChanged 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeReceivedEstablishCommunicationsRequest()***

*Description*

ReceivedEstablishCommunicationsRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeReceivedRemoteCommand()***

*Description*

ReceivedRemoteCommand 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeReceivedEnhancedRemoteCommand()***

*Description*

ReceivedEnhancedRemoteCommand 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeReceivedNewECVSend()***

*Description*

ReceivedNewECVSend 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedLoopback()**

*Description*

ReceivedLoopback 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedTerminalMessage()**

*Description*

ReceivedTerminalMessage 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedTerminalMultiMessage()**

*Description*

ReceivedTerminalMultiMessage 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedRequestOffline()**

*Description*

ReceivedRequestOffline 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedRequestOnline()**

*Description*

ReceivedRequestOnline 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedDefineReport()**

*Description*

ReceivedDefineReport 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedLinkEventReport()**

*Description*

ReceivedLinkEventReport 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedEnableDisableEventReport()**

*Description*

ReceivedEnableDisableEventReport 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedEnableDisableAlarmSend()**

*Description*

ReceivedEnableDisableAlarmSend 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedPPLoadInquire()**

*Description*

ReceivedPPLoadInquire 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedPPSend()**

*Description*

ReceivedPPSend 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedFmtPPSend()**

*Description*

ReceivedFmtPPSend 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedDateTimeRequest()**

*Description*

ReceivedDateTimeRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedDateTimeSetRequest()**

*Description*

ReceivedDateTimeSetRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedPPRequest()**

*Description*

ReceivedPPRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedFmtPPRequest()**

*Description*

ReceivedFmtPPRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedDeletePPSend()**

*Description*

ReceivedDeletePPSend 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedCurrentEPPDRequest()**

*Description*

ReceivedCurrentEPPDRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeUserPrimaryMessageReceived()***

*Description*

UserPrimaryMessageReceived 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeResponseDateTimeRequest()***

*Description*

ResponseDateTimeRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeResponseLoopback()***

*Description*

ResponseLoopback 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeResponseEventReportAcknowledge()***

*Description*

ResponseEventReportAcknowledge 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeResponsePPLoadInquire()**

*Description*

ResponsePPLoadInquire 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeResponsePPSend()**

*Description*

ResponsePPSend 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeResponsePPRequest()***

*Description*

ResponsePPRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeResponseFmtPPSend()***

*Description*

ResponseFmtPPSend 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeResponseFmtPPRequest()**

*Description*

ResponseFmtPPRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeResponseFmtPPVerification()**

*Description*

ResponseFmtPPVerification 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeResponseTerminalRequest()**

*Description*

ResponseTerminalRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeUserSecondaryMessageReceived()**

*Description*

UserSecondaryMessageReceived 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeVariableUpdateRequest()**

*Description*

VariableUpdateRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeTraceDataUpdateRequest()**

*Description*

TraceDataUpdateRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeUserGEMMessageUpdateRequest()***

*Description*

UserGEMMessageUpdateRequest 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeInvalidMessageReceived()***

*Description*

InvalidMessageReceived 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeReceivedUnknownMessage()***

*Description*

ReceivedUnknownMessage 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

***void SubscribeReceivedInvalidRemoteCommand()***

*Description*

ReceivedInvalidRemoteCommand 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

**void SubscribeReceivedInvalidEnhancedRemoteCommand()**

*Description*

ReceivedInvalidEnhancedRemoteCommand 이벤트를 활성화합니다.

*Parameters*

(none)

*Return Value*

(none)

## Event

### *Connection and State Changed Events*

**void OnGEMConnected(LPCTSTR strIpAddress, int nPortNo)**

#### *Description*

GEM Driver가 연결되었을 때 발생하는 이벤트입니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strIpAddress	연결된 IpAddress입니다.
int	nPortNo	연결된 PortNo입니다.

#### *Return Value*

(none)

**void OnGEMDisconnected(LPCTSTR strIpAddress, int nPortNo)**

#### *Description*

GEM Driver가 연결해제되었을 때 발생하는 이벤트입니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strIpAddress	연결 해제된 IpAddress입니다.
int	nPortNo	연결 해제된 PortNo입니다.

