

Gaudi

Gaudi-speak

component - refers to a block of software with a well defined interface and purpose. ex.) an algorithm or a service

Algorithm - an algorithm class which is derived from Gaudi's IAlgorithm base class. All algorithms must include: initialize, execute, and finalize methods.

```
class GAUDI_API IAlgorithm : virtual public INamedInterface {
public:    /// InterfaceID DeclareInterfaceID(IAlgorithm,4,0);  /** The version of the algorithm */
virtual StatusCode initialize() = 0;
virtual StatusCode finalize() = 0;

    /** The action to be performed by the algorithm on an event. This method is invoked once per event for top
    level algorithms by the
    application manager. */
    virtual StatusCode execute() = 0;
```

Converter - provides translation between the transient representation to the persistent representation and back again. Converters are part of the persistency service.

Service - a component that is shared among many algorithms, potentially.

Tool - "a light weight object whose purpose is to help other components to perform their work"

Transient Data Store (TDS) a mechanism for shared memory; a tree that stores all data pertinent to a particular execution of the code. Gaudi provides an event data store, which is cleared on a per event basis, and a detector data store, which handles longer lasting data such as calibrations.

Persistency Service - handles I/O through the use of converters.

Fermi's Use of Gaudi

Started in 2000 as part of our upgrade of the simulation and reconstruction software named Gleam, now often referred to as GlastRelease. See this [diagram](#) from our online workbook.

Gaudi Provides Services/Tools Fermi Uses

ChronoSvc - monitors the cpu usage for all algorithms and provides a report at the end of the job

```
****Chrono*** INFO *****
****Chrono**** INFO The Final CPU consumption ( Chrono ) Table (ordered)
****Chrono*** INFO *****
Digitization:ex... INFO Time User : Tot= 406 [s] Ave/Min/Max= 4.06(+/- 13)/ 0/3.8e+03 [ms] #=100000
G4Generator:exe... INFO Time User : Tot= 429 [s] Ave/Min/Max= 4.29(+/- 17.8)/ 0/5.06e+03 [ms] #=100000
EA_Generation INFO Time User : Tot= 446 [s] Ave/Min/Max= 4.46(+/- 17.8)/ 0/5.06e+03 [ms] #=100000
Generation:execute INFO Time User : Tot= 446 [s] Ave/Min/Max= 4.46(+/- 17.8)/ 0/5.06e+03 [ms] #=100000
FirstPass:execute INFO Time User : Tot= 12.2[min] Ave/Min/Max= 68.8(+/- 106)/ 0/1.36e+03 [ms] #=10602
EA_Tkr INFO Time User : Tot= 12.2[min] Ave/Min/Max= 68.8(+/- 106)/ 0/1.36e+03 [ms] #=10602
Tkr:execute INFO Time User : Tot= 12.2[min] Ave/Min/Max= 68.9(+/- 106)/ 0/1.36e+03 [ms] #=10602
EA_Reconstruction INFO Time User : Tot= 13.9[min] Ave/Min/Max= 78.8(+/- 117)/ 0/1.37e+03 [ms] #=10602
Reconstruction:... INFO Time User : Tot= 13.9[min] Ave/Min/Max= 78.8(+/- 117)/ 0/1.37e+03 [ms] #=10602
Triggered:execute INFO Time User : Tot= 14.9[min] Ave/Min/Max= 84.3(+/- 117)/ 0/1.37e+03 [ms] #=10602
EA_Event INFO Time User : Tot= 29.7[min] Ave/Min/Max= 17.8(+/- 59)/ 0/9.78e+03 [ms] #=100000
Event:execute INFO Time User : Tot= 29.7[min] Ave/Min/Max= 17.8(+/- 59)/ 0/9.78e+03 [ms] #=100000
Top:execute INFO Time User : Tot= 29.7[min] Ave/Min/Max= 17.8(+/- 59)/ 0/9.78e+03 [ms] #=100000
ChronoStatSvc INFO Time User : Tot= 30.8[min] #= 1
****Chrono*** INFO *****
ChronoStatSvc.f... INFO Service finalized succesfully
```

