

## 2015-2016 Test beam Run Control

Generated by Doxygen 1.8.10



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>7</b>
4.1	Data Structures . . . . .	7
<b>5</b>	<b>Module Documentation</b>	<b>9</b>
5.1	Socket communication objects . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.2	FPGA board control . . . . .	10
5.2.1	Detailed Description . . . . .	10
5.3	HPTDC chip control . . . . .	11
5.3.1	Detailed Description . . . . .	11
5.3.2	Enumeration Type Documentation . . . . .	11
5.3.2.1	AcquisitionMode . . . . .	11
5.3.2.2	AcquisitionMode . . . . .	11
<b>6</b>	<b>Namespace Documentation</b>	<b>13</b>
6.1	DAQ Namespace Reference . . . . .	13
6.2	DQM Namespace Reference . . . . .	13
<b>7</b>	<b>Data Structure Documentation</b>	<b>15</b>
7.1	OnlineDBHandler::BurstInfo Struct Reference . . . . .	15
7.1.1	Field Documentation . . . . .	15
7.1.1.1	burst_id . . . . .	15
7.1.1.2	time_start . . . . .	15
7.2	Client Class Reference . . . . .	15
7.2.1	Detailed Description . . . . .	17

7.2.2	Constructor & Destructor Documentation	17
7.2.2.1	Client()	17
7.2.2.2	Client(int port)	17
7.2.2.3	~Client()	18
7.2.3	Member Function Documentation	18
7.2.3.1	Announce()	18
7.2.3.2	Connect(const SocketType &type=CLIENT)	18
7.2.3.3	Disconnect()	19
7.2.3.4	GetType() const	19
7.2.3.5	ParseMessage(const SocketMessage &m)	19
7.2.3.6	Receive()	19
7.2.3.7	Receive(const MessageKey &key)	20
7.2.3.8	Send(const Message &m) const	20
7.2.3.9	Send(const Exception &e) const	21
7.2.3.10	SendAndReceive(const SocketMessage &m, const MessageKey &a) const	21
7.2.4	Field Documentation	21
7.2.4.1	fClientId	21
7.2.4.2	fIsConnected	21
7.2.4.3	fType	21
7.3	DQM::GastofCanvas::Coord Struct Reference	21
7.3.1	Field Documentation	21
7.3.1.1	x	22
7.3.1.2	y	22
7.4	DQM::QuarticCanvas::Coord Struct Reference	22
7.4.1	Field Documentation	22
7.4.1.1	x	22
7.4.1.2	y	22
7.5	DQM::DQMProcess Class Reference	22
7.5.1	Detailed Description	23
7.5.2	Member Enumeration Documentation	24
7.5.2.1	Action	24
7.5.3	Constructor & Destructor Documentation	24
7.5.3.1	DQMProcess(int port, unsigned short order=0, const char *det_type="")	24
7.5.3.2	~DQMProcess()	24
7.5.4	Member Function Documentation	24
7.5.4.1	IsInRun()	25
7.5.4.2	ParseMessage(uint32_t *board_address, std::string *filename)	25
7.5.4.3	Run(bool(*fcn)(unsigned int addr, std::string filename, std::vector< std::string > *outputs), const Action &act=NewPlot)	25
7.5.4.4	Run(bool(*fcn)(std::vector< std::string > *outputs), const Action &act=NewPlot)	26

7.5.5	Field Documentation	26
7.5.5.1	fAddressesCanProcess	26
7.5.5.2	fDetectorType	26
7.5.5.3	fOrder	26
7.5.5.4	fRunNumber	26
7.6	file_header_t Struct Reference	26
7.6.1	Detailed Description	26
7.6.2	Field Documentation	27
7.6.2.1	acq_mode	27
7.6.2.2	det_mode	27
7.6.2.3	magic	27
7.6.2.4	num_hptdc	27
7.6.2.5	run_id	27
7.6.2.6	spill_id	27
7.7	FileReader Class Reference	27
7.7.1	Detailed Description	28
7.7.2	Constructor & Destructor Documentation	28
7.7.2.1	FileReader()	28
7.7.2.2	FileReader(std::string name)	28
7.7.2.3	~FileReader()	29
7.7.3	Member Function Documentation	29
7.7.3.1	Clear()	29
7.7.3.2	Dump() const	29
7.7.3.3	GetAcquisitionMode() const	29
7.7.3.4	GetBurstId() const	29
7.7.3.5	GetDetectionMode() const	29
7.7.3.6	GetNextEvent(TDCEvent *)	29
7.7.3.7	GetNextMeasurement(unsigned int channel_id, TDCMeasurement *mc)	29
7.7.3.8	GetNumEvents() const	30
7.7.3.9	GetNumTDCs() const	30
7.7.3.10	GetRunId() const	30
7.7.3.11	IsOpen() const	30
7.7.3.12	Open(std::string name)	30
7.7.4	Field Documentation	30
7.7.4.1	fFile	30
7.7.4.2	fHeader	30
7.7.4.3	fNumEvents	30
7.7.4.4	fReadoutMode	30
7.7.4.5	fWriteTime	30
7.8	DAQ::FPGAHandler Class Reference	30

7.8.1	Detailed Description	32
7.8.2	Constructor & Destructor Documentation	32
7.8.2.1	FPGAHandler(int port, const char *dev)	32
7.8.2.2	~FPGAHandler()	32
7.8.3	Member Function Documentation	33
7.8.3.1	CloseFile()	33
7.8.3.2	ErrorState()	33
7.8.3.3	GetFilename() const	33
7.8.3.4	GetTDC(unsigned int i=0)	33
7.8.3.5	GetTDCControl() const	33
7.8.3.6	GetTDCStatus() const	33
7.8.3.7	GetType() const	33
7.8.3.8	OpenFile()	34
7.8.3.9	RegisterTest() const	34
7.8.3.10	RetrieveSetupWord() const	34
7.8.3.11	SendSetupWord() const	35
7.8.3.12	SetTDCSetup(const TDCSetup &s)	35
7.8.3.13	StartAcquisition()	35
7.8.3.14	Stop()	35
7.8.3.15	StopAcquisition()	36
7.8.4	Field Documentation	36
7.8.4.1	fFilename	36
7.8.4.2	fIsFileOpen	36
7.8.4.3	fIsTDCInReadout	36
7.8.4.4	fOutput	36
7.8.4.5	fTDC	36
7.9	DQM::GastofCanvas Class Reference	36
7.9.1	Detailed Description	38
7.9.2	Constructor & Destructor Documentation	38
7.9.2.1	GastofCanvas()	38
7.9.2.2	GastofCanvas(TString name, unsigned int width=500, unsigned int height=500, TString upper_label="")	38
7.9.2.3	GastofCanvas(TString name, TString upper_label)	38
7.9.2.4	~GastofCanvas()	39
7.9.3	Member Function Documentation	39
7.9.3.1	Build()	39
7.9.3.2	DrawGrid()	39
7.9.3.3	FillChannel(unsigned short nino_id, unsigned short channel_id, double content)	39
7.9.3.4	GetCoordinates(unsigned short nino_id, unsigned short channel_id) const	39
7.9.3.5	Grid()	39

7.9.3.6	Save(TString ext=""png"", TString path=""."")	39
7.9.3.7	SetRunInfo(unsigned int board_id, unsigned int run_id, unsigned int spill_id, T↵String date)	39
7.9.3.8	SetUpperLabel(TString text)	39
7.9.4	Field Documentation	39
7.9.4.1	c1	40
7.9.4.2	c2	40
7.9.4.3	fBoardId	40
7.9.4.4	fHeight	40
7.9.4.5	fHist	40
7.9.4.6	fLabel1	40
7.9.4.7	fLabel2	40
7.9.4.8	fLabel3	40
7.9.4.9	fLabel4	40
7.9.4.10	fLabelsDrawn	40
7.9.4.11	fLegend	40
7.9.4.12	fLegendNumEntries	40
7.9.4.13	fLegendX	40
7.9.4.14	fLegendY	40
7.9.4.15	fRunDate	40
7.9.4.16	fRunId	40
7.9.4.17	fSpillId	40
7.9.4.18	fUpperLabel	40
7.9.4.19	fUpperLabelText	40
7.9.4.20	fWidth	40
7.10	Logger Class Reference	40
7.10.1	Detailed Description	41
7.10.2	Constructor & Destructor Documentation	41
7.10.2.1	Logger(std::ostream &lhs, std::ostream &rhs=std::cout)	41
7.10.2.2	~Logger()	41
7.10.3	Field Documentation	41
7.10.3.1	fBuffer	41
7.10.3.2	fStream	41
7.11	LogRedirector Class Reference	41
7.11.1	Detailed Description	42
7.11.2	Constructor & Destructor Documentation	42
7.11.2.1	LogRedirector(std::ostream &stm=std::cout)	42
7.11.3	Member Function Documentation	42
7.11.3.1	contents() const	42
7.11.4	Field Documentation	42

7.11.4.1	fRedirect	42
7.11.4.2	fSS	42
7.12	Message Class Reference	42
7.12.1	Detailed Description	43
7.12.2	Constructor & Destructor Documentation	43
7.12.2.1	Message()	43
7.12.2.2	Message(const char *msg)	43
7.12.2.3	Message(std::string msg)	43
7.12.2.4	~Message()	44
7.12.3	Member Function Documentation	44
7.12.3.1	Dump(std::ostream &os=std::cout) const	44
7.12.3.2	GetKey() const	44
7.12.3.3	GetString() const	44
7.12.3.4	IsFromWeb() const	44
7.12.4	Field Documentation	44
7.12.4.1	fString	44
7.13	Messenger Class Reference	44
7.13.1	Detailed Description	46
7.13.2	Constructor & Destructor Documentation	46
7.13.2.1	Messenger()	46
7.13.2.2	Messenger(int port)	46
7.13.2.3	~Messenger()	46
7.13.3	Member Function Documentation	47
7.13.3.1	AddClient()	47
7.13.3.2	Broadcast(const Message &m) const	47
7.13.3.3	Connect()	47
7.13.3.4	Disconnect()	48
7.13.3.5	DisconnectClient(int sid, MessageKey key, bool force=false)	48
7.13.3.6	GetType() const	49
7.13.3.7	ProcessMessage(SocketMessage m, int sid)	49
7.13.3.8	Receive()	49
7.13.3.9	Send(const Message &m, int sid) const	50
7.13.3.10	SendAll(const Socket::SocketType &type, const Message &m) const	50
7.13.3.11	SendAll(const Socket::SocketType &type, const Exception &e) const	50
7.13.3.12	StartAcquisition()	50
7.13.3.13	StopAcquisition()	51
7.13.3.14	SwitchClientType(int sid, Socket::SocketType type)	51
7.13.4	Field Documentation	51
7.13.4.1	fNumAttempts	51
7.13.4.2	fPID	51



7.13.4.3	fStderrPipe	51
7.13.4.4	fStdoutPipe	51
7.14	OnlineDBHandler Class Reference	51
7.14.1	Detailed Description	52
7.14.2	Member Typedef Documentation	53
7.14.2.1	BurstInfos	53
7.14.2.2	RunCollection	53
7.14.2.3	TDCConditionsCollection	53
7.14.3	Constructor & Destructor Documentation	53
7.14.3.1	OnlineDBHandler(std::string path=std::string(std::getenv("PPS_PATH"))+"/run_↵ _infos.db")	53
7.14.3.2	~OnlineDBHandler()	53
7.14.4	Member Function Documentation	53
7.14.4.1	BuildTables()	53
7.14.4.2	GetLastBurst(unsigned int run) const	53
7.14.4.3	GetLastRun() const	53
7.14.4.4	GetRunInfo(unsigned int run) const	53
7.14.4.5	GetRuns() const	53
7.14.4.6	GetTDCConditions(unsigned int run_id) const	53
7.14.4.7	NewBurst()	54
7.14.4.8	NewRun()	54
7.14.4.9	Select(std::string req, int num_fields=-1) const	54
7.14.4.10	SetHVConditions(unsigned short channel_id, unsigned int vmax, unsigned imax)	54
7.14.4.11	SetTDCConditions(unsigned short tdc_id, unsigned long tdc_address, unsigned short tdc_acq_mode, unsigned short tdc_det_mode, std::string detector)	54
7.14.5	Field Documentation	54
7.14.5.1	fDB	54
7.15	DQM::PPSCanvas Class Reference	55
7.15.1	Detailed Description	56
7.15.2	Constructor & Destructor Documentation	56
7.15.2.1	PPSCanvas()	56
7.15.2.2	PPSCanvas(TString name, unsigned int width=500, unsigned int height=500, T↵ String upper_label="")	56
7.15.2.3	PPSCanvas(TString name, TString upper_label)	56
7.15.2.4	~PPSCanvas()	57
7.15.3	Member Function Documentation	57
7.15.3.1	Build()	57
7.15.3.2	DrawGrid()	57
7.15.3.3	Grid()	57
7.15.3.4	Save(TString ext="png", TString path="")	57
7.15.3.5	SetRunInfo(unsigned int run_id, TString date)	57

7.15.3.6	SetUpperLabel(TString text)	57
7.15.4	Field Documentation	57
7.15.4.1	c1	57
7.15.4.2	c2	57
7.15.4.3	fHeight	57
7.15.4.4	fLabel1	57
7.15.4.5	fLabel2	57
7.15.4.6	fLabel3	57
7.15.4.7	fLabelsDrawn	58
7.15.4.8	fLegend	58
7.15.4.9	fLegendNumEntries	58
7.15.4.10	fLegendX	58
7.15.4.11	fLegendY	58
7.15.4.12	fRunDate	58
7.15.4.13	fRunId	58
7.15.4.14	fUpperLabel	58
7.15.4.15	fUpperLabelText	58
7.15.4.16	fWidth	58
7.16	DQM::QuarticCanvas Class Reference	58
7.16.1	Detailed Description	60
7.16.2	Constructor & Destructor Documentation	60
7.16.2.1	QuarticCanvas()	60
7.16.2.2	QuarticCanvas(TString name, unsigned int width=500, unsigned int height=500, TString upper_label="")	60
7.16.2.3	QuarticCanvas(TString name, TString upper_label)	60
7.16.2.4	~QuarticCanvas()	61
7.16.3	Member Function Documentation	61
7.16.3.1	Build()	61
7.16.3.2	DrawGrid()	61
7.16.3.3	FillChannel(unsigned short channel_id, double content)	61
7.16.3.4	GetCoordinates(unsigned short channel_id) const	61
7.16.3.5	Grid()	61
7.16.3.6	Save(TString ext="png", TString path="")	61
7.16.3.7	SetRunInfo(unsigned int board_id, unsigned int run_id, unsigned int spill_id, TString date)	61
7.16.3.8	SetUpperLabel(TString text)	61
7.16.4	Field Documentation	61
7.16.4.1	c1	62
7.16.4.2	c2	62
7.16.4.3	fBoardId	62

7.16.4.4	fHeight	62
7.16.4.5	fHist	62
7.16.4.6	fLabel1	62
7.16.4.7	fLabel2	62
7.16.4.8	fLabel3	62
7.16.4.9	fLabel4	62
7.16.4.10	fLabelsDrawn	62
7.16.4.11	fLegend	62
7.16.4.12	fLegendNumEntries	62
7.16.4.13	fLegendX	62
7.16.4.14	fLegendY	62
7.16.4.15	fRunDate	62
7.16.4.16	fRunId	62
7.16.4.17	fSpillId	62
7.16.4.18	fUpperLabel	62
7.16.4.19	fUpperLabelText	62
7.16.4.20	fWidth	62
7.17	DAQ::QuickUSBHandler Class Reference	62
7.17.1	Detailed Description	64
7.17.2	Constructor & Destructor Documentation	64
7.17.2.1	QuickUSBHandler()	64
7.17.2.2	~QuickUSBHandler()	64
7.17.3	Member Function Documentation	64
7.17.3.1	Fetch(uint16_t addr, uint16_t size) const	64
7.17.3.2	GetDLLVersion() const	64
7.17.3.3	GetDriverVersion() const	64
7.17.3.4	GetFWVersion() const	64
7.17.3.5	Init()	64
7.17.3.6	Reset() const	65
7.17.3.7	StartBulkTransfer(QVOIDRETURN callback(PQBULKSTREAM))	65
7.17.3.8	StopBulkTransfer()	65
7.17.3.9	Write(uint16_t addr, uint8_t word) const	65
7.17.3.10	Write(uint16_t addr, std::vector< uint8_t > &words, uint16_t size) const	65
7.17.4	Field Documentation	65
7.17.4.1	fDevice	65
7.17.4.2	fHandle	65
7.17.4.3	fIsStopping	65
7.17.4.4	fStreamId	65
7.18	Socket Class Reference	65
7.18.1	Detailed Description	67

7.18.2	Member Typedef Documentation	67
7.18.2.1	SocketCollection	67
7.18.3	Member Enumeration Documentation	67
7.18.3.1	SocketType	67
7.18.4	Constructor & Destructor Documentation	67
7.18.4.1	Socket()	67
7.18.4.2	Socket(int port)	67
7.18.4.3	~Socket()	67
7.18.5	Member Function Documentation	68
7.18.5.1	AcceptConnections(Socket &socket)	68
7.18.5.2	Bind()	68
7.18.5.3	Configure()	68
7.18.5.4	Create()	68
7.18.5.5	DumpConnected() const	69
7.18.5.6	FetchMessage(int id=-1) const	69
7.18.5.7	GetPort() const	69
7.18.5.8	GetSocketId() const	69
7.18.5.9	GetSocketType(int sid) const	69
7.18.5.10	IsWebSocket(int sid) const	69
7.18.5.11	Listen(int maxconn)	69
7.18.5.12	PrepareConnection()	70
7.18.5.13	SelectConnections()	70
7.18.5.14	SendMessage(Message message, int id=-1) const	70
7.18.5.15	SetPort(int port)	70
7.18.5.16	SetSocketId(int sid)	70
7.18.5.17	Start()	70
7.18.5.18	Stop()	71
7.18.6	Field Documentation	71
7.18.6.1	fAddress	71
7.18.6.2	fBuffer	71
7.18.6.3	fMaster	71
7.18.6.4	fPort	71
7.18.6.5	fReadFds	71
7.18.6.6	fSocketId	71
7.18.6.7	fSocketsConnected	71
7.19	SocketMessage Class Reference	72
7.19.1	Detailed Description	74
7.19.2	Constructor & Destructor Documentation	74
7.19.2.1	SocketMessage()	74
7.19.2.2	SocketMessage(const Message &msg)	74

7.19.2.3	SocketMessage(const char *msg_s)	74
7.19.2.4	SocketMessage(std::string msg_s)	75
7.19.2.5	SocketMessage(const MessageKey &key)	75
7.19.2.6	SocketMessage(const MessageKey &key, const char *value)	75
7.19.2.7	SocketMessage(const MessageKey &key, std::string value)	75
7.19.2.8	SocketMessage(const MessageKey &key, const short value)	76
7.19.2.9	SocketMessage(const MessageKey &key, const int value)	76
7.19.2.10	SocketMessage(const MessageKey &key, const long value)	76
7.19.2.11	SocketMessage(const MessageKey &key, const float value)	76
7.19.2.12	SocketMessage(const MessageKey &key, const double value)	77
7.19.2.13	SocketMessage(MessageMap msg_m)	77
7.19.2.14	~SocketMessage()	77
7.19.3	Member Function Documentation	77
7.19.3.1	Dump(std::ostream &os=std::cout) const	77
7.19.3.2	GetCleanedValue() const	77
7.19.3.3	GetIntValue() const	77
7.19.3.4	GetKey() const	77
7.19.3.5	GetString() const	77
7.19.3.6	GetValue() const	78
7.19.3.7	GetVectorValue() const	78
7.19.3.8	Object() const	78
7.19.3.9	SetKeyValue(const MessageKey &key, const char *value)	78
7.19.3.10	SetKeyValue(const MessageKey &key, short int_value)	78
7.19.3.11	SetKeyValue(const MessageKey &key, int int_value)	78
7.19.3.12	SetKeyValue(const MessageKey &key, long int_value)	79
7.19.3.13	SetKeyValue(const MessageKey &key, float float_value)	79
7.19.3.14	SetKeyValue(const MessageKey &key, double double_value)	79
7.19.3.15	String() const	79
7.19.4	Field Documentation	79
7.19.4.1	fMessage	80
7.20	DAQ::TDC Class Reference	80
7.20.1	Detailed Description	81
7.20.2	Member Enumeration Documentation	81
7.20.2.1	DetectionMode	81
7.20.3	Constructor & Destructor Documentation	81
7.20.3.1	TDC(unsigned int id, QuickUSBHandler *h)	81
7.20.3.2	~TDC()	82
7.20.4	Member Function Documentation	82
7.20.4.1	CheckFirmwareVersion() const	82
7.20.4.2	FetchEvents()	82

7.20.4.3	GetSetupRegister()	82
7.20.4.4	ReadConfiguration()	82
7.20.4.5	ReadRegister(unsigned int r)	82
7.20.4.6	ReadStatus()	82
7.20.4.7	SendConfiguration()	82
7.20.4.8	SetSetupRegister(const TDCSetup &c)	82
7.20.4.9	SoftReset()	82
7.20.4.10	WriteRegister(unsigned int r, const T &v)	82
7.20.5	Field Documentation	82
7.20.5.1	fBS	82
7.20.5.2	fControl	82
7.20.5.3	fId	82
7.20.5.4	fSetup	82
7.20.5.5	fStatus	82
7.20.5.6	fUSB	82
7.21	TDCBoundaryScan Class Reference	83
7.21.1	Detailed Description	84
7.21.2	Constructor & Destructor Documentation	84
7.21.2.1	TDCBoundaryScan()	84
7.21.2.2	TDCBoundaryScan(const TDCBoundaryScan &bs)	84
7.21.3	Member Function Documentation	85
7.21.3.1	SetConstantValues()	85
7.21.4	Field Documentation	85
7.21.4.1	kAuxClock	85
7.21.4.2	kBunchReset	85
7.21.4.3	kClk	85
7.21.4.4	kDataReady	85
7.21.4.5	kEncodedControl	85
7.21.4.6	kError	85
7.21.4.7	kEventReset	85
7.21.4.8	kGetData	85
7.21.4.9	kHit	85
7.21.4.10	kParallelDataOut	85
7.21.4.11	kParallelEnable	85
7.21.4.12	kReset	85
7.21.4.13	kSerialBypassIn	85
7.21.4.14	kSerialIn	85
7.21.4.15	kSerialOut	85
7.21.4.16	kStrobeOut	85
7.21.4.17	kTest	85

7.21.4.18 kTokenBypassIn . . . . .	85
7.21.4.19 kTokenIn . . . . .	85
7.21.4.20 kTokenOut . . . . .	85
7.21.4.21 kTrigger . . . . .	85
7.22 OnlineDBHandler::TDCConditions Struct Reference . . . . .	86
7.22.1 Member Function Documentation . . . . .	86
7.22.1.1 operator=(const TDCConditions &rhs) . . . . .	86
7.22.1.2 operator==(const TDCConditions &rhs) const . . . . .	86
7.22.2 Field Documentation . . . . .	86
7.22.2.1 detector . . . . .	86
7.22.2.2 run_id . . . . .	86
7.22.2.3 tdc_acq_mode . . . . .	86
7.22.2.4 tdc_address . . . . .	86
7.22.2.5 tdc_det_mode . . . . .	86
7.22.2.6 tdc_id . . . . .	86
7.23 TDCControl Class Reference . . . . .	86
7.23.1 Detailed Description . . . . .	88
7.23.2 Member Enumeration Documentation . . . . .	88
7.23.2.1 EnablePattern . . . . .	88
7.23.2.2 RegisterName . . . . .	88
7.23.3 Constructor & Destructor Documentation . . . . .	88
7.23.3.1 TDCControl() . . . . .	88
7.23.3.2 TDCControl(const TDCControl &c) . . . . .	89
7.23.3.3 TDCControl(const std::vector< uint8_t > &words) . . . . .	89
7.23.4 Member Function Documentation . . . . .	89
7.23.4.1 DisableAllChannels() . . . . .	89
7.23.4.2 DisableChannel(unsigned int id) . . . . .	89
7.23.4.3 Dump(int verb=1, std::ostream &os=std::cout) const . . . . .	90
7.23.4.4 EnableAllChannels() . . . . .	90
7.23.4.5 EnableChannel(unsigned int id) . . . . .	90
7.23.4.6 GetDLLReset() const . . . . .	91
7.23.4.7 GetEnablePattern() const . . . . .	91
7.23.4.8 GetGlobalReset() const . . . . .	91
7.23.4.9 GetPLLReset() const . . . . .	92
7.23.4.10 SetConstantValues() . . . . .	92
7.23.4.11 SetControlParity(const bool cp=true) . . . . .	92
7.23.4.12 SetDLLReset(const bool dr=true) . . . . .	92
7.23.4.13 SetEnablePattern(const EnablePattern &ep) . . . . .	93
7.23.4.14 SetGlobalReset(const bool gr=true) . . . . .	93
7.23.4.15 SetPLLReset(const bool pr=true) . . . . .	93

7.23.5	Field Documentation	93
7.23.5.1	kControlParity	93
7.23.5.2	kDLLReset	93
7.23.5.3	kEnableChannel	93
7.23.5.4	kEnablePattern	93
7.23.5.5	kGlobalReset	93
7.23.5.6	kPLLReset	94
7.24	TDCErrorFlag Class Reference	94
7.24.1	Detailed Description	94
7.24.2	Constructor & Destructor Documentation	95
7.24.2.1	TDCErrorFlag(uint16_t ef)	95
7.24.2.2	~TDCErrorFlag()	95
7.24.3	Member Function Documentation	95
7.24.3.1	Dump() const	95
7.24.3.2	GetWord() const	95
7.24.3.3	HasGroupError(unsigned int group_id) const	95
7.24.3.4	HasInternalChipError() const	95
7.24.3.5	HasL1BufferOverflow(unsigned int group_id) const	95
7.24.3.6	HasReachedEventSizeLimit() const	95
7.24.3.7	HasReadoutFIFOOverflow(unsigned int group_id) const	95
7.24.3.8	HasTriggerFIFOOverflow() const	95
7.24.4	Friends And Related Function Documentation	95
7.24.4.1	operator<<	95
7.24.5	Field Documentation	95
7.24.5.1	fWord	95
7.25	TDCEvent Class Reference	95
7.25.1	Detailed Description	97
7.25.2	Member Enumeration Documentation	97
7.25.2.1	EventType	97
7.25.3	Constructor & Destructor Documentation	97
7.25.3.1	TDCEvent()	97
7.25.3.2	TDCEvent(const TDCEvent &ev)	97
7.25.3.3	TDCEvent(const uint32_t &word)	97
7.25.3.4	TDCEvent(const EventType &ev)	97
7.25.3.5	~TDCEvent()	97
7.25.4	Member Function Documentation	97
7.25.4.1	Dump() const	98
7.25.4.2	GetBunchId() const	98
7.25.4.3	GetChannelId() const	98
7.25.4.4	GetErrorFlags() const	98



7.25.4.5	GetETTT() const	99
7.25.4.6	GetEventCount() const	99
7.25.4.7	GetEventId() const	99
7.25.4.8	GetGeo() const	100
7.25.4.9	GetStatus() const	100
7.25.4.10	GetTDCId() const	100
7.25.4.11	GetTime(bool pair=false) const	101
7.25.4.12	GetType() const	101
7.25.4.13	GetWidth() const	101
7.25.4.14	GetWord() const	102
7.25.4.15	GetWordCount() const	102
7.25.4.16	IsTrailing() const	102
7.25.4.17	SetWord(const uint32_t &word)	102
7.25.5	Field Documentation	102
7.25.5.1	fWord	102
7.26	TDCMeasurement Class Reference	102
7.26.1	Detailed Description	103
7.26.2	Constructor & Destructor Documentation	103
7.26.2.1	TDCMeasurement()	103
7.26.2.2	TDCMeasurement(const std::vector< TDCEvent > &v)	103
7.26.2.3	~TDCMeasurement()	103
7.26.3	Member Function Documentation	103
7.26.3.1	Dump()	104
7.26.3.2	GetBunchId()	104
7.26.3.3	GetChannelId(unsigned short event_id=0)	104
7.26.3.4	GetETTT()	104
7.26.3.5	GetEventId()	104
7.26.3.6	GetLeadingTime(unsigned short event_id=0)	104
7.26.3.7	GetTDCId()	104
7.26.3.8	GetToT(unsigned short event_id=0)	105
7.26.3.9	GetTrailingTime(unsigned short event_id=0)	105
7.26.3.10	NumErrors() const	105
7.26.3.11	NumEvents() const	105
7.26.3.12	SetEventsCollection(const std::vector< TDCEvent > &v)	105
7.26.4	Field Documentation	105
7.26.4.1	fEvents	105
7.26.4.2	fMap	105
7.27	TDCRegister Class Reference	105
7.27.1	Detailed Description	106
7.27.2	Member Typedef Documentation	107

7.27.2.1	bit	107
7.27.2.2	word_t	107
7.27.3	Constructor & Destructor Documentation	107
7.27.3.1	TDCRegister(const unsigned int size)	107
7.27.3.2	TDCRegister(const unsigned int size, const TDCRegister &r)	107
7.27.3.3	TDCRegister(const unsigned int size, const std::vector< uint8_t > words)	107
7.27.3.4	~TDCRegister()	108
7.27.4	Member Function Documentation	108
7.27.4.1	Clear()	108
7.27.4.2	DumpRegister(std::ostream &os=std::cout, const bit max_bits=-1) const	108
7.27.4.3	GetBits(uint16_t lsb, uint8_t size) const	108
7.27.4.4	GetNumWords() const	108
7.27.4.5	GetWord(const unsigned int i) const	108
7.27.4.6	GetWords() const	108
7.27.4.7	operator=(const TDCRegister &r)	108
7.27.4.8	SetBits(uint16_t lsb, uint16_t word, uint8_t size)	108
7.27.4.9	SetConstantValues()=0	108
7.27.4.10	SetWord(const unsigned int i, const word_t word)	109
7.27.5	Field Documentation	109
7.27.5.1	fNumWords	109
7.27.5.2	fWord	109
7.27.5.3	fWordSize	109
7.28	TDCSetup Class Reference	109
7.28.1	Detailed Description	116
7.28.2	Member Enumeration Documentation	116
7.28.2.1	CoreClockSource	116
7.28.2.2	DeadTime	117
7.28.2.3	DLLClockSource	117
7.28.2.4	DLLSpeedMode	117
7.28.2.5	EdgeResolution	117
7.28.2.6	EnabledError	117
7.28.2.7	IOClockSource	118
7.28.2.8	ReadoutSingleCycleSpeed	118
7.28.2.9	ReadoutSpeed	118
7.28.2.10	SerialClockSource	118
7.28.2.11	SerialStrobeType	119
7.28.2.12	WidthResolution	119
7.28.3	Constructor & Destructor Documentation	119
7.28.3.1	TDCSetup()	120
7.28.3.2	TDCSetup(const TDCSetup &c)	121

7.28.4	Member Function Documentation	121
7.28.4.1	Dump(int verb=1, std::ostream &os=std::cout) const	122
7.28.4.2	GetChannelOffset(int channel) const	122
7.28.4.3	GetCoarseCountOffset() const	122
7.28.4.4	GetDeadTime() const	123
7.28.4.5	GetDLLAdjustment(int tap) const	123
7.28.4.6	GetEdgeResolution() const	124
7.28.4.7	GetEdgesPairing() const	124
7.28.4.8	GetEnableError() const	124
7.28.4.9	GetEnableErrorBypass() const	125
7.28.4.10	GetEnableErrorMark() const	125
7.28.4.11	GetEnableJTAGReadout() const	125
7.28.4.12	GetEnableReadoutOccupancy() const	126
7.28.4.13	GetEnableReadoutSeparator() const	126
7.28.4.14	GetEnableSerial() const	126
7.28.4.15	GetLeadingMode() const	126
7.28.4.16	GetMatchWindow() const	127
7.28.4.17	GetMaxEventSize() const	127
7.28.4.18	GetRCAdjustment(int tap)	127
7.28.4.19	GetReadoutFIFOSize() const	128
7.28.4.20	GetRejectCountOffset() const	128
7.28.4.21	GetRejectFIFOFull() const	128
7.28.4.22	GetSearchWindow() const	129
7.28.4.23	GetSetupParity() const	129
7.28.4.24	GetTDCId() const	130
7.28.4.25	GetTestInvert() const	130
7.28.4.26	GetTestMode() const	130
7.28.4.27	GetTrailingMode() const	130
7.28.4.28	GetTriggerCountOffset() const	131
7.28.4.29	GetTriggerLatency() const	131
7.28.4.30	GetTriggerMatchingMode() const	131
7.28.4.31	GetVernierOffset() const	132
7.28.4.32	GetWidthResolution() const	132
7.28.4.33	SetAllChannelsOffset(uint16_t offset)	132
7.28.4.34	SetAllTapsDLLAdjustment(uint8_t adj)	133
7.28.4.35	SetBypassInputs(const bool sbi=true)	133
7.28.4.36	SetChannelOffset(int channel, uint16_t offset)	133
7.28.4.37	SetCoarseCountOffset(uint16_t cco)	133
7.28.4.38	SetConstantValues()	134
7.28.4.39	SetCoreClockDelay(const bool delay_clock, const uint8_t delay)	134

7.28.4.40 SetCoreClockSource(const CoreClockSource ccs)	134
7.28.4.41 SetDeadTime(const DeadTime dt)	135
7.28.4.42 SetDLLAdjustment(int tap, uint8_t adj)	135
7.28.4.43 SetDLLClockDelay(const bool delay_clock, const uint8_t delay)	135
7.28.4.44 SetDLLClockSource(const DLLClockSource dcs)	135
7.28.4.45 SetDLLControl(const uint8_t dc)	136
7.28.4.46 SetDLLMode(const DLLSpeedMode dsm)	136
7.28.4.47 SetEdgeResolution(const EdgeResolution r)	137
7.28.4.48 SetEdgesPairing(const bool pair=true)	137
7.28.4.49 SetEnableAutomaticReject(const bool ear=true)	137
7.28.4.50 SetEnableBytewise(const bool seb=true)	138
7.28.4.51 SetEnableDirectBunchReset(const bool edbr=true)	138
7.28.4.52 SetEnableDirectEventReset(const bool eder=true)	138
7.28.4.53 SetEnableDirectTrigger(const bool edt=true)	138
7.28.4.54 SetEnableError(const uint16_t &err)	139
7.28.4.55 SetEnableErrorBypass(const bool eb)	139
7.28.4.56 SetEnableErrorMark(const bool em)	139
7.28.4.57 SetEnableGlobalHeader(const bool egh=true)	140
7.28.4.58 SetEnableGlobalTrailer(const bool egt=true)	140
7.28.4.59 SetEnableJTAGReadout(const bool jr)	140
7.28.4.60 SetEnableLocalHeader(const bool elh=true)	141
7.28.4.61 SetEnableLocalTrailer(const bool elt=true)	141
7.28.4.62 SetEnableMasterResetCode(const bool emrc=true)	141
7.28.4.63 SetEnableMasterResetOnEventReset(const bool emroer=true)	142
7.28.4.64 SetEnableOverflowDetect(const bool eod=true)	142
7.28.4.65 SetEnableReadoutOccupancy(const bool ro=true)	142
7.28.4.66 SetEnableReadoutSeparator(const bool ro=true)	143
7.28.4.67 SetEnableRelative(const bool er=true)	143
7.28.4.68 SetEnableResetChannelBufferWhenSeparator(const bool ercbws=true)	143
7.28.4.69 SetEnableSeparatorOnBunchReset(const bool esobr=true)	144
7.28.4.70 SetEnableSeparatorOnEventReset(const bool esoer=true)	144
7.28.4.71 SetEnableSerial(const bool es)	144
7.28.4.72 SetEnableSetCountersOnBunchReset(const bool escobr=true)	145
7.28.4.73 SetEnableTTLClock(const bool tc=true)	145
7.28.4.74 SetEnableTTLControl(const bool tc=true)	145
7.28.4.75 SetEnableTTLHit(const bool th=true)	146
7.28.4.76 SetEnableTTLReset(const bool tr=true)	146
7.28.4.77 SetEnableTTLSerial(const bool ts=true)	146
7.28.4.78 SetEventCountOffset(uint16_t eco)	147
7.28.4.79 SetIOClockDelay(const bool delay_clock, const uint8_t delay)	147

7.28.4.80	SetIOClockSource(const IOClockSource ics)	148
7.28.4.81	SetKeepToken(const bool kt=true)	148
7.28.4.82	SetLeadingMode(const bool lead=true)	148
7.28.4.83	SetLowPowerMode(const bool lpm=true)	148
7.28.4.84	SetMaster(const bool m=true)	149
7.28.4.85	SetMatchWindow(uint16_t mw)	149
7.28.4.86	SetMaxEventSize(int sz=-1)	149
7.28.4.87	SetModeRC(const bool mr=true)	150
7.28.4.88	SetModeRCCompression(const bool mrc=true)	150
7.28.4.89	SetPLLControl(const uint8_t charge_pump_current=0x4, const bool power_↵ down_mode=false, const bool enable_test_outputs=false, const bool invert_↵ connection_to_status=false)	150
7.28.4.90	SetRCAdjustment(int tap, uint8_t adj)	151
7.28.4.91	SetReadoutFIFOSize(int rfs)	151
7.28.4.92	SetReadoutSingleCycleSpeed(const ReadoutSingleCycleSpeed rscs=RSC_↵ 40Mbits_s)	151
7.28.4.93	SetReadoutSpeedSelect(const ReadoutSpeed rss=RO_Fixed)	152
7.28.4.94	SetRejectCountOffset(uint16_t rco)	152
7.28.4.95	SetRejectFIFOFull(const bool rej=true)	153
7.28.4.96	SetRollOver(const uint16_t ro=0xFFFF)	153
7.28.4.97	SetSearchWindow(uint16_t sw)	153
7.28.4.98	SetSerialClockDelay(const bool delay_clock, const uint8_t delay)	153
7.28.4.99	SetSerialClockSource(const SerialClockSource scs)	154
7.28.4.100	SetSerialDelay(const uint8_t sd=0x0)	154
7.28.4.101	SetSetupParity(const bool sp=true)	154
7.28.4.102	SetStrobeSelect(const SerialStrobeType ss=SS_NoStrobe)	155
7.28.4.103	SetTDCId(const uint8_t id=0x0)	155
7.28.4.104	SetTest(const bool test=true)	156
7.28.4.105	SetTestInvert(const bool ti=true)	156
7.28.4.106	SetTestMode(const bool tm=true)	156
7.28.4.107	SetTokenDelay(const uint8_t td=0x0)	156
7.28.4.108	SetTrailingMode(const bool trail=true)	157
7.28.4.109	SetTriggerCountOffset(uint16_t tco)	157
7.28.4.110	SetTriggerMatchingMode(const bool trig=true)	157
7.28.4.111	SetVernierOffset(const uint8_t vo)	158
7.28.4.112	SetWidthResolution(const WidthResolution r)	158
7.28.5	Field Documentation	158
7.28.5.1	kCoarseCountOffset	158
7.28.5.2	kCoreClockDelay	158
7.28.5.3	kCoreClockSource	158
7.28.5.4	kDeadTime	158

7.28.5.5	kDLLClockDelay	158
7.28.5.6	kDLLClockSource	159
7.28.5.7	kDLLControl	159
7.28.5.8	kDLLMode	159
7.28.5.9	kDLLTapAdjust0	159
7.28.5.10	kEnableAutomaticReject	159
7.28.5.11	kEnableBytewise	159
7.28.5.12	kEnableDirectBunchReset	159
7.28.5.13	kEnableDirectEventReset	159
7.28.5.14	kEnableDirectTrigger	159
7.28.5.15	kEnableError	159
7.28.5.16	kEnableErrorBypass	159
7.28.5.17	kEnableErrorMark	159
7.28.5.18	kEnableGlobalHeader	159
7.28.5.19	kEnableGlobalTrailer	159
7.28.5.20	kEnableJTAGReadout	159
7.28.5.21	kEnableLocalHeader	159
7.28.5.22	kEnableLocalTrailer	159
7.28.5.23	kEnableMasterResetCode	159
7.28.5.24	kEnableMasterResetOnEventReset	159
7.28.5.25	kEnableMatching	159
7.28.5.26	kEnableOverflowDetect	159
7.28.5.27	kEnablePair	159
7.28.5.28	kEnableReadoutOccupancy	159
7.28.5.29	kEnableReadoutSeparator	159
7.28.5.30	kEnableRelative	159
7.28.5.31	kEnableResetChannelBufferWhenSeparator	159
7.28.5.32	kEnableSeparatorOnBunchReset	159
7.28.5.33	kEnableSeparatorOnEventReset	159
7.28.5.34	kEnableSerial	160
7.28.5.35	kEnableSetCountersOnBunchReset	160
7.28.5.36	kEnableTTLClock	160
7.28.5.37	kEnableTTLControl	160
7.28.5.38	kEnableTTLHit	160
7.28.5.39	kEnableTTLReset	160
7.28.5.40	kEnableTTLSerial	160
7.28.5.41	kEventCountOffset	160
7.28.5.42	kIOClockDelay	160
7.28.5.43	kIOClockSource	160
7.28.5.44	kKeepToken	160