#### *Return Value*

(none)

**void OnGEMSelected(LPCTSTR strIpAddress, int nPortNo)**

*Description*

GEM Driver가 Selected상태로 변경 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
LPCTSTR	strIpAddress	Selected로 변경된 IpAddress입니다.
int	nPortNo	Selected로 변경된 PortNo입니다.

*Return Value*

(none)

**void OnGEMDeselected(LPCTSTR strIpAddress, int nPortNo)**

*Description*

GEM Driver가 Deselected상태로 변경 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
LPCTSTR	strIpAddress	Deselected로 변경된 IpAddress입니다.
int	nPortNo	Deselected로 변경된 PortNo입니다.

*Return Value*

(none)

***void OnCommunicationStateChanged(CommunicationState communicationState)***

*Description*

GEM Driver의 Communication 상태가 변경 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
CommunicationState	communicationState	변경된 Communication 상태입니다.

*Return Value*

(none)

***void OnControlStateChanged(ControlState controlState)***

*Description*

GEM Driver의 Control 상태가 변경 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
ControlState	controlState	변경된 Control 상태입니다.

*Return Value*

(none)

***void OnControlStateOnlineChangeFailed()***

*Description*

GEM Driver가 Control State를 Online으로 변경 실패하였을 때 발생하는 이벤트입니다.

*Parameters*

(none)

*Return Value*

(none)

***void OnEquipmentProcessState(uint8\_t equipmentProcessState)***

*Description*

설비 공정 상태가 변경되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
uint8_t	equipmentProcessState	변경된 설비 공정 상태입니다.

*Return Value*

(none)

***void OnSpoolStateChanged(GEMSpoolState spoolState)***

*Description*

GEM Driver의 Spool 상태가 변경되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
GEMSpoolState	spoolState	변경된 Spool 상태입니다.

*Return Value*

(none)

### **Primary Message Received Events**

*int OnReceivedEstablishCommunicationRequest(LPCTSTR strMDLN, LPCTSTR strSOFTREV)*

#### *Description*

S1F13(Establish Communication Request)가 수신 되었을 때 발생하는 이벤트입니다.

S1F14(Establish Communication Request Acknowledge)를 자동 응답합니다.

Return Value에 따라 Communication State가 변경됩니다.

#### *Parameters*

Type	Name	Description
LPCTSTR	strMDLN	응답할 MDLN입니다.
LPCTSTR	strSOFTREV	응답할 SOFTREV입니다.

#### *Return Value*

Type	Description
int	ack 코드입니다.

*void OnReceivedRemoteCommand(RemoteCommandInfo\*remoteCommandInfo)*

#### *Description*

S2F41(Host Command Send)가 수신 되었을 때 발생하는 이벤트입니다.

S2F42(Host Command Acknowledge)를 송신하기 위해 ReplyRemoteCommandAck를 호출해야 합니다.

#### *Parameters*

Type	Name	Description
RemoteCommandInfo*	remoteCommandInfo	수신된 RemoteCommand입니다.

#### *Return Value*

(none)

**void OnReceivedEnhancedRemoteCommand(EnhancedRemoteCommandInfo\* ercmdInfo)**

*Description*

S2F49(Enhanced Remote Command)가 수신 되었을 때 발생하는 이벤트입니다.

S2F50(Enhanced Remote Command Acknowledge)를 송신하기 위해 ReplyEnhancedRemoteCommandAck를 호출해야합니다.

*Parameters*

Type	Name	Description
EnhancedRemoteCommandInfo*	ercmdInfo	수신된 EnhancedRemoteCommand 입니다.

*Return Value*

(none)

**void OnReceivedNewECVSend(VariableCollection\* newEcInfo)**

*Description*

S2F15(New Equipment Constant Send)가 수신 되었을 때 발생하는 이벤트입니다.

S2F16(New Equipment Constant Acknowledge)를 송신하기 위해 ReplyNewEquipmentConstantSend를 호출해야합니다.

*Parameters*

Type	Name	Description
VariableCollection*	newEcInfo	수신된 EquipmentConstant 입니다.

*Return Value*

(none)

**void OnReceivedLoopback(List<uint8\_t>\* receiveData)**

*Description*

S2F25(Loopback Diagnostic Request)가 수신 되었을 때 발생하는 이벤트입니다.

