

Dorian.NET

Manual

1. Table of Contents

1.	Table of Contents	2
2.	Dorian.AboutDlg Class	25
2.1.	Properties.....	26
2.1.1.	components Property.....	26
2.2.	Methods	27
2.2.1.	~Dorian.AboutDlg Method.....	27
2.2.2.	InitializeComponent Method.....	27
3.	Dorian.Comm Class	28
3.1.	Properties.....	29
3.1.1.	BaudRate Property	29
3.1.2.	ByteSize Property.....	29
3.1.3.	Connect Property.....	29
3.1.4.	Count Property	29
3.1.5.	Data Property	29
3.1.6.	FileName Property.....	30
3.1.7.	Json Property	30
3.1.8.	Parity Property.....	30
3.1.9.	Port Property	30
3.1.10.	StopBit Property	30
3.2.	Methods	31
3.2.1.	Config Method.....	31
3.2.2.	Load Method	31
3.2.3.	Read Method.....	31
3.2.4.	Save Method.....	31
3.2.5.	TimerProc Method	32
3.2.6.	Write Method.....	32
3.3.	Events.....	33
3.3.1.	ReceivedEventEvent	33
4.	Dorian.Comm.ReceivedEventArgs Class	34
4.1.	Properties.....	35
4.1.1.	data Property.....	35
4.1.2.	sender Property.....	35
5.	Dorian.CommConfigDlg Class	36
5.1.	Properties.....	37
5.1.1.	comm Property	37
5.1.2.	components Property.....	37
5.2.	Methods	38
5.2.1.	~Dorian.CommConfigDlg Method.....	38
5.2.2.	CommConfigDlg_Load Method	38
5.2.3.	Dorian.CommConfigDlg Method.....	38
5.2.4.	InitializeComponent Method.....	38
6.	Dorian.Communication Class	40
6.1.	Properties.....	42
6.1.1.	AutoReconnect Property	42
6.1.2.	AutoSelectReq Property.....	42
6.1.3.	AutoSessionID Property.....	42
6.1.4.	AutoSystemBytes Property	42
6.1.5.	BaudRate Property	42
6.1.6.	Clients Property.....	42
6.1.7.	Connect Property.....	43
6.1.8.	FileNameHsms Property	43
6.1.9.	FileNameSecsl Property	43
6.1.10.	HandleCtrlMsg Property.....	43
6.1.11.	Heartbeat Property	43
6.1.12.	Hsms Property	44
6.1.13.	IP Property.....	44
6.1.14.	IPv4 Property	44
6.1.15.	Json Property.....	44
6.1.16.	Master Property.....	44
6.1.17.	MDLN Property	45
6.1.18.	Port Property.....	45
6.1.19.	Retry Property.....	45
6.1.20.	Selected Property.....	45
6.1.21.	SerialPort Property	45

6.1.22.	Server Property	46
6.1.23.	SessionID Property	46
6.1.24.	SOFTREV Property	46
6.1.25.	SystemBytes Property	46
6.1.26.	T1 Property	46
6.1.27.	T2 Property	47
6.1.28.	T3 Property	47
6.1.29.	T4 Property	47
6.1.30.	T5 Property	47
6.1.31.	T6 Property	47
6.1.32.	T7 Property	48
6.1.33.	T8 Property	48
6.1.34.	Transactions Property	48
6.2.	Methods	49
6.2.1.	Config Method	49
6.2.2.	Load Method	49
6.2.3.	Save Method	49
6.2.4.	Send Method	49
6.2.5.	TimerProc Method	50
6.3.	Events	51
6.3.1.	ConnectedEventEvent	51
6.3.2.	DisconnectedEventEvent	51
6.3.3.	ProblemEventEvent	51
6.3.4.	ReceivedEventEvent	51
6.3.5.	ReceivedRawEventEvent	51
6.3.6.	SentEventEvent	52
6.3.7.	SentRawEventEvent	52
7.	Dorian.Forms.Comm Class	53
7.1.	Properties	54
7.1.1.	BaudRate Property	54
7.1.2.	ByteSize Property	54
7.1.3.	Connect Property	54
7.1.4.	Count Property	54
7.1.5.	Data Property	54
7.1.6.	FileName Property	55
7.1.7.	Json Property	55
7.1.8.	Parity Property	55
7.1.9.	Port Property	55
7.1.10.	StopBit Property	55
7.2.	Methods	56
7.2.1.	Config Method	56
7.2.2.	Load Method	56
7.2.3.	Read Method	56
7.2.4.	Save Method	56
7.2.5.	Write Method	57
7.3.	Events	58
7.3.1.	ReceivedEvent Method	58
8.	Dorian.Forms.Communication Class	59
8.1.	Properties	61
8.1.1.	AutoReconnect Property	61
8.1.2.	AutoSelectReq Property	61
8.1.3.	AutoSessionID Property	61
8.1.4.	AutoSystemBytes Property	61
8.1.5.	BaudRate Property	61
8.1.6.	Clients Property	61
8.1.7.	Connect Property	62
8.1.8.	FileNameHsms Property	62
8.1.9.	FileNameSecsl Property	62
8.1.10.	HandleCtrlMsg Property	62
8.1.11.	Heartbeat Property	62
8.1.12.	Hsms Property	63
8.1.13.	IP Property	63
8.1.14.	IPv4 Property	63
8.1.15.	Json Property	63
8.1.16.	Master Property	63

8.1.17.	MDLN Property	64
8.1.18.	Port Property	64
8.1.19.	Retry Property	64
8.1.20.	Selected Property	64
8.1.21.	SerialPort Property	64
8.1.22.	Server Property	65
8.1.23.	SessionID Property	65
8.1.24.	SOFTREV Property	65
8.1.25.	SystemBytes Property	65
8.1.26.	T1 Property	65
8.1.27.	T2 Property	66
8.1.28.	T3 Property	66
8.1.29.	T4 Property	66
8.1.30.	T5 Property	66
8.1.31.	T6 Property	66
8.1.32.	T7 Property	67
8.1.33.	T8 Property	67
8.1.34.	Transactions Property	67
8.2.	Methods	68
8.2.1.	Config Method	68
8.2.2.	Load Method	68
8.2.3.	Save Method	68
8.2.4.	Send Method	68
8.3.	Events	70
8.3.1.	ConnectedEventEvent	70
8.3.2.	DisconnectedEventEvent	70
8.3.3.	ProblemEventEvent	70
8.3.4.	ReceivedEventEvent	70
8.3.5.	ReceivedRawEventEvent	70
8.3.6.	SentEventEvent	71
8.3.7.	SentRawEventEvent	71
9.	Dorian.Forms.CommunicationView Class	72
9.1.	Properties	74
9.1.1.	AutoReconnect Property	74
9.1.2.	AutoSelectReq Property	74
9.1.3.	AutoSessionID Property	74
9.1.4.	AutoSystemBytes Property	74
9.1.5.	BaudRate Property	74
9.1.6.	Clients Property	74
9.1.7.	Connect Property	75
9.1.8.	FileNameHsms Property	75
9.1.9.	FileNameSecsl Property	75
9.1.10.	HandleCtrlMsg Property	75
9.1.11.	Heartbeat Property	75
9.1.12.	Hsms Property	76
9.1.13.	IP Property	76
9.1.14.	IPv4 Property	76
9.1.15.	Json Property	76
9.1.16.	Master Property	76
9.1.17.	MDLN Property	77
9.1.18.	Port Property	77
9.1.19.	Retry Property	77
9.1.20.	Selected Property	77
9.1.21.	SerialPort Property	77
9.1.22.	Server Property	78
9.1.23.	SessionID Property	78
9.1.24.	SOFTREV Property	78
9.1.25.	SystemBytes Property	78
9.1.26.	T1 Property	78
9.1.27.	T2 Property	79
9.1.28.	T3 Property	79
9.1.29.	T4 Property	79
9.1.30.	T5 Property	79
9.1.31.	T6 Property	79
9.1.32.	T7 Property	80

9.1.33.	T8 Property	80
9.1.34.	Transactions Property.....	80
9.2.	Methods	81
9.2.1.	Config Method.....	81
9.2.2.	Load Method	81
9.2.3.	Save Method.....	81
9.2.4.	Send Method.....	81
9.3.	Events.....	83
9.3.1.	ConnectedEventEvent	83
9.3.2.	DisconnectedEventEvent	83
9.3.3.	ProblemEventEvent.....	83
9.3.4.	ReceivedEventEvent	83
9.3.5.	ReceivedRawEventEvent.....	83
9.3.6.	SentEventEvent.....	84
9.3.7.	SentRawEventEvent.....	84
10.	Dorian.Forms.Gem Class.....	85
10.1.	Properties.....	86
10.1.1.	AlarmModel Property	86
10.1.2.	AutoReconnect Property.....	86
10.1.3.	AutoSelectReq Property	86
10.1.4.	AutoSessionID Property.....	86
10.1.5.	AutoSystemBytes Property	86
10.1.6.	CallDefProc Property	86
10.1.7.	Clients Property.....	87
10.1.8.	CommModel Property.....	87
10.1.9.	Connect Property	87
10.1.10.	CtrlModel Property	87
10.1.11.	EventModel Property	87
10.1.12.	FileName Property	88
10.1.13.	HandleCtrlMsg Property.....	88
10.1.14.	IP Property.....	88
10.1.15.	IPv4 Property	88
10.1.16.	Json Property.....	88
10.1.17.	Msg Property.....	88
10.1.18.	Port Property.....	89
10.1.19.	Selected Property.....	89
10.1.20.	Server Property.....	89
10.1.21.	SessionID Property.....	89
10.1.22.	SystemBytes Property	90
10.1.23.	T3 Property	90
10.1.24.	T5 Property	90
10.1.25.	T6 Property	90
10.1.26.	T7 Property	90
10.1.27.	T8 Property	91
10.2.	Methods	92
10.2.1.	Config Method.....	92
10.2.2.	InvokeAlarm Method.....	92
10.2.3.	Load Method.....	92
10.2.4.	Save Method.....	92
10.2.5.	Send Method.....	93
10.3.	Events.....	94
10.3.1.	CommStateChangedEvent Method.....	94
10.3.2.	ConnectedEvent Method	94
10.3.3.	CtrlStateChangedEvent Method.....	94
10.3.4.	DisconnectedEvent Method	94
10.3.5.	InvokeEvent Method.....	94
10.3.6.	ProblemEvent Method	95
10.3.7.	ReceivedEvent Method.....	95
10.3.8.	SentEvent Method.....	95
11.	Dorian.Forms.GemView Class.....	96
11.1.	Properties.....	97
11.1.1.	AlarmModel Property	97
11.1.2.	AutoReconnect Property.....	97
11.1.3.	AutoSelectReq Property	97
11.1.4.	AutoSessionID Property.....	97

11.1.5.	AutoSystemBytes Property	97
11.1.6.	CallDefProc Property	97
11.1.7.	Clients Property.....	98
11.1.8.	CommModel Property.....	98
11.1.9.	Connect Property	98
11.1.10.	CtrlModel Property	98
11.1.11.	EventModel Property	98
11.1.12.	FileName Property	99
11.1.13.	HandleCtrlMsg Property.....	99
11.1.14.	IP Property.....	99
11.1.15.	IPv4 Property	99
11.1.16.	Json Property.....	99
11.1.17.	Msg Property.....	99
11.1.18.	Port Property.....	100
11.1.19.	Selected Property.....	100
11.1.20.	Server Property	100
11.1.21.	SessionID Property.....	100
11.1.22.	SystemBytes Property	101
11.1.23.	T3 Property	101
11.1.24.	T5 Property	101
11.1.25.	T6 Property	101
11.1.26.	T7 Property	101
11.1.27.	T8 Property	102
11.2.	Methods	103
11.2.1.	Config Method.....	103
11.2.2.	InvokeAlarm Method.....	103
11.2.3.	Load Method	103
11.2.4.	Save Method.....	103
11.2.5.	Send Method.....	104
11.3.	Events.....	105
11.3.1.	CommStateChangedEvent Method	105
11.3.2.	ConnectedEvent Method	105
11.3.3.	CtrlStateChangedEvent Method.....	105
11.3.4.	DisconnectedEvent Method	105
11.3.5.	InvokeEvent Method.....	105
11.3.6.	ProblemEvent Method	106
11.3.7.	ReceivedEvent Method.....	106
11.3.8.	SentEvent Method.....	106
12.	Dorian.Forms.Hsms Class.....	107
12.1.	Properties.....	108
12.1.1.	AutoReconnect Property	108
12.1.2.	AutoSelectReq Property	108
12.1.3.	AutoSessionID Property.....	108
12.1.4.	AutoSystemBytes Property	108
12.1.5.	Clients Property.....	108
12.1.6.	Connect Property	108
12.1.7.	FileName Property	109
12.1.8.	HandleCtrlMsg Property.....	109
12.1.9.	Heartbeat Property	109
12.1.10.	IP Property.....	109
12.1.11.	IPv4 Property	110
12.1.12.	Json Property.....	110
12.1.13.	MDLN Property	110
12.1.14.	Port Property.....	110
12.1.15.	Selected Property.....	110
12.1.16.	Server Property	111
12.1.17.	SessionID Property.....	111
12.1.18.	SOFTREV Property	111
12.1.19.	SystemBytes Property	111
12.1.20.	T3 Property	111
12.1.21.	T5 Property	111
12.1.22.	T6 Property	112
12.1.23.	T7 Property	112
12.1.24.	T8 Property	112
12.1.25.	Transactions Property.....	112

12.2.	Methods	113
12.2.1.	Config Method	113
12.2.2.	Load Method	113
12.2.3.	Save Method	113
12.2.4.	Send Method	113
12.3.	Events	115
12.3.1.	ConnectedEvent Method	115
12.3.2.	DisconnectedEvent Method	115
12.3.3.	ProblemEvent Method	115
12.3.4.	ReceivedEvent Method	115
12.3.5.	SentEvent Method	115
13.	Dorian.Forms.HsmsView Class	117
13.1.	Properties	118
13.1.1.	AutoReconnect Property	118
13.1.2.	AutoSelectReq Property	118
13.1.3.	AutoSessionID Property	118
13.1.4.	AutoSystemBytes Property	118
13.1.5.	Clients Property	118
13.1.6.	Connect Property	118
13.1.7.	FileName Property	119
13.1.8.	HandleCtrlMsg Property	119
13.1.9.	Heartbeat Property	119
13.1.10.	IP Property	119
13.1.11.	IPv4 Property	120
13.1.12.	Json Property	120
13.1.13.	MDLN Property	120
13.1.14.	Port Property	120
13.1.15.	Selected Property	120
13.1.16.	Server Property	121
13.1.17.	SessionID Property	121
13.1.18.	SOFTREV Property	121
13.1.19.	SystemBytes Property	121
13.1.20.	T3 Property	121
13.1.21.	T5 Property	121
13.1.22.	T6 Property	122
13.1.23.	T7 Property	122
13.1.24.	T8 Property	122
13.1.25.	Transactions Property	122
13.2.	Methods	123
13.2.1.	Config Method	123
13.2.2.	Load Method	123
13.2.3.	Save Method	123
13.2.4.	Send Method	123
13.3.	Events	125
13.3.1.	ConnectedEvent Method	125
13.3.2.	DisconnectedEvent Method	125
13.3.3.	ProblemEvent Method	125
13.3.4.	ReceivedEvent Method	125
13.3.5.	SentEvent Method	125
14.	Dorian.Forms.Klarf12 Class	127
14.1.	Properties	128
14.1.1.	Count Property	128
14.1.2.	this[int] Property	128
14.1.3.	this[string] Property	128
14.2.	Methods	129
14.2.1.	Find Method	129
14.2.2.	Parse Method	129
15.	Dorian.Forms.Klarf12Map Class	130
15.1.	Properties	131
15.1.1.	Index Property	131
15.1.2.	klarf Property	131
15.1.3.	XY Property	131
16.	Dorian.Forms.Klarf12Table Class	132
16.1.	Properties	133
16.1.1.	Index Property	133

16.1.2.	klarf Property.....	133
17.	Dorian.Forms.Log Class.....	134
17.1.	Properties.....	135
17.1.1.	Count Property.....	135
17.1.2.	Enable Property.....	135
17.1.3.	FileName Property.....	135
17.1.4.	FileSize Property.....	135
17.1.5.	Json Property.....	135
17.1.6.	JsonFileName Property.....	135
17.1.7.	TimeStamp Property.....	136
17.2.	Methods.....	137
17.2.1.	Config Method.....	137
17.2.2.	Load Method.....	137
17.2.3.	Save Method.....	137
17.2.4.	Write Method.....	137
18.	Dorian.Forms.LogView Class.....	139
18.1.	Properties.....	140
18.1.1.	Count Property.....	140
18.1.2.	Enable Property.....	140
18.1.3.	FileName Property.....	140
18.1.4.	FileSize Property.....	140
18.1.5.	Json Property.....	140
18.1.6.	JsonFileName Property.....	140
18.1.7.	TimeStamp Property.....	141
18.2.	Methods.....	142
18.2.1.	Config Method.....	142
18.2.2.	Load Method.....	142
18.2.3.	Save Method.....	142
18.2.4.	Write Method.....	142
19.	Dorian.Forms.SecsI Class.....	144
19.1.1.	AutoSessionID Property.....	145
19.1.2.	AutoSystemBytes Property.....	145
19.1.3.	BaudRate Property.....	145
19.1.4.	Connect Property.....	145
19.1.5.	FileName Property.....	145
19.1.6.	Heartbeat Property.....	145
19.1.7.	Json Property.....	146
19.1.8.	Master Property.....	146
19.1.9.	MDLN Property.....	146
19.1.10.	Port Property.....	146
19.1.11.	Retry Property.....	146
19.1.12.	SessionID Property.....	146
19.1.13.	SOFTREV Property.....	147
19.1.14.	SystemBytes Property.....	147
19.1.15.	T1 Property.....	147
19.1.16.	T2 Property.....	147
19.1.17.	T3 Property.....	147
19.1.18.	T4 Property.....	148
19.2.	Methods.....	149
19.2.1.	Config Method.....	149
19.2.2.	Load Method.....	149
19.2.3.	Save Method.....	149
19.2.4.	Send Method.....	149
19.3.	Events.....	151
19.3.1.	ProblemEvent Method.....	151
19.3.2.	ReceivedEvent Method.....	151
19.3.3.	ReceivedRawEvent Method.....	151
19.3.4.	SentEvent Method.....	151
19.3.5.	SentRawEvent Method.....	151
20.	Dorian.Forms.SecsII Class.....	153
20.1.	Properties.....	154
20.1.1.	BlockNumber Property.....	154
20.1.2.	DeviceID Property.....	154
20.1.3.	Ebit Property.....	154
20.1.4.	Error Property.....	154

20.1.5.	Function Property	154
20.1.6.	Host Property	155
20.1.7.	Hsms Property	155
20.1.8.	Json Property	155
20.1.9.	Msg Property	155
20.1.10.	Node Property	155
20.1.11.	NodeCount Property	156
20.1.12.	NodeType Property	156
20.1.13.	NodeValue Property	156
20.1.14.	NodeValueHex Property	156
20.1.15.	PType Property	156
20.1.16.	Rbit Property	157
20.1.17.	SessionID Property	157
20.1.18.	Sml Property	157
20.1.19.	SmlType Property	157
20.1.20.	SourceID Property	157
20.1.21.	Stream Property	158
20.1.22.	SType Property	158
20.1.23.	SuggestedReplyMsg Property	158
20.1.24.	SystemBytes Property	158
20.1.25.	this[string] Property	158
20.1.26.	TransactionID Property	159
20.1.27.	Verification Property	159
20.1.28.	Wbit Property	159
20.2.	Methods	160
20.2.1.	Reply Method	160
20.2.2.	Reply Method	160
20.2.3.	Reset Method	160
21.	Dorian.Forms.SecsIView Class	161
21.1.	Properties	162
21.1.1.	BlockNumber Property	162
21.1.2.	DeviceID Property	162
21.1.3.	Ebit Property	162
21.1.4.	Error Property	162
21.1.5.	Function Property	162
21.1.6.	Host Property	163
21.1.7.	Hsms Property	163
21.1.8.	Json Property	163
21.1.9.	Msg Property	163
21.1.10.	Node Property	163
21.1.11.	NodeCount Property	164
21.1.12.	Nodeltem Property	164
21.1.13.	NodeType Property	164
21.1.14.	NodeValue Property	164
21.1.15.	NodeValueHex Property	164
21.1.16.	PType Property	165
21.1.17.	Rbit Property	165
21.1.18.	SessionID Property	165
21.1.19.	Sml Property	165
21.1.20.	SmlType Property	165
21.1.21.	SourceID Property	166
21.1.22.	Stream Property	166
21.1.23.	SType Property	166
21.1.24.	SuggestedReplyMsg Property	166
21.1.25.	SystemBytes Property	166
21.1.26.	this[string] Property	167
21.1.27.	TransactionID Property	167
21.1.28.	Verification Property	167
21.1.29.	Wbit Property	167
21.2.	Methods	168
21.2.1.	Reply Method	168
21.2.2.	Reply Method	168
21.2.3.	Reset Method	168
22.	Dorian.Forms.SecsIView Class	169
22.1.	Properties	170

22.1.1.	AutoSessionID Property	170
22.1.2.	AutoSystemBytes Property	170
22.1.3.	BaudRate Property	170
22.1.4.	Connect Property	170
22.1.5.	FileName Property	170
22.1.6.	Heartbeat Property	170
22.1.7.	Json Property	171
22.1.8.	Master Property	171
22.1.9.	MDLN Property	171
22.1.10.	Port Property	171
22.1.11.	Retry Property	171
22.1.12.	SessionID Property	171
22.1.13.	SOFTREV Property	172
22.1.14.	SystemBytes Property	172
22.1.15.	T1 Property	172
22.1.16.	T2 Property	172
22.1.17.	T3 Property	172
22.1.18.	T4 Property	173
22.2.	Methods	174
22.2.1.	Config Method	174
22.2.2.	Load Method	174
22.2.3.	Save Method	174
22.2.4.	Send Method	174
22.3.	Events	176
22.3.1.	ProblemEvent Method	176
22.3.2.	ReceivedEvent Method	176
22.3.3.	ReceivedRawEvent Method	176
22.3.4.	SentEvent Method	176
22.3.5.	SentRawEvent Method	176
23.	Dorian.Gem Class	178
23.1.	Properties	179
23.1.1.	AlarmModel Property	179
23.1.2.	CallDefProc Property	179
23.1.3.	CommModel Property	179
23.1.4.	CtrlModel Property	179
23.1.5.	EventModel Property	179
23.1.6.	FileName Property	179
23.1.7.	Hsms Property	179
23.1.8.	Json Property	180
23.1.9.	Msg Property	180
23.2.	Methods	181
23.2.1.	Config Method	181
23.2.2.	InvokeAlarm Method	181
23.2.3.	Load Method	181
23.2.4.	Save Method	181
23.2.5.	TimerProc Method	182
23.3.	Events	183
23.3.1.	CommStateChangedEventEvent	183
23.3.2.	ConnectedEventEvent	183
23.3.3.	CtrlStateChangedEventEvent	183
23.3.4.	DisconnectedEventEvent	183
23.3.5.	InvokeEvent Method	183
23.3.6.	ProblemEventEvent	184
23.3.7.	ReceivedEventEvent	184
23.3.8.	SentEventEvent	184
24.	Dorian.Gem.AlarmModelClass Class	185
24.1.	Properties	186
24.1.1.	Count Property	186
24.1.2.	this[int] Property	186
24.2.	Methods	187
24.2.1.	Add Method	187
24.2.2.	Remove Method	187
25.	Dorian.Gem.AlarmModelClass.Alarm Class	188
25.1.	Properties	189
25.1.1.	Code Property	189

25.1.2.	Enable Property	189
25.1.3.	ID Property.....	189
25.1.4.	Text Property	189
26.	Dorian.Gem.CommModelClass Class.....	190
26.1.	Properties.....	191
26.1.1.	CommDelayTimer Property.....	191
26.1.2.	EstablishComm Property	191
26.1.3.	InitialState Property	191
26.1.4.	MDLN Property	191
26.1.5.	SOFTREV Property.....	191
26.1.6.	State Property	191
27.	Dorian.Gem.CommStateChangedEventArgs Class.....	192
27.1.	Properties.....	193
27.1.1.	newState Property.....	193
27.1.2.	prevState Property	193
27.1.3.	sender Property.....	193
28.	Dorian.Gem.ConnectedEventArgs Class	194
28.1.	Properties.....	195
28.1.1.	ip Property	195
28.1.2.	port Property	195
28.1.3.	sender Property.....	195
29.	Dorian.Gem.CtrlModelClass Class	196
29.1.	Properties.....	197
29.1.1.	InitialOfflineState Property	197
29.1.2.	InitialState Property	197
29.1.3.	OfflineState Property	197
29.1.4.	OnlineState Property	197
29.1.5.	State Property	197
30.	Dorian.Gem.CtrlStateChangedEventArgs Class.....	198
30.1.	Properties.....	199
30.1.1.	newState Property.....	199
30.1.2.	prevState Property	199
30.1.3.	sender Property.....	199
31.	Dorian.Gem.Data Class	200
31.1.	Properties.....	201
31.1.1.	alarmModel Property	201
31.1.2.	commModel Property.....	201
31.1.3.	ctrlModel Property	201
31.1.4.	eventModel Property	201
31.1.5.	hsms Property.....	201
32.	Dorian.Gem.Data.AlarmModel Class	202
32.1.	Properties.....	203
32.1.1.	alarms Property.....	203
33.	Dorian.Gem.Data.AlarmModel.Alarm Class	204
33.1.	Properties.....	205
33.1.1.	code Property.....	205
33.1.2.	enable Property.....	205
33.1.3.	id Property	205
33.1.4.	text Property.....	205
34.	Dorian.Gem.Data.AlarmModel.Alarms Class.....	206
35.	Dorian.Gem.Data.CommModel Class	207
35.1.	Properties.....	208
35.1.1.	establishComm Property.....	208
35.1.2.	initialState Property	208
35.1.3.	mdlIn Property.....	208
35.1.4.	softrev Property.....	208
36.	Dorian.Gem.Data.CtrlModel Class.....	209
36.1.	Properties.....	210
36.1.1.	initialOfflineState Property.....	210
36.1.2.	initialState Property	210
36.1.3.	offlineState Property	210
36.1.4.	onlineState Property	210
37.	Dorian.Gem.Data.EventModel Class	211
37.1.	Properties.....	212
37.1.1.	events Property.....	212

37.1.2.	reports Property	212
37.1.3.	variables Property	212
38.	Dorian.Gem.Data.EventModel.Event Class	213
38.1.	Properties.....	214
38.1.1.	definition Property	214
38.1.2.	description Property.....	214
38.1.3.	enable Property.....	214
38.1.4.	id Property	214
38.1.5.	reports Property	214
39.	Dorian.Gem.Data.EventModel.Events Class	215
40.	Dorian.Gem.Data.EventModel.Report Class	216
40.1.	Properties.....	217
40.1.1.	description Property.....	217
40.1.2.	id Property	217
40.1.3.	variables Property	217
41.	Dorian.Gem.Data.EventModel.Reports Class.....	218
42.	Dorian.Gem.Data.EventModel.Variable Class	219
42.1.	Properties.....	220
42.1.1.	defaultValue Property	220
42.1.2.	definition Property	220
42.1.3.	description Property.....	220
42.1.4.	id Property	220
42.1.5.	max Property.....	220
42.1.6.	min Property.....	220
42.1.7.	secsType Property	220
42.1.8.	sml Property.....	221
42.1.9.	type Property.....	221
42.1.10.	unit Property.....	221
43.	Dorian.Gem.Data.EventModel.Variables Class	222
44.	Dorian.Gem.DisconnectedEventArgs Class	223
44.1.	Properties.....	224
44.1.1.	ip Property	224
44.1.2.	port Property	224
44.1.3.	sender Property.....	224
45.	Dorian.Gem.EventModelClass Class	225
45.1.	Properties.....	226
45.1.1.	Events Property.....	226
45.1.2.	Reports Property	226
45.1.3.	Variables Property.....	226
46.	Dorian.Gem.EventModelClass.Event Class	227
46.1.	Properties.....	228
46.1.1.	Count Property	228
46.1.2.	Definition Property.....	228
46.1.3.	Description Property	228
46.1.4.	Enable Property	228
46.1.5.	ID Property.....	228
46.1.6.	Reports Property	228
46.1.7.	this[int] Property	228
46.2.	Methods	230
46.2.1.	AddReport Method	230
46.2.2.	ClearReports Method.....	230
47.	Dorian.Gem.EventModelClass.EventsClass Class	231
47.1.	Properties.....	232
47.1.1.	Count Property	232
47.1.2.	this[int] Property	232
47.2.	Methods	233
47.2.1.	Add Method.....	233
47.2.2.	Remove Method.....	233
48.	Dorian.Gem.EventModelClass.Report Class	234
48.1.	Properties.....	235
48.1.1.	Count Property	235
48.1.2.	Description Property	235
48.1.3.	ID Property.....	235
48.1.4.	this[int] Property	235
48.1.5.	Variables Property.....	235