7.28.5.45	kLeading	160
7.28.5.46	kLeadingResolution	160
7.28.5.47	kLowPowerMode	160
7.28.5.48	kMaster	160
7.28.5.49	kMatchWindow	160
7.28.5.50	kMaxEventSize	160
7.28.5.51	kModeRC	160
7.28.5.52	kModeRCCompression	160
7.28.5.53	kOffset0	160
7.28.5.54	kPLLControl	160
7.28.5.55	kRCAdjust0	160
7.28.5.56	kReadoutFIFOSize	160
7.28.5.57	kReadoutSingleCycleSpeed	160
7.28.5.58	kReadoutSpeedSelect	160
7.28.5.59	kRejectCountOffset	160
7.28.5.60	kRejectFIFOFull	160
7.28.5.61	kRollOver	160
7.28.5.62	kSearchWindow	161
7.28.5.63	kSelectBypassInputs	161
7.28.5.64	kSerialClockDelay	161
7.28.5.65	kSerialClockSource	161
7.28.5.66	kSerialDelay	161
7.28.5.67	kSetupParity	161
7.28.5.68	kStrobeSelect	161
7.28.5.69	kTDCId	161
7.28.5.70	kTestInvert	161
7.28.5.71	kTestMode	161
7.28.5.72	kTestSelect	161
7.28.5.73	kTokenDelay	161
7.28.5.74	kTrailing	161
7.28.5.75	kTriggerCountOffset	161
7.28.5.76	kVernierOffset	161
7.28.5.77	kWidthSelect	161
7.29	TDCStatus Class Reference	161
7.29.1	Detailed Description	163
7.29.2	Constructor & Destructor Documentation	163
7.29.2.1	TDCStatus()	163
7.29.2.2	TDCStatus(const TDCStatus &s)	163
7.29.2.3	TDCStatus(const std::vector< uint8_t > &words)	164
7.29.3	Member Function Documentation	164

7.29.3.1	DLLLock() const	164
7.29.3.2	Dump(int verb=1, std::ostream &os=std::cout) const	164
7.29.3.3	Error() const	165
7.29.3.4	FIFOEmpty() const	165
7.29.3.5	FIFOFull() const	165
7.29.3.6	FIFOOccupancy() const	166
7.29.3.7	HaveToken() const	166
7.29.3.8	L1Occupancy() const	166
7.29.3.9	SetConstantValues()	166
7.29.3.10	TriggerFIFOEmpty() const	167
7.29.3.11	TriggerFIFOFull() const	167
7.29.3.12	TriggerFIFOOccupancy() const	167
7.29.4	Field Documentation	167
7.29.4.1	kDLLLock	167
7.29.4.2	kError	167
7.29.4.3	kHaveToken	167
7.29.4.4	kL1Occupancy	167
7.29.4.5	kReadoutFIFOEmpty	167
7.29.4.6	kReadoutFIFOFull	168
7.29.4.7	kReadoutFIFOOccupancy	168
7.29.4.8	kTriggerFIFOEmpty	168
7.29.4.9	kTriggerFIFOFull	168
7.29.4.10	kTriggerFIFOOccupancy	168
7.30	DAQ::QuickUSBHandler::Version Struct Reference	168
7.30.1	Field Documentation	168
7.30.1.1	BuildVersion	168
7.30.1.2	MajorVersion	168
7.30.1.3	MinorVersion	168



# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Socket communication objects . . . . .	9
FPGA board control . . . . .	10
HPTDC chip control . . . . .	11



## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">DAQ</a> . . . . .	<a href="#">13</a>
<a href="#">DQM</a> . . . . .	<a href="#">13</a>



## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

OnlineDBHandler::BurstInfo . . . . .	15
DQM::GastofCanvas::Coord . . . . .	21
DQM::QuarticCanvas::Coord . . . . .	22
file_header_t . . . . .	26
FileReader . . . . .	27
Logger . . . . .	40
LogRedirector . . . . .	41
Message . . . . .	42
SocketMessage . . . . .	72
OnlineDBHandler . . . . .	51
DAQ::QuickUSBHandler . . . . .	62
DAQ::FPGAHandler . . . . .	30
Socket . . . . .	65
Client . . . . .	15
DAQ::FPGAHandler . . . . .	30
DQM::DQMProcess . . . . .	22
Messenger . . . . .	44
TCanvas . . . . .	
DQM::GastofCanvas . . . . .	36
DQM::PPSCanvas . . . . .	55
DQM::QuarticCanvas . . . . .	58
DAQ::TDC . . . . .	80
OnlineDBHandler::TDCConditions . . . . .	86
TDCErrorFlag . . . . .	94
TDCEvent . . . . .	95
TDCMeasurement . . . . .	102
TDCRegister . . . . .	105
TDCBoundaryScan . . . . .	83
TDCControl . . . . .	86
TDCSetup . . . . .	109
TDCStatus . . . . .	161
DAQ::QuickUSBHandler::Version . . . . .	168



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">OnlineDBHandler::BurstInfo</a>	15
<a href="#">Client</a>	
Base client object for the socket	15
<a href="#">DQM::GastofCanvas::Coord</a>	21
<a href="#">DQM::QuarticCanvas::Coord</a>	22
<a href="#">DQM::DQMProcess</a>	
Handler for a common <a href="#">DQM</a> process to run on the socket	22
<a href="#">file_header_t</a>	
Header to the output files	26
<a href="#">FileReader</a>	
Handler for a TDC output file readout	27
<a href="#">DAQ::FPGAHandler</a>	
Driver for timing detectors' FPGA readout	30
<a href="#">DQM::GastofCanvas</a>	36
<a href="#">Logger</a>	
Redirect outputs to another output stream	40
<a href="#">LogRedirector</a>	
Redirect output stream to a string	41
<a href="#">Message</a>	
Base socket message type	42
<a href="#">Messenger</a>	
Base master object for the socket	44
<a href="#">OnlineDBHandler</a>	
Handler for the run information online database	51
<a href="#">DQM::PPSCanvas</a>	55
<a href="#">DQM::QuarticCanvas</a>	58
<a href="#">DAQ::QuickUSBHandler</a>	
Generic QuickUSB communication handler	62
<a href="#">Socket</a>	
Base socket object from which clients/master from a socket inherit	65
<a href="#">SocketMessage</a>	
Socket-passed message type	72
<a href="#">DAQ::TDC</a>	
HPTDC object	80
<a href="#">TDCBoundaryScan</a>	83
<a href="#">OnlineDBHandler::TDCConditions</a>	86
<a href="#">TDCControl</a>	
Control word to be sent to the HPTDC chip	86

<a href="#">TDCErrorFlag</a>	
Error flags handler . . . . .	94
<a href="#">TDCEvent</a>	
HPTDC event parser . . . . .	95
<a href="#">TDCMeasurement</a> . . . . .	102
<a href="#">TDCRegister</a>	
General register object to interact with a HPTDC chip . . . . .	105
<a href="#">TDCSetup</a>	
Setup word to be sent to the HPTDC chip . . . . .	109
<a href="#">TDCStatus</a> . . . . .	161
<a href="#">DAQ::QuickUSBHandler::Version</a> . . . . .	168



## Chapter 5

# Module Documentation

### 5.1 Socket communication objects

#### Data Structures

- class [Client](#)  
*Base client object for the socket.*
- class [Messenger](#)  
*Base master object for the socket.*
- class [Socket](#)  
*Base socket object from which clients/master from a socket inherit.*
- class [SocketMessage](#)  
*Socket-passed message type.*

#### 5.1.1 Detailed Description

## 5.2 FPGA board control

### Data Structures

- class [DAQ::FPGAHandler](#)  
*Driver for timing detectors' FPGA readout.*
- struct [DAQ::QuickUSBHandler::Version](#)
- class [DAQ::QuickUSBHandler](#)  
*Generic QuickUSB communication handler.*

### 5.2.1 Detailed Description

## 5.3 HPTDC chip control

### Data Structures

- class [TDCErrorFlag](#)  
*Error flags handler.*
- class [TDCEvent](#)  
*HPTDC event parser.*
- class [DAQ::TDC](#)  
*HPTDC object.*
- class [TDCBoundaryScan](#)
- class [TDCControl](#)  
*Control word to be sent to the HPTDC chip.*
- class [TDCRegister](#)  
*General register object to interact with a HPTDC chip.*
- class [TDCSetup](#)  
*Setup word to be sent to the HPTDC chip.*
- class [TDCStatus](#)

### Enumerations

- enum [AcquisitionMode](#) { [CONT\\_STORAGE](#), [TRIG\\_MATCH](#) }  
*TDC acquisition mode.*
- enum [DAQ::TDC::AcquisitionMode](#) { [DAQ::TDC::CONT\\_STORAGE](#), [DAQ::TDC::TRIG\\_MATCH](#) }  
*TDC acquisition mode.*

#### 5.3.1 Detailed Description

#### 5.3.2 Enumeration Type Documentation

##### 5.3.2.1 enum AcquisitionMode

TDC acquisition mode.

Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

Enumerator

***CONT\_STORAGE***  
***TRIG\_MATCH***

##### 5.3.2.2 enum DAQ::TDC::AcquisitionMode

[TDC](#) acquisition mode.

Enumerator

***CONT\_STORAGE***  
***TRIG\_MATCH***



## Chapter 6

# Namespace Documentation

### 6.1 DAQ Namespace Reference

#### Data Structures

- class [FPGAHandler](#)  
*Driver for timing detectors' FPGA readout.*
- class [QuickUSBHandler](#)  
*Generic QuickUSB communication handler.*
- class [TDC](#)  
*HPTDC object.*

### 6.2 DQM Namespace Reference

#### Data Structures

- class [DQMProcess](#)  
*Handler for a common [DQM](#) process to run on the socket.*
- class [GastofCanvas](#)
- class [PPSCanvas](#)
- class [QuarticCanvas](#)



## Chapter 7

# Data Structure Documentation

### 7.1 OnlineDBHandler::BurstInfo Struct Reference

```
#include <OnlineDBHandler.h>
```

#### Data Fields

- unsigned int [burst\\_id](#)
- unsigned int [time\\_start](#)

#### 7.1.1 Field Documentation

7.1.1.1 unsigned int OnlineDBHandler::BurstInfo::burst\_id

7.1.1.2 unsigned int OnlineDBHandler::BurstInfo::time\_start

The documentation for this struct was generated from the following file:

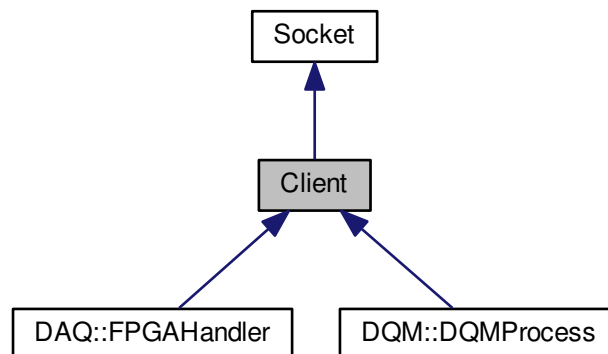
- include/OnlineDBHandler.h

### 7.2 Client Class Reference

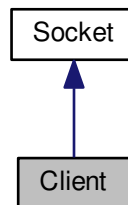
Base client object for the socket.

```
#include <Client.h>
```

Inheritance diagram for Client:



Collaboration diagram for Client:



## Public Member Functions

- [Client](#) ()  
*General void client constructor.*
- [Client](#) (int port)  
*Bind a socket client to a given port.*
- virtual [~Client](#) ()
- bool [Connect](#) (const [SocketType](#) &type=CLIENT)  
*Bind this client to the socket.*
- void [Disconnect](#) ()  
*Unbind this client from the socket.*
- void [Send](#) (const [Message](#) &m) const  
*Send a message to the master through the socket.*
- void [Send](#) (const Exception &e) const
- [SocketMessage](#) [SendAndReceive](#) (const [SocketMessage](#) &m, const MessageKey &a) const
- void [Receive](#) ()



*Receive a socket message from the master.*

- [SocketMessage Receive](#) (const MessageKey &key)
- virtual void [ParseMessage](#) (const [SocketMessage](#) &m)

*Parse a [SocketMessage](#) received from the master.*

- virtual [SocketType GetType](#) () const

*[Socket](#) actor type retrieval method.*

## Private Member Functions

- void [Announce](#) ()

*Announce our entry on the socket to its master.*

## Private Attributes

- int [fClientId](#)
- bool [fIsConnected](#)
- [SocketType](#) [fType](#)

## Additional Inherited Members

### 7.2.1 Detailed Description

Base client object for the socket.

[Client](#) object used by the server to send/receive commands from the messenger/broadcaster.

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

24 Mar 2015

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 [Client::Client](#) ( ) [inline]

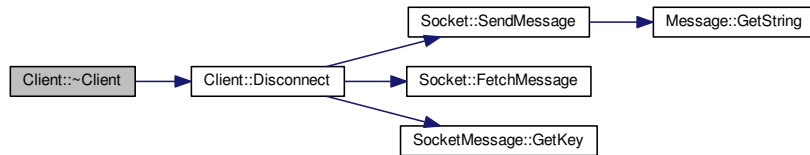
General void client constructor.

#### 7.2.2.2 [Client::Client](#) ( int *port* )

Bind a socket client to a given port.

### 7.2.2.3 Client::~~Client ( ) [virtual]

Here is the call graph for this function:

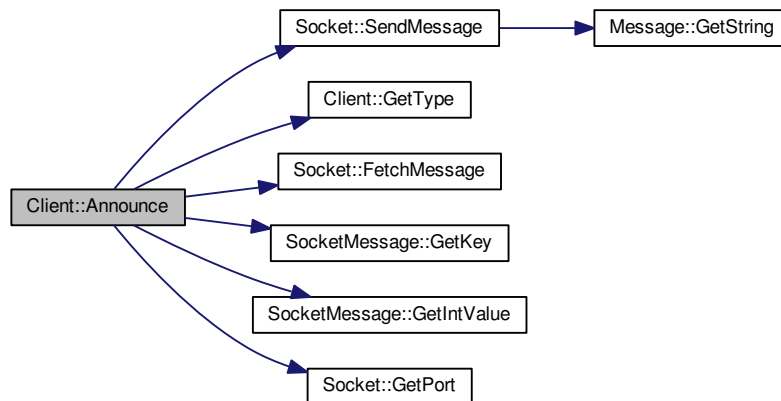


## 7.2.3 Member Function Documentation

### 7.2.3.1 void Client::Announce ( ) [private]

Announce our entry on the socket to its master.

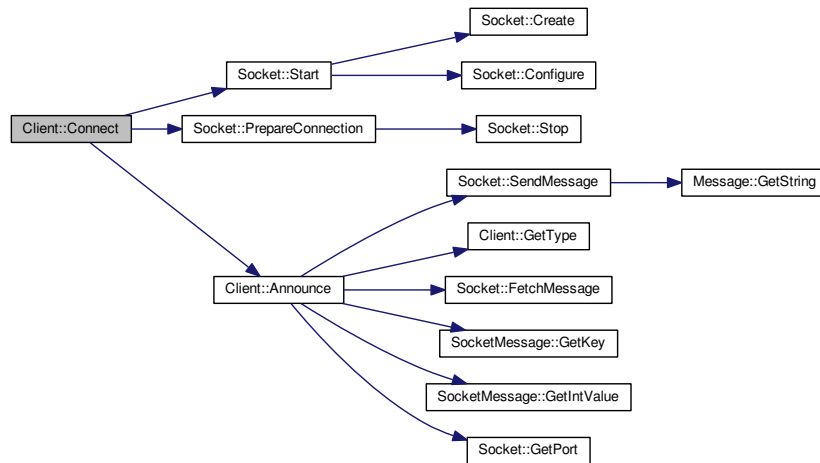
Here is the call graph for this function:



### 7.2.3.2 bool Client::Connect ( const SocketType & type = CLIENT )

Bind this client to the socket.

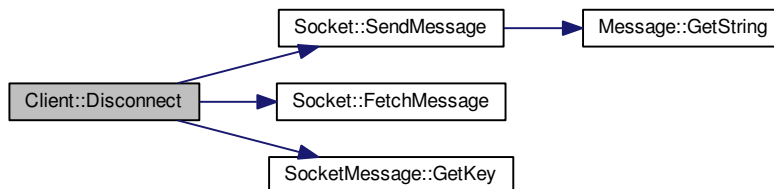
Here is the call graph for this function:



#### 7.2.3.3 void Client::Disconnect ( )

Unbind this client from the socket.

Here is the call graph for this function:



#### 7.2.3.4 virtual SocketType Client::GetType ( ) const [inline],[virtual]

[Socket](#) actor type retrieval method.

Reimplemented in [DAQ::FPGAHandler](#).

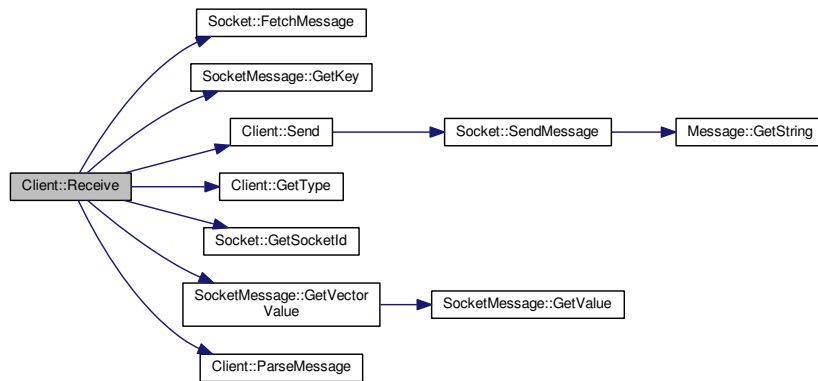
#### 7.2.3.5 virtual void Client::ParseMessage ( const SocketMessage & m ) [inline],[virtual]

Parse a [SocketMessage](#) received from the master.

#### 7.2.3.6 void Client::Receive ( )

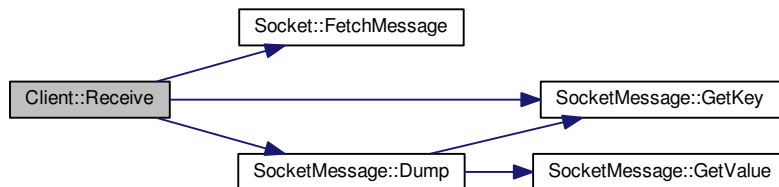
Receive a socket message from the master.

Here is the call graph for this function:



#### 7.2.3.7 SocketMessage Client::Receive ( const MessageKey & key )

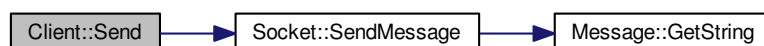
Here is the call graph for this function:



#### 7.2.3.8 void Client::Send ( const Message & m ) const [inline]

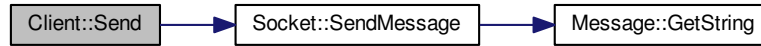
Send a message to the master through the socket.

Here is the call graph for this function:



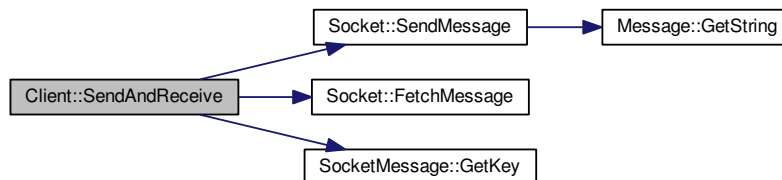
7.2.3.9 `void Client::Send ( const Exception & e ) const` `[inline]`

Here is the call graph for this function:



7.2.3.10 `SocketMessage Client::SendAndReceive ( const SocketMessage & m, const MessageKey & a ) const` `[inline]`

Here is the call graph for this function:



## 7.2.4 Field Documentation

7.2.4.1 `int Client::fClientId` `[private]`

7.2.4.2 `bool Client::fIsConnected` `[private]`

7.2.4.3 `SocketType Client::fType` `[private]`

The documentation for this class was generated from the following files:

- `include/Client.h`
- `src/Client.cpp`

## 7.3 DQM::GastofCanvas::Coord Struct Reference

### Data Fields

- unsigned int [x](#)
- unsigned int [y](#)

### 7.3.1 Field Documentation

7.3.1.1 unsigned int DQM::GastofCanvas::Coord::x

7.3.1.2 unsigned int DQM::GastofCanvas::Coord::y

The documentation for this struct was generated from the following file:

- include/GastofCanvas.h

## 7.4 DQM::QuarticCanvas::Coord Struct Reference

### Data Fields

- unsigned int [x](#)
- unsigned int [y](#)

### 7.4.1 Field Documentation

7.4.1.1 unsigned int DQM::QuarticCanvas::Coord::x

7.4.1.2 unsigned int DQM::QuarticCanvas::Coord::y

The documentation for this struct was generated from the following file:

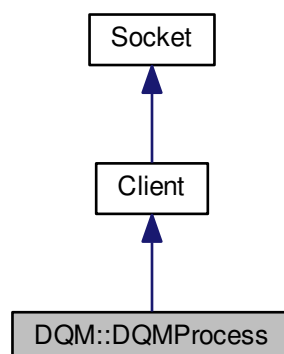
- include/QuarticCanvas.h

## 7.5 DQM::DQMProcess Class Reference

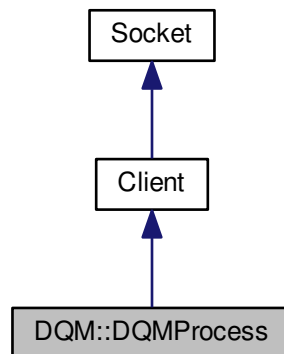
Handler for a common [DQM](#) process to run on the socket.

```
#include <DQMProcess.h>
```

Inheritance diagram for DQM::DQMProcess:



Collaboration diagram for DQM::DQMProcess:



## Public Types

- enum [Action](#) { [NewPlot](#) = 0x0, [UpdatedPlot](#) = 0x1 }

## Public Member Functions

- [DQMProcess](#) (int port, unsigned short order=0, const char \*det\_type="")
- [~DQMProcess](#) ()
- void [Run](#) (bool(\*fcn)(unsigned int addr, std::string filename, std::vector< std::string > \*outputs), const [Action](#) &act=[NewPlot](#))  
*Run a [DQM](#) plotter making use of the board/output filename information.*
- void [Run](#) (bool(\*fcn)(std::vector< std::string > \*outputs), const [Action](#) &act=[NewPlot](#))  
*Run a [DQM](#) plotter without any information on the board/output filename.*

## Private Member Functions

- int [ParseMessage](#) (uint32\_t \*board\_address, std::string \*filename)
- bool [IsInRun](#) ()

## Private Attributes

- unsigned short [fOrder](#)
- unsigned int [fRunNumber](#)
- std::string [fDetectorType](#)
- std::map< unsigned long, std::string > [fAddressesCanProcess](#)

## Additional Inherited Members

### 7.5.1 Detailed Description

Handler for a common [DQM](#) process to run on the socket.

## Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

## Date

3 Aug 2015

## 7.5.2 Member Enumeration Documentation

### 7.5.2.1 enum DQM::DQMProcess::Action

## Enumerator

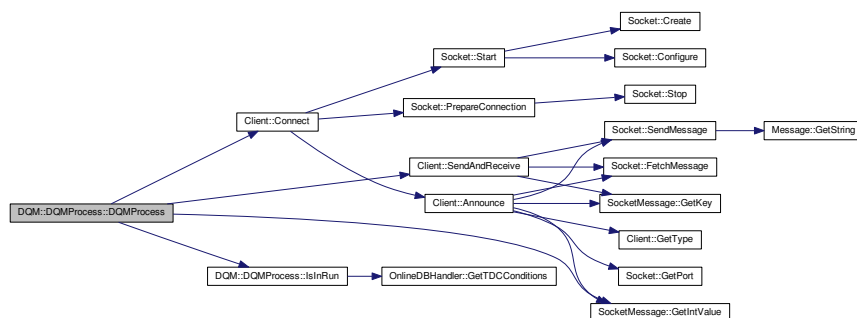
***NewPlot***

***UpdatedPlot***

## 7.5.3 Constructor & Destructor Documentation

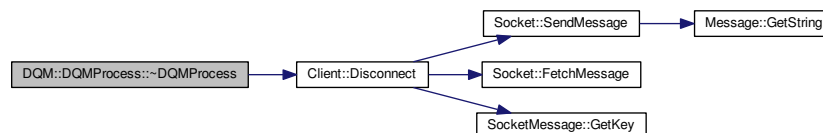
### 7.5.3.1 DQM::DQMProcess::DQMProcess ( int *port*, unsigned short *order* = 0, const char \* *det\_type* = " " ) [inline]

Here is the call graph for this function:



### 7.5.3.2 DQM::DQMProcess::~~DQMProcess ( ) [inline]

Here is the call graph for this function:

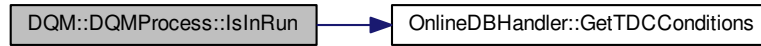


## 7.5.4 Member Function Documentation



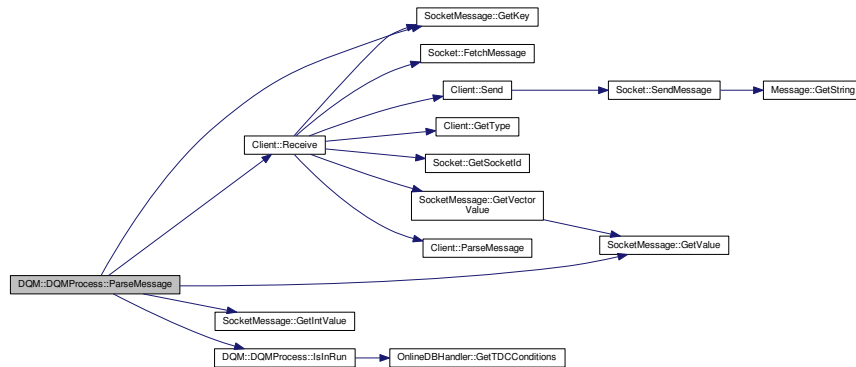
#### 7.5.4.1 bool DQM::DQMProcess::IsInRun ( ) [inline],[private]

Here is the call graph for this function:



#### 7.5.4.2 int DQM::DQMProcess::ParseMessage ( uint32\_t \* board\_address, std::string \* filename ) [inline],[private]

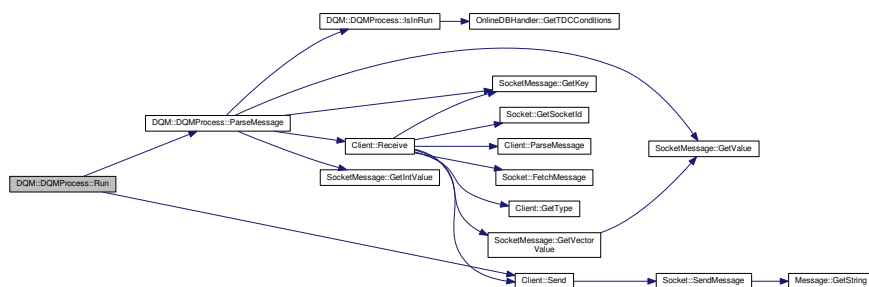
Here is the call graph for this function:



#### 7.5.4.3 void DQM::DQMProcess::Run ( bool(\*)(unsigned int addr, std::string filename, std::vector< std::string > \*outputs) fcn, const Action & act = NewPlot ) [inline]

Run a [DQM](#) plotter making use of the board/output filename information.

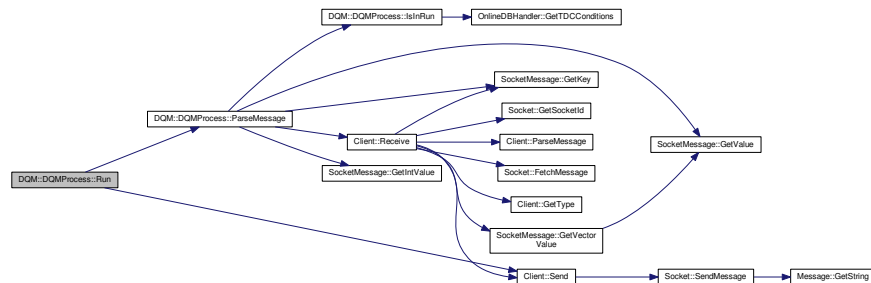
Here is the call graph for this function:



7.5.4.4 `void DQM::DQMProcess::Run ( bool(*) (std::vector< std::string > *outputs) fcn, const Action & act = NewPlot )`  
`[inline]`

Run a [DQM](#) plotter without any information on the board/output filename.

Here is the call graph for this function:



## 7.5.5 Field Documentation

7.5.5.1 `std::map<unsigned long, std::string> DQM::DQMProcess::fAddressesCanProcess` `[private]`

7.5.5.2 `std::string DQM::DQMProcess::fDetectorType` `[private]`

7.5.5.3 `unsigned short DQM::DQMProcess::fOrder` `[private]`

7.5.5.4 `unsigned int DQM::DQMProcess::fRunNumber` `[private]`

The documentation for this class was generated from the following file:

- `include/DQMProcess.h`

## 7.6 file\_header\_t Struct Reference

Header to the output files.

```
#include <FileConstants.h>
```

### Data Fields

- `uint32_t magic`
- `uint32_t run_id`
- `uint32_t spill_id`
- `uint8_t num_hptdc`
- `AcquisitionMode acq_mode`
- `DetectionMode det_mode`

### 7.6.1 Detailed Description

Header to the output files.

General header to store in each collected data file for offline readout. It enable any reader to retrieve the run/spill number, as well as the HPTDC configuration during data collection.

## Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

## Date

14 Apr 2015

## 7.6.2 Field Documentation

7.6.2.1 **AcquisitionMode** `file_header_t::acq_mode`

7.6.2.2 **DetectionMode** `file_header_t::det_mode`

7.6.2.3 **uint32\_t** `file_header_t::magic`

7.6.2.4 **uint8\_t** `file_header_t::num_hptdc`

7.6.2.5 **uint32\_t** `file_header_t::run_id`

7.6.2.6 **uint32\_t** `file_header_t::spill_id`

The documentation for this struct was generated from the following file:

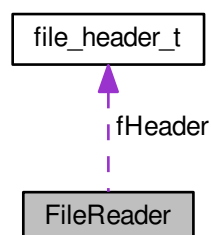
- `include/FileConstants.h`

## 7.7 FileReader Class Reference

Handler for a TDC output file readout.

```
#include <FileReader.h>
```

Collaboration diagram for FileReader:



### Public Member Functions

- [FileReader](#) ()
- [FileReader](#) (std::string name)  
*Class constructor.*
- [~FileReader](#) ()

- void [Open](#) (std::string name)
- bool [IsOpen](#) () const
- void [Clear](#) ()
- void [Dump](#) () const
- unsigned int [GetNumTDCs](#) () const
- unsigned int [GetRunId](#) () const
- unsigned int [GetBurstId](#) () const
- unsigned int [GetAcquisitionMode](#) () const
- unsigned int [GetDetectionMode](#) () const
- unsigned long [GetNumEvents](#) () const
- bool [GetNextEvent](#) (TDCEvent \*)
- bool [GetNextMeasurement](#) (unsigned int channel\_id, [TDCMeasurement](#) \*mc)

*Fetch the next full measurement on a given channel.*

### Private Attributes

- std::ifstream [fFile](#)
- [file\\_header\\_t](#) [fHeader](#)
- [AcquisitionMode](#) [fReadoutMode](#)
- time\_t [fWriteTime](#)
- unsigned long [fNumEvents](#)

### 7.7.1 Detailed Description

Handler for a TDC output file readout.

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

Jun 2015

### 7.7.2 Constructor & Destructor Documentation

#### 7.7.2.1 [FileReader::FileReader](#) ( ) [inline]

#### 7.7.2.2 [FileReader::FileReader](#) ( std::string name )

Class constructor.

#### Parameters

in	<i>name</i>	Path to the file to read
in	<i>ro</i>	Data readout mode (continuous storage or trigger matching)

Here is the call graph for this function:



## 7.7.2.3 FileReader::~FileReader ( )

## 7.7.3 Member Function Documentation

## 7.7.3.1 void FileReader::Clear ( ) [inline]

## 7.7.3.2 void FileReader::Dump ( ) const

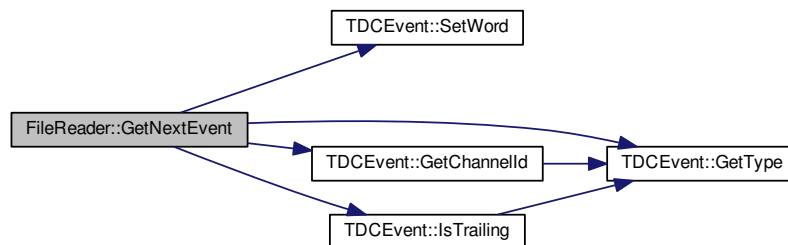
## 7.7.3.3 unsigned int FileReader::GetAcquisitionMode ( ) const [inline]

## 7.7.3.4 unsigned int FileReader::GetBurstId ( ) const [inline]

## 7.7.3.5 unsigned int FileReader::GetDetectionMode ( ) const [inline]

## 7.7.3.6 bool FileReader::GetNextEvent ( TDCEvent \* ev )

Here is the call graph for this function:

7.7.3.7 bool FileReader::GetNextMeasurement ( unsigned int *channel\_id*, TDCMeasurement \* *mc* )

Fetch the next full measurement on a given channel.

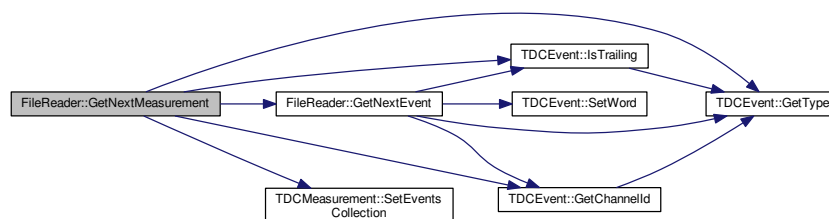
## Parameters

in	<i>channel_id</i>	Unique identifier of the channel number to retrieve
out	<i>m</i>	A full measurement with leading, trailing times, ...

## Returns

A boolean stating the success of retrieval operation

Here is the call graph for this function:



7.7.3.8 `unsigned long FileReader::GetNumEvents ( ) const` `[inline]`

7.7.3.9 `unsigned int FileReader::GetNumTDCs ( ) const` `[inline]`

7.7.3.10 `unsigned int FileReader::GetRunId ( ) const` `[inline]`

7.7.3.11 `bool FileReader::IsOpen ( ) const` `[inline]`

7.7.3.12 `void FileReader::Open ( std::string name )`

## 7.7.4 Field Documentation

7.7.4.1 `std::ifstream FileReader::fFile` `[private]`

7.7.4.2 `file_header_t FileReader::fHeader` `[private]`

7.7.4.3 `unsigned long FileReader::fNumEvents` `[private]`

7.7.4.4 `AcquisitionMode FileReader::fReadoutMode` `[private]`

7.7.4.5 `time_t FileReader::fWriteTime` `[private]`

The documentation for this class was generated from the following files:

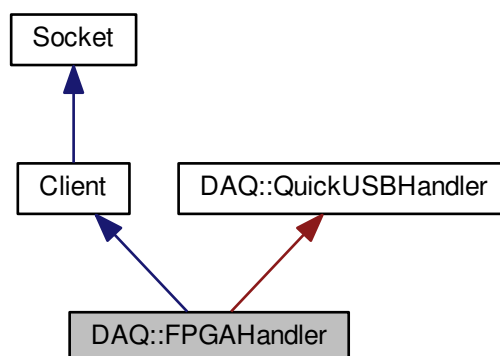
- `include/FileReader.h`
- `src/FileReader.cpp`

## 7.8 DAQ::FPGAHandler Class Reference

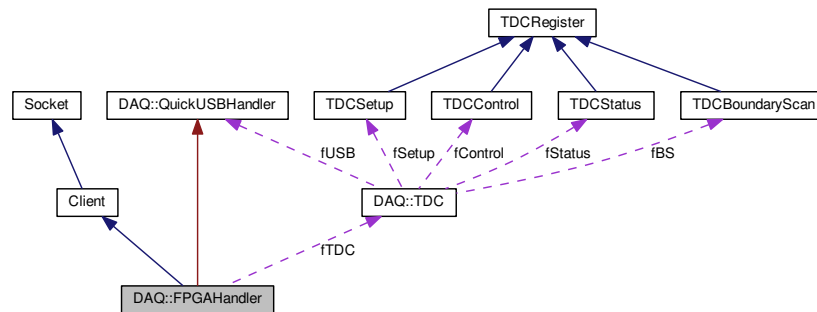
Driver for timing detectors' FPGA readout.

```
#include <FPGAHandler.h>
```

Inheritance diagram for DAQ::FPGAHandler:



Collaboration diagram for DAQ::FPGAHandler:



## Public Member Functions

- [FPGAHandler](#) (int port, const char \*dev)  
*Bind to a FPGA through the USB protocol, and to the socket.*
- [~FPGAHandler](#) ()
- void [Stop](#) ()
- void [OpenFile](#) ()  
*Open an output file to store header/HPTDC events.*
- void [CloseFile](#) ()  
*Close a previously opened output file used to store header/HPTDC events.*
- std::string [GetFilename](#) () const  
*Retrieve the file name used to store data collected from the FPGA.*
- [TDC \\* GetTDC](#) (unsigned int i=0)
- void [SetTDCSetup](#) (const [TDCSetup](#) &s)
- bool [ErrorState](#) ()
- void [StartAcquisition](#) ()
- void [StopAcquisition](#) ()
- [SocketType GetType](#) () const  
*Socket actor type retrieval method.*
- [TDCControl GetTDCControl](#) () const
- [TDCStatus GetTDCStatus](#) () const

## Private Member Functions

- void [RegisterTest](#) () const
- void [SendSetupWord](#) () const
- void [RetrieveSetupWord](#) () const

## Private Attributes

- std::string [fFilename](#)
- std::ofstream [fOutput](#)
- bool [flsFileOpen](#)
- [TDC \\* fTDC](#) [NUM\_HPTDC]
- bool [flsTDCInReadout](#)

## Additional Inherited Members

### 7.8.1 Detailed Description

Driver for timing detectors' FPGA readout.

Main driver for a homebrew FPGA designed for the timing detectors' HPTDC chip readout.

Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

Date

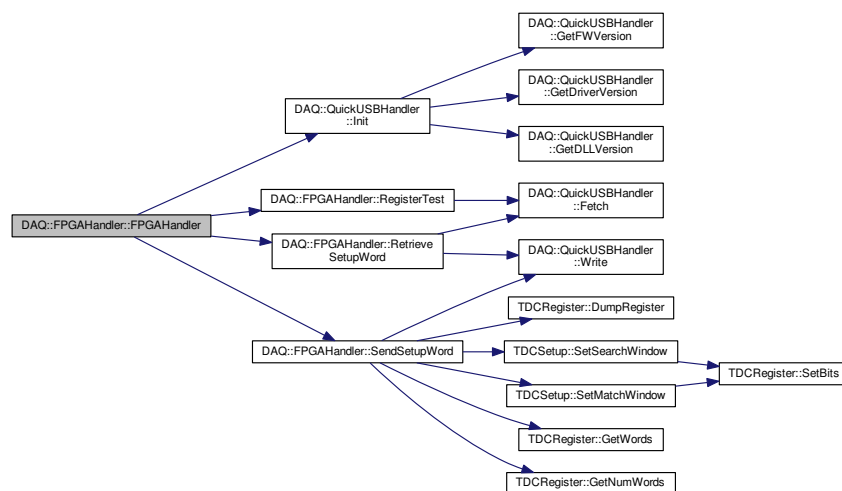
14 Apr 2015

### 7.8.2 Constructor & Destructor Documentation

#### 7.8.2.1 DAQ::FPGAHandler::FPGAHandler ( int *port*, const char \* *dev* )

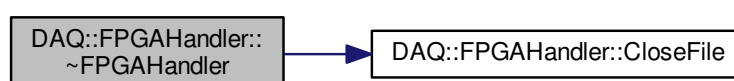
Bind to a FPGA through the USB protocol, and to the socket.

Here is the call graph for this function:



#### 7.8.2.2 DAQ::FPGAHandler::~~FPGAHandler ( )

Here is the call graph for this function:





### 7.8.3 Member Function Documentation

#### 7.8.3.1 void DAQ::FPGAHandler::CloseFile ( )

Close a previously opened output file used to store header/HPTDC events.

#### 7.8.3.2 bool DAQ::FPGAHandler::ErrorState ( )

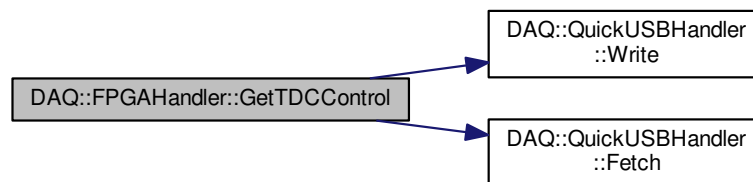
#### 7.8.3.3 std::string DAQ::FPGAHandler::GetFilename ( ) const [inline]

Retrieve the file name used to store data collected from the FPGA.

