

Dorian.NET

マニュアル

1. 目次

1.	目次.....	2
2.	Dorian.AboutDlg Class.....	27
2.1.	プロパティ.....	28
2.1.1.	components プロパティ.....	28
2.2.	メソッド.....	29
2.2.1.	~Dorian.AboutDlg メソッド.....	29
2.2.2.	InitializeComponent メソッド.....	29
3.	Dorian.Comm Class.....	30
3.1.	プロパティ.....	31
3.1.1.	BaudRate プロパティ.....	31
3.1.2.	ByteSize プロパティ.....	31
3.1.3.	Connect プロパティ.....	31
3.1.4.	Count プロパティ.....	31
3.1.5.	Data プロパティ.....	31
3.1.6.	FileName プロパティ.....	32
3.1.7.	Json プロパティ.....	32
3.1.8.	Parity プロパティ.....	32
3.1.9.	Port プロパティ.....	32
3.1.10.	StopBit プロパティ.....	32
3.2.	メソッド.....	33
3.2.1.	Config メソッド.....	33
3.2.2.	Load メソッド.....	33
3.2.3.	Read メソッド.....	33
3.2.4.	Save メソッド.....	33
3.2.5.	TimerProc メソッド.....	34
3.2.6.	Write メソッド.....	34
3.3.	イベント.....	35
3.3.1.	ReceivedEvent イベント.....	35
4.	Dorian.Comm.ReceivedEventArgs Class.....	36
4.1.	プロパティ.....	37
4.1.1.	data プロパティ.....	37
4.1.2.	sender プロパティ.....	37
5.	Dorian.CommConfigDlg Class.....	38
5.1.	プロパティ.....	39
5.1.1.	comm プロパティ.....	39
5.1.2.	components プロパティ.....	39
5.2.	メソッド.....	40
5.2.1.	~Dorian.CommConfigDlg メソッド.....	40
5.2.2.	CommConfigDlg_Load メソッド.....	40
5.2.3.	Dorian.CommConfigDlg メソッド.....	40
5.2.4.	InitializeComponent メソッド.....	40
6.	Dorian.Communication Class.....	42
6.1.	プロパティ.....	44
6.1.1.	AutoReconnect プロパティ.....	44
6.1.2.	AutoSelectReq プロパティ.....	44
6.1.3.	AutoSessionID プロパティ.....	44
6.1.4.	AutoSystemBytes プロパティ.....	44
6.1.5.	BaudRate プロパティ.....	44
6.1.6.	Clients プロパティ.....	45
6.1.7.	Connect プロパティ.....	45
6.1.8.	FileNameHsms プロパティ.....	45
6.1.9.	FileNameSecsl プロパティ.....	45
6.1.10.	HandleCtrlMsg プロパティ.....	45
6.1.11.	Heartbeat プロパティ.....	46
6.1.12.	Hsms プロパティ.....	46
6.1.13.	IP プロパティ.....	46
6.1.14.	IPv4 プロパティ.....	46

6.1.15.	Json プロパティ	46
6.1.16.	Master プロパティ	47
6.1.17.	MDLN プロパティ	47
6.1.18.	Port プロパティ	47
6.1.19.	Retry プロパティ.....	47
6.1.20.	Selected プロパティ	47
6.1.21.	SerialPort プロパティ.....	48
6.1.22.	Server プロパティ.....	48
6.1.23.	SessionID プロパティ	48
6.1.24.	SOFTREV プロパティ.....	48
6.1.25.	SystemBytes プロパティ.....	48
6.1.26.	T1 プロパティ.....	49
6.1.27.	T2 プロパティ.....	49
6.1.28.	T3 プロパティ.....	49
6.1.29.	T4 プロパティ.....	49
6.1.30.	T5 プロパティ.....	49
6.1.31.	T6 プロパティ.....	50
6.1.32.	T7 プロパティ.....	50
6.1.33.	T8 プロパティ.....	50
6.1.34.	Transactions プロパティ	50
6.2.	メソッド.....	51
6.2.1.	Config メソッド	51
6.2.2.	Load メソッド	51
6.2.3.	Save メソッド	51
6.2.4.	Send メソッド.....	51
6.2.5.	TimerProc メソッド	52
6.3.	イベント	53
6.3.1.	ConnectedEvent イベント.....	53
6.3.2.	DisconnectedEvent イベント.....	53
6.3.3.	ProblemEvent イベント.....	53
6.3.4.	ReceivedEvent イベント	53
6.3.5.	ReceivedRawEvent イベント	54
6.3.6.	SentEvent イベント	54
6.3.7.	SentRawEvent イベント	54
7.	Dorian.Forms.Comm Class	55
7.1.	プロパティ.....	56
7.1.1.	BaudRate プロパティ.....	56
7.1.2.	ByteSize プロパティ	56
7.1.3.	Connect プロパティ	56
7.1.4.	Count プロパティ.....	56
7.1.5.	Data プロパティ.....	56
7.1.6.	FileName プロパティ	57
7.1.7.	Json プロパティ.....	57
7.1.8.	Parity プロパティ.....	57
7.1.9.	Port プロパティ.....	57
7.1.10.	StopBit プロパティ.....	57
7.2.	メソッド.....	58
7.2.1.	Config メソッド	58
7.2.2.	Load メソッド	58
7.2.3.	Read メソッド.....	58
7.2.4.	Save メソッド	58
7.2.5.	Write メソッド.....	59
7.3.	イベント	60
7.3.1.	ReceivedEvent メソッド.....	60
8.	Dorian.Forms.Communication Class.....	61
8.1.	プロパティ.....	63
8.1.1.	AutoReconnect プロパティ.....	63
8.1.2.	AutoSelectReq プロパティ	63

8.1.3.	AutoSessionID プロパティ	63
8.1.4.	AutoSystemBytes プロパティ	63
8.1.5.	BaudRate プロパティ	63
8.1.6.	Clients プロパティ	64
8.1.7.	Connect プロパティ	64
8.1.8.	FileNameHsms プロパティ	64
8.1.9.	FileNameSecsl プロパティ	64
8.1.10.	HandleCtrlMsg プロパティ	64
8.1.11.	Heartbeat プロパティ	65
8.1.12.	Hsms プロパティ	65
8.1.13.	IP プロパティ	65
8.1.14.	IPv4 プロパティ	65
8.1.15.	Json プロパティ	65
8.1.16.	Master プロパティ	66
8.1.17.	MDLN プロパティ	66
8.1.18.	Port プロパティ	66
8.1.19.	Retry プロパティ	66
8.1.20.	Selected プロパティ	66
8.1.21.	SerialPort プロパティ	67
8.1.22.	Server プロパティ	67
8.1.23.	SessionID プロパティ	67
8.1.24.	SOFTREV プロパティ	67
8.1.25.	SystemBytes プロパティ	67
8.1.26.	T1 プロパティ	68
8.1.27.	T2 プロパティ	68
8.1.28.	T3 プロパティ	68
8.1.29.	T4 プロパティ	68
8.1.30.	T5 プロパティ	68
8.1.31.	T6 プロパティ	69
8.1.32.	T7 プロパティ	69
8.1.33.	T8 プロパティ	69
8.1.34.	Transactions プロパティ	69
8.2.	メソッド	70
8.2.1.	Config メソッド	70
8.2.2.	Load メソッド	70
8.2.3.	Save メソッド	70
8.2.4.	Send メソッド	70
8.3.	イベント	72
8.3.1.	ConnectedEvent イベント	72
8.3.2.	DisconnectedEvent イベント	72
8.3.3.	ProblemEvent イベント	72
8.3.4.	ReceivedEvent イベント	72
8.3.5.	ReceivedRawEvent イベント	73
8.3.6.	SentEvent イベント	73
8.3.7.	SentRawEvent イベント	73
9.	Dorian.Forms.CommunicationView Class	74
9.1.	プロパティ	76
9.1.1.	AutoReconnect プロパティ	76
9.1.2.	AutoSelectReq プロパティ	76
9.1.3.	AutoSessionID プロパティ	76
9.1.4.	AutoSystemBytes プロパティ	76
9.1.5.	BaudRate プロパティ	76
9.1.6.	Clients プロパティ	77
9.1.7.	Connect プロパティ	77
9.1.8.	FileNameHsms プロパティ	77
9.1.9.	FileNameSecsl プロパティ	77
9.1.10.	HandleCtrlMsg プロパティ	77
9.1.11.	Heartbeat プロパティ	78

9.1.12.	Hsms プロパティ	78
9.1.13.	IP プロパティ	78
9.1.14.	IPv4 プロパティ	78
9.1.15.	Json プロパティ	78
9.1.16.	Master プロパティ	79
9.1.17.	MDLN プロパティ	79
9.1.18.	Port プロパティ	79
9.1.19.	Retry プロパティ	79
9.1.20.	Selected プロパティ	79
9.1.21.	SerialPort プロパティ	80
9.1.22.	Server プロパティ	80
9.1.23.	SessionID プロパティ	80
9.1.24.	SOFTREV プロパティ	80
9.1.25.	SystemBytes プロパティ	80
9.1.26.	T1 プロパティ	81
9.1.27.	T2 プロパティ	81
9.1.28.	T3 プロパティ	81
9.1.29.	T4 プロパティ	81
9.1.30.	T5 プロパティ	81
9.1.31.	T6 プロパティ	82
9.1.32.	T7 プロパティ	82
9.1.33.	T8 プロパティ	82
9.1.34.	Transactions プロパティ	82
9.2.	メソッド	83
9.2.1.	Config メソッド	83
9.2.2.	Load メソッド	83
9.2.3.	Save メソッド	83
9.2.4.	Send メソッド	83
9.3.	イベント	85
9.3.1.	ConnectedEvent イベント	85
9.3.2.	DisconnectedEvent イベント	85
9.3.3.	ProblemEvent イベント	85
9.3.4.	ReceivedEvent イベント	85
9.3.5.	ReceivedRawEvent イベント	86
9.3.6.	SentEvent イベント	86
9.3.7.	SentRawEvent イベント	86
10.	Dorian.Forms.Gem Class	87
10.1.	プロパティ	88
10.1.1.	AlarmModel プロパティ	88
10.1.2.	AutoReconnect プロパティ	88
10.1.3.	AutoSelectReq プロパティ	88
10.1.4.	AutoSessionID プロパティ	88
10.1.5.	AutoSystemBytes プロパティ	88
10.1.6.	CallDefProc プロパティ	88
10.1.7.	Clients プロパティ	89
10.1.8.	CommModel プロパティ	89
10.1.9.	Connect プロパティ	89
10.1.10.	CtrlModel プロパティ	89
10.1.11.	EventModel プロパティ	89
10.1.12.	FileName プロパティ	90
10.1.13.	HandleCtrlMsg プロパティ	90
10.1.14.	IP プロパティ	90
10.1.15.	IPv4 プロパティ	90
10.1.16.	Json プロパティ	90
10.1.17.	Msg プロパティ	91
10.1.18.	Port プロパティ	91
10.1.19.	Selected プロパティ	91
10.1.20.	Server プロパティ	91

10.1.21.	SessionID プロパティ.....	91
10.1.22.	SystemBytes プロパティ.....	92
10.1.23.	T3 プロパティ.....	92
10.1.24.	T5 プロパティ.....	92
10.1.25.	T6 プロパティ.....	92
10.1.26.	T7 プロパティ.....	92
10.1.27.	T8 プロパティ.....	93
10.2.	メソッド.....	94
10.2.1.	Config メソッド.....	94
10.2.2.	InvokeAlarm メソッド.....	94
10.2.3.	Load メソッド.....	94
10.2.4.	Save メソッド.....	94
10.2.5.	Send メソッド.....	95
10.3.	イベント.....	96
10.3.1.	CommStateChangedEvent メソッド.....	96
10.3.2.	ConnectedEvent メソッド.....	96
10.3.3.	CtrlStateChangedEvent メソッド.....	96
10.3.4.	DisconnectedEvent メソッド.....	96
10.3.5.	InvokeEvent メソッド.....	97
10.3.6.	ProblemEvent メソッド.....	97
10.3.7.	ReceivedEvent メソッド.....	97
10.3.8.	SentEvent メソッド.....	97
11.	Dorian.Forms.GemView Class.....	99
11.1.	プロパティ.....	100
11.1.1.	AlarmModel プロパティ.....	100
11.1.2.	AutoReconnect プロパティ.....	100
11.1.3.	AutoSelectReq プロパティ.....	100
11.1.4.	AutoSessionID プロパティ.....	100
11.1.5.	AutoSystemBytes プロパティ.....	100
11.1.6.	CallDefProc プロパティ.....	100
11.1.7.	Clients プロパティ.....	101
11.1.8.	CommModel プロパティ.....	101
11.1.9.	Connect プロパティ.....	101
11.1.10.	CtrlModel プロパティ.....	101
11.1.11.	EventModel プロパティ.....	101
11.1.12.	FileName プロパティ.....	102
11.1.13.	HandleCtrlMsg プロパティ.....	102
11.1.14.	IP プロパティ.....	102
11.1.15.	IPv4 プロパティ.....	102
11.1.16.	Json プロパティ.....	102
11.1.17.	Msg プロパティ.....	103
11.1.18.	Port プロパティ.....	103
11.1.19.	Selected プロパティ.....	103
11.1.20.	Server プロパティ.....	103
11.1.21.	SessionID プロパティ.....	103
11.1.22.	SystemBytes プロパティ.....	104
11.1.23.	T3 プロパティ.....	104
11.1.24.	T5 プロパティ.....	104
11.1.25.	T6 プロパティ.....	104
11.1.26.	T7 プロパティ.....	104
11.1.27.	T8 プロパティ.....	105
11.2.	メソッド.....	106
11.2.1.	Config メソッド.....	106
11.2.2.	InvokeAlarm メソッド.....	106
11.2.3.	Load メソッド.....	106
11.2.4.	Save メソッド.....	106
11.2.5.	Send メソッド.....	107
11.3.	イベント.....	108

11.3.1.	CommStateChangedEvent メソッド	108
11.3.2.	ConnectedEvent メソッド	108
11.3.3.	CtrlStateChangedEvent メソッド	108
11.3.4.	DisconnectedEvent メソッド	108
11.3.5.	InvokeEvent メソッド	109
11.3.6.	ProblemEvent メソッド	109
11.3.7.	ReceivedEvent メソッド	109
11.3.8.	SentEvent メソッド	109
12.	Dorian.Forms.Hsms Class	111
12.1.	プロパティ	112
12.1.1.	AutoReconnect プロパティ	112
12.1.2.	AutoSelectReq プロパティ	112
12.1.3.	AutoSessionID プロパティ	112
12.1.4.	AutoSystemBytes プロパティ	112
12.1.5.	Clients プロパティ	112
12.1.6.	Connect プロパティ	112
12.1.7.	FileName プロパティ	113
12.1.8.	HandleCtrlMsg プロパティ	113
12.1.9.	Heartbeat プロパティ	113
12.1.10.	IP プロパティ	113
12.1.11.	IPv4 プロパティ	114
12.1.12.	Json プロパティ	114
12.1.13.	MDLN プロパティ	114
12.1.14.	Port プロパティ	114
12.1.15.	Selected プロパティ	114
12.1.16.	Server プロパティ	115
12.1.17.	SessionID プロパティ	115
12.1.18.	SOFTREV プロパティ	115
12.1.19.	SystemBytes プロパティ	115
12.1.20.	T3 プロパティ	115
12.1.21.	T5 プロパティ	116
12.1.22.	T6 プロパティ	116
12.1.23.	T7 プロパティ	116
12.1.24.	T8 プロパティ	116
12.1.25.	Transactions プロパティ	116
12.2.	メソッド	118
12.2.1.	Config メソッド	118
12.2.2.	Load メソッド	118
12.2.3.	Save メソッド	118
12.2.4.	Send メソッド	118
12.3.	イベント	120
12.3.1.	ConnectedEvent メソッド	120
12.3.2.	DisconnectedEvent メソッド	120
12.3.3.	ProblemEvent メソッド	120
12.3.4.	ReceivedEvent メソッド	120
12.3.5.	SentEvent メソッド	121
13.	Dorian.Forms.HsmsView Class	122
13.1.	プロパティ	123
13.1.1.	AutoReconnect プロパティ	123
13.1.2.	AutoSelectReq プロパティ	123
13.1.3.	AutoSessionID プロパティ	123
13.1.4.	AutoSystemBytes プロパティ	123
13.1.5.	Clients プロパティ	123
13.1.6.	Connect プロパティ	123
13.1.7.	FileName プロパティ	124
13.1.8.	HandleCtrlMsg プロパティ	124
13.1.9.	Heartbeat プロパティ	124
13.1.10.	IP プロパティ	124

13.1.11.	IPv4 プロパティ.....	125
13.1.12.	Json プロパティ	125
13.1.13.	MDLN プロパティ.....	125
13.1.14.	Port プロパティ.....	125
13.1.15.	Selected プロパティ.....	125
13.1.16.	Server プロパティ.....	126
13.1.17.	SessionID プロパティ.....	126
13.1.18.	SOFTREV プロパティ.....	126
13.1.19.	SystemBytes プロパティ.....	126
13.1.20.	T3 プロパティ.....	126
13.1.21.	T5 プロパティ.....	127
13.1.22.	T6 プロパティ.....	127
13.1.23.	T7 プロパティ.....	127
13.1.24.	T8 プロパティ.....	127
13.1.25.	Transactions プロパティ.....	127
13.2.	メソッド.....	129
13.2.1.	Config メソッド.....	129
13.2.2.	Load メソッド.....	129
13.2.3.	Save メソッド.....	129
13.2.4.	Send メソッド.....	129
13.3.	イベント.....	131
13.3.1.	ConnectedEvent メソッド.....	131
13.3.2.	DisconnectedEvent メソッド.....	131
13.3.3.	ProblemEvent メソッド.....	131
13.3.4.	ReceivedEvent メソッド.....	131
13.3.5.	SentEvent メソッド.....	132
14.	Dorian.Forms.Klarf12 Class.....	133
14.1.	プロパティ.....	134
14.1.1.	Count プロパティ.....	134
14.1.2.	this[int]プロパティ.....	134
14.1.3.	this[string]プロパティ.....	134
14.2.	メソッド.....	135
14.2.1.	Find メソッド.....	135
14.2.2.	Parse メソッド.....	135
15.	Dorian.Forms.Klarf12Map Class.....	136
15.1.	プロパティ.....	137
15.1.1.	Index プロパティ.....	137
15.1.2.	klarf プロパティ.....	137
15.1.3.	XY プロパティ.....	137
16.	Dorian.Forms.Klarf12Table Class.....	138
16.1.	プロパティ.....	139
16.1.1.	Index プロパティ.....	139
16.1.2.	klarf プロパティ.....	139
17.	Dorian.Forms.Log Class.....	140
17.1.	プロパティ.....	141
17.1.1.	Count プロパティ.....	141
17.1.2.	Enable プロパティ.....	141
17.1.3.	FileName プロパティ.....	141
17.1.4.	FileSize プロパティ.....	141
17.1.5.	Json プロパティ.....	141
17.1.6.	JsonFileName プロパティ.....	142
17.1.7.	TimeStamp プロパティ.....	142
17.2.	メソッド.....	143
17.2.1.	Config メソッド.....	143
17.2.2.	Load メソッド.....	143
17.2.3.	Save メソッド.....	143
17.2.4.	Write メソッド.....	143
18.	Dorian.Forms.LogView Class.....	145

18.1.	プロパティ.....	146
18.1.1.	Count プロパティ.....	146
18.1.2.	Enable プロパティ.....	146
18.1.3.	FileName プロパティ.....	146
18.1.4.	FileSize プロパティ.....	146
18.1.5.	Json プロパティ.....	146
18.1.6.	JsonFileName プロパティ.....	147
18.1.7.	TimeStamp プロパティ.....	147
18.2.	メソッド.....	148
18.2.1.	Config メソッド.....	148
18.2.2.	Load メソッド.....	148
18.2.3.	Save メソッド.....	148
18.2.4.	Write メソッド.....	148
19.	Dorian.Forms.SecsI Class.....	150
19.1.1.	AutoSessionID プロパティ.....	151
19.1.2.	AutoSystemBytes プロパティ.....	151
19.1.3.	BaudRate プロパティ.....	151
19.1.4.	Connect プロパティ.....	151
19.1.5.	FileName プロパティ.....	151
19.1.6.	Heartbeat プロパティ.....	152
19.1.7.	Json プロパティ.....	152
19.1.8.	Master プロパティ.....	152
19.1.9.	MDLN プロパティ.....	152
19.1.10.	Port プロパティ.....	152
19.1.11.	Retry プロパティ.....	152
19.1.12.	SessionID プロパティ.....	153
19.1.13.	SOFTREV プロパティ.....	153
19.1.14.	SystemBytes プロパティ.....	153
19.1.15.	T1 プロパティ.....	153
19.1.16.	T2 プロパティ.....	153
19.1.17.	T3 プロパティ.....	154
19.1.18.	T4 プロパティ.....	154
19.2.	メソッド.....	155
19.2.1.	Config メソッド.....	155
19.2.2.	Load メソッド.....	155
19.2.3.	Save メソッド.....	155
19.2.4.	Send メソッド.....	155
19.3.	イベント.....	157
19.3.1.	ProblemEvent メソッド.....	157
19.3.2.	ReceivedEvent メソッド.....	157
19.3.3.	ReceivedRawEvent メソッド.....	157
19.3.4.	SentEvent メソッド.....	157
19.3.5.	SentRawEvent メソッド.....	158
20.	Dorian.Forms.SecsII Class.....	159
20.1.	プロパティ.....	160
20.1.1.	BlockNumber プロパティ.....	160
20.1.2.	DeviceID プロパティ.....	160
20.1.3.	Ebit プロパティ.....	160
20.1.4.	Error プロパティ.....	160
20.1.5.	Function プロパティ.....	160
20.1.6.	Host プロパティ.....	161
20.1.7.	Hsms プロパティ.....	161
20.1.8.	Json プロパティ.....	161
20.1.9.	Msg プロパティ.....	161
20.1.10.	Node プロパティ.....	162
20.1.11.	NodeCount プロパティ.....	162
20.1.12.	NodeType プロパティ.....	162
20.1.13.	NodeValue プロパティ.....	162

20.1.14.	NodeValueHex プロパティ.....	162
20.1.15.	PType プロパティ.....	163
20.1.16.	Rbit プロパティ.....	163
20.1.17.	SessionID プロパティ.....	163
20.1.18.	Sml プロパティ.....	163
20.1.19.	SmlType プロパティ.....	164
20.1.20.	SourcelD プロパティ.....	164
20.1.21.	Stream プロパティ.....	164
20.1.22.	SType プロパティ.....	164
20.1.23.	SuggestedReplyMsg プロパティ.....	164
20.1.24.	SystemBytes プロパティ.....	165
20.1.25.	this[string]プロパティ.....	165
20.1.26.	TransactionID プロパティ.....	165
20.1.27.	Verification プロパティ.....	165
20.1.28.	Wbit プロパティ.....	165
20.2.	メソッド.....	167
20.2.1.	Reply メソッド.....	167
20.2.2.	Reply メソッド.....	167
20.2.3.	Reset メソッド.....	167
21.	Dorian.Forms.SecsIIView Class.....	168
21.1.	プロパティ.....	169
21.1.1.	BlockNumber プロパティ.....	169
21.1.2.	DeviceID プロパティ.....	169
21.1.3.	Ebit プロパティ.....	169
21.1.4.	Error プロパティ.....	169
21.1.5.	Function プロパティ.....	169
21.1.6.	Host プロパティ.....	170
21.1.7.	Hsms プロパティ.....	170
21.1.8.	Json プロパティ.....	170
21.1.9.	Msg プロパティ.....	170
21.1.10.	Node プロパティ.....	171
21.1.11.	NodeCount プロパティ.....	171
21.1.12.	Nodeltem プロパティ.....	171
21.1.13.	NodeType プロパティ.....	171
21.1.14.	NodeValue プロパティ.....	171
21.1.15.	NodeValueHex プロパティ.....	172
21.1.16.	PType プロパティ.....	172
21.1.17.	Rbit プロパティ.....	172
21.1.18.	SessionID プロパティ.....	172
21.1.19.	Sml プロパティ.....	172
21.1.20.	SmlType プロパティ.....	173
21.1.21.	SourcelD プロパティ.....	173
21.1.22.	Stream プロパティ.....	173
21.1.23.	SType プロパティ.....	173
21.1.24.	SuggestedReplyMsg プロパティ.....	174
21.1.25.	SystemBytes プロパティ.....	174
21.1.26.	this[string]プロパティ.....	174
21.1.27.	TransactionID プロパティ.....	174
21.1.28.	Verification プロパティ.....	174
21.1.29.	Wbit プロパティ.....	175
21.2.	メソッド.....	176
21.2.1.	Reply メソッド.....	176
21.2.2.	Reply メソッド.....	176
21.2.3.	Reset メソッド.....	176
22.	Dorian.Forms.SecsIView Class.....	177
22.1.	プロパティ.....	178
22.1.1.	AutoSessionID プロパティ.....	178
22.1.2.	AutoSystemBytes プロパティ.....	178

22.1.3.	BaudRate プロパティ	178
22.1.4.	Connect プロパティ.....	178
22.1.5.	FileName プロパティ	178
22.1.6.	Heartbeat プロパティ.....	179
22.1.7.	Json プロパティ	179
22.1.8.	Master プロパティ	179
22.1.9.	MDLN プロパティ	179
22.1.10.	Port プロパティ	179
22.1.11.	Retry プロパティ.....	179
22.1.12.	SessionID プロパティ	180
22.1.13.	SOFTREV プロパティ.....	180
22.1.14.	SystemBytes プロパティ.....	180
22.1.15.	T1 プロパティ.....	180
22.1.16.	T2 プロパティ.....	180
22.1.17.	T3 プロパティ.....	181
22.1.18.	T4 プロパティ.....	181
22.2.	メソッド.....	182
22.2.1.	Config メソッド.....	182
22.2.2.	Load メソッド.....	182
22.2.3.	Save メソッド.....	182
22.2.4.	Send メソッド.....	182
22.3.	イベント.....	184
22.3.1.	ProblemEvent メソッド	184
22.3.2.	ReceivedEvent メソッド.....	184
22.3.3.	ReceivedRawEvent メソッド.....	184
22.3.4.	SentEvent メソッド	184
22.3.5.	SentRawEvent メソッド	185
23.	Dorian.Gem Class.....	186
23.1.	プロパティ.....	187
23.1.1.	AlarmModel プロパティ	187
23.1.2.	CallDefProc プロパティ	187
23.1.3.	CommModel プロパティ	187
23.1.4.	CtrlModel プロパティ	187
23.1.5.	EventModel プロパティ.....	187
23.1.6.	FileName プロパティ	187
23.1.7.	Hsms プロパティ.....	187
23.1.8.	Json プロパティ	188
23.1.9.	Msg プロパティ	188
23.2.	メソッド.....	189
23.2.1.	Config メソッド.....	189
23.2.2.	InvokeAlarm メソッド.....	189
23.2.3.	Load メソッド.....	189
23.2.4.	Save メソッド.....	189
23.2.5.	TimerProc メソッド.....	190
23.3.	イベント.....	191
23.3.1.	CommStateChangedEvent イベント	191
23.3.2.	ConnectedEvent イベント.....	191
23.3.3.	CtrlStateChangedEvent イベント	191
23.3.4.	DisconnectedEvent イベント	191
23.3.5.	InvokeEvent メソッド.....	192
23.3.6.	ProblemEvent イベント	192
23.3.7.	ReceivedEvent イベント.....	192
23.3.8.	SentEvent イベント.....	192
24.	Dorian.Gem.AlarmModelClass Class.....	194
24.1.	プロパティ.....	195
24.1.1.	Count プロパティ.....	195
24.1.2.	this[int]プロパティ.....	195
24.2.	メソッド.....	196

24.2.1.	Add メソッド.....	196
24.2.2.	Remove メソッド.....	196
25.	Dorian.Gem.AlarmModelClass.Alarm Class.....	197
25.1.	プロパティ.....	198
25.1.1.	Code プロパティ.....	198
25.1.2.	Enable プロパティ.....	198
25.1.3.	ID プロパティ.....	198
25.1.4.	Text プロパティ.....	198
26.	Dorian.Gem.CommModelClass Class.....	199
26.1.	プロパティ.....	200
26.1.1.	CommDelayTimer プロパティ.....	200
26.1.2.	EstablishComm プロパティ.....	200
26.1.3.	InitialState プロパティ.....	200
26.1.4.	MDLN プロパティ.....	200
26.1.5.	SOFTREV プロパティ.....	200
26.1.6.	State プロパティ.....	200
27.	Dorian.Gem.CommStateChangedEventArgs Class.....	201
27.1.	プロパティ.....	202
27.1.1.	newState プロパティ.....	202
27.1.2.	prevState プロパティ.....	202
27.1.3.	sender プロパティ.....	202
28.	Dorian.Gem.ConnectedEventArgs Class.....	203
28.1.	プロパティ.....	204
28.1.1.	ip プロパティ.....	204
28.1.2.	port プロパティ.....	204
28.1.3.	sender プロパティ.....	204
29.	Dorian.Gem.CtrlModelClass Class.....	205
29.1.	プロパティ.....	206
29.1.1.	InitialOfflineState プロパティ.....	206
29.1.2.	InitialState プロパティ.....	206
29.1.3.	OfflineState プロパティ.....	206
29.1.4.	OnlineState プロパティ.....	206
29.1.5.	State プロパティ.....	206
30.	Dorian.Gem.CtrlStateChangedEventArgs Class.....	207
30.1.	プロパティ.....	208
30.1.1.	newState プロパティ.....	208
30.1.2.	prevState プロパティ.....	208
30.1.3.	sender プロパティ.....	208
31.	Dorian.Gem.Data Class.....	209
31.1.	プロパティ.....	210
31.1.1.	alarmModel プロパティ.....	210
31.1.2.	commModel プロパティ.....	210
31.1.3.	ctrlModel プロパティ.....	210
31.1.4.	eventModel プロパティ.....	210
31.1.5.	hsms プロパティ.....	210
32.	Dorian.Gem.Data.AlarmModel Class.....	211
32.1.	プロパティ.....	212
32.1.1.	alarms プロパティ.....	212
33.	Dorian.Gem.Data.AlarmModel.Alarm Class.....	213
33.1.	プロパティ.....	214
33.1.1.	code プロパティ.....	214
33.1.2.	enable プロパティ.....	214
33.1.3.	id プロパティ.....	214
33.1.4.	text プロパティ.....	214
34.	Dorian.Gem.Data.AlarmModel.Alarms Class.....	215
35.	Dorian.Gem.Data.CommModel Class.....	216
35.1.	プロパティ.....	217
35.1.1.	establishComm プロパティ.....	217
35.1.2.	initialState プロパティ.....	217

