

Upgrading to Gaudi v21r7

Notes concerning the transition to Gaudi v21r7

JO Updates

HistogramPersistency = "NONE";

We still do not use Gaudi's native histogram service, and now we must explicitly shut it off in our JO files.

SmartIF Interface Change

SmartIF's interface has changed, and we no longer pass in a IID_XXXX identifier when creating a new SmartIF. This affected all the crnv sources in Overlay.

TkrUtil's unit test reveals a new Gaudi feature detailed in Athena's doc:

Detecting Service Initialization Loops

Sometimes, loops in service initialization are created, eg when ServiceA is created on demand, it uses ServiceB, which needs ServiceC, which then also needs ServiceA. When the ServiceManager tries to initialize all the services sequentially, a loop will be created, and unpredictable results can occur. A new feature of Gaudi allows these loops to be detected, and the job can be configured to halt, or merely print out debug information. The following [JobOptions](#) for the ApplicationMgr are available:

- **InitializationLoopCheck** (Boolean): turn on loop checking
- **LoopCheckIgnore** (StringArray): list of services to ignore when checking for loops
- **ServiceLogFile** (String): name of file to write detailed output to

*If **InitializationLoopCheck** is set to true, and a loop is detected, you will see an error message like:*

ServiceManager ERROR Initialization loop detected when creating service "AuditorSvc"

*If the output level of the ApplicationMgr is set to **DEBUG** or **VERBOSE**, a full traceback will be printed.*

*If **ServiceLogFile** is set to non-null, whenever a service is created, an entry will be written to that file showing who requested the service to be created.*

Some of these JO parameters no longer seem available. Looking at TkrGeometrySvc::initialize, there are a number of other services started up - all of which also set up TkrGeometrySvc in their initialize step. This was probably never a great idea anyhow - but now Gaudi will not allow such behavior. To work around this, it seems that TkrGeometrySvc just provides methods to access these other services to other classes - so now the other services are set up with the initial call to getTkrXXXSvc(), where if the ptr is null, we set up the service, otherwise the ptr is returned as usual. Not elegant, but it should get us through for now.

IObjManager is no more

The removal of IObjManager requires us to modify our use of tools in some of services - where we used the mere existence of tools to trigger some behavior, such as the initialization of random numbers for a particular package. Instead, observers have been introduced, which are triggered when a tool is created. This change affects GlastSvc's GlastRandomSvc, FluxSvc, CRflux, and HepRepSvc.

Gaudi::LTime is now Gaudi::Time

Gaudi's native time class has been changed and this affects CalibSvc. In particular, the methods associated with hour, minute, etc have an interface change where a boolean input parameter is required to denote local time. This affects a number of logging statements throughout packages that utilize CalibSvc.

Rootmap files

The DLL/*.cxx files are no longer necessary - in fact they were only necessary on windows anyhow. Instead, rootmap files must be generated for all gaudi-cognizant shared libraries. This is done using the genmap application available with the ROOT external. Note that we were already creating rootmap files for the xxxRootData packages, however we use the rlibmap application in that case. I'm inclined to leave that as is for now.

A new wrinkle is avoiding the creation of rootmap files when there are more than one library created by a package - where the main library does need to produce a rootmap file. For example, GlastClassify - which needs GlastClassify.rootmap but also generates ClassificationTree.lib, a static library which does not require a rootmap file.