48.2.	Methods	236
48.2.1.	AddVariable Method	236
48.2.2.	ClearVariables Method	236
49.	Dorian.Gem.EventModelClass.ReportsClass Class	237
49.1.	Properties	238
49.1.1.	Count Property	238
49.1.2.	this[int] Property	238
49.2.	Methods	239
49.2.1.	Add Method	239
49.2.2.	Remove Method	239
50.	Dorian.Gem.EventModelClass.Variable Class	240
50.1.	Properties	241
50.1.1.	DefaultValue Property	241
50.1.2.	Definition Property	241
50.1.3.	Description Property	241
50.1.4.	ID Property	241
50.1.5.	Max Property	241
50.1.6.	Min Property	241
50.1.7.	SecsType Property	241
50.1.8.	Sml Property	242
50.1.9.	Type Property	242
50.1.10.	Unit Property	242
51.	Dorian.Gem.EventModelClass.VariablesClass Class	243
51.1.	Properties	244
51.1.1.	Count Property	244
51.1.2.	this[int] Property	244
51.2.	Methods	245
51.2.1.	Add Method	245
51.2.2.	Remove Method	245
52.	Dorian.Gem.HsmsClass Class	246
52.1.	Properties	247
52.1.1.	AutoReconnect Property	247
52.1.2.	AutoSelectReq Property	247
52.1.3.	AutoSessionID Property	247
52.1.4.	AutoSystemBytes Property	247
52.1.5.	Connect Property	247
52.1.6.	HandleCtrlMsg Property	248
52.1.7.	IP Property	248
52.1.8.	IPv4 Property	248
52.1.9.	Port Property	248
52.1.10.	Selected Property	249
52.1.11.	Server Property	249
52.1.12.	SessionID Property	249
52.1.13.	SystemBytes Property	249
52.1.14.	T3 Property	249
52.1.15.	T5 Property	250
52.1.16.	T6 Property	250
52.1.17.	T7 Property	250
52.1.18.	T8 Property	250
52.2.	Methods	251
52.2.1.	Send Method	251
53.	Dorian.Gem.ProblemEventArgs Class	252
53.1.	Properties	253
53.1.1.	code Property	253
53.1.2.	ip Property	253
53.1.3.	message Property	253
53.1.4.	port Property	253
53.1.5.	sender Property	253
54.	Dorian.Gem.ReceivedEventArgs Class	254
54.1.	Properties	255
54.1.1.	ip Property	255
54.1.2.	message Property	255
54.1.3.	port Property	255
54.1.4.	sender Property	255
55.	Dorian.Gem.SecsIIClass Class	256

55.1.	Properties.....	257
55.1.1.	this[int] Property	257
56.	Dorian.Gem.SentEventArgs Class.....	258
56.1.	Properties.....	259
56.1.1.	ip Property	259
56.1.2.	message Property	259
56.1.3.	port Property	259
56.1.4.	sender Property.....	259
56.1.5.	success Property.....	259
57.	Dorian.GemConfigDlg Class.....	260
57.1.	Properties.....	261
57.1.1.	components Property.....	261
57.1.2.	gem Property	261
57.2.	Methods	262
57.2.1.	~Dorian.GemConfigDlg Method	262
57.2.2.	Dorian.GemConfigDlg Method	262
57.2.3.	InitializeComponent Method.....	262
58.	Dorian.Hsms Class	263
58.1.	Properties.....	264
58.1.1.	AutoReconnect Property.....	264
58.1.2.	AutoSelectReq Property	264
58.1.3.	AutoSessionID Property.....	264
58.1.4.	AutoSystemBytes Property	264
58.1.5.	Clients Property.....	264
58.1.6.	Connect Property	264
58.1.7.	FileName Property	265
58.1.8.	HandleCtrlMsg Property.....	265
58.1.9.	Heartbeat Property	265
58.1.10.	IP Property.....	265
58.1.11.	IPv4 Property	266
58.1.12.	Json Property.....	266
58.1.13.	MDLN Property	266
58.1.14.	Port Property.....	266
58.1.15.	Selected Property.....	266
58.1.16.	Server Property	266
58.1.17.	SessionID Property.....	267
58.1.18.	SOFTREV Property	267
58.1.19.	SystemBytes Property	267
58.1.20.	T3 Property	267
58.1.21.	T5 Property.....	267
58.1.22.	T6 Property	268
58.1.23.	T7 Property	268
58.1.24.	T8 Property	268
58.1.25.	Transactions Property.....	268
58.2.	Methods	269
58.2.1.	Config Method.....	269
58.2.2.	Load Method.....	269
58.2.3.	Save Method.....	269
58.2.4.	Send Method.....	269
58.2.5.	TimerProc Method.....	270
58.3.	Events.....	271
58.3.1.	ConnectedEventEvent	271
58.3.2.	DisconnectedEventEvent	271
58.3.3.	ProblemEventEvent.....	271
58.3.4.	ReceivedEventEvent	271
58.3.5.	SentEventEvent	271
59.	Dorian.Hsms.ClientsClass Class	273
59.1.	Properties.....	274
59.1.1.	Count Property	274
59.1.2.	this[int] Property	274
59.2.	Methods	275
59.2.1.	find Method.....	275
60.	Dorian.Hsms.ClientsClass.Client Class.....	276
60.1.	Properties.....	277
60.1.1.	IP Property.....	277

60.1.2.	Port Property.....	277
60.1.3.	Selected Property.....	277
60.2.	Methods.....	278
60.2.1.	Disconnect Method.....	278
60.2.2.	Send Method.....	278
61.	Dorian.Hsms.ConnectedEventArgs Class.....	279
61.1.	Properties.....	280
61.1.1.	ip Property.....	280
61.1.2.	port Property.....	280
61.1.3.	sender Property.....	280
62.	Dorian.Hsms.Data Class.....	281
62.1.	Properties.....	282
62.1.1.	autoSessionID Property.....	282
62.1.2.	autoSystemBytes Property.....	282
62.1.3.	handleCtrlMessage Property.....	282
62.1.4.	sendSelectReq Property.....	282
62.1.5.	sessionID Property.....	282
62.1.6.	socket Property.....	283
62.1.7.	t3 Property.....	283
62.1.8.	t5 Property.....	283
62.1.9.	t6 Property.....	283
62.1.10.	t7 Property.....	283
62.1.11.	t8 Property.....	283
63.	Dorian.Hsms.Data.Socket Class.....	285
63.1.	Properties.....	286
63.1.1.	ip Property.....	286
63.1.2.	ipv4 Property.....	286
63.1.3.	port Property.....	286
63.1.4.	server Property.....	286
64.	Dorian.Hsms.DisconnectedEventArgs Class.....	287
64.1.	Properties.....	288
64.1.1.	ip Property.....	288
64.1.2.	port Property.....	288
64.1.3.	sender Property.....	288
65.	Dorian.Hsms.ProblemEventArgs Class.....	289
65.1.	Properties.....	290
65.1.1.	code Property.....	290
65.1.2.	ip Property.....	290
65.1.3.	message Property.....	290
65.1.4.	port Property.....	290
65.1.5.	sender Property.....	290
66.	Dorian.Hsms.ReceivedEventArgs Class.....	291
66.1.	Properties.....	292
66.1.1.	ip Property.....	292
66.1.2.	message Property.....	292
66.1.3.	port Property.....	292
66.1.4.	sender Property.....	292
67.	Dorian.Hsms.SentEventArgs Class.....	293
67.1.	Properties.....	294
67.1.1.	ip Property.....	294
67.1.2.	message Property.....	294
67.1.3.	port Property.....	294
67.1.4.	sender Property.....	294
67.1.5.	success Property.....	294
68.	Dorian.Hsms.TransactionsClass Class.....	295
68.1.	Properties.....	296
68.1.1.	Count Property.....	296
68.1.2.	this[int] Property.....	296
69.	Dorian.Hsms.TransactionsClass.Transaction Class.....	297
69.1.	Properties.....	298
69.1.1.	Ctrl Property.....	298
69.1.2.	Msg Property.....	298
70.	Dorian.HsmsConfigDlg Class.....	299
70.1.	Properties.....	300
70.1.1.	components Property.....	300

70.1.2.	hsms Property.....	300
70.2.	Methods.....	301
70.2.1.	~Dorian.HsmsConfigDlg Method.....	301
70.2.2.	Dorian.HsmsConfigDlg Method.....	301
70.2.3.	HsmsConfigDlg_Load Method.....	301
70.2.4.	InitializeComponent Method.....	301
71.	Dorian.Klarf12 Class.....	303
71.1.	Properties.....	304
71.1.1.	Count Property.....	304
71.1.2.	this[int] Property.....	304
71.1.3.	this[string] Property.....	304
71.2.	Methods.....	305
71.2.1.	Dorian.Klarf12 Method.....	305
71.2.2.	Find Method.....	305
71.2.3.	Parse Method.....	305
72.	Dorian.Klarf12.AlignmentImages Class.....	306
72.1.	Properties.....	307
72.1.1.	Count Property.....	307
72.1.2.	this[int] Property.....	307
72.2.	Methods.....	308
72.2.1.	Dorian.Klarf12.AlignmentImages Method.....	308
73.	Dorian.Klarf12.AlignmentImages.DataItem Class.....	309
73.1.	Properties.....	310
73.1.1.	ID Property.....	310
73.1.2.	NumOfImage Property.....	310
73.1.3.	X Property.....	310
73.1.4.	Y Property.....	310
74.	Dorian.Klarf12.AlignmentImageTransforms Class.....	311
74.1.	Properties.....	312
74.1.1.	A11 Property.....	312
74.1.2.	A12 Property.....	312
74.1.3.	A21 Property.....	312
74.1.4.	A22 Property.....	312
74.1.5.	Count Property.....	312
74.1.6.	MarkID Property.....	312
74.2.	Methods.....	313
74.2.1.	Dorian.Klarf12.AlignmentImageTransforms Method.....	313
75.	Dorian.Klarf12.AlignmentPoints Class.....	314
75.1.	Properties.....	315
75.1.1.	Count Property.....	315
75.1.2.	this[int] Property.....	315
75.2.	Methods.....	316
75.2.1.	Dorian.Klarf12.AlignmentPoints Method.....	316
76.	Dorian.Klarf12.AlignmentPoints.DataItem Class.....	317
76.1.	Properties.....	318
76.1.1.	ID Property.....	318
76.1.2.	X Property.....	318
76.1.3.	Y Property.....	318
77.	Dorian.Klarf12.AreaPerTest Class.....	319
77.1.	Properties.....	320
77.1.1.	Value Property.....	320
77.2.	Methods.....	321
77.2.1.	Dorian.Klarf12.AreaPerTest Method.....	321
78.	Dorian.Klarf12.ClassLookup Class.....	322
78.1.	Properties.....	323
78.1.1.	Count Property.....	323
78.1.2.	this[int] Property.....	323
78.2.	Methods.....	324
78.2.1.	Dorian.Klarf12.ClassLookup Method.....	324
79.	Dorian.Klarf12.ClassLookup.DataItem Class.....	325
79.1.	Properties.....	326
79.1.1.	ID Property.....	326
79.1.2.	Name Property.....	326
80.	Dorian.Klarf12.ClusterClassificationList Class.....	327
80.1.	Properties.....	328

80.1.1.	Count Property	328
80.1.2.	this[int] Property	328
80.2.	Methods	329
80.2.1.	Dorian.Klarf12.ClusterClassificationList Method.....	329
81.	Dorian.Klarf12.ClusterClassificationList.DataItem Class	330
81.1.	Properties.....	331
81.1.1.	Classification Property	331
81.1.2.	ID Property.....	331
82.	Dorian.Klarf12.CoordinatesMirrored Class	332
82.1.	Properties.....	333
82.1.1.	Value Property	333
82.2.	Methods	334
82.2.1.	Dorian.Klarf12.CoordinatesMirrored Method.....	334
83.	Dorian.Klarf12.DatabaseAlignmentMarks Class	335
83.1.	Properties.....	336
83.1.1.	Count Property	336
83.1.2.	this[int] Property	336
83.2.	Methods	337
83.2.1.	Dorian.Klarf12.DatabaseAlignmentMarks Method.....	337
84.	Dorian.Klarf12.DatabaseAlignmentMarks.DataItem Class	338
84.1.	Properties.....	339
84.1.1.	ID Property.....	339
84.1.2.	OriginX Property.....	339
84.1.3.	OriginY Property.....	339
84.1.4.	PointX Property.....	339
84.1.5.	PointY Property.....	339
85.	Dorian.Klarf12.DefectClusterSetup Class.....	340
85.1.	Properties.....	341
85.1.1.	Count Property	341
85.1.2.	Value Property	341
85.2.	Methods	342
85.2.1.	Dorian.Klarf12.DefectClusterSetup Method	342
86.	Dorian.Klarf12.DefectClusterSpec Class.....	343
86.1.	Properties.....	344
86.1.1.	Count Property	344
86.1.2.	Value Property	344
86.2.	Methods	345
86.2.1.	Dorian.Klarf12.DefectClusterSpec Method	345
87.	Dorian.Klarf12.DefectList Class	346
87.1.	Properties.....	347
87.1.1.	Count Property	347
87.1.2.	this[int] Property	347
87.2.	Methods	348
87.2.1.	Dorian.Klarf12.DefectList Method.....	348
88.	Dorian.Klarf12.DefectList.DataItem Class	349
88.1.	Properties.....	350
88.1.1.	Count Property	350
88.1.2.	this[string] Property.....	350
88.1.3.	Value Property	350
89.	Dorian.Klarf12.DefectRecordSpec Class.....	351
89.1.	Properties.....	352
89.1.1.	Count Property	352
89.1.2.	this[string] Property.....	352
89.1.3.	Value Property	352
89.2.	Methods	353
89.2.1.	Dorian.Klarf12.DefectRecordSpec Method	353
90.	Dorian.Klarf12.DeviceID Class	354
90.1.	Properties.....	355
90.1.1.	Text Property	355
90.2.	Methods	356
90.2.1.	Dorian.Klarf12.DeviceID Method	356
91.	Dorian.Klarf12.DieOrigin Class.....	357
91.1.	Properties.....	358
91.1.1.	X Property.....	358
91.1.2.	Y Property.....	358

91.2.	Methods	359
91.2.1.	Dorian.Klarf12.DieOrigin Method.....	359
92.	Dorian.Klarf12.DiePitch Class	360
92.1.	Properties.....	361
92.1.1.	X Property.....	361
92.1.2.	Y Property.....	361
92.2.	Methods	362
92.2.1.	Dorian.Klarf12.DiePitch Method	362
93.	Dorian.Klarf12.EndOfFile Class.....	363
93.1.	Methods	364
93.1.1.	Dorian.Klarf12.EndOfFile Method.....	364
94.	Dorian.Klarf12.FileTimestamp Class.....	365
94.1.	Properties.....	366
94.1.1.	DateTime Property	366
94.2.	Methods	367
94.2.1.	Dorian.Klarf12.FileTimestamp Method.....	367
95.	Dorian.Klarf12.FileVersion Class	368
95.1.	Properties.....	369
95.1.1.	Major Property.....	369
95.1.2.	Minor Property.....	369
95.2.	Methods	370
95.2.1.	Dorian.Klarf12.FileVersion Method.....	370
96.	Dorian.Klarf12.InspectedArea Class	371
96.1.	Properties.....	372
96.1.1.	Count Property	372
96.1.2.	this[int] Property	372
96.2.	Methods	373
96.2.1.	Dorian.Klarf12.InspectedArea Method	373
97.	Dorian.Klarf12.InspectedArea.DataItem Class	374
97.1.	Properties.....	375
97.1.1.	OffsetX Property.....	375
97.1.2.	OffsetY Property.....	375
97.1.3.	PitchX Property	375
97.1.4.	PitchY Property	375
97.1.5.	RepeatX Property.....	375
97.1.6.	RepeatY Property.....	375
97.1.7.	SizeX Property	375
97.1.8.	SizeY Property	376
98.	Dorian.Klarf12.InspectedAreaOrigin Class	377
98.1.	Properties.....	378
98.1.1.	X Property.....	378
98.1.2.	Y Property.....	378
98.2.	Methods	379
98.2.1.	Dorian.Klarf12.InspectedAreaOrigin Method.....	379
99.	Dorian.Klarf12.InspectionOrientation Class.....	380
99.1.	Properties.....	381
99.1.1.	Text Property	381
99.2.	Methods	382
99.2.1.	Dorian.Klarf12.InspectionOrientation Method.....	382
100.	Dorian.Klarf12.InspectionStationID Class.....	383
100.1.	Properties.....	384
100.1.1.	ID Property.....	384
100.1.2.	Manufacturer Property.....	384
100.1.3.	Model Property.....	384
100.2.	Methods	385
100.2.1.	Dorian.Klarf12.InspectionStationID Method	385
101.	Dorian.Klarf12.InspectionTest Class.....	386
101.1.	Properties.....	387
101.1.1.	ID Property.....	387
101.2.	Methods	388
101.2.1.	Dorian.Klarf12.InspectionTest Method.....	388
102.	Dorian.Klarf12.LotID Class	389
102.1.	Properties.....	390
102.1.1.	Text Property	390
102.2.	Methods.....	391

102.2.1.	Dorian.Klarf12.LotID Method.....	391
103.	Dorian.Klarf12.LotStatus Class.....	392
103.1.	Properties.....	393
103.1.1.	Failed Property.....	393
103.1.2.	Passed Property.....	393
103.1.3.	Total Property.....	393
103.2.	Methods.....	394
103.2.1.	Dorian.Klarf12.LotStatus Method.....	394
104.	Dorian.Klarf12.OrientationInstructions Class.....	395
104.1.	Properties.....	396
104.1.1.	Text Property.....	396
104.2.	Methods.....	397
104.2.1.	Dorian.Klarf12.OrientationInstructions Method.....	397
105.	Dorian.Klarf12.OrientationMarkLocation Class.....	398
105.1.	Properties.....	399
105.1.1.	Text Property.....	399
105.2.	Methods.....	400
105.2.1.	Dorian.Klarf12.OrientationMarkLocation Method.....	400
106.	Dorian.Klarf12.Record Class.....	401
106.1.	Properties.....	402
106.1.1.	Count Property.....	402
106.1.2.	Name Property.....	402
106.1.3.	this[int] Property.....	402
107.	Dorian.Klarf12.RecordBase Class.....	403
107.1.	Properties.....	404
107.1.1.	Available Property.....	404
108.	Dorian.Klarf12.RemovedDieList Class.....	405
108.1.	Properties.....	406
108.1.1.	Count Property.....	406
108.1.2.	this[int] Property.....	406
108.2.	Methods.....	407
108.2.1.	Dorian.Klarf12.RemovedDieList Method.....	407
109.	Dorian.Klarf12.RemovedDieList.DataItem Class.....	408
109.1.	Properties.....	409
109.1.1.	X Property.....	409
109.1.2.	Y Property.....	409
110.	Dorian.Klarf12.ResultsID Class.....	410
110.1.	Properties.....	411
110.1.1.	Text Property.....	411
110.2.	Methods.....	412
110.2.1.	Dorian.Klarf12.ResultsID Method.....	412
111.	Dorian.Klarf12.ResultTimestamp Class.....	413
111.1.	Properties.....	414
111.1.1.	DateTime Property.....	414
111.2.	Methods.....	415
111.2.1.	Dorian.Klarf12.ResultTimestamp Method.....	415
112.	Dorian.Klarf12.SampleCenterLocation Class.....	416
112.1.	Properties.....	417
112.1.1.	X Property.....	417
112.1.2.	Y Property.....	417
112.2.	Methods.....	418
112.2.1.	Dorian.Klarf12.SampleCenterLocation Method.....	418
113.	Dorian.Klarf12.SampleDieMap Class.....	419
113.1.	Properties.....	420
113.1.1.	Count Property.....	420
113.1.2.	this[int] Property.....	420
113.2.	Methods.....	421
113.2.1.	Dorian.Klarf12.SampleDieMap Method.....	421
114.	Dorian.Klarf12.SampleDieMap.DataItem Class.....	422
114.1.	Properties.....	423
114.1.1.	X Property.....	423
114.1.2.	Y Property.....	423
115.	Dorian.Klarf12.SampleOrientationMarkType Class.....	424
115.1.	Properties.....	425
115.1.1.	Text Property.....	425

115.2.	Methods.....	426
115.2.1.	Dorian.Klarf12.SampleOrientationMarkType Method.....	426
116.	Dorian.Klarf12.SampleSize Class.....	427
116.1.	Properties.....	428
116.1.1.	Value Property.....	428
116.2.	Methods.....	429
116.2.1.	Dorian.Klarf12.SampleSize Method.....	429
117.	Dorian.Klarf12.SampleTestPlan Class.....	430
117.1.	Properties.....	431
117.1.1.	Count Property.....	431
117.1.2.	this[int] Property.....	431
117.2.	Methods.....	432
117.2.1.	Dorian.Klarf12.SampleTestPlan Method.....	432
118.	Dorian.Klarf12.SampleTestPlan.DataItem Class.....	433
118.1.	Properties.....	434
118.1.1.	X Property.....	434
118.1.2.	Y Property.....	434
119.	Dorian.Klarf12.SampleTestReferencePlan Class.....	435
119.1.	Properties.....	436
119.1.1.	Count Property.....	436
119.1.2.	this[int] Property.....	436
119.2.	Methods.....	437
119.2.1.	Dorian.Klarf12.SampleTestReferencePlan Method.....	437
120.	Dorian.Klarf12.SampleTestReferencePlan.DataItem Class.....	438
120.1.	Properties.....	439
120.1.1.	ComparedX Property.....	439
120.1.2.	ComparedY Property.....	439
120.1.3.	SampledX Property.....	439
120.1.4.	SampledY Property.....	439
121.	Dorian.Klarf12.SampleType Class.....	440
121.1.	Properties.....	441
121.1.1.	Text Property.....	441
121.2.	Methods.....	442
121.2.1.	Dorian.Klarf12.SampleType Method.....	442
122.	Dorian.Klarf12.SetupID Class.....	443
122.1.	Properties.....	444
122.1.1.	DateTime Property.....	444
122.1.2.	ID Property.....	444
122.2.	Methods.....	445
122.2.1.	Dorian.Klarf12.SetupID Method.....	445
123.	Dorian.Klarf12.Slot Class.....	446
123.1.	Properties.....	447
123.1.1.	Value Property.....	447
123.2.	Methods.....	448
123.2.1.	Dorian.Klarf12.Slot Method.....	448
124.	Dorian.Klarf12.StepID Class.....	449
124.1.	Properties.....	450
124.1.1.	Text Property.....	450
124.2.	Methods.....	451
124.2.1.	Dorian.Klarf12.StepID Method.....	451
125.	Dorian.Klarf12.SummaryList Class.....	452
125.1.	Properties.....	453
125.1.1.	Count Property.....	453
125.1.2.	this[int] Property.....	453
125.2.	Methods.....	454
125.2.1.	Dorian.Klarf12.SummaryList Method.....	454
126.	Dorian.Klarf12.SummaryList.DataItem Class.....	455
126.1.	Properties.....	456
126.1.1.	Count Property.....	456
126.1.2.	Value Property.....	456
127.	Dorian.Klarf12.SummarySpec Class.....	457
127.1.	Properties.....	458
127.1.1.	Count Property.....	458
127.1.2.	Value Property.....	458
127.2.	Methods.....	459

127.2.1.	Dorian.Klarf12.SummarySpec Method.....	459
128.	Dorian.Klarf12.TestParametersList Class.....	460
128.1.	Properties.....	461
128.1.1.	Count Property	461
128.1.2.	Value Property	461
128.2.	Methods.....	462
128.2.1.	Dorian.Klarf12.TestParametersList Method	462
129.	Dorian.Klarf12.TestParametersSpec Class	463
129.1.	Properties.....	464
129.1.1.	Count Property	464
129.1.2.	Value Property	464
129.2.	Methods.....	465
129.2.1.	Dorian.Klarf12.TestParametersSpec Method.....	465
130.	Dorian.Klarf12.TiffFileName Class.....	466
130.1.	Properties.....	467
130.1.1.	Text Property	467
130.2.	Methods.....	468
130.2.1.	Dorian.Klarf12.TiffFileName Method.....	468
131.	Dorian.Klarf12.TiffSpec Class.....	469
131.1.	Properties.....	470
131.1.1.	Alignment Property	470
131.1.2.	Defect Property	470
131.1.3.	Version Property.....	470
131.2.	Methods.....	471
131.2.1.	Dorian.Klarf12.TiffSpec Method	471
132.	Dorian.Klarf12.WaferID Class.....	472
132.1.	Properties.....	473
132.1.1.	Text Property	473
132.2.	Methods.....	474
132.2.1.	Dorian.Klarf12.WaferID Method	474
133.	Dorian.Klarf12.WaferStatus Class	475
133.1.	Properties.....	476
133.1.1.	Text Property	476
133.2.	Methods.....	477
133.2.1.	Dorian.Klarf12.WaferStatus Method	477
134.	Dorian.Klarf18 Class	478
134.1.	Properties.....	479
134.1.1.	Json Property	479
134.2.	Methods.....	480
134.2.1.	Parse Method.....	480
135.	Dorian.Log Class	481
135.1.	Properties.....	482
135.1.1.	Count Property	482
135.1.2.	Enable Property	482
135.1.3.	FileName Property	482
135.1.4.	FileSize Property	482
135.1.5.	Json Property.....	482
135.1.6.	JsonFileName Property	482
135.1.7.	TimeStamp Property.....	483
135.2.	Methods.....	484
135.2.1.	Config Method.....	484
135.2.2.	Load Method.....	484
135.2.3.	Save Method.....	484
135.2.4.	Write Method.....	484
136.	Dorian.LogConfigDlg Class	486
136.1.	Properties.....	487
136.1.1.	components Property.....	487
136.1.2.	log Property.....	487
136.2.	Methods.....	488
136.2.1.	~Dorian.LogConfigDlg Method	488
136.2.2.	Dorian.LogConfigDlg Method.....	488
136.2.3.	InitializeComponent Method.....	488
136.2.4.	LogConfigDlg_Load Method	488
137.	Dorian.Properties.Resources Class	490
137.1.	Properties.....	491