S2F26(Loopback Diagnostic Data)를 자동 응답합니다.

*Parameters*

Type	Name	Description
List<uint8_t>*	receiveData	수신된 ABS입니다.

*Return Value*

(none)

**void OnReceivedTerminalMessage(uint32\_t systemBytes, int tid, LPCTSTR strTerminalMessage)**

*Description*

S10F3(Terminal Display, Single)가 수신 되었을 때 발생하는 이벤트입니다.

S10F4(Terminal Display, Single Acknowledge)를 송신하기 위해 ReplyTerminalMessageAck를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.
int	tid	수신된 TID입니다.
LPCTSTR	strTerminalMessage	수신된 Terminal Message입니다.

*Return Value*

(none)

```
void OnReceivedTerminalMultiMessage(uint32_t systemBytes, int tid, List<LPCTSTR>* strTerminalMessages)
```

#### Description

S10F5(Terminal Display, Multi-Block)가 수신 되었을 때 발생하는 이벤트입니다.

S10F6(Terminal Display, Multi-Block Acknowledge)를 송신하기 위해 ReplyTerminalMultiMessageAck를 호출해야합니다.

#### Parameters

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte 입니다.
int	tid	수신된 TID 입니다.
List<LPCTSTR>*	strTerminalMessages	수신된 Terminal Message 리스트입니다.

#### Return Value

(none)

```
void OnReceivedRequestOffline(uint32_t systemBytes)
```

#### Description

S1F15(Request OFF-LINE)가 수신 되었을 때 발생하는 이벤트입니다.

S1F16(OFF-LINE Acknowledge)를 송신하기 위해 ReplyRequestOfflineAck를 호출해야합니다.

#### Parameters

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte 입니다.

#### Return Value

(none)

***void OnReceivedRequestOnline(uint32\_t systemBytes)***

*Description*

S1F17(Request ON-LINE)가 수신 되었을 때 발생하는 이벤트입니다.

S1F18(ON-LINE Acknowledge)를 송신하기 위해 ReplyRequestOnlineAck를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.

*Return Value*

(none)

***void OnReceivedDefineReport()***

*Description*

S2F33(Define Report)가 수신 되었을 때 발생하는 이벤트입니다.

S2F34(Define Report Acknowledge)를 자동 응답합니다.

*Parameters*

(none)

*Return Value*

(none)

***void OnReceivedLinkEventReport()***

*Description*

S2F35(Link Event Report)가 수신 되었을 때 발생하는 이벤트입니다.

S2F36(Link Event Report Acknowledge)를 자동 응답합니다.

*Parameters*

(none)

*Return Value*

(none)

***void OnReceivedEnableDisableEventReport()***

*Description*

S2F37(Enable/Disable Event Report)가 수신 되었을 때 발생하는 이벤트입니다.

S2F38(Enable/Disable Event Report Acknowledge)를 자동 응답합니다.

*Parameters*

(none)

*Return Value*

(none)

### ***void OnReceivedEnableDisableAlarmSend()***

#### *Description*

S5F3(Enable/Disable Alarm Send)가 수신 되었을 때 발생하는 이벤트입니다.

S5F4(Enable/Disable Alarm Acknowledge)를 자동 응답합니다.

#### *Parameters*

(none)

#### *Return Value*

(none)

### ***void OnReceivedPPLoadInquire(uint32\_t systemBytes, LPCTSTR strPPID, int length)***

#### *Description*

S7F1(Process Program Load Inquire)가 수신 되었을 때 발생하는 이벤트입니다.

S7F2(Process Program Load Grant)를 송신하기 위해 ReplyPPLoadInquireAck를 호출해야합니다.

#### *Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte 입니다.
LPCTSTR	strPPID	수신된 PPID 입니다.
int	length	수신된 Length 입니다.

#### *Return Value*

(none)

**void OnReceivedPPSend(uint32\_t systemBytes, LPCTSTR strPPID, List<uint8\_t>\* ppbody)**

*Description*

S7F3(Process Program Send)가 수신 되었을 때 발생하는 이벤트입니다.

