

artdaq_epics_plugin
1.06.00

Generated by Doxygen 1.8.5

Thu Sep 5 2024 11:02:33

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	artdaq Namespace Reference	7
4.1.1	Detailed Description	7
5	Class Documentation	9
5.1	artdaq::EpicsMetric Class Reference	9
5.1.1	Detailed Description	10
5.1.2	Constructor & Destructor Documentation	10
5.1.2.1	EpicsMetric	10
5.1.3	Member Function Documentation	10
5.1.3.1	getLibName	10
5.1.3.2	sendMetric_	10
5.1.3.3	sendMetric_	11
5.1.3.4	sendMetric_	12
5.1.3.5	sendMetric_	12
5.1.3.6	sendMetric_	12
	Index	14

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

artdaq	The artdaq namespace	7
------------------------	--------------------------------	-------------------

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

MetricPlugin	
artdaq::EpicsMetric	9

Chapter 3

Class Index

3.1 Class List

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

[artdaq::EpicsMetric](#)

An instance of the MetricPlugin class that sends metric data using the Channel Access protocol from
EPICS

Chapter 4

Namespace Documentation

4.1 artdaq Namespace Reference

The artdaq namespace.

Classes

- class [EpicsMetric](#)

An instance of the MetricPlugin class that sends metric data using the Channel Access protocol from EPICS.

4.1.1 Detailed Description

The artdaq namespace.

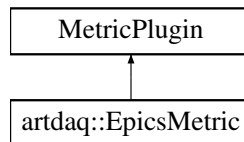
Chapter 5

Class Documentation

5.1 artdaq::EpicsMetric Class Reference

An instance of the MetricPlugin class that sends metric data using the Channel Access protocol from EPICS.

Inheritance diagram for artdaq::EpicsMetric:



Public Member Functions

- [EpicsMetric](#) (fhicl::ParameterSet const &pset, std::string const &app_name, std::string const &metric_name)
Construct an instance of the [EpicsMetric](#) plugin.
- std::string [getLibName](#) () const override
Gets the unique library name of this plugin.
- void [stopMetrics_](#) () override
Clears the registered ChannelAccess channels.
- void [startMetrics_](#) () override
No initialization is needed to start sending metrics.
- void [sendMetric_](#) (const std::string &name, const std::string &value, const std::string &unit, const std::chrono::system_clock::time_point &) override
Send a string metric data point to ChannelAccess.
- void [sendMetric_](#) (const std::string &name, const int &value, const std::string &unit, const std::chrono::system_clock::time_point &) override
Send an integer metric data point to ChannelAccess.
- void [sendMetric_](#) (const std::string &name, const double &value, const std::string &unit, const std::chrono::system_clock::time_point &) override
Send a double metric data point to ChannelAccess.
- void [sendMetric_](#) (const std::string &name, const float &value, const std::string &unit, const std::chrono::system_clock::time_point &) override
Send a float metric data point to ChannelAccess.

- void [sendMetric_](#) (const std::string &name, const uint64_t &value, const std::string &unit, const std::chrono::system_clock::time_point &) override

Send an unsigned integer metric data point to ChannelAccess.

5.1.1 Detailed Description

An instance of the MetricPlugin class that sends metric data using the Channel Access protocol from EPICS.

Definition at line 40 of file epics_metric.cc.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 `artdaq::EpicsMetric::EpicsMetric (fhicl::ParameterSet const & pset, std::string const & app_name, std::string const & metric_name) [inline], [explicit]`

Construct an instance of the [EpicsMetric](#) plugin.

Parameters

<i>pset</i>	Parameter set to configure with. MetricPlugin parameters plus "channel_name_prefix", default "artdaq".
<i>app_name</i>	Name of the application sending metrics
<i>metric_name</i>	Name of this metric instance

Definition at line 105 of file epics_metric.cc.

5.1.3 Member Function Documentation

5.1.3.1 `std::string artdaq::EpicsMetric::getLibName () const [inline], [override]`

Gets the unique library name of this plugin.