#### 7.8.3.4 TDC\* DAQ::FPGAHandler::GetTDC ( unsigned int $i=0$ ) [inline]

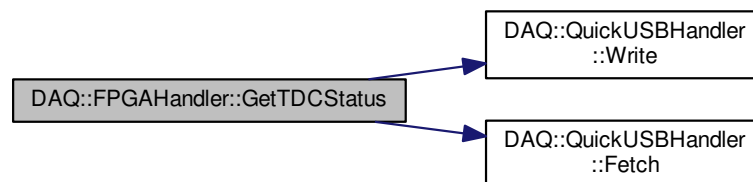
#### 7.8.3.5 TDCControl DAQ::FPGAHandler::GetTDCControl ( ) const

Here is the call graph for this function:



#### 7.8.3.6 TDCStatus DAQ::FPGAHandler::GetTDCStatus ( ) const

Here is the call graph for this function:



#### 7.8.3.7 SocketType DAQ::FPGAHandler::GetType ( ) const [inline],[virtual]

[Socket](#) actor type retrieval method.

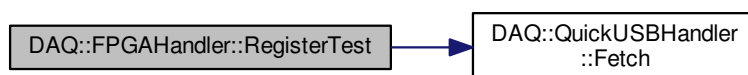
Reimplemented from [Client](#).

### 7.8.3.8 void DAQ::FPGAHandler::OpenFile ( )

Open an output file to store header/HPTDC events.

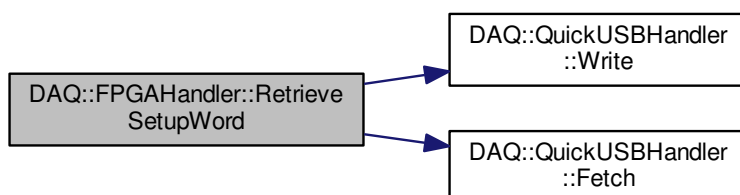
### 7.8.3.9 void DAQ::FPGAHandler::RegisterTest ( ) const [private]

Here is the call graph for this function:



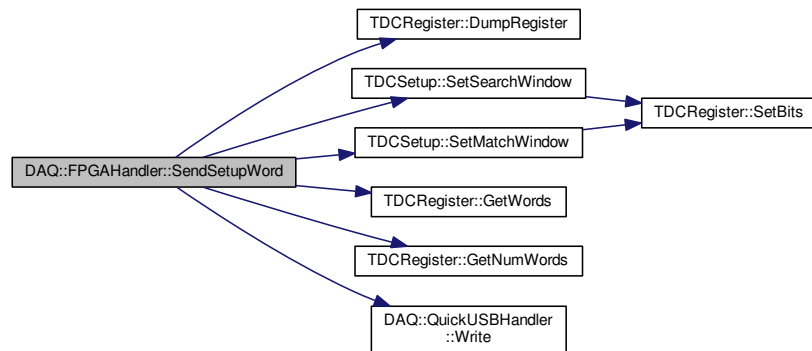
### 7.8.3.10 void DAQ::FPGAHandler::RetrieveSetupWord ( ) const [private]

Here is the call graph for this function:



### 7.8.3.11 void DAQ::FPGAHandler::SendSetupWord ( ) const [private]

Here is the call graph for this function:



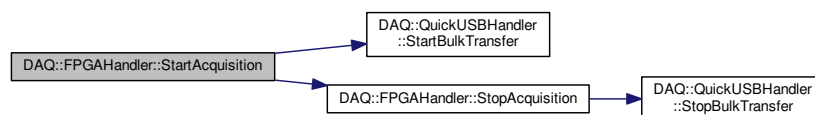
### 7.8.3.12 void DAQ::FPGAHandler::SetTDCSetup ( const TDCSetup & s ) [inline]

Here is the call graph for this function:



### 7.8.3.13 void DAQ::FPGAHandler::StartAcquisition ( )

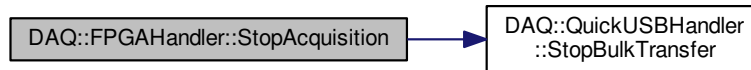
Here is the call graph for this function:



### 7.8.3.14 void DAQ::FPGAHandler::Stop ( ) [inline]

#### 7.8.3.15 void DAQ::FPGAHandler::StopAcquisition ( )

Here is the call graph for this function:



### 7.8.4 Field Documentation

7.8.4.1 `std::string DAQ::FPGAHandler::fFilename` [private]

7.8.4.2 `bool DAQ::FPGAHandler::fIsFileOpen` [private]

7.8.4.3 `bool DAQ::FPGAHandler::fIsTDCInReadout` [private]

7.8.4.4 `std::ofstream DAQ::FPGAHandler::fOutput` [private]

7.8.4.5 `TDC* DAQ::FPGAHandler::fTDC[NUM_HPTDC]` [private]

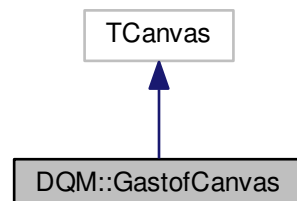
The documentation for this class was generated from the following files:

- `daq/include/FPGAHandler.h`
- `daq/src/FPGAHandler.cpp`

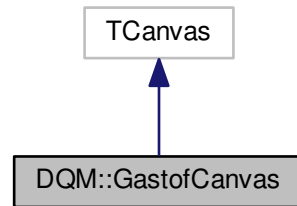
## 7.9 DQM::GastofCanvas Class Reference

```
#include <GastofCanvas.h>
```

Inheritance diagram for `DQM::GastofCanvas`:



Collaboration diagram for DQM::GastofCanvas:



## Data Structures

- struct [Coord](#)

## Public Member Functions

- [GastofCanvas](#) ()
- [GastofCanvas](#) (TString name, unsigned int width=500, unsigned int height=500, TString upper\_label="")
- [GastofCanvas](#) (TString name, TString upper\_label)
- virtual [~GastofCanvas](#) ()
- void [SetRunInfo](#) (unsigned int board\_id, unsigned int run\_id, unsigned int spill\_id, TString date)
- void [SetUpperLabel](#) (TString text)
- void [FillChannel](#) (unsigned short nino\_id, unsigned short channel\_id, double content)
- TH2D \* [Grid](#) ()
- void [Save](#) (TString ext="png", TString path=".")

## Private Member Functions

- void [Build](#) ()
- void [DrawGrid](#) ()
- [Coord](#) [GetCoordinates](#) (unsigned short nino\_id, unsigned short channel\_id) const

## Private Attributes

- TPad \* [c1](#)
- TPad \* [c2](#)
- TH2D \* [fHist](#)
- double [fWidth](#)
- double [fHeight](#)
- TLegend \* [fLegend](#)
- double [fLegendX](#)
- double [fLegendY](#)
- unsigned int [fLegendNumEntries](#)
- TPaveText \* [fLabel1](#)
- TPaveText \* [fLabel2](#)
- TPaveText \* [fLabel3](#)

- TPaveText \* [fLabel4](#)
- TString [fUpperLabelText](#)
- TPaveText \* [fUpperLabel](#)
- bool [fLabelsDrawn](#)
- unsigned int [fBoardId](#)
- unsigned int [fRunId](#)
- unsigned int [fSpillId](#)
- TString [fRunDate](#)

### 7.9.1 Detailed Description

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

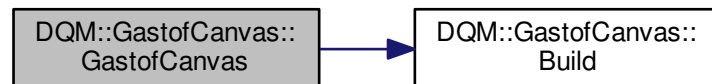
25 Jul 2015

### 7.9.2 Constructor & Destructor Documentation

7.9.2.1 DQM::GastofCanvas::GastofCanvas ( ) [inline]

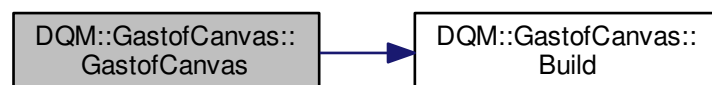
7.9.2.2 DQM::GastofCanvas::GastofCanvas ( TString *name*, unsigned int *width* = 500, unsigned int *height* = 500, TString *upper\_label* = " " ) [inline]

Here is the call graph for this function:



7.9.2.3 DQM::GastofCanvas::GastofCanvas ( TString *name*, TString *upper\_label* ) [inline]

Here is the call graph for this function:



7.9.2.4 `virtual DQM::GastofCanvas::~~GastofCanvas ( ) [inline],[virtual]`

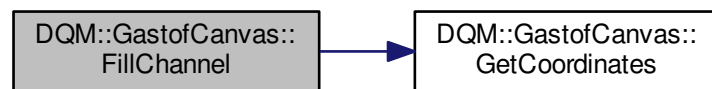
### 7.9.3 Member Function Documentation

7.9.3.1 `void DQM::GastofCanvas::Build ( ) [inline],[private]`

7.9.3.2 `void DQM::GastofCanvas::DrawGrid ( ) [inline],[private]`

7.9.3.3 `void DQM::GastofCanvas::FillChannel ( unsigned short nino_id, unsigned short channel_id, double content ) [inline]`

Here is the call graph for this function:

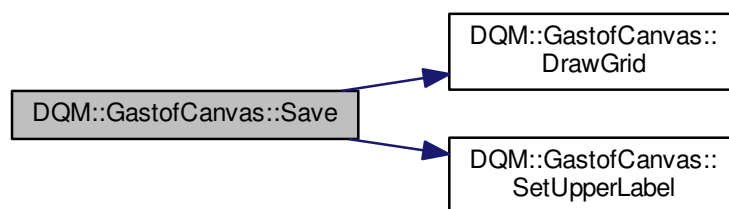


7.9.3.4 `Coord DQM::GastofCanvas::GetCoordinates ( unsigned short nino_id, unsigned short channel_id ) const [inline],[private]`

7.9.3.5 `TH2D* DQM::GastofCanvas::Grid ( ) [inline]`

7.9.3.6 `void DQM::GastofCanvas::Save ( TString ext = "png", TString path = " . " ) [inline]`

Here is the call graph for this function:



7.9.3.7 `void DQM::GastofCanvas::SetRunInfo ( unsigned int board_id, unsigned int run_id, unsigned int spill_id, TString date ) [inline]`

7.9.3.8 `void DQM::GastofCanvas::SetUpperLabel ( TString text ) [inline]`

### 7.9.4 Field Documentation

- 7.9.4.1 TPad\* DQM::GastofCanvas::c1 [private]
- 7.9.4.2 TPad \* DQM::GastofCanvas::c2 [private]
- 7.9.4.3 unsigned int DQM::GastofCanvas::fBoardId [private]
- 7.9.4.4 double DQM::GastofCanvas::fHeight [private]
- 7.9.4.5 TH2D\* DQM::GastofCanvas::fHist [private]
- 7.9.4.6 TPaveText\* DQM::GastofCanvas::fLabel1 [private]
- 7.9.4.7 TPaveText \* DQM::GastofCanvas::fLabel2 [private]
- 7.9.4.8 TPaveText \* DQM::GastofCanvas::fLabel3 [private]
- 7.9.4.9 TPaveText \* DQM::GastofCanvas::fLabel4 [private]
- 7.9.4.10 bool DQM::GastofCanvas::fLabelsDrawn [private]
- 7.9.4.11 TLegend\* DQM::GastofCanvas::fLegend [private]
- 7.9.4.12 unsigned int DQM::GastofCanvas::fLegendNumEntries [private]
- 7.9.4.13 double DQM::GastofCanvas::fLegendX [private]
- 7.9.4.14 double DQM::GastofCanvas::fLegendY [private]
- 7.9.4.15 TString DQM::GastofCanvas::fRunDate [private]
- 7.9.4.16 unsigned int DQM::GastofCanvas::fRunId [private]
- 7.9.4.17 unsigned int DQM::GastofCanvas::fSpillId [private]
- 7.9.4.18 TPaveText\* DQM::GastofCanvas::fUpperLabel [private]
- 7.9.4.19 TString DQM::GastofCanvas::fUpperLabelText [private]
- 7.9.4.20 double DQM::GastofCanvas::fWidth [private]

The documentation for this class was generated from the following file:

- include/GastofCanvas.h

## 7.10 Logger Class Reference

Redirect outputs to another output stream.

```
#include <FileConstants.h>
```

### Public Member Functions

- [Logger](#) (std::ostream &lhs, std::ostream &rhs=std::cout)
- [~Logger](#) ()



## Private Attributes

- `std::ostream & fStream`
- `std::streambuf *const fBuffer`

### 7.10.1 Detailed Description

Redirect outputs to another output stream.

### 7.10.2 Constructor & Destructor Documentation

7.10.2.1 `Logger::Logger ( std::ostream & lhs, std::ostream & rhs = std::cout ) [inline]`

7.10.2.2 `Logger::~Logger ( ) [inline]`

### 7.10.3 Field Documentation

7.10.3.1 `std::streambuf* const Logger::fBuffer [private]`

7.10.3.2 `std::ostream& Logger::fStream [private]`

The documentation for this class was generated from the following file:

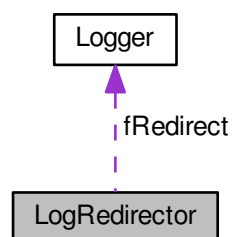
- `include/FileConstants.h`

## 7.11 LogRedirector Class Reference

Redirect output stream to a string.

```
#include <FileConstants.h>
```

Collaboration diagram for LogRedirector:



## Public Member Functions

- `LogRedirector (std::ostream &stm=std::cout)`
- `std::string contents () const`

## Private Attributes

- `std::ostringstream fSS`
- `const Logger fRedirect`

### 7.11.1 Detailed Description

Redirect output stream to a string.

#### Author

Laurent Forthomme `laurent.forthomme@cern.ch`

#### Date

3 Aug 2015

### 7.11.2 Constructor & Destructor Documentation

7.11.2.1 `LogRedirector::LogRedirector ( std::ostream & stm = std::cout ) [inline]`

### 7.11.3 Member Function Documentation

7.11.3.1 `std::string LogRedirector::contents ( ) const [inline]`

### 7.11.4 Field Documentation

7.11.4.1 `const Logger LogRedirector::fRedirect [private]`

7.11.4.2 `std::ostringstream LogRedirector::fSS [private]`

The documentation for this class was generated from the following file:

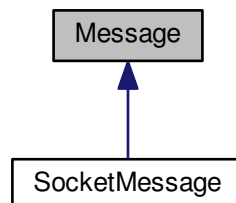
- `include/FileConstants.h`

## 7.12 Message Class Reference

Base socket message type.

```
#include <Message.h>
```

Inheritance diagram for Message:



## Public Member Functions

- [Message](#) ()  
*Void message constructor.*
- [Message](#) (const char \*msg)  
*Construct a message from a string.*
- [Message](#) (std::string msg)  
*Construct a message from a string.*
- virtual [~Message](#) ()
- MessageKey [GetKey](#) () const  
*Placeholder for the MessageKey retrieval method.*
- std::string [GetString](#) () const  
*Retrieve the string carried by this message as a whole.*
- bool [IsFromWeb](#) () const  
*Extract from any message its potential arrival from a WebSocket protocol.*
- void [Dump](#) (std::ostream &os=std::cout) const

## Protected Attributes

- std::string [fString](#)

### 7.12.1 Detailed Description

Base socket message type.

Base handler for messages to be transmitted through the socket

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

6 Apr 2015

### 7.12.2 Constructor & Destructor Documentation

#### 7.12.2.1 [Message::Message](#) ( ) [inline]

Void message constructor.

#### 7.12.2.2 [Message::Message](#) ( const char \* msg ) [inline]

Construct a message from a string.

#### 7.12.2.3 [Message::Message](#) ( std::string msg ) [inline]

Construct a message from a string.

7.12.2.4 `virtual Message::~~Message ( ) [inline],[virtual]`

### 7.12.3 Member Function Documentation

7.12.3.1 `void Message::Dump ( std::ostream & os = std::cout ) const [inline]`

7.12.3.2 `MessageKey Message::GetKey ( ) const [inline]`

Placeholder for the MessageKey retrieval method.

7.12.3.3 `std::string Message::GetString ( ) const [inline]`

Retrieve the string carried by this message as a whole.

7.12.3.4 `bool Message::IsFromWeb ( ) const [inline]`

Extract from any message its potential arrival from a WebSocket protocol.

### 7.12.4 Field Documentation

7.12.4.1 `std::string Message::fString [protected]`

The documentation for this class was generated from the following file:

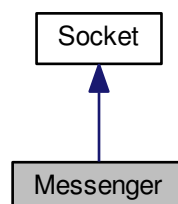
- include/Message.h

## 7.13 Messenger Class Reference

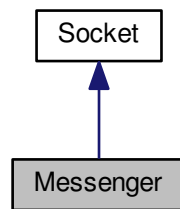
Base master object for the socket.

```
#include <Messenger.h>
```

Inheritance diagram for Messenger:



Collaboration diagram for Messenger:



## Public Member Functions

- [Messenger](#) ()  
*Build a void master object or socket actor.*
- [Messenger](#) (int port)  
*Build a master object to control the socket.*
- [~Messenger](#) ()
- bool [Connect](#) ()  
*Connect the master to the socket.*
- void [Disconnect](#) ()  
*Remove the master and destroy the socket.*
- void [Send](#) (const [Message](#) &m, int sid) const  
*Send any type of message to any client.*
- void [SendAll](#) (const [Socket::SocketType](#) &type, const [Message](#) &m) const  
*Send any type of message to all clients of one type.*
- void [SendAll](#) (const [Socket::SocketType](#) &type, const Exception &e) const
- void [Receive](#) ()  
*Handle a message reception from a client.*
- void [Broadcast](#) (const [Message](#) &m) const  
*Emit a message to all clients connected through the socket.*
- void [StartAcquisition](#) ()  
*Start the data acquisition.*
- void [StopAcquisition](#) ()
- [SocketType](#) [GetType](#) () const  
*[Socket](#) actor type retrieval method.*

## Private Member Functions

- void [AddClient](#) ()  
*Add a client to listen to.*
- void [DisconnectClient](#) (int sid, MessageKey key, bool force=false)  
*Disconnect a client.*
- void [SwitchClientType](#) (int sid, [Socket::SocketType](#) type)
- void [ProcessMessage](#) ([SocketMessage](#) m, int sid)  
*Process a message received from the socket.*

## Private Attributes

- int [fNumAttempts](#)
- pid\_t [fPID](#)
- int [fStdoutPipe](#) [2]
- int [fStderrPipe](#) [2]

## Additional Inherited Members

### 7.13.1 Detailed Description

Base master object for the socket.

Messenger/broadcaster object used by the server to send/receive commands from the clients/listeners.

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

23 Mar 2015

### 7.13.2 Constructor & Destructor Documentation

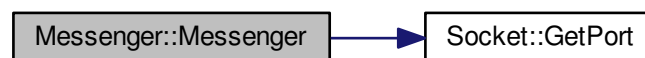
#### 7.13.2.1 Messenger::Messenger ( )

Build a void master object or socket actor.

#### 7.13.2.2 Messenger::Messenger ( int *port* )

Build a master object to control the socket.

Here is the call graph for this function:



#### 7.13.2.3 Messenger::~~Messenger ( )

Here is the call graph for this function:



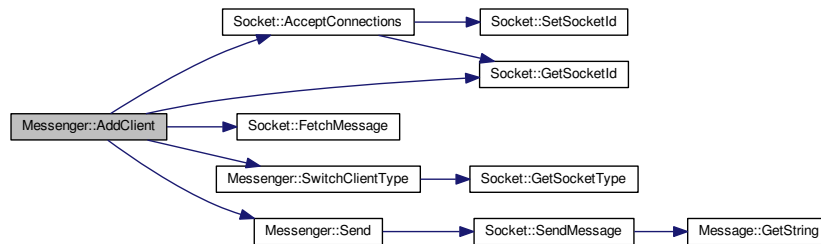
### 7.13.3 Member Function Documentation

#### 7.13.3.1 void Messenger::AddClient ( ) [private]

Add a client to listen to.

Add one client to the list of socket actors to monitor for message retrieval/submission.

Here is the call graph for this function:



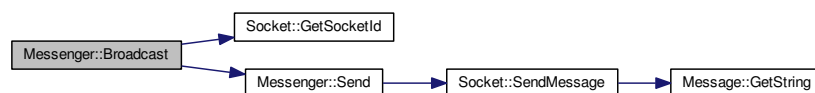
#### 7.13.3.2 void Messenger::Broadcast ( const Message & m ) const

Emit a message to all clients connected through the socket.

Parameters

in	<i>m</i>	Message to transmit
----	----------	---------------------

Here is the call graph for this function:

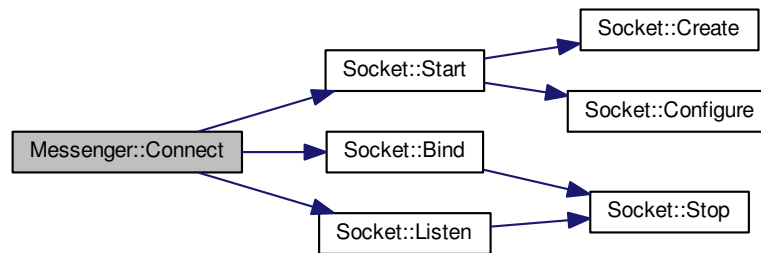


#### 7.13.3.3 bool Messenger::Connect ( )

Connect the master to the socket.

Connect this master to the socket for clients to be able to bind.

Here is the call graph for this function:

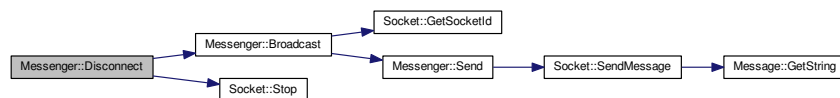


#### 7.13.3.4 void Messenger::Disconnect ( )

Remove the master and destroy the socket.

Remove this master from the socket, thus disconnecting automatically the clients connected.

Here is the call graph for this function:



#### 7.13.3.5 void Messenger::DisconnectClient ( int sid, MessageKey key, bool force = false ) [private]

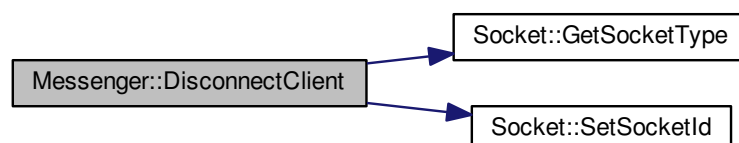
Disconnect a client.

Ask to a client to disconnect from this socket.

##### Parameters

in	<i>sid</i>	Unique identifier of the client to disconnect
in	<i>key</i>	Key to the message to transmit for disconnection
in	<i>force</i>	Do we need to force the client out of this socket ?

Here is the call graph for this function:





### 7.13.3.6 SocketType Messenger::GetType ( ) const [inline]

Socket actor type retrieval method.

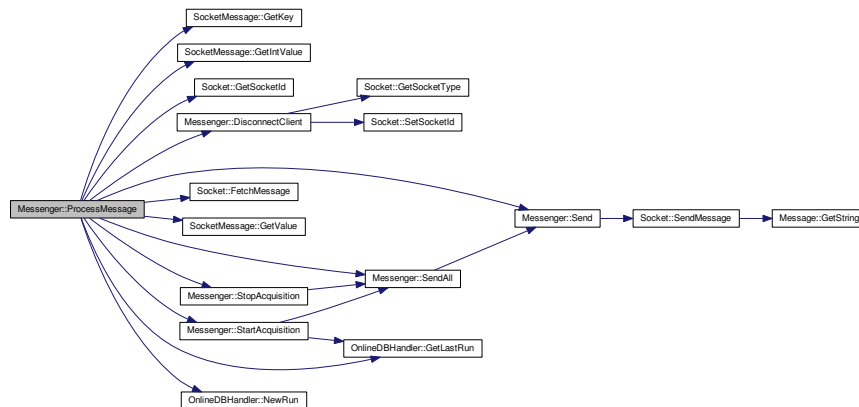
### 7.13.3.7 void Messenger::ProcessMessage ( SocketMessage m, int sid ) [private]

Process a message received from the socket.

Parameters

in	Unique	identifier of the client sending the message
----	--------	--

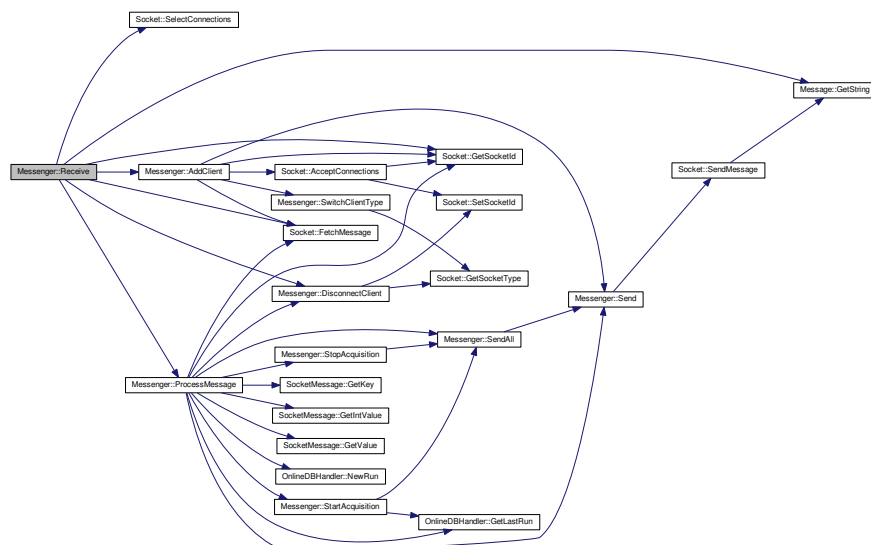
Here is the call graph for this function:



### 7.13.3.8 void Messenger::Receive ( )

Handle a message reception from a client.

Here is the call graph for this function:



#### 7.13.3.9 void Messenger::Send ( const Message & *m*, int *sid* ) const

Send any type of message to any client.

##### Parameters

in	<i>m</i>	<a href="#">Message</a> to transmit
in	<i>sid</i>	Unique identifier of the client on this socket

Here is the call graph for this function:



#### 7.13.3.10 void Messenger::SendAll ( const Socket::SocketType & *type*, const Message & *m* ) const [inline]

Send any type of message to all clients of one type.

##### Parameters

in	<i>type</i>	<a href="#">Client</a> type
in	<i>m</i>	<a href="#">Message</a> to transmit

Here is the call graph for this function:



#### 7.13.3.11 void Messenger::SendAll ( const Socket::SocketType & *type*, const Exception & *e* ) const [inline]

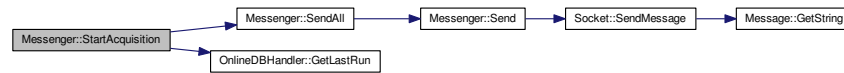
Here is the call graph for this function:



#### 7.13.3.12 void Messenger::StartAcquisition ( )

Start the data acquisition.

Here is the call graph for this function:



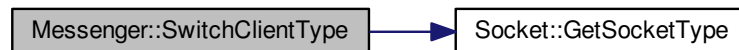
#### 7.13.3.13 void Messenger::StopAcquisition ( )

Here is the call graph for this function:



#### 7.13.3.14 void Messenger::SwitchClientType ( int sid, Socket::SocketType type ) [private]

Here is the call graph for this function:



### 7.13.4 Field Documentation

7.13.4.1 int Messenger::fNumAttempts [private]

7.13.4.2 pid\_t Messenger::fPID [private]

7.13.4.3 int Messenger::fStderrPipe[2] [private]

7.13.4.4 int Messenger::fStdoutPipe[2] [private]

The documentation for this class was generated from the following files:

- include/Messenger.h
- src/Messenger.cpp

## 7.14 OnlineDBHandler Class Reference

Handler for the run information online database.

```
#include <OnlineDBHandler.h>
```

## Data Structures

- struct [BurstInfo](#)
- struct [TDCConditions](#)

## Public Types

- typedef std::map< unsigned int, unsigned int > [RunCollection](#)
- typedef std::vector< [BurstInfo](#) > [BurstInfos](#)
- typedef std::vector< [TDCConditions](#) > [TDCConditionsCollection](#)

## Public Member Functions

- [OnlineDBHandler](#) (std::string path=std::string(std::getenv("PPS\_PATH"))+"/run\_infos.db")
- [~OnlineDBHandler](#) ()
- void [NewRun](#) ()
- void [NewBurst](#) ()
- [RunCollection](#) [GetRuns](#) () const
- unsigned int [GetLastRun](#) () const  
*Retrieve the last run acquired.*
- int [GetLastBurst](#) (unsigned int run) const
- [BurstInfos](#) [GetRunInfo](#) (unsigned int run) const  
*Retrieve information on a given run (spill IDs / timestamp)*
- void [SetTDCConditions](#) (unsigned short tdc\_id, unsigned long tdc\_address, unsigned short tdc\_acq\_mode, unsigned short tdc\_det\_mode, std::string detector)
- [TDCConditionsCollection](#) [GetTDCConditions](#) (unsigned int run\_id) const
- void [SetHVConditions](#) (unsigned short channel\_id, unsigned int vmax, unsigned imax)

## Private Member Functions

- void [BuildTables](#) ()
- template<class T >  
std::vector< std::vector< T > > [Select](#) (std::string req, int num\_fields=-1) const

## Private Attributes

- sqlite3 \* [fDB](#)

### 7.14.1 Detailed Description

Handler for the run information online database.

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

3 Aug 2015

## 7.14.2 Member Typedef Documentation

7.14.2.1 `typedef std::vector<BurstInfo> OnlineDBHandler::BurstInfos`

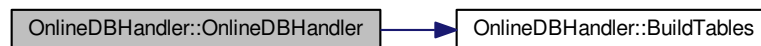
7.14.2.2 `typedef std::map<unsigned int, unsigned int> OnlineDBHandler::RunCollection`

7.14.2.3 `typedef std::vector<TDCConditions> OnlineDBHandler::TDCConditionsCollection`

## 7.14.3 Constructor & Destructor Documentation

7.14.3.1 `OnlineDBHandler::OnlineDBHandler ( std::string path = std::string(std::getenv("PPS_PATH")) + "/run_infos.db" ) [inline]`

Here is the call graph for this function:



7.14.3.2 `OnlineDBHandler::~~OnlineDBHandler ( ) [inline]`

## 7.14.4 Member Function Documentation

7.14.4.1 `void OnlineDBHandler::BuildTables ( ) [inline], [private]`

7.14.4.2 `int OnlineDBHandler::GetLastBurst ( unsigned int run ) const [inline]`

7.14.4.3 `unsigned int OnlineDBHandler::GetLastRun ( ) const [inline]`

Retrieve the last run acquired.

7.14.4.4 `BurstInfos OnlineDBHandler::GetRunInfo ( unsigned int run ) const [inline]`

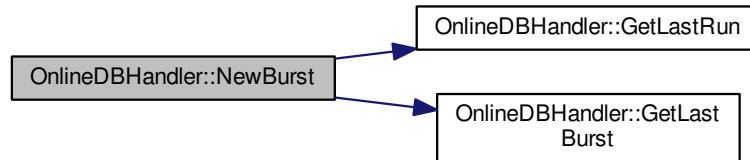
Retrieve information on a given run (spill IDs / timestamp)

7.14.4.5 `RunCollection OnlineDBHandler::GetRuns ( ) const [inline]`

7.14.4.6 `TDCConditionsCollection OnlineDBHandler::GetTDCConditions ( unsigned int run_id ) const [inline]`

7.14.4.7 `void OnlineDBHandler::NewBurst ( ) [inline]`

Here is the call graph for this function:



7.14.4.8 `void OnlineDBHandler::NewRun ( ) [inline]`

7.14.4.9 `template<class T > std::vector< std::vector<T> > OnlineDBHandler::Select ( std::string req, int num_fields = -1 ) const [inline],[private]`

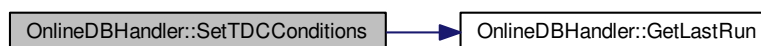
7.14.4.10 `void OnlineDBHandler::SetHVConditions ( unsigned short channel_id, unsigned int vmax, unsigned imax ) [inline]`

Here is the call graph for this function:



7.14.4.11 `void OnlineDBHandler::SetTDCCConditions ( unsigned short tdc_id, unsigned long tdc_address, unsigned short tdc_acq_mode, unsigned short tdc_det_mode, std::string detector ) [inline]`

Here is the call graph for this function:



## 7.14.5 Field Documentation

7.14.5.1 `sqlite3* OnlineDBHandler::fDB [private]`

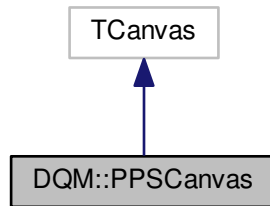
The documentation for this class was generated from the following file:

- `include/OnlineDBHandler.h`

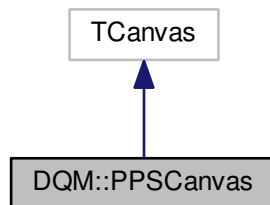
## 7.15 DQM::PPSCanvas Class Reference

```
#include <PPSCanvas.h>
```

Inheritance diagram for DQM::PPSCanvas:



Collaboration diagram for DQM::PPSCanvas:



### Public Member Functions

- `PPSCanvas ()`
- `PPSCanvas (TString name, unsigned int width=500, unsigned int height=500, TString upper_label="")`
- `PPSCanvas (TString name, TString upper_label)`
- `virtual ~PPSCanvas ()`
- `void SetRunInfo (unsigned int run_id, TString date)`
- `void SetUpperLabel (TString text)`
- `TPad * Grid ()`
- `void Save (TString ext="png", TString path=".")`

### Private Member Functions

- `void Build ()`
- `void DrawGrid ()`

## Private Attributes

- TPad \* `c1`
- TPad \* `c2`
- double `fWidth`
- double `fHeight`
- TLegend \* `fLegend`
- double `fLegendX`
- double `fLegendY`
- unsigned int `fLegendNumEntries`
- TPaveText \* `fLabel1`
- TPaveText \* `fLabel2`
- TPaveText \* `fLabel3`
- TString `fUpperLabelText`
- TPaveText \* `fUpperLabel`
- bool `fLabelsDrawn`
- unsigned int `fRunId`
- TString `fRunDate`

### 7.15.1 Detailed Description

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

3 Aug 2015

### 7.15.2 Constructor & Destructor Documentation

7.15.2.1 `DQM::PPSCanvas::PPSCanvas ( )` `[inline]`

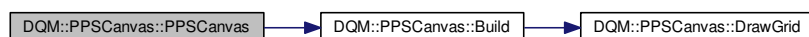
7.15.2.2 `DQM::PPSCanvas::PPSCanvas ( TString name, unsigned int width = 500, unsigned int height = 500, TString upper_label = " " )` `[inline]`

Here is the call graph for this function:



7.15.2.3 `DQM::PPSCanvas::PPSCanvas ( TString name, TString upper_label )` `[inline]`

Here is the call graph for this function:





7.15.2.4 `virtual DQM::PPSCanvas::~~PPSCanvas ( ) [inline],[virtual]`

### 7.15.3 Member Function Documentation

7.15.3.1 `void DQM::PPSCanvas::Build ( ) [inline],[private]`

Here is the call graph for this function:



7.15.3.2 `void DQM::PPSCanvas::DrawGrid ( ) [inline],[private]`

7.15.3.3 `TPad* DQM::PPSCanvas::Grid ( ) [inline]`

7.15.3.4 `void DQM::PPSCanvas::Save ( TString ext = "png", TString path = " . " ) [inline]`

Here is the call graph for this function:



7.15.3.5 `void DQM::PPSCanvas::SetRunInfo ( unsigned int run_id, TString date ) [inline]`

7.15.3.6 `void DQM::PPSCanvas::SetUpperLabel ( TString text ) [inline]`

### 7.15.4 Field Documentation

7.15.4.1 `TPad* DQM::PPSCanvas::c1 [private]`

7.15.4.2 `TPad * DQM::PPSCanvas::c2 [private]`

7.15.4.3 `double DQM::PPSCanvas::fHeight [private]`

7.15.4.4 `TPaveText* DQM::PPSCanvas::fLabel1 [private]`

7.15.4.5 `TPaveText * DQM::PPSCanvas::fLabel2 [private]`

7.15.4.6 `TPaveText * DQM::PPSCanvas::fLabel3 [private]`

- 7.15.4.7 `bool DQM::PPSCanvas::fLabelsDrawn` [private]
- 7.15.4.8 `TLegend* DQM::PPSCanvas::fLegend` [private]
- 7.15.4.9 `unsigned int DQM::PPSCanvas::fLegendNumEntries` [private]
- 7.15.4.10 `double DQM::PPSCanvas::fLegendX` [private]
- 7.15.4.11 `double DQM::PPSCanvas::fLegendY` [private]
- 7.15.4.12 `TString DQM::PPSCanvas::fRunDate` [private]
- 7.15.4.13 `unsigned int DQM::PPSCanvas::fRunId` [private]
- 7.15.4.14 `TPaveText* DQM::PPSCanvas::fUpperLabel` [private]
- 7.15.4.15 `TString DQM::PPSCanvas::fUpperLabelText` [private]
- 7.15.4.16 `double DQM::PPSCanvas::fWidth` [private]

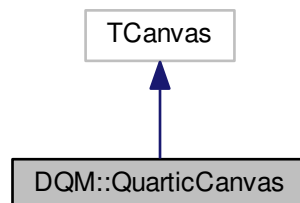
The documentation for this class was generated from the following file:

- `include/PPSCanvas.h`

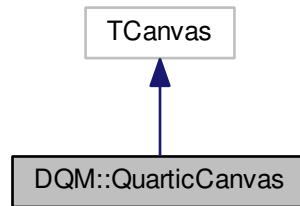
## 7.16 DQM::QuarticCanvas Class Reference

```
#include <QuarticCanvas.h>
```

Inheritance diagram for DQM::QuarticCanvas:



Collaboration diagram for DQM::QuarticCanvas:



## Data Structures

- struct [Coord](#)

## Public Member Functions

- [QuarticCanvas](#) ()
- [QuarticCanvas](#) (TString name, unsigned int width=500, unsigned int height=500, TString upper\_label="")
- [QuarticCanvas](#) (TString name, TString upper\_label)
- virtual [~QuarticCanvas](#) ()
- void [SetRunInfo](#) (unsigned int board\_id, unsigned int run\_id, unsigned int spill\_id, TString date)
- void [SetUpperLabel](#) (TString text)
- void [FillChannel](#) (unsigned short channel\_id, double content)
- TH2D \* [Grid](#) ()
- void [Save](#) (TString ext="png", TString path=".")