S7F4(Process Program Acknowledge)를 송신하기 위해 ReplyPPSendAck를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.
LPCTSTR	strPPID	수신된 PPID입니다.
List<uint8_t>*	ppbody	수신된 Body입니다.

*Return Value*

(none)

**void OnReceivedFmtPPSend(uint32\_t systemBytes, FmtPPCollection\* fmtPPCollection)**

*Description*

S7F23(Formatted Process Program Send)가 수신 되었을 때 발생하는 이벤트입니다.

S7F24(Formatted Process Program Acknowledge)를 송신하기 위해 ReplyFmtPPSendAck를 호출해야 합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.
FmtPPCollection*	fmtPPCollection	수신된 FmtPPCollection입니다.

*Return Value*

(none)

```
void OnReceivedDateTimeRequest(uint32_t systemBytes, DateTime* timeData)
```

*Description*

S2F17(Date and Time Request)가 수신 되었을 때 발생하는 이벤트입니다.

S2F18(Date and Time Data)를 송신하기 위해 ReplyDateTimeRequest를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.
DateTime*	timeData	응답할 DateTime입니다.

*Return Value*

(none)

```
void OnReceivedDateTimeSetRequest(uint32_t systemBytes, DateTime* timeData)
```

*Description*

S2F31(Date and Time Set Request)가 수신 되었을 때 발생하는 이벤트입니다.

S2F32(Date and Time Set Acknowledge)를 송신하기 위해 ReplyDateTimeSetRequest를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.
DateTime*	timeData	수신된 DateTime입니다.

*Return Value*

(none)

**void OnReceivedPPRequest(uint32\_t systemBytes, LPCTSTR strPPID)**

*Description*

S7F5(Process Program Request)가 수신 되었을 때 발생하는 이벤트입니다.

S7F6(Process Program Data)를 송신하기 위해 ReplyPPRequestAck를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.
LPCTSTR	strPPID	수신된 PPID입니다.

*Return Value*

(none)

**void OnReceivedFmtPPRequest(uint32\_t systemBytes, LPCTSTR strPPID)**

*Description*

S7F25(Formatted Process Program Request)가 수신 되었을 때 발생하는 이벤트입니다.

S7F26(Formatted Process Program Data)를 송신하기 위해 ReplyFmtPPRequestAck를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.
LPCTSTR	strPPID	수신된 PPID입니다.

*Return Value*

(none)

**void OnReceivedDeletePPSend(uint32\_t systemBytes, List<LPCTSTR>\* ppids)**

*Description*

S7F17(Delete Process Program Send)가 수신 되었을 때 발생하는 이벤트입니다.

S7F18(Delete Process Program Acknowledge)를 송신하기 위해 ReplyPPDeleteAck를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.
List<LPCTSTR>*	ppids	수신된 PPID 목록입니다.

*Return Value*

(none)

**void OnReceivedCurrentEPPDRequest(uint32\_t systemBytes)**

*Description*

S7F19(Current EPPD Request)가 수신 되었을 때 발생하는 이벤트입니다.

S7F20(Current EPPD Data)를 송신하기 위해 ReplyCurrentEPPDRequestAck를 호출해야합니다.

*Parameters*

Type	Name	Description
uint32_t	systemBytes	수신된 SystemByte입니다.

*Return Value*

(none)

**void OnUserPrimaryMessageReceived(SECSMessage \*message)**

*Description*

등록된 User Custom Message 중 Primary Message가 수신 되었을 때 발생하는 이벤트입니다.

Secondary Message를 응답하기 위해 ReplySECSMessage를 호출해야 합니다.

*Parameters*

Type	Name	Description
SECSMessage*	message	수신된 SECSMessage입니다.

*Return Value*

(none)

## **Secondary Message Received Events**

***bool OnResponseDateTimeRequest(DateTime\* timeData)***

### *Description*

S2F18(Date and Time Data)가 수신 되었을 때 발생하는 이벤트입니다.

Configuration 툴의 설정과 Return Value에 따라 Driver에서 시스템의 시간을 설정합니다.

### *Parameters*

Type	Name	Description
DateTime*	timeData	수신된 DateTime입니다.

### *Return Value*

Type	Description
bool	시스템 설정 결과입니다.