35.1.3.	mdlIn プロパティ	217
35.1.4.	softrev プロパティ	217
36.	Dorian.Gem.Data.CtrlModel Class	218
36.1.	プロパティ	219
36.1.1.	initialOfflineState プロパティ	219
36.1.2.	initialState プロパティ	219
36.1.3.	offlineState プロパティ	219
36.1.4.	onlineState プロパティ	219
37.	Dorian.Gem.Data.EventModel Class	220
37.1.	プロパティ	221
37.1.1.	events プロパティ	221
37.1.2.	reports プロパティ	221
37.1.3.	variables プロパティ	221
38.	Dorian.Gem.Data.EventModel.Event Class	222
38.1.	プロパティ	223
38.1.1.	definition プロパティ	223
38.1.2.	description プロパティ	223
38.1.3.	enable プロパティ	223
38.1.4.	id プロパティ	223
38.1.5.	reports プロパティ	223
39.	Dorian.Gem.Data.EventModel.Events Class	224
40.	Dorian.Gem.Data.EventModel.Report Class	225
40.1.	プロパティ	226
40.1.1.	description プロパティ	226
40.1.2.	id プロパティ	226
40.1.3.	variables プロパティ	226
41.	Dorian.Gem.Data.EventModel.Reports Class	227
42.	Dorian.Gem.Data.EventModel.Variable Class	228
42.1.	プロパティ	229
42.1.1.	defaultValue プロパティ	229
42.1.2.	definition プロパティ	229
42.1.3.	description プロパティ	229
42.1.4.	id プロパティ	229
42.1.5.	max プロパティ	229
42.1.6.	min プロパティ	229
42.1.7.	secsType プロパティ	229
42.1.8.	sml プロパティ	230
42.1.9.	type プロパティ	230
42.1.10.	unit プロパティ	230
43.	Dorian.Gem.Data.EventModel.Variables Class	231
44.	Dorian.Gem.DisconnectedEventArgs Class	232
44.1.	プロパティ	233
44.1.1.	ip プロパティ	233
44.1.2.	port プロパティ	233
44.1.3.	sender プロパティ	233
45.	Dorian.Gem.EventModelClass Class	234
45.1.	プロパティ	235
45.1.1.	Events プロパティ	235
45.1.2.	Reports プロパティ	235
45.1.3.	Variables プロパティ	235
46.	Dorian.Gem.EventModelClass.Event Class	236
46.1.	プロパティ	237
46.1.1.	Count プロパティ	237
46.1.2.	Definition プロパティ	237
46.1.3.	Description プロパティ	237
46.1.4.	Enable プロパティ	237
46.1.5.	ID プロパティ	237
46.1.6.	Reports プロパティ	237
46.1.7.	this[int]プロパティ	237

46.2.	メソッド.....	239
46.2.1.	AddReport メソッド.....	239
46.2.2.	ClearReports メソッド.....	239
47.	Dorian.Gem.EventModelClass.EventsClass Class.....	240
47.1.	プロパティ.....	241
47.1.1.	Count プロパティ.....	241
47.1.2.	this[int]プロパティ.....	241
47.2.	メソッド.....	242
47.2.1.	Add メソッド.....	242
47.2.2.	Remove メソッド.....	242
48.	Dorian.Gem.EventModelClass.Report Class.....	243
48.1.	プロパティ.....	244
48.1.1.	Count プロパティ.....	244
48.1.2.	Description プロパティ.....	244
48.1.3.	ID プロパティ.....	244
48.1.4.	this[int]プロパティ.....	244
48.1.5.	Variables プロパティ.....	244
48.2.	メソッド.....	245
48.2.1.	AddVariable メソッド.....	245
48.2.2.	ClearVariables メソッド.....	245
49.	Dorian.Gem.EventModelClass.ReportsClass Class.....	246
49.1.	プロパティ.....	247
49.1.1.	Count プロパティ.....	247
49.1.2.	this[int]プロパティ.....	247
49.2.	メソッド.....	248
49.2.1.	Add メソッド.....	248
49.2.2.	Remove メソッド.....	248
50.	Dorian.Gem.EventModelClass.Variable Class.....	249
50.1.	プロパティ.....	250
50.1.1.	DefaultValue プロパティ.....	250
50.1.2.	Definition プロパティ.....	250
50.1.3.	Description プロパティ.....	250
50.1.4.	ID プロパティ.....	250
50.1.5.	Max プロパティ.....	250
50.1.6.	Min プロパティ.....	250
50.1.7.	SecsType プロパティ.....	250
50.1.8.	Sml プロパティ.....	251
50.1.9.	Type プロパティ.....	251
50.1.10.	Unit プロパティ.....	251
51.	Dorian.Gem.EventModelClass.VariablesClass Class.....	252
51.1.	プロパティ.....	253
51.1.1.	Count プロパティ.....	253
51.1.2.	this[int]プロパティ.....	253
51.2.	メソッド.....	254
51.2.1.	Add メソッド.....	254
51.2.2.	Remove メソッド.....	254
52.	Dorian.Gem.HsmsClass Class.....	255
52.1.	プロパティ.....	256
52.1.1.	AutoReconnect プロパティ.....	256
52.1.2.	AutoSelectReq プロパティ.....	256
52.1.3.	AutoSessionID プロパティ.....	256
52.1.4.	AutoSystemBytes プロパティ.....	256
52.1.5.	Connect プロパティ.....	256
52.1.6.	HandleCtrlMsg プロパティ.....	257
52.1.7.	IP プロパティ.....	257
52.1.8.	IPv4 プロパティ.....	257
52.1.9.	Port プロパティ.....	257
52.1.10.	Selected プロパティ.....	258

52.1.11.	Server プロパティ.....	258
52.1.12.	SessionID プロパティ.....	258
52.1.13.	SystemBytes プロパティ.....	258
52.1.14.	T3 プロパティ.....	258
52.1.15.	T5 プロパティ.....	259
52.1.16.	T6 プロパティ.....	259
52.1.17.	T7 プロパティ.....	259
52.1.18.	T8 プロパティ.....	259
52.2.	メソッド.....	260
52.2.1.	Send メソッド.....	260
53.	Dorian.Gem.ProblemEventArgs Class.....	261
53.1.	プロパティ.....	262
53.1.1.	code プロパティ.....	262
53.1.2.	ip プロパティ.....	262
53.1.3.	message プロパティ.....	262
53.1.4.	port プロパティ.....	262
53.1.5.	sender プロパティ.....	262
54.	Dorian.Gem.ReceivedEventArgs Class.....	263
54.1.	プロパティ.....	264
54.1.1.	ip プロパティ.....	264
54.1.2.	message プロパティ.....	264
54.1.3.	port プロパティ.....	264
54.1.4.	sender プロパティ.....	264
55.	Dorian.Gem.SecsIIClass Class.....	265
55.1.	プロパティ.....	266
55.1.1.	this[int]プロパティ.....	266
56.	Dorian.Gem.SentEventArgs Class.....	267
56.1.	プロパティ.....	268
56.1.1.	ip プロパティ.....	268
56.1.2.	message プロパティ.....	268
56.1.3.	port プロパティ.....	268
56.1.4.	sender プロパティ.....	268
56.1.5.	success プロパティ.....	268
57.	Dorian.Gem.ConfigDlg Class.....	269
57.1.	プロパティ.....	270
57.1.1.	components プロパティ.....	270
57.1.2.	gem プロパティ.....	270
57.2.	メソッド.....	271
57.2.1.	~Dorian.Gem.ConfigDlg メソッド.....	271
57.2.2.	Dorian.Gem.ConfigDlg メソッド.....	271
57.2.3.	InitializeComponent メソッド.....	271
58.	Dorian.Hsms Class.....	272
58.1.	プロパティ.....	273
58.1.1.	AutoReconnect プロパティ.....	273
58.1.2.	AutoSelectReq プロパティ.....	273
58.1.3.	AutoSessionID プロパティ.....	273
58.1.4.	AutoSystemBytes プロパティ.....	273
58.1.5.	Clients プロパティ.....	273
58.1.6.	Connect プロパティ.....	273
58.1.7.	FileName プロパティ.....	274
58.1.8.	HandleCtrlMsg プロパティ.....	274
58.1.9.	Heartbeat プロパティ.....	274
58.1.10.	IP プロパティ.....	274
58.1.11.	IPv4 プロパティ.....	275
58.1.12.	Json プロパティ.....	275
58.1.13.	MDLN プロパティ.....	275
58.1.14.	Port プロパティ.....	275
58.1.15.	Selected プロパティ.....	275

58.1.16.	Server プロパティ.....	276
58.1.17.	SessionID プロパティ.....	276
58.1.18.	SOFTREV プロパティ.....	276
58.1.19.	SystemBytes プロパティ.....	276
58.1.20.	T3 プロパティ.....	276
58.1.21.	T5 プロパティ.....	277
58.1.22.	T6 プロパティ.....	277
58.1.23.	T7 プロパティ.....	277
58.1.24.	T8 プロパティ.....	277
58.1.25.	Transactions プロパティ.....	277
58.2.	メソッド.....	279
58.2.1.	Config メソッド.....	279
58.2.2.	Load メソッド.....	279
58.2.3.	Save メソッド.....	279
58.2.4.	Send メソッド.....	279
58.2.5.	TimerProc メソッド.....	280
58.3.	イベント.....	281
58.3.1.	ConnectedEvent イベント.....	281
58.3.2.	DisconnectedEvent イベント.....	281
58.3.3.	ProblemEvent イベント.....	281
58.3.4.	ReceivedEvent イベント.....	281
58.3.5.	SentEvent イベント.....	282
59.	Dorian.Hsms.ClientsClass Class.....	283
59.1.	プロパティ.....	284
59.1.1.	Count プロパティ.....	284
59.1.2.	this[int]プロパティ.....	284
59.2.	メソッド.....	285
59.2.1.	find メソッド.....	285
60.	Dorian.Hsms.ClientsClass.Client Class.....	286
60.1.	プロパティ.....	287
60.1.1.	IP プロパティ.....	287
60.1.2.	Port プロパティ.....	287
60.1.3.	Selected プロパティ.....	287
60.2.	メソッド.....	288
60.2.1.	Disconnect メソッド.....	288
60.2.2.	Send メソッド.....	288
61.	Dorian.Hsms.ConnectedEventArgs Class.....	289
61.1.	プロパティ.....	290
61.1.1.	ip プロパティ.....	290
61.1.2.	port プロパティ.....	290
61.1.3.	sender プロパティ.....	290
62.	Dorian.Hsms.Data Class.....	291
62.1.	プロパティ.....	292
62.1.1.	autoSessionID プロパティ.....	292
62.1.2.	autoSystemBytes プロパティ.....	292
62.1.3.	handleCtrlMessage プロパティ.....	292
62.1.4.	sendSelectReq プロパティ.....	292
62.1.5.	sessionID プロパティ.....	292
62.1.6.	socket プロパティ.....	293
62.1.7.	t3 プロパティ.....	293
62.1.8.	t5 プロパティ.....	293
62.1.9.	t6 プロパティ.....	293
62.1.10.	t7 プロパティ.....	293
62.1.11.	t8 プロパティ.....	294
63.	Dorian.Hsms.Data.Socket Class.....	295
63.1.	プロパティ.....	296
63.1.1.	ip プロパティ.....	296
63.1.2.	ipv4 プロパティ.....	296

63.1.3.	port プロパティ.....	296
63.1.4.	server プロパティ.....	296
64.	Dorian.Hsms.DisconnectedEventArgs Class.....	297
64.1.	プロパティ.....	298
64.1.1.	ip プロパティ.....	298
64.1.2.	port プロパティ.....	298
64.1.3.	sender プロパティ.....	298
65.	Dorian.Hsms.ProblemEventArgs Class.....	299
65.1.	プロパティ.....	300
65.1.1.	code プロパティ.....	300
65.1.2.	ip プロパティ.....	300
65.1.3.	message プロパティ.....	300
65.1.4.	port プロパティ.....	300
65.1.5.	sender プロパティ.....	300
66.	Dorian.Hsms.ReceivedEventArgs Class.....	301
66.1.	プロパティ.....	302
66.1.1.	ip プロパティ.....	302
66.1.2.	message プロパティ.....	302
66.1.3.	port プロパティ.....	302
66.1.4.	sender プロパティ.....	302
67.	Dorian.Hsms.SentEventArgs Class.....	303
67.1.	プロパティ.....	304
67.1.1.	ip プロパティ.....	304
67.1.2.	message プロパティ.....	304
67.1.3.	port プロパティ.....	304
67.1.4.	sender プロパティ.....	304
67.1.5.	success プロパティ.....	304
68.	Dorian.Hsms.TransactionsClass Class.....	305
68.1.	プロパティ.....	306
68.1.1.	Count プロパティ.....	306
68.1.2.	this[int]プロパティ.....	306
69.	Dorian.Hsms.TransactionsClass.Transaction Class.....	307
69.1.	プロパティ.....	308
69.1.1.	Ctrl プロパティ.....	308
69.1.2.	Msg プロパティ.....	308
70.	Dorian.HsmsConfigDlg Class.....	309
70.1.	プロパティ.....	310
70.1.1.	components プロパティ.....	310
70.1.2.	hsms プロパティ.....	310
70.2.	メソッド.....	311
70.2.1.	~Dorian.HsmsConfigDlg メソッド.....	311
70.2.2.	Dorian.HsmsConfigDlg メソッド.....	311
70.2.3.	HsmsConfigDlg_Load メソッド.....	311
70.2.4.	InitializeComponent メソッド.....	311
71.	Dorian.Klarf12 Class.....	313
71.1.	プロパティ.....	314
71.1.1.	Count プロパティ.....	314
71.1.2.	this[int]プロパティ.....	314
71.1.3.	this[string]プロパティ.....	314
71.2.	メソッド.....	315
71.2.1.	Dorian.Klarf12 メソッド.....	315
71.2.2.	Find メソッド.....	315
71.2.3.	Parse メソッド.....	315
72.	Dorian.Klarf12.AlignmentImages Class.....	316
72.1.	プロパティ.....	317
72.1.1.	Count プロパティ.....	317
72.1.2.	this[int]プロパティ.....	317
72.2.	メソッド.....	318
72.2.1.	Dorian.Klarf12.AlignmentImages メソッド.....	318

73.	Dorian.Klarf12.AlignmentImages.DataItem Class	319
73.1.	プロパティ.....	320
73.1.1.	ID プロパティ	320
73.1.2.	NumOfImage プロパティ	320
73.1.3.	X プロパティ	320
73.1.4.	Y プロパティ	320
74.	Dorian.Klarf12.AlignmentImageTransforms Class	321
74.1.	プロパティ.....	322
74.1.1.	A11 プロパティ.....	322
74.1.2.	A12 プロパティ.....	322
74.1.3.	A21 プロパティ.....	322
74.1.4.	A22 プロパティ.....	322
74.1.5.	Count プロパティ.....	322
74.1.6.	MarkID プロパティ.....	322
74.2.	メソッド.....	323
74.2.1.	Dorian.Klarf12.AlignmentImageTransforms メソッド	323
75.	Dorian.Klarf12.AlignmentPoints Class.....	324
75.1.	プロパティ.....	325
75.1.1.	Count プロパティ.....	325
75.1.2.	this[int]プロパティ.....	325
75.2.	メソッド.....	326
75.2.1.	Dorian.Klarf12.AlignmentPoints メソッド	326
76.	Dorian.Klarf12.AlignmentPoints.DataItem Class.....	327
76.1.	プロパティ.....	328
76.1.1.	ID プロパティ	328
76.1.2.	X プロパティ	328
76.1.3.	Y プロパティ	328
77.	Dorian.Klarf12.AreaPerTest Class	329
77.1.	プロパティ.....	330
77.1.1.	Value プロパティ.....	330
77.2.	メソッド.....	331
77.2.1.	Dorian.Klarf12.AreaPerTest メソッド.....	331
78.	Dorian.Klarf12.ClassLookup Class	332
78.1.	プロパティ.....	333
78.1.1.	Count プロパティ.....	333
78.1.2.	this[int]プロパティ.....	333
78.2.	メソッド.....	334
78.2.1.	Dorian.Klarf12.ClassLookup メソッド	334
79.	Dorian.Klarf12.ClassLookup.DataItem Class.....	335
79.1.	プロパティ.....	336
79.1.1.	ID プロパティ	336
79.1.2.	Name プロパティ.....	336
80.	Dorian.Klarf12.ClusterClassificationList Class	337
80.1.	プロパティ.....	338
80.1.1.	Count プロパティ.....	338
80.1.2.	this[int]プロパティ.....	338
80.2.	メソッド.....	339
80.2.1.	Dorian.Klarf12.ClusterClassificationList メソッド.....	339
81.	Dorian.Klarf12.ClusterClassificationList.DataItem Class	340
81.1.	プロパティ.....	341
81.1.1.	Classification プロパティ.....	341
81.1.2.	ID プロパティ	341
82.	Dorian.Klarf12.CoordinatesMirrored Class	342
82.1.	プロパティ.....	343
82.1.1.	Value プロパティ.....	343
82.2.	メソッド.....	344
82.2.1.	Dorian.Klarf12.CoordinatesMirrored メソッド.....	344
83.	Dorian.Klarf12.DatabaseAlignmentMarks Class	345
83.1.	プロパティ.....	346

83.1.1.	Count プロパティ.....	346
83.1.2.	this[int]プロパティ.....	346
83.2.	メソッド.....	347
83.2.1.	Dorian.Klarf12.DatabaseAlignmentMarks メソッド.....	347
84.	Dorian.Klarf12.DatabaseAlignmentMarks.DataItem Class.....	348
84.1.	プロパティ.....	349
84.1.1.	ID プロパティ.....	349
84.1.2.	OriginX プロパティ.....	349
84.1.3.	OriginY プロパティ.....	349
84.1.4.	PointX プロパティ.....	349
84.1.5.	PointY プロパティ.....	349
85.	Dorian.Klarf12.DefectClusterSetup Class.....	350
85.1.	プロパティ.....	351
85.1.1.	Count プロパティ.....	351
85.1.2.	Value プロパティ.....	351
85.2.	メソッド.....	352
85.2.1.	Dorian.Klarf12.DefectClusterSetup メソッド.....	352
86.	Dorian.Klarf12.DefectClusterSpec Class.....	353
86.1.	プロパティ.....	354
86.1.1.	Count プロパティ.....	354
86.1.2.	Value プロパティ.....	354
86.2.	メソッド.....	355
86.2.1.	Dorian.Klarf12.DefectClusterSpec メソッド.....	355
87.	Dorian.Klarf12.DefectList Class.....	356
87.1.	プロパティ.....	357
87.1.1.	Count プロパティ.....	357
87.1.2.	this[int]プロパティ.....	357
87.2.	メソッド.....	358
87.2.1.	Dorian.Klarf12.DefectList メソッド.....	358
88.	Dorian.Klarf12.DefectList.DataItem Class.....	359
88.1.	プロパティ.....	360
88.1.1.	Count プロパティ.....	360
88.1.2.	this[string]プロパティ.....	360
88.1.3.	Value プロパティ.....	360
89.	Dorian.Klarf12.DefectRecordSpec Class.....	361
89.1.	プロパティ.....	362
89.1.1.	Count プロパティ.....	362
89.1.2.	this[string]プロパティ.....	362
89.1.3.	Value プロパティ.....	362
89.2.	メソッド.....	363
89.2.1.	Dorian.Klarf12.DefectRecordSpec メソッド.....	363
90.	Dorian.Klarf12.DeviceID Class.....	364
90.1.	プロパティ.....	365
90.1.1.	Text プロパティ.....	365
90.2.	メソッド.....	366
90.2.1.	Dorian.Klarf12.DeviceID メソッド.....	366
91.	Dorian.Klarf12.DieOrigin Class.....	367
91.1.	プロパティ.....	368
91.1.1.	X プロパティ.....	368
91.1.2.	Y プロパティ.....	368
91.2.	メソッド.....	369
91.2.1.	Dorian.Klarf12.DieOrigin メソッド.....	369
92.	Dorian.Klarf12.DiePitch Class.....	370
92.1.	プロパティ.....	371
92.1.1.	X プロパティ.....	371
92.1.2.	Y プロパティ.....	371
92.2.	メソッド.....	372
92.2.1.	Dorian.Klarf12.DiePitch メソッド.....	372
93.	Dorian.Klarf12.EndOfFile Class.....	373

93.1.	メソッド.....	374
93.1.1.	Dorian.Klarf12.EndOfFile メソッド.....	374
94.	Dorian.Klarf12.FileTimestamp Class.....	375
94.1.	プロパティ.....	376
94.1.1.	DateTime プロパティ.....	376
94.2.	メソッド.....	377
94.2.1.	Dorian.Klarf12.FileTimestamp メソッド.....	377
95.	Dorian.Klarf12.FileVersion Class.....	378
95.1.	プロパティ.....	379
95.1.1.	Major プロパティ.....	379
95.1.2.	Minor プロパティ.....	379
95.2.	メソッド.....	380
95.2.1.	Dorian.Klarf12.FileVersion メソッド.....	380
96.	Dorian.Klarf12.InspectedArea Class.....	381
96.1.	プロパティ.....	382
96.1.1.	Count プロパティ.....	382
96.1.2.	this[int]プロパティ.....	382
96.2.	メソッド.....	383
96.2.1.	Dorian.Klarf12.InspectedArea メソッド.....	383
97.	Dorian.Klarf12.InspectedArea.DataItem Class.....	384
97.1.	プロパティ.....	385
97.1.1.	OffsetX プロパティ.....	385
97.1.2.	OffsetY プロパティ.....	385
97.1.3.	PitchX プロパティ.....	385
97.1.4.	PitchY プロパティ.....	385
97.1.5.	RepeatX プロパティ.....	385
97.1.6.	RepeatY プロパティ.....	385
97.1.7.	SizeX プロパティ.....	385
97.1.8.	SizeY プロパティ.....	386
98.	Dorian.Klarf12.InspectedAreaOrigin Class.....	387
98.1.	プロパティ.....	388
98.1.1.	X プロパティ.....	388
98.1.2.	Y プロパティ.....	388
98.2.	メソッド.....	389
98.2.1.	Dorian.Klarf12.InspectedAreaOrigin メソッド.....	389
99.	Dorian.Klarf12.InspectionOrientation Class.....	390
99.1.	プロパティ.....	391
99.1.1.	Text プロパティ.....	391
99.2.	メソッド.....	392
99.2.1.	Dorian.Klarf12.InspectionOrientation メソッド.....	392
100.	Dorian.Klarf12.InspectionStationID Class.....	393
100.1.	プロパティ.....	394
100.1.1.	ID プロパティ.....	394
100.1.2.	Manufacturer プロパティ.....	394
100.1.3.	Model プロパティ.....	394
100.2.	メソッド.....	395
100.2.1.	Dorian.Klarf12.InspectionStationID メソッド.....	395
101.	Dorian.Klarf12.InspectionTest Class.....	396
101.1.	プロパティ.....	397
101.1.1.	ID プロパティ.....	397
101.2.	メソッド.....	398
101.2.1.	Dorian.Klarf12.InspectionTest メソッド.....	398
102.	Dorian.Klarf12.LotID Class.....	399
102.1.	プロパティ.....	400
102.1.1.	Text プロパティ.....	400
102.2.	メソッド.....	401
102.2.1.	Dorian.Klarf12.LotID メソッド.....	401
103.	Dorian.Klarf12.LotStatus Class.....	402
103.1.	プロパティ.....	403

103.1.1.	Failed プロパティ	403
103.1.2.	Passed プロパティ	403
103.1.3.	Total プロパティ	403
103.2.	メソッド	404
103.2.1.	Dorian.Klarf12.LotStatus メソッド	404
104.	Dorian.Klarf12.OrientationInstructions Class	405
104.1.	プロパティ	406
104.1.1.	Text プロパティ	406
104.2.	メソッド	407
104.2.1.	Dorian.Klarf12.OrientationInstructions メソッド	407
105.	Dorian.Klarf12.OrientationMarkLocation Class	408
105.1.	プロパティ	409
105.1.1.	Text プロパティ	409
105.2.	メソッド	410
105.2.1.	Dorian.Klarf12.OrientationMarkLocation メソッド	410
106.	Dorian.Klarf12.Record Class	411
106.1.	プロパティ	412
106.1.1.	Count プロパティ	412
106.1.2.	Name プロパティ	412
106.1.3.	this[int]プロパティ	412
107.	Dorian.Klarf12.RecordBase Class	413
107.1.	プロパティ	414
107.1.1.	Available プロパティ	414
108.	Dorian.Klarf12.RemovedDieList Class	415
108.1.	プロパティ	416
108.1.1.	Count プロパティ	416
108.1.2.	this[int]プロパティ	416
108.2.	メソッド	417
108.2.1.	Dorian.Klarf12.RemovedDieList メソッド	417
109.	Dorian.Klarf12.RemovedDieList.DataItem Class	418
109.1.	プロパティ	419
109.1.1.	X プロパティ	419
109.1.2.	Y プロパティ	419
110.	Dorian.Klarf12.ResultsID Class	420
110.1.	プロパティ	421
110.1.1.	Text プロパティ	421
110.2.	メソッド	422
110.2.1.	Dorian.Klarf12.ResultsID メソッド	422
111.	Dorian.Klarf12.ResultTimestamp Class	423
111.1.	プロパティ	424
111.1.1.	DateTime プロパティ	424
111.2.	メソッド	425
111.2.1.	Dorian.Klarf12.ResultTimestamp メソッド	425
112.	Dorian.Klarf12.SampleCenterLocation Class	426
112.1.	プロパティ	427
112.1.1.	X プロパティ	427
112.1.2.	Y プロパティ	427
112.2.	メソッド	428
112.2.1.	Dorian.Klarf12.SampleCenterLocation メソッド	428
113.	Dorian.Klarf12.SampleDieMap Class	429
113.1.	プロパティ	430
113.1.1.	Count プロパティ	430
113.1.2.	this[int]プロパティ	430
113.2.	メソッド	431
113.2.1.	Dorian.Klarf12.SampleDieMap メソッド	431
114.	Dorian.Klarf12.SampleDieMap.DataItem Class	432
114.1.	プロパティ	433
114.1.1.	X プロパティ	433
114.1.2.	Y プロパティ	433

115.	Dorian.Klarf12.SampleOrientationMarkType Class	434
115.1.	プロパティ	435
115.1.1.	Text プロパティ	435
115.2.	メソッド	436
115.2.1.	Dorian.Klarf12.SampleOrientationMarkType メソッド	436
116.	Dorian.Klarf12.SampleSize Class	437
116.1.	プロパティ	438
116.1.1.	Value プロパティ	438
116.2.	メソッド	439
116.2.1.	Dorian.Klarf12.SampleSize メソッド	439
117.	Dorian.Klarf12.SampleTestPlan Class	440
117.1.	プロパティ	441
117.1.1.	Count プロパティ	441
117.1.2.	this[int]プロパティ	441
117.2.	メソッド	442
117.2.1.	Dorian.Klarf12.SampleTestPlan メソッド	442
118.	Dorian.Klarf12.SampleTestPlan.DataItem Class	443
118.1.	プロパティ	444
118.1.1.	X プロパティ	444
118.1.2.	Y プロパティ	444
119.	Dorian.Klarf12.SampleTestReferencePlan Class	445
119.1.	プロパティ	446
119.1.1.	Count プロパティ	446
119.1.2.	this[int]プロパティ	446
119.2.	メソッド	447
119.2.1.	Dorian.Klarf12.SampleTestReferencePlan メソッド	447
120.	Dorian.Klarf12.SampleTestReferencePlan.DataItem Class	448
120.1.	プロパティ	449
120.1.1.	ComparedX プロパティ	449
120.1.2.	ComparedY プロパティ	449
120.1.3.	SampledX プロパティ	449
120.1.4.	SampledY プロパティ	449
121.	Dorian.Klarf12.SampleType Class	450
121.1.	プロパティ	451
121.1.1.	Text プロパティ	451
121.2.	メソッド	452
121.2.1.	Dorian.Klarf12.SampleType メソッド	452
122.	Dorian.Klarf12.SetupID Class	453
122.1.	プロパティ	454
122.1.1.	DateTime プロパティ	454
122.1.2.	ID プロパティ	454
122.2.	メソッド	455
122.2.1.	Dorian.Klarf12.SetupID メソッド	455
123.	Dorian.Klarf12.Slot Class	456
123.1.	プロパティ	457
123.1.1.	Value プロパティ	457
123.2.	メソッド	458
123.2.1.	Dorian.Klarf12.Slot メソッド	458
124.	Dorian.Klarf12.StepID Class	459
124.1.	プロパティ	460
124.1.1.	Text プロパティ	460
124.2.	メソッド	461
124.2.1.	Dorian.Klarf12.StepID メソッド	461
125.	Dorian.Klarf12.SummaryList Class	462
125.1.	プロパティ	463
125.1.1.	Count プロパティ	463
125.1.2.	this[int]プロパティ	463
125.2.	メソッド	464
125.2.1.	Dorian.Klarf12.SummaryList メソッド	464