JobOptionsSvc - Handles ASCII jobOptions files which allow setting of runtime parameters for all algorithm, services, tools, etc as well as determining the sequence of execution. Allows for conditional execution based on user-defined conditions.

```

ApplicationMgr.DLLs    += {"GaudiAlg", "GaudiAud"};
ApplicationMgr.ExtSvc += {"ChronoStatSvc"};

AuditorSvc.Auditors = {"ChronoAuditor"};

//
// Set up basic event loop:
//
ApplicationMgr.ExtSvc = {"EventSelector/EventSelector" };

EventPersistencySvc.CnvServices = {"EventCnvSvc"};

EventSelector.Input      = "SVC='DbEvtSelector'";
EventSelector.PrintFreq = -1;

//
// A structure for the topalg, using sequencer steps:
//
ApplicationMgr.TopAlg = {"Sequencer/Top"};

//
// Define the top sequence loop:
//
Top.Members = {"Sequencer/Digitization",
               "Sequencer/Calibration",
               "Sequencer/Integrity",
               "Sequencer/NtupleMaker",
               "Sequencer/Output"
              };

//
// Needed for EventIntegrityAlg sequence breaking:
//
Top.StopOverride = true;

//
// Digitization sequence: Read in digis from a ROOT file!
//
Digitization.Members += {"digiRootReaderAlg"};

//
// Need 'Rec' in the name to get the alignment:
//
Calibration.Members = {"TkrCalibAlg/TkrReconCalib"};

//
// Using EventIntegrityAlg to decide if we run or skip recon:
//   If the event passes, proceed as normal.
//   If EventIntegrityAlg is BAD, skip recon but output to file.
//
ApplicationMgr.DLLs += {"EventIntegrity"};

Integrity.Members    = {"EventIntegrityAlg",
                       "Sequencer/Filter",
                       "Sequencer/Reconstruction"};

//
// Detector services:
//
ApplicationMgr.DLLs    += {"GlastSvc"};
ApplicationMgr.ExtSvc += {"GlastDetSvc"};

GlastDetSvc.topVolume  = "LAT";
GlastDetSvc.visitorMode = "recon";

```

MessageSvc - provides logging at prescribed "levels": DEBUG, INFO, WARNING, ERROR

```
log << MSG::DEBUG << "Hello World!" << endreq;
log << MSG::INFO << "Hello World!" << endreq;
log << MSG::WARNING << "Hello World!" << endreq;
```

There are other Gaudi services/tools that we have not made use of, such as the RandomSvc, python interface, etc.

Fermi-defined servies and algorithms

G4Generator - Algorithm that is our interface to Geant4. G4 generates one event, and the resulting data are then posted to the TDS.

Detector Description - stored as XML, and read in via xerces*. * A service provides detailed access to the specifications.

Calibrations - stored on the Detector TDS, where the event time determines the validity window for the calibration data.

ntupleWriterSvc - rather than utilize Gaudi's histogram service we chose to write our own. Over the years it has been extended to handle not only writing our outupt ROOT ntuples, but also provides a mechanism to read in ntuples for reprocessing.

GlastRandomSvc - can't recall why, but we implemented our own random number service that initializes the seeds of the random number generators for all packages that obtain random numbers. We desired to have event by event reproducibility (which we didn't achieve)...rather we can reproduce a run.

Toy Example

```
#include "GaudiKernel/Algorithm.h"
#include "GaudiKernel/Property.h"
#include "GaudiKernel/MsgStream.h"
#include "GaudiKernel/AlgFactory.h"
#include "GaudiKernel/DataObject.h"
#include "GaudiKernel/IDataProviderSvc.h"
#include "GaudiKernel/SmartDataPtr.h"
#include "Event/TopLevel/Event.h"
#include "Event/TopLevel/EventModel.h"
/** @class HelloWorld @brief Simple Hello World example Gaudi algorithm. Prints Hello World with different
priority levels.
$header$
*/
class HelloWorld : public Algorithm {
public:
    /// Constructor of this form must be provided
    HelloWorld(const std::string& name, ISvcLocator* pSvcLocator) : Algorithm(name, pSvcLocator) { };
    /// Three mandatory member functions of any Gaudi algorithm
    StatusCode initialize() { return StatusCode::SUCCESS; };
    StatusCode execute();
    StatusCode finalize() { return StatusCode::SUCCESS; };

private:
};
// Static Factory declaration
static const AlgFactory<HelloWorld> Factory;
const IAlgFactory& HelloWorldFactory = Factory;

StatusCode HelloWorld::execute() {
    MsgStream          log( msgSvc(), name() );
    static int eventCounter = 0;

    // Sending informational log message
    log << MSG::INFO << "Hello World!" << endreq;

    // Access the Event TDS and retrieve the top-level "Event" object
    SmartDataPtr<Event::EventHeader> evtTds(eventSvc(), EventModel::EventHeader);
    if (evtTds) {
        evtTds->setRun(10);
        evtTds->setEvent(eventCounter++);
        log << MSG::INFO << "<RunId, EvtId> = <" << evtTds->run() << ", " << evtTds->event() << ">" << endreq;
    }
    return StatusCode::SUCCESS;
}
```

Input JobOptions

```
ApplicationMgr.DLLs = {"GaudiAlg", "GaudiAud", "GlastSvc", "HelloWorldGaudi"};

ApplicationMgr.ExtSvc ={"EventSelector/EventSelector", "EventLoopSvc" };

ApplicationMgr.Runable= "EventLoopSvc";

AuditorSvc.Auditors = { "ChronoAuditor" };

//-----
// Private Application Configuration options
//-----

ApplicationMgr.TopAlg ={"HelloWorld" };

// Set output level threshold (2=DEBUG, 3=INFO, 4=WARNING, 5=ERROR, 6=FATAL )
MessageSvc.OutputLevel = 3;
//-----
// Event related parameters
//-----
ApplicationMgr.EvtMax = 10; // events to be processed (default is 10)
EventPersistencySvc.CnvServices ={"EventCnvSvc"};
EventSelector.Input = "SVC=DbEvtSelector";
EventSelector.PrintFreq = -1;
```

Output

```
Job options successfully read in from c:\heather\glst\packages\gr\HelloWorldGaudi\18\HelloWorldGaudi\v0\src\test\jobOptions.txt
ApplicationMgr SUCCESS
```

```
=====
                               Welcome to ApplicationMgr $Revision: 1.47 $
                               running on KELLYLAPTOP on Wed Mar 17 13:21:12 2010
=====
ApplicationMgr  INFO Successfully loaded modules : GaudiAlg, GaudiAud, GlastSvc, HelloWorldGaudi
ApplicationMgr  INFO Application Manager Configured successfully
EventPersistenc... INFO "CnvServices": ["EventCnvSvc"]
EventSelector_1  INFO Selection root:/Event CLID:110
EventSelector    INFO Stream:EventSelector_1 Def:SVC='DbEvtSelector'
ApplicationMgr  INFO Application Manager Initialized successfully
EventLoopSvc    INFO Runnable interface starting event loop as : MaxEvt = 10
EventPersistenc... INFO Added successfully Conversion service:EventCnvSvc
ChronoStatSvc    INFO Number of skipped events for MemStat-1
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 0>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 1>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 2>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 3>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 4>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 5>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 6>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 7>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 8>
HelloWorld      INFO Hello World!
HelloWorld      INFO <RunId, EvtId> = <10, 9>
EventLoopSvc    INFO Processing loop terminated by event count
****Chrono****  INFO *****
****Chrono****  INFO The Final CPU consumption ( Chrono ) Table (ordered)
****Chrono****  INFO *****
ChronoStatSvc    INFO Time User : Tot= 0 [us] #= 1
HelloWorld:execute INFO Time User : Tot= 0 [us] Ave/Min/Max= 0(+ 0)/ 0/ 0 [us] #= 10
****Chrono****  INFO *****
ChronoStatSvc.f... INFO Service finalized succesfully
ApplicationMgr    INFO Application Manager Finalized successfully
Current time: Wed Mar 17 13:21:12 2010 ( 0 s elapsed)
```

References

[Some words from Fermi's Online Workbook.](#)

[Gaudi Home](#)

[Recent Developments reported to CHEP 2009 by Gaudi team](#)