ATLAS

ATLAS Dataset Nomenclature

Document Version:	3.1
Document ID:	**ATL-COM-GEN-2007-003**
Document Date:	2010-05-04
Document Status:	Update 2010 , second issue

Abstract

This document describes the dataset nomenclature for ATLAS datasets.

It was approved by the ATLAS Trigger and Offline Board on 2007-11-21[1]

Version 3 of the document is the 2010 update.

Authors: Solveig Albrand, Dario Barberis, Fabiola Gianotti, Claude Guyot, Richard Hawkings, Ian Hinchliffe, Beate Heinemann, Andreas Hoecker, Alexei Klimentov, Giovanna Lehmann, Pavel Nevski, Hans Von der Schmitt.

Additional Authors (2009 edition): James Catmore, Dave Charlton, David Cote, Luc Goossens, Armin Nairz.

Editor: Solveig Albrand

ATL-GEN-INT-2007-001
20/07/2010

Document Change Record

Title:	ATLAS Dataset Nomenclature		
ID:	**ATL-COM-GEN-2007-003**		
Version	**Issue**	**Date**	**Comment**
0		2006	A.K et al (see reference 1)
1	0	2007-07-19	Edited S.A. after meeting 2007-07-17
1	1	2007-08-31	Modified by S.A. after email exchange
1	2	2007-09-20	Modified after nomenclature meeting 2007-08-31. Author list added.
1	3	2007-10-04	Modified by S.A. after email exchange with all authors, and meeting with A.H., B.H &C.G.
1	4	2007-10-18	After further discussion, ProductionStep put back into the nomenclature name.
1	5	2007-11-14	Minor changes:description of merge step changed, max length of prodStep name set to 16.
1	6	2007-11-21	Approved by the TOB[1]
2	0	2008-12-08	Annual revision of the document. New text is marked with a vertical line in the margin. All the tables have been updated to take into account the additions made during the year. A summary has been added. Appendix 1 concerning configuration tags implementation has been re-written to take into account the current usage. Appendix 2 has been added to take into account container datasets.
2	1	2009-02-24	Further modifications after a discussion on 2009-02-10
2	2	2009-07-31	Update, adding references to - groupmcxx - Debug stream event merging - Super reprocessing datasets - Note on production system internal naming
3	0	2010-01-21	Changed definition of user and group dataset naming. TID modification for tasks with >10K jobs.
3	1	2010-05-04	ddo datasets and Physics containers added

Contents

1 Introduction and Scope

This document defines the nomenclature of physics datasets for ATLAS.

It extends and replaces the document “datasetsV152.pdf”(2) (March 2006) which deals principally with Monte-Carlo datasets.

It takes into account the requirements of the real data datasets, and also the recommendations of the Metadata Task Force(3).

The scope of the present nomenclature covers:-

(1) Monte-Carlo datasets
(2) Real Data Datasets.
  a. Primary
  b. Super datasets (including relational event collections)
(3) User datasets
(4) Group datasets
(5) Conditions datasets
(6) Database release datasets
(7) SW release datasets
(8) Application Internal datasets

This document first describes the general rules for the formation of names for all ATLAS datasets, and then goes on to consider each type of dataset in particular.

Official datasets are real data or files generated using the production system; they should be registered in both the DDM catalogues and in AMI.

Group and user datasets are formed in the analysis process. They may be registered in DDM, but they are not registered in AMI at the moment. This may change during 2010.

After the publication of the first edition of this document, other types of datasets were distinguished. Since they must in some cases be catalogued with physics datasets it is useful to define their nomenclature following the same principles.

The tables in this document contain examples of currently approved values. Users should consult AMI for the latest list of values.

The document will be updated at least once a year.

2 Conventions and Definitions

2.1 Conventions

(1) The use of the verb “**must**” implies a requirement upon members of the collaboration.
(2) The use of the verb “**should**” implies a recommendation.
(3) The use of the verb “**may**” implies an option.
(4) Items which are still under discussion or awaiting input from concerned parties are written in *italic text*.
(5) Examples which are given to illustrate the application of the rules are framed.

Example

This is an example

2.2 Definition of some terms used in this document

AKTR The Task Request Interface “Alexei Klimentov Task Request” also known as ATLAS Knowledge managemenT Requests.

AMI Atlas Metadata Interface, used for dataset selection.

DDM	Distributed Data Management
DQ2	The current instance of DDM software. “Don Quixote 2”
Nomenclature	A community’s system of names for things; a systematic naming; catalogue or register
Nomenclature conflict	In computing science when an algorithm designed to provide an identifier from a set of parameters produces a result which has already been assigned to another set of parameters a “collision” is said to occur. Unless all the possible dataset parameters are included in a mnemonic dataset name, there will always be some cases where the nomenclature rules generate a name which is already registered. As it was thought that the use of the word “collision” in this context might confuse some physicists we use the term “nomenclature conflict” to describe this situation in this document. The convention must be able to deal with such conflicts.
Reference table	A database table which is used to restrict the allowed values in the field of other database tables. Also known as a “foreign key constraint”.
Tier 0	The agent which forms the real data datasets and defines and registers their names.
TR	Task Request. The agent which defines the simulated and re-processed datasets and registers their names.
VOMS	Virtual Organization Management System. A layer above grid certificates used for authentication and authorisation. It is used by ATLAS to indicate to applications the physics group membership and the database privileges accorded to users.

3 General Requirements for Dataset Nomenclature

(1) It must be possible to generate names using an algorithm which covers all possible cases.

(2) The name generated must be unique within the community. Once a name has been defined and applied to a set of data it must not be reused for a different set.

(3) All members of the collaboration must be aware of the nomenclature rules, and be able and willing to apply them.

(4) The names should be in part mnemonic.

(5) Since the names are derived from a sub-set of metadata, there may be nomenclature conflicts and we must have rules to deal with them.

(6) The most upstream application is responsible for enforcing the nomenclature and for forming valid names.

(7) Downstream applications may parse the names but they must not be dependent on parsing the names to get metadata information. An exception to this rule is the data placement by DDM at destination sites. Data is organized according to the values of two of the dataset name fields (project and dataType).

(8) The dataset name is logical. It does not necessarily imply anything about the physical organization or data. Similarly, the dataset name does not imply anything about the file names of the files which belong to the dataset.

(9) The procedure for changing the nomenclature is defined.

4 Constraints

4.1 Name Structure

(1) The dataset name consists of a number of fields.
(2) The number of fields may vary according to the project.
(3) Fields may have some semantic meaning.
(4) It is possible for applications to determine the number of fields and their meaning from AMI.

4.2 Maximum Acceptable Length

(1) The dataset container name of an official dataset must not exceed 132 characters.
(2) The total contained dataset name must not exceed 255 characters (the limit set in the DDM catalogues).
(3) Fields which, by their nature, have a variable number of characters, have a maximum length defined.

4.3 Allowed Character Set

Only the following characters are allowed in a dataset name

a-z, A-Z	Lower and upper case letters
0-9	Decimal number characters
-	The “dash” or “minus” sign
_	The underscore character
.	The dot character
/	The slash character

4.4 Use of Characters

(1) The dot character “.” must only be used to separate fields of the dataset name.

(2) The underscore character should be used to separate parts of a field.

(3) A field must not start with the underscore character. This is because a field with a leading underscore is a convention in the file naming, which marks a partition number.

(4) A field must not end with an underscore character.

(5) The dash or minus sign “-“, should not be used to indicate polarity, but only as a separation character to improve readability. Note that the “+” character is forbidden.

(6) Only one field is allowed to be purely numeric. This field is used to indicate the run number or, in the case of Monte Carlo data, the dataset number. This numeric field has a fixed number of characters and it must be left zero filled.

(7) The slash character can only be the **last** character of an official dataset name. It is used for a specific marking of dataset containers (6). Dataset containers are compound datasets which aggregate other datasets. They are defined for the convenience of Distributed Data Management. Monte Carlo physics datasets which combine the output of several production tasks are declared as containers. (see Appendix 2)

4.5 Case Sensitivity

(1) Dataset catalogues must consider that two dataset names which differ only in case are equivalent.

(2) Dataset searches must be case insensitive.

(3) The case given at the registration of a dataset must be respected by all applications.

Example:

The dataset “myFirstDataset” will be registered with an upper case “F” and “D”.

It will not be possible to register another dataset called “myFirstdataset” as this will be a breach of the unique name rule. This would be considered to be a conflict.

A search for a dataset called “myFirstdataset” will find the dataset named “myFirstDataset”.

5 General Dataset Nomenclature Conventions

The general format of an ATLAS dataset name is

Project.[OtherFields.]DataType.Version[/]

The dataType is used by DDM when distributing data to destination sites. The project field is used by DDM to place data on storage elements.

If the dataset has been declared to be a “container” then by convention the last character of the dataset name is a slash “/” (see Appendix 2).

5.1 Project

This field is a string which identifies the particular physics or computing context of a set of datasets.

(1) Only a few projects (<10) should be declared each year.

(2) Sub projects may also be defined.

(3) The project name part of the project identifier precedes the sub project part and is separated from it by the underscore character “_”.

(4) Adding a new project or sub-project requires a strong justification. New projects and sub projects can be defined only by a small number of people (Physics coordinator, Run coordinator, Data preparation coordinator)

(5) Project names are generic and easily understandable by members of the collaboration. The last two characters of the project name denote the beam period, which does not coincide exactly with the calendar year.

(6) The number of underscores in the project identifier must not exceed one.

(7) The length of the project identifier, (project name, underscore separator and sub project name) must not exceed 15 characters.

(8) A reference table of project names is kept in AMI.

(9) **Table 5-1** shows examples of the current list of projects. New projects will be defined for each beam period as appropriate.

(10) No project name must be used to name a dataset until it is registered in the AMI reference table. When a new name is registered the information is sent to DDM mailing lists.

(11) Project identifiers are assigned to a particular nomenclature. Sub projects of the same project follow the same nomenclature.

Examples:

mc08_test and mc08_cos must have the same nomenclature; mc12_test may have a different nomenclature.

Table 5-1 : The currently approved list of project identifiers

Generic Projects	Meaning
mcnn	Monte Carlo production 20nn
datann	Real data 20nn
user	User analysis data
group	Physics group data
groupmcnn	A Monte Carlo dataset which is NOT replicated to all Tier 1 sites example : groupmc08.105807.JF35_pythia_jet_filter.merge.AOD.e418_a84_t53

Specific Projects	Meaning
data10_10TeV	
data10_2TeV	
data10_7TeV	
data10_900GeV	
data10_calib	Tag used for all detector calibration data.
data10_calocomm	2010 Calorimeter combined commissioning runs
data10_cos	2010 cosmics combined runs
data10_idcomm	2010 Inner Detector commissioning runs
data10_larcomm	2010 LAr calorimeter commissioning runs
data10_muoncomm	2010 Muon spectrometer commissioning runs
data10_tilecomm	2010 Tile calorimeter commissioning runs
mc09_10TeV	mc09 production at 10 TeV
mc09_14TeV	subproject tag for upgrade studies
mc09_2TeV	2.36 TeV MC
mc09_7TeV	mc09 production at 7 TeV
mc09_900GeV	mc09 production at 900 GeV

5.2 OtherFields

A suite of dot separated fields; the number and definition of these fields may vary according to the project and dataset type, but it is always well defined and available in the AMI project reference table.

Hence each dataset name has a predefined number of fields which all must be present and non-empty. Details for each dataset type are given in section 6

5.3 DataType

This field is a string which identifies the format of data in the dataset.

(1) Only data type tags approved by the Physics Coordinator and the Data Preparation Coordinator must be used.

(2) New data type tags must be approved and entered in the AMI data type reference table before use. Table 5-2 shows examples of the currently approved list of data types. When a new dataType is defined the information is sent to DDM mailing lists.

(3) All other tags which may have been used in the past are considered deprecated.

Table 5-2 : Examples of the currently approved list of data types. Both base types and compound types are listed.

DATATYPE	DESCRIPTION
AOD	Analysis Object Data
CBNT	Flat Combined Ntuple
COND	ATLAS condition datasets
DPD	Derived Physics Data
DPD_EGAMMA	Derived e/gamma; Performance-group DPD, produced off ESDs
DPD_SUSY	Derived SUSY. Physics-group DPD, produced off AODs
DPD_TAUDIJETS	Derived tau/dijets; Performance-group DPD, produced off ESDs
DPD_TAUWZ	Derived tau/WZ ;Performance-group DPD, produced off ESDs
DPD_TOPEW	Derived top/electro-weak physics; Physics-group DPD, produced off AODs
ESD	Event Summary data. Output of reconstruction. Supports reconstruction, alignment and detailed analysis
ESD_FILTERED	First used for data08 M6
EVNT	Output of event generation, HepMC event record.
HIST	Histograms
HITS	Output of (g4) simulation (and maybe pile_up)
HPTV	HighPtView NTuples - all files merged
LOG	
NTUP	ROOT readable NTUPLE
RAW	Detector output, or simulated bytestream
RDO	Raw Data Objects; either converted detector output or digitization output
SAN	Structured Athena-aware NTuple
TAG	Event Tags
TXT	Output of event generation, translated to ASCII, to make it exportable

(4) Sub sets of data types may also be defined, by the addition of a sub data type.

(5) New sub data types must be approved by the Physics Coordinator and the Data Preparation Coordinator.

(6) Sub data types must be entered in the AMI data type reference table.

(7) The data type part of the dataType identifier must be separated from the sub data type part by an underscore character.

(8) The number of underscores in the dataType must not exceed one.

(9) The length of the dataType identifier (data type, underscore separator and sub data type) must not exceed 15 characters.

Examples:

DPD_TAUDIJETS, DPD_TOPEW, ESD_FILTERED.

5.4 Version

The version field is an encoding of the configuration used for each of the production steps which were passed by the data in the dataset

Table 5-3 : The characters designated to mark the parts of the version tag, for each production step.

Character	Input data type	Production step	Tag setter
a	BS	atlfast	TR
b		bytestream	TR
c		calproc	Tier 0
d		digit	TR
e	-	evgen	TR
f	RAW	recon	Tier 0
g	-	dbgrec	DAQ
m	AOD/DPD	merge	Tier 0
o	-	daq	*obsolete*[i]
p	ESD	select	TR
q	ESD	select	Tier 0
r	BS/RAW	recon	TR
s		simul	TR
t	AOD/DPD	merge	TR
v		super dataset formation	Data Preparation

[i] It was decided that DAQ would not set configuration tags because all configuration information is contained in COOL and can be obtained through the runNumber.

(1) When a transformation is applied to data the version of the input dataset is passed to the output dataset, and the new configuration code is added as a suffix, separated from the preceding part of the version by an underscore character. In this way the configuration codes of successive production steps will be arranged in the order of data processing. Super dataset formation is an exception to this rule.

(2) When a transformation performs more than one "production step", the appended configuration tag will be determined by the input data type. For example the Tier0 reconstruction transform RAW→ESD (&HIST, NTUP)→ AOD & DPD; only an f tag will be applied, determined by the input data type.

(3) Each part of the version field is prefixed by a lower case letter which designates the input datatype to the transformation or production step.

(4) prodStep Tags are case sensitive – only lower case letters should be used

(5) The designated characters are shown in Table 5-3

(6) The configuration codes are defined in AMI. Each production step will have its own reference table of configuration codes

(7) Configuration codes are generated as necessary, by the tag setter shown in the third column of Table 5-3

(8) Configuration codes are calculated from the set of appropriate parameters for each production step. In most cases one of the parameters is the software release.

(9) The codes generated are decimal numbers allocated by the tag setter, or by AMI. No semantic meaning is given to tags.

(10) The number of decimal characters for the code is not fixed, and hence not left zero filled.

Example:

e23 (not e023)

(11) In some cases two actors can set tags for a given production step or transformation (example "recon"). In these cases each actor has a reserved tag character (example "r" for TR and "f" for Tier 0).

(12) The length of the version tag should not exceed 32 characters.

Some implementation details and a detailed example are given in Appendix 1.

6 Specific Dataset Nomenclature Conventions

In this section we give the special conventions for each type of dataset which exists at present or we are currently able to anticipate.

6.1 Monte-Carlo datasets

Monte-Carlo datasets are formed by a production system task request application (currently only AKTR). The Task Request application is responsible for forming the dataset names correctly and registering them in DQ2. Datasets are registered in AMI only after registration in DQ2, to ensure nomenclature coherence. The uniqueness of names is tested when the dataset is registered in DQ2.

The dataset name will have the form:

Project.datasetNumber.physicsShort.prodStep.dataType.Version[/|_tidnnnnnn[_SS]

A slash may be added to the end of the name if the dataset has been registered in DDM catalogues as a container (see Appendix 2). Container datasets are the logical equivalent of physics datasets.

If the dataset name has a suffix "_tidnnnnnn" where nnnnnn is a production system task number (6 digits with left zero filling) then the dataset contains only data made by the production task nnnnnn. The output from several tasks may be placed in one container.

A production task number of more than 6 characters, nnnnnn_SS, indicates that the production task nnnnnn was split into sub tasks, numbered consecutively using 2 characters with leading zeros. This is an internal production system implementation which addresses the problem of not having more than 10k files in a production dataset. The names of files produced by the sub production tasks will reference only the main task, numbered nnnnnn.

Appendix 2 gives more details of container datasets.

6.1.1 datasetNumber

The dataset number is currently a 6 digit decimal number. It must be registered in the "dataset Number broker" [4] by the production manager or by a physics group before being used. The dataset number is associated with a particular physics channel (it corresponds to a set of physics four vectors).

A given dataset number must always be associated with only one physicsShort text comment

6.1.2 physicsShort

This field is a text description of the physics channel.

The length of the physicsShort field must not exceed 40 characters.

6.1.3 prodStep

This is a string which gives the production step which was used to create the data. Although this field is not needed to define the dataset, since all the production steps are encoded in the version field, it renders the name more immediately understood.

Table 6-1 The currently approved list of production steps.

prodStep name	Production Step	Tag character (see rule (3))
atlfast	ATLFAST simulation	a
bytestream	bytestream	b
calproc	Used in Tier-0 reconstruction of reprocessed HLT data.	c
daq	RAW data acquisition	No tag (was previously o)
digit	digitization	d
evgen	Monte Carlo event generation	e
merge	TAG output from the production system	t
	merging at Tier 0	m
recon	reconstruction from the production system	r
	first reconstruction done at Tier 0	f
select	DPD production by the production system	p
	DPD production by Tier 0	q
simul	simulation	s

(1) Only prodStep names approved by the physics coordination must be used.

(2) New prodStep names must be approved and entered in the AMI prodStep reference table before use.

(3) The production step does not necessarily determine the config tag appended to the datasets name. This is defined by the input data type. This rule has been introduced because the production system transformations often perform more than one “production step”.

(4) The currently approved list of production steps is shown in Table 6-1. All other prodStep names which may have been used in the past are considered deprecated.

(5) Production step names must not be longer than 15 characters.

6.2 Real Data Datasets

6.2.1 Primary Datasets

Primary datasets contain all the RAW data which is recorded by the DAQ and the result of the first reconstruction and TAG building. The RAW dataset name and in particular the number of digits in the runNumber are set by DAQ in the SFO DB. Tier 0 management (TOM) forms the datasets, organizes the first reconstruction, the registration in DDM/AMI and the dispatch of data to T1s.

The current dataset model is 1 dataset/run/stream.

The dataset name has the form:

Project.runNumber.streamName.prodStep.dataType.Version[/|_Tnnnnnnnnnn]

Project

Real data datasets will contain the CM energy in the project tag, for example data10_10TeV, data10_7TeV. If the data is from a single beam the project tag is dataNN_1beam, for example data10_1beam, data09_1beam. If the data is for cosmics, the sub tag is “cos” e.g. data10_cos

runNumber

The DAQ run number is a 31 digit number which is incremented for each run. If the connection to the run number server is lost then the runNumber is replaced by a timestamp. It is recorded in the dataset name using 8 digits with left zero filling. If it is necessary to record a timestamp as a run number then the full 10 digits will be used.

streamName

This field is a string identifying the data stream, which is part of the filenames, and is assigned by the Trigger and DAQ configuration.

The length of the streamName string must not exceed 40 characters.

prodStep

This field is included in the primary dataset name for the same reasons as explained in section 6.1.3.

The identifier “daq” is used for RAW data. The other production steps are the same as for Monte Carlo datasets.

Version

As defined in section 5.4, and discussed in Appendix 1

If the dataset name has a suffix “_Tnnnnnnnnnn” then the dataset contains only data which arrived at the Tier 0 at time nnnnnnnnnn. The container dataset, whose name ends with a slash in DDM, may contain several timestamped datasets. Appendix 2 gives more details of container datasets.

6.3 Physics Container datasets

These datasets are also known as “super datasets” or “aggregate datasets”. They are made by grouping together in a single container of the contents of several other datasets or dataset containers.

The first defined use of such containers is to group together Tier 0 processed or reprocessed data for several runs for the same period. The component datasets may have differing reconstruction tags (f tags or r tags).

The following convention is used:

Project.containerName.streamName.PhysCont.dataType.version[/]

Project Tags, streamNames and dataTypes will not be mixed, and are the same as for other data.

The prodStep is always “PhysCont”.

The containerName is a free string of maximum length 40 chatacters, defined by the Data Preparation coordination. It will designate either a run range or a run range period.

The version field of PhysicsContainers is

[t0proc|repro]NN_vM

The first part will be either "t0proc" or "repro” for Tier 0 processing or production system reprocessing respectively, followed by a left zero filled two digit number "NN" , 01<=NN<=99, which is the "ordinal" number of the reprocessing of the data, and represents the reprocessing conditions. The version suffix "MM" is, another left zero filled 2 digit number 01<=MM<=99 to indicate that a different selection of reprocessing results was made for the same reprocessing conditions.

Example:

data09_900GeV.allYear.physics_CosmicMuons.PhysCont.ESD.repro02_v01

This dataset contains a run selection from the whole year 2009 for 900GeV collisions. It was the second reprocessing of the data, and the first selection from this reprocessing.

6.4 Calibration datasets

Calibration datasets exist as RAW data types with customised or non bytestream format. In some cases, the analysis of calibration data may also generate datasets of derived data types.

6.4.1 Calibration using physics datasets

Some systems do calibration using physics data; datasets for this purpose should follow the conventions for real primary datasets, placing the word “calibration” in the streamName part of the name.

Example:

data08_cosmag.00092094.calibration_IDTracks.daq.RAW

The nomenclature rules for these calibration datasets will be expanded when more practical experience is gained.

6.4.2 Calibration using specific data

Systems using specific calibration data should use the following convention.

dataNN_calib.xxxxxxxx.calibration_DetectorPart-meta-information-field.daq.RAW

Example (LAr Electronic Calibration data):

data08_calib.00654321.calibration_LArElec-Pedestal-Medium-EM.daq.RAW

N.B. These datasets have no derived output (reconstruction), only the RAW type will be used.

calibration_DetectorPart-meta-information-field

This special string is the calibration streamName. Its total length is 50 characters. This implies that the **DetectorPart-meta-information-field** part should not exceed 38 characters**.**

6.5 User and group defined datasets

(1) User and group datasets **must** use the following nomenclature for datasets registered with the ATLAS DDM. Distributed Analysis Tools must impose these nomenclature rules[ii].

user.userName.[otherFields]

group.groupName.[otherFields]

(2) The first field must be either “user” or “group”. **Note that the two figures denoting the period will no longer accepted for these datasets once tools convert to this convention.**

(3) The second field of the name identifies the user or the group, owner of the dataset.

(4) The VOMS physics group names [(5)] should be used for the groupName fields.

(5) Users will be assigned a unique ATLAS nickname which will be used for their datasets. DDM will manage the mapping of the nickname to the DN of the user’s certificate. Users will not be able to register datasets until they have been assigned a nickname.

(6) The maximum length of the user nickname is 32 characters.

(7) There must be at least one other field in the dataset name.

(8) It is highly recommended to use at least 3 fields. The first 3 (or 4 if available) fields will be used to form LFC and directory paths. Using only 3 fields will result in a flat structure.

(9) The maximum overall length of the dataset must not exceed 255 characters in length (the limit set in the DDM catalogues). This means that the [otherFields] length must be less than 216 characters.

(10) All other general rules for dataset names must be obeyed (See Section 4)

Example

This will create a flat structure:

user.jeanDupont.dataset.mc09
user.jeanDupont.dataset.data09

This will allow granularity in the file directory

user.jeanDupont.mc09.dataset
user.jeanDupont.data09.dataset

.

6.6 Conditions datasets

The conditions datasets contain POOL files of conditions payload data which are referred to by some of the conditions data itself stored in COOL. They are organised by function (offline MC production, real data cosmics etc) according to the different conditions database instances which exist.

project.internalCondNumber.shortComment.COND

[ii] After a period of transition .

project

An ATLAS conditions project name.

internalCondTag

Currently a 6 digit number but could become a field to group files corresponding to a particular running period (cf. super datasets), example p002_M4

shortComment

A string of not more than 30 characters**.**

COND

A special dataType used only by these datasets

N.B. These datasets are not coupled to any particular software release, so there is no need to include a version field in the nomenclature.

