



# Emulator software code review

Kalanand Mishra, Fermilab

*L1 Calo Stage1 Upgrade Meeting*  
(March 19, 2014)

# Status of the code implementation

[GitHub:cms-l1t-offline/cmssw/](https://github.com/cms-l1t-offline/cmssw/)

	Data format	Algorithm	Isolation
Jets	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/A
SumET	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	N/A
Egamma	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Tau	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Region calibration	N/A	<input type="checkbox"/>	N/A



# Data format: **DataFormats/L1Trigger/interface/L1Candidate.h**

```
class L1Candidate : public reco::LeafCandidate {
```

```
public:  
    L1Candidate();
```

**// construct from \*both\* physical and integer values**

```
L1Candidate( const LorentzVector& p4,  
             int pt=0, int eta=0, int phi=0,  
             int qual=0, int iso=0);
```

```
L1Candidate( const PolarLorentzVector& p4,  
             int pt=0, int eta=0, int phi=0,  
             int qual=0, int iso=0);
```

```
~L1Candidate();
```

**// methods to set integer values  
// in general, these should not be needed**

```
void setHwPt(int pt);  
void setHwEta(int eta);  
void setHwPhi(int phi);  
void setHwQual(int qual);  
void setHwIso(int iso);
```

```
// methods to retrieve integer values
```

```
int hwPt() const;  
int hwEta() const;  
int hwPhi() const;  
int hwQual() const;  
int hwIso() const;
```

```
private:
```

```
// integer "hardware" values
```

```
int hwPt_;  
int hwEta_;  
int hwPhi_;  
int hwQual_;  
int hwIso_;
```

```
};
```

**Inherited by:**

**EGamma.h**

**Tau.h**

**Jet.h**

**EtSum.h**

**Question: How to set precision in emulator to match the firmware bit information ?**



# Algorithm

[GitHub:cms-l1t-offline/cmssw/L1Trigger/L1TCalorimeter](https://github.com/cms-l1t-offline/cmssw/L1Trigger/L1TCalorimeter)

	Algorithm	Class name
Jets	12 x 12 towers	Stage1Layer2JetAlgorithm
SumET	Summing all towers within specified $\eta$	Stage1Layer2EtSumAlgorithm
Egamma	2 x 1 em clusters	Stage1Layer2EGammaAlgorithm
Tau	8 x 4 towers	Stage1Layer2TauAlgorithm

**Caveat: success metric = “code exists and compiles”**

Loose ends to be tightened during the run/validation step