***void OnResponseLoopback(List<uint8\_t>\* receiveData, List<uint8\_t>\* sendData)***

### *Description*

S2F26(Loopback Diagnostic Data)가 수신 되었을 때 발생하는 이벤트입니다.

### *Parameters*

Type	Name	Description
List<uint8_t>*	receiveData	수신된 ABS입니다.
List<uint8_t>*	sendData	송신했던 ABS입니다.

### *Return Value*

(none)

**void OnResponseEventReportAcknowledge(LPCTSTR strCEID, int ack)**

*Description*

S6F12(Event Report Acknowledge)가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
LPCTSTR	strCEID	송신했던 CEID입니다.
int	ack	수신된 ack입니다.

*Return Value*

(none)

**void OnResponsePPLoadInquire(int ppgnt, LPCTSTR strPPID)**

*Description*

S7F2(Process Program Load Grant)가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
int	ppgnt	수신된 PPGNT입니다.
LPCTSTR	strPPID	송신했던 PPID입니다.

*Return Value*

(none)

**void OnResponsePPSend(int ack, LPCTSTR strPPID)**

*Description*

S7F4(Process Program Acknowledge)가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
int	ack	수신된 ACK입니다.
LPCTSTR	strPPID	송신했던 PPID입니다.

*Return Value*

(none)

**void OnResponsePPRequest(LPCTSTR strPPID, List<uint8\_t> \* ppbody)**

*Description*

S7F6(Process Program Data)가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
LPCTSTR	strPPID	수신된 PPID입니다.
List<uint8_t> *	ppbody	수신된 Body입니다.

*Return Value*

(none)

**void OnRequestFmtPPSend(int ack, FmtPPCollection\* fmtPPCollection)**

*Description*

S7F24(Formatted Process Program Acknowledge)가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
int	ack	수신된 ACK입니다.
FmtPPCollection*	fmtPPCollection	송신했던 FmtPPCollection입니다.

*Return Value*

(none)

**void OnResponseFmtPPRequest(FmtPPCollection \*fmtPPCollection)**

*Description*

S7F26(Formatted Process Program Data)가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
FmtPPCollection*	fmtPPCollection	수신된 FmtPPCollection입니다.

*Return Value*

(none)

***void OnResponseFmtPPVerification(FmtPPVerificationCollection \*fmtPPVerificationCollection)***

*Description*

S7F28(Process Program Verification Acknowledge)가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
FmtPPVerificationCollection*	fmtPPVerificationCollection	송신했던 FmtPPVerificationCollection 입니다

*Return Value*

(none)

***void OnResponseTerminalRequest(int ack)***

*Description*

S10F2(Terminal Request Acknowledge)가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
int	ack	수신된 ACK 입니다.

*Return Value*

(none)

```
void OnUserSecondaryMessageReceived(SECSMessage* primaryMessage, SECSMessage* secondaryMessage)
```

*Description*

등록된 User Custom Message 중 Secondary Message 가 수신 되었을 때 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
SECSMessage*	primaryMessage	송신했던 PrimaryMessage 입니다.
SECSMessage*	secondaryMessage	수신된 SecondaryMessage 입니다.

*Return Value*

(none)

### **Variable Value Update Events**

**void OnVariableUpdateRequest(GEMVariableUpdateType updateType, List<LPCTSTR>\* strVIDs)**

#### *Description*

Variable의 값을 Update하기 위해 발생하는 이벤트입니다.

1. S6F11(Event Report Send) 송신 전
2. S1F4(Selected Equipment Status Data) 송신 전
3. S6F16(Event Report Data) 송신 전
4. S6F20(Individual Report Data) 송신 전

#### *Parameters*

Type	Name	Description
GEMVariableUpdateType	updateType	Update Type 입니다.
List<LPCTSTR>*	strVIDs	값 Update 가 필요한 VID 목록입니다.

#### *Return Value*

(none)

**void OnTraceDataUpdateRequest(List<LPCTSTR> \* strVIDs)**

*Description*

S6F1(Trace Data Send) 송신 시 사용하기 위해 발생하는 이벤트입니다.

*Parameters*

Type	Name	Description
List<LPCTSTR> *	strVIDs	값 Update 가 필요한 VID 목록입니다.