126.	Dorian.Klarf12.SummaryList.DataItem Class	465
126.1.	プロパティ.....	466
126.1.1.	Count プロパティ.....	466
126.1.2.	Value プロパティ.....	466
127.	Dorian.Klarf12.SummarySpec Class.....	467
127.1.	プロパティ.....	468
127.1.1.	Count プロパティ.....	468
127.1.2.	Value プロパティ.....	468
127.2.	メソッド.....	469
127.2.1.	Dorian.Klarf12.SummarySpec メソッド.....	469
128.	Dorian.Klarf12.TestParametersList Class.....	470
128.1.	プロパティ.....	471
128.1.1.	Count プロパティ.....	471
128.1.2.	Value プロパティ.....	471
128.2.	メソッド.....	472
128.2.1.	Dorian.Klarf12.TestParametersList メソッド	472
129.	Dorian.Klarf12.TestParametersSpec Class	473
129.1.	プロパティ.....	474
129.1.1.	Count プロパティ.....	474
129.1.2.	Value プロパティ.....	474
129.2.	メソッド.....	475
129.2.1.	Dorian.Klarf12.TestParametersSpec メソッド.....	475
130.	Dorian.Klarf12.TiffFileName Class.....	476
130.1.	プロパティ.....	477
130.1.1.	Text プロパティ.....	477
130.2.	メソッド.....	478
130.2.1.	Dorian.Klarf12.TiffFileName メソッド.....	478
131.	Dorian.Klarf12.TiffSpec Class.....	479
131.1.	プロパティ.....	480
131.1.1.	Alignment プロパティ	480
131.1.2.	Defect プロパティ.....	480
131.1.3.	Version プロパティ.....	480
131.2.	メソッド.....	481
131.2.1.	Dorian.Klarf12.TiffSpec メソッド	481
132.	Dorian.Klarf12.WaferID Class.....	482
132.1.	プロパティ.....	483
132.1.1.	Text プロパティ.....	483
132.2.	メソッド.....	484
132.2.1.	Dorian.Klarf12.WaferID メソッド	484
133.	Dorian.Klarf12.WaferStatus Class	485
133.1.	プロパティ.....	486
133.1.1.	Text プロパティ.....	486
133.2.	メソッド.....	487
133.2.1.	Dorian.Klarf12.WaferStatus メソッド.....	487
134.	Dorian.Klarf18 Class.....	488
134.1.	プロパティ.....	489
134.1.1.	Json プロパティ	489
134.2.	メソッド.....	490
134.2.1.	Parse メソッド.....	490
135.	Dorian.Log Class	491
135.1.	プロパティ.....	492
135.1.1.	Count プロパティ.....	492
135.1.2.	Enable プロパティ.....	492
135.1.3.	FileName プロパティ	492
135.1.4.	FileSize プロパティ.....	492
135.1.5.	Json プロパティ	492
135.1.6.	JsonFileName プロパティ.....	493
135.1.7.	TimeStamp プロパティ	493
135.2.	メソッド.....	494

135.2.1.	Config メソッド.....	494
135.2.2.	Load メソッド.....	494
135.2.3.	Save メソッド.....	494
135.2.4.	Write メソッド.....	494
136.	Dorian.LogConfigDlg Class	496
136.1.	プロパティ.....	497
136.1.1.	components プロパティ	497
136.1.2.	log プロパティ.....	497
136.2.	メソッド.....	498
136.2.1.	~Dorian.LogConfigDlg メソッド	498
136.2.2.	Dorian.LogConfigDlg メソッド	498
136.2.3.	InitializeComponent メソッド.....	498
136.2.4.	LogConfigDlg_Load メソッド.....	498
137.	Dorian.Properties.Resources Class	500
137.1.	プロパティ.....	501
137.1.1.	Add プロパティ.....	501
137.1.2.	Banner プロパティ.....	501
137.1.3.	Culture プロパティ.....	501
137.1.4.	Gear プロパティ	501
137.1.5.	Play プロパティ.....	501
137.1.6.	Remove プロパティ.....	501
137.1.7.	ResourceManager プロパティ.....	501
137.1.8.	Stop プロパティ.....	502
138.	Dorian.Secsl Class	503
138.1.	プロパティ.....	504
138.1.1.	AutoSessionID プロパティ.....	504
138.1.2.	AutoSystemBytes プロパティ.....	504
138.1.3.	BaudRate プロパティ	504
138.1.4.	Connect プロパティ.....	504
138.1.5.	FileName プロパティ	504
138.1.6.	Heartbeat プロパティ.....	505
138.1.7.	Json プロパティ	505
138.1.8.	Master プロパティ	505
138.1.9.	MDLN プロパティ.....	505
138.1.10.	Port プロパティ	505
138.1.11.	Retry プロパティ	505
138.1.12.	SessionID プロパティ.....	506
138.1.13.	SOFTREV プロパティ	506
138.1.14.	SystemBytes プロパティ.....	506
138.1.15.	T1 プロパティ	506
138.1.16.	T2 プロパティ	506
138.1.17.	T3 プロパティ	507
138.1.18.	T4 プロパティ	507
138.2.	メソッド.....	508
138.2.1.	Config メソッド.....	508
138.2.2.	Load メソッド.....	508
138.2.3.	Save メソッド.....	508
138.2.4.	Send メソッド.....	508
138.2.5.	TimerProc メソッド	509
138.3.	イベント.....	510
138.3.1.	ProblemEvent イベント	510
138.3.2.	ReceivedEvent イベント.....	510
138.3.3.	ReceivedRawEvent イベント.....	510
138.3.4.	SentEvent イベント.....	510
138.3.5.	SentRawEvent イベント	511
139.	Dorian.Secsl.Data Class	512
139.1.	プロパティ.....	513
139.1.1.	autoSessionID プロパティ	513

139.1.2.	autoSystemBytes プロパティ	513
139.1.3.	baudRate プロパティ	513
139.1.4.	master プロパティ	513
139.1.5.	port プロパティ	513
139.1.6.	retry プロパティ	514
139.1.7.	sessionId プロパティ	514
139.1.8.	t1 プロパティ	514
139.1.9.	t2 プロパティ	514
139.1.10.	t3 プロパティ	514
139.1.11.	t4 プロパティ	515
140.	Dorian.Secsl.ProblemEventArgs Class	516
140.1.	プロパティ	517
140.1.1.	code プロパティ	517
140.1.2.	message プロパティ	517
140.1.3.	sender プロパティ	517
141.	Dorian.Secsl.ReceivedEventArgs Class	518
141.1.	プロパティ	519
141.1.1.	message プロパティ	519
141.1.2.	sender プロパティ	519
142.	Dorian.Secsl.ReceivedRawEventArgs Class	520
142.1.	プロパティ	521
142.1.1.	data プロパティ	521
142.1.2.	sender プロパティ	521
143.	Dorian.Secsl.SentEventArgs Class	522
143.1.	プロパティ	523
143.1.1.	message プロパティ	523
143.1.2.	sender プロパティ	523
143.1.3.	success プロパティ	523
144.	Dorian.Secsl.SentRawEventArgs Class	524
144.1.	プロパティ	525
144.1.1.	data プロパティ	525
144.1.2.	sender プロパティ	525
145.	Dorian.SecslConfigDlg Class	526
145.1.	プロパティ	527
145.1.1.	components プロパティ	527
145.1.2.	secs プロパティ	527
145.2.	メソッド	528
145.2.1.	~Dorian.SecslConfigDlg メソッド	528
145.2.2.	InitializeComponent メソッド	528
145.2.3.	SecslConfigDlg_Load メソッド	528
146.	Dorian.Secsl Class	530
146.1.	プロパティ	531
146.1.1.	BlockNumber プロパティ	531
146.1.2.	DeviceID プロパティ	531
146.1.3.	Ebit プロパティ	531
146.1.4.	Error プロパティ	531
146.1.5.	Function プロパティ	531
146.1.6.	Host プロパティ	532
146.1.7.	Hsms プロパティ	532
146.1.8.	Json プロパティ	532
146.1.9.	Msg プロパティ	532
146.1.10.	Node プロパティ	533
146.1.11.	NodeCount プロパティ	533
146.1.12.	NodeItem プロパティ	533
146.1.13.	NodeType プロパティ	533
146.1.14.	NodeValue プロパティ	533
146.1.15.	NodeValueHex プロパティ	534
146.1.16.	PType プロパティ	534

146.1.17.	Rbit プロパティ	534
146.1.18.	SessionID プロパティ	534
146.1.19.	Sml プロパティ	534
146.1.20.	SmlType プロパティ	535
146.1.21.	SourceID プロパティ	535
146.1.22.	Stream プロパティ	535
146.1.23.	SType プロパティ	535
146.1.24.	SuggestedReplyMsg プロパティ	536
146.1.25.	SystemBytes プロパティ	536
146.1.26.	this[string]プロパティ	536
146.1.27.	TransactionID プロパティ	536
146.1.28.	Verification プロパティ	536
146.1.29.	Wbit プロパティ	537
146.2.	メソッド	538
146.2.1.	Reply メソッド	538
146.2.2.	Reply メソッド	538
146.2.3.	Reset メソッド	538
147.	Dorian.SecsII.NodeItemClass Class	539
147.1.	プロパティ	540
147.1.1.	Count プロパティ	540
147.1.2.	Hex プロパティ	540
147.1.3.	Index プロパティ	540
147.1.4.	Text プロパティ	540
147.1.5.	this[int]プロパティ	540
147.1.6.	Type プロパティ	540
147.2.	メソッド	542
147.2.1.	Value メソッド	542
147.2.2.	ValueHex メソッド	542
148.	Dorian.SecsType Enum	543
149.	Dorian.SecsVerify Enum	544
150.	Dorian.Utility Class	545
150.1.	メソッド	546
150.1.1.	atof メソッド	546
150.1.2.	atofex メソッド	546
150.1.3.	atoi メソッド	546
150.1.4.	ExpireLicense メソッド	547
150.1.5.	GetLicense メソッド	547

2. Dorian.AboutDlg Class

Dorian.AboutDlg はライセンス情報を表示するヘルパークラスです。

プロパティ:

プロパティ	説明
components	

メソッド:

メソッド	説明
~Dorian.AboutDlg	デストラクタ
InitializeComponent	

2.1. プロパティ

2.1.1. components プロパティ

Required designer variable.

Visual C#:

components

2.2. メソッド

2.2.1. ~Dorian.AboutDlg メソッド

デストラクタ

Visual C#:

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

引数

変数名	説明
disposing	

戻り値

2.2.2. InitializeComponent メソッド

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

Visual C#:

```
InitializeComponent()
```

戻り値

3. Dorian.Comm Class

Dorian.Comm は RS232 シリアル通信のコンポーネントです。

プロパティ:

プロパティ	説明
BaudRate	ボーレート
ByteSize	1 バイトが何ビットか
Connect	通信ポートを開く・閉じる
Count	受信キューに何バイトのデータがあるか
Data	データを受信・送信します
FileName	JSON 設定ファイル名
Json	JSON 設定
Parity	パリティビット
Port	通信ポート番号
StopBit	ストップビット

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Read	データを読み込みます
Save	JSON 設定ファイルを書き込みます。
TimerProc	このメソッドを周期的に呼び出す必要があります。
Write	データを送信します

イベント:

イベント	説明
ReceivedEvent	データを受信した

3.1. プロパティ

3.1.1. BaudRate プロパティ

ボーレート

Visual C#:

```
int BaudRate
```

特記事項

ボーレートを取得または設定します。

3.1.2. ByteSize プロパティ

1バイトが何ビットか

Visual C#:

```
int ByteSize
```

特記事項

1バイトのデータが何ビットかを取得または設定します。

3.1.3. Connect プロパティ

通信ポートを開く・閉じる

Visual C#:

```
bool Connect
```

特記事項

シリアルポートの接続状態を取得または設定します。

3.1.4. Count プロパティ

受信キューに何バイトのデータがあるか

Visual C#:

```
int Count
```

特記事項

受信キューに何バイトのデータがあるかを取得します。

3.1.5. Data プロパティ

データを受信・送信します

Visual C#:

```
Byte[] Data
```

特記事項

データを受信または送信します。

3.1.6. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

3.1.7. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

3.1.8. Parity プロパティ

パリティビット

Visual C#:

```
int Parity
```

特記事項

パリティビットを取得または設定します。

3.1.9. Port プロパティ

通信ポート番号

Visual C#:

```
int Port
```

特記事項

通信ポート番号を取得または設定します。

3.1.10. StopBit プロパティ

ストップビット

Visual C#:

```
int StopBit
```

特記事項

ストップビットを取得または設定します。

3.2. メソッド

3.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

3.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

3.2.3. Read メソッド

データを読み込みます

Visual C#:

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

特記事項

受信キューからデータを読み込みます。

引数

変数名	説明
length	読み込むバイト数

戻り値

受信したデータ

3.2.4. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

3.2.5. TimerProc メソッド

このメソッドを周期的に呼び出す必要があります。

Visual C#:

```
public void TimerProc()
```

3.2.6. Write メソッド

データを送信します

Visual C#:

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

特記事項

データを送信します。

引数

変数名	説明
value	送信データ

戻り値

送信されたバイト数。

3.3. イベント

3.3.1. ReceivedEvent イベント

データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

4. Dorian.Comm.ReceivedEventArgs Class

Received イベント引数

プロパティ:

プロパティ	説明
data	データ
sender	このイベントの送信元

4.1. プロパティ

4.1.1. data プロパティ

データ

Visual C#:

```
Byte[] data
```

4.1.2. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

5. Dorian.CommConfigDlg Class

シリアルポート設定ダイアログボックス

プロパティ:

プロパティ	説明
comm	Comm オブジェクト
components	

メソッド:

メソッド	説明
~Dorian.CommConfigDlg	デストラクタ
CommConfigDlg_Load	Load イベント
Dorian.CommConfigDlg	コンストラクタ
InitializeComponent	

5.1. プロパティ

5.1.1. comm プロパティ

Comm オブジェクト

Visual C#:

Comm comm

5.1.2. components プロパティ

Required designer variable.

Visual C#:

components

5.2. メソッド

5.2.1. ~Dorian.CommConfigDlg メソッド

デストラクタ

Visual C#:

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

引数

変数名	説明
disposing	

戻り値

5.2.2. CommConfigDlg_Load メソッド

Load イベント

Visual C#:

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

引数

変数名	説明
sender	送信元
e	引数

戻り値

5.2.3. Dorian.CommConfigDlg メソッド

コンストラクタ

Visual C#:

```
Dorian.CommConfigDlg()
```

戻り値

5.2.4. InitializeComponent メソッド

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

Visual C#:

```
InitializeComponent()
```


戻り値

6. Dorian.Communication Class

Dorian.Communication コンポーネントは SEMI E37 (HSMS)または SEMI E4 (SECS-I)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。Dorian.SecsII コンポーネントと組み合わせて使います。

プロパティ:

プロパティ	説明
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
BaudRate	ボーレート
Clients	接続されているクライアントのリスト(パッシブエンティティのみ)
Connect	サーバに接続する、またはサーバを開始する。
FileNameHsms	HSMS 用の JSON 設定ファイル名
FileNameSecsI	SECS-I 用の JSON 設定ファイル名
HandleCtrlMsg	自動的に HSMS 制御メッセージを処理します
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
Hsms	HSMS または SECS-I
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Json	JSON 設定
Master	マスターかスレーブか
MDLN	装置モデル名
Port	TCP ポート番号
Retry	リトライ回数
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
SerialPort	シリアルポート番号
Server	サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)
SessionID	セッション ID(HSMS)
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T1	T1 タイマー
T2	T2 タイマー
T3	T3 タイマー
T4	T4 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー
Transactions	オープンなトランザクションのリスト

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージをサーバに送信する
TimerProc	このメソッドを周期的に呼び出す必要があります。

イベント:

イベント	説明
ConnectedEvent	クライアントから接続された
DisconnectedEvent	切断された
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した

ReceivedRawEvent	文字データを受信した
SentEvent	メッセージを送信した
SentRawEvent	文字データを送信した

6.1. プロパティ

6.1.1. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

6.1.2. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

6.1.3. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

6.1.4. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

6.1.5. BaudRate プロパティ

ボーレート

Visual C#:

```
int BaudRate
```

特記事項

ボーレートを取得または設定します。

6.1.6. Clients プロパティ

接続されているクライアントのリスト(パッシブエンティティのみ)

Visual C#:

```
ClientsClass Clients
```

6.1.7. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

Communication の接続状態を取得または設定します。もし Server プロパティが false(クライアント)の時に Connect プロパティが true にセットされた場合、Dorian.Communication コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされます。

もし Server プロパティが true(サーバ)の時に Connect プロパティが true にセットされた場合、Dorian.Communication コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。

Dorian.Communication コンポーネントが自動的に解除してくれるからです。Server プロパティが true(サーバ)の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

6.1.8. FileNameHsms プロパティ

HSMS 用の JSON 設定ファイル名

Visual C#:

```
string FileNameHsms
```

6.1.9. FileNameSecsI プロパティ

SECS-I 用の JSON 設定ファイル名

Visual C#:

```
string FileNameSecsI
```

6.1.10. HandleCtrlMsg プロパティ

自動的に HSMS 制御メッセージを処理します

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

自動的に HSMS 制御メッセージを処理します。

6.1.11. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

6.1.12. Hsms プロパティ

HSMS または SECS-I

Visual C#:

```
bool Hsms
```

特記事項

HSMS か SECS-I かを取得または設定します。デフォルトは HSMS です。

6.1.13. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは"0.0.0.0"に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は"127.0.0.1"もしくは"localhost"を使用してください。

IP アドレスの代わりにコンピュータ名を使用することもできます。

6.1.14. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

6.1.15. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

6.1.16. Master プロパティ

マスターかスレーブか

Visual C#:

```
bool Master
```

特記事項

マスターかスレーブかを取得または設定します。

6.1.17. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

6.1.18. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号(例えば http サーバなどは 80)があるため、一般的には 5000 以上の値を指定すべきです。

6.1.19. Retry プロパティ

リトライ回数

Visual C#:

```
int Retry
```

特記事項

リトライ回数を取得または設定します。

6.1.20. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

Communication 通信のセレクト状態を取得または設定します。

このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。

セレクト要求を受け取ると Dorian.Communication は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

6.1.21. SerialPort プロパティ

シリアルポート番号

Visual C#:

```
int SerialPort
```

特記事項

通信ポート番号を取得または設定します。

6.1.22. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

6.1.23. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

6.1.24. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

6.1.25. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```


特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

6.1.26. **T1 プロパティ**

T1 タイマー

Visual C#:

```
double T1
```

特記事項

SECS-I の T1 タイムアウトを秒単位で取得または設定します。デフォルトは 1 秒です。

6.1.27. **T2 プロパティ**

T2 タイマー

Visual C#:

```
double T2
```

特記事項

SECS-I の T2 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

6.1.28. **T3 プロパティ**

T3 タイマー

Visual C#:

```
double T3
```

特記事項

HSMS/SECS-I の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

6.1.29. **T4 プロパティ**

T4 タイマー

Visual C#:

```
double T4
```

特記事項

SECS-I の T4 タイムアウトを秒単位で取得または設定します。デフォルトは 60 秒です。

6.1.30. **T5 プロパティ**

T5 タイマー

Visual C#:

```
double T5
```

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

6.1.31. T6 プロパティ

T6 タイマー

Visual C#:

```
double T6
```

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

6.1.32. T7 プロパティ

T7 タイマー

Visual C#:

```
double T7
```

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

6.1.33. T8 プロパティ

T8 タイマー

Visual C#:

```
double T8
```

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

6.1.34. Transactions プロパティ

オープンなトランザクションのリスト

Visual C#:

```
TransactionsClass Transactions
```

6.2. メソッド

6.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

6.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

6.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

6.2.4. Send メソッド

メッセージをサーバに送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message

SECS-II メッセージ

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

6.2.5. TimerProc メソッド

このメソッドを周期的に呼び出す必要があります。

Visual C#:

```
public void TimerProc()
```

6.3. イベント

6.3.1. ConnectedEvent イベント

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

6.3.2. DisconnectedEvent イベント

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

6.3.3. ProblemEvent イベント

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

6.3.4. ReceivedEvent イベント

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

6.3.5. ReceivedRawEvent イベント

文字データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

6.3.6. SentEvent イベント

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

6.3.7. SentRawEvent イベント

文字データを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

7. Dorian.Forms.Comm Class

Dorian.Forms.Comm は RS232 シリアル通信のコンポーネントです。

プロパティ:

プロパティ	説明
BaudRate	ボーレート
ByteSize	1 バイトが何ビットか
Connect	通信ポートを開く・閉じる
Count	受信キューに何バイトのデータがあるか
Data	データを受信・送信します
FileName	JSON 設定ファイル名
Json	JSON 設定
Parity	パリティビット
Port	通信ポート番号
StopBit	ストップビット

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Read	データを読み込みます
Save	JSON 設定ファイルを書き込みます。
Write	データを送信します

イベント:

イベント	説明
ReceivedEvent	データを受信した

7.1. プロパティ

7.1.1. BaudRate プロパティ

ボーレート

Visual C#:

```
int BaudRate
```

特記事項

ボーレートを取得または設定します。

7.1.2. ByteSize プロパティ

1バイトが何ビットか

Visual C#:

```
int ByteSize
```

特記事項

1バイトのデータが何ビットかを取得または設定します。

7.1.3. Connect プロパティ

通信ポートを開く・閉じる

Visual C#:

```
bool Connect
```

特記事項

シリアルポートの接続状態を取得または設定します。

7.1.4. Count プロパティ

受信キューに何バイトのデータがあるか

Visual C#:

```
int Count
```

特記事項

受信キューに何バイトのデータがあるかを取得します。

7.1.5. Data プロパティ

データを受信・送信します

Visual C#:

```
Byte[] Data
```

特記事項

データを受信または送信します。

7.1.6. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

7.1.7. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

7.1.8. Parity プロパティ

パリティビット

Visual C#:

```
int Parity
```

特記事項

パリティビットを取得または設定します。

7.1.9. Port プロパティ

通信ポート番号

Visual C#:

```
int Port
```

特記事項

通信ポート番号を取得または設定します。

7.1.10. StopBit プロパティ

ストップビット

Visual C#:

```
int StopBit
```

特記事項

ストップビットを取得または設定します。

7.2. メソッド

7.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

7.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

7.2.3. Read メソッド

データを読み込みます

Visual C#:

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

特記事項

受信キューからデータを読み込みます。

引数

変数名	説明
length	読み込むバイト数

戻り値

受信したデータ

7.2.4. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

7.2.5. Write メソッド

データを送信します

Visual C#:

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

特記事項

データを送信します。

引数

変数名	説明
value	送信データ

戻り値

送信されたバイト数。

7.3. イベント

7.3.1. ReceivedEvent メソッド

データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

8. Dorian.Forms.Communication Class

Dorian.Forms.Communication コンポーネントは SEMI E37 (HSMS)または SEMI E4 (SECS-I)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。Dorian.SecslI コンポーネントと組み合わせて使います。

プロパティ:

プロパティ	説明
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
BaudRate	ボーレート
Clients	接続されているクライアントのリスト(パッシブエンティティのみ)
Connect	サーバに接続する、またはサーバを開始する。
FileNameHsms	HSMS 用の JSON 設定ファイル名
FileNameSecsl	SECS-I 用の JSON 設定ファイル名
HandleCtrlMsg	自動的に HSMS 制御メッセージを処理します
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
Hsms	HSMS または SECS-I
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Json	JSON 設定
Master	マスターかスレーブか
MDLN	装置モデル名
Port	TCP ポート番号
Retry	リトライ回数
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
SerialPort	シリアルポート番号
Server	サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)
SessionID	セッション ID(HSMS)
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T1	T1 タイマー
T2	T2 タイマー
T3	T3 タイマー
T4	T4 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー
Transactions	オープンなトランザクションのリスト

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージをサーバに送信する

イベント:

イベント	説明
ConnectedEvent	クライアントから接続された
DisconnectedEvent	切断された
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
ReceivedRawEvent	文字データを受信した

SentEvent	メッセージを送信した
SentRawEvent	文字データを送信した

8.1. プロパティ

8.1.1. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

8.1.2. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

8.1.3. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

8.1.4. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

8.1.5. BaudRate プロパティ

ボーレート

Visual C#:

```
int BaudRate
```

特記事項

ボーレートを取得または設定します。

8.1.6. Clients プロパティ

接続されているクライアントのリスト(パッシブエンティティのみ)

Visual C#:

```
ClientsClass Clients
```

8.1.7. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

Communication の接続状態を取得または設定します。もし Server プロパティが false(クライアント)の時に Connect プロパティが true にセットされた場合、Dorian.Communication コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされます。

もし Server プロパティが true(サーバ)の時に Connect プロパティが true にセットされた場合、Dorian.Communication コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。

Dorian.Communication コンポーネントが自動的に解除してくれるからです。Server プロパティが true(サーバ)の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

8.1.8. FileNameHsms プロパティ

HSMS 用の JSON 設定ファイル名

Visual C#:

```
string FileNameHsms
```

8.1.9. FileNameSecsI プロパティ

SECS-I 用の JSON 設定ファイル名

Visual C#:

```
string FileNameSecsI
```

8.1.10. HandleCtrlMsg プロパティ

自動的に HSMS 制御メッセージを処理します

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

自動的に HSMS 制御メッセージを処理します。

8.1.11. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

8.1.12. Hsms プロパティ

HSMS または SECS-I

Visual C#:

```
bool Hsms
```

特記事項

HSMS か SECS-I かを取得または設定します。デフォルトは HSMS です。

8.1.13. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは"0.0.0.0"に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は"127.0.0.1"もしくは"localhost"を使用してください。

IP アドレスの代わりにコンピュータ名を使用することもできます。

8.1.14. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

8.1.15. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

8.1.16. Master プロパティ

マスターかスレーブか

Visual C#:

```
bool Master
```

特記事項

マスターかスレーブかを取得または設定します。

8.1.17. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

8.1.18. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号(例えば http サーバなどは 80)があるため、一般的には 5000 以上の値を指定すべきです。

8.1.19. Retry プロパティ

リトライ回数

Visual C#:

```
int Retry
```

特記事項

リトライ回数を取得または設定します。

8.1.20. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

Communication 通信のセレクト状態を取得または設定します。

このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。

セレクト要求を受け取ると Dorian.Communication は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

8.1.21. SerialPort プロパティ

シリアルポート番号

Visual C#:

```
int SerialPort
```

特記事項

通信ポート番号を取得または設定します。

8.1.22. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

8.1.23. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

8.1.24. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

8.1.25. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

8.1.26. **T1 プロパティ**

T1 タイマー

Visual C#:

```
double T1
```

特記事項

SECS-I の T1 タイムアウトを秒単位で取得または設定します。デフォルトは 1 秒です。

8.1.27. **T2 プロパティ**

T2 タイマー

Visual C#:

```
double T2
```

特記事項

SECS-I の T2 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

8.1.28. **T3 プロパティ**

T3 タイマー

Visual C#:

```
double T3
```

特記事項

HSMS/SECS-I の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

8.1.29. **T4 プロパティ**

T4 タイマー

Visual C#:

```
double T4
```

特記事項

SECS-I の T4 タイムアウトを秒単位で取得または設定します。デフォルトは 60 秒です。

8.1.30. **T5 プロパティ**

T5 タイマー

Visual C#:

```
double T5
```

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

8.1.31. T6 プロパティ

T6 タイマー

Visual C#:

```
double T6
```

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

8.1.32. T7 プロパティ

T7 タイマー

Visual C#:

```
double T7
```

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

8.1.33. T8 プロパティ

T8 タイマー

Visual C#:

```
double T8
```

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

8.1.34. Transactions プロパティ

オープンなトランザクションのリスト

Visual C#:

```
TransactionsClass Transactions
```

8.2. メソッド

8.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

8.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

8.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

8.2.4. Send メソッド

メッセージをサーバに送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message	SECS-II メッセージ
---------	---------------

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

8.3. イベント

8.3.1. ConnectedEvent イベント

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

8.3.2. DisconnectedEvent イベント

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

8.3.3. ProblemEvent イベント

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

8.3.4. ReceivedEvent イベント

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

8.3.5. ReceivedRawEvent イベント

文字データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

8.3.6. SentEvent イベント

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

8.3.7. SentRawEvent イベント

文字データを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

9. Dorian.Forms.CommunicationView Class

Dorian.Forms.CommunicationView コンポーネントは SEMI E37 (HSMS)または SEMI E4 (SECS-I)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。Dorian.SecsII コンポーネントと組み合わせて使います。

プロパティ:

プロパティ	説明
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
BaudRate	ボーレート
Clients	接続されているクライアントのリスト(パッシブエンティティのみ)
Connect	サーバに接続する、またはサーバを開始する。
FileNameHsms	HSMS 用の JSON 設定ファイル名
FileNameSecsI	SECS-I 用の JSON 設定ファイル名
HandleCtrlMsg	自動的に HSMS 制御メッセージを処理します
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
Hsms	HSMS または SECS-I
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Json	JSON 設定
Master	マスターかスレーブか
MDLN	装置モデル名
Port	TCP ポート番号
Retry	リトライ回数
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
SerialPort	シリアルポート番号
Server	サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)
SessionID	セッション ID(HSMS)
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T1	T1 タイマー
T2	T2 タイマー
T3	T3 タイマー
T4	T4 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー
Transactions	オープンなトランザクションのリスト

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージをサーバに送信する

イベント:

イベント	説明
ConnectedEvent	クライアントから接続された
DisconnectedEvent	切断された
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
ReceivedRawEvent	文字データを受信した

SentEvent	メッセージを送信した
SentRawEvent	文字データを送信した

9.1. プロパティ

9.1.1. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

9.1.2. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

9.1.3. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

9.1.4. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

9.1.5. BaudRate プロパティ

ボーレート

Visual C#:

```
int BaudRate
```

特記事項

ボーレートを取得または設定します。

9.1.6. Clients プロパティ

接続されているクライアントのリスト(パッシブエンティティのみ)

Visual C#:

```
ClientsClass Clients
```

9.1.7. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

Communication の接続状態を取得または設定します。もし Server プロパティが false(クライアント)の時に Connect プロパティが true にセットされた場合、Dorian.Communication コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされます。

もし Server プロパティが true(サーバ)の時に Connect プロパティが true にセットされた場合、Dorian.Communication コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。

Dorian.Communication コンポーネントが自動的に解除してくれるからです。Server プロパティが true(サーバ)の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

9.1.8. FileNameHsms プロパティ

HSMS 用の JSON 設定ファイル名

Visual C#:

```
string FileNameHsms
```

9.1.9. FileNameSecsI プロパティ

SECS-I 用の JSON 設定ファイル名

Visual C#:

```
string FileNameSecsI
```

9.1.10. HandleCtrlMsg プロパティ

自動的に HSMS 制御メッセージを処理します

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

自動的に HSMS 制御メッセージを処理します。

9.1.11. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

9.1.12. Hsms プロパティ

HSMS または SECS-I

Visual C#:

```
bool Hsms
```

特記事項

HSMS か SECS-I かを取得または設定します。デフォルトは HSMS です。

9.1.13. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは"0.0.0.0"に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は"127.0.0.1"もしくは"localhost"を使用してください。

IP アドレスの代わりにコンピュータ名を使用することもできます。

9.1.14. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

9.1.15. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

9.1.16. Master プロパティ

マスターかスレーブか

Visual C#:

```
bool Master
```

特記事項

マスターかスレーブかを取得または設定します。

9.1.17. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

9.1.18. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号(例えば http サーバなどは 80)があるため、一般的には 5000 以上の値を指定すべきです。

9.1.19. Retry プロパティ

リトライ回数

Visual C#:

```
int Retry
```

特記事項

リトライ回数を取得または設定します。

9.1.20. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

Communication 通信のセレクト状態を取得または設定します。

このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。

セレクト要求を受け取ると Dorian.Communication は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

9.1.21. SerialPort プロパティ

シリアルポート番号

Visual C#:

```
int SerialPort
```

特記事項

通信ポート番号を取得または設定します。

9.1.22. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

9.1.23. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

9.1.24. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

9.1.25. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```


特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

9.1.26. **T1 プロパティ**

T1 タイマー

Visual C#:

```
double T1
```

特記事項

SECS-I の T1 タイムアウトを秒単位で取得または設定します。デフォルトは 1 秒です。

9.1.27. **T2 プロパティ**

T2 タイマー

Visual C#:

```
double T2
```

特記事項

SECS-I の T2 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

9.1.28. **T3 プロパティ**

T3 タイマー

Visual C#:

```
double T3
```

特記事項

HSMS/SECS-I の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

9.1.29. **T4 プロパティ**

T4 タイマー

Visual C#:

```
double T4
```

特記事項

SECS-I の T4 タイムアウトを秒単位で取得または設定します。デフォルトは 60 秒です。

9.1.30. **T5 プロパティ**

T5 タイマー

Visual C#:

```
double T5
```

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

9.1.31. T6 プロパティ

T6 タイマー

Visual C#:

```
double T6
```

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

9.1.32. T7 プロパティ

T7 タイマー

Visual C#:

```
double T7
```

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

9.1.33. T8 プロパティ

T8 タイマー

Visual C#:

```
double T8
```

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

9.1.34. Transactions プロパティ

オープンなトランザクションのリスト

Visual C#:

```
TransactionsClass Transactions
```

9.2. メソッド

9.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

9.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

9.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

9.2.4. Send メソッド

メッセージをサーバに送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message

SECS-II メッセージ

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

9.3. イベント

9.3.1. ConnectedEvent イベント

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

9.3.2. DisconnectedEvent イベント

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

9.3.3. ProblemEvent イベント

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

9.3.4. ReceivedEvent イベント

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

9.3.5. ReceivedRawEvent イベント

文字データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

9.3.6. SentEvent イベント

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

9.3.7. SentRawEvent イベント

文字データを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

10. Dorian.Forms.Gem Class

Dorian.Forms.Gem コンポーネントは SEMI E30 (GEM)の機能を作成するための開発支援製品です。装置側ソフトの開発で使えます。

プロパティ:

プロパティ	説明
AlarmModel	GEM アラームモデルオブジェクト
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
CallDefProc	Received イベントの後に DefProc() を呼ぶ
Clients	接続されているクライアントのリスト (パッシブエンティティのみ)
CommModel	GEM 通信モデルオブジェクト
Connect	サーバに接続する、またはサーバを開始する。
CtrlModel	GEM コントロールモデルオブジェクト
EventModel	GEM イベントモデルオブジェクト
FileName	JSON 設定ファイル名
HandleCtrlMsg	IP v4 または IP v6
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Json	JSON 設定
Msg	SECS-II オブジェクト
Port	TCP ポート番号
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
Server	サーバ (パッシブエンティティ)、またはクライアント (アクティブエンティティ)
SessionID	セッション ID (HSMS)
SystemBytes	システムバイト
T3	T3 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー

メソッド:

メソッド	説明
Config	設定画面を表示します
InvokeAlarm	アラームを送信します。
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージを送信する

イベント:

イベント	説明
CommStateChangedEvent	通信状態が変化した
ConnectedEvent	クライアントから接続された
CtrlStateChangedEvent	コントロール状態が変化した
DisconnectedEvent	切断された
InvokeEvent	イベントを送信します。
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
SentEvent	メッセージを送信した

10.1. プロパティ

10.1.1. AlarmModel プロパティ

GEM アラームモデルオブジェクト

Visual C#:

```
AlarmModelClass AlarmModel
```

10.1.2. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

10.1.3. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

10.1.4. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

10.1.5. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

10.1.6. CallDefProc プロパティ

Received イベントの後で DefProc() を呼ぶ

Visual C#:

```
bool CallDefProc
```

10.1.7. Clients プロパティ

接続されているクライアントのリスト(パッシブエンティティのみ)

Visual C#:

```
ClientsClass Clients
```

10.1.8. CommModel プロパティ

GEM 通信モデルオブジェクト

Visual C#:

```
CommModelClass CommModel
```

10.1.9. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

HSMS の接続状態を取得または設定します。もし Server プロパティが false (クライアント) の時に Connect プロパティが true にセットされた場合、Dorian.Forms.Hsms コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされません。

もし Server プロパティが true (サーバ) の時に Connect プロパティが true にセットされた場合、Dorian.Forms.Hsms コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。

Dorian.Forms.Hsms コンポーネントが自動的に解除してくれるからです。Server プロパティが true (サーバ) の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

10.1.10. CtrlModel プロパティ

GEM コントロールモデルオブジェクト

Visual C#:

```
CtrlModelClass CtrlModel
```

10.1.11. EventModel プロパティ

GEM イベントモデルオブジェクト

Visual C#:

```
EventManagerClass EventMode]
```

10.1.12. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

10.1.13. HandleCtrlMsg プロパティ

IP v4 または IP v6

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

10.1.14. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスシブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは "0.0.0.0" に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は "127.0.0.1" もしくは "localhost" を使用してください。

IP アドレスの代わりにコンピュータ名を使用することもできます。

10.1.15. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

10.1.16. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

10.1.17. Msg プロパティ

SECS-II オブジェクト

Visual C#:

```
SecsIIClass Msg
```

10.1.18. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号 (例えば http サーバなどは 80) があるため、一般的には 5000 以上の値を指定すべきです。

10.1.19. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

HSMS 通信のセレクト状態を取得または設定します。

このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。

セレクト要求を受け取ると Dorian.Forms.Hsms は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

10.1.20. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

10.1.21. SessionID プロパティ

セッション ID (HSMS)

Visual C#:

UInt16 SessionID

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

10.1.22. SystemBytes プロパティ

システムバイト

Visual C#:

UInt32 SystemBytes

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

10.1.23. T3 プロパティ

T3 タイマー

Visual C#:

double T3

特記事項

HSMS の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

10.1.24. T5 プロパティ

T5 タイマー

Visual C#:

double T5

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

10.1.25. T6 プロパティ

T6 タイマー

Visual C#:

double T6

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

10.1.26. T7 プロパティ

T7 タイマー

Visual C#:

double T7

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

10.1.27. T8 プロパティ

T8 タイマー

Visual C#:

double T8

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

10.2. メソッド

10.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

10.2.2. InvokeAlarm メソッド

アラームを送信します。

Visual C#:

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

引数

変数名	説明
alid	ALID
occur	アラーム発生・解除

戻り値

成功した場合は true が、失敗した場合は false が返ります。

10.2.3. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

10.2.4. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

10.2.5. Send メソッド

メッセージを送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
message	SECS-II メッセージ

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

10.3. イベント

10.3.1. CommStateChangedEvent メソッド

通信状態が変化した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

10.3.2. ConnectedEvent メソッド

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

10.3.3. CtrlStateChangedEvent メソッド

コントロール状態が変化した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

10.3.4. DisconnectedEvent メソッド

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

10.3.5. InvokeEvent メソッド

イベントを送信します。

Visual C#:

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

引数

変数名	説明
ceid	CEID

戻り値

成功した場合は true が、失敗した場合は false が返ります。

10.3.6. ProblemEvent メソッド

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

10.3.7. ReceivedEvent メソッド

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

10.3.8. SentEvent メソッド

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

11. Dorian.Forms.GemView Class

Dorian.Forms.GemView コンポーネントは SEMI E30 (GEM)の機能を作成するための開発支援製品です。装置側ソフトの開発で使えます。

プロパティ:

プロパティ	説明
AlarmModel	GEM アラームモデルオブジェクト
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
CallDefProc	Received イベントの後に DefProc() を呼ぶ
Clients	接続されているクライアントのリスト (パッシブエンティティのみ)
CommModel	GEM 通信モデルオブジェクト
Connect	サーバに接続する、またはサーバを開始する。
CtrlModel	GEM コントロールモデルオブジェクト
EventModel	GEM イベントモデルオブジェクト
FileName	JSON 設定ファイル名
HandleCtrlMsg	IP v4 または IP v6
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Json	JSON 設定
Msg	SECS-II オブジェクト
Port	TCP ポート番号
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
Server	サーバ (パッシブエンティティ)、またはクライアント (アクティブエンティティ)
SessionID	セッション ID (HSMS)
SystemBytes	システムバイト
T3	T3 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー

メソッド:

メソッド	説明
Config	設定画面を表示します
InvokeAlarm	アラームを送信します。
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージを送信する

イベント:

イベント	説明
CommStateChangedEvent	通信状態が変化した
ConnectedEvent	クライアントから接続された
CtrlStateChangedEvent	コントロール状態が変化した
DisconnectedEvent	切断された
InvokeEvent	イベントを送信します。
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
SentEvent	メッセージを送信した

11.1. プロパティ

11.1.1. AlarmModel プロパティ

GEM アラームモデルオブジェクト

Visual C#:

```
AlarmModelClass AlarmModel
```

11.1.2. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

11.1.3. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

11.1.4. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

11.1.5. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

11.1.6. CallDefProc プロパティ

Received イベントの後に DefProc() を呼ぶ

Visual C#:

```
bool CallDefProc
```

11.1.7. Clients プロパティ

接続されているクライアントのリスト(パッシブエンティティのみ)

Visual C#:

```
ClientsClass Clients
```

11.1.8. CommModel プロパティ

GEM 通信モデルオブジェクト

Visual C#:

```
CommModelClass CommModel
```

11.1.9. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

HSMS の接続状態を取得または設定します。もし Server プロパティが false (クライアント) の時に Connect プロパティが true にセットされた場合、Dorian.Forms.Hsms コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされます。

もし Server プロパティが true (サーバ) の時に Connect プロパティが true にセットされた場合、Dorian.Forms.Hsms コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。

Dorian.Forms.Hsms コンポーネントが自動的に解除してくれるからです。Server プロパティが true (サーバ) の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

11.1.10. CtrlModel プロパティ

GEM コントロールモデルオブジェクト

Visual C#:

```
CtrlModelClass CtrlModel
```

11.1.11. EventModel プロパティ

GEM イベントモデルオブジェクト

Visual C#:

```
EventManagerClass EventMode]
```

11.1.12. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

11.1.13. HandleCtrlMsg プロパティ

IP v4 または IP v6

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

11.1.14. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスシブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは "0.0.0.0" に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は "127.0.0.1" もしくは "localhost" を使用してください。

IP アドレスの代わりにコンピュータ名を使用することもできます。

11.1.15. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

11.1.16. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

11.1.17. Msg プロパティ

SECS-II オブジェクト

Visual C#:

```
SecsIIClass Msg
```

11.1.18. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号 (例えば http サーバなどは 80) があるため、一般的には 5000 以上の値を指定すべきです。

11.1.19. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

HSMS 通信のセレクト状態を取得または設定します。

このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。

セレクト要求を受け取ると Dorian.Forms.Hsms は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

11.1.20. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

11.1.21. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

UInt16 SessionID

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

11.1.22. SystemBytes プロパティ

システムバイト

Visual C#:

UInt32 SystemBytes

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

11.1.23. T3 プロパティ

T3 タイマー

Visual C#:

double T3

特記事項

HSMS の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

11.1.24. T5 プロパティ

T5 タイマー

Visual C#:

double T5

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

11.1.25. T6 プロパティ

T6 タイマー

Visual C#:

double T6

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

11.1.26. T7 プロパティ

T7 タイマー

Visual C#:

double T7

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

11.1.27. T8 プロパティ

T8 タイマー

Visual C#:

double T8

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

11.2. メソッド

11.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

11.2.2. InvokeAlarm メソッド

アラームを送信します。

Visual C#:

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

引数

変数名	説明
alid	ALID
occur	アラーム発生・解除

戻り値

成功した場合は true が、失敗した場合は false が返ります。

11.2.3. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

11.2.4. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

11.2.5. Send メソッド

メッセージを送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
message	SECS-II メッセージ

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

11.3. イベント

11.3.1. CommStateChangedEvent メソッド

通信状態が変化した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

11.3.2. ConnectedEvent メソッド

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

11.3.3. CtrlStateChangedEvent メソッド

コントロール状態が変化した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

11.3.4. DisconnectedEvent メソッド

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

11.3.5. InvokeEvent メソッド

イベントを送信します。

Visual C#:

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

引数

変数名	説明
ceid	CEID

戻り値

成功した場合は true が、失敗した場合は false が返ります。

11.3.6. ProblemEvent メソッド

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

11.3.7. ReceivedEvent メソッド

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

11.3.8. SentEvent メソッド

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

12. Dorian.Forms.Hsms Class

Dorian.Forms.Hsms コンポーネントは SEMI E37 (HSMS)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。Dorian.SecsII コンポーネントと組み合わせて使います。

プロパティ:

プロパティ	説明
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
Clients	接続されているクライアントのリスト(パッシブエンティティのみ)
Connect	サーバに接続する、またはサーバを開始する。
FileName	JSON 設定ファイル名
HandleCtrlMsg	自動的に HSMS 制御メッセージを処理します
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Json	JSON 設定
MDLN	装置モデル名
Port	TCP ポート番号
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
Server	サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)
SessionID	セッション ID(HSMS)
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T3	T3 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー
Transactions	オープンなトランザクションのリスト

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージをサーバに送信する

イベント:

イベント	説明
ConnectedEvent	クライアントから接続された
DisconnectedEvent	切断された
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
SentEvent	メッセージを送信した

12.1. プロパティ

12.1.1. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

12.1.2. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

12.1.3. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

12.1.4. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

12.1.5. Clients プロパティ

接続されているクライアントのリスト (パッシブエンティティのみ)

Visual C#:

```
ClientsClass Clients
```

12.1.6. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

HSMS の接続状態を取得または設定します。もし Server プロパティが false (クライアント) の時に Connect プロパティが true にセットされた場合、Dorian.Forms.Hsms コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされます。

もし Server プロパティが true (サーバ) の時に Connect プロパティが true にセットされた場合、Dorian.Forms.Hsms コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。

Dorian.Forms.Hsms コンポーネントが自動的に解除してくれるからです。Server プロパティが true (サーバ) の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

12.1.7. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

12.1.8. HandleCtrlMsg プロパティ

自動的に HSMS 制御メッセージを処理します

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

自動的に HSMS 制御メッセージを処理します。

12.1.9. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

12.1.10. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスシブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは"0.0.0.0"に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は"127.0.0.1"もしくは"localhost"を使用してください。IP アドレスの代わりにコンピュータ名を使用することもできます。

12.1.11. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

12.1.12. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

12.1.13. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

12.1.14. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号(例えば http サーバなどは 80)があるため、一般的には 5000 以上の値を指定すべきです。

12.1.15. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

HSMS 通信のセレクト状態を取得または設定します。
このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。
セレクト要求を受け取ると Dorian.Forms.Hsms は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

12.1.16. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

12.1.17. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

12.1.18. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

12.1.19. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

12.1.20. T3 プロパティ

T3 タイマー

Visual C#:

```
double T3
```

特記事項

HSMS の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

12.1.21. T5 プロパティ

T5 タイマー

Visual C#:

```
double T5
```

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

12.1.22. T6 プロパティ

T6 タイマー

Visual C#:

```
double T6
```

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

12.1.23. T7 プロパティ

T7 タイマー

Visual C#:

```
double T7
```

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

12.1.24. T8 プロパティ

T8 タイマー

Visual C#:

```
double T8
```

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

12.1.25. Transactions プロパティ

オープンなトランザクションのリスト

Visual C#:

TransactionsClass Transactions

12.2. メソッド

12.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

12.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

12.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

12.2.4. Send メソッド

メッセージをサーバに送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message	SECS-II メッセージ
---------	---------------

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

12.3. イベント

12.3.1. ConnectedEvent メソッド

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

12.3.2. DisconnectedEvent メソッド

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

12.3.3. ProblemEvent メソッド

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

12.3.4. ReceivedEvent メソッド

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

12.3.5. SentEvent メソッド

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

13. Dorian.Forms.HsmsView Class

Dorian.Forms.HsmsView コンポーネントは SEMI E37 (HSMS)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。Dorian.SecsII コンポーネントと組み合わせて使います。

プロパティ:

プロパティ	説明
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
Clients	接続されているクライアントのリスト(パッシブエンティティのみ)
Connect	サーバに接続する、またはサーバを開始する。
FileName	JSON 設定ファイル名
HandleCtrlMsg	自動的に HSMS 制御メッセージを処理します
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Json	JSON 設定
MDLN	装置モデル名
Port	TCP ポート番号
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
Server	サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)
SessionID	セッション ID(HSMS)
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T3	T3 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー
Transactions	オープンなトランザクションのリスト

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージをサーバに送信する

イベント:

イベント	説明
ConnectedEvent	クライアントから接続された
DisconnectedEvent	切断された
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
SentEvent	メッセージを送信した

13.1. プロパティ

13.1.1. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

13.1.2. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

13.1.3. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

13.1.4. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

13.1.5. Clients プロパティ

接続されているクライアントのリスト (パッシブエンティティのみ)

Visual C#:

```
ClientsClass Clients
```

13.1.6. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

HSMS の接続状態を取得または設定します。もし Server プロパティが false (クライアント) の時に Connect プロパティが true にセットされた場合、Dorian.Forms.Hsms コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされます。

もし Server プロパティが true (サーバ) の時に Connect プロパティが true にセットされた場合、Dorian.Forms.Hsms コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。

Dorian.Forms.Hsms コンポーネントが自動的に解除してくれるからです。Server プロパティが true (サーバ) の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

13.1.7. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

13.1.8. HandleCtrlMsg プロパティ

自動的に HSMS 制御メッセージを処理します

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

自動的に HSMS 制御メッセージを処理します。

13.1.9. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

13.1.10. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスシブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは"0.0.0.0"に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は"127.0.0.1"もしくは"localhost"を使用してください。IP アドレスの代わりにコンピュータ名を使用することもできます。

13.1.11. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

13.1.12. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

13.1.13. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

13.1.14. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号(例えば http サーバなどは 80)があるため、一般的には 5000 以上の値を指定すべきです。

13.1.15. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

HSMS 通信のセレクト状態を取得または設定します。
 このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。
 セレクト要求を受け取ると Dorian.Forms.Hsms は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

13.1.16. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

13.1.17. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

13.1.18. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

13.1.19. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

13.1.20. T3 プロパティ

T3 タイマー

Visual C#:

```
double T3
```

特記事項

HSMS の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

13.1.21. T5 プロパティ

T5 タイマー

Visual C#:

```
double T5
```

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

13.1.22. T6 プロパティ

T6 タイマー

Visual C#:

```
double T6
```

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

13.1.23. T7 プロパティ

T7 タイマー

Visual C#:

```
double T7
```

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

13.1.24. T8 プロパティ

T8 タイマー

Visual C#:

```
double T8
```

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

13.1.25. Transactions プロパティ

オープンなトランザクションのリスト

Visual C#:

TransactionsClass Transactions

13.2. メソッド

13.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

13.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

13.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

13.2.4. Send メソッド

メッセージをサーバに送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message

SECS-II メッセージ

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

13.3. イベント

13.3.1. ConnectedEvent メソッド

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

13.3.2. DisconnectedEvent メソッド

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

13.3.3. ProblemEvent メソッド

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

13.3.4. ReceivedEvent メソッド

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

13.3.5. SentEvent メソッド

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

14. Dorian.Forms.Klarf12 Class

Dorian.Forms.Klarf12 コンポーネント は、KLARF バージョン 1.2 の機能を作成するための開発支援製品です。

プロパティ:

プロパティ	説明
Count	レコード数
this[int]	レコードにアクセスするためのシンタックスシュガー
this[string]	レコードにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Find	レコードを検索します
Parse	KLARF ファイルを解析します

14.1. プロパティ

14.1.1. Count プロパティ

レコード数

Visual C#:

```
int Count
```

特記事項

KLARF ファイルに含まれているレコード数を取得します。

14.1.2. this[int] プロパティ

レコードにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

14.1.3. this[string] プロパティ

レコードにアクセスするためのシンタックスシュガー

Visual C#:

```
this[string]
```

14.2. メソッド

14.2.1. Find メソッド

レコードを検索します

Visual C#:

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

引数

変数名	説明
record	レコード名
from	検索を開始するインデックス
forward	true = 正方向検索、false = 逆方向検索

戻り値

14.2.2. Parse メソッド

KLARF ファイルを解析します

Visual C#:

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

特記事項

KLARF ファイルの内容を解析します。

引数

変数名	説明
klarfText	KLARF ファイルの内容

戻り値

成功した場合は true を、失敗した場合は false を返します。

15. Dorian.Forms.Klarf12Map Class

Dorian.Forms.Klarf12Map コンポーネント は、KLARF バージョン 1.2 の機能を作成するための開発支援製品です。欠陥マップを表示します。

プロパティ:

プロパティ	説明
Index	レコードのインデックス
klarf	Dorian.KLARF12 オブジェクト
XY	REL が利用可能でも、XとYを使用。

15.1. プロパティ

15.1.1. Index プロパティ

レコードのインデックス

Visual C#:

```
int Index
```

15.1.2. klarf プロパティ

Dorian.KLARF12 オブジェクト

Visual C#:

```
Klarf12 klarf
```

15.1.3. XY プロパティ

REL が利用可能でも、XとYを使用。

Visual C#:

```
bool XY
```

16. Dorian.Forms.Klarf12Table Class

Dorian.Forms.Klarf12TableForm コンポーネント は、KLARF バージョン 1.2 の機能を作成するための開発支援製品です。

プロパティ:

プロパティ	説明
Index	レコードのインデックス
klarf	Dorian.KLARF12 オブジェクト

16.1. プロパティ

16.1.1. Index プロパティ

レコードのインデックス

Visual C#:

```
int Index
```

16.1.2. klarf プロパティ

Dorian.KLARF12 オブジェクト

Visual C#:

```
Klarf12 klarf
```

17. Dorian.Forms.Log Class

Dorian.Forms.Log はログファイルのコンポーネントです。

プロパティ:

プロパティ	説明
Count	バックアップファイルの数
Enable	ログ機能の有効・無効
FileName	ログファイル名
FileSize	ログファイルのサイズ
Json	JSON 設定
JsonFileName	JSON 設定ファイル名
TimeStamp	タイムスタンプの有効・無効

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Write	文字列を書き込みます

17.1. プロパティ

17.1.1. Count プロパティ

バックアップファイルの数

Visual C#:

```
int Count
```

特記事項

バックアップファイルの数を取得または設定します。

17.1.2. Enable プロパティ

ログ機能の有効・無効

Visual C#:

```
bool Enable
```

特記事項

ログ機能の有効・無効を取得または設定します。

17.1.3. FileName プロパティ

ログファイル名

Visual C#:

```
string FileName
```

特記事項

ログファイル名を取得または設定します。

17.1.4. FileSize プロパティ

ログファイルのサイズ

Visual C#:

```
int FileSize
```

特記事項

ログファイルのサイズを取得または設定します。

17.1.5. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

17.1.6. JsonFileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string JsonFileName
```

17.1.7. TimeStamp プロパティ

タイムスタンプの有効・無効

Visual C#:

```
bool TimeStamp
```

特記事項

タイムスタンプの有効・無効を取得または設定します。

17.2. メソッド

17.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

17.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

17.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

17.2.4. Write メソッド

文字列を書き込みます

Visual C#:

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

特記事項

文字列をログファイルに書き込みます。

引数

変数名	説明
-----	----

value

文字列

18. Dorian.Forms.LogView Class

Dorian.Forms.LogView はログファイルのコンポーネントです。

プロパティ:

プロパティ	説明
Count	バックアップファイルの数
Enable	ログ機能の有効・無効
FileName	ログファイル名
FileSize	ログファイルのサイズ
Json	JSON 設定
JsonFileName	JSON 設定ファイル名
TimeStamp	タイムスタンプの有効・無効

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Write	文字列を書き込みます

18.1. プロパティ

18.1.1. Count プロパティ

バックアップファイルの数

Visual C#:

```
int Count
```

特記事項

バックアップファイルの数を取得または設定します。

18.1.2. Enable プロパティ

ログ機能の有効・無効

Visual C#:

```
bool Enable
```

特記事項

ログ機能の有効・無効を取得または設定します。

18.1.3. FileName プロパティ

ログファイル名

Visual C#:

```
string FileName
```

特記事項

ログファイル名を取得または設定します。

18.1.4. FileSize プロパティ

ログファイルのサイズ

Visual C#:

```
int FileSize
```

特記事項

ログファイルのサイズを取得または設定します。

18.1.5. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

18.1.6. JsonFileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string JsonFileName
```

18.1.7. TimeStamp プロパティ

タイムスタンプの有効・無効

Visual C#:

```
bool TimeStamp
```

特記事項

タイムスタンプの有効・無効を取得または設定します。

18.2. メソッド

18.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

18.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

18.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

18.2.4. Write メソッド

文字列を書き込みます

Visual C#:

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

特記事項

文字列をログファイルに書き込みます。

引数

変数名	説明
-----	----

value	文字列
-------	-----

19. Dorian.Forms.Secsl Class

Dorian.Forms.Secsl は SEMI E.4 (SECS-I) 通信用のコンポーネントです。

プロパティ:

プロパティ	説明
AutoSessionID	自動的にデバイス ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
BaudRate	ボーレート
Connect	通信ポートを開く・閉じる
FileName	JSON 設定ファイル名
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
Json	JSON 設定
Master	マスターカスレーブか
MDLN	装置モデル名
Port	シリアルポート番号
Retry	リトライ回数
SessionID	デバイス ID
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T1	T1 タイマー
T2	T2 タイマー
T3	T3 タイマー
T4	T4 タイマー

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージを送信する

イベント:

イベント	説明
ProblemEvent	エラーが発生した
ReceivedEvent	データを受信した
ReceivedRawEvent	文字データを受信した
SentEvent	メッセージを受信した
SentRawEvent	文字データを送信した

