IO interface (all within some namespace)

structs	description
• CDMSEVENT	• Contains TRIGGERINFO, vector<TRIGPRIMITIVE>, vector<DETECTOR>
• READOUT_REQUEST	• Simple struct (trig ID, type, timestamp) (needs head…)
• TRIGPRIMITIVE	• Simple struct (status, detector, time, etc etc)
• DETECTOR	• Contains TRIGINFO, vector<WAVEFORM> + detnum, serial, etc
• TRIGINFO	• Simple struct: time_sec, time_100ns, status
• WAVEFORM	• Contains pointer to sample array (not copied!) plus headers (nsamps, sample rate, etc

IO interface (all within some namespace)

structs

- CDMSEVENT
- READOUT_REQUEST
- TRIGPRIMITIVE
- DETECTOR
- TRIGINFO
- WAVEFORM

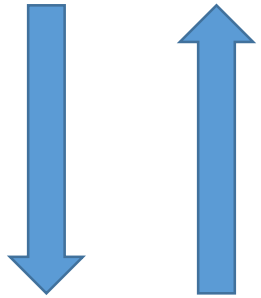
To/from memory buffer

- (un)pack_event(char* b, CDMSEVENT* event)
  - (un)pack_eventlist(char* b, vector<CDMSEVENT>* evtlist)
- (un)pack_readout_request(char* b, READOUT_REQUST*r)
- (un)pack_primitive(char* b, TRIGPRIMITIVE* prim)
  - (un)pack_primitivelist(char* b, vector<TRIGPRIMITIVE>* primlist)
- (un)pack_detector(char* b, DETECTOR* det)
- (un)pack_triginfo(char* b, TRIGINFO*r)
- (un)pack_waveform(char* b, WAVEFORM* wvfm)

```
struct WAVEFORM {
    DWORD prepulseLength;
    DWORD onpulseLength;
    DWORD postpulseLength;
    DWORD pretriggerOffset;
    WORD samplerateHigh;
    WORD samplerateLow;
    BYTE channelType;
    BYTE channelNum;
    WORD* data; //<points to array in mem managed by someone else
};
```

pack,unpack return status

```
int pack_waveform(char* buffer, WAVEFORM* wvfm)
```



```
int unpack_waveform(char* buffer, WAVEFORM* wvfm)
```

0x1	pre-trigger offset (22 bits)	ch num	ch type
n pre-pulse samples			
n on-pulse samples			
n post-pulse samples			
sampling rate high in kHz		sampling rate low in kHz	
samp1		samp0	
samp3		samp2	
⋮			
sampN		sampN-1	

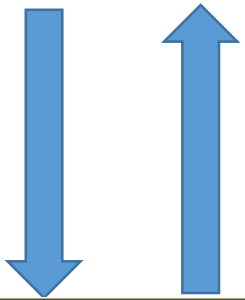
```
struct DETECTOR {
    DWORD detectorType;
    BYTE detectorID;
    BYTE detectorIndex;
    BYTE dcrc0_serial;
    BYTE dcrc1_serial;
    BYTE dcrc0_version;
    BYTE dcrc1_version;

    READOUT_REQUEST triginfo;
    std::vector<WAVEFORM> channels;
};
```

pack,unpack return status

```
int pack_detector(char* buffer, DETECTOR* det)
```

```
int unpack_detector(char* buffer, DETECTOR* det)
```



	0x2	detector type		detector id	index
	DCRC1 serial number	DCRC1 version	DCRC0 serial number	DCRC0 version	
	0x4	readout status	series time in sec		
		series time fraction (100nsec/count)			
	0x0	n channels to follow			
x N channels	0x1	pre-trigger offset (22 bits)		ch num	ch type
	n pre-pulse samples				
	n on-pulse samples				
	n post-pulse samples				
	sampling rate high in kHz		sampling rate low in kHz		
	samp1		samp0		
	samp3		samp2		
	⋮				
	sampN		sampN-1		

IO interface (all within some namespace) other necessaries

- Calculate size of buffer to allocate for packing
- Error handling for receiving data of wrong size
- Determine type of block at memory pointer
- Getters, setters, utility functions for all of these structs (e.g. total_nsamps() {return npresamps+npostsamps; }
- Functionality to optionally copy sample arrays with WAVEFORMs
  - In case buffer gets clobbered underneath you