

Reference Manual

Generated by Doxygen 1.6.1

Fri Aug 14 11:55:34 2015

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	FedException::exBase Class Reference	3
2.2	FedAMC13::FedAMC Class Reference	4
2.2.1	Detailed Description	4
2.3	FedAMC13::FedBlock Class Reference	5
2.3.1	Detailed Description	5
2.3.2	Member Function Documentation	5
2.3.2.1	GetAMCInfo	5
2.3.2.2	GetAMCPtr	6
2.3.2.3	GetMaskedHeader	6
2.3.2.4	Parse	6
2.4	FedAMC13::FedEvent Class Reference	7
2.4.1	Detailed Description	8
2.4.2	Member Function Documentation	8
2.4.2.1	Dump	8
2.4.2.2	ParseEvent	8
2.4.2.3	PreParse	8

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

FedException::exBase	3
FedAMC13::FedAMC (Class to access AMC payloads within an AMC13 event)	4
FedAMC13::FedBlock (Class to access blocks within an AMC13 event)	5
FedAMC13::FedEvent (Class to access items within AMC13 Data with minimal unpacking) . .	7

Chapter 2

Class Documentation

2.1 FedException::exBase Class Reference

Public Member Functions

- const char * **StackTrace** () const throw ()
- void **Append** (const char *buffer) throw ()
- void **Append** (std::string str)
- const char * **Description** () const throw ()
- virtual const char * **what** () const =0 throw ()
- pid_t **GetPID** ()

Protected Member Functions

- void **Copy** (const exBase &rh) throw ()

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

- FedException.hh
- FedException.cc
- FedException_StackTrace.cc

2.2 FedAMC13::FedAMC Class Reference

Class to access AMC payloads within an AMC13 event.

```
#include <FedAMC.hh>
```

Public Member Functions

- void **Clear** ()
Clears AMC pointer and size vectors.
- void **AddPayload** (uint64_t *ptr, uint64_t size)
Pushes back pointer and size onto AMCData and AMCSize vectors, respectively.
- uint64_t **GetWord** (size_t iWord)
Returns the 64-bit word of the AMC payload.
- uint64_t **GetSize** ()
Returns the size of the AMC payload.
- uint32_t **AMCNo** ()
- uint32_t **EvN** ()
- uint32_t **BcN** ()
- uint32_t **OrN** ()
- uint32_t **DataLength** ()
- uint32_t **BoardID** ()
- uint32_t **CRC** ()
- uint32_t **EvN_TR** ()
- uint32_t **DataLength_TR** ()

2.2.1 Detailed Description

Class to access AMC payloads within an AMC13 event.

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

- FedAMC.hh
- FedAMC.cc

2.3 FedAMC13::FedBlock Class Reference

Class to access blocks within an AMC13 event.

```
#include <FedBlock.hh>
```

Public Member Functions

- void **Clear** ()
- uint64_t * **Parse** (uint64_t *ptr, uint64_t size)
- uint64_t * **GetAMCInfo** (int iAMC)
- uint32_t **GetMaskedHeader** (int64_t iAMCword, int bit, int mask)
- uint32_t **GetNAMC** ()
- uint32_t **GetOrN** ()
- uint32_t **GetuFOV** ()
- uint32_t **GetBlockNoTR** ()
- uint32_t **GetEvNTR** ()
- uint32_t **GetBcNTR** ()
- uint32_t **GetMS** (int iAMC)
- uint32_t **GetLMSEPVC** (int iAMC)
- uint32_t **GetAMCSize** (int iAMC)
- uint32_t **GetID** (int iAMC)
- uint64_t **GetBlockNo** (int iAMC)
- uint64_t **GetSize** ()

Used in Parse to compute the number of words to the next block.

- uint64_t * **GetAMCPtr** (int iAMC)
- uint64_t **BlockAMCSize** (int iAMC)

Outputs the actual AMCSize within the block.

- uint64_t **GetEventSize** ()

Returns the size of the event calculated by Mr. Wu's formula.

2.3.1 Detailed Description

Class to access blocks within an AMC13 event.

2.3.2 Member Function Documentation

2.3.2.1 uint64_t * FedAMC13::FedBlock::GetAMCInfo (int iAMC)

Returns a pointer to an AMC header in a block (will return null if the AMC is not active).

Parameters:

iAMC AMC number (1-based)

2.3.2.2 `uint64_t * FedAMC13::FedBlock::GetAMCPtr (int iAMC)`

Returns a pointer to the beginning of an AMC payload. Must be called with {[GetAMCInfo\(\)](#)}.

Parameters:

iAMC AMC number (1-based)

2.3.2.3 `uint32_t FedAMC13::FedBlock::GetMaskedHeader (int64_t iAMCword, int bit, int mask)`

Returns value in the block header by AMC number. Will throw if trying to access inactive AMC.