137.1.1.	Add Property	491
137.1.2.	Banner Property	491
137.1.3.	Culture Property	491
137.1.4.	Gear Property	491
137.1.5.	Play Property	491
137.1.6.	Remove Property	491
137.1.7.	ResourceManager Property	491
137.1.8.	Stop Property	492
138.	Dorian.Secsl Class	493
138.1.	Properties	494
138.1.1.	AutoSessionID Property	494
138.1.2.	AutoSystemBytes Property	494
138.1.3.	BaudRate Property	494
138.1.4.	Connect Property	494
138.1.5.	FileName Property	494
138.1.6.	Heartbeat Property	494
138.1.7.	Json Property	495
138.1.8.	Master Property	495
138.1.9.	MDLN Property	495
138.1.10.	Port Property	495
138.1.11.	Retry Property	495
138.1.12.	SessionID Property	495
138.1.13.	SOFTREV Property	496
138.1.14.	SystemBytes Property	496
138.1.15.	T1 Property	496
138.1.16.	T2 Property	496
138.1.17.	T3 Property	496
138.1.18.	T4 Property	497
138.2.	Methods	498
138.2.1.	Config Method	498
138.2.2.	Load Method	498
138.2.3.	Save Method	498
138.2.4.	Send Method	498
138.2.5.	TimerProc Method	499
138.3.	Events	500
138.3.1.	ProblemEventEvent	500
138.3.2.	ReceivedEventEvent	500
138.3.3.	ReceivedRawEventEvent	500
138.3.4.	SentEventEvent	500
138.3.5.	SentRawEventEvent	500
139.	Dorian.Secsl.Data Class	502
139.1.	Properties	503
139.1.1.	autoSessionID Property	503
139.1.2.	autoSystemBytes Property	503
139.1.3.	baudRate Property	503
139.1.4.	master Property	503
139.1.5.	port Property	503
139.1.6.	retry Property	504
139.1.7.	sessionID Property	504
139.1.8.	t1 Property	504
139.1.9.	t2 Property	504
139.1.10.	t3 Property	504
139.1.11.	t4 Property	504
140.	Dorian.Secsl.ProblemEventArgs Class	506
140.1.	Properties	507
140.1.1.	code Property	507
140.1.2.	message Property	507
140.1.3.	sender Property	507
141.	Dorian.Secsl.ReceivedEventArgs Class	508
141.1.	Properties	509
141.1.1.	message Property	509
141.1.2.	sender Property	509
142.	Dorian.Secsl.ReceivedRawEventArgs Class	510
142.1.	Properties	511
142.1.1.	data Property	511

142.1.2.	sender Property.....	511
143.	Dorian.Secsl.SentEventArgs Class.....	512
143.1.	Properties.....	513
143.1.1.	message Property	513
143.1.2.	sender Property.....	513
143.1.3.	success Property.....	513
144.	Dorian.Secsl.SentRawEventArgs Class.....	514
144.1.	Properties.....	515
144.1.1.	data Property	515
144.1.2.	sender Property.....	515
145.	Dorian.Secsl.ConfigDlg Class	516
145.1.	Properties.....	517
145.1.1.	components Property.....	517
145.1.2.	secs Property	517
145.2.	Methods	518
145.2.1.	~Dorian.Secsl.ConfigDlg Method	518
145.2.2.	InitializeComponent Method.....	518
145.2.3.	SecslConfigDlg_Load Method.....	518
146.	Dorian.SecslII Class	520
146.1.	Properties.....	521
146.1.1.	BlockNumber Property.....	521
146.1.2.	DeviceID Property	521
146.1.3.	Ebit Property	521
146.1.4.	Error Property.....	521
146.1.5.	Function Property	521
146.1.6.	Host Property	522
146.1.7.	Hsms Property	522
146.1.8.	Json Property.....	522
146.1.9.	Msg Property.....	522
146.1.10.	Node Property.....	522
146.1.11.	NodeCount Property.....	523
146.1.12.	NodeItem Property	523
146.1.13.	NodeType Property	523
146.1.14.	NodeValue Property	523
146.1.15.	NodeValueHex Property	523
146.1.16.	PType Property	524
146.1.17.	Rbit Property.....	524
146.1.18.	SessionID Property	524
146.1.19.	Sml Property	524
146.1.20.	SmlType Property	525
146.1.21.	SourceID Property.....	525
146.1.22.	Stream Property	525
146.1.23.	SType Property	525
146.1.24.	SuggestedReplyMsg Property	525
146.1.25.	SystemBytes Property	525
146.1.26.	this[string] Property	526
146.1.27.	TransactionID Property.....	526
146.1.28.	Verification Property.....	526
146.1.29.	Wbit Property	526
146.2.	Methods	528
146.2.1.	Reply Method.....	528
146.2.2.	Reply Method.....	528
146.2.3.	Reset Method.....	528
147.	Dorian.SecslII.NodeItemClass Class	529
147.1.	Properties.....	530
147.1.1.	Count Property	530
147.1.2.	Hex Property	530
147.1.3.	Index Property.....	530
147.1.4.	Text Property	530
147.1.5.	this[int] Property	530
147.1.6.	Type Property.....	530
147.2.	Methods	532
147.2.1.	Value Method.....	532
147.2.2.	ValueHex Method.....	532
148.	Dorian.Secsl.Type Enum	533

149.	Dorian.SecsVerify Enum	534
150.	Dorian.Utility Class.....	535
150.1.	Methods	536
150.1.1.	atof Method.....	536
150.1.2.	atofex Method	536
150.1.3.	atoi Method.....	536
150.1.4.	ExpireLicense Method	536
150.1.5.	GetLicense Method	537

2. Dorian.AboutDlg Class

Dorian.AboutDlg is a helper class to show license info.

Properties:

Properties	Description
components	Required designer variable.

Methods:

Methods	Description
~Dorian.AboutDlg	Clean up any resources being used.
InitializeComponent	Required method for Designer support - do not modify the contents of this method with the code editor.

2.1. Properties

2.1.1. components Property

Required designer variable.

Visual C#:

components

2.2. Methods

2.2.1. ~Dorian.AboutDlg Method

Clean up any resources being used.

Visual C#:

```
~Dorian.AboutDlg(bool disposing)
```

Arguments

Name	Description
disposing	true if managed resources should be disposed; otherwise, false.

Return value:

2.2.2. InitializeComponent Method

Required method for Designer support - do not modify the contents of this method with the code editor.

Visual C#:

```
InitializeComponent()
```

Return value:

3. Dorian.Comm Class

Dorian.Comm is a RS232 Serial communication component.

Properties:

Properties	Description
BaudRate	Baud rate
ByteSize	Number of bits for 1 byte
Connect	Open or close serial port
Count	Number of data in the incoming queue
Data	Send or receive data
FileName	JSON settings file name
Json	JSON settings
Parity	Parity bit
Port	Serial port number
StopBit	Stop bit

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Read	Read data
Save	Save JSON settings file.
TimerProc	This function has to be called periodically in order for this class to work.
Write	Write data

Events:

Events	Description
ReceivedEvent	Received event

3.1. Properties

3.1.1. BaudRate Property

Baud rate

Visual C#:

```
int BaudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

3.1.2. ByteSize Property

Number of bits for 1 byte

Visual C#:

```
int ByteSize
```

Remarks:

Gets or sets the number of bits for 1 byte of data.

3.1.3. Connect Property

Open or close serial port

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the serial port connection status.

3.1.4. Count Property

Number of data in the incoming queue

Visual C#:

```
int Count
```

Remarks:

Gets the number of data in the incoming queue.

3.1.5. Data Property

Send or receive data

Visual C#:

```
Byte[] Data
```

Remarks:

Sends or receives the data.

3.1.6. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

3.1.7. Json Property

JSON settings

Visual C#:

```
string Json
```

3.1.8. Parity Property

Parity bit

Visual C#:

```
int Parity
```

Remarks:

Gets or sets the parity bit.

3.1.9. Port Property

Serial port number

Visual C#:

```
int Port
```

Remarks:

Gets or sets the serial port number.

3.1.10. StopBit Property

Stop bit

Visual C#:

```
int StopBit
```

Remarks:

Gets or sets the stop bit.

3.2. Methods

3.2.1. Config Method

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

3.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

3.2.3. Read Method

Read data

Visual C#:

```
public byte[] Read(int length)
```

Remarks:

Reads data from incoming queue.

Arguments

Name	Description
length	Number of bytes to read

Return value:

Received data

3.2.4. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

3.2.5. TimerProc Method

This function has to be called periodically in order for this class to work.

Visual C#:

```
public void TimerProc()
```

3.2.6. Write Method

Write data

Visual C#:

```
public int write(unsigned char[] value)
```

Remarks:

Writes data.

Arguments

Name	Description
value	Data

Return value:

Number of bytes sent.

3.3. Events

3.3.1. ReceivedEventEvent

Received event

Visual C#:

```
public void ReceivedEvent(object sender, ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

4. Dorian.Comm.ReceivedEventArgs Class

Received event arguments

Properties:

Properties	Description
data	Data
sender	Sender of this event

4.1. Properties

4.1.1. data Property

Data

Visual C#:

```
Byte[] data
```

4.1.2. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

5. Dorian.CommConfigDlg Class

Serial port configuration dialog box

Properties:

Properties	Description
comm	Comm object
components	Required designer variable.

Methods:

Methods	Description
~Dorian.CommConfigDlg	Clean up any resources being used.
CommConfigDlg_Load	On load event
Dorian.CommConfigDlg	Constructor
InitializeComponent	Required method for Designer support - do not modify the contents of this method with the code editor.

5.1. Properties

5.1.1. comm Property

Comm object

Visual C#:

Comm **comm**

5.1.2. components Property

Required designer variable.

Visual C#:

components

5.2. Methods

5.2.1. ~Dorian.CommConfigDlg Method

Clean up any resources being used.

Visual C#:

```
~Dorian.CommConfigDlg(bool disposing)
```

Arguments

Name	Description
disposing	true if managed resources should be disposed; otherwise, false.

Return value:

5.2.2. CommConfigDlg_Load Method

On load event

Visual C#:

```
CommConfigDlg_Load(System.Object sender, System.EventArgs e)
```

Arguments

Name	Description
sender	Sender
e	Argument

Return value:

5.2.3. Dorian.CommConfigDlg Method

Constructor

Visual C#:

```
Dorian.CommConfigDlg()
```

Return value:

5.2.4. InitializeComponent Method

Required method for Designer support - do not modify the contents of this method with the code editor.

Visual C#:

```
InitializeComponent()
```

Return value:

6. Dorian.Communication Class

Dorian.Communication component is an assistant product to develop SEMI E37 (HSMS) or SEMI E4 (SECS-I) compliant communication application software. Dorian.Communication component can be used for either equipment side development or host side development. Usually Dorian.Communication component will be used with Dorian.Secsl component.

Properties:

Properties	Description
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
BaudRate	Baud rate
Clients	List of connected clients (passive entity only)
Connect	Connect to the server or start the server.
FileNameHsms	JSON settings file name for HSMS
FileNameSecsl	JSON settings file name for SECS-I
HandleCtrlMsg	Handles HSMS control message automatically
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
Hsms	HSMS or SECS-I
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Json	JSON settings
Master	Master or slave
MDLN	Equipment model name
Port	TCP port number.
Retry	Number of retry
Selected	Communication connection has been "selected" by select request/response transaction.
SerialPort	Serial port number
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SOFTREV	Equipment software revision
SystemBytes	System bytes
T1	T1 timer
T2	T2 timer
T3	T3 timer.
T4	T4 timer
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.
Transactions	List of open transactions

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message to the server
TimerProc	This function has to be called periodically in order for this class to work.

Events:

Events	Description
ConnectedEvent	Connected event (passive entity only)
DisconnectedEvent	Disconnected event
ProblemEvent	Problem event
ReceivedEvent	Received event
ReceivedRawEvent	ReceivedRaw event
SentEvent	Sent event
SentRawEvent	SentRaw event

6.1. Properties

6.1.1. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

6.1.2. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

6.1.3. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

6.1.4. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

6.1.5. BaudRate Property

Baud rate

Visual C#:

```
int BaudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

6.1.6. Clients Property

List of connected clients (passive entity only)

Visual C#:

```
clientsClass clients
```

6.1.7. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the HSMS/SECS-I connection status. If Connect property is set to true where Server property is false (client), Dorian.Communication component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Communication component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened.

If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Communication component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

6.1.8. FileNameHsms Property

JSON settings file name for HSMS

Visual C#:

```
string FileNameHsms
```

6.1.9. FileNameSecsI Property

JSON settings file name for SECS-I

Visual C#:

```
string FileNameSecsI
```

6.1.10. HandleCtrlMsg Property

Handles HSMS control message automatically

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Handles HSMS control message automatically

6.1.11. Heartbeat Property

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

6.1.12. Hsms Property

HSMS or SECS-I

Visual C#:

```
bool Hsms
```

Remarks:

Gets or sets whether Dorian.Communication component is configured as HSMS or SECS-I. Default value is HSMS.

6.1.13. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

6.1.14. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

6.1.15. Json Property

JSON settings

Visual C#:

```
string Json
```

6.1.16. Master Property

Master or slave

Visual C#:

```
bool Master
```

Remarks:

Gets or sets master/slave.

6.1.17. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

6.1.18. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

6.1.19. Retry Property

Number of retry

Visual C#:

```
int Retry
```

Remarks:

Gets or sets the number of retry.

6.1.20. Selected Property

Communication connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of Communication communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Communication component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Communication received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

6.1.21. SerialPort Property

Serial port number

Visual C#:

```
int SerialPort
```

Remarks:

Gets or sets the port number for serial port connection.

6.1.22. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Communication control will run as server. If Server property is false, Dorian.Communication component will run as client.

6.1.23. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

6.1.24. SOFTREV Property

Equipment software revision

Visual C#:

```
string SOFTREV
```

6.1.25. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

6.1.26. T1 Property

T1 timer

Visual C#:

```
double T1
```

Remarks:

Gets or sets the T1 time out for SECS-I in seconds. The default value is 1 second.

6.1.27. **T2 Property**

T2 timer

Visual C#:

```
double T2
```

Remarks:

Gets or sets the T2 time out for SECS-I in seconds. The default value is 10 seconds.

6.1.28. **T3 Property**

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS/SECS-I in seconds. The default value is 45 seconds.

6.1.29. **T4 Property**

T4 timer

Visual C#:

```
double T4
```

Remarks:

Gets or sets the T4 time out for SECS-I in seconds. The default value is 60 seconds.

6.1.30. **T5 Property**

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

6.1.31. **T6 Property**

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

6.1.32. T7 Property

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

6.1.33. T8 Property

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

6.1.34. Transactions Property

List of open transactions

Visual C#:

```
TransactionsClass Transactions
```


6.2. Methods

6.2.1. Config Method

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

6.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

6.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

6.2.4. Send Method

Send the message to the server

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

6.2.5. **TimerProc Method**

This function has to be called periodically in order for this class to work.

Visual C#:

```
public void TimerProc()
```

6.3. Events

6.3.1. ConnectedEventEvent

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(object sender, ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

6.3.2. DisconnectedEventEvent

Disconnected event

Visual C#:

```
public void DisconnectedEvent(object sender, DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

6.3.3. ProblemEventEvent

Problem event

Visual C#:

```
public void ProblemEvent(object sender, ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

6.3.4. ReceivedEventEvent

Received event

Visual C#:

```
public void ReceivedEvent(object sender, ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

6.3.5. ReceivedRawEventEvent

ReceivedRaw event

Visual C#:

```
public void ReceivedRawEvent(object sender, ReceivedRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

6.3.6. **SentEventEvent**

Sent event

Visual C#:

```
public void SentEvent(object sender, SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

6.3.7. **SentRawEventEvent**

SentRaw event

Visual C#:

```
public void SentRawEvent(object sender, SentRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

7. Dorian.Forms.Comm Class

Dorian.Forms.Comm is a RS232 Serial communication component.

Properties:

Properties	Description
BaudRate	Baud rate
ByteSize	Number of bits for 1 byte
Connect	Open or close serial port
Count	Number of data in the incoming queue
Data	Send or receive data
FileName	JSON settings file name
Json	JSON settings
Parity	Parity bit
Port	Serial port number
StopBit	Stop bit

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Read	Read data
Save	Save JSON settings file.
Write	Write data

Events:

Events	Description
ReceivedEvent	Received event

7.1. Properties

7.1.1. BaudRate Property

Baud rate

Visual C#:

```
int BaudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

7.1.2. ByteSize Property

Number of bits for 1 byte

Visual C#:

```
int ByteSize
```

Remarks:

Gets or sets the number of bits for 1 byte of data.

7.1.3. Connect Property

Open or close serial port

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the serial port connection status.

7.1.4. Count Property

Number of data in the incoming queue

Visual C#:

```
int Count
```

Remarks:

Gets the number of data in the incoming queue.

7.1.5. Data Property

Send or receive data

Visual C#:

```
Byte[] Data
```

Remarks:

Sends or receives the data.

7.1.6. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

7.1.7. Json Property

JSON settings

Visual C#:

```
string Json
```

7.1.8. Parity Property

Parity bit

Visual C#:

```
int Parity
```

Remarks:

Gets or sets the parity bit.

7.1.9. Port Property

Serial port number

Visual C#:

```
int Port
```

Remarks:

Gets or sets the serial port number.

7.1.10. StopBit Property

Stop bit

Visual C#:

```
int StopBit
```

Remarks:

Gets or sets the stop bit.

7.2. Methods

7.2.1. Config Method

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

7.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

7.2.3. Read Method

Read data

Visual C#:

```
public byte[] Read(int length)
```

Remarks:

Reads data from incoming queue.

Arguments

Name	Description
length	Number of bytes to read

Return value:

Received data

7.2.4. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

7.2.5. Write Method

Write data

Visual C#:

```
public int write(unsigned char[] value)
```

Remarks:

Writes data.

Arguments

Name	Description
value	Data

Return value:

Number of bytes sent.

7.3. Events

7.3.1. ReceivedEvent Method

Received event

Visual C#:

```
public void ReceivedEvent(System.Object sender, Dorian.Comm.ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

8. Dorian.Forms.Communication Class

Dorian.Forms.Communication component is an assistant product to develop SEMI E37 (HSMS) or SEMI E4 (SECS-I) compliant communication application software. Dorian.Communication component can be used for either equipment side development or host side development. Usually Dorian.Communication component will be used with Dorian.Secsl component.

Properties:

Properties	Description
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
BaudRate	Baud rate
Clients	List of connected clients (passive entity only)
Connect	Connect to the server or start the server.
FileNameHsms	JSON settings file name for HSMS
FileNameSecsl	JSON settings file name for SECS-I
HandleCtrlMsg	Handles HSMS control message automatically
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
Hsms	HSMS or SECS-I
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Json	JSON settings
Master	Master or slave
MDLN	Equipment model name
Port	TCP port number.
Retry	Number of retry
Selected	Communication connection has been "selected" by select request/response transaction.
SerialPort	Serial port number
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SOFTREV	Equipment software revision
SystemBytes	System bytes
T1	T1 timer
T2	T2 timer
T3	T3 timer.
T4	T4 timer
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.
Transactions	List of open transactions

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message to the server

Events:

Events	Description
ConnectedEvent	Connected event (passive entity only)
DisconnectedEvent	Disconnected event
ProblemEvent	Problem event
ReceivedEvent	Received event
ReceivedRawEvent	ReceivedRaw event
SentEvent	Sent event
SentRawEvent	SentRaw event

8.1. Properties

8.1.1. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

8.1.2. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

8.1.3. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

8.1.4. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

8.1.5. BaudRate Property

Baud rate

Visual C#:

```
int BaudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

8.1.6. Clients Property

List of connected clients (passive entity only)

Visual C#:

```
clientsClass clients
```

8.1.7. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the HSMS/SECS-I connection status. If Connect property is set to true where Server property is false (client), Dorian.Communication component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Communication component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened.

If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Communication component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

8.1.8. FileNameHsms Property

JSON settings file name for HSMS

Visual C#:

```
string FileNameHsms
```

8.1.9. FileNameSecsI Property

JSON settings file name for SECS-I

Visual C#:

```
string FileNameSecsI
```

8.1.10. HandleCtrlMsg Property

Handles HSMS control message automatically

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Handles HSMS control message automatically

8.1.11. Heartbeat Property

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

8.1.12. Hsms Property

HSMS or SECS-I

Visual C#:

```
bool Hsms
```

Remarks:

Gets or sets whether Dorian.Communication component is configured as HSMS or SECS-I. Default value is HSMS.

8.1.13. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

8.1.14. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

8.1.15. Json Property

JSON settings

Visual C#:

```
string Json
```

8.1.16. Master Property

Master or slave

Visual C#:

```
bool Master
```

Remarks:

Gets or sets master/slave.

8.1.17. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

8.1.18. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

8.1.19. Retry Property

Number of retry

Visual C#:

```
int Retry
```

Remarks:

Gets or sets the number of retry.

8.1.20. Selected Property

Communication connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of Communication communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Communication component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Communication received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

8.1.21. SerialPort Property

Serial port number

Visual C#:


```
int SerialPort
```

Remarks:

Gets or sets the port number for serial port connection.

8.1.22. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Communication control will run as server. If Server property is false, Dorian.Communication component will run as client.

8.1.23. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

8.1.24. SOFTREV Property

Equipment software revision

Visual C#:

```
string SOFTREV
```

8.1.25. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

8.1.26. T1 Property

T1 timer

Visual C#:

```
double T1
```

Remarks:

Gets or sets the T1 time out for SECS-I in seconds. The default value is 1 second.

8.1.27. **T2 Property**

T2 timer

Visual C#:

```
double T2
```

Remarks:

Gets or sets the T2 time out for SECS-I in seconds. The default value is 10 seconds.

8.1.28. **T3 Property**

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS/SECS-I in seconds. The default value is 45 seconds.

8.1.29. **T4 Property**

T4 timer

Visual C#:

```
double T4
```

Remarks:

Gets or sets the T4 time out for SECS-I in seconds. The default value is 60 seconds.

8.1.30. **T5 Property**

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

8.1.31. **T6 Property**

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

8.1.32. **T7 Property**

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

8.1.33. **T8 Property**

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

8.1.34. **Transactions Property**

List of open transactions

Visual C#:

```
TransactionsClass Transactions
```

8.2. Methods

8.2.1. Config Method

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

8.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

8.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

8.2.4. Send Method

Send the message to the server

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

8.3. Events

8.3.1. ConnectedEventEvent

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(object sender, ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

8.3.2. DisconnectedEventEvent

Disconnected event

Visual C#:

```
public void DisconnectedEvent(object sender, DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

8.3.3. ProblemEventEvent

Problem event

Visual C#:

```
public void ProblemEvent(object sender, ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

8.3.4. ReceivedEventEvent

Received event

Visual C#:

```
public void ReceivedEvent(object sender, ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

8.3.5. ReceivedRawEventEvent

ReceivedRaw event

Visual C#:

```
public void ReceivedRawEvent(object sender, ReceivedRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

8.3.6. **SentEventEvent**

Sent event

Visual C#:

```
public void SentEvent(object sender, SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

8.3.7. **SentRawEventEvent**

SentRaw event

Visual C#:

```
public void SentRawEvent(object sender, SentRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

9. Dorian.Forms.CommunicationView Class

Dorian.Forms.CommunicationView component is an assistant product to develop SEMI E37 (HSMS) or SEMI E4 (SECS-I) compliant communication application software. Dorian.Communication component can be used for either equipment side development or host side development. Usually Dorian.Communication component will be used with Dorian.Secsl component.

Properties:

Properties	Description
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
BaudRate	Baud rate
Clients	List of connected clients (passive entity only)
Connect	Connect to the server or start the server.
FileNameHsms	JSON settings file name for HSMS
FileNameSecsl	JSON settings file name for SECS-I
HandleCtrlMsg	Handles HSMS control message automatically
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
Hsms	HSMS or SECS-I
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Json	JSON settings
Master	Master or slave
MDLN	Equipment model name
Port	TCP port number.
Retry	Number of retry
Selected	Communication connection has been "selected" by select request/response transaction.
SerialPort	Serial port number
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SOFTREV	Equipment software revision
SystemBytes	System bytes
T1	T1 timer
T2	T2 timer
T3	T3 timer.
T4	T4 timer
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.
Transactions	List of open transactions

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message to the server

Events:

Events	Description
ConnectedEvent	Connected event (passive entity only)
DisconnectedEvent	Disconnected event
ProblemEvent	Problem event
ReceivedEvent	Received event
ReceivedRawEvent	ReceivedRaw event
SentEvent	Sent event
SentRawEvent	SentRaw event

9.1. Properties

9.1.1. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

9.1.2. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

9.1.3. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

9.1.4. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

9.1.5. BaudRate Property

Baud rate

Visual C#:

```
int BaudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

9.1.6. Clients Property

List of connected clients (passive entity only)

Visual C#:

```
clientsClass clients
```

9.1.7. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the HSMS/SECS-I connection status. If Connect property is set to true where Server property is false (client), Dorian.Communication component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Communication component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened.

If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Communication component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

9.1.8. FileNameHsms Property

JSON settings file name for HSMS

Visual C#:

```
string FileNameHsms
```

9.1.9. FileNameSecsI Property

JSON settings file name for SECS-I

Visual C#:

```
string FileNameSecsI
```

9.1.10. HandleCtrlMsg Property

Handles HSMS control message automatically

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Handles HSMS control message automatically

9.1.11. Heartbeat Property

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

9.1.12. Hsms Property

HSMS or SECS-I

Visual C#:

```
bool Hsms
```

Remarks:

Gets or sets whether Dorian.Communication component is configured as HSMS or SECS-I. Default value is HSMS.

9.1.13. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

9.1.14. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

9.1.15. Json Property

JSON settings

Visual C#:

```
string Json
```

9.1.16. Master Property

Master or slave

Visual C#:

```
bool Master
```

Remarks:

Gets or sets master/slave.

9.1.17. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

9.1.18. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

9.1.19. Retry Property

Number of retry

Visual C#:

```
int Retry
```

Remarks:

Gets or sets the number of retry.

9.1.20. Selected Property

Communication connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of Communication communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Communication component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Communication received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

9.1.21. SerialPort Property

Serial port number

Visual C#:

```
int SerialPort
```

Remarks:

Gets or sets the port number for serial port connection.

9.1.22. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Communication control will run as server. If Server property is false, Dorian.Communication component will run as client.

9.1.23. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

9.1.24. SOFTREV Property

Equipment software revision

Visual C#:

```
string SOFTREV
```

9.1.25. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

9.1.26. T1 Property

T1 timer

Visual C#:

```
double T1
```

Remarks:

Gets or sets the T1 time out for SECS-I in seconds. The default value is 1 second.

9.1.27. **T2 Property**

T2 timer

Visual C#:

```
double T2
```

Remarks:

Gets or sets the T2 time out for SECS-I in seconds. The default value is 10 seconds.

9.1.28. **T3 Property**

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS/SECS-I in seconds. The default value is 45 seconds.

9.1.29. **T4 Property**

T4 timer

Visual C#:

```
double T4
```

Remarks:

Gets or sets the T4 time out for SECS-I in seconds. The default value is 60 seconds.

9.1.30. **T5 Property**

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

9.1.31. **T6 Property**

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

9.1.32. **T7 Property**

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

9.1.33. **T8 Property**

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

9.1.34. **Transactions Property**

List of open transactions

Visual C#:

```
TransactionsClass Transactions
```


9.2. Methods

9.2.1. Config Method

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

9.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

9.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

9.2.4. Send Method

Send the message to the server

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

9.3. Events

9.3.1. ConnectedEventEvent

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(object sender, ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

9.3.2. DisconnectedEventEvent

Disconnected event

Visual C#:

```
public void DisconnectedEvent(object sender, DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

9.3.3. ProblemEventEvent

Problem event

Visual C#:

```
public void ProblemEvent(object sender, ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

9.3.4. ReceivedEventEvent

Received event

Visual C#:

```
public void ReceivedEvent(object sender, ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

9.3.5. ReceivedRawEventEvent

ReceivedRaw event

Visual C#:

```
public void ReceivedRawEvent(object sender, ReceivedRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

9.3.6. **SentEventEvent**

Sent event

Visual C#:

```
public void SentEvent(object sender, SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

9.3.7. **SentRawEventEvent**

SentRaw event

Visual C#:

```
public void SentRawEvent(object sender, SentRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

10. Dorian.Forms.Gem Class

Dorian.Forms.Gem component is an assistant product to develop SEMI E30 (GEM) compliant communication application software. Dorian.Forms.Gem component can be used for equipment side development.

Properties:

Properties	Description
AlarmModel	GEM alarm model object
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
CallDefProc	Call DefProc() after Received event
Clients	List of connected clients (passive entity only)
CommModel	GEM communication model object
Connect	Connect to the server or start the server.
CtrlModel	GEM control model object
EventModel	GEM event model object
FileName	JSON settings file name
HandleCtrlMsg	IP v4 or IP v6.
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Json	JSON settings
Msg	SECS-II object
Port	TCP port number.
Selected	HSMS connection has been "selected" by select request/response transaction.
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SystemBytes	System bytes
T3	T3 timer.
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.

Methods:

Methods	Description
Config	Open configuration dialog box
InvokeAlarm	Invoke alarm.
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message

Events:

Events	Description
CommStateChangedEvent	Communication state changed event
ConnectedEvent	Connected event (passive entity only)
CtrlStateChangedEvent	Control state changed event
DisconnectedEvent	Disconnected event
InvokeEvent	Invoke event.
ProblemEvent	Problem event
ReceivedEvent	Received event
SentEvent	Sent event

10.1. Properties

10.1.1. AlarmModel Property

GEM alarm model object

Visual C#:

```
AlarmModelClass AlarmModel
```

10.1.2. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

10.1.3. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

10.1.4. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

10.1.5. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

10.1.6. CallDefProc Property

Call DefProc() after Received event

Visual C#:

```
bool CallDefProc
```

10.1.7. Clients Property

List of connected clients (passive entity only)

Visual C#:

```
ClientsClass Clients
```

10.1.8. CommModel Property

GEM communication model object

Visual C#:

```
CommModelClass CommModel
```

10.1.9. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the HSMS connection status. If Connect property is set to true where Server property is false (client), Dorian.Forms.Hsms component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Forms.Hsms component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened.

If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Forms.Hsms component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

10.1.10. CtrlModel Property

GEM control model object

Visual C#:

```
CtrlModelClass CtrlModel
```

10.1.11. EventModel Property

GEM event model object

Visual C#:

```
EventModelClass EventModel
```

10.1.12. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

10.1.13. HandleCtrlMsg Property

IP v4 or IP v6.

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Set true if IP v4. Set false if IP v6.

10.1.14. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

10.1.15. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

10.1.16. Json Property

JSON settings

Visual C#:

```
string Json
```

10.1.17. Msg Property

SECS-II object

Visual C#:

```
SecsIIClass Msg
```

10.1.18. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

10.1.19. Selected Property

HSMS connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of HSMS communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Forms.Hsms component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Forms.Hsms received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

10.1.20. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Forms.Hsms control will run as server. If Server property is false, Dorian.Hsms component will run as client.

10.1.21. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

10.1.22. **SystemBytes Property**

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

10.1.23. **T3 Property**

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS in seconds. The default value is 45 seconds.

10.1.24. **T5 Property**

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

10.1.25. **T6 Property**

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

10.1.26. **T7 Property**

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

10.1.27. T8 Property

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

10.2. Methods**10.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

10.2.2. InvokeAlarm Method

Invoke alarm.

Visual C#:

```
public bool InvokeAlarm(int alid, bool occur)
```

Arguments

Name	Description
alid	ALID
occur	Alarm occurred or resolved

Return value:

Returns true if successful. Otherwise, false.

10.2.3. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

10.2.4. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

10.2.5. Send Method

Send the message

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

10.3. Events**10.3.1. CommStateChangedEvent Method**

Communication state changed event

Visual C#:

```
public void CommStateChangedEvent(System.Object sender, Dorian.Gem.CommStateChangedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

10.3.2. ConnectedEvent Method

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(System.Object sender, Dorian.Gem.ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

10.3.3. CtrlStateChangedEvent Method

Control state changed event

Visual C#:

```
public void CtrlStateChangedEvent(System.Object sender, Dorian.Gem.CtrlStateChangedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

10.3.4. DisconnectedEvent Method

Disconnected event

Visual C#:

```
public void DisconnectedEvent(System.Object sender, Dorian.Gem.DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

10.3.5. InvokeEvent Method

Invoke event.