*Return Value*

(none)

## *ETC Events*

*void OnUserGEMMessageUpdateRequest(SECSMessage\* message)*

### *Description*

등록된 User Custom Message 가 GEM에서 사용하는 표준 Stream, Function인 경우 발생하는 이벤트입니다.

SECSMessage의 내부 구조를 직접 채워야합니다.

### *Parameters*

Type	Name	Description
SECSMessage*	message	송신할 UserMessage 입니다.

### *Return Value*

(none)

### ***Invalid or Unknown Message Receive Events***

***void OnInvalidMessageReceived(GEMMessageValidationError error, SECSMessage\* message)***

#### *Description*

Message 수신 후 Stream 9 계열 오류 감지 시 발생하는 이벤트입니다.

1. S9F1(Unrecognized Device ID)
2. S9F3(Unrecognized Stream Type)
3. S9F5(Unrecognized Function Type)
4. S9F7(Illegal Data)
5. S9F9(Transaction Timer Timeout)
6. S9F11(Data Too Long)

#### *Parameters*

Type	Name	Description
GEMMessageValidationError	error	발생한 오류입니다.
SECSMessage*	message	수신된 Message입니다.

#### *Return Value*

(none)

***void OnReceivedUnknownMessage(SECSMessage\* message)***

#### *Description*

GEM에서 정의되지 않은 Message 수신 시 발생하는 이벤트입니다.

#### *Parameters*

Type	Name	Description
SECSMessage*	message	수신된 Message입니다.

#### *Return Value*

(none)

**void OnReceivedInvalidRemoteCommand(RemoteCommandInfo\* remoteCommandInfo)**

*Description*

S2F41(Host Command Send) 수신 시 Validation을 실패한 경우 발생하는 이벤트입니다.

1. 등록되지 않은 CommandName
2. 등록되지 않은 CPName
3. CPVAL의 Format 오류

*Parameters*

Type	Name	Description
RemoteCommandInfo*	remoteComnadInfo	수신된 RemoteCommandInfo 입니다.

*Return Value*

(none)

**void OnReceivedInvalidEnhancedRemoteCommand(EnhancedRemoteCommandInfo\* ercmdInfo)**

*Description*

S2F49(Enhanced Remote Command) 수신 시 Validation을 실패한 경우 발생하는 이벤트입니다.

1. 등록되지 않은 CommandName
2. 등록되지 않은 CPName
3. CPVAL의 Format 오류

*Parameters*

Type	Name	Description
EnhancedRemoteCommandInfo*	ercmdInfo	수신된 EnhancedRemoteCommandInfo 입니다.

*Return Value*

(none)

## Error Code

### GEMResult

#### *Description*

GEM driver 처리 결과입니다.

#### *Parameters*

Name	Value	Description
Ok	0	정상 상태입니다.
Unknown	1	알 수 없는 상태입니다.
NotExistDriverName	2	Driver name이 존재하지 않습니다.
NotExistFile	3	환경 설정 파일이 존재하지 않습니다.
FileLoadFailed	4	환경 설정 load에 실패했습니다.
FileSaveFailed	5	환경 설정 저장에 실패했습니다.
InvalidConfiguration	6	올바르지 않은 환경 설정 값입니다.
AlreadyConnected	7	이미 Connected 상태입니다.
SocketException	8	Socket 에러 상태입니다.
LicenseVerificationFailed	9	License 검증 실패입니다.
Disconnected	11	연결 해지 상태입니다.
ControlStatelsOffline	12	HOST offline 모드입니다.
SameState	13	이전과 동일한 상태입니다.
Undefined	14	정의되지 않은 상태입니다.
Disabled	15	Disable 상태입니다.
HSMSDriverError	16	HSMS driver 에러입니다.
HSMSDriverDisconnected	17	HSMS driver disconnected 상태입니다.
NotCommunicating	18	Not communicating 상태입니다.
MessageMakeFailed	19	Message 생성 실패입니다.
Exception	20	예외 발생 상태입니다.
Mismatch	21	Mismatch 발생 상태입니다.
HostDenied	22	Host에서 거부되었습니다.
InvalidFormat	23	Message의 구조가 잘못되었습니다.