## Private Member Functions

- void [Build](#) ()
- void [DrawGrid](#) ()
- [Coord GetCoordinates](#) (unsigned short channel\_id) const

## Private Attributes

- TPad \* [c1](#)
- TPad \* [c2](#)
- TH2D \* [fHist](#)
- double [fWidth](#)
- double [fHeight](#)
- TLegend \* [fLegend](#)
- double [fLegendX](#)
- double [fLegendY](#)
- unsigned int [fLegendNumEntries](#)
- TPaveText \* [fLabel1](#)
- TPaveText \* [fLabel2](#)
- TPaveText \* [fLabel3](#)

- TPaveText \* [fLabel4](#)
- TString [fUpperLabelText](#)
- TPaveText \* [fUpperLabel](#)
- bool [fLabelsDrawn](#)
- unsigned int [fBoardId](#)
- unsigned int [fRunId](#)
- unsigned int [fSpillId](#)
- TString [fRunDate](#)

### 7.16.1 Detailed Description

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

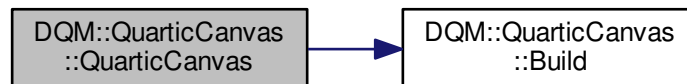
3 Aug 2015

### 7.16.2 Constructor & Destructor Documentation

7.16.2.1 `DQM::QuarticCanvas::QuarticCanvas ( )` `[inline]`

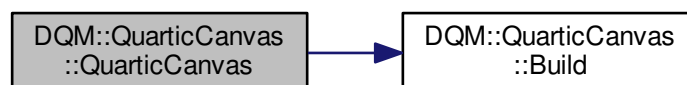
7.16.2.2 `DQM::QuarticCanvas::QuarticCanvas ( TString name, unsigned int width = 500, unsigned int height = 500, TString upper_label = " " )` `[inline]`

Here is the call graph for this function:



7.16.2.3 `DQM::QuarticCanvas::QuarticCanvas ( TString name, TString upper_label )` `[inline]`

Here is the call graph for this function:



7.16.2.4 `virtual DQM::QuarticCanvas::~~QuarticCanvas ( ) [inline], [virtual]`

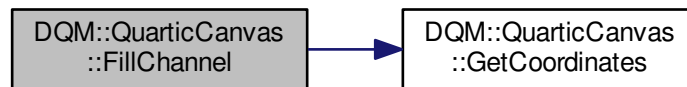
### 7.16.3 Member Function Documentation

7.16.3.1 `void DQM::QuarticCanvas::Build ( ) [inline], [private]`

7.16.3.2 `void DQM::QuarticCanvas::DrawGrid ( ) [inline], [private]`

7.16.3.3 `void DQM::QuarticCanvas::FillChannel ( unsigned short channel_id, double content ) [inline]`

Here is the call graph for this function:

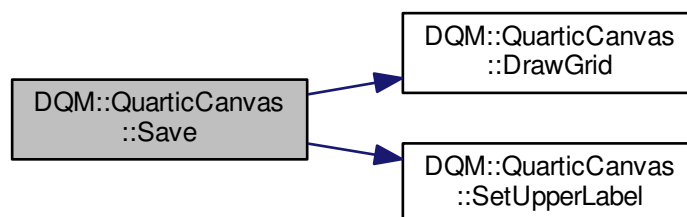


7.16.3.4 `Coord DQM::QuarticCanvas::GetCoordinates ( unsigned short channel_id ) const [inline], [private]`

7.16.3.5 `TH2D* DQM::QuarticCanvas::Grid ( ) [inline]`

7.16.3.6 `void DQM::QuarticCanvas::Save ( TString ext = "png", TString path = " . " ) [inline]`

Here is the call graph for this function:



7.16.3.7 `void DQM::QuarticCanvas::SetRunInfo ( unsigned int board_id, unsigned int run_id, unsigned int spill_id, TString date ) [inline]`

7.16.3.8 `void DQM::QuarticCanvas::SetUpperLabel ( TString text ) [inline]`

### 7.16.4 Field Documentation

```

7.16.4.1  TPad* DQM::QuarticCanvas::c1  [private]

7.16.4.2  TPad * DQM::QuarticCanvas::c2  [private]

7.16.4.3  unsigned int DQM::QuarticCanvas::fBoardId  [private]

7.16.4.4  double DQM::QuarticCanvas::fHeight  [private]

7.16.4.5  TH2D* DQM::QuarticCanvas::fHist  [private]

7.16.4.6  TPaveText* DQM::QuarticCanvas::fLabel1  [private]

7.16.4.7  TPaveText * DQM::QuarticCanvas::fLabel2  [private]

7.16.4.8  TPaveText * DQM::QuarticCanvas::fLabel3  [private]

7.16.4.9  TPaveText * DQM::QuarticCanvas::fLabel4  [private]

7.16.4.10  bool DQM::QuarticCanvas::fLabelsDrawn  [private]

7.16.4.11  TLegend* DQM::QuarticCanvas::fLegend  [private]

7.16.4.12  unsigned int DQM::QuarticCanvas::fLegendNumEntries  [private]

7.16.4.13  double DQM::QuarticCanvas::fLegendX  [private]

7.16.4.14  double DQM::QuarticCanvas::fLegendY  [private]

7.16.4.15  TString DQM::QuarticCanvas::fRunDate  [private]

7.16.4.16  unsigned int DQM::QuarticCanvas::fRunId  [private]

7.16.4.17  unsigned int DQM::QuarticCanvas::fSpillId  [private]

7.16.4.18  TPaveText* DQM::QuarticCanvas::fUpperLabel  [private]

7.16.4.19  TString DQM::QuarticCanvas::fUpperLabelText  [private]

7.16.4.20  double DQM::QuarticCanvas::fWidth  [private]

```

The documentation for this class was generated from the following file:

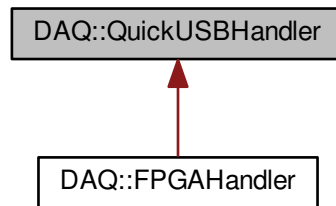
- include/QuarticCanvas.h

## 7.17 DAQ::QuickUSBHandler Class Reference

Generic QuickUSB communication handler.

```
#include <QuickUSBHandler.h>
```

Inheritance diagram for DAQ::QuickUSBHandler:



## Data Structures

- struct [Version](#)

## Public Member Functions

- [QuickUSBHandler](#) ()
- virtual [~QuickUSBHandler](#) ()
- void [Init](#) ()
- void [Reset](#) () const
- [Version](#) [GetFWVersion](#) () const
- [Version](#) [GetDriverVersion](#) () const
- [Version](#) [GetDLLVersion](#) () const
- void [Write](#) (uint16\_t addr, uint8\_t word) const  
*Write a single word to the QuickUSB device.*
- void [Write](#) (uint16\_t addr, std::vector< uint8\_t > &words, uint16\_t size) const  
*Write a set of words to the QuickUSB device.*
- std::vector< uint8\_t > [Fetch](#) (uint16\_t addr, uint16\_t size) const  
*Receive a set of words from the QuickUSB device.*
- void [StartBulkTransfer](#) (QVOIDRETURN callback(PQBULKSTREAM))
- void [StopBulkTransfer](#) ()

## Protected Attributes

- bool [flsStopping](#)

## Private Attributes

- std::string [fDevice](#)
- QHANDLE [fHandle](#)
- uint8\_t [fStreamId](#)

### 7.17.1 Detailed Description

Generic QuickUSB communication handler.

Date

17 May 2016

Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

### 7.17.2 Constructor & Destructor Documentation

7.17.2.1 `DAQ::QuickUSBHandler::QuickUSBHandler ( )`

7.17.2.2 `DAQ::QuickUSBHandler::~~QuickUSBHandler ( )` [virtual]

### 7.17.3 Member Function Documentation

7.17.3.1 `std::vector< uint8_t > DAQ::QuickUSBHandler::Fetch ( uint16_t addr, uint16_t size ) const`

Receive a set of words from the QuickUSB device.

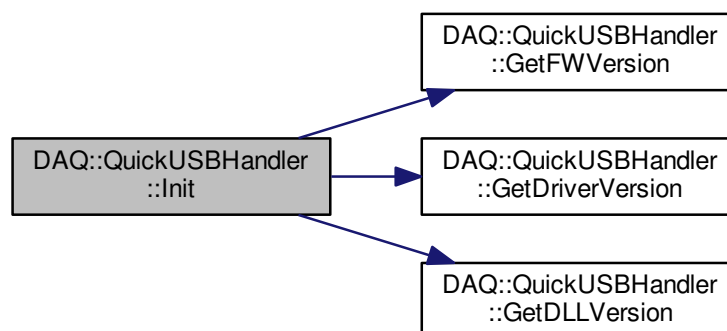
7.17.3.2 `QuickUSBHandler::Version DAQ::QuickUSBHandler::GetDLLVersion ( ) const`

7.17.3.3 `QuickUSBHandler::Version DAQ::QuickUSBHandler::GetDriverVersion ( ) const`

7.17.3.4 `QuickUSBHandler::Version DAQ::QuickUSBHandler::GetFWVersion ( ) const`

7.17.3.5 `void DAQ::QuickUSBHandler::Init ( )`

Here is the call graph for this function:





7.17.3.6 `void DAQ::QuickUSBHandler::Reset ( ) const`

7.17.3.7 `void DAQ::QuickUSBHandler::StartBulkTransfer ( QVOIDRETURN callbackPQBULKSTREAM )`

7.17.3.8 `void DAQ::QuickUSBHandler::StopBulkTransfer ( )`

7.17.3.9 `void DAQ::QuickUSBHandler::Write ( uint16_t addr, uint8_t word ) const [inline]`

Write a single word to the QuickUSB device.

7.17.3.10 `void DAQ::QuickUSBHandler::Write ( uint16_t addr, std::vector< uint8_t > & words, uint16_t size ) const`

Write a set of words to the QuickUSB device.

## 7.17.4 Field Documentation

7.17.4.1 `std::string DAQ::QuickUSBHandler::fDevice [private]`

7.17.4.2 `QHANDLE DAQ::QuickUSBHandler::fHandle [private]`

7.17.4.3 `bool DAQ::QuickUSBHandler::fIsStopping [protected]`

7.17.4.4 `uint8_t DAQ::QuickUSBHandler::fStreamId [private]`

The documentation for this class was generated from the following files:

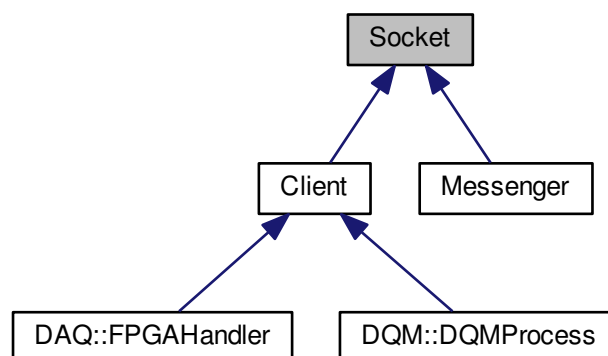
- `daq/include/QuickUSBHandler.h`
- `daq/src/QuickUSBHandler.cpp`

## 7.18 Socket Class Reference

Base socket object from which clients/master from a socket inherit.

```
#include <Socket.h>
```

Inheritance diagram for Socket:



## Public Types

- enum `SocketType` {  
`INVALID` = -1, `MASTER` = 0, `WEBSOCKET_CLIENT`, `CLIENT`,  
`DETECTOR`, `DQM`, `DAQ` }
- *Type of actor playing a role on the socket.*
- typedef `std::set< std::pair< int, SocketType > >` `SocketCollection`

## Public Member Functions

- `Socket` ()
- `Socket` (int port)
- virtual `~Socket` ()
- void `Stop` ()
- *Terminates the socket and all attached communications.*
- void `SetPort` (int port)
- int `GetPort` () const
- *Retrieve the port used for this socket.*
- void `AcceptConnections` (`Socket` &socket)
- *Accept connection from a client.*
- void `SelectConnections` ()
- void `SetSocketId` (int sid)
- int `GetSocketId` () const
- `SocketType` `GetSocketType` (int sid) const
- bool `IsWebSocket` (int sid) const
- void `DumpConnected` () const

## Protected Member Functions

- bool `Start` ()
- *Start the socket.*
- void `Bind` ()
- *Bind a name to a socket.*
- void `PrepareConnection` ()
- void `Listen` (int maxconn)
- *Listen to incoming messages.*
- void `SendMessage` (`Message` message, int id=-1) const
- *Send a message on a socket.*
- `Message` `FetchMessage` (int id=-1) const
- *Receive a message from a socket.*

## Protected Attributes

- int `fPort`
- char `fBuffer` [MAX\_WORD\_LENGTH]
- `SocketCollection` `fSocketsConnected`
- fd\_set `fMaster`
- *Master file descriptor list.*
- fd\_set `fReadFds`
- *Temp file descriptor list for select()*

## Private Member Functions

- void [Create](#) ()  
*Create an endpoint for communication.*
- void [Configure](#) ()  
*Configure the socket object for communication.*

## Private Attributes

- int [fSocketId](#)
- struct sockaddr\_in [fAddress](#)

### 7.18.1 Detailed Description

Base socket object from which clients/master from a socket inherit.

General object providing all useful method to connect/bind/send/receive information through system sockets.

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

23 Mar 2015

### 7.18.2 Member Typedef Documentation

7.18.2.1 `typedef std::set< std::pair<int,SocketType> > Socket::SocketCollection`

### 7.18.3 Member Enumeration Documentation

7.18.3.1 `enum Socket::SocketType`

Type of actor playing a role on the socket.

#### Enumerator

**INVALID**  
**MASTER**  
**WEBSOCKET\_CLIENT**  
**CLIENT**  
**DETECTOR**  
**DQM**  
**DAQ**

### 7.18.4 Constructor & Destructor Documentation

7.18.4.1 `Socket::Socket ( ) [inline]`

7.18.4.2 `Socket::Socket ( int port )`

7.18.4.3 `Socket::~Socket ( ) [virtual]`

## 7.18.5 Member Function Documentation

### 7.18.5.1 void Socket::AcceptConnections ( Socket & socket )

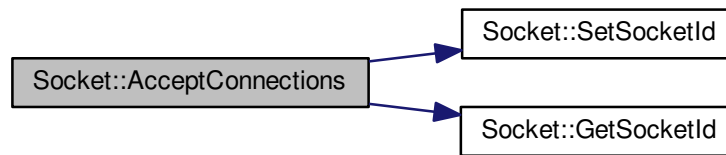
Accept connection from a client.

Set the socket to accept connections any client transmitting through the socket

#### Parameters

in, out	socket	Master/client object to enable on the socket
---------	--------	--

Here is the call graph for this function:



### 7.18.5.2 void Socket::Bind ( ) [protected]

Bind a name to a socket.

#### Returns

Success of the operation

Here is the call graph for this function:



### 7.18.5.3 void Socket::Configure ( ) [private]

Configure the socket object for communication.

### 7.18.5.4 void Socket::Create ( ) [private]

Create an endpoint for communication.

7.18.5.5 `void Socket::DumpConnected ( ) const`

7.18.5.6 `Message Socket::FetchMessage ( int id = -1 ) const` [protected]

Receive a message from a socket.

Returns

Received message as a `std::string`

7.18.5.7 `int Socket::GetPort ( ) const` [inline]

Retrieve the port used for this socket.

7.18.5.8 `int Socket::GetSocketId ( ) const` [inline]

7.18.5.9 `SocketType Socket::GetSocketType ( int sid ) const` [inline]

7.18.5.10 `bool Socket::IsWebSocket ( int sid ) const` [inline]

Here is the call graph for this function:



7.18.5.11 `void Socket::Listen ( int maxconn )` [protected]

Listen to incoming messages.

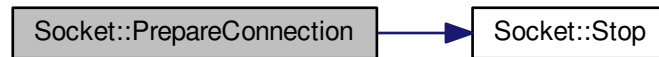
Set the socket to listen to any message coming from outside

Here is the call graph for this function:



#### 7.18.5.12 void Socket::PrepareConnection ( ) [protected]

Here is the call graph for this function:



#### 7.18.5.13 void Socket::SelectConnections ( )

Register all open file descriptors to read their communication through the socket

#### 7.18.5.14 void Socket::SendMessage ( Message *message*, int *id* = -1 ) const [protected]

Send a message on a socket.

Here is the call graph for this function:



#### 7.18.5.15 void Socket::SetPort ( int *port* ) [inline]

#### 7.18.5.16 void Socket::SetSocketId ( int *sid* ) [inline]

#### 7.18.5.17 bool Socket::Start ( ) [protected]

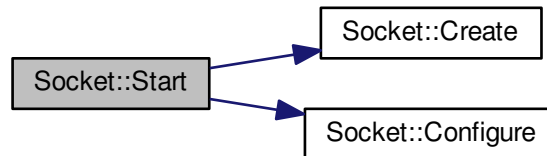
Start the socket.

Launch all mandatory operations to set the socket to be used

**Returns**

Success of the operation

Here is the call graph for this function:

**7.18.5.18 void Socket::Stop ( )**

Terminates the socket and all attached communications.

**7.18.6 Field Documentation****7.18.6.1 struct sockaddr\_in Socket::fAddress [private]****7.18.6.2 char Socket::fBuffer[MAX\_WORD\_LENGTH] [protected]****7.18.6.3 fd\_set Socket::fMaster [protected]**

Master file descriptor list.

**7.18.6.4 int Socket::fPort [protected]****7.18.6.5 fd\_set Socket::fReadFds [protected]**

Temp file descriptor list for select()

**7.18.6.6 int Socket::fSocketId [private]**

A file descriptor for this socket, if *Create* was performed beforehand.

**7.18.6.7 SocketCollection Socket::fSocketsConnected [protected]**

The documentation for this class was generated from the following files:

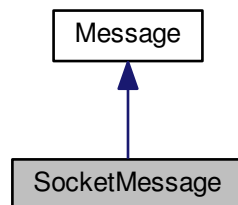
- include/Socket.h
- src/Socket.cpp

## 7.19 SocketMessage Class Reference

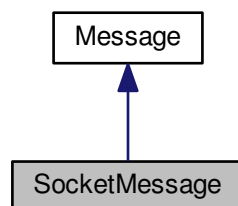
Socket-passed message type.

```
#include <SocketMessage.h>
```

Inheritance diagram for SocketMessage:



Collaboration diagram for SocketMessage:



### Public Member Functions

- [SocketMessage](#) ()
- [SocketMessage](#) (const [Message](#) &msg)
- [SocketMessage](#) (const char \*msg\_s)
- [SocketMessage](#) (std::string msg\_s)
- [SocketMessage](#) (const MessageKey &key)
 

*Construct a socket message out of a key.*
- [SocketMessage](#) (const MessageKey &key, const char \*value)
 

*Construct a socket message out of a key and a string-type value.*
- [SocketMessage](#) (const MessageKey &key, std::string value)
 

*Construct a socket message out of a key and a string-type value.*
- [SocketMessage](#) (const MessageKey &key, const short value)
 

*Construct a socket message out of a key and a short integer-type value.*
- [SocketMessage](#) (const MessageKey &key, const int value)
 

*Construct a socket message out of a key and an integer-type value.*



- [SocketMessage](#) (const MessageKey &key, const long value)  
*Construct a socket message out of a key and a long integer-type value.*
- [SocketMessage](#) (const MessageKey &key, const float value)  
*Construct a socket message out of a key and a float-type value.*
- [SocketMessage](#) (const MessageKey &key, const double value)  
*Construct a socket message out of a key and a double precision-type value.*
- [SocketMessage](#) (MessageMap msg\_m)  
*Construct a socket message out of a map of key/string-type value.*
- [~SocketMessage](#) ()
- void [SetKeyValue](#) (const MessageKey &key, const char \*value)  
*String-valued message.*
- void [SetKeyValue](#) (const MessageKey &key, short int\_value)  
*Send a short integer-valued message.*
- void [SetKeyValue](#) (const MessageKey &key, int int\_value)  
*Send an integer-valued message.*
- void [SetKeyValue](#) (const MessageKey &key, long int\_value)  
*Send a long integer-valued message.*
- void [SetKeyValue](#) (const MessageKey &key, float float\_value)  
*Float-valued message.*
- void [SetKeyValue](#) (const MessageKey &key, double double\_value)  
*Double-valued message.*
- std::string [GetString](#) () const  
*Extract the whole key:value message.*
- MessageKey [GetKey](#) () const  
*Extract the message's key.*
- std::string [GetValue](#) () const  
*Extract the message's string value.*
- std::string [GetCleanedValue](#) () const  
*Extract the message's string value (without the trailing endlines)*
- int [GetIntValue](#) () const  
*Extract the message's integer value.*
- VectorValue [GetVectorValue](#) () const  
*Extract the message's vector of string value.*
- void [Dump](#) (std::ostream &os=std::cout) const

### Private Member Functions

- MessageMap [Object](#) () const
- std::string [String](#) () const

### Private Attributes

- MessageMap [fMessage](#)

## Additional Inherited Members

### 7.19.1 Detailed Description

Socket-passed message type.

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

26 Mar 2015

### 7.19.2 Constructor & Destructor Documentation

7.19.2.1 `SocketMessage::SocketMessage ( )` `[inline]`

7.19.2.2 `SocketMessage::SocketMessage ( const Message & msg )` `[inline]`

Here is the call graph for this function:



7.19.2.3 `SocketMessage::SocketMessage ( const char * msg_s )` `[inline]`

Here is the call graph for this function:



#### 7.19.2.4 SocketMessage::SocketMessage ( std::string msg\_s ) [inline]

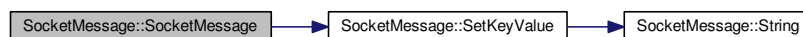
Here is the call graph for this function:



#### 7.19.2.5 SocketMessage::SocketMessage ( const MessageKey & key ) [inline]

Construct a socket message out of a key.

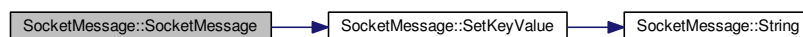
Here is the call graph for this function:



#### 7.19.2.6 SocketMessage::SocketMessage ( const MessageKey & key, const char \* value ) [inline]

Construct a socket message out of a key and a string-type value.

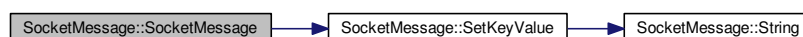
Here is the call graph for this function:



#### 7.19.2.7 SocketMessage::SocketMessage ( const MessageKey & key, std::string value ) [inline]

Construct a socket message out of a key and a string-type value.

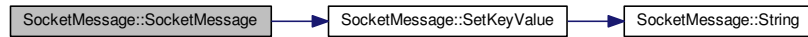
Here is the call graph for this function:



#### 7.19.2.8 `SocketMessage::SocketMessage ( const MessageKey & key, const short value ) [inline]`

Construct a socket message out of a key and a short integer-type value.

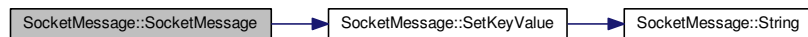
Here is the call graph for this function:



#### 7.19.2.9 `SocketMessage::SocketMessage ( const MessageKey & key, const int value ) [inline]`

Construct a socket message out of a key and an integer-type value.

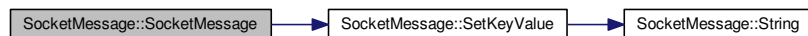
Here is the call graph for this function:



#### 7.19.2.10 `SocketMessage::SocketMessage ( const MessageKey & key, const long value ) [inline]`

Construct a socket message out of a key and a long integer-type value.

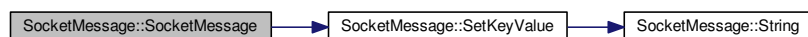
Here is the call graph for this function:



#### 7.19.2.11 `SocketMessage::SocketMessage ( const MessageKey & key, const float value ) [inline]`

Construct a socket message out of a key and a float-type value.

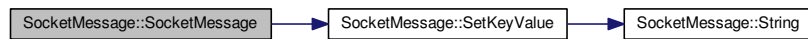
Here is the call graph for this function:



#### 7.19.2.12 SocketMessage::SocketMessage ( const MessageKey & *key*, const double *value* ) [inline]

Construct a socket message out of a key and a double precision-type value.

Here is the call graph for this function:



#### 7.19.2.13 SocketMessage::SocketMessage ( MessageMap *msg\_m* ) [inline]

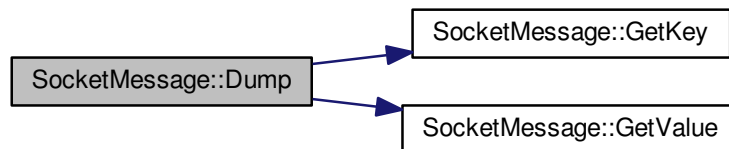
Construct a socket message out of a map of key/string-type value.

#### 7.19.2.14 SocketMessage::~SocketMessage ( ) [inline]

### 7.19.3 Member Function Documentation

#### 7.19.3.1 void SocketMessage::Dump ( std::ostream & *os* = std::cout ) const [inline]

Here is the call graph for this function:



#### 7.19.3.2 std::string SocketMessage::GetCleanedValue ( ) const [inline]

Extract the message's string value (without the trailing endlines)

#### 7.19.3.3 int SocketMessage::GetIntValue ( ) const [inline]

Extract the message's integer value.

#### 7.19.3.4 MessageKey SocketMessage::GetKey ( ) const [inline]

Extract the message's key.

#### 7.19.3.5 std::string SocketMessage::GetString ( ) const [inline]

Extract the whole key:value message.

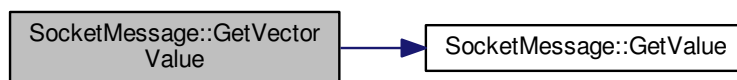
#### 7.19.3.6 `std::string SocketMessage::GetValue ( ) const` `[inline]`

Extract the message's string value.

#### 7.19.3.7 `VectorValue SocketMessage::GetVectorValue ( ) const` `[inline]`

Extract the message's vector of string value.

Here is the call graph for this function:



#### 7.19.3.8 `MessageMap SocketMessage::Object ( ) const` `[inline]`, `[private]`

#### 7.19.3.9 `void SocketMessage::SetKeyValue ( const MessageKey & key, const char * value )` `[inline]`

String-valued message.

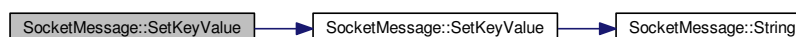
Here is the call graph for this function:



#### 7.19.3.10 `void SocketMessage::SetKeyValue ( const MessageKey & key, short int_value )` `[inline]`

Send a short integer-valued message.

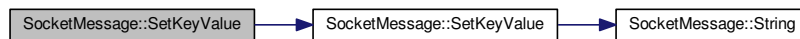
Here is the call graph for this function:



#### 7.19.3.11 `void SocketMessage::SetKeyValue ( const MessageKey & key, int int_value )` `[inline]`

Send an integer-valued message.

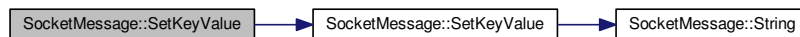
Here is the call graph for this function:



#### 7.19.3.12 void SocketMessage::SetKeyValue ( const MessageKey & key, long int\_value ) [inline]

Send a long integer-valued message.

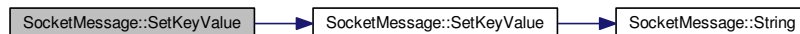
Here is the call graph for this function:



#### 7.19.3.13 void SocketMessage::SetKeyValue ( const MessageKey & key, float float\_value ) [inline]

Float-valued message.

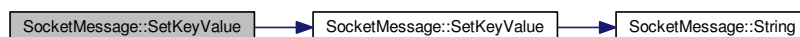
Here is the call graph for this function:



#### 7.19.3.14 void SocketMessage::SetKeyValue ( const MessageKey & key, double double\_value ) [inline]

Double-valued message.

Here is the call graph for this function:



#### 7.19.3.15 std::string SocketMessage::String ( ) const [inline], [private]

### 7.19.4 Field Documentation

#### 7.19.4.1 MessageMap SocketMessage::fMessage [private]

The documentation for this class was generated from the following file:

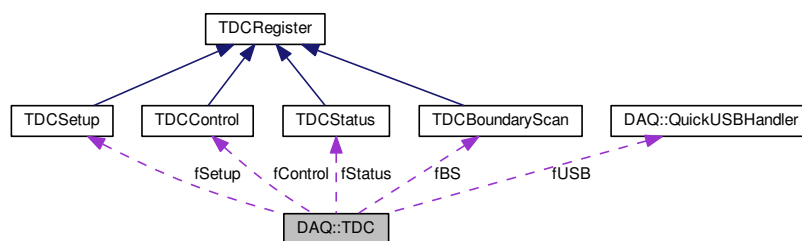
- include/SocketMessage.h

## 7.20 DAQ::TDC Class Reference

HPTDC object.

```
#include <TDC.h>
```

Collaboration diagram for DAQ::TDC:



### Public Types

- enum [AcquisitionMode](#) { [CONT\\_STORAGE](#), [TRIG\\_MATCH](#) }  
*TDC acquisition mode.*
- enum [DetectionMode](#) { [PAIR](#) = 0x0, [OTRILING](#) = 0x1, [OLEADING](#) = 0x2, [TRAILEAD](#) = 0x3 }

### Public Member Functions

- [TDC](#) (unsigned int id, [QuickUSBHandler](#) \*h)
- [~TDC](#) ()
- void [SetSetupRegister](#) (const [TDCSetup](#) &c)  
*Submit the HPTDC setup word as a [TDCSetup](#) object.*
- [TDCSetup](#) [GetSetupRegister](#) ()  
*Retrieve the HPTDC setup word as a [TDCSetup](#) object.*
- bool [CheckFirmwareVersion](#) () const
- void [SoftReset](#) ()
- [TDCEventCollection](#) [FetchEvents](#) ()
- void [ReadStatus](#) ()

### Private Member Functions

- void [SendConfiguration](#) ()  
*Set the setup word to the HPTDC internal setup register.*
- void [ReadConfiguration](#) ()  
*Read the setup word from the HPTDC internal setup register.*



- template<class T >  
void [WriteRegister](#) (unsigned int r, const T &v)  
*Write one register content on the HPTDC inner memory.*
- template<class T >  
T [ReadRegister](#) (unsigned int r)  
*Retrieve one register content from the HPTDC inner memory.*

### Private Attributes

- unsigned int [fld](#)
- [QuickUSBHandler](#) \* [fUSB](#)
- [TDCSetup](#) [fSetup](#)
- [TDCControl](#) [fControl](#)
- [TDCBoundaryScan](#) [fBS](#)
- [TDCStatus](#) [fStatus](#)

### 7.20.1 Detailed Description

HPTDC object.

Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

Date

27 Apr 2015

### 7.20.2 Member Enumeration Documentation

#### 7.20.2.1 enum DAQ::TDC::DetectionMode

Enumerator

***PAIR***  
***OTRILING***  
***OLEADING***  
***TRAILEAD***

### 7.20.3 Constructor & Destructor Documentation

#### 7.20.3.1 DAQ::TDC::TDC ( unsigned int *id*, [QuickUSBHandler](#) \* *h* )

Here is the call graph for this function:



7.20.3.2 `DAQ::TDC::~~TDC ( ) [inline]`

## 7.20.4 Member Function Documentation

7.20.4.1 `bool DAQ::TDC::CheckFirmwareVersion ( ) const`

7.20.4.2 `TDCEventCollection DAQ::TDC::FetchEvents ( )`

7.20.4.3 `TDCSetup DAQ::TDC::GetSetupRegister ( ) [inline]`

Retrieve the HPTDC setup word as a [TDCSetup](#) object.

7.20.4.4 `void DAQ::TDC::ReadConfiguration ( ) [private]`

Read the setup word from the HPTDC internal setup register.

7.20.4.5 `template<class T > T DAQ::TDC::ReadRegister ( unsigned int r ) [private]`

Retrieve one register content from the HPTDC inner memory.

7.20.4.6 `void DAQ::TDC::ReadStatus ( ) [inline]`

7.20.4.7 `void DAQ::TDC::SendConfiguration ( ) [private]`

Set the setup word to the HPTDC internal setup register.

7.20.4.8 `void DAQ::TDC::SetSetupRegister ( const TDCSetup & c ) [inline]`

Submit the HPTDC setup word as a [TDCSetup](#) object.

7.20.4.9 `void DAQ::TDC::SoftReset ( )`

7.20.4.10 `template<class T > void DAQ::TDC::WriteRegister ( unsigned int r, const T & v ) [private]`

Write one register content on the HPTDC inner memory.

## 7.20.5 Field Documentation

7.20.5.1 `TDCBoundaryScan DAQ::TDC::fBS [private]`

7.20.5.2 `TDCControl DAQ::TDC::fControl [private]`

7.20.5.3 `unsigned int DAQ::TDC::fld [private]`

7.20.5.4 `TDCSetup DAQ::TDC::fSetup [private]`

7.20.5.5 `TDCStatus DAQ::TDC::fStatus [private]`

7.20.5.6 `QuickUSBHandler* DAQ::TDC::fUSB [private]`

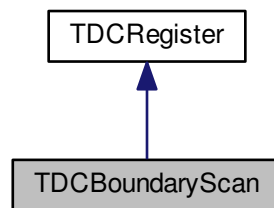
The documentation for this class was generated from the following files:

- `daq/include/TDC.h`
- `daq/src/TDC.cpp`

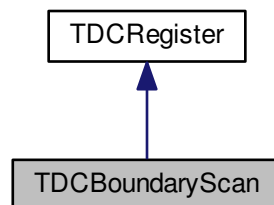
## 7.21 TDCBoundaryScan Class Reference

```
#include <TDCBoundaryScan.h>
```

Inheritance diagram for TDCBoundaryScan:



Collaboration diagram for TDCBoundaryScan:



### Public Member Functions

- [TDCBoundaryScan](#) ()
- [TDCBoundaryScan](#) (const [TDCBoundaryScan](#) &bs)
- void [SetConstantValues](#) ()

### Static Private Attributes

- static const [bit](#) [kTokenOut](#) = 0
- static const [bit](#) [kStrobeOut](#) = 1
- static const [bit](#) [kSerialOut](#) = 2
- static const [bit](#) [kTest](#) = 3
- static const [bit](#) [kError](#) = 4

- static const [bit kDataReady](#) = 5
- static const [bit kParallelEnable](#) = 6
- static const [bit kParallelDataOut](#) = 7
- static const [bit kEncodedControl](#) = 39
- static const [bit kTrigger](#) = 40
- static const [bit kEventReset](#) = 41
- static const [bit kBunchReset](#) = 42
- static const [bit kGetData](#) = 43
- static const [bit kSerialBypassIn](#) = 44
- static const [bit kSerialIn](#) = 45
- static const [bit kTokenBypassIn](#) = 46
- static const [bit kTokenIn](#) = 47
- static const [bit kReset](#) = 48
- static const [bit kAuxClock](#) = 49
- static const [bit kClk](#) = 50
- static const [bit kHit](#) = 51

## Additional Inherited Members

### 7.21.1 Detailed Description

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

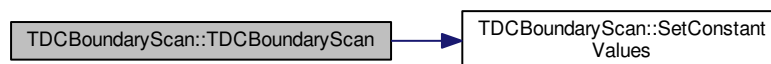
#### Date

24 Apr 2015

### 7.21.2 Constructor & Destructor Documentation

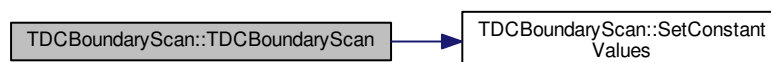
#### 7.21.2.1 TDCBoundaryScan::TDCBoundaryScan ( ) [inline]

Here is the call graph for this function:



#### 7.21.2.2 TDCBoundaryScan::TDCBoundaryScan ( const TDCBoundaryScan & bs ) [inline]

Here is the call graph for this function:



### 7.21.3 Member Function Documentation

7.21.3.1 `void TDCBoundaryScan::SetConstantValues ( )` `[inline]`, `[virtual]`

Ensure that the critical constant values are properly set in the register word

Implements [TDCRegister](#).

### 7.21.4 Field Documentation

7.21.4.1 `const bit TDCBoundaryScan::kAuxClock = 49` `[static]`, `[private]`

7.21.4.2 `const bit TDCBoundaryScan::kBunchReset = 42` `[static]`, `[private]`

7.21.4.3 `const bit TDCBoundaryScan::kClk = 50` `[static]`, `[private]`

7.21.4.4 `const bit TDCBoundaryScan::kDataReady = 5` `[static]`, `[private]`

7.21.4.5 `const bit TDCBoundaryScan::kEncodedControl = 39` `[static]`, `[private]`

7.21.4.6 `const bit TDCBoundaryScan::kError = 4` `[static]`, `[private]`

7.21.4.7 `const bit TDCBoundaryScan::kEventReset = 41` `[static]`, `[private]`

7.21.4.8 `const bit TDCBoundaryScan::kGetData = 43` `[static]`, `[private]`

7.21.4.9 `const bit TDCBoundaryScan::kHit = 51` `[static]`, `[private]`

7.21.4.10 `const bit TDCBoundaryScan::kParallelDataOut = 7` `[static]`, `[private]`

7.21.4.11 `const bit TDCBoundaryScan::kParallelEnable = 6` `[static]`, `[private]`

7.21.4.12 `const bit TDCBoundaryScan::kReset = 48` `[static]`, `[private]`

7.21.4.13 `const bit TDCBoundaryScan::kSerialBypassIn = 44` `[static]`, `[private]`

7.21.4.14 `const bit TDCBoundaryScan::kSerialIn = 45` `[static]`, `[private]`

7.21.4.15 `const bit TDCBoundaryScan::kSerialOut = 2` `[static]`, `[private]`

7.21.4.16 `const bit TDCBoundaryScan::kStrobeOut = 1` `[static]`, `[private]`

7.21.4.17 `const bit TDCBoundaryScan::kTest = 3` `[static]`, `[private]`

7.21.4.18 `const bit TDCBoundaryScan::kTokenBypassIn = 46` `[static]`, `[private]`

7.21.4.19 `const bit TDCBoundaryScan::kTokenIn = 47` `[static]`, `[private]`

7.21.4.20 `const bit TDCBoundaryScan::kTokenOut = 0` `[static]`, `[private]`

7.21.4.21 `const bit TDCBoundaryScan::kTrigger = 40` `[static]`, `[private]`

The documentation for this class was generated from the following file:

- `daq/include/TDCBoundaryScan.h`

## 7.22 OnlineDBHandler::TDCCConditions Struct Reference

```
#include <OnlineDBHandler.h>
```

### Public Member Functions

- bool `operator==` (const `TDCCConditions` &`rhs`) const
- `TDCCConditions` & `operator=` (const `TDCCConditions` &`rhs`)

### Data Fields

- unsigned int `run_id`
- unsigned short `tdc_id`
- unsigned long `tdc_address`
- unsigned short `tdc_acq_mode`
- unsigned short `tdc_det_mode`
- std::string `detector`

### 7.22.1 Member Function Documentation

7.22.1.1 `TDCCConditions& OnlineDBHandler::TDCCConditions::operator= ( const TDCCConditions & rhs )` [inline]

7.22.1.2 `bool OnlineDBHandler::TDCCConditions::operator== ( const TDCCConditions & rhs )` const [inline]

### 7.22.2 Field Documentation

7.22.2.1 `std::string OnlineDBHandler::TDCCConditions::detector`

7.22.2.2 `unsigned int OnlineDBHandler::TDCCConditions::run_id`

7.22.2.3 `unsigned short OnlineDBHandler::TDCCConditions::tdc_acq_mode`

7.22.2.4 `unsigned long OnlineDBHandler::TDCCConditions::tdc_address`

7.22.2.5 `unsigned short OnlineDBHandler::TDCCConditions::tdc_det_mode`

7.22.2.6 `unsigned short OnlineDBHandler::TDCCConditions::tdc_id`

The documentation for this struct was generated from the following file:

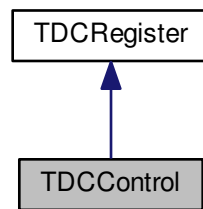
- `include/OnlineDBHandler.h`

## 7.23 TDCCControl Class Reference

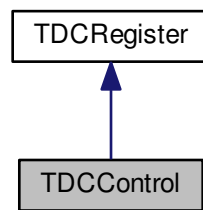
Control word to be sent to the HPTDC chip.

```
#include <TDCCControl.h>
```

Inheritance diagram for TDCControl:



Collaboration diagram for TDCControl:



## Public Types

- enum [EnablePattern](#)
- enum [RegisterName](#) { [R\\_EnablePattern](#), [R\\_GlobalReset](#), [R\\_DLLReset](#), [R\\_PLLReset](#) }

## Public Member Functions

- [TDCControl](#) ()
- [TDCControl](#) (const [TDCControl](#) &c)
- [TDCControl](#) (const std::vector< uint8\_t > &words)
- void [SetEnablePattern](#) (const [EnablePattern](#) &ep)
- [EnablePattern](#) [GetEnablePattern](#) () const
- void [SetGlobalReset](#) (const bool gr=true)
- bool [GetGlobalReset](#) () const
- void [SetDLLReset](#) (const bool dr=true)
- bool [GetDLLReset](#) () const
- void [SetPLLReset](#) (const bool pr=true)
- bool [GetPLLReset](#) () const
- void [EnableChannel](#) (unsigned int id)
- void [EnableAllChannels](#) ()
- void [DisableChannel](#) (unsigned int id)

- void [DisableAllChannels](#) ()
- void [Dump](#) (int verb=1, std::ostream &os=std::cout) const
- void [SetConstantValues](#) ()

### Private Member Functions

- void [SetControlParity](#) (const bool cp=true)

### Static Private Attributes

- static const [bit kEnablePattern](#) = 0
- static const [bit kGlobalReset](#) = 4
- static const [bit kEnableChannel](#) = 5
- static const [bit kDLLReset](#) = 37
- static const [bit kPLLReset](#) = 38
- static const [bit kControlParity](#) = 39

### Additional Inherited Members

#### 7.23.1 Detailed Description

Control word to be sent to the HPTDC chip.

Object handling the control word provided by/to the HPTDC chip

Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

Date

24 Apr 2015

#### 7.23.2 Member Enumeration Documentation

7.23.2.1 enum [TDCCControl::EnablePattern](#)

7.23.2.2 enum [TDCCControl::RegisterName](#)

Enumerator

***R\_EnablePattern***

***R\_GlobalReset***

***R\_DLLReset***

***R\_PLLReset***

#### 7.23.3 Constructor & Destructor Documentation

7.23.3.1 [TDCCControl::TDCCControl](#) ( ) [[inline](#)]

Here is the call graph for this function:





### 7.23.3.2 TDCCControl::TDCCControl ( const TDCCControl & c ) [inline]

Here is the call graph for this function:

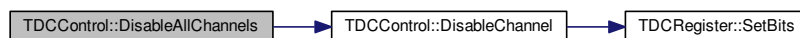


### 7.23.3.3 TDCCControl::TDCCControl ( const std::vector< uint8\_t > & words ) [inline]

## 7.23.4 Member Function Documentation

### 7.23.4.1 void TDCCControl::DisableAllChannels ( ) [inline]

Here is the call graph for this function:



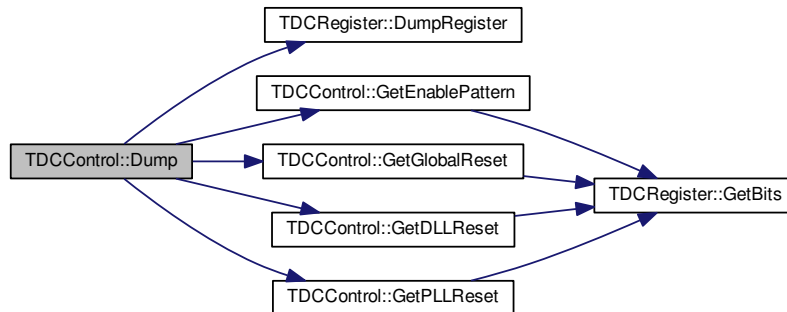
### 7.23.4.2 void TDCCControl::DisableChannel ( unsigned int id ) [inline]

Here is the call graph for this function:



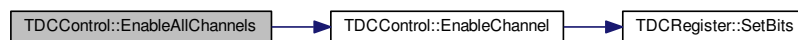
7.23.4.3 `void TDCControl::Dump ( int verb = 1, std::ostream & os = std::cout ) const` `[inline]`

Here is the call graph for this function:



7.23.4.4 `void TDCControl::EnableAllChannels ( )` `[inline]`

Here is the call graph for this function:



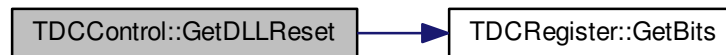
7.23.4.5 `void TDCControl::EnableChannel ( unsigned int id )` `[inline]`

Here is the call graph for this function:



#### 7.23.4.6 `bool TDCControl::GetDLLReset ( ) const [inline]`

Here is the call graph for this function:



#### 7.23.4.7 `EnablePattern TDCControl::GetEnablePattern ( ) const [inline]`

Here is the call graph for this function:



#### 7.23.4.8 `bool TDCControl::GetGlobalReset ( ) const [inline]`

Here is the call graph for this function:



#### 7.23.4.9 `bool TDCControl::GetPLLReset ( ) const [inline]`

Here is the call graph for this function:



#### 7.23.4.10 `void TDCControl::SetConstantValues ( ) [inline],[virtual]`

Ensure that the critical constant values are properly set in the register word

Implements [TDCRegister](#).