Visual C#:

```
public bool InvokeEvent(int ceid)
```

Arguments

Name	Description
ceid	CEID

Return value:

Returns true if successful. Otherwise, false.

10.3.6. ProblemEvent Method

Problem event

Visual C#:

```
public void ProblemEvent(System.Object sender, Dorian.Gem.ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

10.3.7. ReceivedEvent Method

Received event

Visual C#:

```
public void ReceivedEvent(System.Object sender, Dorian.Gem.ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

10.3.8. SentEvent Method

Sent event

Visual C#:

```
public void SentEvent(System.Object sender, Dorian.Gem.SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

11.Dorian.Forms.GemView Class

Dorian.Forms.GemView component is an assistant product to develop SEMI E30 (GEM) compliant communication application software. Dorian.Forms.Gem component can be used for equipment side development.

Properties:

Properties	Description
AlarmModel	GEM alarm model object
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
CallDefProc	Call DefProc() after Received event
Clients	List of connected clients (passive entity only)
CommModel	GEM communication model object
Connect	Connect to the server or start the server.
CtrlModel	GEM control model object
EventModel	GEM event model object
FileName	JSON settings file name
HandleCtrlMsg	IP v4 or IP v6.
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Json	JSON settings
Msg	SECS-II object
Port	TCP port number.
Selected	HSMS connection has been "selected" by select request/response transaction.
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SystemBytes	System bytes
T3	T3 timer.
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.

Methods:

Methods	Description
Config	Open configuration dialog box
InvokeAlarm	Invoke alarm.
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message

Events:

Events	Description
CommStateChangedEvent	Communication state changed event
ConnectedEvent	Connected event (passive entity only)
CtrlStateChangedEvent	Control state changed event
DisconnectedEvent	Disconnected event
InvokeEvent	Invoke event.
ProblemEvent	Problem event
ReceivedEvent	Received event
SentEvent	Sent event

11.1. Properties

11.1.1. AlarmModel Property

GEM alarm model object

Visual C#:

```
AlarmModelClass AlarmModel
```

11.1.2. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

11.1.3. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

11.1.4. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

11.1.5. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

11.1.6. CallDefProc Property

Call DefProc() after Received event

Visual C#:

```
bool CallDefProc
```

11.1.7. Clients Property

List of connected clients (passive entity only)

Visual C#:

```
ClientsClass Clients
```

11.1.8. CommModel Property

GEM communication model object

Visual C#:

```
CommModelClass CommModel
```

11.1.9. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the HSMS connection status. If Connect property is set to true where Server property is false (client), Dorian.Forms.Hsms component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Forms.Hsms component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened.

If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Forms.Hsms component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

11.1.10. CtrlModel Property

GEM control model object

Visual C#:

```
CtrlModelClass CtrlModel
```

11.1.11. EventModel Property

GEM event model object

Visual C#:

```
EventModelClass EventModel
```

11.1.12. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

11.1.13. HandleCtrlMsg Property

IP v4 or IP v6.

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Set true if IP v4. Set false if IP v6.

11.1.14. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

11.1.15. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

11.1.16. Json Property

JSON settings

Visual C#:

```
string Json
```

11.1.17. Msg Property

SECS-II object

Visual C#:

```
SecsIIClass Msg
```

11.1.18. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

11.1.19. Selected Property

HSMS connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of HSMS communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Forms.Hsms component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Forms.Hsms received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

11.1.20. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Forms.Hsms control will run as server. If Server property is false, Dorian.Hsms component will run as client.

11.1.21. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

11.1.22. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

11.1.23. T3 Property

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS in seconds. The default value is 45 seconds.

11.1.24. T5 Property

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

11.1.25. T6 Property

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

11.1.26. T7 Property

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

11.1.27. T8 Property

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

11.2. Methods**11.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

11.2.2. InvokeAlarm Method

Invoke alarm.

Visual C#:

```
public bool InvokeAlarm(int alid, bool occur)
```

Arguments

Name	Description
alid	ALID
occur	Alarm occurred or resolved

Return value:

Returns true if successful. Otherwise, false.

11.2.3. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

11.2.4. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

11.2.5. Send Method

Send the message

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

11.3. Events**11.3.1. CommStateChangedEvent Method**

Communication state changed event

Visual C#:

```
public void CommStateChangedEvent(System.Object sender, Dorian.Gem.CommStateChangedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

11.3.2. ConnectedEvent Method

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(System.Object sender, Dorian.Gem.ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

11.3.3. CtrlStateChangedEvent Method

Control state changed event

Visual C#:

```
public void CtrlStateChangedEvent(System.Object sender, Dorian.Gem.CtrlStateChangedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

11.3.4. DisconnectedEvent Method

Disconnected event

Visual C#:

```
public void DisconnectedEvent(System.Object sender, Dorian.Gem.DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

11.3.5. InvokeEvent Method

Invoke event.

Visual C#:

```
public bool InvokeEvent(int ceid)
```

Arguments

Name	Description
ceid	CEID

Return value:

Returns true if successful. Otherwise, false.

11.3.6. ProblemEvent Method

Problem event

Visual C#:

```
public void ProblemEvent(System.Object sender, Dorian.Gem.ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

11.3.7. ReceivedEvent Method

Received event

Visual C#:

```
public void ReceivedEvent(System.Object sender, Dorian.Gem.ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

11.3.8. SentEvent Method

Sent event

Visual C#:

```
public void SentEvent(System.Object sender, Dorian.Gem.SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

12. Dorian.Forms.Hsms Class

Dorian.Forms.Hsms component is an assistant product to develop SEMI E37 (HSMS) compliant communication application software. Dorian.Forms.Hsms component can be used for either equipment side development or host side development. Usually Dorian.Forms.Hsms component will be used with Dorian.SecsII component.

Properties:

Properties	Description
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
Clients	List of connected clients (passive entity only)
Connect	Connect to the server or start the server.
FileName	JSON settings file name
HandleCtrlMsg	Handles HSMS control message automatically
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Json	JSON settings
MDLN	Equipment model name
Port	TCP port number.
Selected	HSMS connection has been "selected" by select request/response transaction.
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SOFTREV	Equipment software revision
SystemBytes	System bytes
T3	T3 timer.
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.
Transactions	List of open transactions

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message to the server

Events:

Events	Description
ConnectedEvent	Connected event (passive entity only)
DisconnectedEvent	Disconnected event
ProblemEvent	Problem event
ReceivedEvent	Received event
SentEvent	Sent event

12.1. Properties

12.1.1. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

12.1.2. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

12.1.3. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

12.1.4. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

12.1.5. Clients Property

List of connected clients (passive entity only)

Visual C#:

```
ClientsClass Clients
```

12.1.6. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the HSMS connection status. If Connect property is set to true where Server property is false (client), Dorian.Forms.Hsms component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Forms.Hsms component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened.

If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Forms.Hsms component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

12.1.7. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

12.1.8. HandleCtrlMsg Property

Handles HSMS control message automatically

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Handles HSMS control message automatically

12.1.9. Heartbeat Property

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

12.1.10. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

12.1.11. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

12.1.12. Json Property

JSON settings

Visual C#:

```
string Json
```

12.1.13. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

12.1.14. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

12.1.15. Selected Property

HSMS connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of HSMS communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Forms.Hsms component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Forms.Hsms received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

12.1.16. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Forms.Hsms control will run as server. If Server property is false, Dorian.Hsms component will run as client.

12.1.17. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

12.1.18. SOFTREV Property

Equipment software revision

Visual C#:

```
string SOFTREV
```

12.1.19. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

12.1.20. T3 Property

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS in seconds. The default value is 45 seconds.

12.1.21. T5 Property

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

12.1.22. T6 Property

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

12.1.23. T7 Property

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

12.1.24. T8 Property

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

12.1.25. Transactions Property

List of open transactions

Visual C#:

```
TransactionsClass Transactions
```


12.2. Methods**12.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

12.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

12.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

12.2.4. Send Method

Send the message to the server

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

12.3. Events**12.3.1. ConnectedEvent Method**

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(System.Object sender, Dorian.Hsms.ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

12.3.2. DisconnectedEvent Method

Disconnected event

Visual C#:

```
public void DisconnectedEvent(System.Object sender, Dorian.Hsms.DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

12.3.3. ProblemEvent Method

Problem event

Visual C#:

```
public void ProblemEvent(System.Object sender, Dorian.Hsms.ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

12.3.4. ReceivedEvent Method

Received event

Visual C#:

```
public void ReceivedEvent(System.Object sender, Dorian.Hsms.ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

12.3.5. SentEvent Method

Sent event

Visual C#:

```
public void SentEvent(System.Object sender, Dorian.Hsms.SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

13. Dorian.Forms.HsmsView Class

Dorian.Forms.HsmsView component is an assistant product to develop SEMI E37 (HSMS) compliant communication application software. Dorian.Forms.Hsms component can be used for either equipment side development or host side development. Usually Dorian.Forms.Hsms component will be used with Dorian.SecsII component.

Properties:

Properties	Description
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
Clients	List of connected clients (passive entity only)
Connect	Connect to the server or start the server.
FileName	JSON settings file name
HandleCtrlMsg	Handles HSMS control message automatically
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Json	JSON settings
MDLN	Equipment model name
Port	TCP port number.
Selected	HSMS connection has been "selected" by select request/response transaction.
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SOFTREV	Equipment software revision
SystemBytes	System bytes
T3	T3 timer.
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.
Transactions	List of open transactions

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message to the server

Events:

Events	Description
ConnectedEvent	Connected event (passive entity only)
DisconnectedEvent	Disconnected event
ProblemEvent	Problem event
ReceivedEvent	Received event
SentEvent	Sent event

13.1. Properties

13.1.1. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

13.1.2. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

13.1.3. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

13.1.4. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

13.1.5. Clients Property

List of connected clients (passive entity only)

Visual C#:

```
ClientsClass Clients
```

13.1.6. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the HSMS connection status. If Connect property is set to true where Server property is false (client), Dorian.Forms.Hsms component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Forms.Hsms component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened.

If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Forms.Hsms component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

13.1.7. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

13.1.8. HandleCtrlMsg Property

Handles HSMS control message automatically

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Handles HSMS control message automatically

13.1.9. Heartbeat Property

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

13.1.10. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

13.1.11. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

13.1.12. Json Property

JSON settings

Visual C#:

```
string Json
```

13.1.13. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

13.1.14. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

13.1.15. Selected Property

HSMS connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of HSMS communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Forms.Hsms component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Forms.Hsms received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

13.1.16. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Forms.Hsms control will run as server. If Server property is false, Dorian.Hsms component will run as client.

13.1.17. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

13.1.18. SOFTREV Property

Equipment software revision

Visual C#:

```
string SOFTREV
```

13.1.19. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

13.1.20. T3 Property

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS in seconds. The default value is 45 seconds.

13.1.21. T5 Property

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

13.1.22. **T6 Property**

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

13.1.23. **T7 Property**

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

13.1.24. **T8 Property**

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

13.1.25. **Transactions Property**

List of open transactions

Visual C#:

```
TransactionsClass Transactions
```

13.2. Methods**13.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

13.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

13.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

13.2.4. Send Method

Send the message to the server

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

13.3. Events**13.3.1. ConnectedEvent Method**

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(System.Object sender, Dorian.Hsms.ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

13.3.2. DisconnectedEvent Method

Disconnected event

Visual C#:

```
public void DisconnectedEvent(System.Object sender, Dorian.Hsms.DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

13.3.3. ProblemEvent Method

Problem event

Visual C#:

```
public void ProblemEvent(System.Object sender, Dorian.Hsms.ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

13.3.4. ReceivedEvent Method

Received event

Visual C#:

```
public void ReceivedEvent(System.Object sender, Dorian.Hsms.ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

13.3.5. SentEvent Method

Sent event

Visual C#:

```
public void SentEvent(System.Object sender, Dorian.Hsms.SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

14. Dorian.Forms.Klarf12 Class

Dorian.Forms.Klarf12 component is an assistant product to develop application software using KLARF version 1.2.

Properties:

Properties	Description
Count	Number of records
this[int]	Syntax sugar to access the record
this[string]	Syntax sugar to access the record

Methods:

Methods	Description
Find	Search record
Parse	Parse KLARF file

14.1. Properties

14.1.1. Count Property

Number of records

Visual C#:

```
int Count
```

Remarks:

Gets the number of records in the KLARF file.

14.1.2. this[int] Property

Syntax sugar to access the record

Visual C#:

```
this[int]
```

14.1.3. this[string] Property

Syntax sugar to access the record

Visual C#:

```
this[string]
```


14.2. Methods

14.2.1. Find Method

Search record

Visual C#:

```
public int Find(string record, int from, bool forward)
```

Arguments

Name	Description
record	Record name
from	Index where searching starts
forward	true = forward search, false = backward search

Return value:

14.2.2. Parse Method

Parse KLARF file

Visual C#:

```
public bool Parse(string klarfText)
```

Remarks:

Parse the KLARF file content.

Arguments

Name	Description
klarfText	KLARF file content

Return value:

Returns true if successful. Otherwise, false.

15. Dorian.Forms.Klarf12Map Class

Dorian.Forms.Klarf12Map component is an assistant product to develop application software using KLARF version 1.2. It shows defect map.

Properties:

Properties	Description
Index	Record index
klarf	Dorian.KLARF12 object
XY	Use X and Y, when REL is also available.

15.1. Properties

15.1.1. Index Property

Record index

Visual C#:

```
int Index
```

15.1.2. klarf Property

Dorian.KLARF12 object

Visual C#:

```
Klarf12 klarf
```

15.1.3. XY Property

Use X and Y, when REL is also available.

Visual C#:

```
bool XY
```

16. Dorian.Forms.Klarf12Table Class

Dorian.Forms.Klarf12TableForm component is an assistant product to develop application software using KLARF version 1.2.

Properties:

Properties	Description
Index	Record index
klarf	Dorian.KLARF12 object

16.1. Properties

16.1.1. Index Property

Record index

Visual C#:

```
int Index
```

16.1.2. klarf Property

Dorian.KLARF12 object

Visual C#:

```
Klarf12 klarf
```

17. Dorian.Forms.Log Class

Dorian.Forms.Log is a log file component.

Properties:

Properties	Description
Count	Number of backup file
Enable	Enable or disable logging
FileName	Log file name
FileSize	Log file size
Json	JSON settings
JsonFileName	JSON settings file name
TimeStamp	Enable or disable time stamp

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Write	Write text string

17.1. Properties

17.1.1. Count Property

Number of backup file

Visual C#:

```
int Count
```

Remarks:

Gets or sets the number of backup file.

17.1.2. Enable Property

Enable or disable logging

Visual C#:

```
bool Enable
```

Remarks:

Enables or disables logging feature.

17.1.3. FileName Property

Log file name

Visual C#:

```
string FileName
```

Remarks:

Gets or sets the log file name.

17.1.4. FileSize Property

Log file size

Visual C#:

```
int FileSize
```

Remarks:

Gets or sets the log file size.

17.1.5. Json Property

JSON settings

Visual C#:

```
string Json
```

17.1.6. JsonFileName Property

JSON settings file name

Visual C#:

```
string JsonFileName
```

17.1.7. **TimeStamp Property**

Enable or disable time stamp

Visual C#:

```
bool TimeStamp
```

Remarks:

Enables or disables time stamp in the log file.

17.2. Methods**17.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

17.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

17.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

17.2.4. Write Method

Write text string

Visual C#:

```
public void write(string value)
```

Remarks:

Writes text string to log file.

Arguments

Name	Description
value	Text string

18. Dorian.Forms.LogView Class

Dorian.Forms.LogView is a log file component.

Properties:

Properties	Description
Count	Number of backup file
Enable	Enable or disable logging
FileName	Log file name
FileSize	Log file size
Json	JSON settings
JsonFileName	JSON settings file name
TimeStamp	Enable or disable time stamp

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Write	Write text string

18.1. Properties

18.1.1. Count Property

Number of backup file

Visual C#:

```
int Count
```

Remarks:

Gets or sets the number of backup file.

18.1.2. Enable Property

Enable or disable logging

Visual C#:

```
bool Enable
```

Remarks:

Enables or disables logging feature.

18.1.3. FileName Property

Log file name

Visual C#:

```
string FileName
```

Remarks:

Gets or sets the log file name.

18.1.4. FileSize Property

Log file size

Visual C#:

```
int FileSize
```

Remarks:

Gets or sets the log file size.

18.1.5. Json Property

JSON settings

Visual C#:

```
string Json
```

18.1.6. JsonFileName Property

JSON settings file name

Visual C#:

```
string JsonFileName
```

18.1.7. **TimeStamp Property**

Enable or disable time stamp

Visual C#:

```
bool TimeStamp
```

Remarks:

Enables or disables time stamp in the log file.

18.2. Methods**18.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

18.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

18.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

18.2.4. Write Method

Write text string

Visual C#:

```
public void write(string value)
```

Remarks:

Writes text string to log file.

Arguments

Name	Description
value	Text string

19. Dorian.Forms.SecsI Class

Dorian.Forms.SecsI is a component for SEMI E.4 (SECS-I) communications.

Properties:

Properties	Description
AutoSessionID	Automatically adjust device ID
AutoSystemBytes	Automatically adjust system bytes
BaudRate	Baud rate
Connect	Open or close serial port
FileName	JSON settings file name
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
Json	JSON settings
Master	Master or slave
MDLN	Equipment model name
Port	Serial port number
Retry	Number of retry
SessionID	Device ID
SOFTREV	Equipment software revision
SystemBytes	System bytes
T1	T1 timer
T2	T2 timer
T3	T3 timer
T4	T4 timer

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message

Events:

Events	Description
ProblemEvent	Problem event
ReceivedEvent	Received event
ReceivedRawEvent	ReceivedRaw event
SentEvent	Sent event
SentRawEvent	SentRaw event

Properties

19.1.1. **AutoSessionID Property**

Automatically adjust device ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust device ID when sending.

19.1.2. **AutoSystemBytes Property**

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

19.1.3. **BaudRate Property**

Baud rate

Visual C#:

```
int BaudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

19.1.4. **Connect Property**

Open or close serial port

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the SECS-I serial port connection status.

19.1.5. **FileName Property**

JSON settings file name

Visual C#:

```
string FileName
```

19.1.6. **Heartbeat Property**

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

19.1.7. Json Property

JSON settings

Visual C#:

```
string Json
```

19.1.8. Master Property

Master or slave

Visual C#:

```
bool Master
```

Remarks:

Gets or sets master/slave.

19.1.9. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

19.1.10. Port Property

Serial port number

Visual C#:

```
int Port
```

Remarks:

Gets or sets the port number for serial port connection.

19.1.11. Retry Property

Number of retry

Visual C#:

```
int Retry
```

Remarks:

Gets or sets the number of retry.

19.1.12. SessionID Property

Device ID

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the device ID for SECS-I.

19.1.13. **SOFTREV Property**

Equipment software revision

Visual C#:

```
string SOFTREV
```

19.1.14. **SystemBytes Property**

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

19.1.15. **T1 Property**

T1 timer

Visual C#:

```
double T1
```

Remarks:

Gets or sets the T1 time out for SECS-I in seconds. The default value is 1 second.

19.1.16. **T2 Property**

T2 timer

Visual C#:

```
double T2
```

Remarks:

Gets or sets the T2 time out for SECS-I in seconds. The default value is 10 seconds.

19.1.17. **T3 Property**

T3 timer

Visual C#:

double T3

Remarks:

Gets or sets the T3 time out for SECS-I in seconds. The default value is 45 seconds.

19.1.18. T4 Property

T4 timer

Visual C#:

double T4

Remarks:

Gets or sets the T4 time out for SECS-I in seconds. The default value is 60 seconds.

19.2. Methods**19.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

19.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

19.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

19.2.4. Send Method

Send the message

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

19.3. Events**19.3.1. ProblemEvent Method**

Problem event

Visual C#:

```
public void ProblemEvent(System.Object sender, Dorian.SecsI.ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

19.3.2. ReceivedEvent Method

Received event

Visual C#:

```
public void ReceivedEvent(System.Object sender, Dorian.SecsI.ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

19.3.3. ReceivedRawEvent Method

ReceivedRaw event

Visual C#:

```
public void ReceivedRawEvent(System.Object sender, Dorian.SecsI.ReceivedRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

19.3.4. SentEvent Method

Sent event

Visual C#:

```
public void SentEvent(System.Object sender, Dorian.SecsI.SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

19.3.5. SentRawEvent Method

SentRaw event

Visual C#:

```
public void SentRawEvent(System.Object sender, Dorian.SecsI.SentRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

20. Dorian.Forms.SecsII Class

Dorian.Forms.SecsII component is an assistant product to develop SEMI E5 (SECS-II) compliant application software. Dorian.Forms.SecsII component can be used for either equipment side development or host side development. Usually Dorian.Forms.SecsII component can be used with Dorian.Hsms or Dorian.SecsI component.

Properties:

Properties	Description
BlockNumber	Block number (SECS-I)
DeviceID	Device ID (SECS-I)
Ebit	End bit
Error	Parse error on SML parsing
Function	Function number
Host	Host or equipment
Hsms	HSMS or SECS-I
Json	JSON
Msg	Message (binary)
Node	Node
NodeCount	Node count
NodeType	Node type
NodeValue	Node value in decimal
NodeValueHex	Node value in hexadecimal
PType	P-type (HSMS)
Rbit	Reverse bit (SECS-I)
SessionID	Session ID (HSMS)
Sml	SML
SmlType	SML format (under construction)
SourceID	Source ID (SECS-I)
Stream	Stream number
SType	S-type (HSMS)
SuggestedReplyMsg	Suggested reply message
SystemBytes	System bytes
this[string]	Syntax sugar for Node and NodeValue
TransactionID	Transaction ID (SECS-I)
Verification	Verification of Sml or Msg
Wbit	Wait bit

Methods:

Methods	Description
Reply	Set header for the reply
Reply	Set header for the reply
Reset	Reset message

20.1. Properties

20.1.1. BlockNumber Property

Block number (SECS-I)

Visual C#:

```
UInt16 BlockNumber
```

Remarks:

Gets or sets the block number in SECS-II header. This property is used only for SECS-I.

If BlockNumber property is not 1 on received SECS-I message, the message was multi-block message.

BlockNumber property should always be 1, when sending message. If message size exceeds the maximum size of one block, Dorian.SecsI component will automatically convert it into multi-block messages.

20.1.2. DeviceID Property

Device ID (SECS-I)

Visual C#:

```
UInt16 DeviceID
```

Remarks:

Gets or sets the device ID. Device ID is 15 bits starting at second bit of SECS-II header.

Device ID parameter will be reset by calling Reset method.

Device ID and session ID are almost same, but device ID is 15-bit, where session ID is 16-bit.

20.1.3. Ebit Property

End bit

Visual C#:

```
bool Ebit
```

Remarks:

Gets or sets the end bit in SECS-II header. This property is used only for SECS-I.

End bit of incoming SECS-I message is always true. Because Dorian.SecsI component will notify Received event after the final block was received.

20.1.4. Error Property

Parse error on SML parsing

Visual C#:

```
bool Error
```

Remarks:

Gets whether or not SML string processing was failed.

Read-only property.

20.1.5. Function Property

Function number

Visual C#:

Byte Function**Remarks:**

Gets or sets the function number in SECS-II header.

20.1.6. Host Property

Host or equipment

Visual C#:

```
bool Host
```

Remarks:

Gets or sets the role of Dorian.Forms.SecsII component. This property will affect to the result verified by Verify method, and SuggestedReplyMsg property.

20.1.7. Hsms Property

HSMS or SECS-I

Visual C#:

```
bool Hsms
```

Remarks:

Gets or sets whether Dorian.Forms.SecsII component is best match for HSMS or SECS-I. Default value is HSMS.

20.1.8. Json Property

JSON

Visual C#:

```
string Json
```

Remarks:

Gets or sets the message in JSON string. Readin Json property will convert message structure into JSON literal string. See appendix C for the details.

20.1.9. Msg Property

Message (binary)

Visual C#:

```
Byte[] Msg
```

Remarks:

Gets or sets the message data of SECS-II.

20.1.10. Node Property

Node

Visual C#:

```
string Node
```

Remarks:

Gets or sets the node for operation. Node consists of "/" (slash), node number, "[" (left bracket) and "]" (right bracket). Node number is a numeric expression starting at 1. Index number starts at 0. If node is "" (empty), it means root.

20.1.11. NodeCount Property

Node count

Visual C#:

```
int NodeCount
```

Remarks:

Gets or sets the number of sub items. If node type is list, this property means the number of sub node. Otherwise, it means number of array.
Read-only property.

20.1.12. NodeType Property

Node type

Visual C#:

```
SecsType NodeType
```

Remarks:

Gets the node type.
Read-only property.

20.1.13. NodeValue Property

Node value in decimal

Visual C#:

```
string NodeValue
```

Remarks:

Gets or sets the node value. If node is numeric type, the number will be converted into decimal literal expression.
Read-only property.

20.1.14. NodeValueHex Property

Node value in hexadecimal

Visual C#:

```
string NodeValueHex
```

20.1.15. PType Property

P-type (HSMS)

Visual C#:

Byte **PType**

Remarks:

Gets or sets the presentation type in SECS-II header.

This property should always be 0, since SEMI E37 defines only SECS-II type at the moment.

20.1.16. Rbit Property

Reverse bit (SECS-I)

Visual C#:

bool **Rbit**

Remarks:

Gets or sets the reverse bit in SECS-II header.

20.1.17. SessionID Property

Session ID (HSMS)

Visual C#:

UInt16 **SessionID**

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

20.1.18. Sml Property

SML

Visual C#:

string **Sml**

Remarks:

Gets or sets the message in SML string. Reading SML property will convert message structure into SML literal string. It is possible to insert CR (carriage return), LF (line feed), space code, tab code in SML string to set it in SML property. They would be ignored except in some context.

See appendix A for the details.

20.1.19. SmlType Property

SML format (under construction)

Visual C#:

Byte **SmlType**

Remarks:

Gets or sets the SML format.

20.1.20. SourceID Property

Source ID (SECS-I)

Visual C#:UInt16 **SourceID****Remarks:**

Gets or sets the source ID in SECS-II header.

20.1.21. **Stream Property**

Stream number

Visual C#:Byte **Stream****Remarks:**

Gets or sets the stream in SECS-II header.

20.1.22. **SType Property**

S-type (HSMS)

Visual C#:Byte **SType****Remarks:**

Gets or sets the session type in SECS-II header.

20.1.23. **SuggestedReplyMsg Property**

Suggested reply message

Visual C#:Byte[] **SuggestedReplyMsg****Remarks:**

Gets the most appropriate reply message determined after verifying the message structure.

20.1.24. **SystemBytes Property**

System bytes

Visual C#:UInt32 **SystemBytes****Remarks:**

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

20.1.25. **this[string] Property**

Syntax sugar for Node and NodeValue

Visual C#:

```
this[string]
```

Remarks:

Gets or sets the node value in hexadecimal expression.
Read-only property.

20.1.26. TransactionID Property

Transaction ID (SECS-I)

Visual C#:

```
UInt16 TransactionID
```

Remarks:

Gets or sets the transaction ID in SECS-II header.

20.1.27. Verification Property

Verification of Sml or Msg

Visual C#:

```
SecsVerify Verification
```

Remarks:

Verification result of the message structure on setting Sml property or Msg property. If the message is a primary message, suggested reply message will be set to SuggestedReplyMsg property.

20.1.28. Wbit Property

Wait bit

Visual C#:

```
bool wbit
```

Remarks:

Gets or sets the wait bit in SECS-II header.
If primary message requested reply message, wait bit will be true.

20.2. Methods

20.2.1. Reply Method

Set header for the reply

Visual C#:

```
public void Reply(unsigned char[] primary)
```

Remarks:

Initializes SECS-II header as reply message of specified message. If specified message is a HSMS control message, Dorian.Forms.SecsII component will remove message body. Otherwise, message body will not be affected.

Arguments

Name	Description
primary	Primary message

20.2.2. Reply Method

Set header for the reply

Visual C#:

```
public void Reply(Dorian.ISecsII primary)
```

Remarks:

Initializes SECS-II header as reply message of specified message. If specified message is a HSMS control message, Dorian.Forms.SecsII component will remove message body. Otherwise, message body will not be affected.

Arguments

Name	Description
primary	Primary message

20.2.3. Reset Method

Reset message

Visual C#:

```
public void Reset()
```

Remarks:

Initializes internal data structure and parameters.

21. Dorian.Forms.SecsIIView Class

Dorian.Forms.SecsIIView component is an assistant product to develop SEMI E5 (SECS-II) compliant application software. Dorian.Forms.SecsII component can be used for either equipment side development or host side development. Usually Dorian.Forms.SecsII component can be used with Dorian.Hsms or Dorian.SecsI component.

Properties:

Properties	Description
BlockNumber	Block number (SECS-I)
DeviceID	Device ID (SECS-I)
Ebit	End bit
Error	Parse error on SML parsing
Function	Function number
Host	Host or equipment
Hsms	HSMS or SECS-I
Json	JSON
Msg	Message (binary)
Node	Node
NodeCount	Node count
NodeItem	Node item
NodeType	Node type
NodeValue	Node value in decimal
NodeValueHex	Node value in hexadecimal
PType	P-type (HSMS)
Rbit	Reverse bit (SECS-I)
SessionID	Session ID (HSMS)
Sml	SML
SmlType	SML format (under construction)
SourceID	Source ID (SECS-I)
Stream	Stream number
SType	S-type (HSMS)
SuggestedReplyMsg	Suggested reply message
SystemBytes	System bytes
this[string]	Syntax sugar for Node and NodeValue
TransactionID	Transaction ID (SECS-I)
Verification	Verification of Sml or Msg
Wbit	Wait bit

Methods:

Methods	Description
Reply	Set header for the reply
Reply	Set header for the reply
Reset	Reset message

21.1. Properties

21.1.1. BlockNumber Property

Block number (SECS-I)

Visual C#:

```
UInt16 BlockNumber
```

Remarks:

Gets or sets the block number in SECS-II header. This property is used only for SECS-I.

If BlockNumber property is not 1 on received SECS-I message, the message was multi-block message.

BlockNumber property should always be 1, when sending message. If message size exceeds the maximum size of one block, Dorian.SecsI component will automatically convert it into multi-block messages.

21.1.2. DeviceID Property

Device ID (SECS-I)

Visual C#:

```
UInt16 DeviceID
```

Remarks:

Gets or sets the device ID. Device ID is 15 bits starting at second bit of SECS-II header.

Device ID parameter will be reset by calling Reset method.

Device ID and session ID are almost same, but device ID is 15-bit, where session ID is 16-bit.

21.1.3. Ebit Property

End bit

Visual C#:

```
bool Ebit
```

Remarks:

Gets or sets the end bit in SECS-II header. This property is used only for SECS-I.

End bit of incoming SECS-I message is always true. Because Dorian.SecsI component will notify Received event after the final block was received.

21.1.4. Error Property

Parse error on SML parsing

Visual C#:

```
bool Error
```

Remarks:

Gets whether or not SML string processing was failed.

Read-only property.

21.1.5. Function Property

Function number

Visual C#:

Byte Function**Remarks:**

Gets or sets the function number in SECS-II header.

21.1.6. Host Property

Host or equipment

Visual C#:

```
bool Host
```

Remarks:

Gets or sets the role of Dorian.Forms.SecsII component. This property will affect to the result verified by Verify method, and SuggestedReplyMsg property.

21.1.7. Hsms Property

HSMS or SECS-I

Visual C#:

```
bool Hsms
```

Remarks:

Gets or sets whether Dorian.Forms.SecsII component is best match for HSMS or SECS-I. Default value is HSMS.

21.1.8. Json Property

JSON

Visual C#:

```
string Json
```

Remarks:

Gets or sets the message in JSON string. Readin Json property will convert message structure into JSON literal string. See appendix C for the details.

21.1.9. Msg Property

Message (binary)

Visual C#:

```
Byte[] Msg
```

Remarks:

Gets or sets the message data of SECS-II.

21.1.10. Node Property

Node

Visual C#:

```
string Node
```

Remarks:

Gets or sets the node for operation. Node consists of "/" (slash), node number, "[" (left bracket) and "]" (right bracket). Node number is a numeric expression starting at 1. Index number starts at 0. If node is "" (empty), it means root.

21.1.11. NodeCount Property

Node count

Visual C#:

```
int NodeCount
```

Remarks:

Gets or sets the number of sub items. If node type is list, this property means the number of sub node. Otherwise, it means number of array.
Read-only property.

21.1.12. NodeItem Property

Node item

Visual C#:

```
NodeItemClass NodeItem
```

Remarks:

Gets the node item for operation.

21.1.13. NodeType Property

Node type

Visual C#:

```
SecsType NodeType
```

Remarks:

Gets the node type.
Read-only property.

21.1.14. NodeValue Property

Node value in decimal

Visual C#:

```
string NodeValue
```

Remarks:

Gets or sets the node value. If node is numeric type, the number will be converted into decimal literal expression.
Read-only property.

21.1.15. NodeValueHex Property

Node value in hexadecimal

Visual C#:

```
string NodeValueHex
```

21.1.16. PType Property

P-type (HSMS)

Visual C#:

```
Byte PType
```

Remarks:

Gets or sets the presentation type in SECS-II header.

This property should always be 0, since SEMI E37 defines only SECS-II type at the moment.

21.1.17. Rbit Property

Reverse bit (SECS-I)

Visual C#:

```
bool Rbit
```

Remarks:

Gets or sets the reverse bit in SECS-II header.

21.1.18. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

21.1.19. Sml Property

SML

Visual C#:

```
string Sml
```

Remarks:

Gets or sets the message in SML string. Reading SML property will convert message structure into SML literal string. It is possible to insert CR (carriage return), LF (line feed), space code, tab code in SML string to set it in SML property. They would be ignored except in some context.

See appendix A for the details.

21.1.20. SmlType Property

SML format (under construction)

Visual C#:Byte **SmlType****Remarks:**

Gets or sets the SML format.

21.1.21. **SourceID Property**

Source ID (SECS-I)

Visual C#:UInt16 **SourceID****Remarks:**

Gets or sets the source ID in SECS-II header.

21.1.22. **Stream Property**

Stream number

Visual C#:Byte **Stream****Remarks:**

Gets or sets the stream in SECS-II header.

21.1.23. **SType Property**

S-type (HSMS)

Visual C#:Byte **SType****Remarks:**

Gets or sets the session type in SECS-II header.

21.1.24. **SuggestedReplyMsg Property**

Suggested reply message

Visual C#:Byte[] **SuggestedReplyMsg****Remarks:**

Gets the most appropriate reply message determined after verifying the message structure.

21.1.25. **SystemBytes Property**

System bytes

Visual C#:

UInt32 SystemBytes**Remarks:**

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

21.1.26. this[string] Property

Syntax sugar for Node and NodeValue

Visual C#:

```
this[string]
```

Remarks:

Gets or sets the node value in hexadecimal expression.

Read-only property.

21.1.27. TransactionID Property

Transaction ID (SECS-I)

Visual C#:

```
UInt16 TransactionID
```

Remarks:

Gets or sets the transaction ID in SECS-II header.

21.1.28. Verification Property

Verification of Sml or Msg

Visual C#:

```
SecsVerify Verification
```

Remarks:

Verification result of the message structure on setting Sml property or Msg property. If the message is a primary message, suggested reply message will be set to SuggestedReplyMsg property.

21.1.29. Wbit Property

Wait bit

Visual C#:

```
bool wbit
```

Remarks:

Gets or sets the wait bit in SECS-II header.

If primary message requested reply message, wait bit will be true.

21.2. Methods

21.2.1. Reply Method

Set header for the reply

Visual C#:

```
public void Reply(unsigned char[] primary)
```

Remarks:

Initializes SECS-II header as reply message of specified message. If specified message is a HSMS control message, Dorian.Forms.SecsII component will remove message body. Otherwise, message body will not be affected.

Arguments

Name	Description
primary	Primary message

21.2.2. Reply Method

Set header for the reply

Visual C#:

```
public void Reply(Dorian.ISecsII primary)
```

Remarks:

Initializes SECS-II header as reply message of specified message. If specified message is a HSMS control message, Dorian.Forms.SecsII component will remove message body. Otherwise, message body will not be affected.

Arguments

Name	Description
primary	Primary message

21.2.3. Reset Method

Reset message

Visual C#:

```
public void Reset()
```

Remarks:

Initializes internal data structure and parameters.

22. Dorian.Forms.SecsIView Class

Dorian.Forms.SecsIView is a component for SEMI E.4 (SECS-I) communications.

Properties:

Properties	Description
AutoSessionID	Automatically adjust device ID
AutoSystemBytes	Automatically adjust system bytes
BaudRate	Baud rate
Connect	Open or close serial port
FileName	JSON settings file name
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
Json	JSON settings
Master	Master or slave
MDLN	Equipment model name
Port	Serial port number
Retry	Number of retry
SessionID	Device ID
SOFTREV	Equipment software revision
SystemBytes	System bytes
T1	T1 timer
T2	T2 timer
T3	T3 timer
T4	T4 timer

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message

Events:

Events	Description
ProblemEvent	Problem event
ReceivedEvent	Received event
ReceivedRawEvent	ReceivedRaw event
SentEvent	Sent event
SentRawEvent	SentRaw event

22.1. Properties

22.1.1. AutoSessionID Property

Automatically adjust device ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust device ID when sending.

22.1.2. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

22.1.3. BaudRate Property

Baud rate

Visual C#:

```
int BaudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

22.1.4. Connect Property

Open or close serial port

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the SECS-I serial port connection status.

22.1.5. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

22.1.6. Heartbeat Property

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

22.1.7. Json Property

JSON settings

Visual C#:

```
string Json
```

22.1.8. Master Property

Master or slave

Visual C#:

```
bool Master
```

Remarks:

Gets or sets master/slave.

22.1.9. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

22.1.10. Port Property

Serial port number

Visual C#:

```
int Port
```

Remarks:

Gets or sets the port number for serial port connection.

22.1.11. Retry Property

Number of retry

Visual C#:

```
int Retry
```

Remarks:

Gets or sets the number of retry.

22.1.12. SessionID Property

Device ID

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the device ID for SECS-I.

22.1.13. **SOFTREV Property**

Equipment software revision

Visual C#:

```
string SOFTREV
```

22.1.14. **SystemBytes Property**

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

22.1.15. **T1 Property**

T1 timer

Visual C#:

```
double T1
```

Remarks:

Gets or sets the T1 time out for SECS-I in seconds. The default value is 1 second.

22.1.16. **T2 Property**

T2 timer

Visual C#:

```
double T2
```

Remarks:

Gets or sets the T2 time out for SECS-I in seconds. The default value is 10 seconds.

22.1.17. **T3 Property**

T3 timer

Visual C#:

double T3

Remarks:

Gets or sets the T3 time out for SECS-I in seconds. The default value is 45 seconds.

22.1.18. **T4 Property**

T4 timer

Visual C#:

double T4

Remarks:

Gets or sets the T4 time out for SECS-I in seconds. The default value is 60 seconds.

22.2. Methods**22.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

22.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

22.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

22.2.4. Send Method

Send the message

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

22.3. Events**22.3.1. ProblemEvent Method**

Problem event

Visual C#:

```
public void ProblemEvent(System.Object sender, Dorian.SecsI.ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

22.3.2. ReceivedEvent Method

Received event

Visual C#:

```
public void ReceivedEvent(System.Object sender, Dorian.SecsI.ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

22.3.3. ReceivedRawEvent Method

ReceivedRaw event

Visual C#:

```
public void ReceivedRawEvent(System.Object sender, Dorian.SecsI.ReceivedRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

22.3.4. SentEvent Method

Sent event

Visual C#:

```
public void SentEvent(System.Object sender, Dorian.SecsI.SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

22.3.5. SentRawEvent Method

SentRaw event

Visual C#:


```
public void SentRawEvent(System.Object sender, Dorian.SecsI.SentRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

23. Dorian.Gem Class

Dorian.Gem component is an assistant product to develop SEMI E30 (GEM) compliant communication application software. Dorian.Forms.Gem component can be used for equipment side development.

Properties:

Properties	Description
AlarmModel	GEM alarm model object
CallDefProc	Call DefProc() after Received event
CommModel	GEM communication model object
CtrlModel	GEM control model object
EventModel	GEM event model object
FileName	JSON settings file name
Hsms	HSMS object
Json	JSON settings
Msg	SECS-II object

Methods:

Methods	Description
Config	Open configuration dialog box
InvokeAlarm	Invoke alarm.
Load	Load JSON settings file.
Save	Save JSON settings file.
TimerProc	This function has to be called periodically in order for this class to work.

Events:

Events	Description
CommStateChangedEvent	Communication state changed event
ConnectedEvent	Connected event (passive entity only)
CtrlStateChangedEvent	Contrl state changed event
DisconnectedEvent	Disconnected event
InvokeEvent	Invoke event.
ProblemEvent	Problem event
ReceivedEvent	Received event
SentEvent	Sent event

23.1. Properties

23.1.1. AlarmModel Property

GEM alarm model object

Visual C#:

```
AlarmModelClass AlarmModel
```

23.1.2. CallDefProc Property

Call DefProc() after Received event

Visual C#:

```
bool CallDefProc
```

23.1.3. CommModel Property

GEM communication model object

Visual C#:

```
CommModelClass CommModel
```

23.1.4. CtrlModel Property

GEM control model object

Visual C#:

```
CtrlModelClass CtrlModel
```

23.1.5. EventModel Property

GEM event model object

Visual C#:

```
EventModelClass EventModel
```

23.1.6. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

23.1.7. Hsms Property

HSMS object

Visual C#:

```
HsmsClass Hsms
```

23.1.8. **Json Property**

JSON settings

Visual C#:

string **Json**

23.1.9. **Msg Property**

SECS-II object

Visual C#:

SecsIIClass **Msg**

23.2. Methods**23.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

23.2.2. InvokeAlarm Method

Invoke alarm.

Visual C#:

```
public bool InvokeAlarm(int alid, bool occur)
```

Arguments

Name	Description
alid	ALID
occur	Alarm occurred or resolved

Return value:

Returns true if successful. Otherwise, false.

23.2.3. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

23.2.4. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

23.2.5. TimerProc Method

This function has to be called periodically in order for this class to work.

Visual C#:

```
public void TimerProc()
```

23.3. Events**23.3.1. CommStateChangedEventEvent**

Communication state changed event

Visual C#:

```
public void CommStateChangedEvent(object sender, CommStateChangedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

23.3.2. ConnectedEventEvent

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(object sender, ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

23.3.3. CtrlStateChangedEventEvent

Contrl state changed event

Visual C#:

```
public void CtrlStateChangedEvent(object sender, CtrlStateChangedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

23.3.4. DisconnectedEventEvent

Disconnected event

Visual C#:

```
public void DisconnectedEvent(object sender, DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

23.3.5. InvokeEvent Method

Invoke event.

Visual C#:

```
public bool InvokeEvent(int ceid)
```

Arguments

Name	Description
ceid	CEID

Return value:

Returns true if successful. Otherwise, false.

23.3.6. ProblemEventEvent

Problem event

Visual C#:

```
public void ProblemEvent(object sender, ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

23.3.7. ReceivedEventEvent

Received event

Visual C#:

```
public void ReceivedEvent(object sender, ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

23.3.8. SentEventEvent

Sent event

Visual C#:

```
public void SentEvent(object sender, SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

24. Dorian.Gem.AlarmModelClass Class

GEM alarm model

Properties:

Properties	Description
Count	Number of alarms
this[int]	Get alarm object

Methods:

Methods	Description
Add	Add new alarm
Remove	Remove alarm

24.1. Properties

24.1.1. Count Property

Number of alarms

Visual C#:

```
int Count
```

24.1.2. this[int] Property

Get alarm object

Visual C#:

```
this[int]
```

24.2. Methods

24.2.1. Add Method

Add new alarm

Visual C#:

```
public Alarm Add()
```

Return value:

24.2.2. Remove Method

Remove alarm

Visual C#:

```
public bool Remove(int index)
```

Arguments

Name	Description
index	index

Return value:

Returns true if successful. Otherwise, false.

25. Dorian.Gem.AlarmModelClass.Alarm Class

GEM alarm

Properties:

Properties	Description
Code	Alarm code
Enable	Enable or disable alarm
ID	Alarm ID
Text	Alarm text

25.1. Properties

25.1.1. Code Property

Alarm code

Visual C#:

```
Byte Code
```

25.1.2. Enable Property

Enable or disable alarm

Visual C#:

```
bool Enable
```

25.1.3. ID Property

Alarm ID

Visual C#:

```
int ID
```

25.1.4. Text Property

Alarm text

Visual C#:

```
string Text
```

26. Dorian.Gem.CommModelClass Class

GEM communication model

Properties:

Properties	Description
CommDelayTimer	Communication delay timer
EstablishComm	Equipment-initiated establish communication
InitialState	Initial communication state
MDLN	Equipment model name
SOFTREV	Equipment software revision
State	Current communication state

26.1. Properties

26.1.1. CommDelayTimer Property

Communication delay timer

Visual C#:

```
int CommDelayTimer
```

26.1.2. EstablishComm Property

Equipment-initiated establish communication

Visual C#:

```
bool EstablishComm
```

26.1.3. InitialState Property

Initial communication state

Visual C#:

```
GemCommState InitialState
```

26.1.4. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

26.1.5. SOFTREV Property

Equipment software revision

Visual C#:

```
string SOFTREV
```

26.1.6. State Property

Current communication state

Visual C#:

```
GemCommState State
```

27. Dorian.Gem.CommStateChangedEventArgs Class

CommStateChanged event arguments

Properties:

Properties	Description
newState	New state
prevState	Previous state
sender	Sender of this event

27.1. Properties

27.1.1. newState Property

New state

Visual C#:

```
GemCtrlState newState
```

27.1.2. prevState Property

Previous state

Visual C#:

```
GemCtrlState prevState
```

27.1.3. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

28. Dorian.Gem.ConnectedEventArgs Class

Connected event arguments

Properties:

Properties	Description
ip	IP address
port	TCP port number
sender	Sender of this event

28.1. Properties

28.1.1. ip Property

IP address

Visual C#:

```
string ip
```

28.1.2. port Property

TCP port number

Visual C#:

```
string port
```

28.1.3. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

29. Dorian.Gem.CtrlModelClass Class

GEM control model

Properties:

Properties	Description
InitialOfflineState	Initial offline state
InitialState	Initial control state
OfflineState	Default offline state
OnlineState	Default online state
State	Current control state

29.1. Properties

29.1.1. InitialOfflineState Property

Initial offline state

Visual C#:

```
GemCtrlState InitialOfflineState
```

29.1.2. InitialState Property

Initial control state

Visual C#:

```
GemCtrlState InitialState
```

29.1.3. OfflineState Property

Default offline state

Visual C#:

```
GemCtrlState offlineState
```

29.1.4. OnlineState Property

Default online state

Visual C#:

```
GemCtrlState OnlineState
```

29.1.5. State Property

Current control state

Visual C#:

```
GemCtrlState State
```

30. Dorian.Gem.CtrlStateChangedEventArgs Class

CtrlStateChanged event arguments

Properties:

Properties	Description
newState	New state
prevState	Previous state
sender	Sender of this event

30.1. Properties

30.1.1. newState Property

New state

Visual C#:

```
GemCtrlState newState
```

30.1.2. prevState Property

Previous state

Visual C#:

```
GemCtrlState prevState
```

30.1.3. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

31. Dorian.Gem.Data Class

JSON data

Properties:

Properties	Description
alarmModel	GEM alarm model object
commModel	GEM communication model object
ctrlModel	GEM control model object
eventModel	GEM event model object
hsms	HSMS object

31.1. Properties

31.1.1. alarmModel Property

GEM alarm model object

Visual C#:

```
AlarmModel alarmModel
```

31.1.2. commModel Property

GEM communication model object

Visual C#:

```
CommModel commModel
```

31.1.3. ctrlModel Property

GEM control model object

Visual C#:

```
CtrlModel ctrlModel
```

31.1.4. eventModel Property

GEM event model object

Visual C#:

```
EventModel eventModel
```

31.1.5. hsms Property

HSMS object

Visual C#:

```
Data hsms
```

32. Dorian.Gem.Data.AlarmModel Class

GEM alarm model

Properties:

Properties	Description
alarms	GEM alarm list

32.1. Properties

32.1.1. alarms Property

GEM alarm list

Visual C#:

Alarms **alarms**

33. Dorian.Gem.Data.AlarmModel.Alarm Class

GEM alarm

Properties:

Properties	Description
code	Alarm code
enable	Enable or disable alarm
id	Alarm ID
text	Alarm text

33.1. Properties

33.1.1. code Property

Alarm code

Visual C#:

```
Byte code
```

33.1.2. enable Property

Enable or disable alarm

Visual C#:

```
bool enable
```

33.1.3. id Property

Alarm ID

Visual C#:

```
int id
```

33.1.4. text Property

Alarm text

Visual C#:

```
string text
```

34. Dorian.Gem.Data.AlarmModel.Alarms Class

GEM alarm list

35. Dorian.Gem.Data.CommModel Class

GEM communication model

Properties:

Properties	Description
establishComm	Equipment-initiated establish communication
initialState	Initial communication state
mdlIn	Equipment model name
softrev	Equipment software revision

35.1. Properties

35.1.1. establishComm Property

Equipment-initiated establish communication

Visual C#:

```
bool establishComm
```

35.1.2. initialState Property

Initial communication state

Visual C#:

```
GemCommState initialState
```

35.1.3. mdlName Property

Equipment model name

Visual C#:

```
string mdlName
```

35.1.4. softwareRevision Property

Equipment software revision

Visual C#:

```
string softwareRevision
```


36. Dorian.Gem.Data.CtrlModel Class

GEM control model

Properties:

Properties	Description
initialOfflineState	Initial offline state
initialState	Initial control state
offlineState	Default offline state
onlineState	Default online state

36.1. Properties

36.1.1. initialOfflineState Property

Initial offline state

Visual C#:

```
GemCtrlState initialOfflineState
```

36.1.2. initialState Property

Initial control state

Visual C#:

```
GemCtrlState initialState
```

36.1.3. offlineState Property

Default offline state

Visual C#:

```
GemCtrlState offlineState
```

36.1.4. onlineState Property

Default online state

Visual C#:

```
GemCtrlState onlineState
```

37. Dorian.Gem.Data.EventModel Class

GEM event model

Properties:

Properties	Description
events	GEM event list
reports	GEM report list
variables	GEM variable list

37.1. Properties

37.1.1. events Property

GEM event list

Visual C#:

Events **events**

37.1.2. reports Property

GEM report list

Visual C#:

Reports **reports**

37.1.3. variables Property

GEM variable list

Visual C#:

variables **variables**

38. Dorian.Gem.Data.EventModel.Event Class

GEM event

Properties:

Properties	Description
definition	Pre-defined event ID
description	Event description
enable	Enable or disable event
id	Event ID
reports	Report IDs

38.1. Properties

38.1.1. definition Property

Pre-defined event ID

Visual C#:

```
int definition
```

38.1.2. description Property

Event description

Visual C#:

```
string description
```

38.1.3. enable Property

Enable or disable event

Visual C#:

```
bool enable
```

38.1.4. id Property

Event ID

Visual C#:

```
int id
```

38.1.5. reports Property

Report IDs

Visual C#:

```
List<int> reports
```

39. Dorian.Gem.Data.EventModel.Events Class

GEM event list

40. Dorian.Gem.Data.EventModel.Report Class

GEM report

Properties:

Properties	Description
description	Report description
id	Report ID
variables	VIDs

40.1. Properties

40.1.1. description Property

Report description

Visual C#:

```
string description
```

40.1.2. id Property

Report ID

Visual C#:

```
int id
```

40.1.3. variables Property

VIDs

Visual C#:

