

ATLAS Event Monitoring System C++ Framework Reference Manual
User's Guide (C++)

Generated by Doxygen 1.3.5

Sat Apr 29 10:55:49 2006

Contents

1	ATLAS Event Monitoring System C++ Framework Class Index	Event Monitoring C++ Framework Class Index
----------	--	---

Chapter 1

ATLAS Event Monitoring System C++ Framework Class Index

1.1 ATLAS Event Monitoring System C++ Framework Class List

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

[emon::Event](#) .ent

Chapter 2

ATLAS Event Monitoring System C++ Framework Class Documentation

2.1 `emon::Event` Class Reference

A class representing an event.

Public Member Functions

- `const unsigned long data() const`

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

- EventChannel.h

Returns:

2.4.3 Member Function Documentation

2.4.3.1 `void emon::EventSampler::stop ()` [inline]

Stops the event sampling process.

Examples:

[PullSamplerMain.cc](#), and [PushSamplerMain.cc](#).

2.4.3.2 `void emon::EventSampler::wait ()` [inline]

Blocks the current thread until the stop method is called.

Examples:

[PullSamplerMain.cc](#), and [PushSamplerMain.cc](#).

The documentation for this class was generated by Doxygen 1.8.13 for version 2.11.1 of the library.

2.5 `emon::MaskedValue` Struct Reference

A class representing a masked value.

Public Member Functions

- `MaskedValue` (long value, bool ignore=false)
Constructor.
- `MaskedValue` (const EventMonitoring::MaskedValue &mv)
Copy constructor.

2.5.1 Detailed Description

A class representing a masked value.

This class represents a masked value. It consists of a long value specifying the value of the selection criteria value and a boolean flag telling whether this value shall1 3.48]TJ 0 -e630slue338Td[(This5-209(class1 3inheth1(itlass1 310(th

2.6 `emon::PullSamplingFactory` Struct Reference

Abstract base class for user's custom pull factory.

Public Member Functions

2.7 `emon::PushSamplingFactory` Struct Reference

Abstract base class for user's custom push factory.

Public Member Functions

- virtual `PushSamplingFactory` ()
Virtual destructor.
- virtual `PushSampling` `startSampling` (const `SamplingAddress` &sa, const `SelectionCriteria` &sc, `EventChannel` channel)q81 9rest

Exceptions:*emon::BadAddress**emon::BadCriteria**emon::NoResources*

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

- PushSamplingFactory.h

2.8 `emon::SamplingAddress` Struct Reference

2.8.3 Member Function Documentation

2.8.3.1 `void emon::SamplingAddress::addComponent (const std::string & key, const std::string & value)`

Parameters:

detector_type long value specifying the type of the detector
lvl1_trigger_type long value specifying the type of the level1 trigger
lvl2_trigger_info long value specifying information for level2 trigger
status_word long value specifying status word in event header
statistics optional parameter, specifying how many events to skip between two events that shall be sampled. a value of x means: sample every xth event

2.9.2.3 `emon::SelectionCriteria::SelectionCriteria (const EventMonitoring::SelectionCriteria & sc`**Parameters:**

sc the selection criteria to copy

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

- Common.h

Chapter 3

3.2 PullSamplerMain.cc


```

CmdArgvIter arg_iter(--argc, ++argv);

cmd.description("pjll model example event sampler");

    // default values for this Event Sampler...
event_size = 1024;
max_channels = 10;
fragment_number = 1;

cmd.parse(arg_iter);

signal(SIGINT, signal_han_ler);
OWLTimer timer;
IPCPartition partition( partition_name signal(SIGINT, signal_long size = fragment_number;
iovec * event = new iovec[size]al(SIGINT, signal_if sig0 event_file.flagssig0sign & CmdArg::GIVEN s
{
    OWreturn600(1;))TJ 0 9.128 -9.464 Td((ev)050SIGINT, signal_ifng)-600(sifl)-ize = 10 = de
    int fd = og0 eve PjllSamp-600(++O_RDONLY600(signal(SIGIN464 Td({))-600(sig0)-600(ev)-600(=)--6

```



```

        {
            event_ = event;
            size_ = size;
        }
    }

    emon::PushSampling * startSampling( const emon::SamplingAddress & ,
        const emon::SelectionCriteria & criteria,
        emon::EventChannel * channel ) throw (emon::BadAddress, emon::BadCriteria, emon::NoRes
{hannel )1,
{hannel ) 00(*)-6ushSamnnel t;
{hannel } =
{hannel 9.464 IPCCore::init600(00(*)-argcer0(*)-argvnnel)-600(;))TJ 0 -5.464 CmdArgStmlingsie

```