Here is the call graph for this function:



#### 7.23.4.11 `void TDCControl::SetControlParity ( const bool cp = true ) [inline],[private]`

Here is the call graph for this function:



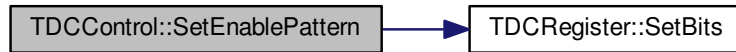
#### 7.23.4.12 `void TDCControl::SetDLLReset ( const bool dr = true ) [inline]`

Here is the call graph for this function:



7.23.4.13 `void TDCControl::SetEnablePattern ( const EnablePattern & ep ) [inline]`

Here is the call graph for this function:



7.23.4.14 `void TDCControl::SetGlobalReset ( const bool gr = true ) [inline]`

Here is the call graph for this function:



7.23.4.15 `void TDCControl::SetPLLReset ( const bool pr = true ) [inline]`

Here is the call graph for this function:



## 7.23.5 Field Documentation

7.23.5.1 `const bit TDCControl::kControlParity = 39 [static], [private]`

7.23.5.2 `const bit TDCControl::kDLLReset = 37 [static], [private]`

7.23.5.3 `const bit TDCControl::kEnableChannel = 5 [static], [private]`

7.23.5.4 `const bit TDCControl::kEnablePattern = 0 [static], [private]`

7.23.5.5 `const bit TDCControl::kGlobalReset = 4 [static], [private]`

7.23.5.6 `const bit TDCControl::kPLLReset = 38` `[static], [private]`

The documentation for this class was generated from the following file:

- `daq/include/TDCControl.h`

## 7.24 TDCErrorFlag Class Reference

Error flags handler.

```
#include <TDCEvent.h>
```

### Public Member Functions

- [TDCErrorFlag](#) (uint16\_t ef)
- virtual [~TDCErrorFlag](#) ()
- uint16\_t [GetWord](#) () const
- void [Dump](#) () const
- bool [HasReadoutFIFOOverflow](#) (unsigned int group\_id) const  
*Check whether hits have been lost from read-out FIFO overflow in a given group.*
- bool [HasL1BufferOverflow](#) (unsigned int group\_id) const  
*Check whether hits have been lost from L1 buffer overflow in a given group.*
- bool [HasGroupError](#) (unsigned int group\_id) const  
*Check whether hits have been lost due to error in a given group.*
- bool [HasReachedEventSizeLimit](#) () const  
*Hits rejected because of programmed event size limit.*
- bool [HasTriggerFIFOOverflow](#) () const  
*Event lost (trigger FIFO overflow)*
- bool [HasInternalChipError](#) () const  
*Internal fatal chip error has been detected.*

### Private Attributes

- uint16\_t [fWord](#)

### Friends

- `std::ostream & operator<<` (`std::ostream &os`, const [TDCErrorFlag](#) &ef)

#### 7.24.1 Detailed Description

Error flags handler.

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

22 Jun 2015

## 7.24.2 Constructor & Destructor Documentation

7.24.2.1 `TDCErrorFlag::TDCErrorFlag ( uint16_t ef )` `[inline]`

7.24.2.2 `virtual TDCErrorFlag::~~TDCErrorFlag ( )` `[inline],[virtual]`

## 7.24.3 Member Function Documentation

7.24.3.1 `void TDCErrorFlag::Dump ( ) const` `[inline]`

7.24.3.2 `uint16_t TDCErrorFlag::GetWord ( ) const` `[inline]`

7.24.3.3 `bool TDCErrorFlag::HasGroupError ( unsigned int group_id ) const` `[inline]`

Check whether hits have been lost due to error in a given group.

7.24.3.4 `bool TDCErrorFlag::HasInternalChipError ( ) const` `[inline]`

Internal fatal chip error has been detected.

7.24.3.5 `bool TDCErrorFlag::HasL1BufferOverflow ( unsigned int group_id ) const` `[inline]`

Check whether hits have been lost from L1 buffer overflow in a given group.

7.24.3.6 `bool TDCErrorFlag::HasReachedEventSizeLimit ( ) const` `[inline]`

Hits rejected because of programmed event size limit.

7.24.3.7 `bool TDCErrorFlag::HasReadoutFIFOOverflow ( unsigned int group_id ) const` `[inline]`

Check whether hits have been lost from read-out FIFO overflow in a given group.

7.24.3.8 `bool TDCErrorFlag::HasTriggerFIFOOverflow ( ) const` `[inline]`

Event lost (trigger FIFO overflow)

## 7.24.4 Friends And Related Function Documentation

7.24.4.1 `std::ostream& operator<< ( std::ostream & os, const TDCErrorFlag & ef )` `[friend]`

## 7.24.5 Field Documentation

7.24.5.1 `uint16_t TDCErrorFlag::fWord` `[private]`

The documentation for this class was generated from the following file:

- include/TDCEvent.h

## 7.25 TDCEvent Class Reference

HPTDC event parser.

```
#include <TDCEvent.h>
```

## Public Types

- enum [EventType](#) {  
[TDCMeasurement](#) = 0x0, [TDCHeader](#) = 0x1, [TDCTrailer](#) = 0x3, [TDCError](#) = 0x4,  
[GlobalHeader](#) = 0x8, [GlobalTrailer](#) = 0x10, [ETTT](#) = 0x11, [Filler](#) = 0x18,  
[Trigger](#) = 0x1f }

## Public Member Functions

- [TDCEvent](#) ()
- [TDCEvent](#) (const [TDCEvent](#) &ev)
- [TDCEvent](#) (const uint32\_t &word)
- [TDCEvent](#) (const [EventType](#) &ev)
- virtual [~TDCEvent](#) ()
- void [Dump](#) () const
- void [SetWord](#) (const uint32\_t &word)
- uint32\_t [GetWord](#) () const
- [EventType](#) [GetType](#) () const  
*Type of packet read out from the TDC.*
- unsigned int [GetTDCId](#) () const  
*Programmed identifier of master TDC providing the event.*
- uint16\_t [GetEventId](#) () const  
*Event identifier from event counter.*
- uint16\_t [GetWordCount](#) () const  
*Total number of words in event (including headers and trailers)*
- unsigned int [GetGeo](#) () const
- unsigned int [GetChannelId](#) () const  
*Channel number for.*
- uint32\_t [GetEventCount](#) () const  
*Total number of events.*
- uint16\_t [GetBunchId](#) () const  
*Bunch identifier of trigger (or trigger time tag)*
- bool [IsTrailing](#) () const  
*Are we dealing with a trailing or a leading measurement?*
- uint32\_t [GetETTT](#) () const  
*Extended trigger time tag.*
- uint32\_t [GetTime](#) (bool pair=false) const  
*Edge measurement in programmed time resolution.*
- unsigned int [GetWidth](#) () const  
*Width of pulse in programmed time resolution.*
- unsigned int [GetStatus](#) () const
- [TDCErrorFlag](#) [GetErrorFlags](#) () const  
*Return error flags if an error condition has been detected.*

## Private Attributes

- uint32\_t [fWord](#)



### 7.25.1 Detailed Description

HPTDC event parser.

Object enabling to decipher any measurement/error/debug event returned by the HPTDC chip

Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

Date

4 May 2015

### 7.25.2 Member Enumeration Documentation

#### 7.25.2.1 enum TDCEvent::EventType

Enumerator

***TDCTMeasurement***

***TDCHheader***

***TDCTTrailer***

***TDCEerror***

***GlobalHeader***

***GlobalTrailer***

***ETTT***

***Filler***

***Trigger***

### 7.25.3 Constructor & Destructor Documentation

7.25.3.1 TDCEvent::TDCEvent ( ) [inline]

7.25.3.2 TDCEvent::TDCEvent ( const TDCEvent & ev ) [inline]

7.25.3.3 TDCEvent::TDCEvent ( const uint32\_t & word ) [inline]

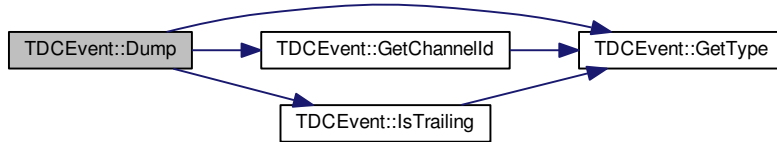
7.25.3.4 TDCEvent::TDCEvent ( const EventType & ev ) [inline]

7.25.3.5 virtual TDCEvent::~TDCEvent ( ) [inline],[virtual]

### 7.25.4 Member Function Documentation

#### 7.25.4.1 void TDCEvent::Dump ( ) const [inline]

Here is the call graph for this function:



#### 7.25.4.2 uint16\_t TDCEvent::GetBunchId ( ) const [inline]

Bunch identifier of trigger (or trigger time tag)

Here is the call graph for this function:



#### 7.25.4.3 unsigned int TDCEvent::GetChannelId ( ) const [inline]

Channel number for.

Here is the call graph for this function:



#### 7.25.4.4 TDCErrorFlag TDCEvent::GetErrorFlags ( ) const [inline]

Return error flags if an error condition has been detected.

Here is the call graph for this function:



#### 7.25.4.5 `uint32_t TDCEvent::GetETTT ( ) const [inline]`

Extended trigger time tag.

Here is the call graph for this function:



#### 7.25.4.6 `uint32_t TDCEvent::GetEventCount ( ) const [inline]`

Total number of events.

Here is the call graph for this function:



#### 7.25.4.7 `uint16_t TDCEvent::GetEventId ( ) const [inline]`

Event identifier from event counter.

Here is the call graph for this function:



7.25.4.8 `unsigned int TDCEvent::GetGeo ( ) const [inline]`

Here is the call graph for this function:



7.25.4.9 `unsigned int TDCEvent::GetStatus ( ) const [inline]`

Here is the call graph for this function:



7.25.4.10 `unsigned int TDCEvent::GetTDCId ( ) const [inline]`

Programmed identifier of master TDC providing the event.

Here is the call graph for this function:



**7.25.4.11** `uint32_t TDCEvent::GetTime ( bool pair = false ) const [inline]`

Edge measurement in programmed time resolution.

Parameters

<i>in</i>	<i>pair</i>	Are we dealing with a pair measurement? (only for leading time word)
-----------	-------------	--

Here is the call graph for this function:



**7.25.4.12** `EventType TDCEvent::GetType ( ) const [inline]`

Type of packet read out from the TDC.

**7.25.4.13** `unsigned int TDCEvent::GetWidth ( ) const [inline]`

Width of pulse in programmed time resolution.

Here is the call graph for this function:



7.25.4.14 `uint32_t TDCEvent::GetWord ( ) const [inline]`

7.25.4.15 `uint16_t TDCEvent::GetWordCount ( ) const [inline]`

Total number of words in event (including headers and trailers)

Here is the call graph for this function:



7.25.4.16 `bool TDCEvent::IsTrailing ( ) const [inline]`

Are we dealing with a trailing or a leading measurement?

Here is the call graph for this function:



7.25.4.17 `void TDCEvent::SetWord ( const uint32_t & word ) [inline]`

## 7.25.5 Field Documentation

7.25.5.1 `uint32_t TDCEvent::fWord [private]`

The documentation for this class was generated from the following file:

- `include/TDCEvent.h`

## 7.26 TDCMeasurement Class Reference

```
#include <TDCMeasurement.h>
```

### Public Member Functions

- [TDCMeasurement \(\)](#)
- [TDCMeasurement \(const std::vector< \[TDCEvent\]\(#\) > &v\)](#)

- [~TDCMeasurement](#) ()
- void [Dump](#) ()
- void [SetEventsCollection](#) (const std::vector< [TDCEvent](#) > &v)
- uint32\_t [GetLeadingTime](#) (unsigned short event\_id=0)
- uint32\_t [GetTrailingTime](#) (unsigned short event\_id=0)
- uint16\_t [GetToT](#) (unsigned short event\_id=0)
- uint16\_t [GetChannelId](#) (unsigned short event\_id=0)
- uint16\_t [GetTDCId](#) ()
- uint16\_t [GetEventId](#) ()
- uint16\_t [GetBunchId](#) ()
- uint32\_t [GetETTT](#) ()
- size\_t [NumEvents](#) () const
- size\_t [NumErrors](#) () const

### Private Attributes

- std::map< [TDCEvent::EventType](#), [TDCEvent](#) > [fMap](#)
- std::vector< std::pair< [TDCEvent](#), [TDCEvent](#) > > [fEvents](#)

### 7.26.1 Detailed Description

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

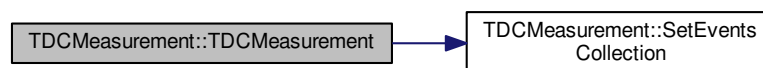
Jun 2015

### 7.26.2 Constructor & Destructor Documentation

7.26.2.1 [TDCMeasurement::TDCMeasurement \( \)](#) `[inline]`

7.26.2.2 [TDCMeasurement::TDCMeasurement \( const std::vector< \[TDCEvent\]\(#\) > & v \)](#) `[inline]`

Here is the call graph for this function:

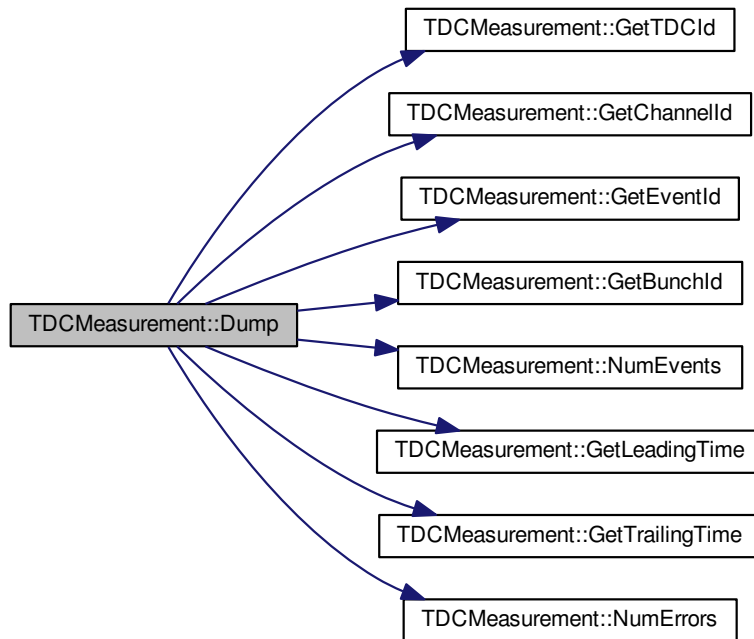


7.26.2.3 [TDCMeasurement::~~TDCMeasurement \( \)](#) `[inline]`

### 7.26.3 Member Function Documentation

### 7.26.3.1 void TDCMeasurement::Dump ( ) [inline]

Here is the call graph for this function:



### 7.26.3.2 uint16\_t TDCMeasurement::GetBunchId ( ) [inline]

### 7.26.3.3 uint16\_t TDCMeasurement::GetChannelId ( unsigned short *event\_id* = 0 ) [inline]

### 7.26.3.4 uint32\_t TDCMeasurement::GetETTT ( ) [inline]

### 7.26.3.5 uint16\_t TDCMeasurement::GetEventId ( ) [inline]

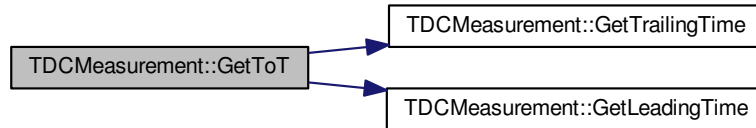
### 7.26.3.6 uint32\_t TDCMeasurement::GetLeadingTime ( unsigned short *event\_id* = 0 ) [inline]

### 7.26.3.7 uint16\_t TDCMeasurement::GetTDCId ( ) [inline]



7.26.3.8 `uint16_t TDCMeasurement::GetToT ( unsigned short event_id = 0 ) [inline]`

Here is the call graph for this function:



7.26.3.9 `uint32_t TDCMeasurement::GetTrailingTime ( unsigned short event_id = 0 ) [inline]`

7.26.3.10 `size_t TDCMeasurement::NumErrors ( ) const [inline]`

7.26.3.11 `size_t TDCMeasurement::NumEvents ( ) const [inline]`

7.26.3.12 `void TDCMeasurement::SetEventsCollection ( const std::vector< TDCEvent > & v ) [inline]`

## 7.26.4 Field Documentation

7.26.4.1 `std::vector< std::pair<TDCEvent,TDCEvent> > TDCMeasurement::fEvents [private]`

7.26.4.2 `std::map<TDCEvent::EventType,TDCEvent> TDCMeasurement::fMap [private]`

The documentation for this class was generated from the following file:

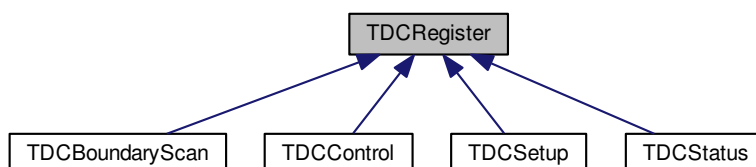
- `include/TDCMeasurement.h`

## 7.27 TDCRegister Class Reference

General register object to interact with a HPTDC chip.

```
#include <TDCRegister.h>
```

Inheritance diagram for TDCRegister:



## Public Types

- typedef uint16\_t [bit](#)  
*LSB index.*
- typedef uint32\_t [word\\_t](#)  
*Unit of the TDC register word to be successfully contained on any machine.*

## Public Member Functions

- [TDCRegister](#) (const unsigned int size)
- [TDCRegister](#) (const unsigned int size, const [TDCRegister](#) &r)
- [TDCRegister](#) (const unsigned int size, const std::vector< uint8\_t > words)
- virtual ~[TDCRegister](#) ()
- [TDCRegister](#) & operator= (const [TDCRegister](#) &r)
- void [SetWord](#) (const unsigned int i, const [word\\_t](#) word)  
*Set one bit(s) subset in the register word.*
- [word\\_t](#) [GetWord](#) (const unsigned int i) const  
*Retrieve one subset from the register word.*
- [word\\_t](#) \* [GetWords](#) () const
- uint8\_t [GetNumWords](#) () const  
*Number of words in the register.*
- void [DumpRegister](#) (std::ostream &os=std::cout, const [bit](#) max\_bits=-1) const
- virtual void [SetConstantValues](#) ()=0

## Protected Member Functions

- void [SetBits](#) (uint16\_t lsb, uint16\_t word, uint8\_t size)  
*Set bits in the register word.*
- uint16\_t [GetBits](#) (uint16\_t lsb, uint8\_t size) const  
*Extract bits from the register word.*
- void [Clear](#) ()  
*Set all bits in this register to '0'.*

## Protected Attributes

- [word\\_t](#) \* [fWord](#)  
*Pointer to this register's word.*
- unsigned int [fNumWords](#)
- unsigned int [fWordSize](#)  
*Number of bits in this register.*

### 7.27.1 Detailed Description

General register object to interact with a HPTDC chip.

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

24 Apr 2015

## 7.27.2 Member Typedef Documentation

### 7.27.2.1 `typedef uint16_t TDCRegister::bit`

LSB index.

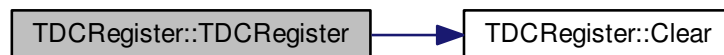
### 7.27.2.2 `typedef uint32_t TDCRegister::word_t`

Unit of the TDC register word to be successfully contained on any machine.

## 7.27.3 Constructor & Destructor Documentation

### 7.27.3.1 `TDCRegister::TDCRegister ( const unsigned int size ) [inline]`

Here is the call graph for this function:



### 7.27.3.2 `TDCRegister::TDCRegister ( const unsigned int size, const TDCRegister & r ) [inline]`

Here is the call graph for this function:



### 7.27.3.3 `TDCRegister::TDCRegister ( const unsigned int size, const std::vector< uint8_t > words ) [inline]`

Here is the call graph for this function:



7.27.3.4 `virtual TDCRegister::~~TDCRegister ( ) [inline], [virtual]`

## 7.27.4 Member Function Documentation

7.27.4.1 `void TDCRegister::Clear ( ) [inline], [protected]`

Set all bits in this register to '0'.

7.27.4.2 `void TDCRegister::DumpRegister ( std::ostream & os = std::cout, const bit max_bits = -1 ) const [inline]`

7.27.4.3 `uint16_t TDCRegister::GetBits ( uint16_t lsb, uint8_t size ) const [inline], [protected]`

Extract bits from the register word.

Extract a fixed amount of bits from the full register word

### Parameters

<i>in</i>	<i>lsb</i>	Least significant bit of the word to retrieve
<i>in</i>	<i>size</i>	Size of the word to retrieve

7.27.4.4 `uint8_t TDCRegister::GetNumWords ( ) const [inline]`

Number of words in the register.

Return the number of words making up the full register word.

7.27.4.5 `word_t TDCRegister::GetWord ( const unsigned int i ) const [inline]`

Retrieve one subset from the register word.

7.27.4.6 `word_t* TDCRegister::GetWords ( ) const [inline]`

7.27.4.7 `TDCRegister& TDCRegister::operator= ( const TDCRegister & r ) [inline]`

7.27.4.8 `void TDCRegister::SetBits ( uint16_t lsb, uint16_t word, uint8_t size ) [inline], [protected]`

Set bits in the register word.

Set a fixed amount of bits in the full register word

### Parameters

<i>in</i>	<i>lsb</i>	Least significant bit of the word to set
<i>in</i>	<i>word</i>	Word to set
<i>in</i>	<i>size</i>	Size of the word to set

7.27.4.9 `virtual void TDCRegister::SetConstantValues ( ) [pure virtual]`

Ensure that the critical constant values are properly set in the register word

Implemented in [TDCSetup](#), [TDCControl](#), [TDCBoundaryScan](#), and [TDCStatus](#).

7.27.4.10 `void TDCRegister::SetWord ( const unsigned int i, const word_t word )` `[inline]`

Set one bit(s) subset in the register word.

### 7.27.5 Field Documentation

7.27.5.1 `unsigned int TDCRegister::fNumWords` `[protected]`

Number of words to fit the *fWordSize* bits of this register to this object

7.27.5.2 `word_t* TDCRegister::fWord` `[protected]`

Pointer to this register's word.

7.27.5.3 `unsigned int TDCRegister::fWordSize` `[protected]`

Number of bits in this register.

The documentation for this class was generated from the following file:

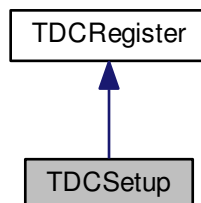
- `daq/include/TDCRegister.h`

## 7.28 TDCSetup Class Reference

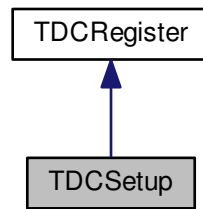
Setup word to be sent to the HPTDC chip.

```
#include <TDCSetup.h>
```

Inheritance diagram for TDCSetup:



Collaboration diagram for TDCSetup:



## Public Types

- enum [EdgeResolution](#) {  
[E\\_100ps](#) =0, [E\\_200ps](#), [E\\_400ps](#), [E\\_800ps](#),  
[E\\_1p6ns](#), [E\\_3p12ns](#), [E\\_6p25ns](#), [E\\_12p5ns](#) }
- enum [DeadTime](#) { [DT\\_5ns](#) =0, [DT\\_10ns](#), [DT\\_30ns](#), [DT\\_100ns](#) }
- enum [WidthResolution](#) {  
[W\\_100ps](#) =0, [W\\_200ps](#), [W\\_400ps](#), [W\\_800ps](#),  
[W\\_1p6ns](#), [W\\_3p2ns](#), [W\\_6p25ns](#), [W\\_12p5ns](#),  
[W\\_25ns](#), [W\\_50ns](#), [W\\_100ns](#), [W\\_200ns](#),  
[W\\_400ns](#), [W\\_800ns](#) }
- enum [EnabledError](#) {  
[VernierError](#) =0x1, [CoarseError](#) =0x2, [ChannelSelectError](#) =0x4, [L1BufferParityError](#) =0x8,  
[TriggerFIFOParityError](#) =0x10, [TriggerMatchingError](#) =0x20, [ReadoutFIFOParityError](#) =0x40, [ReadoutStateError](#) =0x80,  
[SetupParityError](#) =0x100, [ControlParityError](#) =0x200, [JTAGInstructionParityError](#) =0x400 }
- enum [DLLSpeedMode](#) { [DLL\\_40MHz](#) =0x0, [DLL\\_160MHz](#) =0x1, [DLL\\_320MHz](#) =0x2, [DLL\\_Illegal](#) =0x3 }
- enum [SerialClockSource](#) { [Serial\\_pll\\_clock\\_80](#) =0x0, [Serial\\_pll\\_clock\\_160](#) =0x1, [Serial\\_pll\\_clock\\_40](#) =0x2, [Serial\\_aux\\_clock](#) =0x3 }
- enum [IOClockSource](#) { [IO\\_clock\\_40](#) =0x0, [IO\\_pll\\_clock\\_80](#) =0x1, [IO\\_pll\\_clock\\_160](#) =0x2, [IO\\_aux\\_clock](#) =0x3 }
- enum [CoreClockSource](#) { [Core\\_clock\\_40](#) =0x0, [Core\\_pll\\_clock\\_80](#) =0x1, [Core\\_pll\\_clock\\_160](#) =0x2, [Core\\_aux\\_clock](#) =0x3 }
- enum [DLLClockSource](#) {  
[DLL\\_clock\\_40](#) =0x0, [DLL\\_pll\\_clock\\_40](#) =0x1, [DLL\\_pll\\_clock\\_160](#) =0x2, [DLL\\_pll\\_clock\\_320](#) =0x3,  
[DLL\\_aux\\_clock](#) =0x4 }
- enum [ReadoutSpeed](#) { [RO\\_Fixed](#) =0x0, [RO\\_pll\\_80Mbits\\_s](#) =0x1 }
- enum [SerialStrobeType](#) { [SS\\_NoStrobe](#) =0x0, [SS\\_DSStrobe](#) =0x1, [SS\\_LeadingTrailingStrobe](#) =0x2, [SS\\_LeadingEdge](#) =0x3 }
- enum [ReadoutSingleCycleSpeed](#) {  
[RSC\\_40Mbits\\_s](#) =0x0, [RSC\\_20Mbits\\_s](#) =0x1, [RSC\\_10Mbits\\_s](#) =0x2, [RSC\\_5Mbits\\_s](#) =0x3,  
[RSC\\_2p5Mbits\\_s](#) =0x4, [RSC\\_1p25Mbits\\_s](#) =0x5, [RSC\\_625kbits\\_s](#) =0x6, [RSC\\_312p5kbits\\_s](#) =0x7 }

## Public Member Functions

- [TDCSetup](#) ()
- [TDCSetup](#) (const [TDCSetup](#) &c)
- void [SetEnableErrorMark](#) (const bool em)  
*Mark events with error if global error signal is set.*

- bool [GetEnableErrorMark](#) () const
- void [SetEnableErrorBypass](#) (const bool eb)  
*Bypass TDC chip if global error signal is set.*
- bool [GetEnableErrorBypass](#) () const
- void [SetEnableError](#) (const uint16\_t &err)  
*Enable internal error types for generation of global error signals.*
- uint16\_t [GetEnableError](#) () const
- void [SetEnableSerial](#) (const bool es)  
*Enable of serial read-out (otherwise parallel read-out)*
- bool [GetEnableSerial](#) () const
- void [SetEnableJTAGReadout](#) (const bool jr)  
*Enable of read-out via JTAG.*
- bool [GetEnableJTAGReadout](#) () const
- void [SetReadoutFIFOSize](#) (int rfs)  
*Effective size of readout FIFO.*
- int [GetReadoutFIFOSize](#) () const
- void [SetRejectCountOffset](#) (uint16\_t rco)  
*Set the offset in reject counter (defines reject latency together with coarse count offset)*
- uint16\_t [GetRejectCountOffset](#) () const  
*Extract the offset in reject counter.*
- void [SetSearchWindow](#) (uint16\_t sw)  
*Set the search window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)*
- uint16\_t [GetSearchWindow](#) () const  
*Extract the search window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)*
- void [SetMatchWindow](#) (uint16\_t mw)  
*Set the matching window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)*
- uint16\_t [GetMatchWindow](#) () const  
*Extract the matching window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)*
- void [SetEdgeResolution](#) (const [EdgeResolution](#) r)
- [EdgeResolution](#) [GetEdgeResolution](#) () const
- void [SetMaxEventSize](#) (int sz=-1)  
*Set the maximum number of hits per event.*
- uint8\_t [GetMaxEventSize](#) () const  
*Extract the maximum number of hits per event.*
- void [SetRejectFIFOFull](#) (const bool rej=true)  
*Reject hits when readout FIFO full.*
- bool [GetRejectFIFOFull](#) () const  
*Are hits rejected when readout FIFO is full?*
- void [SetEnableReadoutOccupancy](#) (const bool ro=true)  
*Enable the readout of buffer occupancies for each event (for debugging purposes)*
- bool [GetEnableReadoutOccupancy](#) () const
- void [SetEnableReadoutSeparator](#) (const bool ro=true)  
*Enable the readout of separators for each event (for debugging purposes, valid if readout of occupancies is enabled)*
- bool [GetEnableReadoutSeparator](#) () const
- void [SetEventCountOffset](#) (uint16\_t eco)  
*Set offset for the event counter.*
- void [SetTriggerCountOffset](#) (uint16\_t tco)  
*Set offset for the trigger time tag counter to set effective trigger latency.*
- uint16\_t [GetTriggerCountOffset](#) () const  
*Extract trigger time tag count offset.*
- void [SetChannelOffset](#) (int channel, uint16\_t offset)

- Set the time offset for one single channel.*

  - `uint16_t GetChannelOffset (int channel) const`

*Return the offset for one single channel.*
- `void SetAllChannelsOffset (uint16_t offset)`

*Set the time offset for all channels.*
- `void SetCoarseCountOffset (uint16_t cco)`

*Set offset for the coarse time counter.*
- `uint16_t GetCoarseCountOffset () const`

*Extract offset for the coarse time counter.*
- `void SetDLLAdjustment (int tap, uint8_t adj)`

*Set the DLL taps adjustments with a resolution of  $\sim 10$  ps.*
- `uint8_t GetDLLAdjustment (int tap) const`

*Set the adjustment of DLL taps.*
- `void SetAllTapsDLLAdjustment (uint8_t adj)`

*Extract the adjustment of DLL taps.*
- `void SetRCAdjustment (int tap, uint8_t adj)`

*Set the adjustment of the RC delay line.*
- `uint8_t GetRCAdjustment (int tap)`

*Extract the adjustment of the RC delay line.*
- `void SetWidthResolution (const WidthResolution r)`

*Set the pulse width resolution when paired measurements are performed.*
- `WidthResolution GetWidthResolution () const`

*Extract the pulse width resolution when paired measurements are performed.*
- `void SetVernierOffset (const uint8_t vo)`

*Set the offset in vernier decoding.*
- `uint8_t GetVernierOffset () const`

*Extract the offset in vernier decoding.*
- `void SetDeadTime (const DeadTime dt)`

*Channel dead time between hits.*
- `DeadTime GetDeadTime () const`
- `void SetTestInvert (const bool ti=true)`

*Automatic inversion of test pattern. Only used during production testing.*
- `bool GetTestInvert () const`
- `void SetTestMode (const bool tm=true)`

*Test mode where hit data are taken from coretest. Only used during production testing.*
- `bool GetTestMode () const`
- `void SetTrailingMode (const bool trail=true)`

*Enable/disable the detection of trailing edges.*
- `bool GetTrailingMode () const`

*Extract the status for the detection of trailing edges.*
- `void SetLeadingMode (const bool lead=true)`

*Enable the detection of leading edges.*
- `bool GetLeadingMode () const`

*Extract the status for the detection of leading edges.*
- `void SetTriggerMatchingMode (const bool trig=true)`

*Set the enable status of trigger matching mode.*
- `bool GetTriggerMatchingMode () const`

*Extract the enable status of trigger matching mode.*
- `void SetEdgesPairing (const bool pair=true)`

*Enable the pairing of leading and trailing edges (overrides individual enable of leading/trailing edges)*
- `bool GetEdgesPairing () const`



- void [SetSetupParity](#) (const bool sp=true)  
*Set the parity of setup data (should be an even parity)*
- bool [GetSetupParity](#) () const  
*Extract the parity of setup data (should be an even parity)*
- void [SetConstantValues](#) ()  
*Ensure that the critical constant values are properly set in the setup word.*
- uint16\_t [GetTriggerLatency](#) () const  
*Effective trigger latency in number of clock cycles (when no counter roll-over is used)*
- void [SetTDCId](#) (const uint8\_t id=0x0)
- uint16\_t [GetTDCId](#) () const
- void [Dump](#) (int verb=1, std::ostream &os=std::cout) const

### Private Member Functions

- void [SetReadoutSingleCycleSpeed](#) (const [ReadoutSingleCycleSpeed](#) rscs=[RSC\\_40Mbits\\_s](#))  
*Serial transmission speed in single cycle mode.*
- void [SetSerialDelay](#) (const uint8\_t sd=0x0)  
*Programmable delay of serial input, in time unit ~ 1 ns.*
- void [SetStrobeSelect](#) (const [SerialStrobeType](#) ss=[SS\\_NoStrobe](#))
- void [SetReadoutSpeedSelect](#) (const [ReadoutSpeed](#) rss=[RO\\_Fixed](#))  
*Selection of serial read-out speed.*
- void [SetTokenDelay](#) (const uint8\_t td=0x0)  
*Programmable delay of token input, in time unit ~ 1 ns.*
- void [SetEnableLocalTrailer](#) (const bool elt=true)  
*Enable of local trailers in read-out.*
- void [SetEnableLocalHeader](#) (const bool elh=true)  
*Enable of local headers in read-out.*
- void [SetEnableGlobalTrailer](#) (const bool egt=true)  
*Enable of global trailers in read-out (only valid for master TDC)*
- void [SetEnableGlobalHeader](#) (const bool egh=true)  
*Enable of global headers in read-out (only valid for master TDC)*
- void [SetKeepToken](#) (const bool kt=true)
- void [SetMaster](#) (const bool m=true)
- void [SetEnableBytewise](#) (const bool seb=true)
- void [SetBypassInputs](#) (const bool sbi=true)  
*Select serial in and token in from bypass inputs.*
- void [SetEnableOverflowDetect](#) (const bool eod=true)  
*Enable overflow detection of L1 buffers (should always be enabled!)*
- void [SetEnableRelative](#) (const bool er=true)
- void [SetEnableAutomaticReject](#) (const bool ear=true)  
*Enable of automatic rejection (should always be enabled if trigger matching mode!)*
- void [SetEnableSetCountersOnBunchReset](#) (const bool escobr=true)  
*Enable all counters to be set on bunch count reset.*
- void [SetEnableMasterResetCode](#) (const bool emrc=true)  
*Enable master reset code on encoded\_control.*
- void [SetEnableMasterResetOnEventReset](#) (const bool emroer=true)  
*Enable master reset of whole TDC on event reset.*
- void [SetEnableResetChannelBufferWhenSeparator](#) (const bool ercbws=true)  
*Enable reset channel buffers when separator.*
- void [SetEnableSeparatorOnEventReset](#) (const bool esoer=true)  
*Enable generation of separator on event reset.*

- void [SetEnableSeparatorOnBunchReset](#) (const bool esobr=true)  
*Enable generation of separator on bunch reset.*
- void [SetEnableDirectEventReset](#) (const bool eder=true)  
*Enable of direct event reset input pin (1), otherwise taken from encoded control.*
- void [SetEnableDirectBunchReset](#) (const bool edbr=true)  
*Enable of direct bunch reset input pin (1), otherwise taken from encoded control.*
- void [SetEnableDirectTrigger](#) (const bool edt=true)  
*Enable of direct trigger input pin.*
- void [SetLowPowerMode](#) (const bool lpm=true)  
*Low power mode of channel buffers.*
- void [SetDLLControl](#) (const uint8\_t dc)  
*Control of DLL (DLL charge pump levels)*
- void [SetModeRCCompression](#) (const bool mrc=true)  
*Perform RC interpolation on-chip (only valid in very high resolution mode)*
- void [SetModeRC](#) (const bool mr=true)  
*Enable of RR delay lines mode (in very high resolution mode) ; only for channels 0-4-8-12-16-20-24-28 active.*
- void [SetDLLMode](#) (const [DLLSpeedMode](#) dsm)  
*Selection of DLL speed mode.*
- void [SetPLLControl](#) (const uint8\_t charge\_pump\_current=0x4, const bool power\_down\_mode=false, const bool enable\_test\_outputs=false, const bool invert\_connection\_to\_status=false)  
*Control of PLL.*
- void [SetSerialClockDelay](#) (const bool delay\_clock, const uint8\_t delay)  
*Delay of internal serial clock.*
- void [SetIOClockDelay](#) (const bool delay\_clock, const uint8\_t delay)  
*Delay of internal I/O clock.*
- void [SetCoreClockDelay](#) (const bool delay\_clock, const uint8\_t delay)  
*Delay of internal core clock.*
- void [SetDLLClockDelay](#) (const bool delay\_clock, const uint8\_t delay)  
*Delay of internal DLL clock.*
- void [SetSerialClockSource](#) (const [SerialClockSource](#) scs)  
*Selection of source for serial clock.*
- void [SetIOClockSource](#) (const [IOClockSource](#) ics)  
*Selection of clock source for I/O signals.*
- void [SetCoreClockSource](#) (const [CoreClockSource](#) ccs)  
*Selection of clock source for internal logic.*
- void [SetDLLClockSource](#) (const [DLLClockSource](#) dcs)  
*Selection of clock source for DLL.*
- void [SetRollOver](#) (const uint16\_t ro=0xFFFF)  
*Counter roll over value, defining maximal count value from where counters will be reset to 0.*
- void [SetEnableTTLSerial](#) (const bool ts=true)  
*Enable LV TTL inputs on serial registers, and disable their drivers.*
- void [SetEnableTTLControl](#) (const bool tc=true)  
*Enable LV TTL inputs on control registers.*
- void [SetEnableTTLReset](#) (const bool tr=true)  
*Enable LV TTL input on reset, otherwise uses LVDS input levels.*
- void [SetEnableTTLClock](#) (const bool tc=true)  
*Enable LV TTL inputs on: clk, aux\_clock, otherwise uses LVDS input levels.*
- void [SetEnableTTLHit](#) (const bool th=true)  
*Enable LV TTL input on hit[31:0], otherwise uses LVDS input levels.*
- void [SetTest](#) (const bool test=true)

### Static Private Attributes

- static const [bit kTestSelect](#) = 0
- static const [bit kEnableErrorMark](#) = 4
- static const [bit kEnableErrorBypass](#) = 5
- static const [bit kEnableError](#) = 6
- static const [bit kReadoutSingleCycleSpeed](#) = 17
- static const [bit kSerialDelay](#) = 20
- static const [bit kStrobeSelect](#) = 24
- static const [bit kReadoutSpeedSelect](#) = 26
- static const [bit kTokenDelay](#) = 27
- static const [bit kEnableLocalTrailer](#) = 31
- static const [bit kEnableLocalHeader](#) = 32
- static const [bit kEnableGlobalTrailer](#) = 33
- static const [bit kEnableGlobalHeader](#) = 34
- static const [bit kKeepToken](#) = 35
- static const [bit kMaster](#) = 36
- static const [bit kEnableBytewise](#) = 37
- static const [bit kEnableSerial](#) = 38
- static const [bit kEnableJTAGReadout](#) = 39
- static const [bit kTDCId](#) = 40
- static const [bit kSelectBypassInputs](#) = 44
- static const [bit kReadoutFIFOSize](#) = 45
- static const [bit kRejectCountOffset](#) = 48
- static const [bit kSearchWindow](#) = 60
- static const [bit kMatchWindow](#) = 72
- static const [bit kLeadingResolution](#) = 84
- static const [bit kMaxEventSize](#) = 116
- static const [bit kRejectFIFOFull](#) = 120
- static const [bit kEnableReadoutOccupancy](#) = 121
- static const [bit kEnableReadoutSeparator](#) = 122
- static const [bit kEnableOverflowDetect](#) = 123
- static const [bit kEnableRelative](#) = 124
- static const [bit kEnableAutomaticReject](#) = 125
- static const [bit kEventCountOffset](#) = 126
- static const [bit kTriggerCountOffset](#) = 138
- static const [bit kEnableSetCountersOnBunchReset](#) = 150
- static const [bit kEnableMasterResetCode](#) = 151
- static const [bit kEnableMasterResetOnEventReset](#) = 152
- static const [bit kEnableResetChannelBufferWhenSeparator](#) = 153
- static const [bit kEnableSeparatorOnEventReset](#) = 154
- static const [bit kEnableSeparatorOnBunchReset](#) = 155
- static const [bit kEnableDirectEventReset](#) = 156
- static const [bit kEnableDirectBunchReset](#) = 157
- static const [bit kEnableDirectTrigger](#) = 158
- static const [bit kOffset0](#) = 438
- static const [bit kCoarseCountOffset](#) = 447
- static const [bit kDLLTapAdjust0](#) = 459
- static const [bit kRCAdjust0](#) = 555
- static const [bit kLowPowerMode](#) = 570
- static const [bit kWidthSelect](#) = 571
- static const [bit kVernierOffset](#) = 575
- static const [bit kDLLControl](#) = 580
- static const [bit kDeadTime](#) = 584
- static const [bit kTestInvert](#) = 586

- static const [bit kTestMode](#) = 587
- static const [bit kTrailing](#) = 588
- static const [bit kLeading](#) = 589
- static const [bit kModeRCCompression](#) = 590
- static const [bit kModeRC](#) = 591
- static const [bit kDLLMode](#) = 592
- static const [bit kPLLControl](#) = 594
- static const [bit kSerialClockDelay](#) = 602
- static const [bit kIOClockDelay](#) = 606
- static const [bit kCoreClockDelay](#) = 610
- static const [bit kDLLClockDelay](#) = 614
- static const [bit kSerialClockSource](#) = 618
- static const [bit kIOClockSource](#) = 620
- static const [bit kCoreClockSource](#) = 622
- static const [bit kDLLClockSource](#) = 624
- static const [bit kRollOver](#) = 627
- static const [bit kEnableMatching](#) = 639
- static const [bit kEnablePair](#) = 640
- static const [bit kEnableTTLSerial](#) = 641
- static const [bit kEnableTTLControl](#) = 642
- static const [bit kEnableTTLReset](#) = 643
- static const [bit kEnableTTLClock](#) = 644
- static const [bit kEnableTTLHit](#) = 645
- static const [bit kSetupParity](#) = 646

## Additional Inherited Members

### 7.28.1 Detailed Description

Setup word to be sent to the HPTDC chip.