Returns

The library name of this plugin, "epics".

Definition at line 118 of file epics_metric.cc.

5.1.3.2 `void artdaq::EpicsMetric::sendMetric_ (const std::string & name, const std::string & value, const std::string & unit, const std::chrono::system_clock::time_point &) [inline], [override]`

Send a string metric data point to ChannelAccess.

Parameters

<i>name</i>	Name of the metric
<i>value</i>	Value of the metric
<i>unit</i>	Units used (not really relevant for string metrics)

Send a string metric data point to ChannelAccess. The name will be channel_name_prefix:name. If the named channel is not yet open, it will be opened. If the channel is not registered with an IOC, then the metric data will not be sent and a warning message will be printed the first time.

Definition at line 155 of file epics_metric.cc.

```
5.1.3.3 void artdaq::EpicsMetric::sendMetric_( const std::string & name, const int & value, const std::string & unit, const  
std::chrono::system_clock::time_point & ) [inline],[override]
```

Send an integer metric data point to ChannelAccess.

Parameters

<i>name</i>	Name of the metric
<i>value</i>	Value of the metric
<i>unit</i>	Units used

Send a string metric data point to ChannelAccess. The name will be channel_name_prefix:name. If the named channel is not yet open, it will be opened. If the channel is not registered with an IOC, then the metric data will not be sent and a warning message will be printed the first time.

Definition at line 185 of file epics_metric.cc.

```
5.1.3.4 void artdaq::EpicsMetric::sendMetric_ ( const std::string & name, const double & value, const std::string & unit, const
std::chrono::system_clock::time_point & ) [inline],[override]
```

Send a double metric data point to ChannelAccess.

Parameters

<i>name</i>	Name of the metric
<i>value</i>	Value of the metric
<i>unit</i>	Units used

Send a string metric data point to ChannelAccess. The name will be channel_name_prefix:name. If the named channel is not yet open, it will be opened. If the channel is not registered with an IOC, then the metric data will not be sent and a warning message will be printed the first time.

Definition at line 215 of file epics_metric.cc.

```
5.1.3.5 void artdaq::EpicsMetric::sendMetric_ ( const std::string & name, const float & value, const std::string & unit, const
std::chrono::system_clock::time_point & ) [inline],[override]
```

Send a float metric data point to ChannelAccess.

Parameters

<i>name</i>	Name of the metric
<i>value</i>	Value of the metric
<i>unit</i>	Units used

Send a string metric data point to ChannelAccess. The name will be channel_name_prefix:name. If the named channel is not yet open, it will be opened. If the channel is not registered with an IOC, then the metric data will not be sent and a warning message will be printed the first time.

Definition at line 245 of file epics_metric.cc.

```
5.1.3.6 void artdaq::EpicsMetric::sendMetric_ ( const std::string & name, const uint64_t & value, const std::string & unit, const
std::chrono::system_clock::time_point & ) [inline],[override]
```

Send an unsigned integer metric data point to ChannelAccess.

Parameters

<i>name</i>	Name of the metric
-------------	--------------------

<i>value</i>	Value of the metric. Will be truncated to fit in the size of a <code>dbr_ulong_t</code> , a 32-bit unsigned integer.
<i>unit</i>	Units used

Send a string metric data point to ChannelAccess. The name will be `channel_name_prefix:name`. If the named channel is not yet open, it will be opened. If the channel is not registered with an IOC, then the metric data will not be sent and a warning message will be printed the first time.

Definition at line 275 of file `epics_metric.cc`.

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

- `artdaq_epics_plugin/artdaq-epics-plugin/MetricPlugins/epics_metric.cc`

Index

artdaq, [7](#)

artdaq::EpicsMetric, [9](#)

EpicsMetric, [10](#)

getLibName, [10](#)

sendMetric_, [10](#), [12](#)

EpicsMetric

artdaq::EpicsMetric, [10](#)

getLibName

artdaq::EpicsMetric, [10](#)

sendMetric_

artdaq::EpicsMetric, [10](#), [12](#)