プロパティ

19.1.1. AutoSessionID プロパティ

自動的にデバイス ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にデバイス ID を補正する。

19.1.2. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

19.1.3. BaudRate プロパティ

ボーレート

Visual C#:

```
int BaudRate
```

特記事項

ボーレートを取得または設定します。

19.1.4. Connect プロパティ

通信ポートを開く・閉じる

Visual C#:

```
bool Connect
```

特記事項

SECS-I の接続状態を取得または設定します。

19.1.5. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

19.1.6. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

19.1.7. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

19.1.8. Master プロパティ

マスターかスレーブか

Visual C#:

```
bool Master
```

特記事項

マスターかスレーブかを取得または設定します。

19.1.9. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

19.1.10. Port プロパティ

シリアルポート番号

Visual C#:

```
int Port
```

特記事項

通信ポート番号を取得または設定します。

19.1.11. Retry プロパティ

リトライ回数

Visual C#:

```
int Retry
```


特記事項

リトライ回数を取得または設定します。

19.1.12. SessionID プロパティ

デバイス ID

Visual C#:

```
UInt16 SessionID
```

特記事項

デバイス ID を取得または設定します。

19.1.13. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

19.1.14. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

19.1.15. T1 プロパティ

T1 タイマー

Visual C#:

```
double T1
```

特記事項

SECS-I の T1 タイムアウトを秒単位で取得または設定します。デフォルトは 1 秒です。

19.1.16. T2 プロパティ

T2 タイマー

Visual C#:

```
double T2
```

特記事項

SECS-I の T2 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

19.1.17. T3 プロパティ

T3 タイマー

Visual C#:

```
double T3
```

特記事項

SECS-I の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

19.1.18. T4 プロパティ

T4 タイマー

Visual C#:

```
double T4
```

特記事項

SECS-I の T4 タイムアウトを秒単位で取得または設定します。デフォルトは 60 秒です。

19.2. メソッド

19.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

19.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

19.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

19.2.4. Send メソッド

メッセージを送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message	SECS-II メッセージ
---------	---------------

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

19.3. イベント

19.3.1. ProblemEvent メソッド

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

19.3.2. ReceivedEvent メソッド

データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

19.3.3. ReceivedRawEvent メソッド

文字データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

19.3.4. SentEvent メソッド

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

19.3.5. SentRawEvent メソッド

文字データを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

20. Dorian.Forms.SecsII Class

Dorian.Forms.SecsII コンポーネント は、SEMI E5 (SECS-II)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。

Dorian.Hsms コンポーネントや Dorian.SecsI コンポーネントなどと組み合わせて使います。

プロパティ:

プロパティ	説明
BlockNumber	ブロック番号 (SECS-I)
DeviceID	デバイス ID (SECS-I)
Ebit	エンドビット
Error	SML のコンパイルに失敗したか
Function	ファンクション番号
Host	ホストまたは装置
Hsms	HSMS または SECS-I
Json	JSON
Msg	メッセージデータ (バイナリ)
Node	ノード
NodeCount	ノードの要素数
NodeType	ノードの型
NodeValue	ノードの値
NodeValueHex	ノードの値 (16 進数)
PType	P タイプ (HSMS)
Rbit	リバースビット (SECS-I)
SessionID	セッション ID (HSMS)
Sml	SML
SmlType	SML 書式 (未サポート)
SourceID	ソース ID (SECS-I)
Stream	ストリーム番号
SType	S タイプ (HSMS)
SuggestedReplyMsg	推奨される返信メッセージ
SystemBytes	システムバイト
this[string]	指定したノードの値を取得します
TransactionID	トランザクション ID (SECS-I)
Verification	認証結果
Wbit	ウェイトビット

メソッド:

メソッド	説明
Reply	返信用にヘッダをセットします
Reply	返信用にヘッダをセットします
Reset	メッセージをリセットします

20.1. プロパティ

20.1.1. BlockNumber プロパティ

ブロック番号 (SECS-I)

Visual C#:

```
UInt16 BlockNumber
```

特記事項

ブロック番号を取得または設定します。SECS-I でのみ使用されます。

SECS-I の受信メッセージで BlockNumber が 1 以外の場合は、マルチブロックメッセージだったことを示します。送信時は必ず 1 にセットします。もしメッセージのサイズが大きくて 1 ブロックに収まりきらない場合は、自動的にマルチブロックにして送信されます。

20.1.2. DeviceID プロパティ

デバイス ID (SECS-I)

Visual C#:

```
UInt16 DeviceID
```

特記事項

デバイス ID を取得または設定します。デバイス ID は SECS-II ヘッダの先頭 2 ビット目から 15 ビットです。

デバイス ID とセッション ID はほぼ同じですが、デバイス ID は 15 ビット、セッション ID は 16 ビットです。

20.1.3. Ebit プロパティ

エンドビット

Visual C#:

```
bool Ebit
```

特記事項

エンドビットを取得または設定します。SECS-I でのみ使用されます。

SECS-I の受信メッセージのエンドビットは常に True となります。これは Dorian.SecsI コンポーネントが最終ブロックまで受信してから Received イベントを発生させるからです。

20.1.4. Error プロパティ

SML のコンパイルに失敗したか

Visual C#:

```
bool Error
```

特記事項

SML 文字列の処理でエラーが発生したかどうかを取得します。

読み出し専用プロパティ。

20.1.5. Function プロパティ

ファンクション番号

Visual C#:

```
Byte Function
```

特記事項

ファンクション番号を取得または設定します。

20.1.6. **Host プロパティ**

ホストまたは装置

Visual C#:

```
bool Host
```

特記事項

このメッセージを送信してきた側がホストなら true、装置なら false をセットします。Dorian.Forms.SecsII コンポーネントの役割を取得または設定します。このプロパティは Verification プロパティを使ってメッセージの構造を検証する場合と、検証の結果として SuggestedReplyMsg プロパティにセットされるメッセージにのみ影響します。

20.1.7. **Hsms プロパティ**

HSMS または SECS-I

Visual C#:

```
bool Hsms
```

特記事項

HSMS か SECS-I かを取得または設定します。デフォルトは HSMS です。

20.1.8. **Json プロパティ**

JSON

Visual C#:

```
string Json
```

特記事項

SECS-II メッセージを JSON 文字列として取得または設定します。Json プロパティを読み出すとコントロールにセットされているメッセージの構造をツリー形式の JSON 文字列で取得します。詳細については付録 C を参照のこと。

20.1.9. **Msg プロパティ**

メッセージデータ(バイナリ)

Visual C#:

```
Byte[] Msg
```

特記事項

メッセージデータを取得または設定します。

20.1.10. Node プロパティ

ノード

Visual C#:

```
string Node
```

特記事項

メッセージ内容を操作するためのノードを取得または設定します。ノードは"/" (スラッシュ)とノード番号と"["、"]" (カギカッコ)で構成されます。ノード番号は 1 から始まる数字です。ノードが空文字列の場合はルートが指定されたときみなされます。

20.1.11. NodeCount プロパティ

ノードの要素数

Visual C#:

```
int NodeCount
```

特記事項

サブアイテムの個数を取得します。もしノードプロパティがリスト型の場合、このプロパティはサブノードの数を意味します。それ以外は配列の個数です。

読み出し専用プロパティ。

20.1.12. NodeType プロパティ

ノードの型

Visual C#:

```
SecsType NodeType
```

特記事項

ノードタイプを取得します。

読み出し専用プロパティ。

20.1.13. NodeValue プロパティ

ノードの値

Visual C#:

```
string NodeValue
```

特記事項

ノードの値を取得します。もしノードが数値型の場合、値は 10 進法文字列に変換されます。

読み出し専用プロパティ。

20.1.14. NodeValueHex プロパティ

ノードの値(16 進数)

Visual C#:

```
string NodeValueHex
```

特記事項

ノードの値を 16 進法表現で取得します。
読み出し専用プロパティ。

20.1.15. PType プロパティ

P タイプ(HSMS)

Visual C#:

```
Byte PType
```

特記事項

プレゼンテーションタイプを取得または設定します。
SECS-II メッセージを使いますので、0 を指定します。SEMI E.5 では現在は SECS-II のみが規定されているため、PType プロパティは常に 0 です。

20.1.16. Rbit プロパティ

リバースビット(SECS-I)

Visual C#:

```
bool Rbit
```

特記事項

リバースビットを取得または設定します。

20.1.17. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

20.1.18. Sml プロパティ

SML

Visual C#:

```
string Sml
```

特記事項

SECS-II メッセージを SML 文字列として取得または設定します。SML プロパティを読み出すとコントロールにセットされているメッセージの構造をツリー形式の SML 文字列で取得します。また SML プロパティに SML 文字列をセットする場合、文字列中に改行コードやスペース、タブなどを自由にに入れても構いません(無視されます)。
詳細については付録 A を参照のこと。

20.1.19. SmlType プロパティ

SML 書式 (未サポート)

Visual C#:

```
Byte SmlType
```

特記事項

SML の書式。

20.1.20. SourceID プロパティ

ソース ID (SECS-I)

Visual C#:

```
UInt16 SourceID
```

特記事項

ソース ID を取得または設定します。

20.1.21. Stream プロパティ

ストリーム番号

Visual C#:

```
Byte Stream
```

特記事項

ストリーム番号を取得または設定します。

20.1.22. SType プロパティ

S タイプ (HSMS)

Visual C#:

```
Byte SType
```

特記事項

セッションタイプを取得または設定します。

20.1.23. SuggestedReplyMsg プロパティ

推奨される返信メッセージ

Visual C#:

```
Byte[] SuggestedReplyMsg
```

特記事項

メッセージ構造を検証した結果、もっとも適切だと考えられる応答メッセージを取得または設定します。

20.1.24. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

20.1.25. this[string] プロパティ

指定したノードの値を取得します

Visual C#:

```
this[string]
```

特記事項

Node プロパティと NodeValue プロパティを組み合わせたシンタックス・シュガーです。読み出し専用プロパティ。

20.1.26. TransactionID プロパティ

トランザクション ID (SECS-I)

Visual C#:

```
UInt16 TransactionID
```

特記事項

トランザクション ID を取得または設定します。

20.1.27. Verification プロパティ

認証結果

Visual C#:

```
SecsVerify Verification
```

特記事項

Sml プロパティまたは Msg プロパティにセットしたした際、メッセージが正しいかどうかを認証し、その認証結果がセットされます。検証したメッセージが一次メッセージの場合、SuggestedReplyMsg プロパティに推奨される二次メッセージがセットされます。

20.1.28. Wbit プロパティ

ウェイトビット

Visual C#:

```
bool wbit
```

特記事項

ウェイトビットを取得または設定します。
返信要求がある場合は True となります。

20.2. メソッド

20.2.1. Reply メソッド

返信用にヘッダをセットします

Visual C#:

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

特記事項

指定されたメッセージの返信メッセージとして SECS-II ヘッダを初期化します。もし指定されたメッセージが HSMS コントロールメッセージの場合、Dorian.Forms.SecsII コンポーネントはメッセージボディを消去します。その他の場合はメッセージボディは影響を受けません。

引数

変数名	説明
primary	一次メッセージ

20.2.2. Reply メソッド

返信用にヘッダをセットします

Visual C#:

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

特記事項

指定されたメッセージの返信メッセージとして SECS-II ヘッダを初期化します。もし指定されたメッセージが HSMS コントロールメッセージの場合、Dorian.Forms.SecsII コンポーネントはメッセージボディを消去します。その他の場合はメッセージボディは影響を受けません。

引数

変数名	説明
primary	一次メッセージ

20.2.3. Reset メソッド

メッセージをリセットします

Visual C#:

```
public void Reset()
```

特記事項

内部のデータ構造とパラメータを初期化します。

21. Dorian.Forms.SecsIIView Class

Dorian.Forms.SecsIIView コンポーネント は、SEMI E5 (SECS-II)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。

Dorian.Hsms コンポーネントや Dorian.SecsI コンポーネントなどと組み合わせて使います。

プロパティ:

プロパティ	説明
BlockNumber	ブロック番号 (SECS-I)
DeviceID	デバイス ID (SECS-I)
Ebit	エンドビット
Error	SML のコンパイルに失敗したか
Function	ファンクション番号
Host	ホストまたは装置
Hsms	HSMS または SECS-I
Json	JSON
Msg	メッセージデータ (バイナリ)
Node	ノード
NodeCount	ノードの要素数
NodeItem	ノードアイテム
NodeType	ノードの型
NodeValue	ノードの値
NodeValueHex	ノードの値 (16 進数)
PType	P タイプ (HSMS)
Rbit	リバースビット (SECS-I)
SessionID	セッション ID (HSMS)
Sml	SML
SmlType	SML 書式 (未サポート)
SourceID	ソース ID (SECS-I)
Stream	ストリーム番号
SType	S タイプ (HSMS)
SuggestedReplyMsg	推奨される返信メッセージ
SystemBytes	システムバイト
this[string]	指定したノードの値を取得します
TransactionID	トランザクション ID (SECS-I)
Verification	認証結果
Wbit	ウェイトビット

メソッド:

メソッド	説明
Reply	返信用にヘッダをセットします
Reply	返信用にヘッダをセットします
Reset	メッセージをリセットします

21.1. プロパティ

21.1.1. BlockNumber プロパティ

ブロック番号 (SECS-I)

Visual C#:

```
UInt16 BlockNumber
```

特記事項

ブロック番号を取得または設定します。SECS-I でのみ使用されます。

SECS-I の受信メッセージで BlockNumber が 1 以外の場合は、マルチブロックメッセージだったことを示します。送信時は必ず 1 にセットします。もしメッセージのサイズが大きくて 1 ブロックに収まりきらない場合は、自動的にマルチブロックにして送信されます。

21.1.2. DeviceID プロパティ

デバイス ID (SECS-I)

Visual C#:

```
UInt16 DeviceID
```

特記事項

デバイス ID を取得または設定します。デバイス ID は SECS-II ヘッダの先頭 2 ビット目から 15 ビットです。

デバイス ID とセッション ID はほぼ同じですが、デバイス ID は 15 ビット、セッション ID は 16 ビットです。

21.1.3. Ebit プロパティ

エンドビット

Visual C#:

```
bool Ebit
```

特記事項

エンドビットを取得または設定します。SECS-I でのみ使用されます。

SECS-I の受信メッセージのエンドビットは常に True となります。これは Dorian.SecsI コンポーネントが最終ブロックまで受信してから Received イベントを発生させるからです。

21.1.4. Error プロパティ

SML のコンパイルに失敗したか

Visual C#:

```
bool Error
```

特記事項

SML 文字列の処理でエラーが発生したかどうかを取得します。

読み出し専用プロパティ。

21.1.5. Function プロパティ

ファンクション番号

Visual C#:

```
Byte Function
```

特記事項

ファンクション番号を取得または設定します。

21.1.6. Host プロパティ

ホストまたは装置

Visual C#:

```
bool Host
```

特記事項

このメッセージを送信してきた側がホストなら true、装置なら false をセットします。Dorian.Forms.SecsII コンポーネントの役割を取得または設定します。このプロパティは Verification プロパティを使ってメッセージの構造を検証する場合と、検証の結果として SuggestedReplyMsg プロパティにセットされるメッセージにのみ影響します。

21.1.7. Hsms プロパティ

HSMS または SECS-I

Visual C#:

```
bool Hsms
```

特記事項

HSMS か SECS-I かを取得または設定します。デフォルトは HSMS です。

21.1.8. Json プロパティ

JSON

Visual C#:

```
string Json
```

特記事項

SECS-II メッセージを JSON 文字列として取得または設定します。Json プロパティを読み出すとコントロールにセットされているメッセージの構造をツリー形式の JSON 文字列で取得します。

詳細については付録 C を参照のこと。

21.1.9. Msg プロパティ

メッセージデータ(バイナリ)

Visual C#:

```
Byte[] Msg
```

特記事項

メッセージデータを取得または設定します。

21.1.10. Node プロパティ

ノード

Visual C#:

```
string Node
```

特記事項

メッセージ内容进行操作するためのノードを取得または設定します。ノードは"/" (スラッシュ)とノード番号と"["、"]" (カギカッコ)で構成されます。ノード番号は 1 から始まる数字です。ノードが空文字列の場合はルートが指定されたときみなされます。

21.1.11. NodeCount プロパティ

ノードの要素数

Visual C#:

```
int NodeCount
```

特記事項

サブアイテムの個数を取得します。もしノードプロパティがリスト型の場合、このプロパティはサブノードの数を意味します。それ以外は配列の個数です。

読み出し専用プロパティ。

21.1.12. NodeItem プロパティ

ノードアイテム

Visual C#:

```
NodeItemClass NodeItem
```

特記事項

メッセージ内容进行操作するためのノードを取得します。

21.1.13. NodeType プロパティ

ノードの型

Visual C#:

```
SecsType NodeType
```

特記事項

ノードタイプを取得します。

読み出し専用プロパティ。

21.1.14. NodeValue プロパティ

ノードの値

Visual C#:

```
string NodeValue
```

特記事項

ノードの値を取得します。もしノードが数値型の場合、値は 10 進法文字列に変換されます。
読み出し専用プロパティ。

21.1.15. NodeValueHex プロパティ

ノードの値(16 進数)

Visual C#:

```
string NodeValueHex
```

特記事項

ノードの値を 16 進法表現で取得します。
読み出し専用プロパティ。

21.1.16. PType プロパティ

P タイプ(HSMS)

Visual C#:

```
Byte PType
```

特記事項

プレゼンテーションタイプを取得または設定します。
SECS-II メッセージを使いますので、0 を指定します。SEMI E.5 では現在は SECS-II のみが規定されているため、PType プロパティは常に 0 です。

21.1.17. Rbit プロパティ

リバースビット(SECS-I)

Visual C#:

```
bool Rbit
```

特記事項

リバースビットを取得または設定します。

21.1.18. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

21.1.19. Sml プロパティ

SML

Visual C#:

```
string SmI
```

特記事項

SECS-II メッセージを SML 文字列として取得または設定します。SML プロパティを読み出すとコントロールにセットされているメッセージの構造をツリー形式の SML 文字列で取得します。また SML プロパティに SML 文字列をセットする場合、文字列中に改行コードやスペース、タブなどを自由に入れても構いません(無視されます)。

詳細については付録 A を参照のこと。

21.1.20. SmIType プロパティ

SML 書式(未サポート)

Visual C#:

```
Byte SmIType
```

特記事項

SML の書式。

21.1.21. SourceID プロパティ

ソース ID(SECS-I)

Visual C#:

```
UInt16 SourceID
```

特記事項

ソース ID を取得または設定します。

21.1.22. Stream プロパティ

ストリーム番号

Visual C#:

```
Byte Stream
```

特記事項

ストリーム番号を取得または設定します。

21.1.23. SType プロパティ

S タイプ(HSMS)

Visual C#:

```
Byte SType
```

特記事項

セッションタイプを取得または設定します。

21.1.24. SuggestedReplyMsg プロパティ

推奨される返信メッセージ

Visual C#:

```
Byte[] SuggestedReplyMsg
```

特記事項

メッセージ構造を検証した結果、もっとも適切だと考えられる応答メッセージを取得または設定します。

21.1.25. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

21.1.26. this[string] プロパティ

指定したノードの値を取得します

Visual C#:

```
this[string]
```

特記事項

Node プロパティと NodeValue プロパティを組み合わせたシンタックス・シュガーです。読み出し専用プロパティ。

21.1.27. TransactionID プロパティ

トランザクション ID (SECS-I)

Visual C#:

```
UInt16 TransactionID
```

特記事項

トランザクション ID を取得または設定します。

21.1.28. Verification プロパティ

認証結果

Visual C#:

```
SecsVerify Verification
```

特記事項

Sml プロパティまたは Msg プロパティにセットした際、メッセージが正しいかどうかを認証し、その認証結果がセットされます。検証したメッセージが一次メッセージの場合、SuggestedReplyMsg プロパティに推奨される二次メッセージがセットされます。

21.1.29. Wbit プロパティ

ウェイトビット

Visual C#:

```
bool wbit
```

特記事項

ウェイトビットを取得または設定します。
返信要求がある場合は True となります。

21.2. メソッド

21.2.1. Reply メソッド

返信用にヘッダをセットします

Visual C#:

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

特記事項

指定されたメッセージの返信メッセージとして SECS-II ヘッダを初期化します。もし指定されたメッセージが HSMS コントロールメッセージの場合、Dorian.Forms.SecsII コンポーネントはメッセージボディを消去します。その他の場合はメッセージボディは影響を受けません。

引数

変数名	説明
primary	一次メッセージ

21.2.2. Reply メソッド

返信用にヘッダをセットします

Visual C#:

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

特記事項

指定されたメッセージの返信メッセージとして SECS-II ヘッダを初期化します。もし指定されたメッセージが HSMS コントロールメッセージの場合、Dorian.Forms.SecsII コンポーネントはメッセージボディを消去します。その他の場合はメッセージボディは影響を受けません。

引数

変数名	説明
primary	一次メッセージ

21.2.3. Reset メソッド

メッセージをリセットします

Visual C#:

```
public void Reset()
```

特記事項

内部のデータ構造とパラメータを初期化します。

22. Dorian.Forms.SecsIView Class

Dorian.Forms.SecsIView は SEMI E.4(SECS-I) 通信用のコンポーネントです。

プロパティ:

プロパティ	説明
AutoSessionID	自動的にデバイス ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
BaudRate	ボーレート
Connect	通信ポートを開く・閉じる
FileName	JSON 設定ファイル名
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
Json	JSON 設定
Master	マスターカスレーブか
MDLN	装置モデル名
Port	シリアルポート番号
Retry	リトライ回数
SessionID	デバイス ID
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T1	T1 タイマー
T2	T2 タイマー
T3	T3 タイマー
T4	T4 タイマー

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージを送信する

イベント:

イベント	説明
ProblemEvent	エラーが発生した
ReceivedEvent	データを受信した
ReceivedRawEvent	文字データを受信した
SentEvent	メッセージを受信した
SentRawEvent	文字データを送信した

22.1. プロパティ

22.1.1. AutoSessionID プロパティ

自動的にデバイス ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にデバイス ID を補正する。

22.1.2. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

22.1.3. BaudRate プロパティ

ボーレート

Visual C#:

```
int BaudRate
```

特記事項

ボーレートを取得または設定します。

22.1.4. Connect プロパティ

通信ポートを開く・閉じる

Visual C#:

```
bool Connect
```

特記事項

SECS-I の接続状態を取得または設定します。

22.1.5. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

22.1.6. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

22.1.7. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

22.1.8. Master プロパティ

マスターかスレーブか

Visual C#:

```
bool Master
```

特記事項

マスターかスレーブかを取得または設定します。

22.1.9. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

22.1.10. Port プロパティ

シリアルポート番号

Visual C#:

```
int Port
```

特記事項

通信ポート番号を取得または設定します。

22.1.11. Retry プロパティ

リトライ回数

Visual C#:

```
int Retry
```

特記事項

リトライ回数を取得または設定します。

22.1.12. SessionID プロパティ

デバイス ID

Visual C#:

```
UInt16 SessionID
```

特記事項

デバイス ID を取得または設定します。

22.1.13. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

22.1.14. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

22.1.15. T1 プロパティ

T1 タイマー

Visual C#:

```
double T1
```

特記事項

SECS-I の T1 タイムアウトを秒単位で取得または設定します。デフォルトは 1 秒です。

22.1.16. T2 プロパティ

T2 タイマー

Visual C#:

```
double T2
```

特記事項

SECS-I の T2 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

22.1.17. T3 プロパティ

T3 タイマー

Visual C#:

```
double T3
```

特記事項

SECS-I の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

22.1.18. T4 プロパティ

T4 タイマー

Visual C#:

```
double T4
```

特記事項

SECS-I の T4 タイムアウトを秒単位で取得または設定します。デフォルトは 60 秒です。

22.2. メソッド

22.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

22.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

22.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

22.2.4. Send メソッド

メッセージを送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message	SECS-II メッセージ
---------	---------------

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

22.3. イベント

22.3.1. ProblemEvent メソッド

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

22.3.2. ReceivedEvent メソッド

データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

22.3.3. ReceivedRawEvent メソッド

文字データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

22.3.4. SentEvent メソッド

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

22.3.5. SentRawEvent メソッド

文字データを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

23. Dorian.Gem Class

Dorian.Gem コンポーネントは SEMI E30 (GEM)の機能を作成するための開発支援製品です。装置側ソフトの開発で使えます。

プロパティ:

プロパティ	説明
AlarmModel	GEM アラームモデルオブジェクト
CallDefProc	Received イベントの後に DefProc()を呼ぶ
CommModel	GEM 通信モデルオブジェクト
CtrlModel	GEM コントロールモデルオブジェクト
EventModel	GEM イベントモデルオブジェクト
FileName	JSON 設定ファイル名
Hsms	HSMS オブジェクト
Json	JSON 設定
Msg	SECS-II オブジェクト

メソッド:

メソッド	説明
Config	設定画面を表示します
InvokeAlarm	アラームを送信します。
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
TimerProc	このメソッドを周期的に呼び出す必要があります。

イベント:

イベント	説明
CommStateChangedEvent	通信状態が変化した
ConnectedEvent	クライアントから接続された
CtrlStateChangedEvent	コントロール状態が変化した
DisconnectedEvent	切断された
InvokeEvent	イベントを送信します。
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
SentEvent	メッセージを送信した

23.1. プロパティ

23.1.1. AlarmModel プロパティ

GEM アラームモデルオブジェクト

Visual C#:

```
AlarmModelClass AlarmModel
```

23.1.2. CallDefProc プロパティ

Received イベントの後に DefProc() を呼ぶ

Visual C#:

```
bool CallDefProc
```

23.1.3. CommModel プロパティ

GEM 通信モデルオブジェクト

Visual C#:

```
CommModelClass CommModel
```

23.1.4. CtrlModel プロパティ

GEM コントロールモデルオブジェクト

Visual C#:

```
CtrlModelClass CtrlModel
```

23.1.5. EventModel プロパティ

GEM イベントモデルオブジェクト

Visual C#:

```
EventModelClass EventModel
```

23.1.6. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

23.1.7. Hsms プロパティ

HSMS オブジェクト

Visual C#:

```
HsmsClass Hsms
```

23.1.8. **Json プロパティ**

JSON 設定

Visual C#:

```
string Json
```

23.1.9. **Msg プロパティ**

SECS-II オブジェクト

Visual C#:

```
SecsIIClass Msg
```

23.2. メソッド

23.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

23.2.2. InvokeAlarm メソッド

アラームを送信します。

Visual C#:

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

引数

変数名	説明
alid	ALID
occur	アラーム発生・解除

戻り値

成功した場合は true が、失敗した場合は false が返ります。

23.2.3. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

23.2.4. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

23.2.5. TimerProc メソッド

このメソッドを周期的に呼び出す必要があります。

Visual C#:

```
public void TimerProc()
```

23.3. イベント

23.3.1. CommStateChangedEvent イベント

通信状態が変化した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

23.3.2. ConnectedEvent イベント

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

23.3.3. CtrlStateChangedEvent イベント

コントロール状態が変化した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

23.3.4. DisconnectedEvent イベント

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

23.3.5. InvokeEvent メソッド

イベントを送信します。

Visual C#:

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

引数

変数名	説明
ceid	CEID

戻り値

成功した場合は true が、失敗した場合は false が返ります。

23.3.6. ProblemEvent イベント

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

23.3.7. ReceivedEvent イベント

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

23.3.8. SentEvent イベント

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

24. Dorian.Gem.AlarmModelClass Class

GEM アラームモデル

プロパティ:

プロパティ	説明
Count	アラームの数
this[int]	アラームオブジェクトを取得する

メソッド:

メソッド	説明
Add	アラームを追加する
Remove	アラームを削除する

24.1. プロパティ

24.1.1. Count プロパティ

アラームの数

Visual C#:

```
int Count
```

24.1.2. this[int] プロパティ

アラームオブジェクトを取得する

Visual C#:

```
this[int]
```

24.2. メソッド

24.2.1. Add メソッド

アラームを追加する

Visual C#:

```
public Alarm Add()
```

戻り値

24.2.2. Remove メソッド

アラームを削除する

Visual C#:

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

引数

変数名	説明
index	インデックス

戻り値

成功した場合は true が、失敗した場合は false が返ります。

25. Dorian.Gem.AlarmModelClass.Alarm Class

GEM アラーム

プロパティ:

プロパティ	説明
Code	アラームコード
Enable	アラームの有効・無効
ID	アラーム ID
Text	アラームテキスト

25.1. プロパティ

25.1.1. Code プロパティ

アラームコード

Visual C#:

```
Byte Code
```

25.1.2. Enable プロパティ

アラームの有効・無効

Visual C#:

```
bool Enable
```

25.1.3. ID プロパティ

アラーム ID

Visual C#:

```
int ID
```

25.1.4. Text プロパティ

アラームテキスト

Visual C#:

```
string Text
```

26. Dorian.Gem.CommModelClass Class

GEM 通信モデル

プロパティ:

プロパティ	説明
CommDelayTimer	S1F13 の送信間隔
EstablishComm	装置からの通信開始要求
InitialState	初期通信状態
MDLN	装置モデル名
SOFTREV	装置ソフトバージョン
State	現在の通信状態

26.1. プロパティ

26.1.1. CommDelayTimer プロパティ

S1F13 の送信間隔

Visual C#:

```
int CommDelayTimer
```

26.1.2. EstablishComm プロパティ

装置からの通信開始要求

Visual C#:

```
bool EstablishComm
```

26.1.3. InitialState プロパティ

初期通信状態

Visual C#:

```
GemCommState InitialState
```

26.1.4. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

26.1.5. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

26.1.6. State プロパティ

現在の通信状態

Visual C#:

```
GemCommState State
```


27. Dorian.Gem.CommStateChangedEventArgs Class

CommStateChanged イベント引数

プロパティ:

プロパティ	説明
newState	新しい状態
prevState	以前の状態
sender	このイベントの送信元

27.1. プロパティ

27.1.1. newState プロパティ

新しい状態

Visual C#:

```
GemCtrlState newState
```

27.1.2. prevState プロパティ

以前の状態

Visual C#:

```
GemCtrlState prevState
```

27.1.3. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

28. Dorian.Gem.ConnectedEventArgs Class

Connected イベント引数

プロパティ:

プロパティ	説明
ip	IP アドレス
port	TCP ポート番号
sender	このイベントの送信元

28.1. プロパティ

28.1.1. ip プロパティ

IP アドレス

Visual C#:

```
string ip
```

28.1.2. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

28.1.3. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

29. Dorian.Gem.CtrlModelClass Class

GEM コントロールモデル

プロパティ:

プロパティ	説明
InitialOfflineState	初期オフライン状態
InitialState	初期コントロール状態
OfflineState	デフォルトオフライン状態
OnlineState	デフォルトオンライン状態
State	現在のコントロール状態

29.1. プロパティ

29.1.1. InitialOfflineState プロパティ

初期オフライン状態

Visual C#:

```
GemCtrlState InitialOfflineState
```

29.1.2. InitialState プロパティ

初期コントロール状態

Visual C#:

```
GemCtrlState InitialState
```

29.1.3. OfflineState プロパティ

デフォルトオフライン状態

Visual C#:

```
GemCtrlState OfflineState
```

29.1.4. OnlineState プロパティ

デフォルトオンライン状態

Visual C#:

```
GemCtrlState OnlineState
```

29.1.5. State プロパティ

現在のコントロール状態

Visual C#:

```
GemCtrlState State
```

30. Dorian.Gem.CtrlStateChangedEventArgs Class

CtrlStateChanged イベント引数

プロパティ:

プロパティ	説明
newState	新しい状態
prevState	以前の状態
sender	このイベントの送信元

30.1. プロパティ

30.1.1. newState プロパティ

新しい状態

Visual C#:

```
GemCtrlState newState
```

30.1.2. prevState プロパティ

以前の状態

Visual C#:

```
GemCtrlState prevState
```

30.1.3. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```


31. Dorian.Gem.Data Class

JSON データ

プロパティ:

プロパティ	説明
alarmModel	GEM アラームモデルオブジェクト
commModel	GEM 通信モデルオブジェクト
ctrlModel	GEM コントロールモデルオブジェクト
eventModel	GEM イベントモデルオブジェクト
hsms	HSMS オブジェクト

31.1. プロパティ

31.1.1. alarmModel プロパティ

GEM アラームモデルオブジェクト

Visual C#:

```
AlarmModel alarmModel
```

31.1.2. commModel プロパティ

GEM 通信モデルオブジェクト

Visual C#:

```
CommModel commModel
```

31.1.3. ctrlModel プロパティ

GEM コントロールモデルオブジェクト

Visual C#:

```
CtrlModel ctrlModel
```

31.1.4. eventModel プロパティ

GEM イベントモデルオブジェクト

Visual C#:

```
EventModel eventModel
```

31.1.5. hsms プロパティ

HSMS オブジェクト

Visual C#:

```
Data hsms
```

32. Dorian.Gem.Data.AlarmModel Class

GEM アラームモデル

プロパティ:

プロパティ	説明
alarms	GEM アラームリスト

32.1. プロパティ

32.1.1. alarms プロパティ

GEM アラームリスト

Visual C#:

Alarms **alarms**

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

GEM アラーム

プロパティ:

プロパティ	説明
code	アラームコード
enable	アラームの有効・無効
id	アラーム ID
text	アラームテキスト

33.1. プロパティ**33.1.1. code プロパティ**

アラームコード

Visual C#:

```
Byte code
```

33.1.2. enable プロパティ

アラームの有効・無効

Visual C#:

```
bool enable
```

33.1.3. id プロパティ

アラームID

Visual C#:

```
int id
```

33.1.4. text プロパティ

アラームテキスト

Visual C#:

```
string text
```

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

GEM アラームリスト

35. Dorian.Gem.Data.CommModel Class

GEM 通信モデル。

プロパティ:

プロパティ	説明
establishComm	装置からの通信開始要求
initialState	初期通信状態
mdlIn	装置モデル名
softrev	装置ソフトバージョン

35.1. プロパティ

35.1.1. establishComm プロパティ

装置からの通信開始要求

Visual C#:

```
bool establishComm
```

35.1.2. initialState プロパティ

初期通信状態

Visual C#:

```
GemCommState initialState
```

35.1.3. mdlName プロパティ

装置モデル名

Visual C#:

```
string mdlName
```

35.1.4. softwareVersion プロパティ

装置ソフトウェアバージョン

Visual C#:

```
string softwareVersion
```

36. Dorian.Gem.Data.CtrlModel Class

GEM コントロールモデル。

プロパティ:

プロパティ	説明
initialOfflineState	初期オフライン状態
initialState	初期コントロール状態
offlineState	デフォルトオフライン状態
onlineState	デフォルトオンライン状態

36.1. プロパティ

36.1.1. initialOfflineState プロパティ

初期オフライン状態

Visual C#:

```
GemCtrlState initialOfflineState
```

36.1.2. initialState プロパティ

初期コントロール状態

Visual C#:

```
GemCtrlState initialState
```

36.1.3. offlineState プロパティ

デフォルトオフライン状態

Visual C#:

```
GemCtrlState offlineState
```

36.1.4. onlineState プロパティ

デフォルトオンライン状態

Visual C#:

```
GemCtrlState onlineState
```

37. Dorian.Gem.Data.EventModel Class

GEM イベントモデル

プロパティ:

プロパティ	説明
events	GEM イベントリスト
reports	GEM レポートリスト
variables	GEM 変数リスト

37.1. プロパティ

37.1.1. events プロパティ

GEM イベントリスト

Visual C#:

Events **events**

37.1.2. reports プロパティ

GEM レポートリスト

Visual C#:

Reports **reports**

37.1.3. variables プロパティ

GEM 変数リスト

Visual C#:

Variables **variables**

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

GEM イベント

プロパティ:

プロパティ	説明
definition	定義済みイベント ID
description	イベントの説明
enable	イベントの有効・無効
id	イベント ID
reports	レポート ID リスト

38.1. プロパティ

38.1.1. definition プロパティ

定義済みイベント ID

Visual C#:

```
int definition
```

38.1.2. description プロパティ

イベントの説明

Visual C#:

```
string description
```

38.1.3. enable プロパティ

イベントの有効・無効

Visual C#:

```
bool enable
```

38.1.4. id プロパティ

イベント ID

Visual C#:

```
int id
```

38.1.5. reports プロパティ

レポート ID リスト

Visual C#:

```
List<int> reports
```

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

GEM イベントリスト

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

GEM レポート

プロパティ:

プロパティ	説明
description	レポートの説明
id	レポート ID
variables	VID リスト

40.1. プロパティ

40.1.1. description プロパティ

レポートの説明

Visual C#:

```
string description
```

40.1.2. id プロパティ

レポートID

Visual C#:

```
int id
```

40.1.3. variables プロパティ

VIDリスト

Visual C#:

```
List<int> variables
```

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

GEM レポートリスト

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

GEM 変数

プロパティ:

プロパティ	説明
defaultValue	変数のデフォルト値
definition	定義済み変数 ID
description	変数の説明
id	変数 ID
max	変数の最大値
min	変数の最小値
secsType	変数 SECS-II タイプ
sml	変数の SML
type	変数タイプ
unit	変数の単位

42.1. プロパティ

42.1.1. defaultValue プロパティ

変数のデフォルト値

Visual C#:

```
string defaultValue
```

42.1.2. definition プロパティ

定義済み変数 ID

Visual C#:

```
PredefinedVID definition
```

42.1.3. description プロパティ

変数の説明

Visual C#:

```
string description
```

42.1.4. id プロパティ

変数 ID

Visual C#:

```
int id
```

42.1.5. max プロパティ

変数の最大値

Visual C#:

```
string max
```

42.1.6. min プロパティ

変数の最小値

Visual C#:

```
string min
```

42.1.7. secsType プロパティ

変数 SECS-II タイプ

Visual C#:

```
SecsType secsType
```

42.1.8. sml プロパティ

変数の SML

Visual C#:

```
string sml
```

42.1.9. type プロパティ

変数タイプ

Visual C#:

```
variableType type
```

42.1.10. unit プロパティ

変数の単位

Visual C#:

```
string unit
```

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

GEM 変数リスト

44. Dorian.Gem.DisconnectedEventArgs Class

Disconnected イベント引数

プロパティ:

プロパティ	説明
ip	IP アドレス
port	TCP ポート番号
sender	このイベントの送信元

44.1. プロパティ

44.1.1. ip プロパティ

IP アドレス

Visual C#:

```
string ip
```

44.1.2. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

44.1.3. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

45. Dorian.Gem.EventModelClass Class

GEM イベントモデル

プロパティ:

プロパティ	説明
Events	GEM イベントリスト
Reports	GEM レポートリスト
Variables	GEM 変数リスト

45.1. プロパティ

45.1.1. Events プロパティ

GEM イベントリスト

Visual C#:

```
EventsClass Events
```

45.1.2. Reports プロパティ

GEM レポートリスト

Visual C#:

```
ReportsClass Reports
```

45.1.3. Variables プロパティ

GEM 変数リスト

Visual C#:

```
VariablesClass Variables
```

46. Dorian.Gem.EventModelClass.Event Class

GEM イベント

プロパティ:

プロパティ	説明
Count	レポートの数
Definition	定義済みイベント ID
Description	イベントの説明
Enable	イベントの有効・無効
ID	イベント ID
Reports	レポート ID リスト
this[int]	レポート ID リスト

メソッド:

メソッド	説明
AddReport	レポート ID を1つ追加する
ClearReports	全てのレポート ID を消去する

46.1. プロパティ

46.1.1. Count プロパティ

レポートの数

Visual C#:

```
int Count
```

46.1.2. Definition プロパティ

定義済みイベント ID

Visual C#:

```
PredefinedCEID Definition
```

46.1.3. Description プロパティ

イベントの説明

Visual C#:

```
string Description
```

46.1.4. Enable プロパティ

イベントの有効・無効

Visual C#:

```
bool Enable
```

46.1.5. ID プロパティ

イベント ID

Visual C#:

```
int ID
```

46.1.6. Reports プロパティ

レポート ID リスト

Visual C#:

```
Int32[] Reports
```

46.1.7. this[int] プロパティ

レポート ID リスト

Visual C#:

```
this[int]
```

46.2. メソッド

46.2.1. AddReport メソッド

レポートIDを1つ追加する

Visual C#:

```
public void AddReport()
```

46.2.2. ClearReports メソッド

全てのレポートIDを消去する

Visual C#:

```
public bool ClearReports()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

47. Dorian.Gem.EventModelClass.EventsClass Class

GEM イベントリスト

プロパティ:

プロパティ	説明
Count	イベントの数
this[int]	イベントオブジェクトを取得する

メソッド:

メソッド	説明
Add	イベントを追加する
Remove	イベントを削除する

47.1. プロパティ

47.1.1. Count プロパティ

イベントの数

Visual C#:

```
int Count
```

47.1.2. this[int] プロパティ

イベントオブジェクトを取得する

Visual C#:

```
this[int]
```

47.2. メソッド

47.2.1. Add メソッド

イベントを追加する

Visual C#:

```
public Event Add()
```

戻り値

47.2.2. Remove メソッド

イベントを削除する

Visual C#:

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

引数

変数名	説明
index	インデックス

戻り値

成功した場合は true が、失敗した場合は false が返ります。

48. Dorian.Gem.EventModelClass.Report Class

GEM レポート

プロパティ:

プロパティ	説明
Count	変数の数
Description	レポートの説明
ID	レポート ID
this[int]	変数 ID リスト
Variables	変数 ID リスト

メソッド:

メソッド	説明
AddVariable	VID を1つ追加する
ClearVariables	全ての VID を消去する

48.1. プロパティ

48.1.1. Count プロパティ

変数の数

Visual C#:

```
int Count
```

48.1.2. Description プロパティ

レポートの説明

Visual C#:

```
string Description
```

48.1.3. ID プロパティ

レポート ID

Visual C#:

```
int ID
```

48.1.4. this[int] プロパティ

変数 ID リスト

Visual C#:

```
this[int]
```

48.1.5. Variables プロパティ

変数 ID リスト

Visual C#:

```
Int32[] Variables
```

48.2. メソッド

48.2.1. AddVariable メソッド

VID を1つ追加する

Visual C#:

```
public void AddVariable()
```

48.2.2. ClearVariables メソッド

全ての VID を消去する

Visual C#:

```
public bool ClearVariables()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

49. Dorian.Gem.EventModelClass.ReportsClass Class

GEM レポートリスト

プロパティ:

プロパティ	説明
Count	レポートの数
this[int]	レポートオブジェクトを取得する

メソッド:

メソッド	説明
Add	レポートを追加する
Remove	レポートを削除する

49.1. プロパティ

49.1.1. Count プロパティ

レポートの数

Visual C#:

```
int Count
```

49.1.2. this[int]プロパティ

レポートオブジェクトを取得する

Visual C#:

```
this[int]
```

49.2. メソッド

49.2.1. Add メソッド

レポートを追加する

Visual C#:

```
public Report Add()
```

戻り値

49.2.2. Remove メソッド

レポートを削除する

Visual C#:

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

引数

変数名	説明
index	インデックス

戻り値

成功した場合は true が、失敗した場合は false が返ります。

50. Dorian.Gem.EventModelClass.Variable Class

GEM 変数

プロパティ:

プロパティ	説明
DefaultValue	変数のデフォルト値
Definition	定義済み変数 ID
Description	変数の説明
ID	変数 ID
Max	変数の最大値
Min	変数の最小値
SecsType	変数 SECS-II タイプ
Sml	変数の SML
Type	変数タイプ
Unit	変数の単位

50.1. プロパティ

50.1.1. DefaultValue プロパティ

変数のデフォルト値

Visual C#:

```
string DefaultValue
```

50.1.2. Definition プロパティ

定義済み変数 ID

Visual C#:

```
PredefinedVID Definition
```

50.1.3. Description プロパティ

変数の説明

Visual C#:

```
string Description
```

50.1.4. ID プロパティ

変数 ID

Visual C#:

```
int ID
```

50.1.5. Max プロパティ

変数の最大値

Visual C#:

```
string Max
```

50.1.6. Min プロパティ

変数の最小値

Visual C#:

```
string Min
```

50.1.7. SecsType プロパティ

変数 SECS-II タイプ

Visual C#:

```
SecsType SecsType
```

50.1.8. **Sml プロパティ**

変数の SML

Visual C#:

```
string Sml
```

50.1.9. **Type プロパティ**

変数タイプ

Visual C#:

```
variableType Type
```

50.1.10. **Unit プロパティ**

変数の単位

Visual C#:

```
string Unit
```

51. Dorian.Gem.EventModelClass.VariablesClass Class

GEM 変数リスト

プロパティ:

プロパティ	説明
Count	変数の数
this[int]	変数オブジェクトを取得する

メソッド:

メソッド	説明
Add	変数を追加する
Remove	変数を削除する

51.1. プロパティ

51.1.1. Count プロパティ

変数の数

Visual C#:

```
int Count
```

51.1.2. this[int]プロパティ

変数オブジェクトを取得する

Visual C#:

```
this[int]
```

51.2. メソッド

51.2.1. Add メソッド

変数を追加する

Visual C#:

```
public variable Add()
```

戻り値

51.2.2. Remove メソッド

変数を削除する

Visual C#:

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

引数

変数名	説明
index	インデックス

戻り値

成功した場合は true が、失敗した場合は false が返ります。

52. Dorian.Gem.HsmsClass Class

SEMI E37 (HSMS)の通信レイヤー。

プロパティ:

プロパティ	説明
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
Connect	サーバに接続する、またはサーバを開始する。
HandleCtrlMsg	IP v4 または IP v6
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Port	TCP ポート番号
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
Server	サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)
SessionID	セッション ID (HSMS)
SystemBytes	システムバイト
T3	T3 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー

メソッド:

メソッド	説明
Send	メッセージをサーバに送信する

52.1. プロパティ

52.1.1. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

52.1.2. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

52.1.3. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

52.1.4. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

52.1.5. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

HSMS の接続状態を取得または設定します。もし Server プロパティが false (クライアント) の時に Connect プロパティが true にセットされた場合、Dorian.Hsms コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされます。もし Server プロパティが true (サーバ) の時に Connect プロパティが true にセットされた場合、Dorian.Hsms コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成

立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。Dorian.Hsms コンポーネントが自動的に解除してくれるからです。Server プロパティが true (サーバ) の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

52.1.6. HandleCtrlMsg プロパティ

IP v4 または IP v6

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

52.1.7. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスシブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは "0.0.0.0" に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は "127.0.0.1" もしくは "localhost" を使用してください。

IP アドレスの代わりにコンピュータ名を使用することもできます。

52.1.8. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

52.1.9. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号 (例えば http サーバなどは 80) があるため、一般的には 5000 以上の値を指定すべきです。

52.1.10. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

HSMS 通信のセレクト状態を取得または設定します。

このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。

セレクト要求を受け取ると Dorian.Hsms は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にしたいくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

52.1.11. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

52.1.12. SessionID プロパティ

セッション ID (HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

52.1.13. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

52.1.14. T3 プロパティ

T3 タイマー

Visual C#:

```
double T3
```

特記事項

HSMS の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

52.1.15. T5 プロパティ

T5 タイマー

Visual C#:

```
double T5
```

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

52.1.16. T6 プロパティ

T6 タイマー

Visual C#:

```
double T6
```

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

52.1.17. T7 プロパティ

T7 タイマー

Visual C#:

```
double T7
```

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

52.1.18. T8 プロパティ

T8 タイマー

Visual C#:

```
double T8
```

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

52.2. メソッド

52.2.1. Send メソッド

メッセージをサーバに送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
message	SECS-II メッセージ

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

53. Dorian.Gem.ProblemEventArgs Class

Problem イベント引数

プロパティ:

プロパティ	説明
code	エラーコード
ip	IP アドレス
message	追加情報
port	TCP ポート番号
sender	このイベントの送信元

53.1. プロパティ

53.1.1. code プロパティ

エラーコード

Visual C#:

```
int code
```

53.1.2. ip プロパティ

IP アドレス

Visual C#:

```
string ip
```

53.1.3. message プロパティ

追加情報

Visual C#:

```
Byte[] message
```

53.1.4. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

53.1.5. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

54. Dorian.Gem.ReceivedEventArgs Class

Received イベント引数

プロパティ:

プロパティ	説明
ip	IP アドレス
message	SECS-II メッセージ
port	TCP ポート番号
sender	このイベントの送信元

54.1. プロパティ

54.1.1. ip プロパティ

IP アドレス

Visual C#:

```
string ip
```

54.1.2. message プロパティ

SECS-II メッセージ

Visual C#:

```
Byte[] message
```

54.1.3. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

54.1.4. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

55. Dorian.Gem.SecsIIClass Class

SECS-II

プロパティ:

プロパティ	説明
this[int]	SECS-II オブジェクトを取得する

55.1. プロパティ

55.1.1. this[int]プロパティ

SECS-II オブジェクトを取得する

Visual C#:

```
this[int]
```

56. Dorian.Gem.SentEventArgs Class

Sent イベント引数

プロパティ:

プロパティ	説明
ip	IP アドレス
message	SECS-II メッセージ
port	TCP ポート番号
sender	このイベントの送信元
success	成功か失敗か

56.1. プロパティ**56.1.1. ip プロパティ**

IP アドレス

Visual C#:`string ip`**56.1.2. message プロパティ**

SECS-II メッセージ

Visual C#:`Byte[] message`**56.1.3. port プロパティ**

TCP ポート番号

Visual C#:`string port`**56.1.4. sender プロパティ**

このイベントの送信元

Visual C#:`IntPtr sender`**56.1.5. success プロパティ**

成功か失敗か

Visual C#:`bool success`

57. Dorian.GemConfigDlg Class

GEM 設定ダイアログボックス

プロパティ:

プロパティ	説明
components	
gem	GEM オブジェクト

メソッド:

メソッド	説明
~Dorian.GemConfigDlg	デストラクタ
Dorian.GemConfigDlg	コンストラクタ
InitializeComponent	

57.1. プロパティ

57.1.1. components プロパティ

Required designer variable.

Visual C#:

`components`

57.1.2. gem プロパティ

GEM オブジェクト

Visual C#:

Gem `gem`

57.2. メソッド

57.2.1. ~Dorian.GemConfigDlg メソッド

デストラクタ

Visual C#:

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

引数

変数名	説明
disposing	

戻り値

57.2.2. Dorian.GemConfigDlg メソッド

コンストラクタ

Visual C#:

```
Dorian.GemConfigDlg()
```

戻り値

57.2.3. InitializeComponent メソッド

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

Visual C#:

```
InitializeComponent()
```

戻り値

58. Dorian.Hsms Class

Dorian.Hsms コンポーネントは SEMI E37 (HSMS)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。Dorian.Secsl コンポーネントと組み合わせて使います。

プロパティ:

プロパティ	説明
AutoReconnect	T5 タイムアウト後に自動的に再接続する
AutoSelectReq	自動的にセレクト要求を送る
AutoSessionID	自動的にセッション ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
Clients	接続されているクライアントのリスト(パッシブエンティティのみ)
Connect	サーバに接続する、またはサーバを開始する。
FileName	JSON 設定ファイル名
HandleCtrlMsg	自動的に HSMS 制御メッセージを処理します
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
IP	接続するリモートコンピュータの IP アドレス
IPv4	IP v4 または IP v6
Json	JSON 設定
MDLN	装置モデル名
Port	TCP ポート番号
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。
Server	サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)
SessionID	セッション ID(HSMS)
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T3	T3 タイマー
T5	T5 タイマー
T6	T6 タイマー
T7	T7 タイマー
T8	T8 タイマー
Transactions	オープンなトランザクションのリスト

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージをサーバに送信する
TimerProc	このメソッドを周期的に呼び出す必要があります。

イベント:

イベント	説明
ConnectedEvent	クライアントから接続された
DisconnectedEvent	切断された
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
SentEvent	メッセージを送信した

58.1. プロパティ

58.1.1. AutoReconnect プロパティ

T5 タイムアウト後に自動的に再接続する

Visual C#:

```
bool AutoReconnect
```

58.1.2. AutoSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool AutoSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

58.1.3. AutoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

58.1.4. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

58.1.5. Clients プロパティ

接続されているクライアントのリスト (パッシブエンティティのみ)

Visual C#:

```
ClientsClass Clients
```

58.1.6. Connect プロパティ

サーバに接続する、またはサーバを開始する。

Visual C#:

```
bool Connect
```

特記事項

HSMS の接続状態を取得または設定します。もし Server プロパティが false (クライアント) の時に Connect プロパティが true にセットされた場合、Dorian.Hsms コンポーネントは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは false にセットされます。もし Server プロパティが true (サーバ) の時に Connect プロパティが true にセットされた場合、Dorian.Hsms コンポーネントはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は true になります。

false をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。Dorian.Hsms コンポーネントが自動的に解除してくれるからです。Server プロパティが true (サーバ) の時に Connect プロパティに false をセットすると接続していた全てのクライアントとのコネクションが解消されます。

接続相手の Connect プロパティが false になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に false になるので注意が必要です。

58.1.7. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```

58.1.8. HandleCtrlMsg プロパティ

自動的に HSMS 制御メッセージを処理します

Visual C#:

```
bool HandleCtrlMsg
```

特記事項

自動的に HSMS 制御メッセージを処理します。

58.1.9. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

58.1.10. IP プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string IP
```

特記事項

HSMS 接続のパスシブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは"0.0.0.0"に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は"127.0.0.1"もしくは"localhost"を使用してください。IP アドレスの代わりにコンピュータ名を使用することもできます。

58.1.11. IPv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool IPv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

58.1.12. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

58.1.13. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

58.1.14. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号(例えば http サーバなどは 80)があるため、一般的には 5000 以上の値を指定すべきです。

58.1.15. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

HSMS 通信のセレクト状態を取得または設定します。

このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。

セレクト要求を受け取ると Dorian.Hsms は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

58.1.16. Server プロパティ

サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool Server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パッシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

58.1.17. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

58.1.18. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

58.1.19. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

58.1.20. T3 プロパティ

T3 タイマー

Visual C#:

```
double T3
```

特記事項

HSMS の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

58.1.21. T5 プロパティ

T5 タイマー

Visual C#:

```
double T5
```

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

58.1.22. T6 プロパティ

T6 タイマー

Visual C#:

```
double T6
```

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

58.1.23. T7 プロパティ

T7 タイマー

Visual C#:

```
double T7
```

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルト値は 10 秒です。

58.1.24. T8 プロパティ

T8 タイマー

Visual C#:

```
double T8
```

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルト値は 5 秒です。

58.1.25. Transactions プロパティ

オープンなトランザクションのリスト

Visual C#:

TransactionsClass Transactions

58.2. メソッド

58.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

58.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

58.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

58.2.4. Send メソッド

メッセージをサーバに送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message

SECS-II メッセージ

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

58.2.5. TimerProc メソッド

このメソッドを周期的に呼び出す必要があります。

Visual C#:

```
public void TimerProc()
```


58.3. イベント

58.3.1. ConnectedEvent イベント

クライアントから接続された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

58.3.2. DisconnectedEvent イベント

切断された

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

58.3.3. ProblemEvent イベント

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

58.3.4. ReceivedEvent イベント

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

58.3.5. SentEvent イベント

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

59. Dorian.Hsms.ClientsClass Class

接続されているクライアントのリスト(パンプエンティティのみ)

プロパティ:

プロパティ	説明
Count	接続されているクライアントの数
this[int]	クライアントにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
find	クライアントを検索する。

59.1. プロパティ

59.1.1. Count プロパティ

接続されているクライアントの数

Visual C#:

```
int Count
```

59.1.2. this[int] プロパティ

クライアントにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

59.2. メソッド

59.2.1. find メソッド

クライアントを検索する。

Visual C#:

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

引数

変数名	説明
ip	IP アドレス
port	TCP ポート番号

戻り値

60. Dorian.Hsms.ClientsClass.Client Class

接続されているクライアント

プロパティ:

プロパティ	説明
IP	IP アドレス
Port	TCP ポート番号
Selected	HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

メソッド:

メソッド	説明
Disconnect	クライアントを切断する
Send	クライアントにメッセージを送信する。

60.1. プロパティ

60.1.1. IP プロパティ

IP アドレス

Visual C#:

```
string IP
```

60.1.2. Port プロパティ

TCP ポート番号

Visual C#:

```
string Port
```

60.1.3. Selected プロパティ

HSMS 接続がセレクト要求・応答のトランザクションによってセレクトされたかどうかを返します。

Visual C#:

```
bool Selected
```

特記事項

HSMS 通信のセレクト状態を取得します。

このプロパティが false の場合、HSMS 接続はセレクトされていません。true の場合はセレクトされています。

セレクト要求を受け取ると Dorian.Hsms は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にしたいくない場合は Selected プロパティを false にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

60.2. メソッド

60.2.1. Disconnect メソッド

クライアントを切断する

Visual C#:

```
public bool Disconnect()
```

特記事項

指定された HSMS 接続を切断します。

戻り値

切断に成功した場合は true を、失敗した場合は false を返します。失敗する可能性としては、指定された IP アドレスとポート番号が間違っているか、既に切断されている場合が考えられます。

60.2.2. Send メソッド

クライアントにメッセージを送信する。

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
message	メッセージ

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

61. Dorian.Hsms.ConnectedEventArgs Class

Connected イベント引数

プロパティ:

プロパティ	説明
ip	IP アドレス
port	TCP ポート番号
sender	このイベントの送信元

61.1. プロパティ

61.1.1. ip プロパティ

IP アドレス

Visual C#:

```
string ip
```

61.1.2. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

61.1.3. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

62. Dorian.Hsms.Data Class

JSON データ

プロパティ:

プロパティ	説明
autoSessionID	自動的にセッション ID を補正する
autoSystemBytes	自動的にシステムバイトを補正する
handleCtrlMessage	自動的に HSMS 制御メッセージを処理します
sendSelectReq	自動的にセレクト要求を送る
sessionID	セッション ID (HSMS)
socket	HSMS オブジェクト
t3	T3 タイマー
t5	T5 タイマー
t6	T6 タイマー
t7	T7 タイマー
t8	T8 タイマー

62.1. プロパティ

62.1.1. autoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool autoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

62.1.2. autoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool autoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

62.1.3. handleCtrlMessage プロパティ

自動的に HSMS 制御メッセージを処理します

Visual C#:

```
bool handleCtrlMessage
```

特記事項

自動的に HSMS 制御メッセージを処理します。

62.1.4. sendSelectReq プロパティ

自動的にセレクト要求を送る

Visual C#:

```
bool sendSelectReq
```

特記事項

接続時に自動的にセレクト要求を送る。

62.1.5. sessionID プロパティ

セッション ID (HSMS)

Visual C#:

```
UInt16 sessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

62.1.6. socket プロパティ

HSMS オブジェクト

Visual C#:

```
Socket socket
```

62.1.7. t3 プロパティ

T3 タイマー

Visual C#:

```
double t3
```

特記事項

HSMS の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

62.1.8. t5 プロパティ

T5 タイマー

Visual C#:

```
double t5
```

特記事項

HSMS の T5 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

62.1.9. t6 プロパティ

T6 タイマー

Visual C#:

```
double t6
```

特記事項

HSMS の T6 タイムアウトを秒単位で取得または設定します。デフォルトは 5 秒です。

62.1.10. t7 プロパティ

T7 タイマー

Visual C#:

```
double t7
```

特記事項

HSMS の T7 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

62.1.11. t8 プロパティ

T8 タイマー

Visual C#:

```
double t8
```

特記事項

HSMS の T8 タイムアウトを秒単位で取得または設定します。デフォルトは 5 秒です。

63. Dorian.Hsms.Data.Socket Class

ソケット

プロパティ:

プロパティ	説明
ip	接続するリモートコンピュータの IP アドレス
ipv4	IP v4 または IP v6
port	TCP ポート番号
server	サーバ(パッシブエンティティ)、またはクライアント(アクティブエンティティ)

63.1. プロパティ

63.1.1. ip プロパティ

接続するリモートコンピュータの IP アドレス

Visual C#:

```
string ip
```

特記事項

HSMS 接続のパスシブエンティティのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けるため IP アドレスは "0.0.0.0" に設定します。特定のネットワークアダプタの IP アドレスに限定したい場合は、その IP アダプタでの IP アドレスを使用することもできます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は "127.0.0.1" もしくは "localhost" を使用してください。

IP アドレスの代わりにコンピュータ名を使用することもできます。

63.1.2. ipv4 プロパティ

IP v4 または IP v6

Visual C#:

```
bool ipv4
```

特記事項

IP v4 の場合は true を、IP v6 の場合は false を設定します。

63.1.3. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

特記事項

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号(例えば http サーバなどは 80)があるため、一般的には 5000 以上の値を指定すべきです。

63.1.4. server プロパティ

サーバ(パスシブエンティティ)、またはクライアント(アクティブエンティティ)

Visual C#:

```
bool server
```

特記事項

エンティティのタイプを取得または設定します。もしこのプロパティが true の場合、パスシブエンティティを意味します。もしこのプロパティが false の場合、アクティブエンティティを意味します。

64. Dorian.Hsms.DisconnectedEventArgs Class

Disconnected イベント引数

プロパティ:

プロパティ	説明
ip	IP アドレス
port	TCP ポート番号
sender	このイベントの送信元

64.1. プロパティ

64.1.1. ip プロパティ

IP アドレス

Visual C#:

```
string ip
```

64.1.2. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

64.1.3. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

65. Dorian.Hsms.ProblemEventArgs Class

Problem イベント引数

プロパティ:

プロパティ	説明
code	エラーコード
ip	IP アドレス
message	追加情報
port	TCP ポート番号
sender	このイベントの送信元

65.1. プロパティ

65.1.1. code プロパティ

エラーコード

Visual C#:

```
int code
```

65.1.2. ip プロパティ

IP アドレス

Visual C#:

```
string ip
```

65.1.3. message プロパティ

追加情報

Visual C#:

```
Byte[] message
```

65.1.4. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

65.1.5. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

66. Dorian.Hsms.ReceivedEventArgs Class

Received イベント引数

プロパティ:

プロパティ	説明
ip	IP アドレス
message	SECS-II メッセージ
port	TCP ポート番号
sender	このイベントの送信元

66.1. プロパティ

66.1.1. ip プロパティ

IP アドレス

Visual C#:

```
string ip
```

66.1.2. message プロパティ

SECS-II メッセージ

Visual C#:

```
Byte[] message
```

66.1.3. port プロパティ

TCP ポート番号

Visual C#:

```
string port
```

66.1.4. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

67. Dorian.Hsms.SentEventArgs Class

Sent イベント引数

プロパティ:

プロパティ	説明
ip	IP アドレス
message	SECS-II メッセージ
port	TCP ポート番号
sender	このイベントの送信元
success	成功か失敗か

67.1. プロパティ**67.1.1. ip プロパティ**

IP アドレス

Visual C#:`string ip`**67.1.2. message プロパティ**

SECS-II メッセージ

Visual C#:`Byte[] message`**67.1.3. port プロパティ**

TCP ポート番号

Visual C#:`string port`**67.1.4. sender プロパティ**

このイベントの送信元

Visual C#:`IntPtr sender`**67.1.5. success プロパティ**

成功か失敗か

Visual C#:`bool success`

68. Dorian.Hsms.TransactionsClass Class

オープンなトランザクションのリスト

プロパティ:

	プロパティ	説明
Count		オープンなトランザクションの数
this[int]		トランザクションにアクセスするためのシンタックスシュガー

68.1. プロパティ

68.1.1. Count プロパティ

オープンなトランザクションの数

Visual C#:

```
int Count
```

68.1.2. this[int] プロパティ

トランザクションにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

69. Dorian.Hsms.TransactionsClass.Transaction Class

オープンなトランザクション

プロパティ:

プロパティ	説明
Ctrl	HSMS 制御メッセージ
Msg	メッセージ

69.1. プロパティ

69.1.1. Ctrl プロパティ

HSMS 制御メッセージ

Visual C#:

```
bool Ctrl
```

特記事項

このプロパティが false の場合、SECS-II データメッセージです。true の場合は HSMS 制御メッセージです。

69.1.2. Msg プロパティ

メッセージ

Visual C#:

```
Byte[] Msg
```

70. Dorian.HsmsConfigDlg Class

HSMS 設定ダイアログボックス

プロパティ:

プロパティ	説明
components	
hsms	HSMS オブジェクト

メソッド:

メソッド	説明
~Dorian.HsmsConfigDlg	デストラクタ
Dorian.HsmsConfigDlg	コンストラクタ
HsmsConfigDlg_Load	Load イベント
InitializeComponent	

70.1. プロパティ

70.1.1. components プロパティ

Required designer variable.

Visual C#:

components

70.1.2. hsms プロパティ

HSMS オブジェクト

Visual C#:

Hsms hsms

70.2. メソッド

70.2.1. ~Dorian.HsmsConfigDlg メソッド

デストラクタ

Visual C#:

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

引数

変数名	説明
disposing	

戻り値

70.2.2. Dorian.HsmsConfigDlg メソッド

コンストラクタ

Visual C#:

```
Dorian.HsmsConfigDlg()
```

戻り値

70.2.3. HsmsConfigDlg_Load メソッド

Load イベント

Visual C#:

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

引数

変数名	説明
sender	送信元
e	引数

戻り値

70.2.4. InitializeComponent メソッド

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

Visual C#:

```
InitializeComponent()
```

戻り値

71. Dorian.Klarf12 Class

Dorian.Klarf12 コンポーネント は、KLARF バージョン 1.2 の機能を作成するための開発支援製品です。

プロパティ:

プロパティ	説明
Count	レコード数
this[int]	レコードにアクセスするためのシンタックスシュガー
this[string]	レコードにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12	コンストラクタ
Find	レコードを検索します
Parse	KLARF ファイルを解析します

71.1. プロパティ

71.1.1. Count プロパティ

レコード数

Visual C#:

```
int Count
```

特記事項

KLARF ファイルに含まれているレコード数を取得します。

71.1.2. this[int] プロパティ

レコードにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

71.1.3. this[string] プロパティ

レコードにアクセスするためのシンタックスシュガー

Visual C#:

```
this[string]
```

71.2. メソッド

71.2.1. Dorian.Klarf12 メソッド

コンストラクタ

Visual C#:

```
Dorian.Klarf12()
```

戻り値

71.2.2. Find メソッド

レコードを検索します

Visual C#:

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

引数

変数名	説明
record	レコード名
from	検索を開始するインデックス
forward	true = 正方向検索、false = 逆方向検索

戻り値

レコードのインデックス。見つからなかった場合は-1 が返ります。

71.2.3. Parse メソッド

KLARF ファイルを解析します

Visual C#:

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

特記事項

KLARF ファイルの内容を解析します。

引数

変数名	説明
klarfText	KLARF ファイルの内容

戻り値

成功した場合は true を、失敗した場合は false を返します。

72. Dorian.Klarf12.AlignmentImages Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.AlignmentImages	コンストラクタ

72.1. プロパティ

72.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

72.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

72.2. メソッド

72.2.1. Dorian.Klarf12.AlignmentImages メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

73. Dorian.Klarf12.AlignmentImages.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
ID	ID
NumOfImage	画像番号
X	X
Y	Y

73.1. プロパティ

73.1.1. ID プロパティ

ID

Visual C#:

```
int ID
```

73.1.2. NumOfImage プロパティ

画像番号

Visual C#:

```
int NumOfImage
```

73.1.3. X プロパティ

X

Visual C#:

```
double X
```

73.1.4. Y プロパティ

Y

Visual C#:

```
double Y
```


74. Dorian.Klarf12.AlignmentImageTransforms Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
A11	A11 (2x2 行列の左上)
A12	A12 (2x2 行列の右上)
A21	A21 (2x2 行列の左下)
A22	A21 (2x2 行列の右下)
Count	アイテム数
MarkID	アイテム

メソッド:

メソッド	説明
Dorian.Klarf12.AlignmentImageTransforms	コンストラクタ

74.1. プロパティ

74.1.1. A11 プロパティ

A11 (2x2 行列の左上)

Visual C#:

```
double A11
```

74.1.2. A12 プロパティ

A12 (2x2 行列の右上)

Visual C#:

```
double A12
```

74.1.3. A21 プロパティ

A21 (2x2 行列の左下)

Visual C#:

```
double A21
```

74.1.4. A22 プロパティ

A22 (2x2 行列の右下)

Visual C#:

```
double A22
```

74.1.5. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

74.1.6. MarkID プロパティ

アイテム

Visual C#:

```
Int32[] MarkID
```

74.2. メソッド

74.2.1. Dorian.Klarf12.AlignmentImageTransforms メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
<code>record</code>	レコード

戻り値

75. Dorian.Klarf12.AlignmentPoints Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.AlignmentPoints	コンストラクタ

75.1. プロパティ

75.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

75.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

75.2. メソッド

75.2.1. Dorian.Klarf12.AlignmentPoints メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

76. Dorian.Klarf12.AlignmentPoints.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
ID	ID
X	X
Y	Y

76.1. プロパティ

76.1.1. ID プロパティ

ID

Visual C#:

```
int ID
```

76.1.2. X プロパティ

X

Visual C#:

```
double X
```

76.1.3. Y プロパティ

Y

Visual C#:

```
double Y
```


77. Dorian.Klarf12.AreaPerTest Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.AreaPerTest	コンストラクタ

77.1. プロパティ

77.1.1. Value プロパティ

値

Visual C#:

```
double value
```

77.2. メソッド

77.2.1. Dorian.Klarf12.AreaPerTest メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

78. Dorian.Klarf12.ClassLookup Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.ClassLookup	コンストラクタ

78.1. プロパティ

78.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

78.1.2. this[int]プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

78.2. メソッド

78.2.1. Dorian.Klarf12.ClassLookup メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

79. Dorian.Klarf12.ClassLookup.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
ID	ID
Name	クラス名

79.1. プロパティ

79.1.1. ID プロパティ

ID

Visual C#:

```
int ID
```

79.1.2. Name プロパティ

クラス名

Visual C#:

```
string Name
```


80. Dorian.Klarf12.ClusterClassificationList Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.ClusterClassificationList	コンストラクタ

80.1. プロパティ

80.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

80.1.2. this[int]プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

80.2. メソッド

80.2.1. Dorian.Klarf12.ClusterClassificationList メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
<code>record</code>	レコード

戻り値

81. Dorian.Klarf12.ClusterClassificationList.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Classification	クラス番号
ID	ID

81.1. プロパティ

81.1.1. Classification プロパティ

クラス番号

Visual C#:

```
int Classification
```

81.1.2. ID プロパティ

ID

Visual C#:

```
int ID
```

82. Dorian.Klarf12.CoordinatesMirrored Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.CoordinatesMirrored	コンストラクタ

82.1. プロパティ

82.1.1. Value プロパティ

値

Visual C#:

bool value

82.2. メソッド

82.2.1. Dorian.Klarf12.CoordinatesMirrored メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
<code>record</code>	レコード

戻り値

83. Dorian.Klarf12.DatabaseAlignmentMarks Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.DatabaseAlignmentMarks	コンストラクタ

83.1. プロパティ

83.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

83.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

83.2. メソッド

83.2.1. Dorian.Klarf12.DatabaseAlignmentMarks メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

84. Dorian.Klarf12.DatabaseAlignmentMarks.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
ID	ID
OriginX	原点 X
OriginY	原点 Y
PointX	ポイント X
PointY	ポイント Y

84.1. プロパティ

84.1.1. ID プロパティ

ID

Visual C#:

```
int ID
```

84.1.2. OriginX プロパティ

原点 X

Visual C#:

```
double originX
```

84.1.3. OriginY プロパティ

原点 Y

Visual C#:

```
double originY
```

84.1.4. PointX プロパティ

ポイント X

Visual C#:

```
double PointX
```

84.1.5. PointY プロパティ

ポイント Y

Visual C#:

```
double PointY
```

85. Dorian.Klarf12.DefectClusterSetup Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.DefectClusterSetup	コンストラクタ

85.1. プロパティ

85.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

85.1.2. Value プロパティ

値

Visual C#:

```
string[] Value
```

85.2. メソッド

85.2.1. Dorian.Klarf12.DefectClusterSetup メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
<code>record</code>	レコード
<code>menu</code>	TestParametersSpec レコード

戻り値

86. Dorian.Klarf12.DefectClusterSpec Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.DefectClusterSpec	コンストラクタ

86.1. プロパティ

86.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

86.1.2. Value プロパティ

値

Visual C#:

```
string[] Value
```

86.2. メソッド

86.2.1. Dorian.Klarf12.DefectClusterSpec メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

87. Dorian.Klarf12.DefectList Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.DefectList	コンストラクタ

87.1. プロパティ

87.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

87.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

87.2. メソッド

87.2.1. Dorian.Klarf12.DefectList メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
<code>record</code>	レコード
<code>menu</code>	TestParametersSpec レコード

戻り値

88. Dorian.Klarf12.DefectList.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[string]	アイテムにアクセスするためのシンタックスシュガー
Value	値

88.1. プロパティ

88.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

88.1.2. this[string]プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[string]
```

88.1.3. Value プロパティ

値

Visual C#:

```
string[] Value
```


89. Dorian.Klarf12.DefectRecordSpec Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[string]	アイテムにアクセスするためのシンタックスシュガー
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.DefectRecordSpec	コンストラクタ

89.1. プロパティ

89.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

89.1.2. this[string] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[string]
```

89.1.3. Value プロパティ

値

Visual C#:

```
string[] Value
```

89.2. メソッド

89.2.1. Dorian.Klarf12.DefectRecordSpec メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

90. Dorian.Klarf12.DeviceID Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.DeviceID	コンストラクタ

90.1. プロパティ

90.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

90.2. メソッド

90.2.1. Dorian.Klarf12.DeviceID メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

91. Dorian.Klarf12.DieOrigin Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
X	X
Y	Y

メソッド:

メソッド	説明
Dorian.Klarf12.DieOrigin	コンストラクタ

91.1. プロパティ

91.1.1. Xプロパティ

X

Visual C#:

```
double X
```

91.1.2. Yプロパティ

Y

Visual C#:

```
double Y
```


91.2. メソッド

91.2.1. Dorian.Klarf12.DieOrigin メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

92. Dorian.Klarf12.DiePitch Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
X	X
Y	Y

メソッド:

メソッド	説明
Dorian.Klarf12.DiePitch	コンストラクタ

92.1. プロパティ

92.1.1. X プロパティ

X

Visual C#:

```
double X
```

92.1.2. Y プロパティ

Y

Visual C#:

```
double Y
```

92.2. メソッド

92.2.1. Dorian.Klarf12.DiePitch メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

93. Dorian.Klarf12.EndOfFile Class

個別クラスにアクセスするためのラッパークラスです。

メソッド:

メソッド	説明
Dorian.Klarf12.EndOfFile	コンストラクタ

93.1. メソッド

93.1.1. Dorian.Klarf12.EndOfFile メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

94. Dorian.Klarf12.FileTimestamp Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
DateTime	日時

メソッド:

メソッド	説明
Dorian.Klarf12.FileTimestamp	コンストラクタ

94.1. プロパティ

94.1.1. DateTime プロパティ

日時

Visual C#:

DateTime **DateTime**

94.2. メソッド

94.2.1. Dorian.Klarf12.FileTimestamp メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

95. Dorian.Klarf12.FileVersion Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Major	メジャーバージョン番号
Minor	マイナーバージョン番号

メソッド:

メソッド	説明
Dorian.Klarf12.FileVersion	コンストラクタ

95.1. プロパティ

95.1.1. Major プロパティ

メジャーバージョン番号

Visual C#:

```
int Major
```

95.1.2. Minor プロパティ

マイナーバージョン番号

Visual C#:

```
int Minor
```

95.2. メソッド

95.2.1. Dorian.Klarf12.FileVersion メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

96. Dorian.Klarf12.InspectedArea Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.InspectedArea	コンストラクタ

96.1. プロパティ

96.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

96.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

96.2. メソッド

96.2.1. Dorian.Klarf12.InspectedArea メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

97. Dorian.Klarf12.InspectedArea.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
OffsetX	オフセット X
OffsetY	オフセット Y
PitchX	ピッチ X
PitchY	ピッチ Y
RepeatX	リピート X
RepeatY	リピート Y
SizeX	サイズ X
SizeY	サイズ Y

97.1. プロパティ**97.1.1. OffsetX プロパティ**

オフセット X

Visual C#:`double OffsetX`**97.1.2. OffsetY プロパティ**

オフセット Y

Visual C#:`double OffsetY`**97.1.3. PitchX プロパティ**

ピッチ X

Visual C#:`double PitchX`**97.1.4. PitchY プロパティ**

ピッチ Y

Visual C#:`double PitchY`**97.1.5. RepeatX プロパティ**

リピート X

Visual C#:`int RepeatX`**97.1.6. RepeatY プロパティ**

リピート Y

Visual C#:`int RepeatY`**97.1.7. SizeX プロパティ**

サイズ X

Visual C#:

double **SizeX**

97.1.8. **SizeY プロパティ**

サイズY

Visual C#:

double **SizeY**

98. Dorian.Klarf12.InspectedAreaOrigin Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
X	X
Y	Y

メソッド:

メソッド	説明
Dorian.Klarf12.InspectedAreaOrigin	コンストラクタ

98.1. プロパティ

98.1.1. Xプロパティ

X

Visual C#:

double X

98.1.2. Yプロパティ

Y

Visual C#:

double Y

98.2. メソッド

98.2.1. Dorian.Klarf12.InspectedAreaOrigin メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

99. Dorian.Klarf12.InspectionOrientation Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.InspectionOrientation	コンストラクタ

99.1. プロパティ

99.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

99.2. メソッド

99.2.1. Dorian.Klarf12.InspectionOrientation メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

100. Dorian.Klarf12.InspectionStationID Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
ID	装置 ID
Manufacturer	メーカー
Model	型番

メソッド:

メソッド	説明
Dorian.Klarf12.InspectionStationID	コンストラクタ

100.1. プロパティ

100.1.1. ID プロパティ

装置 ID

Visual C#:

```
string ID
```

100.1.2. Manufacturer プロパティ

メーカー

Visual C#:

```
string Manufacturer
```

100.1.3. Model プロパティ

型番

Visual C#:

```
string Model
```

100.2. メソッド

100.2.1. Dorian.Klarf12.InspectionStationID メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

101. Dorian.Klarf12.InspectionTest Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
ID	ID

メソッド:

メソッド	説明
Dorian.Klarf12.InspectionTest	コンストラクタ

101.1. プロパティ

101.1.1. ID プロパティ

ID

Visual C#:

```
int ID
```

101.2. メソッド

101.2.1. Dorian.Klarf12.InspectionTest メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

102. Dorian.Klarf12.LotID Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.LotID	コンストラクタ

102.1. プロパティ

102.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```


102.2. メソッド

102.2.1. Dorian.Klarf12.LotID メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

103. Dorian.Klarf12.LotStatus Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Failed	失敗数
Passed	成功数
Total	合計数

メソッド:

メソッド	説明
Dorian.Klarf12.LotStatus	コンストラクタ

103.1. プロパティ

103.1.1. Failed プロパティ

失敗数

Visual C#:

```
int Failed
```

103.1.2. Passed プロパティ

成功数

Visual C#:

```
int Passed
```

103.1.3. Total プロパティ

合計数

Visual C#:

```
int Total
```

103.2. メソッド

103.2.1. Dorian.Klarf12.LotStatus メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

104. Dorian.Klarf12.OrientationInstructions Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.OrientationInstructions	コンストラクタ

104.1. プロパティ

104.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

104.2. メソッド

104.2.1. Dorian.Klarf12.OrientationInstructions メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

105. Dorian.Klarf12.OrientationMarkLocation Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.OrientationMarkLocation	コンストラクタ

105.1. プロパティ

105.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

105.2. メソッド

105.2.1. Dorian.Klarf12.OrientationMarkLocation メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
<code>record</code>	レコード

戻り値

106. Dorian.Klarf12.Record Class

レコードデータ

プロパティ:

プロパティ	説明
Count	アイテム数
Name	名前
this[int]	アイテム

106.1. プロパティ

106.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

106.1.2. Name プロパティ

名前

Visual C#:

```
string Name
```

106.1.3. this[int] プロパティ

アイテム

Visual C#:

```
this[int]
```

107. Dorian.Klarf12.RecordBase Class

個別クラスにアクセスするためのラッパークラスのベースクラスです。

プロパティ:

プロパティ	説明
Available	このオブジェクトが使用できるか

107.1. プロパティ

107.1.1. Available プロパティ

このオブジェクトが利用できるか

Visual C#:

```
bool Available
```

108. Dorian.Klarf12.RemovedDieList Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.RemovedDieList	コンストラクタ

108.1. プロパティ

108.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

108.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```


108.2. メソッド

108.2.1. Dorian.Klarf12.RemovedDieList メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

109. Dorian.Klarf12.RemovedDieList.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
X	X
Y	Y

109.1. プロパティ

109.1.1. X プロパティ

X

Visual C#:

```
int X
```

109.1.2. Y プロパティ

Y

Visual C#:

```
int Y
```

110. Dorian.Klarf12.ResultsID Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.ResultsID	コンストラクタ

110.1. プロパティ

110.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

110.2. メソッド

110.2.1. Dorian.Klarf12.ResultsID メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

111. Dorian.Klarf12.ResultTimestamp Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
DateTime	日時

メソッド:

メソッド	説明
Dorian.Klarf12.ResultTimestamp	コンストラクタ

111.1. プロパティ

111.1.1. DateTime プロパティ

日時

Visual C#:

DateTime **DateTime**

111.2. メソッド

111.2.1. Dorian.Klarf12.ResultTimestamp メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

112. Dorian.Klarf12.SampleCenterLocation Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
X	X
Y	Y

メソッド:

メソッド	説明
Dorian.Klarf12.SampleCenterLocation	コンストラクタ

112.1. プロパティ

112.1.1. X プロパティ

X

Visual C#:

```
double X
```

112.1.2. Y プロパティ

Y

Visual C#:

```
double Y
```

112.2. メソッド

112.2.1. Dorian.Klarf12.SampleCenterLocation メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

113. Dorian.Klarf12.SampleDieMap Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.SampleDieMap	コンストラクタ

113.1. プロパティ

113.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

113.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

113.2. メソッド

113.2.1. Dorian.Klarf12.SampleDieMap メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

114. Dorian.Klarf12.SampleDieMap.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
X	X
Y	Y

114.1. プロパティ

114.1.1. X プロパティ

X

Visual C#:

```
int X
```

114.1.2. Y プロパティ

Y

Visual C#:

```
int Y
```

115. Dorian.Klarf12.SampleOrientationMarkType Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.SampleOrientationMarkType	コンストラクタ

115.1. プロパティ

115.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

115.2. メソッド

115.2.1. Dorian.Klarf12.SampleOrientationMarkType メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

116. Dorian.Klarf12.SampleSize Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.SampleSize	コンストラクタ

116.1. プロパティ

116.1.1. Value プロパティ

値

Visual C#:

```
Int32[] Value
```

116.2. メソッド

116.2.1. Dorian.Klarf12.SampleSize メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

117. Dorian.Klarf12.SampleTestPlan Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.SampleTestPlan	コンストラクタ

117.1. プロパティ

117.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

117.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

117.2. メソッド

117.2.1. Dorian.Klarf12.SampleTestPlan メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

118. Dorian.Klarf12.SampleTestPlan.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
X	X
Y	Y

118.1. プロパティ

118.1.1. X プロパティ

X

Visual C#:

```
int X
```

118.1.2. Y プロパティ

Y

Visual C#:

```
int Y
```

119. Dorian.Klarf12.SampleTestReferencePlan Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.SampleTestReferencePlan	コンストラクタ

119.1. プロパティ

119.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

119.1.2. this[int]プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

119.2. メソッド

119.2.1. Dorian.Klarf12.SampleTestReferencePlan メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

120. Dorian.Klarf12.SampleTestReferencePlan.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
ComparedX	比較 X
ComparedY	比較 Y
SampledX	サンプル X
SampledY	サンプル Y

120.1. プロパティ

120.1.1. ComparedX プロパティ

比較 X

Visual C#:

```
int ComparedX
```

120.1.2. ComparedY プロパティ

比較 Y

Visual C#:

```
int ComparedY
```

120.1.3. SampledX プロパティ

サンプル X

Visual C#:

```
int SampledX
```

120.1.4. SampledY プロパティ

サンプル Y

Visual C#:

```
int SampledY
```

121. Dorian.Klarf12.SampleType Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.SampleType	コンストラクタ

121.1. プロパティ

121.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

121.2. メソッド

121.2.1. Dorian.Klarf12.SampleType メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

122. Dorian.Klarf12.SetupID Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
DateTime	日時
ID	ID

メソッド:

メソッド	説明
Dorian.Klarf12.SetupID	コンストラクタ

122.1. プロパティ

122.1.1. DateTime プロパティ

日時

Visual C#:

```
DateTime DateTime
```

122.1.2. ID プロパティ

ID

Visual C#:

```
string ID
```

122.2. メソッド

122.2.1. Dorian.Klarf12.SetupID メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

123. Dorian.Klarf12.Slot Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.Slot	コンストラクタ

123.1. プロパティ

123.1.1. Value プロパティ

値

Visual C#:

```
int Value
```

123.2. メソッド

123.2.1. Dorian.Klarf12.Slot メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

124. Dorian.Klarf12.StepID Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.StepID	コンストラクタ

124.1. プロパティ

124.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

124.2. メソッド

124.2.1. Dorian.Klarf12.StepID メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

125. Dorian.Klarf12.SummaryList Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
this[int]	アイテムにアクセスするためのシンタックスシュガー

メソッド:

メソッド	説明
Dorian.Klarf12.SummaryList	コンストラクタ

125.1. プロパティ

125.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

125.1.2. this[int] プロパティ

アイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

125.2. メソッド

125.2.1. Dorian.Klarf12.SummaryList メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
<code>record</code>	レコード
<code>menu</code>	TestParametersSpec レコード

戻り値

126. Dorian.Klarf12.SummaryList.DataItem Class

個別アイテムにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
Value	値

126.1. プロパティ

126.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

126.1.2. Value プロパティ

値

Visual C#:

```
string[] Value
```

127. Dorian.Klarf12.SummarySpec Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.SummarySpec	コンストラクタ

127.1. プロパティ

127.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

127.1.2. Value プロパティ

値

Visual C#:

```
string[] Value
```

127.2. メソッド

127.2.1. Dorian.Klarf12.SummarySpec メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

128. Dorian.Klarf12.TestParametersList Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.TestParametersList	コンストラクタ

128.1. プロパティ

128.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

128.1.2. Value プロパティ

値

Visual C#:

```
string[] Value
```

128.2. メソッド

128.2.1. Dorian.Klarf12.TestParametersList メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
<code>record</code>	レコード
<code>menu</code>	TestParametersSpec レコード

戻り値

129. Dorian.Klarf12.TestParametersSpec Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Count	アイテム数
Value	値

メソッド:

メソッド	説明
Dorian.Klarf12.TestParametersSpec	コンストラクタ

129.1. プロパティ

129.1.1. Count プロパティ

アイテム数

Visual C#:

```
int Count
```

129.1.2. Value プロパティ

値

Visual C#:

```
string[] Value
```

129.2. メソッド

129.2.1. Dorian.Klarf12.TestParametersSpec メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

130. Dorian.Klarf12.TiffFileName Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.TiffFileName	コンストラクタ

130.1. プロパティ

130.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

130.2. メソッド

130.2.1. Dorian.Klarf12.TiffFileName メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

131. Dorian.Klarf12.TiffSpec Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Alignment	TIFF クラスで用いられるアライメント画像
Defect	TIFF クラスで用いられる欠陥画像
Version	TIFF バージョン

メソッド:

メソッド	説明
Dorian.Klarf12.TiffSpec	コンストラクタ

131.1. プロパティ

131.1.1. Alignment プロパティ

TIFF クラスで用いられるアライメント画像

Visual C#:

```
string Alignment
```

131.1.2. Defect プロパティ

TIFF クラスで用いられる欠陥画像

Visual C#:

```
string Defect
```

131.1.3. Version プロパティ

TIFF バージョン

Visual C#:

```
string Version
```


131.2. メソッド

131.2.1. Dorian.Klarf12.TiffSpec メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

132. Dorian.Klarf12.WaferID Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.WaferID	コンストラクタ

132.1. プロパティ

132.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

132.2. メソッド

132.2.1. Dorian.Klarf12.WaferID メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

133. Dorian.Klarf12.WaferStatus Class

個別クラスにアクセスするためのラッパークラスです。

プロパティ:

プロパティ	説明
Text	文字列

メソッド:

メソッド	説明
Dorian.Klarf12.WaferStatus	コンストラクタ

133.1. プロパティ

133.1.1. Text プロパティ

文字列

Visual C#:

```
string Text
```

133.2. メソッド

133.2.1. Dorian.Klarf12.WaferStatus メソッド

コンストラクタ

Visual C#:

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

引数

変数名	説明
record	レコード

戻り値

134. Dorian.Klarf18 Class

Dorian.Klarf18 コンポーネント は、KLARF バージョン 1.8 の機能を作成するための開発支援製品です。

プロパティ:

プロパティ	説明
Json	JSON 設定

メソッド:

メソッド	説明
Parse	KLARF ファイルを解析します

134.1. プロパティ

134.1.1. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

134.2. メソッド

134.2.1. Parse メソッド

KLARF ファイルを解析します

Visual C#:

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

特記事項

KLARF ファイルの内容を解析します。

引数

変数名	説明
<code>klarfText</code>	KLARF ファイルの内容

戻り値

成功した場合は true を、失敗した場合は false を返します。

135. Dorian.Log Class

Dorian.Log はログファイルのコンポーネントです。

プロパティ:

プロパティ	説明
Count	バックアップファイルの数
Enable	ログ機能の有効・無効
FileName	ログファイル名
FileSize	ログファイルのサイズ
Json	JSON 設定
JsonFileName	JSON 設定ファイル名
TimeStamp	タイムスタンプの有効・無効

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Write	文字列を書き込みます

135.1. プロパティ

135.1.1. Count プロパティ

バックアップファイルの数

Visual C#:

```
int Count
```

特記事項

バックアップファイルの数を取得または設定します。

135.1.2. Enable プロパティ

ログ機能の有効・無効

Visual C#:

```
bool Enable
```

特記事項

ログ機能の有効・無効を取得または設定します。

135.1.3. FileName プロパティ

ログファイル名

Visual C#:

```
string FileName
```

特記事項

ログファイル名を取得または設定します。

135.1.4. FileSize プロパティ

ログファイルのサイズ

Visual C#:

```
int FileSize
```

特記事項

ログファイルのサイズを取得または設定します。

135.1.5. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

135.1.6. JsonFileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string JsonFileName
```

135.1.7. TimeStamp プロパティ

タイムスタンプの有効・無効

Visual C#:

```
bool TimeStamp
```

特記事項

タイムスタンプの有効・無効を取得または設定します。

135.2. メソッド

135.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

135.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

135.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

135.2.4. Write メソッド

文字列を書き込みます

Visual C#:

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

特記事項

文字列をログファイルに書き込みます。

引数

変数名	説明
-----	----

value

文字列

136. Dorian.LogConfigDlg Class

ログ設定ダイアログボックス

プロパティ:

プロパティ	説明
components	
log	Log オブジェクト

メソッド:

メソッド	説明
~Dorian.LogConfigDlg	デストラクタ
Dorian.LogConfigDlg	コンストラクタ
InitializeComponent	
LogConfigDlg_Load	Load イベント

136.1. プロパティ

136.1.1. components プロパティ

Required designer variable.

Visual C#:

components

136.1.2. log プロパティ

Log オブジェクト

Visual C#:

Log log

136.2. メソッド

136.2.1. ~Dorian.LogConfigDlg メソッド

デストラクタ

Visual C#:

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

引数

変数名	説明
disposing	

戻り値

136.2.2. Dorian.LogConfigDlg メソッド

コンストラクタ

Visual C#:

```
Dorian.LogConfigDlg()
```

戻り値

136.2.3. InitializeComponent メソッド

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

Visual C#:

```
InitializeComponent()
```

戻り値

136.2.4. LogConfigDlg_Load メソッド

Load イベント

Visual C#:

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

引数

変数名	説明
sender	送信元
e	引数

戻り値

137. Dorian.Properties.Resources Class

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

プロパティ:

プロパティ	説明
Add	
Banner	
Culture	
Gear	
Play	
Remove	
ResourceManager	
Stop	

137.1. プロパティ

137.1.1. Add プロパティ

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

Visual C#:

```
Add
```

137.1.2. Banner プロパティ

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

Visual C#:

```
Banner
```

137.1.3. Culture プロパティ

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

Visual C#:

```
Culture
```

137.1.4. Gear プロパティ

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

Visual C#:

```
Gear
```

137.1.5. Play プロパティ

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

Visual C#:

```
Play
```

137.1.6. Remove プロパティ

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

Visual C#:

```
Remove
```

137.1.7. ResourceManager プロパティ

Returns the cached ResourceManager instance used by this class.

Visual C#:

ResourceManager

137.1.8. Stop プロパティ

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

Visual C#:

Stop

138. Dorian.SecsI Class

Dorian.SecsI は SEMI E.4(SECS-I) 通信用のコンポーネントです。

プロパティ:

プロパティ	説明
AutoSessionID	自動的にデバイス ID を補正する
AutoSystemBytes	自動的にシステムバイトを補正する
BaudRate	ボーレート
Connect	通信ポートを開く・閉じる
FileName	JSON 設定ファイル名
Heartbeat	S1F1 の送信周期(秒)。0 以下は無効。
Json	JSON 設定
Master	マスターカスレーブか
MDLN	装置モデル名
Port	シリアルポート番号
Retry	リトライ回数
SessionID	デバイス ID
SOFTREV	装置ソフトバージョン
SystemBytes	システムバイト
T1	T1 タイマー
T2	T2 タイマー
T3	T3 タイマー
T4	T4 タイマー

メソッド:

メソッド	説明
Config	設定画面を表示します
Load	JSON 設定ファイルを読み込みます。
Save	JSON 設定ファイルを書き込みます。
Send	メッセージを送信する
TimerProc	このメソッドを周期的に呼び出す必要があります。

イベント:

イベント	説明
ProblemEvent	エラーが発生した
ReceivedEvent	メッセージを受信した
ReceivedRawEvent	文字データを受信した
SentEvent	メッセージを送信した
SentRawEvent	文字データを送信した

138.1. プロパティ

138.1.1. AutoSessionID プロパティ

自動的にデバイス ID を補正する

Visual C#:

```
bool AutoSessionID
```

特記事項

送信時に自動的にデバイス ID を補正する。

138.1.2. AutoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool AutoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

138.1.3. BaudRate プロパティ

ボーレート

Visual C#:

```
int BaudRate
```

特記事項

ボーレートを取得または設定します。

138.1.4. Connect プロパティ

通信ポートを開く・閉じる

Visual C#:

```
bool Connect
```

特記事項

SECS-I の接続状態を取得または設定します。

138.1.5. FileName プロパティ

JSON 設定ファイル名

Visual C#:

```
string FileName
```


138.1.6. Heartbeat プロパティ

S1F1 の送信周期(秒)。0 以下は無効。

Visual C#:

```
int Heartbeat
```

138.1.7. Json プロパティ

JSON 設定

Visual C#:

```
string Json
```

138.1.8. Master プロパティ

マスターかスレーブか

Visual C#:

```
bool Master
```

特記事項

マスターかスレーブかを取得または設定します。

138.1.9. MDLN プロパティ

装置モデル名

Visual C#:

```
string MDLN
```

138.1.10. Port プロパティ

シリアルポート番号

Visual C#:

```
int Port
```

特記事項

通信ポート番号を取得または設定します。

138.1.11. Retry プロパティ

リトライ回数

Visual C#:

```
int Retry
```

特記事項

リトライ回数を取得または設定します。

138.1.12. SessionID プロパティ

デバイス ID

Visual C#:

```
UInt16 SessionID
```

特記事項

デバイス ID を取得または設定します。

138.1.13. SOFTREV プロパティ

装置ソフトバージョン

Visual C#:

```
string SOFTREV
```

138.1.14. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

138.1.15. T1 プロパティ

T1 タイマー

Visual C#:

```
double T1
```

特記事項

SECS-I の T1 タイムアウトを秒単位で取得または設定します。デフォルトは 1 秒です。

138.1.16. T2 プロパティ

T2 タイマー

Visual C#:

```
double T2
```

特記事項

SECS-I の T2 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

138.1.17. T3 プロパティ

T3 タイマー

Visual C#:

```
double T3
```

特記事項

SECS-I の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

138.1.18. T4 プロパティ

T4 タイマー

Visual C#:

```
double T4
```

特記事項

SECS-I の T4 タイムアウトを秒単位で取得または設定します。デフォルトは 60 秒です。

138.2. メソッド

138.2.1. Config メソッド

設定画面を表示します

Visual C#:

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

引数

変数名	説明
caption	キャプションタイトル

戻り値

成功した場合は true が、失敗した場合は false が返ります。

138.2.2. Load メソッド

JSON 設定ファイルを読み込みます。

Visual C#:

```
public bool Load()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

138.2.3. Save メソッド

JSON 設定ファイルを書き込みます。

Visual C#:

```
public bool Save()
```

戻り値

成功した場合は true が、失敗した場合は false が返ります。

138.2.4. Send メソッド

メッセージを送信する

Visual C#:

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

特記事項

指定されたメッセージを送信します。

引数

変数名	説明
-----	----

message	SECS-II メッセージ
---------	---------------

戻り値

正しく送信できた場合は true が、失敗した場合は false が返ります。

138.2.5. TimerProc メソッド

このメソッドを周期的に呼び出す必要があります。

Visual C#:

```
public void TimerProc()
```

138.3. イベント

138.3.1. ProblemEvent イベント

エラーが発生した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

138.3.2. ReceivedEvent イベント

メッセージを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

138.3.3. ReceivedRawEvent イベント

文字データを受信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

138.3.4. SentEvent イベント

メッセージを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

138.3.5. SentRawEvent イベント

文字データを送信した

Visual C#:

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

引数

変数名	説明
sender	送信元
e	付帯情報

139. Dorian.Secsl.Data Class

JSON データ

プロパティ:

プロパティ	説明
autoSessionID	自動的にセッション ID を補正する
autoSystemBytes	自動的にシステムバイトを補正する
baudRate	ボーレート
master	マスターかスレーブか
port	シリアルポート番号
retry	リトライ回数
sessionID	セッション ID (HSMS)
t1	T1 タイマー
t2	T2 タイマー
t3	T3 タイマー
t4	T4 タイマー

139.1. プロパティ

139.1.1. autoSessionID プロパティ

自動的にセッション ID を補正する

Visual C#:

```
bool autoSessionID
```

特記事項

送信時に自動的にセッション ID を補正する。

139.1.2. autoSystemBytes プロパティ

自動的にシステムバイトを補正する

Visual C#:

```
bool autoSystemBytes
```

特記事項

送信時に自動的にシステムバイトを補正する。

139.1.3. baudRate プロパティ

ボーレート

Visual C#:

```
int baudRate
```

特記事項

ボーレートを取得または設定します。

139.1.4. master プロパティ

マスターかスレーブか

Visual C#:

```
bool master
```

特記事項

マスターかスレーブかを取得または設定します。

139.1.5. port プロパティ

シリアルポート番号

Visual C#:

```
int port
```

特記事項

通信ポート番号を取得または設定します。

139.1.6. retry プロパティ

リトライ回数

Visual C#:

```
int retry
```

特記事項

リトライ回数を取得または設定します。

139.1.7. sessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 sessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

139.1.8. t1 プロパティ

T1 タイマー

Visual C#:

```
double t1
```

特記事項

HSMS の T1 タイムアウトを秒単位で取得または設定します。デフォルトは 1 秒です。

139.1.9. t2 プロパティ

T2 タイマー

Visual C#:

```
double t2
```

特記事項

HSMS の T2 タイムアウトを秒単位で取得または設定します。デフォルトは 10 秒です。

139.1.10. t3 プロパティ

T3 タイマー

Visual C#:

```
double t3
```

特記事項

HSMS の T3 タイムアウトを秒単位で取得または設定します。デフォルトは 45 秒です。

139.1.11. t4 プロパティ

T4 タイマー

Visual C#:

```
double t4
```

特記事項

HSMS の T4 タイムアウトを秒単位で取得または設定します。デフォルトは 60 秒です。

140. Dorian.Secsl.ProblemEventArgs Class

Problem イベント引数

プロパティ:

プロパティ	説明
code	エラーコード
message	SECS-II メッセージ
sender	このイベントの送信元

140.1. プロパティ

140.1.1. code プロパティ

エラーコード

Visual C#:

```
int code
```

140.1.2. message プロパティ

SECS-II メッセージ

Visual C#:

```
Byte[] message
```

140.1.3. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

141. Dorian.SecsI.ReceivedEventArgs Class

Received イベント引数

プロパティ:

プロパティ	説明
message	SECS-II メッセージ
sender	このイベントの送信元

141.1. プロパティ

141.1.1. message プロパティ

SECS-II メッセージ

Visual C#:

```
Byte[] message
```

141.1.2. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

142. Dorian.Secsl.ReceivedRawEventArgs Class

ReceivedRaw イベント引数

プロパティ:

プロパティ	説明
data	文字データ
sender	このイベントの送信元

142.1. プロパティ

142.1.1. data プロパティ

文字データ

Visual C#:

Byte **data**

142.1.2. sender プロパティ

このイベントの送信元

Visual C#:

IntPtr **sender**

143. Dorian.Secsl.SentEventArgs Class

Sent イベント引数

プロパティ:

プロパティ	説明
message	SECS-II メッセージ
sender	このイベントの送信元
success	成功か失敗か

143.1. プロパティ

143.1.1. message プロパティ

SECS-II メッセージ

Visual C#:

```
Byte[] message
```

143.1.2. sender プロパティ

このイベントの送信元

Visual C#:

```
IntPtr sender
```

143.1.3. success プロパティ

成功か失敗か

Visual C#:

```
bool success
```

144. Dorian.Secsl.SentRawEventArgs Class

SentRaw イベント引数

プロパティ:

プロパティ	説明
<code>data</code>	文字データ
<code>sender</code>	このイベントの送信元

144.1. プロパティ

144.1.1. data プロパティ

文字データ

Visual C#:

Byte **data**

144.1.2. sender プロパティ

このイベントの送信元

Visual C#:

IntPtr **sender**

145. Dorian.SecslConfigDlg Class

SECS-I 設定ダイアログボックス

プロパティ:

プロパティ	説明
components	
secs	Secsl オブジェクト

メソッド:

メソッド	説明
~Dorian.SecslConfigDlg	デストラクタ
Dorian.SecslConfigDlg	コンストラクタ
InitializeComponent	
SecslConfigDlg_Load	Load イベント

145.1. プロパティ

145.1.1. components プロパティ

Required designer variable.

Visual C#:

`components`

145.1.2. secs プロパティ

SecsI オブジェクト

Visual C#:

`SecsI secs`

145.2. メソッド

145.2.1. ~Dorian.SecsIConfigDlg メソッド

デストラクタ

Visual C#:

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

引数

変数名	説明
disposing	

戻り値

Dorian.SecsIConfigDlg メソッド

コンストラクタ

Visual C#:

```
Dorian.SecsIConfigDlg()
```

戻り値

145.2.2. InitializeComponent メソッド

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

Visual C#:

```
InitializeComponent()
```

戻り値

145.2.3. SecsIConfigDlg_Load メソッド

Load イベント

Visual C#:

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

引数

変数名	説明
sender	送信元
e	引数

戻り値

146. Dorian.SecsII Class

Dorian.SecsII コンポーネント は、SEMI E5 (SECS-II)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。

Dorian.Hsms コンポーネントや Dorian.SecsI コンポーネントなどと組み合わせて使います。

プロパティ:

プロパティ	説明
BlockNumber	ブロック番号 (SECS-I)
DeviceID	デバイス ID (SECS-I)
Ebit	エンドビット
Error	SML のコンパイルに失敗したか
Function	ファンクション番号
Host	ホストまたは装置
Hsms	HSMS または SECS-I
Json	JSON
Msg	メッセージデータ (バイナリ)
Node	ノード
NodeCount	ノードの要素数
NodeItem	ノードアイテム
NodeType	ノードの型
NodeValue	ノードの値
NodeValueHex	ノードの値 (16 進数)
PType	P タイプ (HSMS)
Rbit	リバースビット (SECS-I)
SessionID	セッション ID (HSMS)
Sml	SML
SmlType	SML 書式 (作成中)
SourceID	ソース ID (SECS-I)
Stream	ストリーム番号
SType	S タイプ (HSMS)
SuggestedReplyMsg	推奨される返信メッセージ
SystemBytes	システムバイト
this[string]	指定したノードの値を取得します
TransactionID	トランザクション ID (SECS-I)
Verification	認証結果
Wbit	ウェイトビット

メソッド:

メソッド	説明
Reply	返信用にヘッダをセットします
Reply	返信用にヘッダをセットします
Reset	メッセージをリセットします

146.1. プロパティ

146.1.1. BlockNumber プロパティ

ブロック番号 (SECS-I)

Visual C#:

```
UInt16 BlockNumber
```

特記事項

ブロック番号を取得または設定します。SECS-I でのみ使用されます。

SECS-I の受信メッセージで BlockNumber が 1 以外の場合は、マルチブロックメッセージだったことを示します。送信時は必ず 1 にセットします。もしメッセージのサイズが大きくて 1 ブロックに収まりきらない場合は、自動的にマルチブロックにして送信されます。

146.1.2. DeviceID プロパティ

デバイス ID (SECS-I)

Visual C#:

```
UInt16 DeviceID
```

特記事項

デバイス ID を取得または設定します。デバイス ID は SECS-II ヘッダの先頭 2 ビット目から 15 ビットです。

デバイス ID とセッション ID はほぼ同じですが、デバイス ID は 15 ビット、セッション ID は 16 ビットです。

146.1.3. Ebit プロパティ

エンドビット

Visual C#:

```
bool Ebit
```

特記事項

エンドビットを取得または設定します。SECS-I でのみ使用されます。

SECS-I の受信メッセージのエンドビットは常に True となります。これは Dorian.SecsI コンポーネントが最終ブロックまで受信してから Received イベントを発生させるからです。

146.1.4. Error プロパティ

SML のコンパイルに失敗したか

Visual C#:

```
bool Error
```

特記事項

SML 文字列の処理でエラーが発生したかどうかを取得します。

読み出し専用プロパティ。

146.1.5. Function プロパティ

ファンクション番号

Visual C#:

```
Byte Function
```

特記事項

ファンクション番号を取得または設定します。

146.1.6. Host プロパティ

ホストまたは装置

Visual C#:

```
bool Host
```

特記事項

このメッセージを送信してきた側がホストなら true、装置なら false をセットします。Dorian.SecsII コンポーネントの役割を取得または設定します。このプロパティは Verification プロパティを使ってメッセージの構造を検証する場合と、検証の結果として SuggestedReplyMsg プロパティにセットされるメッセージにのみ影響します。

146.1.7. Hsms プロパティ

HSMS または SECS-I

Visual C#:

```
bool Hsms
```

特記事項

HSMS か SECS-I かを取得または設定します。デフォルトは HSMS です。

146.1.8. Json プロパティ

JSON

Visual C#:

```
string Json
```

特記事項

SECS-II メッセージを JSON 文字列として取得または設定します。Json プロパティを読み出すとコントロールにセットされているメッセージの構造をツリー形式の JSON 文字列で取得します。詳細については付録 C を参照のこと。

146.1.9. Msg プロパティ

メッセージデータ(バイナリ)

Visual C#:

```
Byte[] Msg
```

特記事項

メッセージデータを取得または設定します。

146.1.10. Node プロパティ

ノード

Visual C#:

```
string Node
```

特記事項

メッセージ内容进行操作するためのノードを取得または設定します。ノードは"/" (スラッシュ)とノード番号と"["、"]" (カギカッコ)で構成されます。ノード番号は 1 から始まる数字です。ノードが空文字列の場合はルートが指定されたときみなされます。

146.1.11. NodeCount プロパティ

ノードの要素数

Visual C#:

```
int NodeCount
```

特記事項

サブアイテムの個数を取得します。もしノードプロパティがリスト型の場合、このプロパティはサブノードの数を意味します。それ以外は配列の個数です。

読み出し専用プロパティ。

146.1.12. NodeItem プロパティ

ノードアイテム

Visual C#:

```
NodeItemClass NodeItem
```

特記事項

メッセージ内容进行操作するためのノードを取得します。

146.1.13. NodeType プロパティ

ノードの型

Visual C#:

```
SecsType NodeType
```

特記事項

ノードタイプを取得します。

読み出し専用プロパティ。

146.1.14. NodeValue プロパティ

ノードの値

Visual C#:

```
string NodeValue
```

特記事項

ノードの値を取得します。もしノードが数値型の場合、値は 10 進法文字列に変換されます。
読み出し専用プロパティ。

146.1.15. NodeValueHex プロパティ

ノードの値(16 進数)

Visual C#:

```
string NodeValueHex
```

特記事項

ノードの値を 16 進法表現で取得します。
読み出し専用プロパティ。

146.1.16. PType プロパティ

P タイプ(HSMS)

Visual C#:

```
Byte PType
```

特記事項

プレゼンテーションタイプを取得または設定します。
SECS-II メッセージを使用しますので、0 を指定します。SEMI E.5 では現在は SECS-II のみが規定されているため、PType プロパティは常に 0 です。

146.1.17. Rbit プロパティ

リバースビット(SECS-I)

Visual C#:

```
bool Rbit
```

特記事項

リバースビットを取得または設定します。

146.1.18. SessionID プロパティ

セッション ID(HSMS)

Visual C#:

```
UInt16 SessionID
```

特記事項

HSMS のセッション ID を取得または設定します。セッション ID は SECS-II ヘッダの先頭 16 ビットです。

146.1.19. Sml プロパティ

SML

Visual C#:

```
string SmI
```

特記事項

SECS-II メッセージを SML 文字列として取得または設定します。SML プロパティを読み出すとコントロールにセットされているメッセージの構造をツリー形式の SML 文字列で取得します。また SML プロパティに SML 文字列をセットする場合、文字列中に改行コードやスペース、タブなどを自由に入れても構いません(無視されます)。

詳細については付録 A を参照のこと。

146.1.20. SmIType プロパティ

SML 書式(作成中)

Visual C#:

```
Byte SmIType
```

特記事項

SML の書式。

146.1.21. SourceID プロパティ

ソース ID(SECS-I)

Visual C#:

```
UInt16 SourceID
```

特記事項

ソース ID を取得または設定します。

146.1.22. Stream プロパティ

ストリーム番号

Visual C#:

```
Byte Stream
```

特記事項

ストリーム番号を取得または設定します。

146.1.23. SType プロパティ

S タイプ(HSMS)

Visual C#:

```
Byte SType
```

特記事項

セッションタイプを取得または設定します。

146.1.24. SuggestedReplyMsg プロパティ

推奨される返信メッセージ

Visual C#:

```
Byte[] SuggestedReplyMsg
```

特記事項

メッセージ構造を検証した結果、もっとも適切だと考えられる応答メッセージを取得または設定します。

146.1.25. SystemBytes プロパティ

システムバイト

Visual C#:

```
UInt32 SystemBytes
```

特記事項

システムバイトを取得または設定します。ソースIDとトランザクションIDを合わせた4バイトの事を指します。二次メッセージは一次メッセージのシステムバイトと同じでなければなりません。

146.1.26. this[string] プロパティ

指定したノードの値を取得します

Visual C#:

```
this[string]
```

特記事項

Node プロパティと NodeValue プロパティを組み合わせたシンタックス・シュガーです。読み出し専用プロパティ。

146.1.27. TransactionID プロパティ

トランザクション ID (SECS-I)

Visual C#:

```
UInt16 TransactionID
```

特記事項

トランザクション ID を取得または設定します。

146.1.28. Verification プロパティ

認証結果

Visual C#:

```
SecsVerify Verification
```


特記事項

Sml プロパティまたは Msg プロパティにセットした際、メッセージが正しいかどうかを認証し、その認証結果がセットされます。検証したメッセージが一次メッセージの場合、SuggestedReplyMsg プロパティに推奨される二次メッセージがセットされます。

146.1.29. Wbit プロパティ

ウェイトビット

Visual C#:

```
bool wbit
```

特記事項

ウェイトビットを取得または設定します。
返信要求がある場合は True となります。

146.2. メソッド

146.2.1. Reply メソッド

返信用にヘッダをセットします

Visual C#:

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

特記事項

指定されたメッセージの返信メッセージとして SECS-II ヘッダを初期化します。もし指定されたメッセージが HSMS コントロールメッセージの場合、Dorian.SecsII コンポーネントはメッセージボディを消去します。その他の場合はメッセージボディは影響を受けません。

引数

変数名	説明
primary	一次メッセージ

146.2.2. Reply メソッド

返信用にヘッダをセットします

Visual C#:

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

特記事項

指定されたメッセージの返信メッセージとして SECS-II ヘッダを初期化します。もし指定されたメッセージが HSMS コントロールメッセージの場合、Dorian.SecsII コンポーネントはメッセージボディを消去します。その他の場合はメッセージボディは影響を受けません。

引数

変数名	説明
primary	一次メッセージ

146.2.3. Reset メソッド

メッセージをリセットします

Visual C#:

```
public void Reset()
```

特記事項

内部のデータ構造とパラメータを初期化します。

147. Dorian.SecsII.NodeItemClass Class

ノードアイテム

プロパティ:

プロパティ	説明
Count	サブアイテム数
Hex	ノードの値 (16 進数)
Index	Text と Hex プロパティ用のインデックス
Text	ノードの値
this[int]	サブアイテムにアクセスするためのシンタックスシュガー
Type	ノードの型

メソッド:

メソッド	説明
Value	ノードの値
ValueHex	ノードの値 (16 進数)

147.1. プロパティ

147.1.1. Count プロパティ

サブアイテム数

Visual C#:

```
int Count
```

147.1.2. Hex プロパティ

ノードの値(16 進数)

Visual C#:

```
string Hex
```

特記事項

ノードの値を取得します。もしノードが数値型の場合、値は 16 進法文字列に変換されます。
読み出し専用プロパティ。

147.1.3. Index プロパティ

Text と Hex プロパティ用のインデックス

Visual C#:

```
int Index
```

147.1.4. Text プロパティ

ノードの値

Visual C#:

```
string Text
```

特記事項

ノードの値を取得します。もしノードが数値型の場合、値は 10 進法文字列に変換されます。
読み出し専用プロパティ。

147.1.5. this[int] プロパティ

サブアイテムにアクセスするためのシンタックスシュガー

Visual C#:

```
this[int]
```

147.1.6. Type プロパティ

ノードの型

Visual C#:

SecsType Type

特記事項

ノードタイプを取得します。
読み出し専用プロパティ。

147.2. メソッド

147.2.1. Value メソッド

ノードの値

Visual C#:

```
public string Value(int index)
```

特記事項

ノードの値を取得します。もしノードが数値型の場合、値は 10 進法文字列に変換されます。

引数

変数名	説明
index	インデックス

戻り値

147.2.2. ValueHex メソッド

ノードの値(16 進数)

Visual C#:

```
public string ValueHex(int index)
```

特記事項

ノードの値を取得します。もしノードが数値型の場合、値は 16 進法文字列に変換されます。

引数

変数名	説明
index	インデックス

戻り値

148. Dorian.SecsType Enum

ノード種別

値:

値	説明
Ascii	アスキー文字列
Ascii2	マルチバイト文字列
Binary	バイナリ
Boolean	ブーリアン
F4	4 バイト浮動小数点
F8	8 バイト浮動小数点
I1	1 バイト符号付き整数
I2	2 バイト符号付き整数
I4	4 バイト符号付き整数
I8	8 バイト符号付き整数
Invalid	無効なタイプ
Jis	JIS8 コード(カタカナ文字列)
List	リスト
U1	1 バイト符号なし整数
U2	2 バイト符号なし整数
U4	4 バイト符号なし整数
U8	8 バイト符号なし整数

149. Dorian.SecsVerify Enum

検証結果

値:

値	説明
Correct	正常
Incorrect	不正
IncorrectAndReply	不正だが返信が必要
NoWBit	Wビットが必要なのにない
UnrecognizedFunction	認識できないファンクション
UnrecognizedStream	認識できないストリーム
UserDefined	ユーザ定義メッセージ
WBit	Wビットが不要なのに付いている
WrongDirection	送信方向が逆

150. Dorian.Utility Class

Dorian.Utility は共通ユーティリティです。

メソッド:

メソッド	説明
atof	文字列を浮動小数点に変換します
atofex	ポインタのバイト長を返します
atoi	文字列を整数に変換します
ExpireLicense	ライセンスを即時無効にします
GetLicense	ライセンスカウンター

150.1. メソッド

150.1.1. atof メソッド

文字列を浮動小数点に変換します

Visual C#:

```
public static double atof(string value)
```

特記事項

文字列を浮動小数点に変換します。

引数

変数名	説明
value	値

戻り値

150.1.2. atofex メソッド

ポインタのバイト長を返します

Visual C#:

```
public static double atofex(string value)
```

特記事項

64ビットの場合は 64 を返し、それ以外は 32 を返します。

引数

変数名	説明
value	値

戻り値

150.1.3. atoi メソッド

文字列を整数に変換します

Visual C#:

```
public static int atoi(string value)
```

特記事項

文字列を整数に変換します。

引数

変数名	説明
value	値

戻り値

150.1.4. ExpireLicense メソッド

ライセンスを即時無効にします

Visual C#:

```
public static void ExpireLicense()
```

特記事項

ライセンスを即時無効にします

150.1.5. GetLicense メソッド

ライセンスカウンター

Visual C#:

```
public static int GetLicense()
```

特記事項

ライセンスが有効な場合は 0 より大きい値を返します。

戻り値