```
List<int> variables
```

41. Dorian.Gem.Data.EventModel.Reports Class

GEM report list

42. Dorian.Gem.Data.EventModel.Variable Class

GEM variable

Properties:

Properties	Description
defaultValue	Default value of the variable
definition	Pre-defined variable ID
description	Variable description
id	Variable ID
max	Maximum value of the variable
min	Minimum value of the variable
secsType	Variable SECS-II type
sml	Variable SML
type	Variable type
unit	Unit of the variable

42.1. Properties

42.1.1. default Value Property

Default value of the variable

Visual C#:

```
string defaultValue
```

42.1.2. definition Property

Pre-defined variable ID

Visual C#:

```
PredefinedVID definition
```

42.1.3. description Property

Variable description

Visual C#:

```
string description
```

42.1.4. id Property

Variable ID

Visual C#:

```
int id
```

42.1.5. max Property

Maximum value of the variable

Visual C#:

```
string max
```

42.1.6. min Property

Minimum value of the variable

Visual C#:

```
string min
```

42.1.7. secsType Property

Variable SECS-II type

Visual C#:

```
SecsType secsType
```

42.1.8. **sml Property**

Variable SML

Visual C#:

```
string sml
```

42.1.9. **type Property**

Variable type

Visual C#:

```
variableType type
```

42.1.10. **unit Property**

Unit of the variable

Visual C#:

```
string unit
```

43. Dorian.Gem.Data.EventModel.Variables Class

GEM variable list

44. Dorian.Gem.DisconnectedEventArgs Class

Disconnected event arguments

Properties:

Properties	Description
ip	IP address
port	TCP port number
sender	Sender of this event

44.1. Properties

44.1.1. ip Property

IP address

Visual C#:

```
string ip
```

44.1.2. port Property

TCP port number

Visual C#:

```
string port
```

44.1.3. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```


45. Dorian.Gem.EventModelClass Class

GEM event model

Properties:

Properties	Description
Events	GEM event list
Reports	GEM report list
Variables	GEM variable list

45.1. Properties

45.1.1. Events Property

GEM event list

Visual C#:

```
EventsClass Events
```

45.1.2. Reports Property

GEM report list

Visual C#:

```
ReportsClass Reports
```

45.1.3. Variables Property

GEM variable list

Visual C#:

```
variablesClass Variables
```

46. Dorian.Gem.EventModelClass.Event Class

GEM event

Properties:

Properties	Description
Count	Number of reports
Definition	Pre-defined event ID
Description	Event description
Enable	Enable or disable event
ID	Event ID
Reports	Report IDs
this[int]	Report IDs

Methods:

Methods	Description
AddReport	Add one report ID
ClearReports	Clear all report IDs

46.1. Properties

46.1.1. Count Property

Number of reports

Visual C#:

```
int Count
```

46.1.2. Definition Property

Pre-defined event ID

Visual C#:

```
PredefinedCEID Definition
```

46.1.3. Description Property

Event description

Visual C#:

```
string Description
```

46.1.4. Enable Property

Enable or disable event

Visual C#:

```
bool Enable
```

46.1.5. ID Property

Event ID

Visual C#:

```
int ID
```

46.1.6. Reports Property

Report IDs

Visual C#:

```
Int32[] Reports
```

46.1.7. this[int] Property

Report IDs

Visual C#:

```
this[int]
```


46.2. Methods

46.2.1. AddReport Method

Add one report ID

Visual C#:

```
public void AddReport()
```

46.2.2. ClearReports Method

Clear all report IDs

Visual C#:

```
public bool ClearReports()
```

Return value:

Returns true if successful. Otherwise, false.

47. Dorian.Gem.EventModelClass.EventsClass Class

GEM event list

Properties:

Properties	Description
Count	Number of events
this[int]	Get event object

Methods:

Methods	Description
Add	Add new event
Remove	Remove event

47.1. Properties

47.1.1. Count Property

Number of events

Visual C#:

```
int Count
```

47.1.2. this[int] Property

Get event object

Visual C#:

```
this[int]
```


47.2. Methods

47.2.1. Add Method

Add new event

Visual C#:

```
public Event Add()
```

Return value:

47.2.2. Remove Method

Remove event

Visual C#:

```
public bool Remove(int index)
```

Arguments

Name	Description
index	index

Return value:

Returns true if successful. Otherwise, false.

48. Dorian.Gem.EventModelClass.Report Class

GEM report

Properties:

Properties	Description
Count	Number of variables
Description	Report description
ID	Report ID
this[int]	Variable IDs
Variables	Variable IDs

Methods:

Methods	Description
AddVariable	Add one VID
ClearVariables	Clear all VIDs

48.1. Properties

48.1.1. Count Property

Number of variables

Visual C#:

```
int Count
```

48.1.2. Description Property

Report description

Visual C#:

```
string Description
```

48.1.3. ID Property

Report ID

Visual C#:

```
int ID
```

48.1.4. this[int] Property

Variable IDs

Visual C#:

```
this[int]
```

48.1.5. Variables Property

Variable IDs

Visual C#:

```
Int32[] Variables
```

48.2. Methods

48.2.1. AddVariable Method

Add one VID

Visual C#:

```
public void AddVariable()
```

48.2.2. ClearVariables Method

Clear all VIDs

Visual C#:

```
public bool ClearVariables()
```

Return value:

Returns true if successful. Otherwise, false.

49. Dorian.Gem.EventModelClass.ReportsClass Class

GEM report list

Properties:

Properties	Description
Count	Number of reports
this[int]	Get report object

Methods:

Methods	Description
Add	Add new report
Remove	Remove report

49.1. Properties

49.1.1. Count Property

Number of reports

Visual C#:

```
int Count
```

49.1.2. this[int] Property

Get report object

Visual C#:

```
this[int]
```

49.2. Methods

49.2.1. Add Method

Add new report

Visual C#:

```
public Report Add()
```

Return value:

49.2.2. Remove Method

Remove report

Visual C#:

```
public bool Remove(int index)
```

Arguments

Name	Description
index	index

Return value:

Returns true if successful. Otherwise, false.

50. Dorian.Gem.EventModelClass.Variable Class

GEM variable

Properties:

Properties	Description
DefaultValue	Default value of the variable
Definition	Pre-defined variable ID
Description	Variable description
ID	Variable ID
Max	Maximum value of the variable
Min	Minimum value of the variable
SecsType	Variable SECS-II type
Sml	Variable SML
Type	Variable type
Unit	Unit of the variable

50.1. Properties

50.1.1. DefaultValue Property

Default value of the variable

Visual C#:

```
string DefaultValue
```

50.1.2. Definition Property

Pre-defined variable ID

Visual C#:

```
PredefinedVID Definition
```

50.1.3. Description Property

Variable description

Visual C#:

```
string Description
```

50.1.4. ID Property

Variable ID

Visual C#:

```
int ID
```

50.1.5. Max Property

Maximum value of the variable

Visual C#:

```
string Max
```

50.1.6. Min Property

Minimum value of the variable

Visual C#:

```
string Min
```

50.1.7. SecsType Property

Variable SECS-II type

Visual C#:

```
SecsType SecsType
```

50.1.8. **Sml Property**

Variable SML

Visual C#:

```
string Sml
```

50.1.9. **Type Property**

Variable type

Visual C#:

```
variableType Type
```

50.1.10. **Unit Property**

Unit of the variable

Visual C#:

```
string Unit
```

51. Dorian.Gem.EventModelClass.VariablesClass Class

GEM variable list

Properties:

Properties	Description
Count	Number of variables
this[int]	Get variable object

Methods:

Methods	Description
Add	Add new variable
Remove	Remove variable

51.1. Properties

51.1.1. Count Property

Number of variables

Visual C#:

```
int Count
```

51.1.2. this[int] Property

Get variable object

Visual C#:

```
this[int]
```

51.2. Methods

51.2.1. Add Method

Add new variable

Visual C#:

```
public variable Add()
```

Return value:

51.2.2. Remove Method

Remove variable

Visual C#:

```
public bool Remove(int index)
```

Arguments

Name	Description
index	index

Return value:

Returns true if successful. Otherwise, false.

52. Dorian.Gem.HsmsClass Class

SEMI E37 (HSMS) compliant communication layer.

Properties:

Properties	Description
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
Connect	Connect to the server or start the server.
HandleCtrlMsg	IP v4 or IP v6.
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Port	TCP port number.
Selected	HSMS connection has been "selected" by select request/response transaction.
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SystemBytes	System bytes
T3	T3 timer.
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.

Methods:

Methods	Description
Send	Send the message to the server

52.1. Properties

52.1.1. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

52.1.2. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

52.1.3. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

52.1.4. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

52.1.5. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the HSMS connection status. If Connect property is set to true where Server property is false (client), Dorian.Hsms component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Hsms component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened.

If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Hsms component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

52.1.6. HandleCtrlMsg Property

IP v4 or IP v6.

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Set true if IP v4. Set false if IP v6.

52.1.7. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

52.1.8. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

52.1.9. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

52.1.10. Selected Property

HSMS connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of HSMS communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Hsms component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Hsms received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

52.1.11. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Hsms control will run as server. If Server property is false, Dorian.Hsms component will run as client.

52.1.12. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

52.1.13. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

52.1.14. T3 Property

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS in seconds. The default value is 45 seconds.

52.1.15. T5 Property

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

52.1.16. T6 Property

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

52.1.17. T7 Property

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

52.1.18. T8 Property

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

52.2. Methods

52.2.1. Send Method

Send the message to the server

Visual C#:

```
public bool send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

53. Dorian.Gem.ProblemEventArgs Class

Problem event arguments

Properties:

Properties	Description
code	Error code
ip	IP address
message	Additional info
port	TCP port number
sender	Sender of this event

53.1. Properties

53.1.1. code Property

Error code

Visual C#:

```
int code
```

53.1.2. ip Property

IP address

Visual C#:

```
string ip
```

53.1.3. message Property

Additional info

Visual C#:

```
Byte[] message
```

53.1.4. port Property

TCP port number

Visual C#:

```
string port
```

53.1.5. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

54. Dorian.Gem.ReceivedEventArgs Class

Received event arguments

Properties:

Properties	Description
ip	IP address
message	SECS-II message
port	TCP port number
sender	Sender of this event

54.1. Properties

54.1.1. ip Property

IP address

Visual C#:

```
string ip
```

54.1.2. message Property

SECS-II message

Visual C#:

```
Byte[] message
```

54.1.3. port Property

TCP port number

Visual C#:

```
string port
```

54.1.4. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

55. Dorian.Gem.SecsIIClass Class

SECS-II

Properties:

Properties	Description
this[int]	Get SECS-II object

55.1. Properties

55.1.1. this[int] Property

Get SECS-II object

Visual C#:

```
this[int]
```

56. Dorian.Gem.SentEventArgs Class

Sent event arguments

Properties:

Properties	Description
ip	IP address
message	SECS-II message
port	TCP port number
sender	Sender of this event
success	Success or failure

56.1. Properties

56.1.1. ip Property

IP address

Visual C#:

```
string ip
```

56.1.2. message Property

SECS-II message

Visual C#:

```
Byte[] message
```

56.1.3. port Property

TCP port number

Visual C#:

```
string port
```

56.1.4. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

56.1.5. success Property

Success or failure

Visual C#:

```
bool success
```

57. Dorian.GemConfigDlg Class

GEM configuration dialog box

Properties:

Properties	Description
components	Required designer variable.
gem	GEM object

Methods:

Methods	Description
~Dorian.GemConfigDlg	Clean up any resources being used.
Dorian.GemConfigDlg	Constructor
InitializeComponent	Required method for Designer support - do not modify the contents of this method with the code editor.

57.1. Properties

57.1.1. components Property

Required designer variable.

Visual C#:

components

57.1.2. gem Property

GEM object

Visual C#:

Gem **gem**

57.2. Methods

57.2.1. ~Dorian.GemConfigDlg Method

Clean up any resources being used.

Visual C#:

```
~Dorian.GemConfigDlg(bool disposing)
```

Arguments

Name	Description
disposing	true if managed resources should be disposed; otherwise, false.

Return value:

57.2.2. Dorian.GemConfigDlg Method

Constructor

Visual C#:

```
Dorian.GemConfigDlg()
```

Return value:

57.2.3. InitializeComponent Method

Required method for Designer support - do not modify the contents of this method with the code editor.

Visual C#:

```
InitializeComponent()
```

Return value:

58. Dorian.Hsms Class

Dorian.Hsms component is an assistant product to develop SEMI E37 (HSMS) compliant communication application software. Dorian.Hsms component can be used for either equipment side development or host side development. Usually Dorian.Hsms component will be used with Dorian.SecsII component.

Properties:

Properties	Description
AutoReconnect	Automatically reconnect after T5 timeout
AutoSelectReq	Automatically send select request
AutoSessionID	Automatically adjust session ID
AutoSystemBytes	Automatically adjust system bytes
Clients	List of connected clients (passive entity only)
Connect	Connect to the server or start the server.
FileName	JSON settings file name
HandleCtrlMsg	Handles HSMS control message automatically
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
IP	IP address of remote computer.
IPv4	IP v4 or IP v6.
Json	JSON settings
MDLN	Equipment model name
Port	TCP port number.
Selected	HSMS connection has been "selected" by select request/response transaction.
Server	Server (passive entity) or client (active entity).
SessionID	Session ID (HSMS)
SOFTREV	Equipment software revision
SystemBytes	System bytes
T3	T3 timer.
T5	T5 timer.
T6	T6 timer.
T7	T7 timer.
T8	T8 timer.
Transactions	List of open transactions

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message to the server
TimerProc	This function has to be called periodically in order for this class to work.

Events:

Events	Description
ConnectedEvent	Connected event (passive entity only)
DisconnectedEvent	Disconnected event
ProblemEvent	Problem event
ReceivedEvent	Received event
SentEvent	Sent event

58.1. Properties

58.1.1. AutoReconnect Property

Automatically reconnect after T5 timeout

Visual C#:

```
bool AutoReconnect
```

58.1.2. AutoSelectReq Property

Automatically send select request

Visual C#:

```
bool AutoSelectReq
```

Remarks:

Automatically send select request when connected.

58.1.3. AutoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust session ID when sending.

58.1.4. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

58.1.5. Clients Property

List of connected clients (passive entity only)

Visual C#:

```
ClientsClass Clients
```

58.1.6. Connect Property

Connect to the server or start the server.

Visual C#:

```
bool Connect
```


Remarks:

Gets or sets the HSMS connection status. If Connect property is set to true where Server property is false (client), Dorian.Hsms component will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), Dorian.Hsms component will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be true, when server port was opened. If user sets false to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. Dorian.Hsms component will take care of the disconnection automatically. When Server property was set to true and Connect property is set to false, all existing connections with clients will be disconnected.

If Connect property on server side was set to false, connection would be disconnected. Therefore, Connect property on client side would also be set to false.

58.1.7. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

58.1.8. HandleCtrlMsg Property

Handles HSMS control message automatically

Visual C#:

```
bool HandleCtrlMsg
```

Remarks:

Handles HSMS control message automatically

58.1.9. Heartbeat Property

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

58.1.10. IP Property

IP address of remote computer.

Visual C#:

```
string IP
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

58.1.11. IPv4 Property

IP v4 or IP v6.

Visual C#:

```
bool IPv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

58.1.12. Json Property

JSON settings

Visual C#:

```
string Json
```

58.1.13. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

58.1.14. Port Property

TCP port number.

Visual C#:

```
string Port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

58.1.15. Selected Property

HSMS connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets or sets the selection status of HSMS communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Hsms component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Hsms received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

58.1.16. Server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool Server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Hsms control will run as server. If Server property is false, Dorian.Hsms component will run as client.

58.1.17. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

58.1.18. SOFTREV Property

Equipment software revision

Visual C#:

```
string SOFTREV
```

58.1.19. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

58.1.20. T3 Property

T3 timer.

Visual C#:

```
double T3
```

Remarks:

Gets or sets the T3 time out for HSMS in seconds. The default value is 45 seconds.

58.1.21. T5 Property

T5 timer.

Visual C#:

```
double T5
```

Remarks:

Gets or sets the T5 time out in seconds. The default value is 10 seconds.

58.1.22. T6 Property

T6 timer.

Visual C#:

```
double T6
```

Remarks:

Gets or sets the T6 time out in seconds. The default value is 5 seconds.

58.1.23. T7 Property

T7 timer.

Visual C#:

```
double T7
```

Remarks:

Gets or sets the T7 time out in seconds. The default value is 10 seconds.

58.1.24. T8 Property

T8 timer.

Visual C#:

```
double T8
```

Remarks:

Gets or sets the T8 time out in seconds. The default value is 5 seconds.

58.1.25. Transactions Property

List of open transactions

Visual C#:

```
TransactionsClass Transactions
```

58.2. Methods**58.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

58.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

58.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

58.2.4. Send Method

Send the message to the server

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Return true if transmission was successful. Otherwise return false.

58.2.5. **TimerProc Method**

This function has to be called periodically in order for this class to work.

Visual C#:

```
public void TimerProc()
```

58.3. Events**58.3.1. ConnectedEventEvent**

Connected event (passive entity only)

Visual C#:

```
public void ConnectedEvent(object sender, ConnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

58.3.2. DisconnectedEventEvent

Disconnected event

Visual C#:

```
public void DisconnectedEvent(object sender, DisconnectedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

58.3.3. ProblemEventEvent

Problem event

Visual C#:

```
public void ProblemEvent(object sender, ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

58.3.4. ReceivedEventEvent

Received event

Visual C#:

```
public void ReceivedEvent(object sender, ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

58.3.5. SentEventEvent

Sent event

Visual C#:

```
public void SentEvent(object sender, EventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

59. Dorian.Hsms.ClientsClass Class

List of connected clients (passive entity only)

Properties:

Properties	Description
Count	Number of connected clients
this[int]	Syntax sugar to access the client

Methods:

Methods	Description
find	Find the client

59.1. Properties

59.1.1. Count Property

Number of connected clients

Visual C#:

```
int Count
```

59.1.2. this[int] Property

Syntax sugar to access the client

Visual C#:

```
this[int]
```

59.2. Methods

59.2.1. find Method

Find the client

Visual C#:

```
public int find(string ip, string port)
```

Arguments

Name	Description
ip	IP address
port	TCP port number

Return value:

60. Dorian.Hsms.ClientsClass.Client Class

Connected client

Properties:

Properties	Description
IP	IP address
Port	TCP port number
Selected	HSMS connection has been "selected" by select request/response transaction.

Methods:

Methods	Description
Disconnect	Disconnect the client
Send	Send a message to the client

60.1. Properties

60.1.1. IP Property

IP address

Visual C#:

```
string IP
```

60.1.2. Port Property

TCP port number

Visual C#:

```
string Port
```

60.1.3. Selected Property

HSMS connection has been "selected" by select request/response transaction.

Visual C#:

```
bool Selected
```

Remarks:

Gets the selection status of HSMS communication.

If false, HSMS connection was not selected. If true, HSMS connection was selected.

If Dorian.Hsms component received Select Request message, the connection would switch to "selected" status. If user doesn't want to change to "selected" by some reason, set Selected property to false. When Dorian.Hsms received Select Response message, judge by "reason code" whether "selected" is appropriate. This property will affect T7 timeout.

60.2. Methods

60.2.1. Disconnect Method

Disconnect the client

Visual C#:

```
public bool Disconnect()
```

Remarks:

Disconnects specified TCP/IP connection.

Return value:

Return true if disconnection was successfully done. Otherwise return false. If false was returned, IP address or port number might be incorrect. Or connection had already disconnected.

60.2.2. Send Method

Send a message to the client

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Send specified message.

Arguments

Name	Description
message	Message

Return value:

Return true if transmission was successful. Otherwise return false.

61. Dorian.Hsms.ConnectedEventArgs Class

Connected event arguments

Properties:

Properties	Description
ip	IP address
port	TCP port number
sender	Sender of this event

61.1. Properties

61.1.1. ip Property

IP address

Visual C#:

```
string ip
```

61.1.2. port Property

TCP port number

Visual C#:

```
string port
```

61.1.3. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```


62. Dorian.Hsms.Data Class

JSON data

Properties:

Properties	Description
autoSessionID	Automatically adjust session ID
autoSystemBytes	Automatically adjust system bytes
handleCtrlMessage	Handles HSMS control message automatically
sendSelectReq	Automatically send select request
sessionID	Session ID (HSMS)
socket	HSMS object
t3	T3 timer.
t5	T5 timer.
t6	T6 timer.
t7	T7 timer.
t8	T8 timer.

62.1. Properties

62.1.1. autoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool autoSessionID
```

Remarks:

Automatically adjust session ID when sending.

62.1.2. autoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool autoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

62.1.3. handleCtrlMessage Property

Handles HSMS control message automatically

Visual C#:

```
bool handleCtrlMessage
```

Remarks:

Handles HSMS control message automatically

62.1.4. sendSelectReq Property

Automatically send select request

Visual C#:

```
bool sendSelectReq
```

Remarks:

Automatically send select request when connected.

62.1.5. sessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 sessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

62.1.6. socket Property

HSMS object

Visual C#:

```
Socket socket
```

62.1.7. t3 Property

T3 timer.

Visual C#:

```
double t3
```

Remarks:

Gets or sets the T3 time out for HSMS in seconds. The default value is 45 seconds.

62.1.8. t5 Property

T5 timer.

Visual C#:

```
double t5
```

Remarks:

Gets or sets the T5 time out for HSMS in seconds. The default value is 10 seconds.

62.1.9. t6 Property

T6 timer.

Visual C#:

```
double t6
```

Remarks:

Gets or sets the T6 time out for HSMS in seconds. The default value is 5 seconds.

62.1.10. t7 Property

T7 timer.

Visual C#:

```
double t7
```

Remarks:

Gets or sets the T7 time out for HSMS in seconds. The default value is 10 seconds.

62.1.11. t8 Property

T8 timer.

Visual C#:

double **t8**

Remarks:

Gets or sets the T8 time out for HSMS in seconds. The default value is 5 seconds.

63. Dorian.Hsms.Data.Socket Class

Socket

Properties:

Properties	Description
ip	IP address of remote computer.
ipv4	IP v4 or IP v6.
port	TCP port number.
server	Server (passive entity) or client (active entity).

63.1. Properties

63.1.1. ip Property

IP address of remote computer.

Visual C#:

```
string ip
```

Remarks:

Gets or sets the IP address of passive entity computer for HSMS connection. IP property should be "0.0.0.0" if the Server property is set to true, because server listens incoming connection. If you want to use specific network adapter's IP address, use the IP address on that network adapter.

When connecting local computer(same computer), use "127.0.0.1" or "localhost".

It is possible to use computer name instead of IP address.

63.1.2. ipv4 Property

IP v4 or IP v6.

Visual C#:

```
bool ipv4
```

Remarks:

Set true if IP v4. Set false if IP v6.

63.1.3. port Property

TCP port number.

Visual C#:

```
string port
```

Remarks:

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 5000 in general. For example http server uses port number 80.

63.1.4. server Property

Server (passive entity) or client (active entity).

Visual C#:

```
bool server
```

Remarks:

Gets or sets the entity type. If Server property is true, Dorian.Hsms control will run as server. If Server property is false, Dorian.Hsms component will run as client.

64. Dorian.Hsms.DisconnectedEventArgs Class

Disconnected event arguments

Properties:

Properties	Description
ip	IP address
port	TCP port number
sender	Sender of this event

64.1. Properties

64.1.1. ip Property

IP address

Visual C#:

```
string ip
```

64.1.2. port Property

TCP port number

Visual C#:

```
string port
```

64.1.3. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```


65. Dorian.Hsms.ProblemEventArgs Class

Problem event arguments

Properties:

Properties	Description
code	Error code
ip	IP address
message	Additional info
port	TCP port number
sender	Sender of this event

65.1. Properties

65.1.1. code Property

Error code

Visual C#:

```
int code
```

65.1.2. ip Property

IP address

Visual C#:

```
string ip
```

65.1.3. message Property

Additional info

Visual C#:

```
Byte[] message
```

65.1.4. port Property

TCP port number

Visual C#:

```
string port
```

65.1.5. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

66. Dorian.Hsms.ReceivedEventArgs Class

Received event arguments

Properties:

Properties	Description
ip	IP address
message	SECS-II message
port	TCP port number
sender	Sender of this event

66.1. Properties

66.1.1. ip Property

IP address

Visual C#:

```
string ip
```

66.1.2. message Property

SECS-II message

Visual C#:

```
Byte[] message
```

66.1.3. port Property

TCP port number

Visual C#:

```
string port
```

66.1.4. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

67. Dorian.Hsms.SentEventArgs Class

Sent event arguments

Properties:

Properties	Description
ip	IP address
message	SECS-II message
port	TCP port number
sender	Sender of this event
success	Success or failure

67.1. Properties

67.1.1. ip Property

IP address

Visual C#:

```
string ip
```

67.1.2. message Property

SECS-II message

Visual C#:

```
Byte[] message
```

67.1.3. port Property

TCP port number

Visual C#:

```
string port
```

67.1.4. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

67.1.5. success Property

Success or failure

Visual C#:

```
bool success
```

68. Dorian.Hsms.TransactionsClass Class

List of open transactions

Properties:

Properties	Description
Count	Number of open transactions
this[int]	Syntax sugar to access the transaction

68.1. Properties

68.1.1. Count Property

Number of open transactions

Visual C#:

```
int Count
```

68.1.2. this[int] Property

Syntax sugar to access the transaction

Visual C#:

```
this[int]
```

69. Dorian.Hsms.TransactionsClass.Transaction Class

Open transaction

Properties:

Properties	Description
Ctrl	HSMS control message
Msg	Message

69.1. Properties

69.1.1. Ctrl Property

HSMS control message

Visual C#:

```
bool Ctrl
```

Remarks:

If false, HSMS data message. If true, HSMS control message.

69.1.2. Msg Property

Message

Visual C#:

```
Byte[] Msg
```

70. Dorian.HsmsConfigDlg Class

HSMS configuration dialog box

Properties:

Properties	Description
components	Required designer variable.
hsms	HSMS object

Methods:

Methods	Description
~Dorian.HsmsConfigDlg	Clean up any resources being used.
Dorian.HsmsConfigDlg	Constructor
HsmsConfigDlg_Load	On load event
InitializeComponent	Required method for Designer support - do not modify the contents of this method with the code editor.

70.1. Properties

70.1.1. components Property

Required designer variable.

Visual C#:

`components`

70.1.2. hsms Property

HSMS object

Visual C#:

`Hsms hsms`

70.2. Methods**70.2.1. ~Dorian.HsmsConfigDlg Method**

Clean up any resources being used.

Visual C#:

```
~Dorian.HsmsConfigDlg(bool disposing)
```

Arguments

Name	Description
disposing	true if managed resources should be disposed; otherwise, false.

Return value:

70.2.2. Dorian.HsmsConfigDlg Method

Constructor

Visual C#:

```
Dorian.HsmsConfigDlg()
```

Return value:

70.2.3. HsmsConfigDlg_Load Method

On load event

Visual C#:

```
HsmsConfigDlg_Load(System.Object sender, System.EventArgs e)
```

Arguments

Name	Description
sender	Sender
e	Argument

Return value:

70.2.4. InitializeComponent Method

Required method for Designer support - do not modify the contents of this method with the code editor.

Visual C#:

```
InitializeComponent()
```

Return value:

71. Dorian.Klarf12 Class

Dorian.Klarf12 component is an assistant product to develop application software using KLARF version 1.2.

Properties:

Properties	Description
Count	Number of records
this[int]	Syntax sugar to access the record
this[string]	Syntax sugar to access the record

Methods:

Methods	Description
Dorian.Klarf12	Constructor
Find	Search record
Parse	Parse KLARF file

71.1. Properties

71.1.1. Count Property

Number of records

Visual C#:

```
int Count
```

Remarks:

Gets the number of records in the KLARF file.

71.1.2. this[int] Property

Syntax sugar to access the record

Visual C#:

```
this[int]
```

71.1.3. this[string] Property

Syntax sugar to access the record

Visual C#:

```
this[string]
```


71.2. Methods**71.2.1. Dorian.Klarf12 Method**

Constructor

Visual C#:

```
Dorian.Klarf12()
```

Return value:

71.2.2. Find Method

Search record

Visual C#:

```
public int Find(string record, int from, bool forward)
```

Arguments

Name	Description
record	Record name
from	Index where searching starts
forward	true = forward search, false = backward search

Return value:

Index of the record. Returns -1, if not found.

71.2.3. Parse Method

Parse KLARF file

Visual C#:

```
public bool Parse(string klarfText)
```

Remarks:

Parse the KLARF file content.

Arguments

Name	Description
klarfText	KLARF file content

Return value:

Returns true if successful. Otherwise, false.

72. Dorian.Klarf12.AlignmentImages Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.AlignmentImages	Constructor

72.1. Properties

72.1.1. Count Property

Item count

Visual C#:

```
int Count
```

72.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

72.2. Methods

72.2.1. Dorian.Klarf12.AlignmentImages Method

Constructor

Visual C#:

```
Dorian.Klarf12.AlignmentImages(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

73. Dorian.Klarf12.AlignmentImages.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
ID	ID
NumOfImage	Number of image
X	X
Y	Y

73.1. Properties

73.1.1. ID Property

ID

Visual C#:

```
int ID
```

73.1.2. NumOfImage Property

Number of image

Visual C#:

```
int NumOfImage
```

73.1.3. X Property

X

Visual C#:

```
double X
```

73.1.4. Y Property

Y

Visual C#:

```
double Y
```

74. Dorian.Klarf12.AlignmentImageTransforms Class

Wrapper class to access specific record.