Object handling the setup word provided by/to the HPTDC chip

Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

Date

16 Apr 2015

### 7.28.2 Member Enumeration Documentation

#### 7.28.2.1 enum TDCSetup::CoreClockSource

Enumerator

***Core\_clock\_40***

***Core\_pll\_clock\_80***

***Core\_pll\_clock\_160***

***Core\_aux\_clock***

## 7.28.2.2 enum TDCSetup::DeadTime

Enumerator

*DT\_5ns*  
*DT\_10ns*  
*DT\_30ns*  
*DT\_100ns*

## 7.28.2.3 enum TDCSetup::DLLClockSource

Enumerator

*DLL\_clock\_40*  
*DLL\_pll\_clock\_40*  
*DLL\_pll\_clock\_160*  
*DLL\_pll\_clock\_320*  
*DLL\_aux\_clock*

## 7.28.2.4 enum TDCSetup::DLLSpeedMode

Enumerator

*DLL\_40MHz*  
*DLL\_160MHz*  
*DLL\_320MHz*  
*DLL\_Illegal*

## 7.28.2.5 enum TDCSetup::EdgeResolution

Enumerator

*E\_100ps*  
*E\_200ps*  
*E\_400ps*  
*E\_800ps*  
*E\_1p6ns*  
*E\_3p12ns*  
*E\_6p25ns*  
*E\_12p5ns*

## 7.28.2.6 enum TDCSetup::EnabledError

Enumerator

*VernierError*  
*CoarseError*  
*ChannelSelectError*  
*L1BufferParityError*

*TriggerFIFOParityError*  
*TriggerMatchingError*  
*ReadoutFIFOParityError*  
*ReadoutStateError*  
*SetupParityError*  
*ControlParityError*  
*JTAGInstructionParityError*

#### 7.28.2.7 enum TDCSetup::IOClockSource

Enumerator

*IO\_clock\_40*  
*IO\_pll\_clock\_80*  
*IO\_pll\_clock\_160*  
*IO\_aux\_clock*

#### 7.28.2.8 enum TDCSetup::ReadoutSingleCycleSpeed

Enumerator

*RSC\_40Mbits\_s*  
*RSC\_20Mbits\_s*  
*RSC\_10Mbits\_s*  
*RSC\_5Mbits\_s*  
*RSC\_2p5Mbits\_s*  
*RSC\_1p25Mbits\_s*  
*RSC\_625kbits\_s*  
*RSC\_312p5kbits\_s*

#### 7.28.2.9 enum TDCSetup::ReadoutSpeed

Enumerator

*RO\_Fixed*  
*RO\_pll\_80Mbits\_s*

#### 7.28.2.10 enum TDCSetup::SerialClockSource

Enumerator

*Serial\_pll\_clock\_80*  
*Serial\_pll\_clock\_160*  
*Serial\_pll\_clock\_40*  
*Serial\_aux\_clock*

## 7.28.2.11 enum TDCSetup::SerialStrobeType

Enumerator

*SS\_NoStrobe**SS\_DSStrobe**SS\_LeadingTrailingStrobe**SS\_LeadingEdge*

## 7.28.2.12 enum TDCSetup::WidthResolution

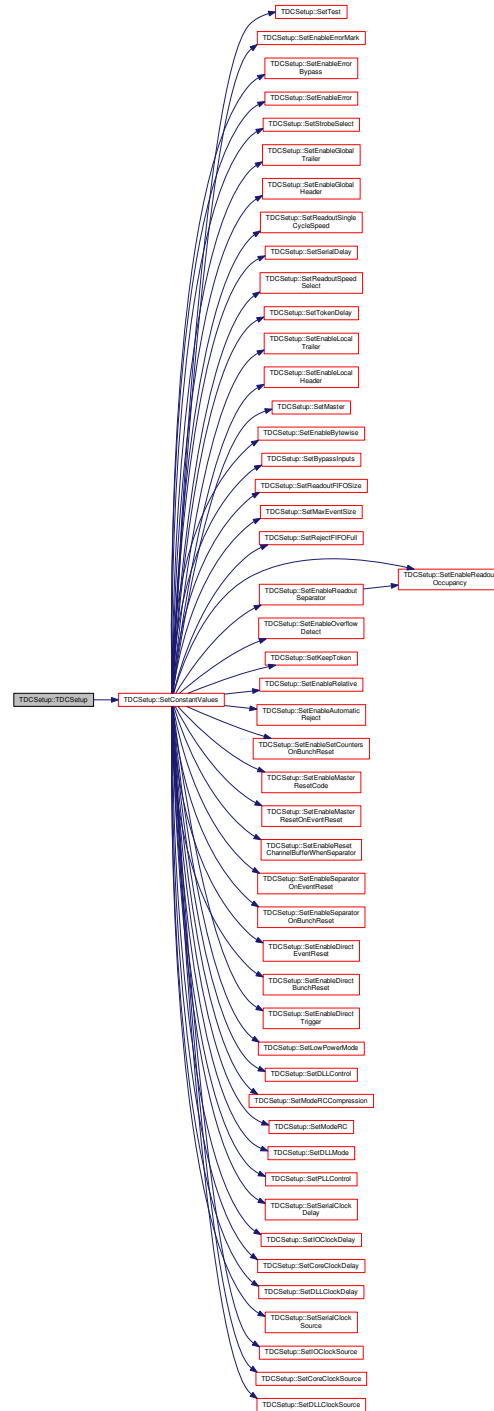
Enumerator

*W\_100ps**W\_200ps**W\_400ps**W\_800ps**W\_1p6ns**W\_3p2ns**W\_6p25ns**W\_12p5ns**W\_25ns**W\_50ns**W\_100ns**W\_200ns**W\_400ns**W\_800ns*

## 7.28.3 Constructor &amp; Destructor Documentation

### 7.28.3.1 TDCSetup::TDCSetup ( ) [inline]

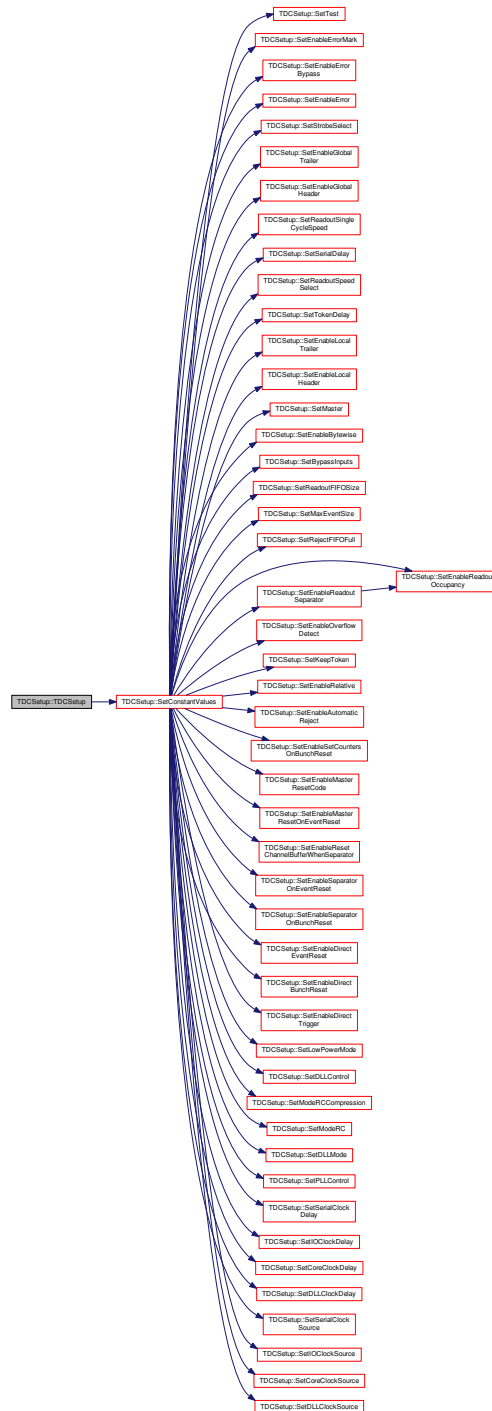
Here is the call graph for this function:





## 7.28.3.2 TDCSetup::TDCSetup ( const TDCSetup &amp; c ) [inline]

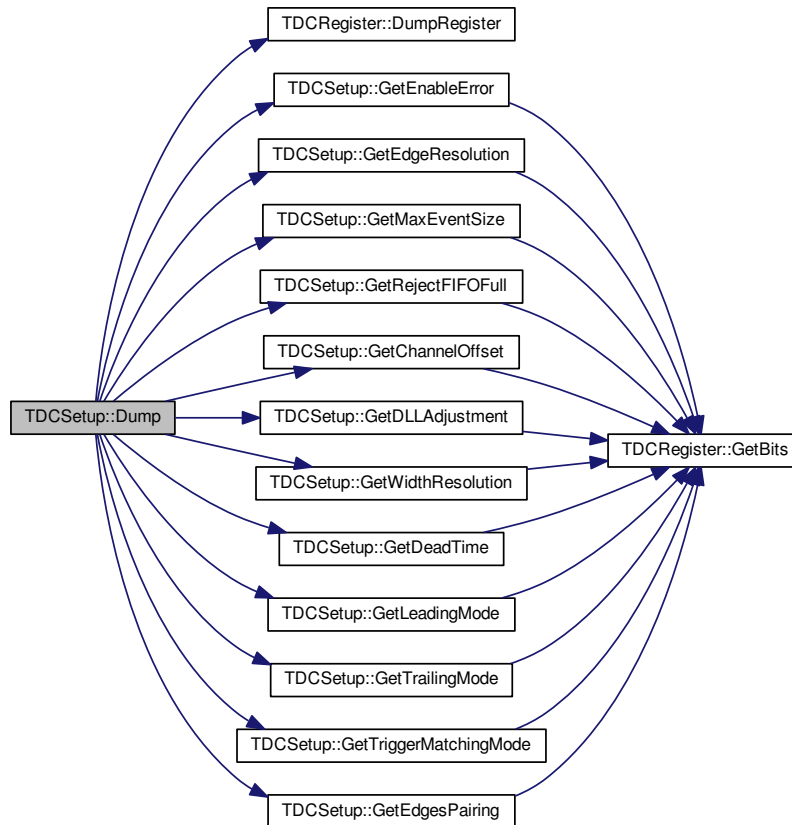
Here is the call graph for this function:



## 7.28.4 Member Function Documentation

7.28.4.1 `void TDCSetup::Dump ( int verb = 1, std::ostream & os = std::cout ) const`

Here is the call graph for this function:



7.28.4.2 `uint16_t TDCSetup::GetChannelOffset ( int channel ) const [inline]`

Return the offset for one single channel.

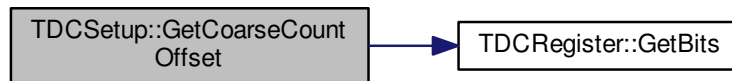
Here is the call graph for this function:



7.28.4.3 `uint16_t TDCSetup::GetCoarseCountOffset ( ) const [inline]`

Extract offset for the coarse time counter.

Here is the call graph for this function:



#### 7.28.4.4 DeadTime TDCSetup::GetDeadTime ( ) const [inline]

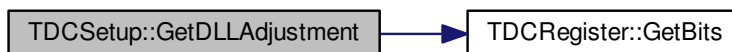
Here is the call graph for this function:



#### 7.28.4.5 uint8\_t TDCSetup::GetDLLAdjustment ( int tap ) const [inline]

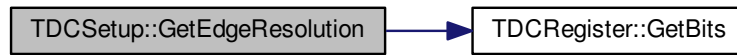
Set the adjustment of DLL taps.

Here is the call graph for this function:



#### 7.28.4.6 `EdgeResolution` `TDCSetup::GetEdgeResolution ( ) const` `[inline]`

Here is the call graph for this function:



#### 7.28.4.7 `bool` `TDCSetup::GetEdgesPairing ( ) const` `[inline]`

Here is the call graph for this function:



#### 7.28.4.8 `uint16_t` `TDCSetup::GetEnableError ( ) const` `[inline]`

Here is the call graph for this function:



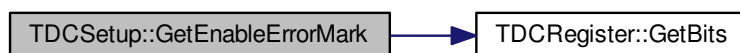
#### 7.28.4.9 bool TDCSetup::GetEnableErrorBypass ( ) const [inline]

Here is the call graph for this function:



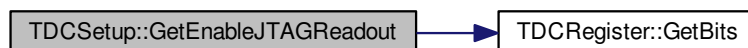
#### 7.28.4.10 bool TDCSetup::GetEnableErrorMark ( ) const [inline]

Here is the call graph for this function:



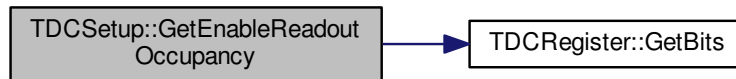
#### 7.28.4.11 bool TDCSetup::GetEnableJTAGReadout ( ) const [inline]

Here is the call graph for this function:



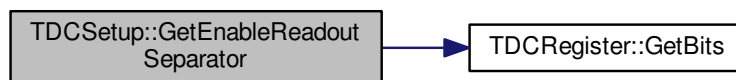
7.28.4.12 `bool TDCSetup::GetEnableReadoutOccupancy ( ) const [inline]`

Here is the call graph for this function:



7.28.4.13 `bool TDCSetup::GetEnableReadoutSeparator ( ) const [inline]`

Here is the call graph for this function:



7.28.4.14 `bool TDCSetup::GetEnableSerial ( ) const [inline]`

Here is the call graph for this function:



7.28.4.15 `bool TDCSetup::GetLeadingMode ( ) const [inline]`

Extract the status for the detection of leading edges.

Here is the call graph for this function:



#### 7.28.4.16 `uint16_t TDCSetup::GetMatchWindow ( ) const [inline]`

Extract the matching window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)

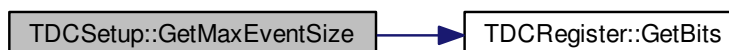
Here is the call graph for this function:



#### 7.28.4.17 `uint8_t TDCSetup::GetMaxEventSize ( ) const [inline]`

Extract the maximum number of hits per event.

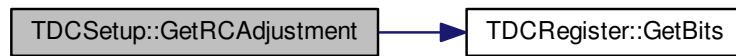
Here is the call graph for this function:



#### 7.28.4.18 `uint8_t TDCSetup::GetRCAdjustment ( int tap ) [inline]`

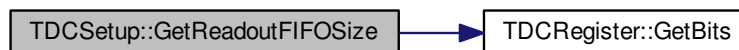
Extract the adjustment of the RC delay line.

Here is the call graph for this function:



**7.28.4.19** `int TDCSetup::GetReadoutFIFOSize ( ) const [inline]`

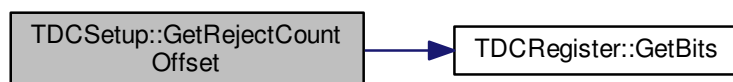
Here is the call graph for this function:



**7.28.4.20** `uint16_t TDCSetup::GetRejectCountOffset ( ) const [inline]`

Extract the offset in reject counter.

Here is the call graph for this function:



**7.28.4.21** `bool TDCSetup::GetRejectFIFOFull ( ) const [inline]`

Are hits rejected when readout FIFO is full?

Extract whether or not hits are rejected once FIFO is full.



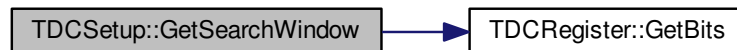
Here is the call graph for this function:



**7.28.4.22** `uint16_t TDCSetup::GetSearchWindow ( ) const [inline]`

Extract the search window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)

Here is the call graph for this function:



**7.28.4.23** `bool TDCSetup::GetSetupParity ( ) const [inline]`

Extract the parity of setup data (should be an even parity)

Here is the call graph for this function:



7.28.4.24 `uint16_t TDCSetup::GetTDCId ( ) const [inline]`

Here is the call graph for this function:



7.28.4.25 `bool TDCSetup::GetTestInvert ( ) const [inline]`

Here is the call graph for this function:



7.28.4.26 `bool TDCSetup::GetTestMode ( ) const [inline]`

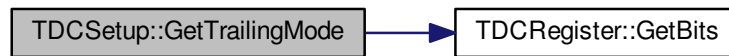
Here is the call graph for this function:



7.28.4.27 `bool TDCSetup::GetTrailingMode ( ) const [inline]`

Extract the status for the detection of trailing edges.

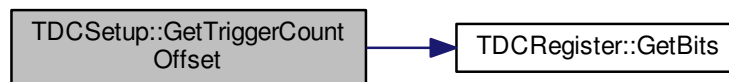
Here is the call graph for this function:



#### 7.28.4.28 `uint16_t TDCSetup::GetTriggerCountOffset ( ) const [inline]`

Extract trigger time tag count offset.

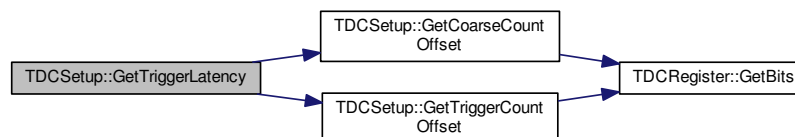
Here is the call graph for this function:



#### 7.28.4.29 `uint16_t TDCSetup::GetTriggerLatency ( ) const [inline]`

Effective trigger latency in number of clock cycles (when no counter roll-over is used)

Here is the call graph for this function:



#### 7.28.4.30 `bool TDCSetup::GetTriggerMatchingMode ( ) const [inline]`

Extract the enable status of trigger matching mode.

Here is the call graph for this function:



#### 7.28.4.31 `uint8_t TDCSetup::GetVernierOffset ( ) const [inline]`

Extract the offset in vernier decoding.

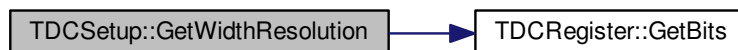
Here is the call graph for this function:



#### 7.28.4.32 `WidthResolution TDCSetup::GetWidthResolution ( ) const [inline]`

Extract the pulse width resolution when paired measurements are performed.

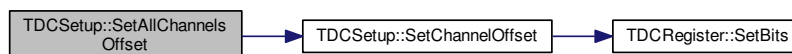
Here is the call graph for this function:



#### 7.28.4.33 `void TDCSetup::SetAllChannelsOffset ( uint16_t offset ) [inline]`

Set the time offset for all channels.

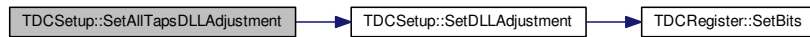
Here is the call graph for this function:



7.28.4.34 `void TDCSetup::SetAllTapsDLLAdjustment ( uint8_t adj ) [inline]`

Extract the adjustment of DLL taps.

Here is the call graph for this function:



7.28.4.35 `void TDCSetup::SetBypassInputs ( const bool sbi=true ) [inline], [private]`

Select serial in and token in from bypass inputs.

Here is the call graph for this function:



7.28.4.36 `void TDCSetup::SetChannelOffset ( int channel, uint16_t offset ) [inline]`

Set the time offset for one single channel.

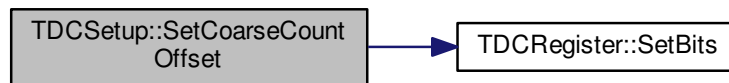
Here is the call graph for this function:



7.28.4.37 `void TDCSetup::SetCoarseCountOffset ( uint16_t cco ) [inline]`

Set offset for the coarse time counter.

Here is the call graph for this function:



**7.28.4.38** `void TDCSetup::SetConstantValues ( )` `[virtual]`

Ensure that the critical constant values are properly set in the setup word.

Implements [TDCRegister](#).

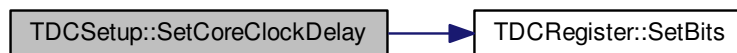
**7.28.4.39** `void TDCSetup::SetCoreClockDelay ( const bool delay_clock, const uint8_t delay )` `[inline], [private]`

Delay of internal core clock.

Parameters

in	<i>delay_clock</i>	Use of direct clock (0) or delayed clock (1)
in	<i>delay</i>	Delay in steps of (typically) 0.13 ns

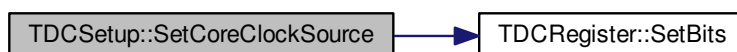
Here is the call graph for this function:



**7.28.4.40** `void TDCSetup::SetCoreClockSource ( const CoreClockSource ccs )` `[inline], [private]`

Selection of clock source for internal logic.

Here is the call graph for this function:



7.28.4.41 `void TDCSetup::SetDeadTime ( const DeadTime dt ) [inline]`

Channel dead time between hits.

Here is the call graph for this function:



7.28.4.42 `void TDCSetup::SetDLLAdjustment ( int tap, uint8_t adj ) [inline]`

Set the DLL taps adjustments with a resolution of  $\sim 10$  ps.

Here is the call graph for this function:



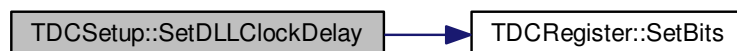
7.28.4.43 `void TDCSetup::SetDLLClockDelay ( const bool delay_clock, const uint8_t delay ) [inline],[private]`

Delay of internal DLL clock.

Parameters

in	<i>delay_clock</i>	Use of direct clock (0) or delayed clock (1)
in	<i>delay</i>	Delay in steps of (typically) 0.13 ns

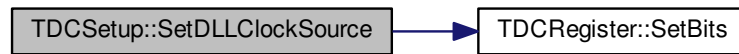
Here is the call graph for this function:



7.28.4.44 `void TDCSetup::SetDLLClockSource ( const DLLClockSource dcs ) [inline],[private]`

Selection of clock source for DLL.

Here is the call graph for this function:



**7.28.4.45** `void TDCSetup::SetDLLControl ( const uint8_t dc )` `[inline], [private]`

Control of DLL (DLL charge pump levels)

Here is the call graph for this function:



**7.28.4.46** `void TDCSetup::SetDLLMode ( const DLLSpeedMode dsm )` `[inline], [private]`

Selection of DLL speed mode.

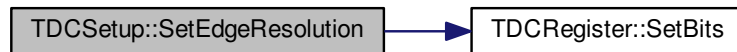
Here is the call graph for this function:





7.28.4.47 `void TDCSetup::SetEdgeResolution ( const EdgeResolution r ) [inline]`

Here is the call graph for this function:



7.28.4.48 `void TDCSetup::SetEdgesPairing ( const bool pair = true ) [inline]`

Enable the pairing of leading and trailing edges (overrides individual enable of leading/trailing edges)

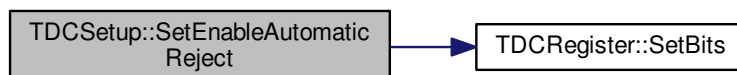
Here is the call graph for this function:



7.28.4.49 `void TDCSetup::SetEnableAutomaticReject ( const bool ear = true ) [inline],[private]`

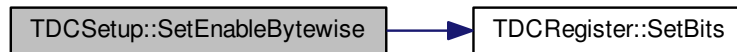
Enable of automatic rejection (should always be enabled if trigger matching mode!)

Here is the call graph for this function:



7.28.4.50 `void TDCSetup::SetEnableBytewise ( const bool seb = true ) [inline],[private]`

Here is the call graph for this function:



7.28.4.51 `void TDCSetup::SetEnableDirectBunchReset ( const bool edbr = true ) [inline],[private]`

Enable of direct bunch reset input pin (1), otherwise taken from encoded control.

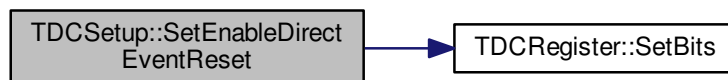
Here is the call graph for this function:



7.28.4.52 `void TDCSetup::SetEnableDirectEventReset ( const bool eder = true ) [inline],[private]`

Enable of direct event reset input pin (1), otherwise taken from encoded control.

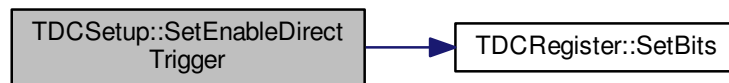
Here is the call graph for this function:



7.28.4.53 `void TDCSetup::SetEnableDirectTrigger ( const bool edt = true ) [inline],[private]`

Enable of direct trigger input pin.

Here is the call graph for this function:



**7.28.4.54** `void TDCSetup::SetEnableError ( const uint16_t & err ) [inline]`

Enable internal error types for generation of global error signals.

Here is the call graph for this function:



**7.28.4.55** `void TDCSetup::SetEnableErrorBypass ( const bool eb ) [inline]`

Bypass TDC chip if global error signal is set.

Here is the call graph for this function:



**7.28.4.56** `void TDCSetup::SetEnableErrorMark ( const bool em ) [inline]`

Mark events with error if global error signal is set.

Here is the call graph for this function:



**7.28.4.57** `void TDCSetup::SetEnableGlobalHeader ( const bool egh = true )` `[inline], [private]`

Enable of global headers in read-out (only valid for master TDC)

Here is the call graph for this function:



**7.28.4.58** `void TDCSetup::SetEnableGlobalTrailer ( const bool egt = true )` `[inline], [private]`

Enable of global trailers in read-out (only valid for master TDC)

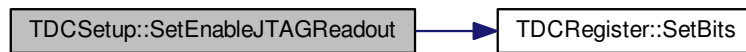
Here is the call graph for this function:



**7.28.4.59** `void TDCSetup::SetEnableJTAGReadout ( const bool jr )` `[inline]`

Enable of read-out via JTAG.

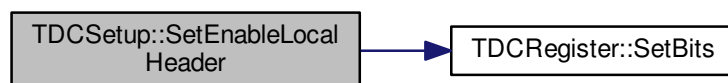
Here is the call graph for this function:



7.28.4.60 `void TDCSetup::SetEnableLocalHeader ( const bool elh = true )` `[inline], [private]`

Enable of local headers in read-out.

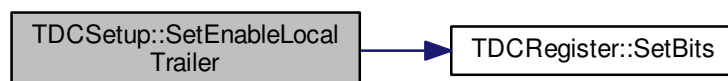
Here is the call graph for this function:



7.28.4.61 `void TDCSetup::SetEnableLocalTrailer ( const bool elt = true )` `[inline], [private]`

Enable of local trailers in read-out.

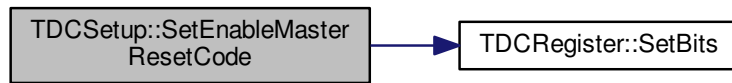
Here is the call graph for this function:



7.28.4.62 `void TDCSetup::SetEnableMasterResetCode ( const bool emrc = true )` `[inline], [private]`

Enable master reset code on encoded\_control.

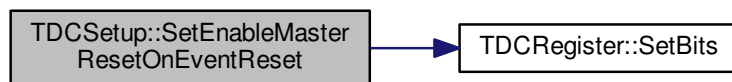
Here is the call graph for this function:



**7.28.4.63** `void TDCSetup::SetEnableMasterResetOnEventReset ( const bool emroer = true )` `[inline], [private]`

Enable master reset of whole TDC on event reset.

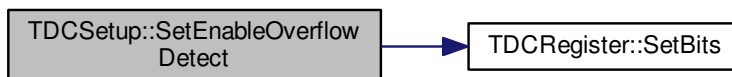
Here is the call graph for this function:



**7.28.4.64** `void TDCSetup::SetEnableOverflowDetect ( const bool eod = true )` `[inline], [private]`

Enable overflow detection of L1 buffers (should always be enabled!)

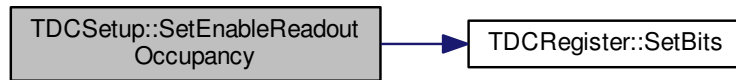
Here is the call graph for this function:



**7.28.4.65** `void TDCSetup::SetEnableReadoutOccupancy ( const bool ro = true )` `[inline]`

Enable the readout of buffer occupancies for each event (for debugging purposes)

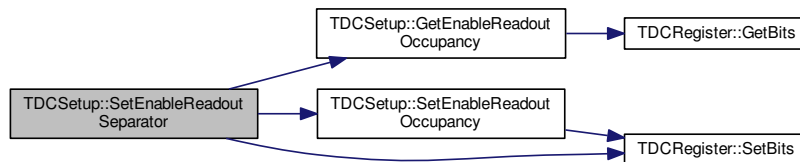
Here is the call graph for this function:



**7.28.4.66** `void TDCSetup::SetEnableReadoutSeparator ( const bool ro = true ) [inline]`

Enable the readout of separators for each event (for debugging purposes, valid if readout of occupancies is enabled)

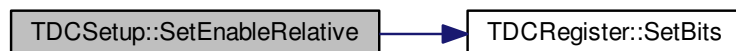
Here is the call graph for this function:



**7.28.4.67** `void TDCSetup::SetEnableRelative ( const bool er = true ) [inline], [private]`

Enable read-out of relative time to trigger time tag. Only valid when using trigger matching mode.

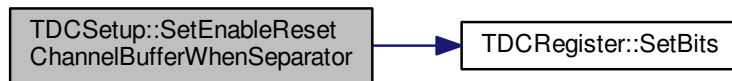
Here is the call graph for this function:



**7.28.4.68** `void TDCSetup::SetEnableResetChannelBufferWhenSeparator ( const bool ercbws = true ) [inline], [private]`

Enable reset channel buffers when separator.

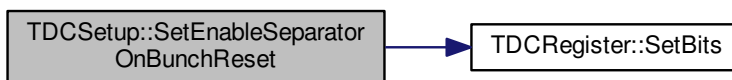
Here is the call graph for this function:



**7.28.4.69** `void TDCSetup::SetEnableSeparatorOnBunchReset ( const bool esobr = true )` `[inline],[private]`

Enable generation of separator on bunch reset.

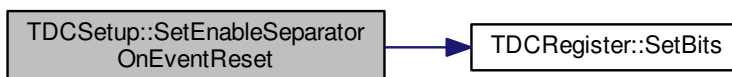
Here is the call graph for this function:



**7.28.4.70** `void TDCSetup::SetEnableSeparatorOnEventReset ( const bool esoer = true )` `[inline],[private]`

Enable generation of separator on event reset.

Here is the call graph for this function:

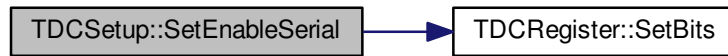


**7.28.4.71** `void TDCSetup::SetEnableSerial ( const bool es )` `[inline]`

Enable of serial read-out (otherwise parallel read-out)



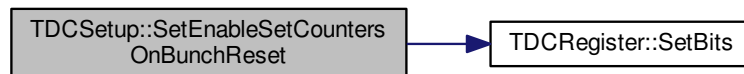
Here is the call graph for this function:



**7.28.4.72** `void TDCSetup::SetEnableSetCountersOnBunchReset ( const bool escobr = true ) [inline],[private]`

Enable all counters to be set on bunch count reset.

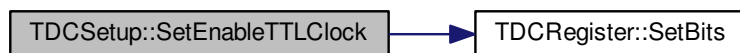
Here is the call graph for this function:



**7.28.4.73** `void TDCSetup::SetEnableTTLClock ( const bool tc = true ) [inline],[private]`

Enable LV TTL inputs on: clk, aux\_clock, otherwise uses LVDS input levels.

Here is the call graph for this function:



**7.28.4.74** `void TDCSetup::SetEnableTTLControl ( const bool tc = true ) [inline],[private]`

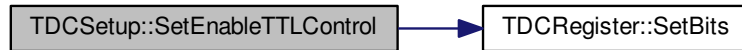
Enable LV TTL inputs on control registers.

Enable LV TTL input on:

- trigger,
- bunch\_reset,
- event\_reset,

- encoded\_control, otherwise uses LVDS input levels.

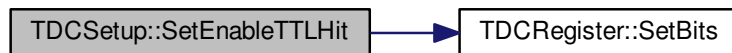
Here is the call graph for this function:



**7.28.4.75** `void TDCSetup::SetEnableTTLHit ( const bool th = true ) [inline], [private]`

Enable LV TTL input on hit[31:0], otherwise uses LVDS input levels.

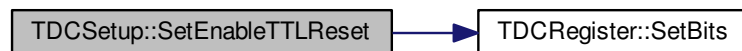
Here is the call graph for this function:



**7.28.4.76** `void TDCSetup::SetEnableTTLReset ( const bool tr = true ) [inline], [private]`

Enable LV TTL input on reset, otherwise uses LVDS input levels.

Here is the call graph for this function:



**7.28.4.77** `void TDCSetup::SetEnableTTLSerial ( const bool ts = true ) [inline], [private]`

Enable LV TTL inputs on serial registers, and disable their drivers.

Enable LV TTL input on:

- serial\_in,
- serial\_bypass\_in,

- token\_in,
- token\_bypass\_in, otherwise uses LVDS input levels. Disable LVDS drivers on:
- serial\_out,
- strobe\_out,
- token\_out.

Here is the call graph for this function:



**7.28.4.78** `void TDCSetup::SetEventCountOffset ( uint16_t eco ) [inline]`

Set offset for the event counter.

Here is the call graph for this function:



**7.28.4.79** `void TDCSetup::SetIOClockDelay ( const bool delay_clock, const uint8_t delay ) [inline],[private]`

Delay of internal I/O clock.

Parameters

in	<i>delay_clock</i>	Use of direct clock (0) or delayed clock (1)
in	<i>delay</i>	Delay in steps of (typically) 0.13 ns

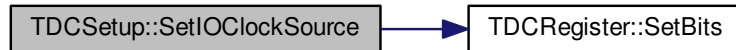
Here is the call graph for this function:



7.28.4.80 `void TDCSetup::SetIOClockSource ( const IOClockSource ics ) [inline],[private]`

Selection of clock source for I/O signals.

Here is the call graph for this function:



7.28.4.81 `void TDCSetup::SetKeepToken ( const bool kt = true ) [inline],[private]`

Keep token until end of event or no more data, otherwise pass token after each word read. Must be enabled when using trigger matching.

Here is the call graph for this function:



7.28.4.82 `void TDCSetup::SetLeadingMode ( const bool lead = true ) [inline]`

Enable the detection of leading edges.

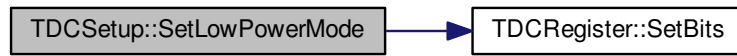
Here is the call graph for this function:



7.28.4.83 `void TDCSetup::SetLowPowerMode ( const bool lpm = true ) [inline],[private]`

Low power mode of channel buffers.

Here is the call graph for this function:



7.28.4.84 `void TDCSetup::SetMaster ( const bool m = true ) [inline], [private]`

Here is the call graph for this function:



7.28.4.85 `void TDCSetup::SetMatchWindow ( uint16_t mw ) [inline]`

Set the matching window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)

Here is the call graph for this function:

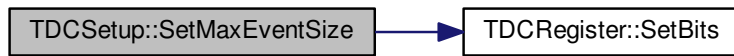


7.28.4.86 `void TDCSetup::SetMaxEventSize ( int sz = -1 ) [inline]`

Set the maximum number of hits per event.

Set the maximum number of hits that can be recorded for each event. It is always rounded to the next power of 2 (in the range 0-128), and if lower than 0 or bigger than 128 then set to unlimited.

Here is the call graph for this function:



**7.28.4.87** `void TDCSetup::SetModeRC ( const bool mr = true ) [inline], [private]`

Enable of RR delay lines mode (in very high resolution mode) ; only for channels 0-4-8-12-16-20-24-28 active.

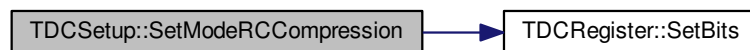
Here is the call graph for this function:



**7.28.4.88** `void TDCSetup::SetModeRCCompression ( const bool mrc = true ) [inline], [private]`

Perform RC interpolation on-chip (only valid in very high resolution mode)

Here is the call graph for this function:



**7.28.4.89** `void TDCSetup::SetPLLControl ( const uint8_t charge_pump_current = 0x4, const bool power_down_mode = false, const bool enable_test_outputs = false, const bool invert_connection_to_status = false ) [inline], [private]`

Control of PLL.

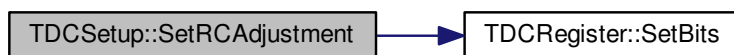
Here is the call graph for this function:



**7.28.4.90** void TDCSetup::SetRCAdjustment ( int *tap*, uint8\_t *adj* ) [inline]

Set the adjustment of the RC delay line.

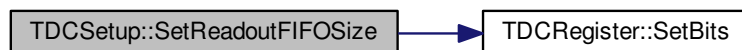
Here is the call graph for this function:



**7.28.4.91** void TDCSetup::SetReadoutFIFOSize ( int *rfs* ) [inline]

Effective size of readout FIFO.

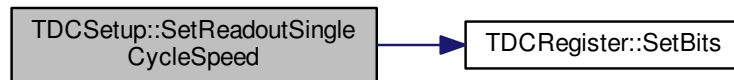
Here is the call graph for this function:



**7.28.4.92** void TDCSetup::SetReadoutSingleCycleSpeed ( const ReadoutSingleCycleSpeed *rscs* = RSC\_40Mbits\_s ) [inline], [private]

Serial transmission speed in single cycle mode.

Here is the call graph for this function:



**7.28.4.93** `void TDCSetup::SetReadoutSpeedSelect ( const ReadoutSpeed rss = RO_Fixed ) [inline], [private]`

Selection of serial read-out speed.

#### Parameters

in	rss	
		<ul style="list-style-type: none"> <li>• 0: Selection of serial read-out speed (as defined by setup[19:17], <i>SetReadoutSingleCycleSpeed</i>)</li> <li>• 1: 80 Mbits/s (PLL lock required)</li> </ul>

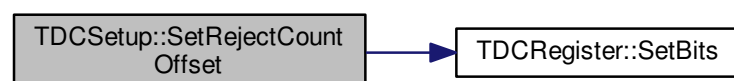
Here is the call graph for this function:



**7.28.4.94** `void TDCSetup::SetRejectCountOffset ( uint16_t rco ) [inline]`

Set the offset in reject counter (defines reject latency together with coarse count offset)

Here is the call graph for this function:



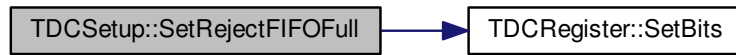


7.28.4.95 `void TDCSetup::SetRejectFIFOFull ( const bool rej = true ) [inline]`

Reject hits when readout FIFO full.

Set whether or not hits are rejected once FIFO is full.

Here is the call graph for this function:



7.28.4.96 `void TDCSetup::SetRollOver ( const uint16_t ro = 0xFFFF ) [inline],[private]`

Counter roll over value, defining maximal count value from where counters will be reset to 0.

