Dual-use tool requirements

The Athena Framework is used for all official production in ATLAS, including reconstruction, simulation, etc.

Most (~75%) ATLAS analysis code is written in alternate frameworks or in a basic ROOT environment.

Dual-use tools need to support the appropriate components depending on the environment.

- **Build system**: CMT, ROOT
- **Event store**: Storegate, TEvent, TStore
- **Tool Retrieval**: ToolSvc, Simple ToolStore
- **Messaging**: Gaudi, simpleMsgStream

### Combined performance tools

Tools that implement the recommendations and calibrations of the combined performance groups of ATLAS, including:

- Object corrections
  - Energy calibrations
  - Resolution smearing
- Object selections
  - Cleaning bad objects
  - Identification criteria
- Object weights
  - Reconstruction/identification efficiencies

### Anatomy of a dual-use tool

- **Interface class** inheriting from IAsgTool
  - Supports the component design pattern
- **Implementation class** inheriting from AsgTool
  - Provides event store access through common interface
  - Provides messaging functionality
- **Compiler directives** switch underlying framework machinery

### Example workflow

1. **Input xAOD**
   - Electrons
   - Muons
   - Jets

2. **Generate shallow copies** for each systematic
   - Must be flat N-tuple or even a hash table

3. **Output xAOD**
   - Electrons
   - Muons
   - Jets

Management of tools may be handled by high level analysis tools (super-tools) or algorithms

- **Correction, selection, efficiency tools**
- **Multi-object selection, calculation tools**
- **Write corrected shallow copies to output file for later analysis**

Output file may instead be flat N-tuple or even a simple histogram file

### Outlook and future work

The dual-use tool design has transformed the way analysis code is written:

- Harmonized interfaces and behaviors make life easier for the user
- Dual-use concept gives users flexibility in choice of analysis framework
- Underlying tool machinery eases burden on tool developers to support multiple frameworks

Possible future developments:

- Improved handling of meta-data in ROOT
- Additional dual-use framework components

---

Steven Farrell, for the ATLAS Collaboration