Properties:

Properties	Description
A11	A11 (upper left of 2x2 matrix)
A12	A12 (upper right of 2x2 matrix)
A21	A21 (lower left of 2x2 matrix)
A22	A21 (lower right of 2x2 matrix)
Count	Item count
MarkID	Items

Methods:

Methods	Description
Dorian.Klarf12.AlignmentImageTransforms	Constructor

74.1. Properties

74.1.1. A11 Property

A11 (upper left of 2x2 matrix)

Visual C#:

```
double A11
```

74.1.2. A12 Property

A12 (upper right of 2x2 matrix)

Visual C#:

```
double A12
```

74.1.3. A21 Property

A21 (lower left of 2x2 matrix)

Visual C#:

```
double A21
```

74.1.4. A22 Property

A21 (lower right of 2x2 matrix)

Visual C#:

```
double A22
```

74.1.5. Count Property

Item count

Visual C#:

```
int Count
```

74.1.6. MarkID Property

Items

Visual C#:

```
Int32[] MarkID
```


74.2. Methods

74.2.1. Dorian.Klarf12.AlignmentImageTransforms Method

Constructor

Visual C#:

```
Dorian.Klarf12.AlignmentImageTransforms(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
<code>record</code>	Record

Return value:

75. Dorian.Klarf12.AlignmentPoints Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.AlignmentPoints	Constructor

75.1. Properties

75.1.1. Count Property

Item count

Visual C#:

```
int Count
```

75.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

75.2. Methods**75.2.1. Dorian.Klarf12.AlignmentPoints Method**

Constructor

Visual C#:

```
Dorian.Klarf12.AlignmentPoints(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

76. Dorian.Klarf12.AlignmentPoints.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
ID	ID
X	X
Y	Y

76.1. Properties

76.1.1. ID Property

ID

Visual C#:

```
int ID
```

76.1.2. X Property

X

Visual C#:

```
double X
```

76.1.3. Y Property

Y

Visual C#:

```
double Y
```

77. Dorian.Klarf12.AreaPerTest Class

Wrapper class to access specific record.

Properties:

Properties	Description
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.AreaPerTest	Constructor

77.1. Properties

77.1.1. Value Property

Value

Visual C#:

```
double value
```


77.2. Methods**77.2.1. Dorian.Klarf12.AreaPerTest Method**

Constructor

Visual C#:

```
Dorian.Klarf12.AreaPerTest(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

78. Dorian.Klarf12.ClassLookup Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.ClassLookup	Constructor

78.1. Properties

78.1.1. Count Property

Item count

Visual C#:

```
int Count
```

78.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

78.2. Methods

78.2.1. Dorian.Klarf12.ClassLookup Method

Constructor

Visual C#:

```
Dorian.Klarf12.ClassLookup(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

79. Dorian.Klarf12.ClassLookup.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
ID	ID
Name	Class name

79.1. Properties

79.1.1. ID Property

ID

Visual C#:

```
int ID
```

79.1.2. Name Property

Class name

Visual C#:

```
string Name
```

80. Dorian.Klarf12.ClusterClassificationList Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.ClusterClassificationList	Constructor

80.1. Properties

80.1.1. Count Property

Item count

Visual C#:

```
int Count
```

80.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```


80.2. Methods

80.2.1. Dorian.Klarf12.ClusterClassificationList Method

Constructor

Visual C#:

```
Dorian.Klarf12.ClusterClassificationList(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

81. Dorian.Klarf12.ClusterClassificationList.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
Classification	Classification number
ID	ID

81.1. Properties

81.1.1. Classification Property

Classification number

Visual C#:

```
int Classification
```

81.1.2. ID Property

ID

Visual C#:

```
int ID
```

82. Dorian.Klarf12.CoordinatesMirrored Class

Wrapper class to access specific record.

Properties:

Properties	Description
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.CoordinatesMirrored	Constructor

82.1. Properties

82.1.1. Value Property

Value

Visual C#:

```
bool value
```

82.2. Methods**82.2.1. Dorian.Klarf12.CoordinatesMirrored Method**

Constructor

Visual C#:

```
Dorian.Klarf12.CoordinatesMirrored(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

83. Dorian.Klarf12.DatabaseAlignmentMarks Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.DatabaseAlignmentMarks	Constructor

83.1. Properties

83.1.1. Count Property

Item count

Visual C#:

```
int Count
```

83.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```


83.2. Methods**83.2.1. Dorian.Klarf12.DatabaseAlignmentMarks Method**

Constructor

Visual C#:

```
Dorian.Klarf12.DatabaseAlignmentMarks(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
<code>record</code>	Record

Return value:

84. Dorian.Klarf12.DatabaseAlignmentMarks.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
ID	ID
OriginX	Origin X
OriginY	Origin Y
PointX	Point X
PointY	Point Y

84.1. Properties

84.1.1. ID Property

ID

Visual C#:

```
int ID
```

84.1.2. OriginX Property

Origin X

Visual C#:

```
double OriginX
```

84.1.3. OriginY Property

Origin Y

Visual C#:

```
double OriginY
```

84.1.4. PointX Property

Point X

Visual C#:

```
double PointX
```

84.1.5. PointY Property

Point Y

Visual C#:

```
double PointY
```

85. Dorian.Klarf12.DefectClusterSetup Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.DefectClusterSetup	Constructor

85.1. Properties

85.1.1. Count Property

Item count

Visual C#:

```
int Count
```

85.1.2. Value Property

Value

Visual C#:

```
string[] value
```

85.2. Methods

85.2.1. Dorian.Klarf12.DefectClusterSetup Method

Constructor

Visual C#:

```
Dorian.Klarf12.DefectClusterSetup(Dorian.Klarf12.Record record, Dorian.Klarf12.Record menu)
```

Arguments

Name	Description
record	Record
menu	TestParametersSpec record

Return value:

86. Dorian.Klarf12.DefectClusterSpec Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.DefectClusterSpec	Constructor

86.1. Properties

86.1.1. Count Property

Item count

Visual C#:

```
int Count
```

86.1.2. Value Property

Value

Visual C#:

```
string[] value
```


86.2. Methods**86.2.1. Dorian.Klarf12.DefectClusterSpec Method**

Constructor

Visual C#:

```
Dorian.Klarf12.DefectClusterSpec(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

87. Dorian.Klarf12.DefectList Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.DefectList	Constructor

87.1. Properties

87.1.1. Count Property

Item count

Visual C#:

```
int Count
```

87.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

87.2. Methods**87.2.1. Dorian.Klarf12.DefectList Method**

Constructor

Visual C#:

```
Dorian.Klarf12.DefectList(Dorian.Klarf12.Record record, Dorian.Klarf12.Record menu)
```

Arguments

Name	Description
record	Record
menu	TestParametersSpec record

Return value:

88. Dorian.Klarf12.DefectList.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
Count	Item count
this[string]	Syntax sugar to access the item
Value	Value

88.1. Properties

88.1.1. Count Property

Item count

Visual C#:

```
int Count
```

88.1.2. this[string] Property

Syntax sugar to access the item

Visual C#:

```
this[string]
```

88.1.3. Value Property

Value

Visual C#:

```
string[] value
```

89. Dorian.Klarf12.DefectRecordSpec Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[string]	Syntax sugar to access the item
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.DefectRecordSpec	Constructor

89.1. Properties

89.1.1. Count Property

Item count

Visual C#:

```
int Count
```

89.1.2. this[string] Property

Syntax sugar to access the item

Visual C#:

```
this[string]
```

89.1.3. Value Property

Value

Visual C#:

```
string[] value
```


89.2. Methods**89.2.1. Dorian.Klarf12.DefectRecordSpec Method**

Constructor

Visual C#:

```
Dorian.Klarf12.DefectRecordSpec(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

90. Dorian.Klarf12.DeviceID Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.DeviceID	Constructor

90.1. Properties

90.1.1. Text Property

Text

Visual C#:

```
string Text
```

90.2. Methods**90.2.1. Dorian.Klarf12.DeviceID Method**

Constructor

Visual C#:

```
Dorian.Klarf12.DeviceID(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

91. Dorian.Klarf12.DieOrigin Class

Wrapper class to access specific record.

Properties:

Properties	Description
X	X
Y	Y

Methods:

Methods	Description
Dorian.Klarf12.DieOrigin	Constructor

91.1. Properties

91.1.1. X Property

X

Visual C#:

double X

91.1.2. Y Property

Y

Visual C#:

double Y

91.2. Methods

91.2.1. Dorian.Klarf12.DieOrigin Method

Constructor

Visual C#:

```
Dorian.Klarf12.DieOrigin(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

92. Dorian.Klarf12.DiePitch Class

Wrapper class to access specific record.

Properties:

Properties	Description
X	X
Y	Y

Methods:

Methods	Description
Dorian.Klarf12.DiePitch	Constructor

92.1. Properties

92.1.1. X Property

X

Visual C#:

```
double X
```

92.1.2. Y Property

Y

Visual C#:

```
double Y
```

92.2. Methods

92.2.1. Dorian.Klarf12.DiePitch Method

Constructor

Visual C#:

```
Dorian.Klarf12.DiePitch(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

93. Dorian.Klarf12.EndOfFile Class

Wrapper class to access specific record.

Methods:

Methods	Description
Dorian.Klarf12.EndOfFile	Constructor

93.1. Methods**93.1.1. Dorian.Klarf12.EndOfFile Method**

Constructor

Visual C#:

```
Dorian.Klarf12.EndOfFile(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

94. Dorian.Klarf12.FileTimestamp Class

Wrapper class to access specific record.

Properties:

Properties	Description
DateTime	Date and time

Methods:

Methods	Description
Dorian.Klarf12.FileTimestamp	Constructor

94.1. Properties

94.1.1. DateTime Property

Date and time

Visual C#:

```
DateTime DateTime
```

94.2. Methods

94.2.1. Dorian.Klarf12.FileTimestamp Method

Constructor

Visual C#:

```
Dorian.Klarf12.FileTimestamp(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

95. Dorian.Klarf12.FileVersion Class

Wrapper class to access specific record.

Properties:

Properties	Description
Major	Major version number
Minor	Minor version number

Methods:

Methods	Description
Dorian.Klarf12.FileVersion	Constructor

95.1. Properties

95.1.1. Major Property

Major version number

Visual C#:

```
int Major
```

95.1.2. Minor Property

Minor version number

Visual C#:

```
int Minor
```

95.2. Methods

95.2.1. Dorian.Klarf12.FileVersion Method

Constructor

Visual C#:

```
Dorian.Klarf12.FileVersion(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

96. Dorian.Klarf12.InspectedArea Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.InspectedArea	Constructor

96.1. Properties

96.1.1. Count Property

Item count

Visual C#:

```
int Count
```

96.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

96.2. Methods**96.2.1. Dorian.Klarf12.InspectedArea Method**

Constructor

Visual C#:

```
Dorian.Klarf12.InspectedArea(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

97. Dorian.Klarf12.InspectedArea.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
OffsetX	Offset X
OffsetY	Offset Y
PitchX	Pitch X
PitchY	Pitch Y
RepeatX	Repeat X
RepeatY	Repeat Y
SizeX	Size X
SizeY	Size Y

97.1. Properties

97.1.1. OffsetX Property

Offset X

Visual C#:

```
double OffsetX
```

97.1.2. OffsetY Property

Offset Y

Visual C#:

```
double OffsetY
```

97.1.3. PitchX Property

Pitch X

Visual C#:

```
double PitchX
```

97.1.4. PitchY Property

Pitch Y

Visual C#:

```
double PitchY
```

97.1.5. RepeatX Property

Repeat X

Visual C#:

```
int RepeatX
```

97.1.6. RepeatY Property

Repeat Y

Visual C#:

```
int RepeatY
```

97.1.7. SizeX Property

Size X

Visual C#:

```
double SizeX
```

97.1.8. **SizeY Property**

Size Y

Visual C#:

double **SizeY**

98. Dorian.Klarf12.InspectedAreaOrigin Class

Wrapper class to access specific record.

Properties:

Properties	Description
X	X
Y	Y

Methods:

Methods	Description
Dorian.Klarf12.InspectedAreaOrigin	Constructor

98.1. Properties

98.1.1. X Property

X

Visual C#:

double X

98.1.2. Y Property

Y

Visual C#:

double Y

98.2. Methods**98.2.1. Dorian.Klarf12.InspectedAreaOrigin Method**

Constructor

Visual C#:

```
Dorian.Klarf12.InspectedAreaOrigin(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

99. Dorian.Klarf12.InspectionOrientation Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.InspectionOrientation	Constructor

99.1. Properties

99.1.1. Text Property

Text

Visual C#:

```
string Text
```

99.2. Methods**99.2.1. Dorian.Klarf12.InspectionOrientation Method**

Constructor

Visual C#:

```
Dorian.Klarf12.InspectionOrientation(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

100. Dorian.Klarf12.InspectionStationID Class

Wrapper class to access specific record.

Properties:

Properties	Description
ID	Equipment ID
Manufacturer	Manufacturer
Model	Model

Methods:

Methods	Description
Dorian.Klarf12.InspectionStationID	Constructor

100.1. Properties

100.1.1. ID Property

Equipment ID

Visual C#:

```
string ID
```

100.1.2. Manufacturer Property

Manufacturer

Visual C#:

```
string Manufacturer
```

100.1.3. Model Property

Model

Visual C#:

```
string Model
```


100.2. Methods**100.2.1. Dorian.Klarf12.InspectionStationID Method**

Constructor

Visual C#:

```
Dorian.Klarf12.InspectionStationID(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

101. Dorian.Klarf12.InspectionTest Class

Wrapper class to access specific record.

Properties:

Properties	Description
ID	ID

Methods:

Methods	Description
Dorian.Klarf12.InspectionTest	Constructor

101.1. Properties

101.1.1. ID Property

ID

Visual C#:

```
int ID
```

101.2. Methods**101.2.1. Dorian.Klarf12.InspectionTest Method**

Constructor

Visual C#:

```
Dorian.Klarf12.InspectionTest(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

102. Dorian.Klarf12.LotID Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.LotID	Constructor

102.1. Properties

102.1.1. Text Property

Text

Visual C#:

```
string Text
```

102.2. Methods**102.2.1. Dorian.Klarf12.LotID Method**

Constructor

Visual C#:

```
Dorian.Klarf12.LotID(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

103. Dorian.Klarf12.LotStatus Class

Wrapper class to access specific record.

Properties:

Properties	Description
Failed	Number of Failed
Passed	Number of passed
Total	Number of Total

Methods:

Methods	Description
Dorian.Klarf12.LotStatus	Constructor

103.1. Properties

103.1.1. Failed Property

Number of Failed

Visual C#:

```
int Failed
```

103.1.2. Passed Property

Number of passed

Visual C#:

```
int Passed
```

103.1.3. Total Property

Number of Total

Visual C#:

```
int Total
```

103.2. Methods

103.2.1. Dorian.Klarf12.LotStatus Method

Constructor

Visual C#:

```
Dorian.Klarf12.LotStatus(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

104. Dorian.Klarf12.OrientationInstructions Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.OrientationInstructions	Constructor

104.1. Properties

104.1.1. Text Property

Text

Visual C#:

```
string Text
```

104.2. Methods**104.2.1. Dorian.Klarf12.OrientationInstructions Method**

Constructor

Visual C#:

```
Dorian.Klarf12.OrientationInstructions(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

105. Dorian.Klarf12.OrientationMarkLocation Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.OrientationMark Location	Constructor

105.1. Properties

105.1.1. Text Property

Text

Visual C#:

```
string Text
```

105.2. Methods**105.2.1. Dorian.Klarf12.OrientationMarkLocation Method**

Constructor

Visual C#:

```
Dorian.Klarf12.OrientationMarkLocation(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

106. Dorian.Klarf12.Record Class

Record data

Properties:

Properties	Description
Count	Number of item
Name	Name
this[int]	Item

106.1. Properties

106.1.1. Count Property

Number of item

Visual C#:

```
int Count
```

106.1.2. Name Property

Name

Visual C#:

```
string Name
```

106.1.3. this[int] Property

Item

Visual C#:

```
this[int]
```

107. Dorian.Klarf12.RecordBase Class

Base class for the wrapper class to access specific record.

Properties:

Properties	Description
Available	Availability of this object

107.1. Properties

107.1.1. Available Property

Availability of this object

Visual C#:

```
bool Available
```

108. Dorian.Klarf12.RemovedDieList Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.RemovedDieList	Constructor

108.1. Properties

108.1.1. Count Property

Item count

Visual C#:

```
int Count
```

108.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

108.2. Methods

108.2.1. Dorian.Klarf12.RemovedDieList Method

Constructor

Visual C#:

```
Dorian.Klarf12.RemovedDieList(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

109. Dorian.Klarf12.RemovedDieList.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
X	X
Y	Y

109.1. Properties

109.1.1. X Property

X

Visual C#:

```
int X
```

109.1.2. Y Property

Y

Visual C#:

```
int Y
```

110. Dorian.Klarf12.ResultsID Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.ResultsID	Constructor

110.1. Properties

110.1.1. Text Property

Text

Visual C#:

```
string Text
```

110.2. Methods

110.2.1. Dorian.Klarf12.ResultsID Method

Constructor

Visual C#:

```
Dorian.Klarf12.ResultsID(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

111. Dorian.Klarf12.ResultTimestamp Class

Wrapper class to access specific record.

Properties:

Properties	Description
DateTime	Date and time

Methods:

Methods	Description
Dorian.Klarf12.ResultTimestamp	Constructor

111.1. Properties

111.1.1. DateTime Property

Date and time

Visual C#:

```
DateTime DateTime
```

111.2. Methods

111.2.1. Dorian.Klarf12.ResultTimestamp Method

Constructor

Visual C#:

```
Dorian.Klarf12.ResultTimestamp(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

112. Dorian.Klarf12.SampleCenterLocation Class

Wrapper class to access specific record.

Properties:

Properties	Description
X	X
Y	Y

Methods:

Methods	Description
Dorian.Klarf12.SampleCenterLocation	Constructor

112.1. Properties

112.1.1. X Property

X

Visual C#:

```
double X
```

112.1.2. Y Property

Y

Visual C#:

```
double Y
```

112.2. Methods

112.2.1. Dorian.Klarf12.SampleCenterLocation Method

Constructor

Visual C#:

```
Dorian.Klarf12.SampleCenterLocation(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

113. Dorian.Klarf12.SampleDieMap Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.SampleDieMap	Constructor

113.1. Properties

113.1.1. Count Property

Item count

Visual C#:

```
int Count
```

113.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

113.2. Methods

113.2.1. Dorian.Klarf12.SampleDieMap Method

Constructor

Visual C#:

```
Dorian.Klarf12.SampleDieMap(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

114. Dorian.Klarf12.SampleDieMap.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
X	X
Y	Y

114.1. Properties

114.1.1. X Property

X

Visual C#:

```
int X
```

114.1.2. Y Property

Y

Visual C#:

```
int Y
```

115. Dorian.Klarf12.SampleOrientationMarkType Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.SampleOrientationMarkType	Constructor

115.1. Properties

115.1.1. Text Property

Text

Visual C#:

```
string Text
```

115.2. Methods

115.2.1. Dorian.Klarf12.SampleOrientationMarkType Method

Constructor

Visual C#:

```
Dorian.Klarf12.SampleOrientationMarkType(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

116. Dorian.Klarf12.SampleSize Class

Wrapper class to access specific record.

Properties:

Properties	Description
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.SampleSize	Constructor

116.1. Properties

116.1.1. Value Property

Value

Visual C#:

```
Int32[] value
```

116.2. Methods

116.2.1. Dorian.Klarf12.SampleSize Method

Constructor

Visual C#:

```
Dorian.Klarf12.SampleSize(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

117. Dorian.Klarf12.SampleTestPlan Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.SampleTestPlan	Constructor

117.1. Properties

117.1.1. Count Property

Item count

Visual C#:

```
int Count
```

117.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

117.2. Methods**117.2.1. Dorian.Klarf12.SampleTestPlan Method**

Constructor

Visual C#:

```
Dorian.Klarf12.SampleTestPlan(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

118. Dorian.Klarf12.SampleTestPlan.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
X	X
Y	Y

118.1. Properties

118.1.1. X Property

X

Visual C#:

```
int X
```

118.1.2. Y Property

Y

Visual C#:

```
int Y
```

119. Dorian.Klarf12.SampleTestReferencePlan Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.SampleTestReferencePlan	Constructor

119.1. Properties

119.1.1. Count Property

Item count

Visual C#:

```
int Count
```

119.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

119.2. Methods**119.2.1. Dorian.Klarf12.SampleTestReferencePlan Method**

Constructor

Visual C#:

```
Dorian.Klarf12.SampleTestReferencePlan(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

120. Dorian.Klarf12.SampleTestReferencePlan.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
ComparedX	Compared X
ComparedY	Compared Y
SampledX	Sampled X
SampledY	Sampled Y

120.1. Properties

120.1.1. ComparedX Property

Compared X

Visual C#:

```
int ComparedX
```

120.1.2. ComparedY Property

Compared Y

Visual C#:

```
int ComparedY
```

120.1.3. SampledX Property

Sampled X

Visual C#:

```
int SampledX
```

120.1.4. SampledY Property

Sampled Y

Visual C#:

```
int SampledY
```

121. Dorian.Klarf12.SampleType Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.SampleType	Constructor

121.1. Properties

121.1.1. Text Property

Text

Visual C#:

```
string Text
```

121.2. Methods

121.2.1. Dorian.Klarf12.SampleType Method

Constructor

Visual C#:

```
Dorian.Klarf12.SampleType(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

122. Dorian.Klarf12.SetupID Class

Wrapper class to access specific record.

Properties:

Properties	Description
DateTime	Date and time
ID	ID

Methods:

Methods	Description
Dorian.Klarf12.SetupID	Constructor

122.1. Properties

122.1.1. DateTime Property

Date and time

Visual C#:

```
DateTime DateTime
```

122.1.2. ID Property

ID

Visual C#:

```
string ID
```

122.2. Methods

122.2.1. Dorian.Klarf12.SetupID Method

Constructor

Visual C#:

```
Dorian.Klarf12.SetupID(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

123. Dorian.Klarf12.Slot Class

Wrapper class to access specific record.

Properties:

Properties	Description
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.Slot	Constructor

123.1. Properties

123.1.1. Value Property

Value

Visual C#:

```
int Value
```

123.2. Methods

123.2.1. Dorian.Klarf12.Slot Method

Constructor

Visual C#:

```
Dorian.Klarf12.Slot(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

124. Dorian.Klarf12.StepID Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.StepID	Constructor

124.1. Properties

124.1.1. Text Property

Text

Visual C#:

```
string Text
```

124.2. Methods

124.2.1. Dorian.Klarf12.StepID Method

Constructor

Visual C#:

```
Dorian.Klarf12.StepID(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

125. Dorian.Klarf12.SummaryList Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
this[int]	Syntax sugar to access the item

Methods:

Methods	Description
Dorian.Klarf12.SummaryList	Constructor

125.1. Properties

125.1.1. Count Property

Item count

Visual C#:

```
int Count
```

125.1.2. this[int] Property

Syntax sugar to access the item

Visual C#:

```
this[int]
```

125.2. Methods

125.2.1. Dorian.Klarf12.SummaryList Method

Constructor

Visual C#:

```
Dorian.Klarf12.SummaryList(Dorian.Klarf12.Record record, Dorian.Klarf12.Record menu)
```

Arguments

Name	Description
<code>record</code>	Record
<code>menu</code>	TestParametersSpec record

Return value:

126. Dorian.Klarf12.SummaryList.DataItem Class

Wrapper class to access specific item.

Properties:

Properties	Description
Count	Item count
Value	Value

126.1. Properties

126.1.1. Count Property

Item count

Visual C#:

```
int Count
```

126.1.2. Value Property

Value

Visual C#:

```
string[] value
```


127. Dorian.Klarf12.SummarySpec Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.SummarySpec	Constructor

127.1. Properties

127.1.1. Count Property

Item count

Visual C#:

```
int Count
```

127.1.2. Value Property

Value

Visual C#:

```
string[] value
```

127.2. Methods

127.2.1. Dorian.Klarf12.SummarySpec Method

Constructor

Visual C#:

```
Dorian.Klarf12.SummarySpec(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

128. Dorian.Klarf12.TestParametersList Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.TestParametersList	Constructor

128.1. Properties

128.1.1. Count Property

Item count

Visual C#:

```
int Count
```

128.1.2. Value Property

Value

Visual C#:

```
string[] value
```

128.2. Methods

128.2.1. Dorian.Klarf12.TestParametersList Method

Constructor

Visual C#:

```
Dorian.Klarf12.TestParametersList(Dorian.Klarf12.Record record, Dorian.Klarf12.Record menu)
```

Arguments

Name	Description
record	Record
menu	TestParametersSpec record

Return value:

129. Dorian.Klarf12.TestParametersSpec Class

Wrapper class to access specific record.

Properties:

Properties	Description
Count	Item count
Value	Value

Methods:

Methods	Description
Dorian.Klarf12.TestParametersSpec	Constructor

129.1. Properties

129.1.1. Count Property

Item count

Visual C#:

```
int Count
```

129.1.2. Value Property

Value

Visual C#:

```
string[] Value
```


129.2. Methods

129.2.1. Dorian.Klarf12.TestParametersSpec Method

Constructor

Visual C#:

```
Dorian.Klarf12.TestParametersSpec(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

130. Dorian.Klarf12.TiffFileName Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.TiffFileName	Constructor

130.1. Properties

130.1.1. Text Property

Text

Visual C#:

```
string Text
```

130.2. Methods

130.2.1. Dorian.Klarf12.TiffFileName Method

Constructor

Visual C#:

```
Dorian.Klarf12.TiffFileName(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

131. Dorian.Klarf12.TiffSpec Class

Wrapper class to access specific record.

Properties:

Properties	Description
Alignment	TIFF class for alignment images
Defect	TIFF class for defect images
Version	TIFF Version

Methods:

Methods	Description
Dorian.Klarf12.TiffSpec	Constructor

131.1. Properties

131.1.1. Alignment Property

TIFF class for alignment images

Visual C#:

```
string Alignment
```

131.1.2. Defect Property

TIFF class for defect images

Visual C#:

```
string Defect
```

131.1.3. Version Property

TIFF Version

Visual C#:

```
string Version
```

131.2. Methods

131.2.1. Dorian.Klarf12.TiffSpec Method

Constructor

Visual C#:

```
Dorian.Klarf12.TiffSpec(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

132. Dorian.Klarf12.WaferID Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.WaferID	Constructor

132.1. Properties

132.1.1. Text Property

Text

Visual C#:

```
string Text
```

132.2. Methods

132.2.1. Dorian.Klarf12.WaferID Method

Constructor

Visual C#:

```
Dorian.Klarf12.WaferID(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

133. Dorian.Klarf12.WaferStatus Class

Wrapper class to access specific record.