Here is the call graph for this function:



7.28.4.97 `void TDCSetup::SetSearchWindow ( uint16_t sw ) [inline]`

Set the search window (in multiples of clock cycles: 0=25 ns, 1=50 ns, ...)

Here is the call graph for this function:



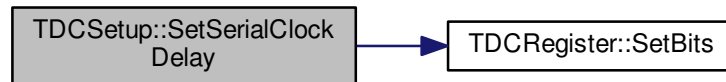
7.28.4.98 `void TDCSetup::SetSerialClockDelay ( const bool delay_clock, const uint8_t delay ) [inline],[private]`

Delay of internal serial clock.

## Parameters

in	<i>delay_clock</i>	Use of direct clock (0) or delayed clock (1)
in	<i>delay</i>	Delay in steps of (typically) 0.13 ns

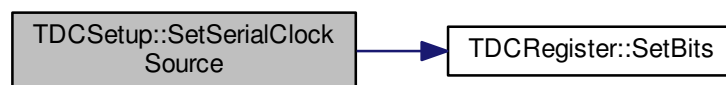
Here is the call graph for this function:



**7.28.4.99** `void TDCSetup::SetSerialClockSource ( const SerialClockSource scs ) [inline], [private]`

Selection of source for serial clock.

Here is the call graph for this function:



**7.28.4.100** `void TDCSetup::SetSerialDelay ( const uint8_t sd = 0x0 ) [inline], [private]`

Programmable delay of serial input, in time unit  $\sim 1$  ns.

Here is the call graph for this function:



**7.28.4.101** `void TDCSetup::SetSetupParity ( const bool sp = true ) [inline]`

Set the parity of setup data (should be an even parity)

Here is the call graph for this function:



**7.28.4.102** `void TDCSetup::SetStrobeSelect ( const SerialStrobeType ss = SS_NoStrobe ) [inline],  
[private]`

Here is the call graph for this function:



**7.28.4.103** `void TDCSetup::SetTDCId ( const uint8_t id = 0x0 ) [inline]`

Here is the call graph for this function:



7.28.4.104 void TDCSetup::SetTest ( const bool *test* = true ) [inline],[private]

Here is the call graph for this function:



7.28.4.105 void TDCSetup::SetTestInvert ( const bool *ti* = true ) [inline]

Automatic inversion of test pattern. Only used during production testing.

Here is the call graph for this function:



7.28.4.106 void TDCSetup::SetTestMode ( const bool *tm* = true ) [inline]

Test mode where hit data are taken from coretest. Only used during production testing.

Here is the call graph for this function:



7.28.4.107 void TDCSetup::SetTokenDelay ( const uint8\_t *td* = 0x0 ) [inline],[private]

Programmable delay of token input, in time unit  $\sim 1$  ns.

Here is the call graph for this function:



**7.28.4.108** `void TDCSetup::SetTrailingMode ( const bool trail = true ) [inline]`

Enable/disable the detection of trailing edges.

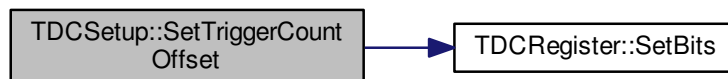
Here is the call graph for this function:



**7.28.4.109** `void TDCSetup::SetTriggerCountOffset ( uint16_t tco ) [inline]`

Set offset for the trigger time tag counter to set effective trigger latency.

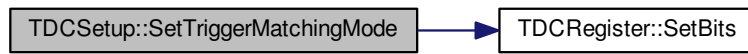
Here is the call graph for this function:



**7.28.4.110** `void TDCSetup::SetTriggerMatchingMode ( const bool trig = true ) [inline]`

Set the enable status of trigger matching mode.

Here is the call graph for this function:



7.28.4.111 `void TDCSetup::SetVernierOffset ( const uint8_t vo ) [inline]`

Set the offset in vernier decoding.

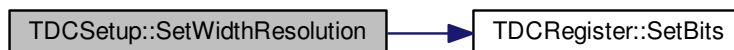
Here is the call graph for this function:



7.28.4.112 `void TDCSetup::SetWidthResolution ( const WidthResolution r ) [inline]`

Set the pulse width resolution when paired measurements are performed.

Here is the call graph for this function:



## 7.28.5 Field Documentation

7.28.5.1 `const bit TDCSetup::kCoarseCountOffset = 447 [static], [private]`

7.28.5.2 `const bit TDCSetup::kCoreClockDelay = 610 [static], [private]`

7.28.5.3 `const bit TDCSetup::kCoreClockSource = 622 [static], [private]`

7.28.5.4 `const bit TDCSetup::kDeadTime = 584 [static], [private]`

7.28.5.5 `const bit TDCSetup::kDLLClockDelay = 614 [static], [private]`

- 7.28.5.6 `const bit TDCSetup::kDLLClockSource = 624` [static], [private]
- 7.28.5.7 `const bit TDCSetup::kDLLControl = 580` [static], [private]
- 7.28.5.8 `const bit TDCSetup::kDLLMode = 592` [static], [private]
- 7.28.5.9 `const bit TDCSetup::kDLLTapAdjust0 = 459` [static], [private]
- 7.28.5.10 `const bit TDCSetup::kEnableAutomaticReject = 125` [static], [private]
- 7.28.5.11 `const bit TDCSetup::kEnableBytewise = 37` [static], [private]
- 7.28.5.12 `const bit TDCSetup::kEnableDirectBunchReset = 157` [static], [private]
- 7.28.5.13 `const bit TDCSetup::kEnableDirectEventReset = 156` [static], [private]
- 7.28.5.14 `const bit TDCSetup::kEnableDirectTrigger = 158` [static], [private]
- 7.28.5.15 `const bit TDCSetup::kEnableError = 6` [static], [private]
- 7.28.5.16 `const bit TDCSetup::kEnableErrorBypass = 5` [static], [private]
- 7.28.5.17 `const bit TDCSetup::kEnableErrorMark = 4` [static], [private]
- 7.28.5.18 `const bit TDCSetup::kEnableGlobalHeader = 34` [static], [private]
- 7.28.5.19 `const bit TDCSetup::kEnableGlobalTrailer = 33` [static], [private]
- 7.28.5.20 `const bit TDCSetup::kEnableJTAGReadout = 39` [static], [private]
- 7.28.5.21 `const bit TDCSetup::kEnableLocalHeader = 32` [static], [private]
- 7.28.5.22 `const bit TDCSetup::kEnableLocalTrailer = 31` [static], [private]
- 7.28.5.23 `const bit TDCSetup::kEnableMasterResetCode = 151` [static], [private]
- 7.28.5.24 `const bit TDCSetup::kEnableMasterResetOnEventReset = 152` [static], [private]
- 7.28.5.25 `const bit TDCSetup::kEnableMatching = 639` [static], [private]
- 7.28.5.26 `const bit TDCSetup::kEnableOverflowDetect = 123` [static], [private]
- 7.28.5.27 `const bit TDCSetup::kEnablePair = 640` [static], [private]
- 7.28.5.28 `const bit TDCSetup::kEnableReadoutOccupancy = 121` [static], [private]
- 7.28.5.29 `const bit TDCSetup::kEnableReadoutSeparator = 122` [static], [private]
- 7.28.5.30 `const bit TDCSetup::kEnableRelative = 124` [static], [private]
- 7.28.5.31 `const bit TDCSetup::kEnableResetChannelBufferWhenSeparator = 153` [static], [private]
- 7.28.5.32 `const bit TDCSetup::kEnableSeparatorOnBunchReset = 155` [static], [private]
- 7.28.5.33 `const bit TDCSetup::kEnableSeparatorOnEventReset = 154` [static], [private]

7.28.5.34 `const bit TDCSetup::kEnableSerial = 38` [static], [private]

7.28.5.35 `const bit TDCSetup::kEnableSetCountersOnBunchReset = 150` [static], [private]

7.28.5.36 `const bit TDCSetup::kEnableTTLClock = 644` [static], [private]

7.28.5.37 `const bit TDCSetup::kEnableTTLControl = 642` [static], [private]

7.28.5.38 `const bit TDCSetup::kEnableTTLHit = 645` [static], [private]

7.28.5.39 `const bit TDCSetup::kEnableTTLReset = 643` [static], [private]

7.28.5.40 `const bit TDCSetup::kEnableTTLSerial = 641` [static], [private]

7.28.5.41 `const bit TDCSetup::kEventCountOffset = 126` [static], [private]

7.28.5.42 `const bit TDCSetup::kIOClockDelay = 606` [static], [private]

7.28.5.43 `const bit TDCSetup::kIOClockSource = 620` [static], [private]

7.28.5.44 `const bit TDCSetup::kKeepToken = 35` [static], [private]

7.28.5.45 `const bit TDCSetup::kLeading = 589` [static], [private]

7.28.5.46 `const bit TDCSetup::kLeadingResolution = 84` [static], [private]

7.28.5.47 `const bit TDCSetup::kLowPowerMode = 570` [static], [private]

7.28.5.48 `const bit TDCSetup::kMaster = 36` [static], [private]

7.28.5.49 `const bit TDCSetup::kMatchWindow = 72` [static], [private]

7.28.5.50 `const bit TDCSetup::kMaxEventSize = 116` [static], [private]

7.28.5.51 `const bit TDCSetup::kModeRC = 591` [static], [private]

7.28.5.52 `const bit TDCSetup::kModeRCCompression = 590` [static], [private]

7.28.5.53 `const bit TDCSetup::kOffset0 = 438` [static], [private]

7.28.5.54 `const bit TDCSetup::kPLLControl = 594` [static], [private]

7.28.5.55 `const bit TDCSetup::kRCAdjust0 = 555` [static], [private]

7.28.5.56 `const bit TDCSetup::kReadoutFIFOSize = 45` [static], [private]

7.28.5.57 `const bit TDCSetup::kReadoutSingleCycleSpeed = 17` [static], [private]

7.28.5.58 `const bit TDCSetup::kReadoutSpeedSelect = 26` [static], [private]

7.28.5.59 `const bit TDCSetup::kRejectCountOffset = 48` [static], [private]

7.28.5.60 `const bit TDCSetup::kRejectFIFOFull = 120` [static], [private]

7.28.5.61 `const bit TDCSetup::kRollOver = 627` [static], [private]



- 7.28.5.62 `const bit TDCSetup::kSearchWindow = 60` [static], [private]
- 7.28.5.63 `const bit TDCSetup::kSelectBypassInputs = 44` [static], [private]
- 7.28.5.64 `const bit TDCSetup::kSerialClockDelay = 602` [static], [private]
- 7.28.5.65 `const bit TDCSetup::kSerialClockSource = 618` [static], [private]
- 7.28.5.66 `const bit TDCSetup::kSerialDelay = 20` [static], [private]
- 7.28.5.67 `const bit TDCSetup::kSetupParity = 646` [static], [private]
- 7.28.5.68 `const bit TDCSetup::kStrobeSelect = 24` [static], [private]
- 7.28.5.69 `const bit TDCSetup::kTDCId = 40` [static], [private]
- 7.28.5.70 `const bit TDCSetup::kTestInvert = 586` [static], [private]
- 7.28.5.71 `const bit TDCSetup::kTestMode = 587` [static], [private]
- 7.28.5.72 `const bit TDCSetup::kTestSelect = 0` [static], [private]
- 7.28.5.73 `const bit TDCSetup::kTokenDelay = 27` [static], [private]
- 7.28.5.74 `const bit TDCSetup::kTrailing = 588` [static], [private]
- 7.28.5.75 `const bit TDCSetup::kTriggerCountOffset = 138` [static], [private]
- 7.28.5.76 `const bit TDCSetup::kVernierOffset = 575` [static], [private]
- 7.28.5.77 `const bit TDCSetup::kWidthSelect = 571` [static], [private]

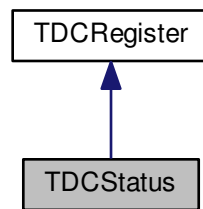
The documentation for this class was generated from the following files:

- `daq/include/TDCSetup.h`
- `daq/src/TDCSetup.cpp`

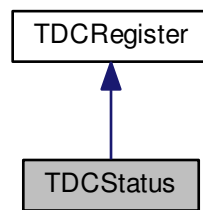
## 7.29 TDCStatus Class Reference

```
#include <TDCStatus.h>
```

Inheritance diagram for TDCStatus:



Collaboration diagram for TDCStatus:



## Public Member Functions

- [TDCStatus](#) ()
- [TDCStatus](#) (const [TDCStatus](#) &s)
- [TDCStatus](#) (const std::vector< uint8\_t > &words)
- void [SetConstantValues](#) ()
- uint16\_t [Error](#) () const
- bool [HaveToken](#) () const
- uint8\_t [FIFOOccupancy](#) () const
- bool [FIFOFull](#) () const
- bool [FIFOEmpty](#) () const
- uint32\_t [L1Occupancy](#) () const
- uint8\_t [TriggerFIFOOccupancy](#) () const
- bool [TriggerFIFOFull](#) () const
- bool [TriggerFIFOEmpty](#) () const
- bool [DLLLock](#) () const
- void [Dump](#) (int verb=1, std::ostream &os=std::cout) const

## Static Private Attributes

- static const [bit kError](#) = 0

- static const `bit kHaveToken` = 11
- static const `bit kReadoutFIFOOccupancy` = 12
- static const `bit kReadoutFIFOFull` = 20
- static const `bit kReadoutFIFOEmpty` = 21
- static const `bit kL1Occupancy` = 22
- static const `bit kTriggerFIFOOccupancy` = 54
- static const `bit kTriggerFIFOFull` = 58
- static const `bit kTriggerFIFOEmpty` = 59
- static const `bit kDLLLock` = 60

## Additional Inherited Members

### 7.29.1 Detailed Description

#### Author

Laurent Forthomme [laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

#### Date

27 Apr 2015

### 7.29.2 Constructor & Destructor Documentation

#### 7.29.2.1 `TDCStatus::TDCStatus ( )` `[inline]`

Here is the call graph for this function:



#### 7.29.2.2 `TDCStatus::TDCStatus ( const TDCStatus & s )` `[inline]`

Here is the call graph for this function:



7.29.2.3 `TDCStatus::TDCStatus ( const std::vector< uint8_t > & words ) [inline]`

### 7.29.3 Member Function Documentation

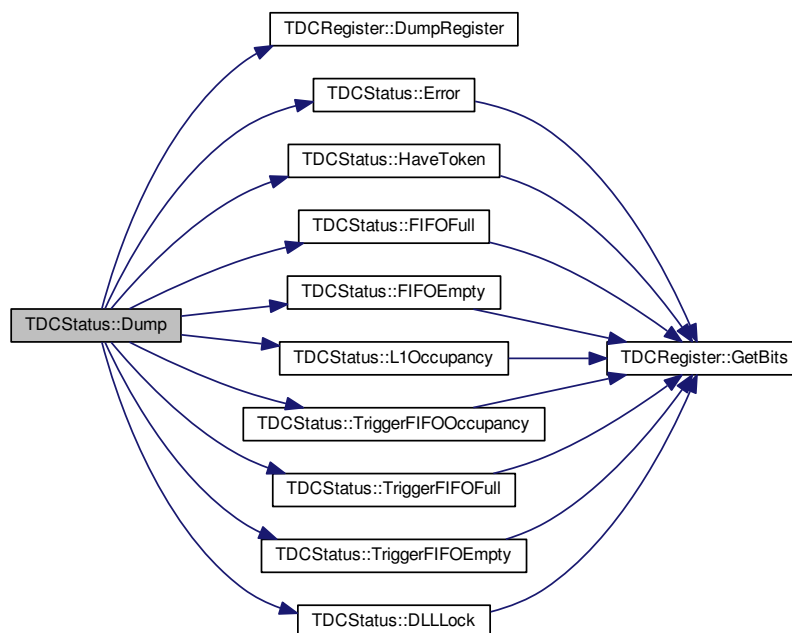
7.29.3.1 `bool TDCStatus::DLLLock ( ) const [inline]`

Here is the call graph for this function:



7.29.3.2 `void TDCStatus::Dump ( int verb = 1, std::ostream & os = std::cout ) const [inline]`

Here is the call graph for this function:



### 7.29.3.3 `uint16_t TDCStatus::Error ( ) const [inline]`

Here is the call graph for this function:



### 7.29.3.4 `bool TDCStatus::FIFOEmpty ( ) const [inline]`

Here is the call graph for this function:



### 7.29.3.5 `bool TDCStatus::FIFOFull ( ) const [inline]`

Here is the call graph for this function:



#### 7.29.3.6 `uint8_t TDCStatus::FIFOOccupancy ( ) const [inline]`

Here is the call graph for this function:



#### 7.29.3.7 `bool TDCStatus::HaveToken ( ) const [inline]`

Here is the call graph for this function:



#### 7.29.3.8 `uint32_t TDCStatus::L1Occupancy ( ) const [inline]`

Here is the call graph for this function:



#### 7.29.3.9 `void TDCStatus::SetConstantValues ( ) [inline], [virtual]`

Ensure that the critical constant values are properly set in the register word

Implements [TDCRegister](#).

7.29.3.10 `bool TDCStatus::TriggerFIFOEmpty ( ) const [inline]`

Here is the call graph for this function:



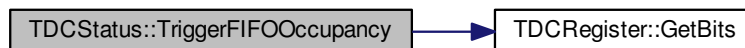
7.29.3.11 `bool TDCStatus::TriggerFIFOFull ( ) const [inline]`

Here is the call graph for this function:



7.29.3.12 `uint8_t TDCStatus::TriggerFIFOOccupancy ( ) const [inline]`

Here is the call graph for this function:



## 7.29.4 Field Documentation

7.29.4.1 `const bit TDCStatus::kDLLLock = 60 [static], [private]`

7.29.4.2 `const bit TDCStatus::kError = 0 [static], [private]`

7.29.4.3 `const bit TDCStatus::kHaveToken = 11 [static], [private]`

7.29.4.4 `const bit TDCStatus::kL1Occupancy = 22 [static], [private]`

7.29.4.5 `const bit TDCStatus::kReadoutFIFOEmpty = 21 [static], [private]`

7.29.4.6 `const bit TDCStatus::kReadoutFIFOFull = 20` `[static], [private]`

7.29.4.7 `const bit TDCStatus::kReadoutFIFOOccupancy = 12` `[static], [private]`

7.29.4.8 `const bit TDCStatus::kTriggerFIFOEmpty = 59` `[static], [private]`

7.29.4.9 `const bit TDCStatus::kTriggerFIFOFull = 58` `[static], [private]`

7.29.4.10 `const bit TDCStatus::kTriggerFIFOOccupancy = 54` `[static], [private]`

The documentation for this class was generated from the following file:

- `daq/include/TDCStatus.h`

## 7.30 DAQ::QuickUSBHandler::Version Struct Reference

```
#include <QuickUSBHandler.h>
```

### Data Fields

- QWORD [MajorVersion](#)
- QWORD [MinorVersion](#)
- QWORD [BuildVersion](#)

### 7.30.1 Field Documentation

7.30.1.1 QWORD `DAQ::QuickUSBHandler::Version::BuildVersion`

7.30.1.2 QWORD `DAQ::QuickUSBHandler::Version::MajorVersion`

7.30.1.3 QWORD `DAQ::QuickUSBHandler::Version::MinorVersion`

The documentation for this struct was generated from the following file:

- `daq/include/QuickUSBHandler.h`



# Index

- ~Client
  - Client, [17](#)
- ~DQMProcess
  - DQM::DQMProcess, [24](#)
- ~FPGAHandler
  - DAQ::FPGAHandler, [32](#)
- ~FileReader
  - FileReader, [29](#)
- ~GastofCanvas
  - DQM::GastofCanvas, [38](#)
- ~Logger
  - Logger, [41](#)
- ~Message
  - Message, [43](#)
- ~Messenger
  - Messenger, [46](#)
- ~OnlineDBHandler
  - OnlineDBHandler, [53](#)
- ~PPSCanvas
  - DQM::PPSCanvas, [56](#)
- ~QuarticCanvas
  - DQM::QuarticCanvas, [60](#)
- ~QuickUSBHandler
  - DAQ::QuickUSBHandler, [64](#)
- ~Socket
  - Socket, [67](#)
- ~SocketMessage
  - SocketMessage, [77](#)
- ~TDC
  - DAQ::TDC, [81](#)
- ~TDCErrorFlag
  - TDCErrorFlag, [95](#)
- ~TDCEvent
  - TDCEvent, [97](#)
- ~TDCMeasurement
  - TDCMeasurement, [103](#)
- ~TDCRegister
  - TDCRegister, [107](#)
- AcceptConnections
  - Socket, [68](#)
- acq\_mode
  - file\_header\_t, [27](#)
- AcquisitionMode
  - HPTDC chip control, [11](#)
- Action
  - DQM::DQMProcess, [24](#)
- AddClient
  - Messenger, [47](#)
- Announce
  - Client, [18](#)
- Bind
  - Socket, [68](#)
- bit
  - TDCRegister, [107](#)
- Broadcast
  - Messenger, [47](#)
- Build
  - DQM::GastofCanvas, [39](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [61](#)
- BuildTables
  - OnlineDBHandler, [53](#)
- BuildVersion
  - DAQ::QuickUSBHandler::Version, [168](#)
- burst\_id
  - OnlineDBHandler::BurstInfo, [15](#)
- BurstInfos
  - OnlineDBHandler, [53](#)
- c1
  - DQM::GastofCanvas, [39](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [61](#)
- c2
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [62](#)
- CLIENT
  - Socket, [67](#)
- CONT\_STORAGE
  - HPTDC chip control, [11](#)
- ChannelSelectError
  - TDCSetup, [117](#)
- CheckFirmwareVersion
  - DAQ::TDC, [82](#)
- Clear
  - FileReader, [29](#)
  - TDCRegister, [108](#)
- Client, [15](#)
  - ~Client, [17](#)
  - Announce, [18](#)
  - Client, [17](#)
  - Connect, [18](#)
  - Disconnect, [19](#)
  - fClientId, [21](#)
  - flsConnected, [21](#)
  - fType, [21](#)
  - GetType, [19](#)

- ParseMessage, 19
- Receive, 19, 20
- Send, 20
- SendAndReceive, 21
- CloseFile
  - DAQ::FPGAHandler, 33
- CoarseError
  - TDCSetup, 117
- Configure
  - Socket, 68
- Connect
  - Client, 18
  - Messenger, 47
- contents
  - LogRedirector, 42
- ControlParityError
  - TDCSetup, 118
- Core\_aux\_clock
  - TDCSetup, 116
- Core\_clock\_40
  - TDCSetup, 116
- Core\_pll\_clock\_160
  - TDCSetup, 116
- Core\_pll\_clock\_80
  - TDCSetup, 116
- CoreClockSource
  - TDCSetup, 116
- Create
  - Socket, 68
- DAQ, 13
  - Socket, 67
- DAQ::FPGAHandler, 30
  - ~FPGAHandler, 32
  - CloseFile, 33
  - ErrorState, 33
  - fFilename, 36
  - flsFileOpen, 36
  - flsTDCInReadout, 36
  - fOutput, 36
  - FPGAHandler, 32
  - fTDC, 36
  - GetFilename, 33
  - GetTDC, 33
  - GetTDCControl, 33
  - GetTDCStatus, 33
  - GetType, 33
  - OpenFile, 33
  - RegisterTest, 34
  - RetrieveSetupWord, 34
  - SendSetupWord, 34
  - SetTDCSetup, 35
  - StartAcquisition, 35
  - Stop, 35
  - StopAcquisition, 35
- DAQ::QuickUSBHandler, 62
  - ~QuickUSBHandler, 64
  - fDevice, 65
  - fHandle, 65
  - flsStopping, 65
  - fStreamId, 65
  - Fetch, 64
  - GetDLLVersion, 64
  - GetDriverVersion, 64
  - GetFWVersion, 64
  - Init, 64
  - QuickUSBHandler, 64
  - Reset, 64
  - StartBulkTransfer, 65
  - StopBulkTransfer, 65
  - Write, 65
- DAQ::QuickUSBHandler::Version, 168
  - BuildVersion, 168
  - MajorVersion, 168
  - MinorVersion, 168
- DAQ::TDC, 80
  - ~TDC, 81
  - CheckFirmwareVersion, 82
  - DetectionMode, 81
  - fBS, 82
  - fControl, 82
  - fId, 82
  - fSetup, 82
  - fStatus, 82
  - fUSB, 82
  - FetchEvents, 82
  - GetSetupRegister, 82
  - OLEADING, 81
  - OTRILING, 81
  - PAIR, 81
  - ReadConfiguration, 82
  - ReadRegister, 82
  - ReadStatus, 82
  - SendConfiguration, 82
  - SetSetupRegister, 82
  - SoftReset, 82
  - TDC, 81
  - TRAILEAD, 81
  - WriteRegister, 82
- DETECTOR
  - Socket, 67
- DLL\_160MHz
  - TDCSetup, 117
- DLL\_320MHz
  - TDCSetup, 117
- DLL\_40MHz
  - TDCSetup, 117
- DLL\_Illegal
  - TDCSetup, 117
- DLL\_aux\_clock
  - TDCSetup, 117
- DLL\_clock\_40
  - TDCSetup, 117
- DLL\_pll\_clock\_160
  - TDCSetup, 117
- DLL\_pll\_clock\_320
  - TDCSetup, 117

DLL\_pll\_clock\_40  
     TDCSetup, 117  
 DLLClockSource  
     TDCSetup, 117  
 DLLLock  
     TDCStatus, 164  
 DLLSpeedMode  
     TDCSetup, 117  
 DQM, 13  
     Socket, 67  
 DQM::DQMProcess, 22  
     ~DQMProcess, 24  
     Action, 24  
     DQMProcess, 24  
     fAddressesCanProcess, 26  
     fDetectorType, 26  
     fOrder, 26  
     fRunNumber, 26  
     IsInRun, 24  
     NewPlot, 24  
     ParseMessage, 25  
     Run, 25  
     UpdatedPlot, 24  
 DQM::GastofCanvas, 36  
     ~GastofCanvas, 38  
     Build, 39  
     c1, 39  
     c2, 40  
     DrawGrid, 39  
     fBoardId, 40  
     fHeight, 40  
     fHist, 40  
     fLabel1, 40  
     fLabel2, 40  
     fLabel3, 40  
     fLabel4, 40  
     fLabelsDrawn, 40  
     fLegend, 40  
     fLegendNumEntries, 40  
     fLegendX, 40  
     fLegendY, 40  
     fRunDate, 40  
     fRunId, 40  
     fSpillId, 40  
     fUpperLabel, 40  
     fUpperLabelText, 40  
     fWidth, 40  
     FillChannel, 39  
     GastofCanvas, 38  
     GetCoordinates, 39  
     Grid, 39  
     Save, 39  
     SetRunInfo, 39  
     SetUpperLabel, 39  
 DQM::GastofCanvas::Coord, 21  
     x, 21  
     y, 22  
 DQM::PPSCanvas, 55  
     ~PPSCanvas, 56  
     Build, 57  
     c1, 57  
     c2, 57  
     DrawGrid, 57  
     fHeight, 57  
     fLabel1, 57  
     fLabel2, 57  
     fLabel3, 57  
     fLabelsDrawn, 57  
     fLegend, 58  
     fLegendNumEntries, 58  
     fLegendX, 58  
     fLegendY, 58  
     fRunDate, 58  
     fRunId, 58  
     fUpperLabel, 58  
     fUpperLabelText, 58  
     fWidth, 58  
     Grid, 57  
     PPSCanvas, 56  
     Save, 57  
     SetRunInfo, 57  
     SetUpperLabel, 57  
 DQM::QuarticCanvas, 58  
     ~QuarticCanvas, 60  
     Build, 61  
     c1, 61  
     c2, 62  
     DrawGrid, 61  
     fBoardId, 62  
     fHeight, 62  
     fHist, 62  
     fLabel1, 62  
     fLabel2, 62  
     fLabel3, 62  
     fLabel4, 62  
     fLabelsDrawn, 62  
     fLegend, 62  
     fLegendNumEntries, 62  
     fLegendX, 62  
     fLegendY, 62  
     fRunDate, 62  
     fRunId, 62  
     fSpillId, 62  
     fUpperLabel, 62  
     fUpperLabelText, 62  
     fWidth, 62  
     FillChannel, 61  
     GetCoordinates, 61  
     Grid, 61  
     QuarticCanvas, 60  
     Save, 61  
     SetRunInfo, 61  
     SetUpperLabel, 61  
 DQM::QuarticCanvas::Coord, 22  
     x, 22  
     y, 22

- DQMProcess
  - DQM::DQMProcess, [24](#)
- DT\_100ns
  - TDCSetup, [117](#)
- DT\_10ns
  - TDCSetup, [117](#)
- DT\_30ns
  - TDCSetup, [117](#)
- DT\_5ns
  - TDCSetup, [117](#)
- DeadTime
  - TDCSetup, [116](#)
- det\_mode
  - file\_header\_t, [27](#)
- DetectionMode
  - DAQ::TDC, [81](#)
- detector
  - OnlineDBHandler::TDCConditions, [86](#)
- DisableAllChannels
  - TDCControl, [89](#)
- DisableChannel
  - TDCControl, [89](#)
- Disconnect
  - Client, [19](#)
  - Messenger, [48](#)
- DisconnectClient
  - Messenger, [48](#)
- DrawGrid
  - DQM::GastofCanvas, [39](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [61](#)
- Dump
  - FileReader, [29](#)
  - Message, [44](#)
  - SocketMessage, [77](#)
  - TDCControl, [89](#)
  - TDCErrorFlag, [95](#)
  - TDCEvent, [97](#)
  - TDCMeasurement, [103](#)
  - TDCSetup, [121](#)
  - TDCStatus, [164](#)
- DumpConnected
  - Socket, [68](#)
- DumpRegister
  - TDCRegister, [108](#)
- E\_100ps
  - TDCSetup, [117](#)
- E\_12p5ns
  - TDCSetup, [117](#)
- E\_1p6ns
  - TDCSetup, [117](#)
- E\_200ps
  - TDCSetup, [117](#)
- E\_3p12ns
  - TDCSetup, [117](#)
- E\_400ps
  - TDCSetup, [117](#)
- E\_6p25ns
  - TDCSetup, [117](#)
- E\_800ps
  - TDCSetup, [117](#)
- ETTT
  - TDCEvent, [97](#)
- EdgeResolution
  - TDCSetup, [117](#)
- EnableAllChannels
  - TDCControl, [90](#)
- EnableChannel
  - TDCControl, [90](#)
- EnablePattern
  - TDCControl, [88](#)
- EnabledError
  - TDCSetup, [117](#)
- Error
  - TDCStatus, [164](#)
- ErrorState
  - DAQ::FPGAHandler, [33](#)
- EventType
  - TDCEvent, [97](#)
- fAddress
  - Socket, [71](#)
- fAddressesCanProcess
  - DQM::DQMProcess, [26](#)
- fBS
  - DAQ::TDC, [82](#)
- fBoardId
  - DQM::GastofCanvas, [40](#)
  - DQM::QuarticCanvas, [62](#)
- fBuffer
  - Logger, [41](#)
  - Socket, [71](#)
- fClientId
  - Client, [21](#)
- fControl
  - DAQ::TDC, [82](#)
- fDB
  - OnlineDBHandler, [54](#)
- fDetectorType
  - DQM::DQMProcess, [26](#)
- fDevice
  - DAQ::QuickUSBHandler, [65](#)
- fEvents
  - TDCMeasurement, [105](#)
- fFile
  - FileReader, [30](#)
- fFilename
  - DAQ::FPGAHandler, [36](#)
- fHandle
  - DAQ::QuickUSBHandler, [65](#)
- fHeader
  - FileReader, [30](#)
- fHeight
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [62](#)
- fHist

- DQM::GastofCanvas, [40](#)
- DQM::QuarticCanvas, [62](#)
- FIFOEmpty
  - TDCStatus, [165](#)
- FIFOFull
  - TDCStatus, [165](#)
- FIFOOccupancy
  - TDCStatus, [165](#)
- fId
  - DAQ::TDC, [82](#)
- flsConnected
  - Client, [21](#)
- flsFileOpen
  - DAQ::FPGAHandler, [36](#)
- flsStopping
  - DAQ::QuickUSBHandler, [65](#)
- flsTDCInReadout
  - DAQ::FPGAHandler, [36](#)
- fLabel1
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [62](#)
- fLabel2
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [62](#)
- fLabel3
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [62](#)
- fLabel4
  - DQM::GastofCanvas, [40](#)
  - DQM::QuarticCanvas, [62](#)
- fLabelsDrawn
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [57](#)
  - DQM::QuarticCanvas, [62](#)
- fLegend
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [58](#)
  - DQM::QuarticCanvas, [62](#)
- fLegendNumEntries
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [58](#)
  - DQM::QuarticCanvas, [62](#)
- fLegendX
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [58](#)
  - DQM::QuarticCanvas, [62](#)
- fLegendY
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [58](#)
  - DQM::QuarticCanvas, [62](#)
- fMap
  - TDCMeasurement, [105](#)
- fMaster
  - Socket, [71](#)
- fMessage
  - SocketMessage, [79](#)
- fNumAttempts
  - Messenger, [51](#)
- fNumEvents
  - FileReader, [30](#)
- fNumWords
  - TDCRegister, [109](#)
- fOrder
  - DQM::DQMProcess, [26](#)
- fOutput
  - DAQ::FPGAHandler, [36](#)
- FPGA board control, [10](#)
- FPGAHandler
  - DAQ::FPGAHandler, [32](#)
- fPID
  - Messenger, [51](#)
- fPort
  - Socket, [71](#)
- fReadFds
  - Socket, [71](#)
- fReadoutMode
  - FileReader, [30](#)
- fRedirect
  - LogRedirector, [42](#)
- fRunDate
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [58](#)
  - DQM::QuarticCanvas, [62](#)
- fRunId
  - DQM::GastofCanvas, [40](#)
  - DQM::PPSCanvas, [58](#)
  - DQM::QuarticCanvas, [62](#)
- fRunNumber
  - DQM::DQMProcess, [26](#)
- fSS
  - LogRedirector, [42](#)
- fSetup
  - DAQ::TDC, [82](#)
- fSocketId
  - Socket, [71](#)
- fSocketsConnected
  - Socket, [71](#)
- fSpillId
  - DQM::GastofCanvas, [40](#)
  - DQM::QuarticCanvas, [62](#)
- fStatus
  - DAQ::TDC, [82](#)
- fStderrPipe
  - Messenger, [51](#)
- fStdoutPipe
  - Messenger, [51](#)
- fStream
  - Logger, [41](#)
- fStreamId
  - DAQ::QuickUSBHandler, [65](#)
- fString
  - Message, [44](#)
- fTDC

- DAQ::FPGAHandler, 36
- fType
  - Client, 21
- fUSB
  - DAQ::TDC, 82
- fUpperLabel
  - DQM::GastofCanvas, 40
  - DQM::PPSCanvas, 58
  - DQM::QuarticCanvas, 62
- fUpperLabelText
  - DQM::GastofCanvas, 40
  - DQM::PPSCanvas, 58
  - DQM::QuarticCanvas, 62
- fWidth
  - DQM::GastofCanvas, 40
  - DQM::PPSCanvas, 58
  - DQM::QuarticCanvas, 62
- fWord
  - TDCErrorFlag, 95
  - TDCEvent, 102
  - TDCRegister, 109
- fWordSize
  - TDCRegister, 109
- fWriteTime
  - FileReader, 30
- Fetch
  - DAQ::QuickUSBHandler, 64
- FetchEvents
  - DAQ::TDC, 82
- FetchMessage
  - Socket, 69
- file\_header\_t, 26
  - acq\_mode, 27
  - det\_mode, 27
  - magic, 27
  - num\_hptdc, 27
  - run\_id, 27
  - spill\_id, 27
- FileReader, 27
  - ~FileReader, 29
  - Clear, 29
  - Dump, 29
  - fFile, 30
  - fHeader, 30
  - fNumEvents, 30
  - fReadoutMode, 30
  - fWriteTime, 30
  - FileReader, 28
  - GetAcquisitionMode, 29
  - GetBurstId, 29
  - GetDetectionMode, 29
  - GetNextEvent, 29
  - GetNextMeasurement, 29
  - GetNumEvents, 30
  - GetNumTDCs, 30
  - GetRunId, 30
  - IsOpen, 30
  - Open, 30
- FillChannel
  - DQM::GastofCanvas, 39
  - DQM::QuarticCanvas, 61
- Filler
  - TDCEvent, 97
- GastofCanvas
  - DQM::GastofCanvas, 38
- GetAcquisitionMode
  - FileReader, 29
- GetBits
  - TDCRegister, 108
- GetBunchId
  - TDCEvent, 98
  - TDCMeasurement, 104
- GetBurstId
  - FileReader, 29
- GetChannelId
  - TDCEvent, 98
  - TDCMeasurement, 104
- GetChannelOffset
  - TDCSetup, 122
- GetCleanedValue
  - SocketMessage, 77
- GetCoarseCountOffset
  - TDCSetup, 122
- GetCoordinates
  - DQM::GastofCanvas, 39
  - DQM::QuarticCanvas, 61
- GetDLLAdjustment
  - TDCSetup, 123
- GetDLLReset
  - TDCControl, 90
- GetDLLVersion
  - DAQ::QuickUSBHandler, 64
- GetDeadTime
  - TDCSetup, 123
- GetDetectionMode
  - FileReader, 29
- GetDriverVersion
  - DAQ::QuickUSBHandler, 64
- GetETTT
  - TDCEvent, 99
  - TDCMeasurement, 104
- GetEdgeResolution
  - TDCSetup, 123
- GetEdgesPairing
  - TDCSetup, 124
- GetEnableError
  - TDCSetup, 124
- GetEnableErrorBypass
  - TDCSetup, 124
- GetEnableErrorMark
  - TDCSetup, 125
- GetEnableJTAGReadout
  - TDCSetup, 125
- GetEnablePattern
  - TDCControl, 91
- GetEnableReadoutOccupancy

- TDCSetup, [125](#)
- GetEnableReadoutSeparator
  - TDCSetup, [126](#)
- GetEnableSerial
  - TDCSetup, [126](#)
- GetErrorFlags
  - TDCEvent, [98](#)
- GetEventCount
  - TDCEvent, [99](#)
- GetEventId
  - TDCEvent, [99](#)
  - TDCMeasurement, [104](#)
- GetFWVersion
  - DAQ::QuickUSBHandler, [64](#)
- GetFilename
  - DAQ::FPGAHandler, [33](#)
- GetGeo
  - TDCEvent, [100](#)
- GetGlobalReset
  - TDCControl, [91](#)
- GetIntValue
  - SocketMessage, [77](#)
- GetKey
  - Message, [44](#)
  - SocketMessage, [77](#)
- GetLastBurst
  - OnlineDBHandler, [53](#)
- GetLastRun
  - OnlineDBHandler, [53](#)
- GetLeadingMode
  - TDCSetup, [126](#)
- GetLeadingTime
  - TDCMeasurement, [104](#)
- GetMatchWindow
  - TDCSetup, [127](#)
- GetMaxEventSize
  - TDCSetup, [127](#)
- GetNextEvent
  - FileReader, [29](#)
- GetNextMeasurement
  - FileReader, [29](#)
- GetNumEvents
  - FileReader, [30](#)
- GetNumTDCs
  - FileReader, [30](#)
- GetNumWords
  - TDCRegister, [108](#)
- GetPLLReset
  - TDCControl, [91](#)
- GetPort
  - Socket, [69](#)
- GetRCAdjustment
  - TDCSetup, [127](#)
- GetReadoutFIFOSize
  - TDCSetup, [128](#)
- GetRejectCountOffset
  - TDCSetup, [128](#)
- GetRejectFIFOFull
  - TDCSetup, [128](#)
- GetRunId
  - FileReader, [30](#)
- GetRunInfo
  - OnlineDBHandler, [53](#)
- GetRuns
  - OnlineDBHandler, [53](#)
- GetSearchWindow
  - TDCSetup, [129](#)
- GetSetupParity
  - TDCSetup, [129](#)
- GetSetupRegister
  - DAQ::TDC, [82](#)
- GetSocketId
  - Socket, [69](#)
- GetSocketType
  - Socket, [69](#)
- GetStatus
  - TDCEvent, [100](#)
- GetString
  - Message, [44](#)
  - SocketMessage, [77](#)
- GetTDC
  - DAQ::FPGAHandler, [33](#)
- GetTDCConditions
  - OnlineDBHandler, [53](#)
- GetTDCCControl
  - DAQ::FPGAHandler, [33](#)
- GetTDCId
  - TDCEvent, [100](#)
  - TDCMeasurement, [104](#)
  - TDCSetup, [129](#)
- GetTDCStatus
  - DAQ::FPGAHandler, [33](#)
- GetTestInvert
  - TDCSetup, [130](#)
- GetTestMode
  - TDCSetup, [130](#)
- GetTime
  - TDCEvent, [101](#)
- GetToT
  - TDCMeasurement, [104](#)
- GetTrailingMode
  - TDCSetup, [130](#)
- GetTrailingTime
  - TDCMeasurement, [105](#)
- GetTriggerCountOffset
  - TDCSetup, [131](#)
- GetTriggerLatency
  - TDCSetup, [131](#)
- GetTriggerMatchingMode
  - TDCSetup, [131](#)
- GetType
  - Client, [19](#)
  - DAQ::FPGAHandler, [33](#)
  - Messenger, [49](#)
  - TDCEvent, [101](#)
- GetValue

