

NSCLDAQ 11.0 - Built Ring Item Format

Author: Jeromy Tompkins

Date: 9/4/2014

Below is an example of a built event consisting of 2 fragments. Note that this is a big ring item, of type PHYSICS_EVENT, with a body consisting of a body size value and two fragments. The fragments are just the original ring items with fragment headers prepended to them. Note that in 11.0 the body header has been added and may or may not exist. The size of the body header is always present and indicates whether the ring item contains a body header. If the body header size is 20, then there is a body header. If it is not 20, then there is no body header data. Currently, a body header size of 0 indicates that there is no body header, but this may change in the future to be 4 for consistency. At the moment, the body header size of 0 means no body header. The user can save themselves from have to deal with this by converting the data into an actual ring item. One would do that by passing a pointer to a ring item header to the CRingItemFactory::createRingItem method. Some example code demonstrating this is below:

```
void MyUnpacker::handlePhysicsEventItem(CPhysicsEventItem* item) {  
  
    // get the pointer to the built body  
    uint16_t* pBody = reinterpret_cast<uint16_t*>(item->getBodyPointer());  
  
    // assume there exists some function that has the ability to locate  
    // and return the location of the first fragment's payload (i.e. the  
    // header of the ring item it wraps)  
    uint16_t* pFirstFragPayload = findPayloadOfFirstFragment(pBody);  
  
    // Convert to a CRingItem object.  
    CRingItem* frag0Item =  
        CRingItemFactory::createRingItem(pFirstFragPayload);  
  
    // get a pointer to this item's body  
    pBody = reinterpret_cast<uint16_t*>(frag0Item->getBodyPointer());  
}
```

Also see the DataFormat.h and fragment.h file in the /usr/opt/nscldaq/11.0/include directory.

	Component	Type of data	Bytes/Shorts associated	Description				
Header of built ring item	RingItemHeader	UInt32_t	4 / 2	Size of entire item (in bytes)				
		UInt32_t	4 / 2	Item type = PHYSICS_EVENT				
Body Header of built	BodyHeader	UInt32_t	4/2	Total size of Body Header (in bytes)				
	(optional)	UInt64_t	8/4	Timestamp				
	(optional)	UInt32_t	4/2	Source Id				
	(optional)	UInt32_t	4/2	Barrier				
Body of built ring item	Size of event	UInt32_t	4 / 2	Total size of body (in bytes)			First Fragment	First raw ring item
	Fragment Header	Unit64_t	8 / 4	Timestamp				
		UInt32_t	4 / 2	Source id				
		UInt32_t	4 / 2	Fragment payload size (in bytes)				
		UInt32_t	4 / 2	Barrier				
	RingItem Header	UInt32_t	4 / 2	Total size of ring item (in bytes)				
		UInt32_t	4 / 2	Item type = PHYSICS_EVENT				
	BodyHeader	UInt32_t	4/2	Total size of Body header (in bytes)				
		UInt64_t	8/4	(optional) Timestamp				
		UInt32_t	4/2	(optional) Source Id				
		UInt32_t	4/2	(optional) Barrier				
	RingItem Body	Data				
				

Item	Fragment Header	Unit64_t	8 / 4	Timestamp	Second Fragment				
		Uint32_t	4 / 2	Source id					
		Uint32_t	4 / 2	Fragment payload size in bytes					
		Uint32_t	4 / 2	Barrier					
	RingItem Header	Uint32_t	4 / 2	Total size of ring item (in bytes)		Second Fragment			
		Uint32_t	4 / 2	Item type = PHYSICS_EVENT					
	Body Header	Uint32_t	4 / 2	Total body header size (in bytes)			Second Fragment	Second raw ring item	
		Unit64_t	8 / 4	(optional) Timestamp					
		Uint32_t	4 / 2	(optional) Source Id					
		Uint32_t	4 / 2	(optional) Barrier					
	RingItem Body	Data				Second Fragment	
					