Parameters:

iAMCword 0 for first word in block header. 1-12 to access AMC headers.

bit Bit location of value in 64-bit word.

mask Mask of value.

2.3.2.4 `uint64_t * FedAMC13::FedBlock::Parse (uint64_t * ptr, uint64_t size)`

Parse returns a pointer to the next block given the beginning of a block and sets private variables `m_ptr` and `sizeLeft`.

Parameters:

ptr pointer to the beginning of a block

size number of 64-bit words left in the total event to prevent reading off the end of an event

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

- FedBlock.hh
- FedBlock.cc

2.4 FedAMC13::FedEvent Class Reference

Class to access items within AMC13 Data with minimal unpacking.

```
#include <FedEvent.hh>
```

Public Member Functions

- void [Clear](#) ()
Clear all values from [FedEvent](#) object.
- uint16_t [CalcCRC](#) ()
Calculate CMS CRC.
- uint16_t [CRC](#) ()
Get CRC from CDF trailer.
- uint64_t [PreParse](#) (uint64_t *ptr, uint64_t size)
- void [ParseEvent](#) (uint64_t *curPtr, uint64_t size)
- void [Dump](#) (int dumpLevel, bool CRCLevel, int errorLevel)
- [FedAMC13::FedBlock](#) [GetBlock](#) (uint32_t iBlock)
Function to access [FedBlock](#) objects.
- size_t [GetnBlocks](#) ()
Returns the number of blocks in a [FedEvent](#).
- [FedAMC13::FedAMC](#) [GetAMC](#) (int iAMC)
Function to access [FedAMC](#) objects.
- uint64_t [GetMaskedField](#) (uint64_t word, uint32_t bit, uint32_t mask)
- size_t [GetBOE](#) ()
- size_t [GetEventType](#) ()
- size_t [GetNAMC](#) ()
- size_t [GetEvN](#) ()
- size_t [GetBcN](#) ()
- size_t [GetOrN](#) ()
- size_t [GetBXID](#) ()
- size_t [GetAMCSize](#) (int n)
- size_t [GetEventSize](#) ()
Returns calculated event size by Mr. Wu's formula.
- void [ErrorChecking](#) ()
Function to create error array.
- uint32_t [GetAMC13Errors](#) (uint32_t AMC, uint32_t error, uint32_t value)
Function to get elements from the error array.

2.4.1 Detailed Description

Class to access items within AMC13 Data with minimal unpacking. First PreParse must be called with a pointer to the raw event and the number of 64-bit words safe to read(two in the first call) two times to read an event. Parse can then be called with a pointer to the raw event and its size in 64-bit words. AMC objects are created to easily access any of the AMC payloads. Accessor functions are provided to the various parts of the data payload.

See documentation for data format: <http://bucms.bu.edu/twiki/bin/view/BUCMSPublic/AMC13CommonFir>

Error-checking functions is provided.

2.4.2 Member Function Documentation

2.4.2.1 void FedAMC13::FedEvent::Dump (int *dumpLevel*, bool *CRCLevel*, int *errorLevel*)

Outputs summary of events onto the screen with various levels of detail.

Parameters:

dumpLevel input to determine how much to print, ranging from 0-4 (default 0, or least amount)

CRCLevel input to turn CRC calculation on/off, false = off, true = on

errorLevel input to turn error checking on/off, off = default, on = 1

2.4.2.2 void FedAMC13::FedEvent::ParseEvent (uint64_t * *curPtr*, uint64_t *size*)

Sets up pointers to each AMC payload for fast access.

Parameters:

curPtr pointer to the header of a raw AMC event

size size of event to be parsed

2.4.2.3 uint64_t FedAMC13::FedEvent::PreParse (uint64_t * *ptr*, uint64_t *size*)

Calculates the number of words should be read next from the file plus some error checking.

Parameters:

ptr pointer to the header of a raw AMC event

size number of 64-bit words allowed to be read from the file

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

- FedEvent.hh
- FedEvent.cc

Index

Dump

FedAMC13::FedEvent, [8](#)

FedAMC13::FedAMC, [4](#)

FedAMC13::FedBlock, [5](#)

GetAMCInfo, [5](#)

GetAMCPtr, [5](#)

GetMaskedHeader, [6](#)

Parse, [6](#)

FedAMC13::FedEvent, [7](#)

Dump, [8](#)

ParseEvent, [8](#)

PreParse, [8](#)

FedException::exBase, [3](#)

GetAMCInfo

FedAMC13::FedBlock, [5](#)

GetAMCPtr

FedAMC13::FedBlock, [5](#)

GetMaskedHeader

FedAMC13::FedBlock, [6](#)

Parse

FedAMC13::FedBlock, [6](#)

ParseEvent

FedAMC13::FedEvent, [8](#)

PreParse

FedAMC13::FedEvent, [8](#)