Properties:

Properties	Description
Text	Text

Methods:

Methods	Description
Dorian.Klarf12.WaferStatus	Constructor

133.1. Properties

133.1.1. Text Property

Text

Visual C#:

```
string Text
```

133.2. Methods**133.2.1. Dorian.Klarf12.WaferStatus Method**

Constructor

Visual C#:

```
Dorian.Klarf12.WaferStatus(Dorian.Klarf12.Record record)
```

Arguments

Name	Description
record	Record

Return value:

134. Dorian.Klarf18 Class

Dorian.Klarf18 component is an assistant product to develop application software using KLARF version 1.8.

Properties:

Properties	Description
Json	JSON settings

Methods:

Methods	Description
Parse	Parse KLARF file

134.1. Properties

134.1.1. Json Property

JSON settings

Visual C#:

```
string Json
```

134.2. Methods

134.2.1. Parse Method

Parse KLARF file

Visual C#:

```
public bool Parse(string klarfText)
```

Remarks:

Parse the KLARF file content.

Arguments

Name	Description
<code>klarfText</code>	KLARF file content

Return value:

Returns true if successful. Otherwise, false.

135. Dorian.Log Class

Dorian.Log is a log file component.

Properties:

Properties	Description
Count	Number of backup file
Enable	Enable or disable logging
FileName	Log file name
FileSize	Log file size
Json	JSON settings
JsonFileName	JSON settings file name
TimeStamp	Enable or disable time stamp

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Write	Write text string

135.1. Properties

135.1.1. Count Property

Number of backup file

Visual C#:

```
int Count
```

Remarks:

Gets or sets the number of backup file.

135.1.2. Enable Property

Enable or disable logging

Visual C#:

```
bool Enable
```

Remarks:

Enables or disables logging feature.

135.1.3. FileName Property

Log file name

Visual C#:

```
string FileName
```

Remarks:

Gets or sets the log file name.

135.1.4. FileSize Property

Log file size

Visual C#:

```
int FileSize
```

Remarks:

Gets or sets the log file size.

135.1.5. Json Property

JSON settings

Visual C#:

```
string Json
```

135.1.6. JsonFileName Property

JSON settings file name

Visual C#:

```
string JsonFileName
```

135.1.7. **TimeStamp Property**

Enable or disable time stamp

Visual C#:

```
bool TimeStamp
```

Remarks:

Enables or disables time stamp in the log file.

135.2. Methods**135.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

135.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

135.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

135.2.4. Write Method

Write text string

Visual C#:

```
public void write(string value)
```

Remarks:

Writes text string to log file.

Arguments

Name	Description
value	Text string

136. Dorian.LogConfigDlg Class

Log configuration dialog box

Properties:

Properties	Description
components	Required designer variable.
log	Log object

Methods:

Methods	Description
~Dorian.LogConfigDlg	Clean up any resources being used.
Dorian.LogConfigDlg	Constructor
InitializeComponent	Required method for Designer support - do not modify the contents of this method with the code editor.
LogConfigDlg_Load	On load event

136.1. Properties

136.1.1. components Property

Required designer variable.

Visual C#:

components

136.1.2. log Property

Log object

Visual C#:

Log log

136.2. Methods**136.2.1. ~Dorian.LogConfigDlg Method**

Clean up any resources being used.

Visual C#:

```
~Dorian.LogConfigDlg(bool disposing)
```

Arguments

Name	Description
disposing	true if managed resources should be disposed; otherwise, false.

Return value:

136.2.2. Dorian.LogConfigDlg Method

Constructor

Visual C#:

```
Dorian.LogConfigDlg()
```

Return value:

136.2.3. InitializeComponent Method

Required method for Designer support - do not modify the contents of this method with the code editor.

Visual C#:

```
InitializeComponent()
```

Return value:

136.2.4. LogConfigDlg_Load Method

On load event

Visual C#:

```
LogConfigDlg_Load(System.Object sender, System.EventArgs e)
```

Arguments

Name	Description
sender	Sender
e	Argument

Return value:

137. Dorian.Properties.Resources Class

A strongly-typed resource class, for looking up localized strings, etc.

Properties:

Properties	Description
Add	Looks up a localized resource of type System.Drawing.Bitmap .
Banner	Looks up a localized resource of type System.Drawing.Bitmap .
Culture	Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.
Gear	Looks up a localized resource of type System.Drawing.Bitmap .
Play	Looks up a localized resource of type System.Drawing.Bitmap .
Remove	Looks up a localized resource of type System.Drawing.Bitmap .
ResourceManager	Returns the cached ResourceManager instance used by this class.
Stop	Looks up a localized resource of type System.Drawing.Bitmap .

137.1. Properties

137.1.1. Add Property

Looks up a localized resource of type System.Drawing.Bitmap.

Visual C#:

```
Add
```

137.1.2. Banner Property

Looks up a localized resource of type System.Drawing.Bitmap.

Visual C#:

```
Banner
```

137.1.3. Culture Property

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

Visual C#:

```
Culture
```

137.1.4. Gear Property

Looks up a localized resource of type System.Drawing.Bitmap.

Visual C#:

```
Gear
```

137.1.5. Play Property

Looks up a localized resource of type System.Drawing.Bitmap.

Visual C#:

```
Play
```

137.1.6. Remove Property

Looks up a localized resource of type System.Drawing.Bitmap.

Visual C#:

```
Remove
```

137.1.7. ResourceManager Property

Returns the cached ResourceManager instance used by this class.

Visual C#:

ResourceManager

137.1.8. Stop Property

Looks up a localized resource of type System.Drawing.Bitmap.

Visual C#:

Stop

138. Dorian.SecsI Class

Dorian.SecsI is a component for SEMI E.4 (SECS-I) communications.

Properties:

Properties	Description
AutoSessionID	Automatically adjust device ID
AutoSystemBytes	Automatically adjust system bytes
BaudRate	Baud rate
Connect	Open or close serial port
FileName	JSON settings file name
Heartbeat	Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.
Json	JSON settings
Master	Master or slave
MDLN	Equipment model name
Port	Serial port number
Retry	Number of retry
SessionID	Device ID
SOFTREV	Equipment software revision
SystemBytes	System bytes
T1	T1 timer
T2	T2 timer
T3	T3 timer
T4	T4 timer

Methods:

Methods	Description
Config	Open configuration dialog box
Load	Load JSON settings file.
Save	Save JSON settings file.
Send	Send the message
TimerProc	This function has to be called periodically in order for this class to work.

Events:

Events	Description
ProblemEvent	Problem event
ReceivedEvent	Received event
ReceivedRawEvent	ReceivedRaw event
SentEvent	Sent event
SentRawEvent	SentRaw event

138.1. Properties

138.1.1. AutoSessionID Property

Automatically adjust device ID

Visual C#:

```
bool AutoSessionID
```

Remarks:

Automatically adjust device ID when sending.

138.1.2. AutoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool AutoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

138.1.3. BaudRate Property

Baud rate

Visual C#:

```
int BaudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

138.1.4. Connect Property

Open or close serial port

Visual C#:

```
bool Connect
```

Remarks:

Gets or sets the SECS-I serial port connection status.

138.1.5. FileName Property

JSON settings file name

Visual C#:

```
string FileName
```

138.1.6. Heartbeat Property

Heartbeat of S1F1 in seconds. Disabled, if 0 or lower.

Visual C#:

```
int Heartbeat
```

138.1.7. Json Property

JSON settings

Visual C#:

```
string Json
```

138.1.8. Master Property

Master or slave

Visual C#:

```
bool Master
```

Remarks:

Gets or sets master/slave.

138.1.9. MDLN Property

Equipment model name

Visual C#:

```
string MDLN
```

138.1.10. Port Property

Serial port number

Visual C#:

```
int Port
```

Remarks:

Gets or sets the port number for serial port connection.

138.1.11. Retry Property

Number of retry

Visual C#:

```
int Retry
```

Remarks:

Gets or sets the number of retry.

138.1.12. SessionID Property

Device ID

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the device ID for SECS-I.

138.1.13. SOFTREV Property

Equipment software revision

Visual C#:

```
string SOFTREV
```

138.1.14. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

138.1.15. T1 Property

T1 timer

Visual C#:

```
double T1
```

Remarks:

Gets or sets the T1 time out for SECS-I in seconds. The default value is 1 second.

138.1.16. T2 Property

T2 timer

Visual C#:

```
double T2
```

Remarks:

Gets or sets the T2 time out for SECS-I in seconds. The default value is 10 seconds.

138.1.17. T3 Property

T3 timer

Visual C#:


```
double T3
```

Remarks:

Gets or sets the T3 time out for SECS-I in seconds. The default value is 45 seconds.

138.1.18. T4 Property

T4 timer

Visual C#:

```
double T4
```

Remarks:

Gets or sets the T4 time out for SECS-I in seconds. The default value is 60 seconds.

138.2. Methods**138.2.1. Config Method**

Open configuration dialog box

Visual C#:

```
public bool Config(string caption)
```

Arguments

Name	Description
caption	Caption title

Return value:

Returns true if successful. Otherwise, false.

138.2.2. Load Method

Load JSON settings file.

Visual C#:

```
public bool Load()
```

Return value:

Returns true if successful. Otherwise, false.

138.2.3. Save Method

Save JSON settings file.

Visual C#:

```
public bool Save()
```

Return value:

Returns true if successful. Otherwise, false.

138.2.4. Send Method

Send the message

Visual C#:

```
public bool Send(unsigned char[] message)
```

Remarks:

Sends specified message.

Arguments

Name	Description
message	SECS-II message

Return value:

Returns true if transmission was successful. Otherwise return false.

138.2.5. **TimerProc Method**

This function has to be called periodically in order for this class to work.

Visual C#:

```
public void TimerProc()
```

138.3. Events**138.3.1. ProblemEventEvent**

Problem event

Visual C#:

```
public void ProblemEvent(object sender, ProblemEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

138.3.2. ReceivedEventEvent

Received event

Visual C#:

```
public void ReceivedEvent(object sender, ReceivedEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

138.3.3. ReceivedRawEventEvent

ReceivedRaw event

Visual C#:

```
public void ReceivedRawEvent(object sender, ReceivedRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

138.3.4. SentEventEvent

Sent event

Visual C#:

```
public void SentEvent(object sender, SentEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

138.3.5. SentRawEventEvent

SentRaw event

Visual C#:

```
public void SentRawEvent(object sender, SentRawEventArgs e)
```

Arguments

Name	Description
sender	Sender of this event
e	Additional info

139. Dorian.Secsl.Data Class

JSON data

Properties:

Properties	Description
autoSessionID	Automatically adjust session ID
autoSystemBytes	Automatically adjust system bytes
baudRate	Baud rate
master	Master or slave
port	Serial port number
retry	Number of retry
sessionID	Session ID (HSMS)
t1	T1 timer.
t2	T2 timer.
t3	T3 timer.
t4	T4 timer.

139.1. Properties

139.1.1. autoSessionID Property

Automatically adjust session ID

Visual C#:

```
bool autoSessionID
```

Remarks:

Automatically adjust session ID when sending.

139.1.2. autoSystemBytes Property

Automatically adjust system bytes

Visual C#:

```
bool autoSystemBytes
```

Remarks:

Automatically adjust system bytes when sending.

139.1.3. baudRate Property

Baud rate

Visual C#:

```
int baudRate
```

Remarks:

Gets or sets the baud rate of the serial port.

139.1.4. master Property

Master or slave

Visual C#:

```
bool master
```

Remarks:

Gets or sets master/slave.

139.1.5. port Property

Serial port number

Visual C#:

```
int port
```

Remarks:

Gets or sets the port number for serial port connection.

139.1.6. retry Property

Number of retry

Visual C#:

```
int retry
```

Remarks:

Gets or sets the number of retry.

139.1.7. sessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 sessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

139.1.8. t1 Property

T1 timer.

Visual C#:

```
double t1
```

Remarks:

Gets or sets the T1 time out for HSMS in seconds. The default value is 1 second.

139.1.9. t2 Property

T2 timer.

Visual C#:

```
double t2
```

Remarks:

Gets or sets the T2 time out for HSMS in seconds. The default value is 10 seconds.

139.1.10. t3 Property

T3 timer.

Visual C#:

```
double t3
```

Remarks:

Gets or sets the T3 time out for HSMS in seconds. The default value is 45 seconds.

139.1.11. t4 Property

T4 timer.

Visual C#:

```
double t4
```

Remarks:

Gets or sets the T4 time out for HSMS in seconds. The default value is 60 seconds.

140. Dorian.Secsl.ProblemEventArgs Class

Problem event arguments

Properties:

Properties	Description
code	Error code
message	SECS-II message
sender	Sender of this event

140.1. Properties

140.1.1. code Property

Error code

Visual C#:

```
int code
```

140.1.2. message Property

SECS-II message

Visual C#:

```
Byte[] message
```

140.1.3. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

141. Dorian.SecsI.ReceivedEventArgs Class

Received event arguments

Properties:

Properties	Description
message	SECS-II message
sender	Sender of this event

141.1. Properties

141.1.1. message Property

SECS-II message

Visual C#:

```
Byte[] message
```

141.1.2. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

142. Dorian.SecsI.ReceivedRawEventArgs Class

ReceivedRaw event arguments

Properties:

Properties	Description
data	Character data
sender	Sender of this event

142.1. Properties

142.1.1. data Property

Character data

Visual C#:

Byte **data**

142.1.2. sender Property

Sender of this event

Visual C#:

IntPtr **sender**

143. Dorian.SecsI.SentEventArgs Class

Sent event arguments

Properties:

Properties	Description
message	SECS-II message
sender	Sender of this event
success	Succeeded or failed

143.1. Properties

143.1.1. message Property

SECS-II message

Visual C#:

```
Byte[] message
```

143.1.2. sender Property

Sender of this event

Visual C#:

```
IntPtr sender
```

143.1.3. success Property

Succeeded or failed

Visual C#:

```
bool success
```

144. Dorian.SecsI.SentRawEventArgs Class

SentRaw event arguments

Properties:

Properties	Description
data	Character data
sender	Sender of this event

144.1. Properties

144.1.1. data Property

Character data

Visual C#:

Byte **data**

144.1.2. sender Property

Sender of this event

Visual C#:

IntPtr **sender**

145. Dorian.SecslConfigDlg Class

SECS-I configuration dialog box

Properties:

Properties	Description
components	Required designer variable.
secs	Secsl object

Methods:

Methods	Description
~Dorian.SecslConfigDlg	Clean up any resources being used.
Dorian.SecslConfigDlg	Constructor
InitializeComponent	Required method for Designer support - do not modify the contents of this method with the code editor.
SecslConfigDlg_Load	On load event

145.1. Properties

145.1.1. components Property

Required designer variable.

Visual C#:

components

145.1.2. secs Property

SecsI object

Visual C#:

SecsI **secs**

145.2. Methods**145.2.1. ~Dorian.SecsIConfigDlg Method**

Clean up any resources being used.

Visual C#:

```
~Dorian.SecsIConfigDlg(bool disposing)
```

Arguments

Name	Description
disposing	true if managed resources should be disposed; otherwise, false.

Return value:

Dorian.SecsIConfigDlg Method

Constructor

Visual C#:

```
Dorian.SecsIConfigDlg()
```

Return value:

145.2.2. InitializeComponent Method

Required method for Designer support - do not modify the contents of this method with the code editor.

Visual C#:

```
InitializeComponent()
```

Return value:

145.2.3. SecsIConfigDlg_Load Method

On load event

Visual C#:

```
SecsIConfigDlg_Load(System.Object sender, System.EventArgs e)
```

Arguments

Name	Description
sender	Sender
e	Argument

Return value:

146. Dorian.SecsII Class

Dorian.SecsII component is an assistant product to develop SEMI E5 (SECS-II) compliant application software. Dorian.SecsII component can be used for either equipment side development or host side development. Usually Dorian.SecsII component can be used with Dorian.Hsms or Dorian.SecsI component.

Properties:

Properties	Description
BlockNumber	Block number (SECS-I)
DeviceID	Device ID (SECS-I)
Ebit	End bit
Error	Parse error on SML parsing
Function	Function number
Host	Host or equipment
Hsms	HSMS or SECS-I
Json	JSON
Msg	Message (binary)
Node	Node
NodeCount	Node count
NodeItem	Node item
NodeType	Node type
NodeValue	Node value in decimal
NodeValueHex	Node value in hexadecimal
PType	P-type (HSMS)
Rbit	Reverse bit (SECS-I)
SessionID	Session ID (HSMS)
Sml	SML
SmlType	SML format (under construction)
SourceID	Source ID (SECS-I)
Stream	Stream number
SType	S-type (HSMS)
SuggestedReplyMsg	Suggested reply message
SystemBytes	System bytes
this[string]	Syntax sugar for Node and NodeValue
TransactionID	Transaction ID (SECS-I)
Verification	Verification of Sml or Msg
Wbit	Wait bit

Methods:

Methods	Description
Reply	Set header for the reply
Reply	Set header for the reply
Reset	Reset message

146.1. Properties

146.1.1. BlockNumber Property

Block number (SECS-I)

Visual C#:

```
UInt16 BlockNumber
```

Remarks:

Gets or sets the block number in SECS-II header. This property is used only for SECS-I.

If BlockNumber property is not 1 on received SECS-I message, the message was multi-block message.

BlockNumber property should always be 1, when sending message. If message size exceeds the maximum size of one block, Dorian.SecsI component will automatically convert it into multi-block messages.

146.1.2. DeviceID Property

Device ID (SECS-I)

Visual C#:

```
UInt16 DeviceID
```

Remarks:

Gets or sets the device ID. Device ID is 15 bits starting at second bit of SECS-II header.

Device ID parameter will be reset by calling Reset method.

Device ID and session ID are almost same, but device ID is 15-bit, where session ID is 16-bit.

146.1.3. Ebit Property

End bit

Visual C#:

```
bool Ebit
```

Remarks:

Gets or sets the end bit in SECS-II header. This property is used only for SECS-I.

End bit of incoming SECS-I message is always true. Because Dorian.SecsI component will notify Received event after the final block was received.

146.1.4. Error Property

Parse error on SML parsing

Visual C#:

```
bool Error
```

Remarks:

Gets whether or not SML string processing was failed.

Read-only property.

146.1.5. Function Property

Function number

Visual C#:

Byte Function**Remarks:**

Gets or sets the function number in SECS-II header.

146.1.6. Host Property

Host or equipment

Visual C#:

```
bool Host
```

Remarks:

Gets or sets the role of Dorian.SecsII component. This property will affect to the result verified by Verify method, and SuggestedReplyMsg property.

146.1.7. Hsms Property

HSMS or SECS-I

Visual C#:

```
bool Hsms
```

Remarks:

Gets or sets whether Dorian.SecsII component is best match for HSMS or SECS-I. Default value is HSMS.

146.1.8. Json Property

JSON

Visual C#:

```
string Json
```

Remarks:

Gets or sets the message in JSON string. Readin Json property will convert message structure into JSON literal string. See appendix C for the details.

146.1.9. Msg Property

Message (binary)

Visual C#:

```
Byte[] Msg
```

Remarks:

Gets or sets the message data of SECS-II.

146.1.10. Node Property

Node

Visual C#:

```
string Node
```

Remarks:

Gets or sets the node for operation. Node consists of "/" (slash), node number, "[" (left bracket) and "]" (right bracket). Node number is a numeric expression starting at 1. Index number starts at 0. If node is "" (empty), it means root.

146.1.11. NodeCount Property

Node count

Visual C#:

```
int NodeCount
```

Remarks:

Gets or sets the number of sub items. If node type is list, this property means the number of sub node. Otherwise, it means number of array.
Read-only property.

146.1.12. NodeItem Property

Node item

Visual C#:

```
NodeItemClass NodeItem
```

Remarks:

Gets the node item for operation.

146.1.13. NodeType Property

Node type

Visual C#:

```
SecsType NodeType
```

Remarks:

Gets the node type.
Read-only property.

146.1.14. NodeValue Property

Node value in decimal

Visual C#:

```
string NodeValue
```

Remarks:

Gets the node value. If node is numeric type, the number will be converted into decimal literal expression.
Read-only property.

146.1.15. NodeValueHex Property

Node value in hexadecimal

Visual C#:

```
string NodeValueHex
```

Remarks:

Gets the node value. If node is numeric type, the number will be converted into hexadecimal literal expression.
Read-only property.

146.1.16. PType Property

P-type (HSMS)

Visual C#:

```
Byte PType
```

Remarks:

Gets or sets the presentation type in SECS-II header.
This property should always be 0, since SEMI E37 defines only SECS-II type at the moment.

146.1.17. Rbit Property

Reverse bit (SECS-I)

Visual C#:

```
bool Rbit
```

Remarks:

Gets or sets the reverse bit in SECS-II header.

146.1.18. SessionID Property

Session ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

Remarks:

Gets or sets the session ID for HSMS. Session ID is first 16 bits of SECS-II header.

146.1.19. Sml Property

SML

Visual C#:

```
string Sml
```

Remarks:

Gets or sets the message in SML string. Reading SML property will convert message structure into SML literal string. It is possible to insert CR (carriage return), LF (line feed), space code, tab code in SML string to set it in SML property. They would be ignored except in some context.
See appendix A for the details.

146.1.20. SmIType Property

SML format (under construction)

Visual C#:

Byte **SmIType**

Remarks:

Gets or sets the SML format.

146.1.21. SourceID Property

Source ID (SECS-I)

Visual C#:

UInt16 **SourceID**

Remarks:

Gets or sets the source ID in SECS-II header.

146.1.22. Stream Property

Stream number

Visual C#:

Byte **Stream**

Remarks:

Gets or sets the stream in SECS-II header.

146.1.23. SType Property

S-type (HSMS)

Visual C#:

Byte **SType**

Remarks:

Gets or sets the session type in SECS-II header.

146.1.24. SuggestedReplyMsg Property

Suggested reply message

Visual C#:

Byte[] **SuggestedReplyMsg**

Remarks:

Gets the most appropriate reply message determined after verifying the message structure.

146.1.25. SystemBytes Property

System bytes

Visual C#:

```
UInt32 SystemBytes
```

Remarks:

Gets or sets the system bytes in SECS-II header.

System bytes are 4-byte area and consist of source ID and transaction ID. System bytes in reply message should be identical with the ones in primary message.

146.1.26. **this[string] Property**

Syntax sugar for Node and NodeValue

Visual C#:

```
this[string]
```

Remarks:

Gets or sets the node value in hexadecimal expression.

Read-only property.

146.1.27. **TransactionID Property**

Transaction ID (SECS-I)

Visual C#:

```
UInt16 TransactionID
```

Remarks:

Gets or sets the transaction ID in SECS-II header.

146.1.28. **Verification Property**

Verification of Sml or Msg

Visual C#:

```
SecsVerify Verification
```

Remarks:

Verification result of the message structure on setting Sml property or Msg property. If the message is a primary message, suggested reply message will be set to SuggestedReplyMsg property.

146.1.29. **Wbit Property**

Wait bit

Visual C#:

```
bool wbit
```

Remarks:

Gets or sets the wait bit in SECS-II header.

If primary message requested reply message, wait bit will be true.

146.2. Methods**146.2.1. Reply Method**

Set header for the reply

Visual C#:

```
public void Reply(unsigned char[] primary)
```

Remarks:

Initializes SECS-II header as reply message of specified message. If specified message is a HSMS control message, Dorian.SecsII component will remove message body. Otherwise, message body will not be affected.

Arguments

Name	Description
primary	Primary message

146.2.2. Reply Method

Set header for the reply

Visual C#:

```
public void Reply(Dorian.ISecsII primary)
```

Remarks:

Initializes SECS-II header as reply message of specified message. If specified message is a HSMS control message, Dorian.SecsII component will remove message body. Otherwise, message body will not be affected.

Arguments

Name	Description
primary	Primary message

146.2.3. Reset Method

Reset message

Visual C#:

```
public void Reset()
```

Remarks:

Initializes internal data structure and parameters.

147. Dorian.SecsII.NodeItemClass Class

Node item

Properties:

Properties	Description
Count	Number of sub-item
Hex	Node value in hexadecimal
Index	Index for Text and Hex properties
Text	Node value
this[int]	Syntax sugar to access the sub-items
Type	Node type

Methods:

Methods	Description
Value	Node value
ValueHex	Node value in hexadecimal

147.1. Properties

147.1.1. Count Property

Number of sub-item

Visual C#:

```
int Count
```

147.1.2. Hex Property

Node value in hexadecimal

Visual C#:

```
string Hex
```

Remarks:

Gets the node value. If node is numeric type, the number will be converted into hexadecimal literal expression.
Read-only property.

147.1.3. Index Property

Index for Text and Hex properties

Visual C#:

```
int Index
```

147.1.4. Text Property

Node value

Visual C#:

```
string Text
```

Remarks:

Gets the node value. If node is numeric type, the number will be converted into decimal literal expression.
Read-only property.

147.1.5. this[int] Property

Syntax sugar to access the sub-items

Visual C#:

```
this[int]
```

147.1.6. Type Property

Node type

Visual C#:

```
SecsType Type
```

Remarks:

Gets the node type.
Read-only property.

147.2. Methods

147.2.1. Value Method

Node value

Visual C#:

```
public string Value(int index)
```

Remarks:

Gets the node value. If node is numeric type, the number will be converted into decimal literal expression.

Arguments

Name	Description
index	Index

Return value:

147.2.2. ValueHex Method

Node value in hexadecimal

Visual C#:

```
public string ValueHex(int index)
```

Remarks:

Gets the node value. If node is numeric type, the number will be converted into hexadecimal literal expression.

Arguments

Name	Description
index	Index

Return value:

148. Dorian.SecsType Enum

Node type

Values:

Values	Description
Ascii	Ascii string
Ascii2	Multi-byte character string
Binary	Binary
Boolean	Boolean
F4	4-byte floating point
F8	8-byte floating point
I1	1-byte signed integer
I2	2-byte signed integer
I4	4-byte signed integer
I8	8-byte signed integer
Invalid	Invalid type
Jis	JIS8 code (string)
List	List
U1	1-byte unsigned integer
U2	2-byte unsigned integer
U4	4-byte unsigned integer
U8	8-byte unsigned integer

149. Dorian.SecsVerify Enum

Verification result

Values:

Values	Description
Correct	Correct
Incorrect	Incorrect
IncorrectAndReply	Incorrect and need reply
NoWBit	No wait bit where it supposedly has it.
UnrecognizedFunction	Unrecognized function
UnrecognizedStream	Unrecognized stream
UserDefined	User-defined message
WBit	Wait bit where it supposedly should not have it.
WrongDirection	The direction of message is wrong.

150. Dorian.Utility Class

Dorian.Utility is a common utility.

Methods:

Methods	Description
atof	Converts string to floating point
atofex	Returns byte length of pointer
atoi	Converts string to integer
ExpireLicense	Expire license immediately
GetLicense	License counter

150.1. Methods**150.1.1. atof Method**

Converts string to floating point

Visual C#:

```
public static double atof(string value)
```

Remarks:

Converts string to floating point.

Arguments

Name	Description
value	Value

Return value:**150.1.2. atofex Method**

Returns byte length of pointer

Visual C#:

```
public static double atofex(string value)
```

Remarks:

Returns 64 if 64-bit. Otherwise, 32.

Arguments

Name	Description
value	Value

Return value:**150.1.3. atoi Method**

Converts string to integer

Visual C#:

```
public static int atoi(string value)
```

Remarks:

Converts string to integer.

Arguments

Name	Description
value	Value

Return value:**150.1.4. ExpireLicense Method**

Expire license immediately

Visual C#:

```
public static void ExpireLicense()
```

Remarks:

Expire license immediately.

150.1.5. **GetLicense Method**

License counter

Visual C#:

```
public static int GetLicense()
```

Remarks:

Returns more than 0 if license is valid.

Return value: