

# Report of the Event Tag Review and Recommendation Group

Kétévi A. Assamagan (chair), Dario Barberis, Stan Bentvelsen, Gustaaf Brooijmans, Kyle Cranmer, Jack Cranshaw (chair), Andrea Dell'Acqua, Amir Farbin, Daniel Froidevaux, Fabiola Gianotti, Ian Hinchliffe, Tom LeCompte, Tadashi Maeno, David Malon, Frank Paige, Giacomo Polesello, David Quarrie, David Rousseau, R. D. Schaffer, Maria Smizanska, Guillaume Unal, Kai Voss, Monika Wielers

Final Version, 12 April 2006

*Charge:*

- Review the current TAG definition.
- Evaluate its suitability for analysis, looking at most relevant use cases taking into account constraints from database and requirements from the computing model.
- Recommend appropriate modifications in the light of the above
- Look at the issues related the TAG builders - from the AOD; how to build the TAG to best suit the analysis: for instance the tools for pre-selection should be the same used in the tag builder?
- Any other relevant issues.

# 1 Introduction

In order to facilitate access to the large volumes of data (multiple petabytes per year) which will be produced during data taking and Monte Carlo production at ATLAS, work has proceeded on building a system of event-level metadata to allow selections of a subset of events to use as input to an analysis. This was included in the ATLAS Computing Model and was first studied and implemented by the Physics Analysis Tools group based on the decisions of the ESD/AOD Task Force. They used tools developed and supported by the CERN IT group and the ATLAS Database group. During 2005 this structure was put through various tests and evaluations. Also, work by physicists on reconstruction and analysis led to an improved understanding of the requirements on the TAG. This report addresses the effect of these new inputs on the previous work with regard to content and the infrastructure needed to support it.

## 2 Background to Discussions

### 2.1 TAG in the Computing Model

In a multi-petabyte event store, a means to determine which events are of interest to a given analysis without first opening event data files and searching them, and to retrieve exactly those events and no others, is an important capability. In the ATLAS computing model[1], an event-level metadata system—a TAG database—is the component that addresses this need. When an event is written, metadata about the event—an event “tag”—is exported, along with a reference to the event data, for publication in the metadata system. The principal criterion for inclusion of information in the TAG is therefore its utility in supporting selection of events for physics analysis. TAG must also contain sufficient navigational information to support location and retrieval of events at upstream levels of processing, including RAW, ESD, and AOD.

The computing model requires that each event data product depend only upon its immediate parent for upstream event information: AOD should be producible from ESD without reference to RAW, and TAG from AOD without reference to ESD. As an operational optimization, the TAG data may be written when AOD are produced, or when small AOD files are merged into larger ones.

The TAG data will be written to local files during AOD production at the Tier 0 center, and published into a central repository that can then be replicated at every site hosting a full copy of the AOD (Tier 2 and lower), as shown in figure 1. The computing model estimates that AOD size will be approximately 100k per event, and budgets on the order of 1k per event for the TAG (TAG size = 1 % AOD size). It is possible that a Tier 2 center might host the “TAG database” simply by hosting the TAG files, rather than by offering a database service.

At the time of this report, the streaming model for ATLAS is still being considered. Several points are already apparent though:

- Streaming will occur at multiple levels. For example, calibration/physics at Event Filter, express/physics at Tier 0, and individual physics group streams at Tier 0/Tier 1/Tier 2.
- The streaming model may allow events to appear in multiple streams. The potential overlap is not yet known but believed to be small.

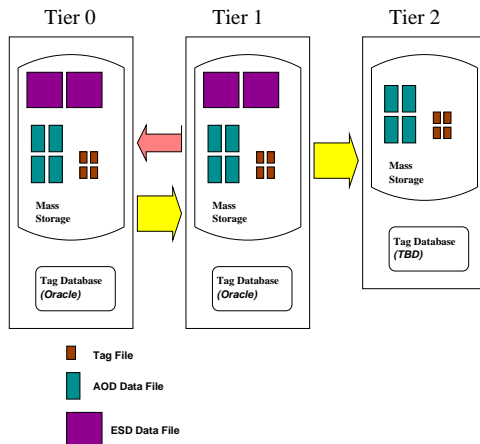


Figure 1: Schematic of distribution of data and TAG among Tier0/1/2 sites. Sizes of rectangles are not indicative of data size.

- Overlap of event content across the TAG for different data streams is not an issue. It was always assumed that TAG references could cross physical stream boundaries. This was, in fact, part of the purpose for having the TAG.
- The TAG should allow groups that do not qualify for a dedicated event data stream to mark events for their virtual streams in the TAG, or to write their own collections of "TAG"s.
- The TAG may be used to support higher level streaming where physicists 'skim' off data based on object properties, e.g. number of electrons, trigger + run quality, etc.

In the computing model, both event data objects and event TAG may be streamed at the output of Tier 0 reconstruction. Streams may be based upon triggers, and, more generally, upon physics and reconstruction group definitions of interesting events. The current model requires that each event in the bulk physics sample be written to exactly one data stream, but *references* to an event may be written to multiple TAG streams. It is expected that it will be possible to make queries using multiple TAG streams. If the model is revisited to allow writing events to more than one data stream, the consequences for the event-level metadata system, which would then need to manage references to a variable number of "versions" of the event at a given processing stage, would need to be considered. A general schematic is shown in figure 2.

## 2.2 Current TAG Infrastructure

ATLAS uses the LCG POOL collection infrastructure[2, 3] to store TAG data. Two persistence formats are supported: ROOT and Relational. The ROOT collections are stored in files and the Relational collections are stored in a relational database. The supported database flavors are MySQL and Oracle. The underlying schema has undergone some changes as we have upgraded versions of POOL, but the collection interface has remained backward compatible. Nevertheless, applications which bypass the POOL interface, e.g. direct SQL queries to a relational collection, have required changes. POOL collections support SQL predicates[4, 5]. In this instance SQL predicate indicates the terms one would put after WHERE in an SQL selection.

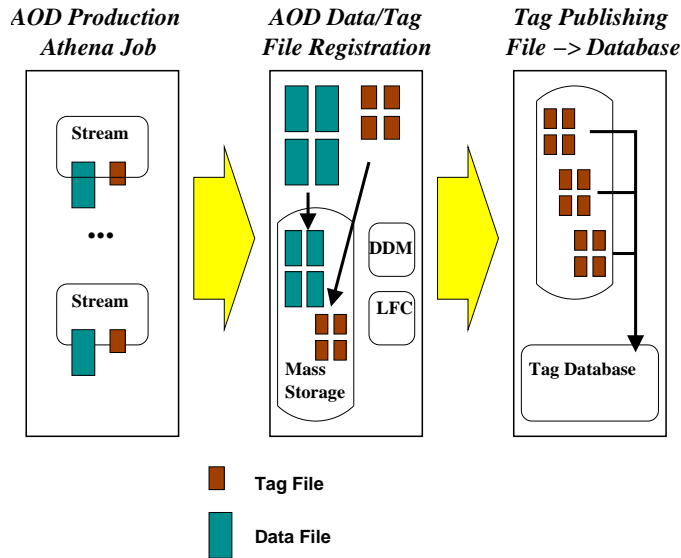


Figure 2: General schematic of production and publishing of TAG data. Sizes of rectangles are not indicative of data size.

The structure of a POOL collection is similar to a flattened struct. The user defines a list of attributes with simple types (bool, int, string, etc.). Arrays are supported only through attribute names, e.g. ElectronPt1. The resulting list of attributes is mapped onto a structure in ROOT (TTree) or in a relational database (data + link tables) by the POOL collection code. In the TAG building code the application simply sets values for the pseudo-struct (AttributeList) and fills it with a row per event. Although the underlying technologies support zero suppression, POOL collections require a value for every element of the AttributeList. In job options for the RegistrationStream, the user then designates which AttributeList/TAG builder to use and which DataHeader (with streams there may be multiple DataHeaders in StoreGate concurrently) to use as the primary reference. The provenance (Back Navigation) record for that primary reference is also unpacked and stored, although currently that is via a token string which takes up more space than it could.

As mentioned in the computing model section, the current usage of collections uses the ROOT format as a way of storing collection data locally during job execution for jobs which generate TAG data. The ROOT collection is then stored as a normal file/dataset and published into a relational collection with data from other ROOT collections.

### 2.3 External Dependencies of TAG Creation and Use

- **DDM:** The event reference used by POOL collections contains a pointer to the file with the event data and an offset within the file. This pointer takes the form of a unique ID, the POOL GUID. This generates an implicit dependency on an external service to resolve the pointer to the file into a physical file from which to fetch the data. In the current ATLAS data management system, this means querying the Distributed Data Management (DDM)[6] system or the local file catalog.
- **AMI:** The TAG attributes themselves will be stored in files at various sites and tables

in various database servers. A system for locating the appropriate files and/or tables is expected to be based on the same system used to locate other ATLAS resources. This implies a dependency on AMI[7, 8].

At the time of this report, neither of these external dependencies had been exercised except in the most rudimentary tests. Even with these limited tests, further integration work was indicated, which should receive a high priority in 2006 and hopefully form part of the Computing System Commissioning (CSC).

## 2.4 Current TAG Content

Based on input from the computing model, the current (late 2005) TAG was developed by the ESD/AOD Task Force[9] in 2004. It contains some event identification information (run, event numbers) and information on reconstruction level physics objects (photons, electrons, muon, jets, etc.) The code used to build the TAG was put in the PhysicsAnalysis/EventTAG container package. The event content evolved in minor ways from the Task Force report to now. One of the main issues was that, for variables like  $P_T$ ,  $\eta$ ,  $\phi$ , etc., the size of these arrays continued to grow in a way which exceeded the size budget and the ability of the infrastructure to handle it. Some duplication was also noted for the muon quantities.

## 2.5 Existing Tests and Evaluations of Current Implementation/Content

- **November 2004 Tier 0 Preparedness Test[10]:** This exercise was done at CERN using purely local resources as part of preparations for the DC2 Tier0 exercise. It used files with a mix of events from DC2 Monte Carlo data. Due to problems with CERN resources, the exercise was very small scale ( $\sim 10k$  events/stream). It used the original computing model Tier 0 process where  $n$  ESD files serve as input to a job which produces  $m$  AOD files for  $m$  streams. In this case TAG streams were produced with a 1-to-1 association with the AOD streams. A TAG stream with references to bad events and one for all events was also produced. As noted before, current requirements are that events are written to only one AOD stream, but references to an event can be put in multiple TAG streams. In this test, the streaming model used had large overlaps, so that, on average, events appeared in 3 TAG streams, and some AOD streams had very few events. The TAG became the only means of accessing some of the AOD streams. So far, this is the only exercise which has actually used TAG with a streaming model.
- **Spring 2005 Rome Workshop[11]:** the TAG data was produced for a sub-sample of Rome Monte Carlo data which had been transferred to CERN by the end of May 2005. The content of the TAG was slightly different from that used in November 2004. Specifically, an integer field was added for the dataset identifier and one for a random value used in performance testing. The results were stored in a MySQL database at CERN which also contained a File Catalog with pointers to the files in castor.cern.ch (2.3M events). A small number of users tried to use this facility to do typical physics queries.<sup>1</sup> The Rome TAG also provided a realistic sample for performance studies by the Database group[12].

---

<sup>1</sup>Tom LeCompte reported that he found the TAG useful for counting the number of events in a certain category and selecting events for analysis in ATHENA.

- **Summer 2005 Use Case Studies[13]:** The PAT and DB groups did an analysis of the use cases for TAG. The actors considered were Tier 0 Operations, Tier 1 Operations, Tier 2+ Operations, Physics Groups, and Physicists. Six *areas of use* were identified:
  - Official TAG Building
  - Content Evolution
  - Schema Evolution
  - Simple End User Needs
  - User Defined TAG
  - Extension of Existing TAG Data or Queries

and two *modes of use* were identified:

1. Local use of the TAG files collocated with data files, e.g. TAG files in ROOT format distributed with the AOD merged files.
2. Usage of a remote, central TAG database server to find data locations or run jobs.

This led to the following requirements on the deployed TAG database:

1. User is not required to know which persistency technology is being used. **Status:** satisfied.
2. A standard, technology independent querying system is available. **Status:** satisfied
3. Metadata format is documented for all supported formats/technologies. **Status:** Available, but limited.
4. Ability to query on TAG attributes for distributed data, i.e. data which is not local to the TAG database. **Status:** Untested.
5. Web service access to global TAG database. **Status:** Prototype available, limited testing.
6. Ability to define new attributes as understanding of the detector and physics improves. **Status:** Available, but some inefficiencies.
7. Queries are reproducible, which implies unique keys/qualifiers in the schema. **Status:** Limited to collections which have stopped accepting new events.
8. Previous selections can be stored for later use. **Status:** Limited to writing out a new collection.
9. Selection results can be catalogued and published to other sites. **Status:** Limited to web page documentation.

## 2.6 Event Level Metadata in Other Experiments

Below is a short list of experience with event level metadata by other experiments.

- Babar[14]: The TAG was used, but attributes were objects in the event processing, and not just outputs used for later selection.

- Zeus[15]: The TAG was added after the experiment was commissioned. They used  $\sim 300$  attributes which gave 1.2kb/event. The TAG attributes were up-to-date within a few weeks of data taking and contained detector and physics quality information.
- CDF[16]: CDF did not make serious use of any form of TAG. TAG capabilities were added after commissioning when the SAM system was adopted.
- D0: The TAG was part of the SAM system used by D0. The TAG could be produced with content specific to a physics group, or even a physics process. File level summaries of event metadata were also used.
- LHCb[17]: The TAG is planned to be produced during physics analyses and updated roughly every 3 months.
- CMS: At this time, CMS has no plans to use event level metadata, but will instead rely on streams from the Tier 0 to produce manageable datasets.

## 3 TAG Content versus Computing Model Constraints

### 3.1 Strawman TAG Definition

The committee considered a Strawman TAG definition which fits within the constraints of size and which generally matches what seemed to be reasonable TAG use cases. The content was divided into the following sections:

1. Collection Information.
2. Global Event Quantities.
3. Trigger Decisions.
4. Electrons, Photons, and Muons.
5. Jets, Taus, and Missing  $E_T$ .
6. Information specific to Physics and Combined Performance.

For the object quantities one of the main issues was how to handle multiple particles. The first problem (and one of the triggers for the formation of this group) was a technological issue. The mapping onto a relational database currently allows only for the equivalent of fixed length arrays. When this is coupled with the need to keep the TAG size below 1 kbyte per event, a compromise is required on the number of objects which have data in the TAG. Also, the query interface only supports simple, sql-like queries, which makes a large number of objects becomes difficult to query.

## 3.2 Collection Information

This concerns two bits of information. The first is the references used to find the event data related to the metadata. This is a one-to-many relationship between metadata row and data references to different stages of processing. Based on discussion of the user requirements, the recommendation is that in addition to the primary reference to the AOD data that references to ESD and RDO data should be sufficient.

The Collection ID information is needed to determine the origins or relevance of a set of rows of metadata. Consequently, these do not need to be repeated for every event, but the current infrastructure needs expanding before this can be implemented.

As indicated in Table 1, these objects are rather large, so there are multiple reasons to store this in a more efficient manner. Upcoming releases of POOL should shrink the space requirements of ESD and RAW references to the size of AOD references (10 bytes).

Name	Attributes	Size
Data References	AOD	10, link table
	ESD	150, string
	RDO	150, string
Collection ID <i>(To be Moved)</i>	Software version	16, string
	Processing version	32, string
	TAG builder version	32, string
<b>Total Size</b>		<b>390 bytes</b> $\rightarrow$ $\sim$ <b>30</b>

Table 1: List of Collection information and data references: definition, sizes, and types.

Collection level metadata should be extensible, and it should be possible to list all collections available based on any of these fields, such as software version.

## 3.3 Global Event

The TAG content associated to the event consists of the following:

- All information included in the EventInfo record in the data stream should also be included in the TAG. Currently, this includes the event number, the run number and the event type. The event type could be calibration, simulation, test beam, real data or a combination thereof.
- Streaming criteria and results.
- The number of tracks, the number of vertices and the coordinates of the primary vertex.
- The missing energy: it is proposed that the missing  $E_T$ , the  $\phi$  of the missing energy and the summed cell  $E_T$  should be in the TAG.



Attribute definition	Attribute Name	Size
Run Number	RunNumber	unsigned int
Event Number	EventNumber	unsigned int
Event Type	Calibration	bool
Event Type	TestBeam	bool
Event Type	Simulation	bool
Event Type	RealData	bool
Number of tracks	NTrk	unsigned int
Number of vertices	NVtx	unsigned int
Primary Vertex	VtxX, VtxY, VtxZ	3 floats
Streaming Criteria/Results	Stream	unsigned int
Random Number	RandomNumber	unsigned int
Time Stamp	TimeStamp	float
Bunch by bunch luminosity	Luminosity	float
Luminosity Block Number	LumBlockNumber	unsigned int
Missing $E_T$	MET	float
$\phi$ of the MissingET	METPhi	float
Summed cell $E_T$	SumET	float
Detector Status	PIX	unsigned int
Detector Status	SCT	unsigned int
Detector Status	TRT	unsigned int
Detector Status	PSB	unsigned int
Detector Status	EMB	unsigned int
Detector Status	TCB	unsigned int
Detector Status	TCE	unsigned int
Detector Status	PSE	unsigned int
Detector Status	EME	unsigned int
Detector Status	HEC	unsigned int
Detector Status	FCA	unsigned int
Detector Status	GSC	unsigned int
Detector Status	MDT	unsigned int
Detector Status	CSC	unsigned int
Detector Status	RPC	unsigned int
Detector Status	TGC	unsigned int
Detector Status	MBS	unsigned int
Good for physics	GoodRun	unsigned int
<b>Total Size</b>		<b>134 bytes</b>

Table 2: The TAG attribute associated to the global event data. The "good for physics" word is determined from a combination of the detector status (data quality) words. In general, we propose to copy all the event identification information into the TAG.

- The detector status and the "good for physics" information: words indicating the status of each detector and a "good for physics" word derived from some combination of these status words should be present in the TAG. It is recommended that one should know how these status words are/will be handled online and in the data stream. The detectors providing status words are:
  1. In the Inner Detector: Pixel, SCT and TRT — one word from each.
  2. In the Calorimeter: Barrel Pre-Sampler, Barrel EM, Tile Barrel, Tile Extended Barrel, End-cap Pre-Sampler, End-cap EM, HCAL, FCAL, and Gap Scintillators — one word from each.
  3. In the Muon Spectrometer: MDT, CSC, RPC and TGC — one word from each.
  4. Minimum bias scintillators — one word.
- The bunch by bunch luminosity. This is however a beam related attribute.
- A random number field to process the TAG for a limited number of events, a specific sample or a random sample selection.
- The time stamp.

The proposed TAG content for the global quantities associated to the event is listed in Table 2.

### 3.4 Trigger

There was no final agreement on the Trigger content of the TAG. Part of this was due to the fact that the definition of triggers will change over time, and the exact form and use of trigger menus has not been defined yet. The following guidelines were established though:

- The trigger representation in the TAG should be independent, preferably static, of the representation in the data stream or the trigger database. Thus, the trigger menu could be read as low-pT electron, high pT-electron, double-electrons, low-pT muon, high-pT muon, double muon etc.
- In addition to knowing which triggers an event satisfied, it might be useful to know whether an identified 'offline' particle was as well identified as a 'trigger' object. This might be a useful feature at start-up to understand the trigger performance. For example one could select events in which a good offline particle was identified, which was either rejected by the trigger or for which no L1 trigger object was found. An alternative way would be to ask the TAG for events in which the 'expected' trigger item rejected the event and the offline found a 'good' particle. Which way would be the preferred one at start-up needs to be further looked at in detail in the future.

In the current definition of the TAG content, the proposed trigger words, containing yes/no bits for the trigger items, are included as shown in Table 3.

The correspondence between a specific trigger bit and the trigger menu can be derived from the configuration database. At the time of building the TAG the trigger configuration data has

Attribute Definition	Attribute Name	Size
CTP Decisions	CTPWord,	3 words, "unsigned int" each
Level 2 Trigger Masks	L2PassedTrigMask	3 words, "unsigned int" each
Event Filter Masks	EFPassedTrigMask	3 words, "unsigned int" each
<b>Total Size</b>		<b>36 bytes</b>

Table 3: Trigger words (indexed from 0 to 2) proposed to be included in the TAGs. The above the attributes contain the yes/no bit for the trigger decisions.

to be read in, interpreted and then mapped onto the trigger menu definitions of the TAG. The 'physicist' can then ask for the events passing for example the high-Pt Tau trigger or passing one of the jet triggers etc. Alternatively he/she may simply ask for the events passing bit XX.

### 3.5 Electrons, Photons and Muons

Two methods were discussed for TAG attributes of a certain class of objects such as off-line electrons, photons, and muons, which require different selections. The first is to have multiple categories, e.g. loose or tight; while the second consists of having only loosely selected objects with an additional word to encode the results of the various, increasing levels of tightness cuts as a bit pattern. The criteria for the loose selections and the increasing levels of the tightness cuts would be defined by the respective combined performance groups. For the electron and the photon in the steady state operations, the looseness and tightness criteria may be considered along these lines:

- Mask the isEM bits to select on all (loose) and  $P_T > 7$  GeV.
- Medium tight: loose with tighter hadronic leakage cuts.
- Tight electron: Medium tight with tighter track-matching criteria (e.g. including E/p).
- Very tight electron: tight with TR-cuts and explicit conversion rejection cuts.
- Photons between loose and very tight criteria: choose intermediate efficiency and purity points, to be specified perhaps once the loose and tight definitions are in place.
- Very tight photon: includes explicit photon/ $\pi^0$  rejection procedure.

It was noted that for the electron and the photon, there is no need for separate fields for their isolation since the isolation is inherently a part of their identification, unlike the case of the muons. For the muons, it is recommended that there should not be any multiple reconstruction output candidates in the TAG:

- Loose muon: a stand-alone muon candidate from the muon spectrometer.
- Tight muon: a loose muon successfully combined with an Inner Detector track.
- Various levels may be considered in-between the loose and the tight criteria. For example, a low Pt muon, i.e., an Inner Detector track associated to a muon spectrometer segment.

Attribute definition	Attribute Name	Size
Total number of loose electrons	NLooseElectron	unsigned int
Loose electron $P_T$	LooseElectronPt	float
Loose electron $\eta$	LooseElectronEta	float
Loose electron $\phi$	LooseElectronPhi	float
Loose electron Tightness	LooseElectronTightness	float
<i>(repeated for 4 electrons)</i>		
Total number of loose photons	NLoosePhoton	unsigned int
Total number of loose converted photons	NLooseConvertedPhoton	unsigned int
Loose photon $P_T$	LoosePhotonPt	float
Loose photon $\eta$	LoosePhotonEta	float
Loose photon $\phi$	LoosePhotonPhi	float
Loose photon Tightness	LoosePhotonTightness	unsigned int
<i>(repeated for 2 photons)</i>		
Total number of loose muons	NLooseMuon	unsigned int
Loose muon $P_T$	LooseMuonPt	float
Loose muon $\eta$	LooseMuonEta	float
Loose muon $\phi$	LooseMuonPhi	float
Loose muon Tightness	LooseMuonTightness	unsigned int
Loose muon Isolation $E_T$	LooseMuonIsolationEt	float
Loose muon Track Isolation	LooseMuonIsolationN	unsigned int
<i>(repeated for 4 muons)</i>		
<b>Total Size</b>		<b>240 bytes</b>

Table 4: Definition of electron, photon and muon TAG attributes for steady state operations. The  $P_T$ ,  $\eta$  and  $\phi$  attributes are indexed from 1 to  $N$  (where  $N = 4$  for electrons and muons, and  $N = 2$  for photons)

- The muon isolation is in a separate field: one field for track isolation and another for calorimeter isolation.

The suggested contents of the TAG for electrons, photons and muons, for steady-state operations are listed in Table 4. Wherever relevant the  $P_T$  is signed. Totals of four loose electrons and muons and of two loose photons are proposed. The objects are ordered in decreasing  $P_T$ . The total number of objects passing the loose selections may be greater than four (electrons, muons) or two (photons).

### 3.6 Jets and $\tau$ -jets

Tau candidates and jets were handled in a similar fashion in the TAG. The proposed common TAG attributes for accepted jets are the  $P_T$ ,  $\eta$ ,  $\phi$  and the total number of jets passing the selection. For each  $\tau$  candidate, the number of charged tracks and the  $\tau$ -jet likelihood are also included. For the jets, the total number of b-tagged jets, the b-likelihood of each jet and the

Attribute definition	Attribute Name	Size
Total number of $\tau$ -jets	NTauJet	unsigned int
$\tau$ candidate $P_T$	TauJetPt	float
$\tau$ candidate $\eta$	TauJetEta	float
$\tau$ candidate $\phi$	TauJetPhi	float
$\tau$ candidate number of tracks	TauJetNTrk	unsigned int
$\tau$ candidate likelihood (repeated for 2 $\tau$ candidates)	TauJetLikelihoodRatio	float
Total number Jets	NJet	unsigned int
Total number b-tagged Jets	NBJet	unsigned int
Jet $E_T$	JetET	float
Jet $\eta$	JetEta	float
Jet $\phi$	JetPhi	float
B-tag likelihood	JetBLkh	float
Summed $E_T$ over Jets (repeated for 6 jets)	JetSumET	float
<b>Total Size</b>		<b>216 bytes</b>

Table 5: Jet and  $\tau$  TAG attributes: two  $\tau$ -jet objects and six standard jet objects are reserved. Attributes associated to a  $\tau$ -jet are index from 1 to 2 whereas those associated to standard jet are indexed from 1 to 6. The B-tag likelihood for jets is the combined weight of various tagging algorithms.

summed  $E_T$  over all jets are stored in the TAG. Totals of two  $\tau$ -jets and six jets are proposed. The  $\tau$ -jet candidate transverse momenta are signed, and the objects are ordered in falling  $P_T$ .

### 3.7 Physics and Combined Performance Groups

It is recommended to have words in the TAG for each physics or performance group where each group is free to encode any "yes/no" analysis decisions that they find necessary or useful. For example, the B-physics group may want to encode in their word(s) "yes/no" decisions on different hypotheses about exclusive processes. Table 6 shows the proposed TAG content list for the analysis words.

For simulated data, an additional set of words to be specified at a later stage should be added with the purpose of including truth information concerning the hard-scattering process and possibly interesting objects produced by the event generator (high-pT leptons, neutrinos, etc.).

Attribute definition	Attribute Name	Size
Electron/Photon Identification	EgammaID	unsigned int
Muon Identification	CombinedMuonID	unsigned int
Jet MissingEt Identification	JetMissingEtID	unsigned int
Tau Identification	TauID	unsigned int
Jet Tagging	JetTagging	unsigned int
B-Physics Analysis	BPhys	unsigned int
Exotic Physics Analysis	ExoticPhys	unsigned int
Higgs Physics Analysis	HiggsPhys	unsigned int
SUSY Physics Analysis	SUSYPhys	unsigned int
SM Physics Analysis	SMPhys	unsigned int
Top Physics Analysis	TopPhys	unsigned int
Heavy Ion Analysis	HeavyIonPhys	unsigned int
<b>Total Size</b>		<b>48 bytes</b>

Table 6: The TAG attribute for each physics and combined performance group to encode identification or "yes/no" decisions. One word is reserved for each of the twelve groups.

## 4 TAG Building Tools

The following clarifications and recommendations were made on the design and implementation of the TAG building tools.

1. *The implementation of the TAG and of the builders are the responsibility of Physics Analysis Tools (PAT).*
2. *The main purpose of the TAG database is to support event selection for physics analysis of AOD.*
3. *The TAG content and structure is a component of the ATLAS EDM, and as such, all rules defined for the ATLAS EDM apply to the TAG, in particular modifications and schema evolution requirements. The primary building code will be based on selection mechanisms/algorithms and the definition of some user words determined by the Physics and Combined Performance Groups (CPW). The scope of the selection mechanisms is **far from clear**. The Event TAG Review group believes that the selection mechanism ought to be simple but the CPG will still have to implement codes (under the PAT supervision). This is a crucial point that needs further discussion outside the Event TAG Review.*
4. *The tools for producing the TAG must be stable. This is turned into a requirement to have ad-hoc tools (people want to be sure of the boundary condition according to which the TAG was produced). The TAG proper must be stable throughout big productions. The TAG must be protected against "driver's errors" (which points to the definition of a well defined interface to be presented to the users and most of the TAG building process being hidden behind it).*

5. *The TAG building process should not have any unnecessary dependencies*, e.g. on pre-AOD stages of processing. There was a consensus that lightweight tools which do not depend on any specific framework/analysis tools were preferred. It is a reiterated requirement that, were we to re-use any specific component of e.g. the EventView, this be made independent and decoupled so as to be usable in "standalone" mode.

The current definition of the TAG, as presented in this document, has been implemented by PAT, according to the recommendations and requirements listed above. Several tests are being carried out toward a full deployment by the release 12.0.0. The general organization of the TAG tools implementation is as follows:

- **The TAG EDM package:** this is located in PhysicsAnalysis/EventTag/TagEvent: this contains the definition of the TAG contents to be built, the definition of the attribute names and the implementation of the container classes for the TAG attribute data. Modifications of the TAG content — for example additions to the TAG, modifications of the TAG attribute names — are to be applied in this EDM package.
- **The TAG building tools:** the tools are implemented in a modular way so as to be able to build fragments of TAG. There are several tools, organized according to the Physics and Combined Performance Groups — it is expected that after the initial implementation, these groups will take control of the maintenance and evolution of their respective tools in collaboration with PAT. The packages for the TAG building tools are in the respective container packages of the Physics and CWP under the PhysicsAnalysis top container package.
- **TAG aggregation algorithms:** these algorithms use the above mentioned tools to build the various fragments of TAG, fragments which are subsequently combined to construct the full TAG. The algorithms themselves are modular with full runtime control of what goes into the TAG. Checks are also made to ensure that the tools are building the expected TAG fragments as defined in the EDM package. The TAG aggregation algorithms are implemented in PhysicsAnalysis/EventTag/EventTagAlgs.

At the time of this writing, some of these tools are at the skeleton stage and it is expected that the detailed implementation of selection tools will require input from the Physics and Combined Performance Groups.

## 5 Event TAG Content Management

TAG attributes are either copies of their AOD equivalents, or in a few cases are computed as simple combinations of these quantities. No recalibration or recomputation is done on these quantities. The TAG aggregation has been restructured so that responsible groups can maintain the software relating to their fragments of the TAG. This means that the selection criteria for Electrons, Photons, Muons and Jet/Tau will be the responsibility of the relevant Combined Performance groups, and the details of the analysis words in the TAG will be the responsibility of the appropriate Physics groups. The other quantities and the structure for building the TAG fragments from the AOD will be the responsibility of the PAT group.

Changes to the TAG will be subject to the same change review as changes to the EDM for AOD/ESD. The nature of the TAG is to provide small, fast access to the data. Additions to the payload solely for backward compatibility conflicts with this requirement. Because of this, because the TAG selection code is relatively simple and because the resources needed to regenerate the TAG are relatively small, we expect a lower threshold for non-backwards compatible changes.

## 6 Requests to the Database Group

Another output of the discussions was a list of new or extended functionality that users would like from the TAG infrastructure. These have been classified into the standard three categories: essential (will fail otherwise), important (severe usability or performance penalties if missing), and desired (would be used if available).

- Essential

1. *Support of all basic numeric types.* Most numeric types are already supported along with strings. Problems arose with bool values because they are not represented and queried in the same way by the current underlying tag database implementation technologies
2. *Cataloguing of TAG files/tables by software version, TAG builder version, and processing version.* Other than naming conventions within a known database, this capability does not yet exist. Note that this should be done in an efficient manner, and that it should be possible to stick a foreign key into a different metadata system, e.g. AMI, where version information is catalogued more generally.
3. *Run dependent information availability.* There is certain information which changes on a run-by-run basis which may be needed to understand other information in the TAG such as trigger and quality information. This must be available for TAG queries. Discussions led to the conclusion that this is a job for an Interval-of-Validity (IOV) database, but the committee wanted to avoid any dependencies on the Conditions database, so a means to store this within the TAG database was requested. Supplemental structures are probably needed to support this.
4. *More efficient storage of back navigation references.* As noted in the section on Collection Information, the references to pre-AOD stages are currently stored as attributes of an event. This is inefficient and uses up space needed for TAG selection data. Storage-efficient representations of references to upstream data are expected in upcoming releases of the POOL collections infrastructure.
5. *Extensibility of an existing TAG database.* The committee discussions led to the requirement that the TAG database be extensible in two ways. First, one must be able to add more events to an existing database. Second, one must be able to add new attributes to rows in an existing database.

The first of these is already possible with the existing infrastructure, although for larger TAG databases, it may be necessary to partition it. This has not been



satisfactorily studied for large databases yet. It would be good if tests of this capability were part of Computing Services Commissioning.

The second is currently only crudely supported in that one may define a new list of attributes and then re-write the entire TAG with then new attributes filled. Ideas on how to do this in a way which avoids re-writing of the data and allows modest schema evolution are being investigated by the Database group.

- Important

1. *Ability to do Loops/define Procedure/define Functions.* The SQL standard does not support loops across columns solely with WHERE clause content. Yet with the implicit arrays in the metadata, most queries would benefit from having this ability added in some way. One of the ways of doing this is to define a server-side procedure or function[19].

- Desired

1. *Zero suppression.* The ability to leave quantities with no meaningful value as NULL would help storage. An example is when there are only three electrons, but one must put some value in ElectronPt4.
2. *Variable length arrays.* As noted earlier, the current infrastructure only supports flattened, fixed length 'arrays'. This can be inefficient.

## 7 Conclusions

The Review group examined the content of the TAG and has presented here a strawman proposal for the content that fits within the computing model budget (706 bytes) and provides relevant data for doing event selection. During our discussions among ourselves and with experts outside the committee, it became clear that we did not have, at this time, a detailed understanding of initial operations and steady state running necessary to finalize the TAG content. Much of this will have to come from the Combined Performance Groups, Trigger Groups and Physics Groups. Nevertheless, we believe we have improved the TAG content significantly from the original implementation towards the requirements as they have been reviewed by the group.

The TAG building process was also examined and various alternatives, such as using EventView, were discussed. Any further discussions of this topic will occur in the PAT group meetings.

The group feels that this constitutes a successful fulfillment of the Charge. Certain areas such as clarifying the Trigger content and getting more input on typical user queries require ongoing work. The agendas and minutes can be found in the CDS agenda system at

<http://agenda.cern.ch/displayLevel.php?fid=279>

under "Event Tag Review & Recommendation Team".

## 8 Summary of the Main Recommendations

- TAG is provided to efficiently collect the datasets needed for physics analysis and not to do analysis.
- Divide structure of TAG building into pieces that can be turned over to the groups responsible for reconstructing and validating that information. **Status: Complete for Release 12.0.0**
- Minimize dependencies on other services, such as the Conditions or IOV databases, for interpreting TAG content, such as triggers.
- Content will be managed in a similar fashion to event content.
- A system for locating appropriate collections, either database tables or TAG files, must be available.
- Collection-level and run-level metadata must be removed from the TAG content and stored elsewhere, but in a way that it can be queried.
- The Trigger content was difficult to define due to a variety of unknowns. A placeholder has been set aside, but it is imperative that follow-up work in the Trigger group provide a more concrete implementation.

## References

- [1] ATLAS Computing Group, ATLAS Computing Technical Design Report, ATLAS TDR-017, CERN-LHCC-2005-022.
- [2] D. Duellmann, "The LCG POOL Project General Overview and Project Structure", Proceedings of Computing in High Energy and Nuclear Physics, 24-28 March 2003.
- [3] <http://lcgapp.cern.ch/project/persist/>
- [4] C. Fehily, SQL: Visual Quickstart Guide, 2nd Ed., Peachpit Press, 2005.
- [5] A. Beaulieu, Learning SQL, O'Reilly Media, 2005.
- [6] T. Wenaus, <https://uimon.cern.ch/twiki/bin/view/Atlas/UserData?topic=DDM>
- [7] S. Albrand, et. al., "The AMI Database Project: Atlas Data Challenge Bookkeeping, and the Tag Collector, a new tool for Release Management", Proceedings of Computing in High Energy and Nuclear Physics, 24-28 March 2003.
- [8] S. Albrand, et. al., <https://atlastagcollector.in2p3.fr:8443/AMI/>
- [9] K. Assamagan, et. al., "Final Report of the ATLAS AOD/ESD Definition Task Force", ATL-SOFT-2004-006
- [10] J. Cranshaw, et. al., <http://www.hep.anl.gov/cranshaw/DC2-Tier0-DB.html>
- [11] J. Cranshaw, et. al., <http://www.hep.anl.gov/cranshaw/RomeTagCheckList.html>
- [12] K. Karr, et. al., <https://uimon.cern.ch/twiki/bin/view/Atlas/CollectionPerformanceTests>
- [13] J. Cranshaw, et. al., <https://uimon.cern.ch/twiki/bin/view/Atlas/AtlasEventTagTools>
- [14] Private communication Amir Farbin.
- [15] Private communication Kai Voss.
- [16] Private communication Jack Cranshaw.
- [17] Private communication Maria Smizanska.
- [18] T. Maeno, et. al., <https://uimon.cern.ch/twiki/bin/view/Atlas/AthenaAwareNTuple>
- [19] SQL functions/procedures
- [20] D. Bales, Java Programming with Oracle JDBC, O'Reilly Media, 2001.

# Sample Queries

- General Collection Information

1. How many events are in the collection of name A?
2. Which production task or series of tasks produced these TAGs?
3. What are the names and types of the queryable attributes?
4. Where can I find documentation on what these variables mean and how they were produced?
5. What named ref's are available? Only AOD?
6. What (type,key) pairs are available if I retrieve a ref of name aaa?
7. Is there a new version of the TAG database and should I be using it?
8. Is my local server up-to-date or should I connect to CERN?

- Content Queries

1. Give me all the events with at least 2 electrons and missing  $E_T$  greater than 10 GeV.
2. .. Limit the previous query to events which are marked as good for physics, and that have 'good' (*how to define this???*) SCT and TRT.

- Summary Queries

1. Give me the number of events for some content query.
2. Give me the sum of the luminosity for some content query.
3. Give me the list of files for some content query.
4. What fraction of the events are lost due to coincident problems with PIX and SCT?