

Update on Offline Pixel Calibration

Wei-Ming Yao (LBNL)

LBNL ID Software Meeting, 11/21/2006

Progresses since last meeting:

- PixelCalibration package in InnerDetector/InDetConditions:
 - PixelCalibSvc – Providing service to access the calibration constants
 - * PixelCalibSvc
 - * PixelCalibDbSvc
 - PixelCalibTool – Providing tool to create PixelCalib TDS via call back
 - * PixelCalibDbTool
 - * PixelCalibDbTestWriteRead
 - PixelCalibData – Pixel Calibration Data Model
- Code can be found at </afs/cern.ch/user/w/weiming/public/InDetConditions/PixelCalibration>, and will be committed to CVS shortly.



Project Overview

- Calibration infrastructure
 - Calibration Data (Threshold, ToT) – Weiming Yao (LBNL)
 - Dead Channel Map – Christoph Ruwiedel (Bonn)
- Calibration Content – Georges Aad, Remy Zaidan (Marseille)
 - Online Calibration at SR1
 - Production Database
- Better parametrization of ToT calibration – Serena Psoroulas, Attilio (Milano)
- Implementation in Digitization – Fredrik Tegenfeldt (Iowa)
- Implementation in Clusterization – Tommaso Lari (Fisica)



Technology

- Using LCG COOL conditions database software
 - For calibration data, store data both directly in COOL as CLOB's, and in C++ objects stored in POOL files, referenced from COOL.
 - COOL tags can be used to identify multiple versions of a calibration
- Calibrations can be written from Athena jobs, read back, and verified.
- Automatic loading of valid conditions data on read-back, matching interval of validity to current event being analysed, loading new versions when required.
- Client code can register call-backs to be notified when conditions change.

Pixel Calibration Data Model

Storage	Definition	Unites	Typical range
1B	threshold	30e	2000-5000 e
1B	dispersion	3e	80 - 600 e
1B	noise	3e	0 - 600 e
1B	timewalk	30e	2000-8000 e
float	A for ToT		0-300
float	B for ToT		
float	C for ToT		
1B	P1 dispersion of ToT	1/100	+ -100
1B	P2 dispersion of ToT	1/1000	+ -100

- 38B/per chip, corresponding to 1 MB/Detector
- It took about few sec to read and unpack whole pixel detector into a text file

Pixel Calibration Text File

49152							
I0	3750	200	200	3750	3750	200	200
	3750	3750	200	200	3750	70.2	-2.0752e+06
	26000	70.2	-2.0752e+06	26000	-0.68	0.17	
I1	3750	200	200	3750	3750	200	200
	3750	3750	200	200	3750	70.2	-2.0752e+06
	26000	70.2	-2.0752e+06	26000	-0.68	0.17	
			.	.	.		
I15	3750	200	200	3750	3750	200	200
	3750	3750	200	200	3750	70.2	-2.0752e+06
	26000	70.2	-2.0752e+06	26000	-0.68	0.17	

180224							
I0	3750	200	200	3750	3750	200	200
	3750	3750	200	200	3750	70.2	-2.0752e+06
	26000	70.2	-2.0752e+06	26000	-0.68	0.17	
			.	.	.		
			.	.	.		
			.	.	.		

- There are 1744 Modules for whole Pixel detector
- Stored in folder /TESTCOOL/PixCalib



Object Class in TDS

- PixelChipSummaryData to hold all the data for a chip
 - chipID m_ident; // std::map<Wafer_id, n(0-15) >
 - char m_threshold[4]; // (thres, sigma, Noise, TW):
 - char m_thresholdlong[4]; // (thres, sigma, Noise, TW) for long pixel
 - char m_thresholdganged[4]; // (thres, sigma, Noise, TW) for ganged pixel
 - float m_q2tot[3]; // a,b,c for standard pixel and long
 - float m_q2totganged[3]; // a,b,c for ganged
 - char m_totres[2]; // for (p1, p2) each chip
- PixelCalibData contains 16 chips per module
- Accessories:
 - const chipID& getChipID() const;
 - int getThreshold(int type); //type=0/1/2 for standard, long and ganged pixels
 - int getThresholdSigma(int type), getNoise(int type), getTimeWalk(int type)
 - float getQ2TotA(int type), getQ2TotB(int type), getQ2TotC(int type);
 - float getTot(float Q), getTotP1(), getTotP2(), getTotRes(float Q);
 - void setChipID(const chipID& ident);
 - void setThreshold(int type, int thres, int sigma, int noise, int timewalk);
 - void setQ2Tot(int type, float A, float B, float C);
 - void setTotRes(int p1, int p2);