- SocketMessage, 77
- GetVectorValue
  - SocketMessage, 78
- GetVernierOffset
  - TDCSetup, 132
- GetWidth
  - TDCEvent, 101
- GetWidthResolution
  - TDCSetup, 132
- GetWord
  - TDCErrorFlag, 95
  - TDCEvent, 101
  - TDCRegister, 108
- GetWordCount
  - TDCEvent, 102
- GetWords
  - TDCRegister, 108
- GlobalHeader
  - TDCEvent, 97
- GlobalTrailer
  - TDCEvent, 97
- Grid
  - DQM::GastofCanvas, 39
  - DQM::PPSCanvas, 57
  - DQM::QuarticCanvas, 61
- HPTDC chip control, 11
  - AcquisitionMode, 11
  - CONT\_STORAGE, 11
  - TRIG\_MATCH, 11
- HasGroupError
  - TDCErrorFlag, 95
- HasInternalChipError
  - TDCErrorFlag, 95
- HasL1BufferOverflow
  - TDCErrorFlag, 95
- HasReachedEventSizeLimit
  - TDCErrorFlag, 95
- HasReadoutFIFOOverflow
  - TDCErrorFlag, 95
- HasTriggerFIFOOverflow
  - TDCErrorFlag, 95
- HaveToken
  - TDCStatus, 166
- INVALID
  - Socket, 67
- IO\_aux\_clock
  - TDCSetup, 118
- IO\_clock\_40
  - TDCSetup, 118
- IO\_pll\_clock\_160
  - TDCSetup, 118
- IO\_pll\_clock\_80
  - TDCSetup, 118
- IOClockSource
  - TDCSetup, 118
- Init
  - DAQ::QuickUSBHandler, 64
- IsFromWeb
  - Message, 44
- IsInRun
  - DQM::DQMProcess, 24
- IsOpen
  - FileReader, 30
- IsTrailing
  - TDCEvent, 102
- IsWebSocket
  - Socket, 69
- JTAGInstructionParityError
  - TDCSetup, 118
- kAuxClock
  - TDCBoundaryScan, 85
- kBunchReset
  - TDCBoundaryScan, 85
- kClk
  - TDCBoundaryScan, 85
- kCoarseCountOffset
  - TDCSetup, 158
- kControlParity
  - TDCControl, 93
- kCoreClockDelay
  - TDCSetup, 158
- kCoreClockSource
  - TDCSetup, 158
- kDLLClockDelay
  - TDCSetup, 158
- kDLLClockSource
  - TDCSetup, 158
- kDLLControl
  - TDCSetup, 159
- kDLLLock
  - TDCStatus, 167
- kDLLMode
  - TDCSetup, 159
- kDLLReset
  - TDCControl, 93
- kDLLTapAdjust0
  - TDCSetup, 159
- kDataReady
  - TDCBoundaryScan, 85
- kDeadTime
  - TDCSetup, 158
- kEnableAutomaticReject
  - TDCSetup, 159
- kEnableBytewise
  - TDCSetup, 159
- kEnableChannel
  - TDCControl, 93
- kEnableDirectBunchReset
  - TDCSetup, 159
- kEnableDirectEventReset
  - TDCSetup, 159
- kEnableDirectTrigger
  - TDCSetup, 159
- kEnableError



- TDCSetup, 159
- kEnableErrorBypass
  - TDCSetup, 159
- kEnableErrorMark
  - TDCSetup, 159
- kEnableGlobalHeader
  - TDCSetup, 159
- kEnableGlobalTrailer
  - TDCSetup, 159
- kEnableJTAGReadout
  - TDCSetup, 159
- kEnableLocalHeader
  - TDCSetup, 159
- kEnableLocalTrailer
  - TDCSetup, 159
- kEnableMasterResetCode
  - TDCSetup, 159
- kEnableMasterResetOnEventReset
  - TDCSetup, 159
- kEnableMatching
  - TDCSetup, 159
- kEnableOverflowDetect
  - TDCSetup, 159
- kEnablePair
  - TDCSetup, 159
- kEnablePattern
  - TDCControl, 93
- kEnableReadoutOccupancy
  - TDCSetup, 159
- kEnableReadoutSeparator
  - TDCSetup, 159
- kEnableRelative
  - TDCSetup, 159
- kEnableResetChannelBufferWhenSeparator
  - TDCSetup, 159
- kEnableSeparatorOnBunchReset
  - TDCSetup, 159
- kEnableSeparatorOnEventReset
  - TDCSetup, 159
- kEnableSerial
  - TDCSetup, 159
- kEnableSetCountersOnBunchReset
  - TDCSetup, 160
- kEnableTTLClock
  - TDCSetup, 160
- kEnableTTLControl
  - TDCSetup, 160
- kEnableTTLHit
  - TDCSetup, 160
- kEnableTTLReset
  - TDCSetup, 160
- kEnableTTLSerial
  - TDCSetup, 160
- kEncodedControl
  - TDCBoundaryScan, 85
- kError
  - TDCBoundaryScan, 85
  - TDCStatus, 167
- kEventCountOffset
  - TDCSetup, 160
- kEventReset
  - TDCBoundaryScan, 85
- kGetData
  - TDCBoundaryScan, 85
- kGlobalReset
  - TDCControl, 93
- kHaveToken
  - TDCStatus, 167
- kHit
  - TDCBoundaryScan, 85
- kIOClockDelay
  - TDCSetup, 160
- kIOClockSource
  - TDCSetup, 160
- kKeepToken
  - TDCSetup, 160
- kL1Occupancy
  - TDCStatus, 167
- kLeading
  - TDCSetup, 160
- kLeadingResolution
  - TDCSetup, 160
- kLowPowerMode
  - TDCSetup, 160
- kMaster
  - TDCSetup, 160
- kMatchWindow
  - TDCSetup, 160
- kMaxEventSize
  - TDCSetup, 160
- kModeRC
  - TDCSetup, 160
- kModeRCCompression
  - TDCSetup, 160
- kOffset0
  - TDCSetup, 160
- kPLLControl
  - TDCSetup, 160
- kPLLReset
  - TDCControl, 93
- kParallelDataOut
  - TDCBoundaryScan, 85
- kParallelEnable
  - TDCBoundaryScan, 85
- kRCAdjust0
  - TDCSetup, 160
- kReadoutFIFOEmpty
  - TDCStatus, 167
- kReadoutFIFOFull
  - TDCStatus, 167
- kReadoutFIFOOccupancy
  - TDCStatus, 168
- kReadoutFIFOSize
  - TDCSetup, 160
- kReadoutSingleCycleSpeed
  - TDCSetup, 160

- kReadoutSpeedSelect
  - TDCSetup, [160](#)
- kRejectCountOffset
  - TDCSetup, [160](#)
- kRejectFIFOFull
  - TDCSetup, [160](#)
- kReset
  - TDCBoundaryScan, [85](#)
- kRollOver
  - TDCSetup, [160](#)
- kSearchWindow
  - TDCSetup, [160](#)
- kSelectBypassInputs
  - TDCSetup, [161](#)
- kSerialBypassIn
  - TDCBoundaryScan, [85](#)
- kSerialClockDelay
  - TDCSetup, [161](#)
- kSerialClockSource
  - TDCSetup, [161](#)
- kSerialDelay
  - TDCSetup, [161](#)
- kSerialIn
  - TDCBoundaryScan, [85](#)
- kSerialOut
  - TDCBoundaryScan, [85](#)
- kSetupParity
  - TDCSetup, [161](#)
- kStrobeOut
  - TDCBoundaryScan, [85](#)
- kStrobeSelect
  - TDCSetup, [161](#)
- kTDCId
  - TDCSetup, [161](#)
- kTest
  - TDCBoundaryScan, [85](#)
- kTestInvert
  - TDCSetup, [161](#)
- kTestMode
  - TDCSetup, [161](#)
- kTestSelect
  - TDCSetup, [161](#)
- kTokenBypassIn
  - TDCBoundaryScan, [85](#)
- kTokenDelay
  - TDCSetup, [161](#)
- kTokenIn
  - TDCBoundaryScan, [85](#)
- kTokenOut
  - TDCBoundaryScan, [85](#)
- kTrailing
  - TDCSetup, [161](#)
- kTrigger
  - TDCBoundaryScan, [85](#)
- kTriggerCountOffset
  - TDCSetup, [161](#)
- kTriggerFIFOEmpty
  - TDCStatus, [168](#)
- kTriggerFIFOFull
  - TDCStatus, [168](#)
- kTriggerFIFOOccupancy
  - TDCStatus, [168](#)
- kVernierOffset
  - TDCSetup, [161](#)
- kWidthSelect
  - TDCSetup, [161](#)
- L1BufferParityError
  - TDCSetup, [117](#)
- L1Occupancy
  - TDCStatus, [166](#)
- Listen
  - Socket, [69](#)
- LogRedirector, [41](#)
  - contents, [42](#)
  - fRedirect, [42](#)
  - fSS, [42](#)
  - LogRedirector, [42](#)
- Logger, [40](#)
  - ~Logger, [41](#)
  - fBuffer, [41](#)
  - fStream, [41](#)
  - Logger, [41](#)
- MASTER
  - Socket, [67](#)
- magic
  - file\_header\_t, [27](#)
- MajorVersion
  - DAQ::QuickUSBHandler::Version, [168](#)
- Message, [42](#)
  - ~Message, [43](#)
  - Dump, [44](#)
  - fString, [44](#)
  - GetKey, [44](#)
  - GetString, [44](#)
  - IsFromWeb, [44](#)
  - Message, [43](#)
- Messenger, [44](#)
  - ~Messenger, [46](#)
  - AddClient, [47](#)
  - Broadcast, [47](#)
  - Connect, [47](#)
  - Disconnect, [48](#)
  - DisconnectClient, [48](#)
  - fNumAttempts, [51](#)
  - fPID, [51](#)
  - fStderrPipe, [51](#)
  - fStdoutPipe, [51](#)
  - GetType, [49](#)
  - Messenger, [46](#)
  - ProcessMessage, [49](#)
  - Receive, [49](#)
  - Send, [49](#)
  - SendAll, [50](#)
  - StartAcquisition, [50](#)
  - StopAcquisition, [51](#)

- SwitchClientType, 51
- MinorVersion
  - DAQ::QuickUSBHandler::Version, 168
- NewBurst
  - OnlineDBHandler, 53
- NewPlot
  - DQM::DQMProcess, 24
- NewRun
  - OnlineDBHandler, 54
- num\_hptdc
  - file\_header\_t, 27
- NumErrors
  - TDCMeasurement, 105
- NumEvents
  - TDCMeasurement, 105
- OLEADING
  - DAQ::TDC, 81
- OTRAILING
  - DAQ::TDC, 81
- Object
  - SocketMessage, 78
- OnlineDBHandler, 51
  - ~OnlineDBHandler, 53
  - BuildTables, 53
  - BurstInfos, 53
  - fDB, 54
  - GetLastBurst, 53
  - GetLastRun, 53
  - GetRunInfo, 53
  - GetRuns, 53
  - GetTDCConditions, 53
  - NewBurst, 53
  - NewRun, 54
  - OnlineDBHandler, 53
  - RunCollection, 53
  - Select, 54
  - SetHVConditions, 54
  - SetTDCConditions, 54
  - TDCConditionsCollection, 53
- OnlineDBHandler::BurstInfo, 15
  - burst\_id, 15
  - time\_start, 15
- OnlineDBHandler::TDCConditions, 86
  - detector, 86
  - operator=, 86
  - operator==, 86
  - run\_id, 86
  - tdc\_acq\_mode, 86
  - tdc\_address, 86
  - tdc\_det\_mode, 86
  - tdc\_id, 86
- Open
  - FileReader, 30
- OpenFile
  - DAQ::FPGAHandler, 33
- operator<<
  - TDCErrorFlag, 95
- operator=
  - OnlineDBHandler::TDCConditions, 86
  - TDCRegister, 108
- operator==
  - OnlineDBHandler::TDCConditions, 86
- PAIR
  - DAQ::TDC, 81
- PPSCanvas
  - DQM::PPSCanvas, 56
- ParseMessage
  - Client, 19
  - DQM::DQMProcess, 25
- PrepareConnection
  - Socket, 69
- ProcessMessage
  - Messenger, 49
- QuarticCanvas
  - DQM::QuarticCanvas, 60
- QuickUSBHandler
  - DAQ::QuickUSBHandler, 64
- R\_DLLReset
  - TDCControl, 88
- R\_EnablePattern
  - TDCControl, 88
- R\_GlobalReset
  - TDCControl, 88
- R\_PLLReset
  - TDCControl, 88
- RO\_Fixed
  - TDCSetup, 118
- RO\_pll\_80Mbits\_s
  - TDCSetup, 118
- RSC\_10Mbits\_s
  - TDCSetup, 118
- RSC\_1p25Mbits\_s
  - TDCSetup, 118
- RSC\_20Mbits\_s
  - TDCSetup, 118
- RSC\_2p5Mbits\_s
  - TDCSetup, 118
- RSC\_312p5kbits\_s
  - TDCSetup, 118
- RSC\_40Mbits\_s
  - TDCSetup, 118
- RSC\_5Mbits\_s
  - TDCSetup, 118
- RSC\_625kbits\_s
  - TDCSetup, 118
- ReadConfiguration
  - DAQ::TDC, 82
- ReadRegister
  - DAQ::TDC, 82
- ReadStatus
  - DAQ::TDC, 82
- ReadoutFIFOParityError
  - TDCSetup, 118

- ReadoutSingleCycleSpeed
  - TDCSetup, 118
- ReadoutSpeed
  - TDCSetup, 118
- ReadoutStateError
  - TDCSetup, 118
- Receive
  - Client, 19, 20
  - Messenger, 49
- RegisterName
  - TDCControl, 88
- RegisterTest
  - DAQ::FPGAHandler, 34
- Reset
  - DAQ::QuickUSBHandler, 64
- RetrieveSetupWord
  - DAQ::FPGAHandler, 34
- Run
  - DQM::DQMProcess, 25
- run\_id
  - file\_header\_t, 27
  - OnlineDBHandler::TDCConditions, 86
- RunCollection
  - OnlineDBHandler, 53
- SS\_DSStrobe
  - TDCSetup, 119
- SS\_LeadingEdge
  - TDCSetup, 119
- SS\_LeadingTrailingStrobe
  - TDCSetup, 119
- SS\_NoStrobe
  - TDCSetup, 119
- Save
  - DQM::GastofCanvas, 39
  - DQM::PPSCanvas, 57
  - DQM::QuarticCanvas, 61
- Select
  - OnlineDBHandler, 54
- SelectConnections
  - Socket, 70
- Send
  - Client, 20
  - Messenger, 49
- SendAll
  - Messenger, 50
- SendAndReceive
  - Client, 21
- SendConfiguration
  - DAQ::TDC, 82
- SendMessage
  - Socket, 70
- SendSetupWord
  - DAQ::FPGAHandler, 34
- Serial\_aux\_clock
  - TDCSetup, 118
- Serial\_pll\_clock\_160
  - TDCSetup, 118
- Serial\_pll\_clock\_40
  - TDCSetup, 118
- Serial\_pll\_clock\_80
  - TDCSetup, 118
- SerialClockSource
  - TDCSetup, 118
- SerialStrobeType
  - TDCSetup, 118
- SetAllChannelsOffset
  - TDCSetup, 132
- SetAllTapsDLLAdjustment
  - TDCSetup, 133
- SetBits
  - TDCRegister, 108
- SetBypassInputs
  - TDCSetup, 133
- SetChannelOffset
  - TDCSetup, 133
- SetCoarseCountOffset
  - TDCSetup, 133
- SetConstantValues
  - TDCBoundaryScan, 85
  - TDCControl, 92
  - TDCRegister, 108
  - TDCSetup, 134
  - TDCStatus, 166
- SetControlParity
  - TDCControl, 92
- SetCoreClockDelay
  - TDCSetup, 134
- SetCoreClockSource
  - TDCSetup, 134
- SetDLLAdjustment
  - TDCSetup, 135
- SetDLLClockDelay
  - TDCSetup, 135
- SetDLLClockSource
  - TDCSetup, 135
- SetDLLControl
  - TDCSetup, 136
- SetDLLMode
  - TDCSetup, 136
- SetDLLReset
  - TDCControl, 92
- SetDeadTime
  - TDCSetup, 134
- SetEdgeResolution
  - TDCSetup, 136
- SetEdgesPairing
  - TDCSetup, 137
- SetEnableAutomaticReject
  - TDCSetup, 137
- SetEnableBytewise
  - TDCSetup, 137
- SetEnableDirectBunchReset
  - TDCSetup, 138
- SetEnableDirectEventReset
  - TDCSetup, 138
- SetEnableDirectTrigger

- TDCSetup, 138
- SetEnableError
  - TDCSetup, 139
- SetEnableErrorBypass
  - TDCSetup, 139
- SetEnableErrorMark
  - TDCSetup, 139
- SetEnableGlobalHeader
  - TDCSetup, 140
- SetEnableGlobalTrailer
  - TDCSetup, 140
- SetEnableJTAGReadout
  - TDCSetup, 140
- SetEnableLocalHeader
  - TDCSetup, 141
- SetEnableLocalTrailer
  - TDCSetup, 141
- SetEnableMasterResetCode
  - TDCSetup, 141
- SetEnableMasterResetOnEventReset
  - TDCSetup, 142
- SetEnableOverflowDetect
  - TDCSetup, 142
- SetEnablePattern
  - TDCControl, 93
- SetEnableReadoutOccupancy
  - TDCSetup, 142
- SetEnableReadoutSeparator
  - TDCSetup, 143
- SetEnableRelative
  - TDCSetup, 143
- SetEnableResetChannelBufferWhenSeparator
  - TDCSetup, 143
- SetEnableSeparatorOnBunchReset
  - TDCSetup, 144
- SetEnableSeparatorOnEventReset
  - TDCSetup, 144
- SetEnableSerial
  - TDCSetup, 144
- SetEnableSetCountersOnBunchReset
  - TDCSetup, 145
- SetEnableTTLClock
  - TDCSetup, 145
- SetEnableTTLControl
  - TDCSetup, 145
- SetEnableTTLHit
  - TDCSetup, 146
- SetEnableTTLReset
  - TDCSetup, 146
- SetEnableTTLSerial
  - TDCSetup, 146
- SetEventCountOffset
  - TDCSetup, 147
- SetEventsCollection
  - TDCMeasurement, 105
- SetGlobalReset
  - TDCControl, 93
- SetHVConditions
  - OnlineDBHandler, 54
- SetIOClockDelay
  - TDCSetup, 147
- SetIOClockSource
  - TDCSetup, 147
- SetKeepToken
  - TDCSetup, 148
- SetKeyValue
  - SocketMessage, 78, 79
- SetLeadingMode
  - TDCSetup, 148
- SetLowPowerMode
  - TDCSetup, 148
- SetMaster
  - TDCSetup, 149
- SetMatchWindow
  - TDCSetup, 149
- SetMaxEventSize
  - TDCSetup, 149
- SetModeRC
  - TDCSetup, 150
- SetModeRCCompression
  - TDCSetup, 150
- SetPLLControl
  - TDCSetup, 150
- SetPLLReset
  - TDCControl, 93
- SetPort
  - Socket, 70
- SetRCAdjustment
  - TDCSetup, 151
- SetReadoutFIFOSize
  - TDCSetup, 151
- SetReadoutSingleCycleSpeed
  - TDCSetup, 151
- SetReadoutSpeedSelect
  - TDCSetup, 152
- SetRejectCountOffset
  - TDCSetup, 152
- SetRejectFIFOFull
  - TDCSetup, 152
- SetRollOver
  - TDCSetup, 153
- SetRunInfo
  - DQM::GastofCanvas, 39
  - DQM::PPSCanvas, 57
  - DQM::QuarticCanvas, 61
- SetSearchWindow
  - TDCSetup, 153
- SetSerialClockDelay
  - TDCSetup, 153
- SetSerialClockSource
  - TDCSetup, 154
- SetSerialDelay
  - TDCSetup, 154
- SetSetupParity
  - TDCSetup, 154
- SetSetupRegister

- DAQ::TDC, 82
- SetSocketId
  - Socket, 70
- SetStrobeSelect
  - TDCSetup, 155
- SetTDCConditions
  - OnlineDBHandler, 54
- SetTDCId
  - TDCSetup, 155
- SetTDCSetup
  - DAQ::FPGAHandler, 35
- SetTest
  - TDCSetup, 155
- SetTestInvert
  - TDCSetup, 156
- SetTestMode
  - TDCSetup, 156
- SetTokenDelay
  - TDCSetup, 156
- SetTrailingMode
  - TDCSetup, 157
- SetTriggerCountOffset
  - TDCSetup, 157
- SetTriggerMatchingMode
  - TDCSetup, 157
- SetUpperLabel
  - DQM::GastofCanvas, 39
  - DQM::PPSCanvas, 57
  - DQM::QuarticCanvas, 61
- SetVernierOffset
  - TDCSetup, 158
- SetWidthResolution
  - TDCSetup, 158
- SetWord
  - TDCEvent, 102
  - TDCRegister, 108
- SetupParityError
  - TDCSetup, 118
- Socket, 65
  - ~Socket, 67
  - AcceptConnections, 68
  - Bind, 68
  - CLIENT, 67
  - Configure, 68
  - Create, 68
  - DAQ, 67
  - DETECTOR, 67
  - DQM, 67
  - DumpConnected, 68
  - fAddress, 71
  - fBuffer, 71
  - fMaster, 71
  - fPort, 71
  - fReadFds, 71
  - fSocketId, 71
  - fSocketsConnected, 71
  - FetchMessage, 69
  - GetPort, 69
  - GetSocketId, 69
  - GetSocketType, 69
  - INVALID, 67
  - IsWebSocket, 69
  - Listen, 69
  - MASTER, 67
  - PrepareConnection, 69
  - SelectConnections, 70
  - SendMessage, 70
  - SetPort, 70
  - SetSocketId, 70
  - Socket, 67
  - SocketCollection, 67
  - SocketType, 67
  - Start, 70
  - Stop, 71
  - WEBSOCKET\_CLIENT, 67
- Socket communication objects, 9
- SocketCollection
  - Socket, 67
- SocketMessage, 72
  - ~SocketMessage, 77
  - Dump, 77
  - fMessage, 79
  - GetCleanedValue, 77
  - GetIntValue, 77
  - GetKey, 77
  - GetString, 77
  - GetValue, 77
  - GetVectorValue, 78
  - Object, 78
  - SetKeyValue, 78, 79
  - SocketMessage, 74–77
  - String, 79
- SocketType
  - Socket, 67
- SoftReset
  - DAQ::TDC, 82
- spill\_id
  - file\_header\_t, 27
- Start
  - Socket, 70
- StartAcquisition
  - DAQ::FPGAHandler, 35
  - Messenger, 50
- StartBulkTransfer
  - DAQ::QuickUSBHandler, 65
- Stop
  - DAQ::FPGAHandler, 35
  - Socket, 71
- StopAcquisition
  - DAQ::FPGAHandler, 35
  - Messenger, 51
- StopBulkTransfer
  - DAQ::QuickUSBHandler, 65
- String
  - SocketMessage, 79
- SwitchClientType

- Messenger, 51
- TDC
  - DAQ::TDC, 81
- TDCBoundaryScan, 83
  - kAuxClock, 85
  - kBunchReset, 85
  - kClk, 85
  - kDataReady, 85
  - kEncodedControl, 85
  - kError, 85
  - kEventReset, 85
  - kGetData, 85
  - kHit, 85
  - kParallelDataOut, 85
  - kParallelEnable, 85
  - kReset, 85
  - kSerialBypassIn, 85
  - kSerialIn, 85
  - kSerialOut, 85
  - kStrobeOut, 85
  - kTest, 85
  - kTokenBypassIn, 85
  - kTokenIn, 85
  - kTokenOut, 85
  - kTrigger, 85
  - SetConstantValues, 85
  - TDCBoundaryScan, 84
- TDCConditionsCollection
  - OnlineDBHandler, 53
- TDCControl, 86
  - DisableAllChannels, 89
  - DisableChannel, 89
  - Dump, 89
  - EnableAllChannels, 90
  - EnableChannel, 90
  - EnablePattern, 88
  - GetDLLReset, 90
  - GetEnablePattern, 91
  - GetGlobalReset, 91
  - GetPLLReset, 91
  - kControlParity, 93
  - kDLLReset, 93
  - kEnableChannel, 93
  - kEnablePattern, 93
  - kGlobalReset, 93
  - kPLLReset, 93
  - R\_DLLReset, 88
  - R\_EnablePattern, 88
  - R\_GlobalReset, 88
  - R\_PLLReset, 88
  - RegisterName, 88
  - SetConstantValues, 92
  - SetControlParity, 92
  - SetDLLReset, 92
  - SetEnablePattern, 93
  - SetGlobalReset, 93
  - SetPLLReset, 93
  - TDCControl, 88, 89
- TDCErrors
  - TDCErrors, 97
- TDCErrorsFlag, 94
  - ~TDCErrorsFlag, 95
  - Dump, 95
  - fWord, 95
  - GetWord, 95
  - HasGroupError, 95
  - HasInternalChipError, 95
  - HasL1BufferOverflow, 95
  - HasReachedEventSizeLimit, 95
  - HasReadoutFIFOOverflow, 95
  - HasTriggerFIFOOverflow, 95
  - operator<<, 95
  - TDCErrorsFlag, 95
- TDCErrorsEvent, 95
  - ~TDCErrorsEvent, 97
  - Dump, 97
  - ETTT, 97
  - EventType, 97
  - fWord, 102
  - Filler, 97
  - GetBunchId, 98
  - GetChannelId, 98
  - GetETTT, 99
  - GetErrorFlags, 98
  - GetEventCount, 99
  - GetEventId, 99
  - GetGeo, 100
  - GetStatus, 100
  - GetTDCErrorsId, 100
  - GetTime, 101
  - GetType, 101
  - GetWidth, 101
  - GetWord, 101
  - GetWordCount, 102
  - GlobalHeader, 97
  - GlobalTrailer, 97
  - IsTrailing, 102
  - SetWord, 102
  - TDCErrors, 97
  - TDCErrorsEvent, 97
  - TDCErrorsHeader, 97
  - TDCErrorsMeasurement, 97
  - TDCErrorsTrailer, 97
  - Trigger, 97
- TDCErrorsHeader
  - TDCErrorsEvent, 97
- TDCErrorsMeasurement, 102
  - ~TDCErrorsMeasurement, 103
  - Dump, 103
  - fEvents, 105
  - fMap, 105
  - GetBunchId, 104
  - GetChannelId, 104
  - GetETTT, 104
  - GetEventId, 104
  - GetLeadingTime, 104

- GetTDCId, [104](#)
- GetToT, [104](#)
- GetTrailingTime, [105](#)
- NumErrors, [105](#)
- NumEvents, [105](#)
- SetEventsCollection, [105](#)
- TDCEvent, [97](#)
- TDCMeasurement, [103](#)
- TDCRegister, [105](#)
  - ~TDCRegister, [107](#)
  - bit, [107](#)
  - Clear, [108](#)
  - DumpRegister, [108](#)
  - fNumWords, [109](#)
  - fWord, [109](#)
  - fWordSize, [109](#)
  - GetBits, [108](#)
  - GetNumWords, [108](#)
  - GetWord, [108](#)
  - GetWords, [108](#)
  - operator=, [108](#)
  - SetBits, [108](#)
  - SetConstantValues, [108](#)
  - SetWord, [108](#)
  - TDCRegister, [107](#)
  - word\_t, [107](#)
- TDCSetup, [109](#)
  - ChannelSelectError, [117](#)
  - CoarseError, [117](#)
  - ControlParityError, [118](#)
  - Core\_aux\_clock, [116](#)
  - Core\_clock\_40, [116](#)
  - Core\_pll\_clock\_160, [116](#)
  - Core\_pll\_clock\_80, [116](#)
  - CoreClockSource, [116](#)
  - DLL\_160MHz, [117](#)
  - DLL\_320MHz, [117](#)
  - DLL\_40MHz, [117](#)
  - DLL\_Illegal, [117](#)
  - DLL\_aux\_clock, [117](#)
  - DLL\_clock\_40, [117](#)
  - DLL\_pll\_clock\_160, [117](#)
  - DLL\_pll\_clock\_320, [117](#)
  - DLL\_pll\_clock\_40, [117](#)
  - DLLClockSource, [117](#)
  - DLLSpeedMode, [117](#)
  - DT\_100ns, [117](#)
  - DT\_10ns, [117](#)
  - DT\_30ns, [117](#)
  - DT\_5ns, [117](#)
  - DeadTime, [116](#)
  - Dump, [121](#)
  - E\_100ps, [117](#)
  - E\_12p5ns, [117](#)
  - E\_1p6ns, [117](#)
  - E\_200ps, [117](#)
  - E\_3p12ns, [117](#)
  - E\_400ps, [117](#)
  - E\_6p25ns, [117](#)
  - E\_800ps, [117](#)
  - EdgeResolution, [117](#)
  - EnabledError, [117](#)
  - GetChannelOffset, [122](#)
  - GetCoarseCountOffset, [122](#)
  - GetDLLAdjustment, [123](#)
  - GetDeadTime, [123](#)
  - GetEdgeResolution, [123](#)
  - GetEdgesPairing, [124](#)
  - GetEnableError, [124](#)
  - GetEnableErrorBypass, [124](#)
  - GetEnableErrorMark, [125](#)
  - GetEnableJTAGReadout, [125](#)
  - GetEnableReadoutOccupancy, [125](#)
  - GetEnableReadoutSeparator, [126](#)
  - GetEnableSerial, [126](#)
  - GetLeadingMode, [126](#)
  - GetMatchWindow, [127](#)
  - GetMaxEventSize, [127](#)
  - GetRCAdjustment, [127](#)
  - GetReadoutFIFOSize, [128](#)
  - GetRejectCountOffset, [128](#)
  - GetRejectFIFOFull, [128](#)
  - GetSearchWindow, [129](#)
  - GetSetupParity, [129](#)
  - GetTDCId, [129](#)
  - GetTestInvert, [130](#)
  - GetTestMode, [130](#)
  - GetTrailingMode, [130](#)
  - GetTriggerCountOffset, [131](#)
  - GetTriggerLatency, [131](#)
  - GetTriggerMatchingMode, [131](#)
  - GetVernierOffset, [132](#)
  - GetWidthResolution, [132](#)
  - IO\_aux\_clock, [118](#)
  - IO\_clock\_40, [118](#)
  - IO\_pll\_clock\_160, [118](#)
  - IO\_pll\_clock\_80, [118](#)
  - IOClockSource, [118](#)
  - JTAGInstructionParityError, [118](#)
  - kCoarseCountOffset, [158](#)
  - kCoreClockDelay, [158](#)
  - kCoreClockSource, [158](#)
  - kDLLClockDelay, [158](#)
  - kDLLClockSource, [158](#)
  - kDLLControl, [159](#)
  - kDLLMode, [159](#)
  - kDLLTapAdjust0, [159](#)
  - kDeadTime, [158](#)
  - kEnableAutomaticReject, [159](#)
  - kEnableBytewise, [159](#)
  - kEnableDirectBunchReset, [159](#)
  - kEnableDirectEventReset, [159](#)
  - kEnableDirectTrigger, [159](#)
  - kEnableError, [159](#)
  - kEnableErrorBypass, [159](#)
  - kEnableErrorMark, [159](#)



kEnableGlobalHeader, 159  
kEnableGlobalTrailer, 159  
kEnableJTAGReadout, 159  
kEnableLocalHeader, 159  
kEnableLocalTrailer, 159  
kEnableMasterResetCode, 159  
kEnableMasterResetOnEventReset, 159  
kEnableMatching, 159  
kEnableOverflowDetect, 159  
kEnablePair, 159  
kEnableReadoutOccupancy, 159  
kEnableReadoutSeparator, 159  
kEnableRelative, 159  
kEnableResetChannelBufferWhenSeparator, 159  
kEnableSeparatorOnBunchReset, 159  
kEnableSeparatorOnEventReset, 159  
kEnableSerial, 159  
kEnableSetCountersOnBunchReset, 160  
kEnableTTLClock, 160  
kEnableTTLControl, 160  
kEnableTTLHit, 160  
kEnableTTLReset, 160  
kEnableTTLSerial, 160  
kEventCountOffset, 160  
kIOClockDelay, 160  
kIOClockSource, 160  
kKeepToken, 160  
kLeading, 160  
kLeadingResolution, 160  
kLowPowerMode, 160  
kMaster, 160  
kMatchWindow, 160  
kMaxEventSize, 160  
kModeRC, 160  
kModeRCCompression, 160  
kOffset0, 160  
kPLLControl, 160  
kRCAdjust0, 160  
kReadoutFIFOSize, 160  
kReadoutSingleCycleSpeed, 160  
kReadoutSpeedSelect, 160  
kRejectCountOffset, 160  
kRejectFIFOFull, 160  
kRollOver, 160  
kSearchWindow, 160  
kSelectBypassInputs, 161  
kSerialClockDelay, 161  
kSerialClockSource, 161  
kSerialDelay, 161  
kSetupParity, 161  
kStrobeSelect, 161  
kTDCId, 161  
kTestInvert, 161  
kTestMode, 161  
kTestSelect, 161  
kTokenDelay, 161  
kTrailing, 161  
kTriggerCountOffset, 161  
kVernierOffset, 161  
kWidthSelect, 161  
L1BufferParityError, 117  
RO\_Fixed, 118  
RO\_pll\_80Mbits\_s, 118  
RSC\_10Mbits\_s, 118  
RSC\_1p25Mbits\_s, 118  
RSC\_20Mbits\_s, 118  
RSC\_2p5Mbits\_s, 118  
RSC\_312p5kbits\_s, 118  
RSC\_40Mbits\_s, 118  
RSC\_5Mbits\_s, 118  
RSC\_625kbits\_s, 118  
ReadoutFIFOParityError, 118  
ReadoutSingleCycleSpeed, 118  
ReadoutSpeed, 118  
ReadoutStateError, 118  
SS\_DSStrobe, 119  
SS\_LeadingEdge, 119  
SS\_LeadingTrailingStrobe, 119  
SS\_NoStrobe, 119  
Serial\_aux\_clock, 118  
Serial\_pll\_clock\_160, 118  
Serial\_pll\_clock\_40, 118  
Serial\_pll\_clock\_80, 118  
SerialClockSource, 118  
SerialStrobeType, 118  
SetAllChannelsOffset, 132  
SetAllTapsDLLAdjustment, 133  
SetBypassInputs, 133  
SetChannelOffset, 133  
SetCoarseCountOffset, 133  
SetConstantValues, 134  
SetCoreClockDelay, 134  
SetCoreClockSource, 134  
SetDLLAdjustment, 135  
SetDLLClockDelay, 135  
SetDLLClockSource, 135  
SetDLLControl, 136  
SetDLLMode, 136  
SetDeadTime, 134  
SetEdgeResolution, 136  
SetEdgesPairing, 137  
SetEnableAutomaticReject, 137  
SetEnableBytewise, 137  
SetEnableDirectBunchReset, 138  
SetEnableDirectEventReset, 138  
SetEnableDirectTrigger, 138  
SetEnableError, 139  
SetEnableErrorBypass, 139  
SetEnableErrorMark, 139  
SetEnableGlobalHeader, 140  
SetEnableGlobalTrailer, 140  
SetEnableJTAGReadout, 140  
SetEnableLocalHeader, 141  
SetEnableLocalTrailer, 141  
SetEnableMasterResetCode, 141  
SetEnableMasterResetOnEventReset, 142

- SetEnableOverflowDetect, 142
- SetEnableReadoutOccupancy, 142
- SetEnableReadoutSeparator, 143
- SetEnableRelative, 143
- SetEnableResetChannelBufferWhenSeparator, 143
- SetEnableSeparatorOnBunchReset, 144
- SetEnableSeparatorOnEventReset, 144
- SetEnableSerial, 144
- SetEnableSetCountersOnBunchReset, 145
- SetEnableTTLClock, 145
- SetEnableTTLControl, 145
- SetEnableTTLHit, 146
- SetEnableTTLReset, 146
- SetEnableTTLSerial, 146
- SetEventCountOffset, 147
- SetIOClockDelay, 147
- SetIOClockSource, 147
- SetKeepToken, 148
- SetLeadingMode, 148
- SetLowPowerMode, 148
- SetMaster, 149
- SetMatchWindow, 149
- SetMaxEventSize, 149
- SetModeRC, 150
- SetModeRCCompression, 150
- SetPLLControl, 150
- SetRCAdjustment, 151
- SetReadoutFIFOSize, 151
- SetReadoutSingleCycleSpeed, 151
- SetReadoutSpeedSelect, 152
- SetRejectCountOffset, 152
- SetRejectFIFOFull, 152
- SetRollOver, 153
- SetSearchWindow, 153
- SetSerialClockDelay, 153
- SetSerialClockSource, 154
- SetSerialDelay, 154
- SetSetupParity, 154
- SetStrobeSelect, 155
- SetTDCId, 155
- SetTest, 155
- SetTestInvert, 156
- SetTestMode, 156
- SetTokenDelay, 156
- SetTrailingMode, 157
- SetTriggerCountOffset, 157
- SetTriggerMatchingMode, 157
- SetVernierOffset, 158
- SetWidthResolution, 158
- SetupParityError, 118
- TDCSetup, 119, 120
- TriggerFIFOParityError, 117
- TriggerMatchingError, 118
- VernierError, 117
- W\_100ns, 119
- W\_100ps, 119
- W\_12p5ns, 119
- W\_1p6ns, 119
- W\_200ns, 119
- W\_200ps, 119
- W\_25ns, 119
- W\_3p2ns, 119
- W\_400ns, 119
- W\_400ps, 119
- W\_50ns, 119
- W\_6p25ns, 119
- W\_800ns, 119
- W\_800ps, 119
- WidthResolution, 119
- TDCStatus, 161
  - DLLLock, 164
  - Dump, 164
  - Error, 164
  - FIFOEmpty, 165
  - FIFOFull, 165
  - FIFOOccupancy, 165
  - HaveToken, 166
  - kDLLLock, 167
  - kError, 167
  - kHaveToken, 167
  - kL1Occupancy, 167
  - kReadoutFIFOEmpty, 167
  - kReadoutFIFOFull, 167
  - kReadoutFIFOOccupancy, 168
  - kTriggerFIFOEmpty, 168
  - kTriggerFIFOFull, 168
  - kTriggerFIFOOccupancy, 168
  - L1Occupancy, 166
  - SetConstantValues, 166
  - TDCStatus, 163
  - TriggerFIFOEmpty, 166
  - TriggerFIFOFull, 167
  - TriggerFIFOOccupancy, 167
- TDCTrailer
  - TDCEvent, 97
- TRAILLEAD
  - DAQ::TDC, 81
- TRIG\_MATCH
  - HPTDC chip control, 11
- tdc\_acq\_mode
  - OnlineDBHandler::TDCConditions, 86
- tdc\_address
  - OnlineDBHandler::TDCConditions, 86
- tdc\_det\_mode
  - OnlineDBHandler::TDCConditions, 86
- tdc\_id
  - OnlineDBHandler::TDCConditions, 86
- time\_start
  - OnlineDBHandler::BurstInfo, 15
- Trigger
  - TDCEvent, 97
- TriggerFIFOEmpty
  - TDCStatus, 166
- TriggerFIFOFull
  - TDCStatus, 167

- TriggerFIFOOccupancy
  - TDCStatus, [167](#)
- TriggerFIFOParityError
  - TDCSetup, [117](#)
- TriggerMatchingError
  - TDCSetup, [118](#)
- UpdatedPlot
  - DQM::DQMProcess, [24](#)
- VernierError
  - TDCSetup, [117](#)
- W\_100ns
  - TDCSetup, [119](#)
- W\_100ps
  - TDCSetup, [119](#)
- W\_12p5ns
  - TDCSetup, [119](#)
- W\_1p6ns
  - TDCSetup, [119](#)
- W\_200ns
  - TDCSetup, [119](#)
- W\_200ps
  - TDCSetup, [119](#)
- W\_25ns
  - TDCSetup, [119](#)
- W\_3p2ns
  - TDCSetup, [119](#)
- W\_400ns
  - TDCSetup, [119](#)
- W\_400ps
  - TDCSetup, [119](#)
- W\_50ns
  - TDCSetup, [119](#)
- W\_6p25ns
  - TDCSetup, [119](#)
- W\_800ns
  - TDCSetup, [119](#)
- W\_800ps
  - TDCSetup, [119](#)
- WEBSOCKET\_CLIENT
  - Socket, [67](#)
- WidthResolution
  - TDCSetup, [119](#)
- word\_t
  - TDCRegister, [107](#)
- Write
  - DAQ::QuickUSBHandler, [65](#)
- WriteRegister
  - DAQ::TDC, [82](#)
- x
  - DQM::GastofCanvas::Coord, [21](#)
  - DQM::QuarticCanvas::Coord, [22](#)
- y
  - DQM::GastofCanvas::Coord, [22](#)
  - DQM::QuarticCanvas::Coord, [22](#)