PixelCalibSvc

- Load and Configure PixelCalibSvc

```
theApp.Dlls += [ "PixelCalibSvc" ]
theApp.ExtSvc =+ [ "PixelCalibSvc" ]
PixelCalibSvc = Service( "PixelCalibSvc" )
```
- Accessories:
 - int GetThreshold(const Identifier& wafer_id, int irow, int icol, int circ);
 - int GetThresholdSigma(const Identifier& wafer_id, int irow, int icol, int circ);
 - int GetNoise(const Identifier& wafer_id, int irow, int icol, int circ);
 - int GetTimeWalkThreshold(const Identifier& wafer_id, int irow, int icol, int circ);
 - float GetQ2TotA(const Identifier& wafer_id, int irow, int icol, int circ);
 - float GetQ2TotB(const Identifier& wafer_id, int irow, int icol, int circ);
 - float GetQ2TotC(const Identifier& wafer_id, int irow, int icol, int circ);
 - float GetTotP1(const Identifier& wafer_id, int circ);
 - float GetTotP2(const Identifier& wafer_id, int circ);
 - float GetTotRes(const Identifier& wafer_id, int circ, float Q);

- int GetThreshold(const Identifier& pix_id);
- int GetThresholdSigma(const Identifier& pix_id);
- int GetNoise(const Identifier& pix_id);
- int GetTimeWalk(const Identifier& pix_id);
- float GetQ2TotA(const Identifier& pix_id);
- float GetQ2TotB(const Identifier& pix_id);
- float GetQ2TotC(const Identifier& pix_id);
- float GetTotP1(const Identifier& pix_id);
- float GetTotP2(const Identifier& pix_id);
- float GetTotRes(const Identifier& pix_id, float Q);
- int PixelType(int row, int col, int circ);
- Private:
 - PixelType(int row, int col, int circ);



PixelCalibDbSvc

- Configure PixelCalibDbSvc

```
theApp.ExtSvc += [ "PixelCalibDbSvc" ]  
PixelCalibDbSvc = Service( "PixelCalibDbSvc" )  
PixelCalibDbSvc.DBToolType = "PixelCalib::PixelCalibDbTool"  
PixelCalibDbSvc.DBToolName = "PixelCalib_PixelCalibDbTool"  
PixelCalibDbSvc.CalibFolder = "/TESTCOOL/PixCalib"  
PixelCalibDbSvc.CalibLocation = "PixCalibKey"
```

- Accessories:

- virtual StatusCode IOVCallback(IOVSVC_CALLBACK_ARGS);
- const PixelCalib::PixelCalibData* getCalibration(const Identifier & id) const;

PixelCalibDbTool

- Load and Configure PixelCalibDbTool

```
theApp.DLLs += [ "PixelCalibTool" ]
```

```
PixelCalib_PixelCalibDbTool = Service( "ToolSvc.PixelCalib_PixelCalibDbTool" )
```

```
PixelCalib_PixelCalibDbTool.CalibFolder = " /TESTCOOL/PixCalib"
```

```
PixelCalib_PixelCalibDbTool.CalibLocation = " PixCalibKey"
```

```
PixelCalib_PixelCalibDbTool.WriteDB = False
```

- Accessories:

```
– virtual StatusCode IOVCallback(IOVSVC_CALLBACK_ARGS);
```

```
– StatusCode createPixelCalibObjects() const;
```

```
– StatusCode writePixelCalibTextFiletoDB(std::string file) const;
```

```
– StatusCode readPixelCalibDBtoTextFile(std::string file) const;
```

```
– void printPixelCalibObjects() const;
```

```
– PixelCalib::PixelCalibData* getCalibPtr(const Identifier& idet) const;
```

```
– const PixelCalib::PixelCalibData* cgetCalibPtr(const Identifier& idet ) const;
```



PixelCalibDbTestWriteRead – an Example

- DataBase Connection

```
IOVDbSvc.dbConnection="impl=cool;techno=oracle;devdb10:TESTCOOL:  
WEIMING:dbconnection00"  
IOVDbSvc.Folders+=["/TESTCOOL/PixCalib"]  
IOVDbSvc.GlobalTag="TESTCOOL_dbtest"
```

- WritePixelCalibDB.py

Write pixelcalibdb_dummy.txt into /TESTCOOL/PixCalib

- ReadPixelCalibDB.py

Read CondAttrListCollection from /TESTCOOL/PixCalib and convert to PixelCalibDataColl (TDS)