6.7 Database Release datasets

These datasets are formed for the purpose of transporting a database to a grid site. They are distinguished by the project tag “ddo” .

ddo.NNNNNN.[otherFields].vDBReleaseVersion

6.8 Software Release datasets

These datasets are formed so that software can be distributed by ATLAS DDM. Each dataset of the transformation release type contains only one file.

Datasets of the base release type are declared as containers, and they contain the set of transformation release datasets.

sitNN.nnnnnn.AtlasSWRelease.PAC.vMMmmp[cc]

The project name designates the dataset as a software release dataset for a particular beam period.

The second field is the pacball version.

The dataType is PAC because the dataset are formed by the PACMAN tool.

The last field is either the ATLAS software Transformation release (**.vMMmmpcc**) or the Base release.(**.vMMmmp**)

6.9 Application Internal datasets

A dataset name should only be altered within a sub-system by the addition of a suffix. The aim is that the name stays identifiable to the outside world.

Example from panda:

Panda splits datasets in pieces adding suffixes '_disNN','_subNN'

The lifetime of these datasets is limited by the lifetime of production tasks

7 Modification of this Convention.

Requests for changes to this convention must be addressed to the ATLAS Trigger and Offline Board (TOB).

8 Summary

Table 8-1 Summary of the different dataset nomenclatures defined in this document

Dataset type	Nomenclature	Example
Monte Carlo Datasets	mcNN_subProject.datasetNumber.physicsShort.prodStep.dataType.Version	mc08.105010.J1_pythia_jetjet.recon.ESD.e344_s456_r456
Real Data (Primary)	DataNN_subProject.runNumber.streamName.prodStep.dataType.Version	data08_cos.00079123.physics_HLT_Cosmics_NIM4.daq.RAW data08_cos.00079123.physics_HLT_Cosmics_NIM4.daq.AOD.f35
Physics Container datasets	Project.runRange.StreamName.PhysCont.dataType.version	
Calibration dataset	dataNN_calib.xxxxxxxx.calibration_DetectorPart-meta-information-field.daq.RAW	data08_calib.00654321.calibration_LArElec-Pedestal-Medium-EM.daq.RAW
User dataset	user.userName.[otherFields]	
Group dataset	group.groupName.[otherFields]	
Conditions dataset	Project.internalCondNumber.shortComment.COND	
Database Release datasets	ddo.NNNNNN.[otherFields].vDBReleaseVersion	ddo.000001.Atlas.Ideal.DBRelease.v09010207
SW Release datasets	sitNN.nnnnnn.AtlasSWRelease.PAC.vMMmmp[cc]	

Table 8-2 Summary of the different nomenclature components defined in this document

Name Component	Maximum length (chars)	How is it defined?
Project	15	By the data preparation coordinator, the physics coordinator, or the production manager
datasetNumber	6	Should be reserved by physics groups or overall MC production manager.
physicsShort	40	Should be reserved by physics groups or overall MC production manager. Linked to the dataset number
prodStep	15	Physics coordination
dataType	15	Physics coordination, Data Preparation coordination, DPD coordination
Version	32	Run coordination, Tier 0 or Task Request.
runNumber	10	Run number server
streamName	40	Run coordination
Calibration stream	52	Run coordination
Period	25	Physics coordination
containerName	40	Data Preparation coordination

9 References

(1) http://cdsweb.cern.ch/record-restricted/1070318/

(2) https://twiki.cern.ch/twiki/bin/viewfile/Atlas/SoftwareIntegration?rev=1.1;filename=datasetsV152.pdf

(3) https://edms.cern.ch/document/833723/1

(4) http://lpsc1168x.in2p3.fr:8080/opencms/opencms/AMI/www/RequestedDatasets/RequestedDatasets.html (This application is an example of an AMI reference table.)

(5) https://atlastagcollector.in2p3.fr:8443/AMI/servlet/net.hep.atlas.Database.Bookkeeping.AMI.Servlet.Command?linkId=434 (The list of the 23 VOMS physics and detector working groups of ATLAS)

(6) https://twiki.cern.ch/twiki/bin/viewfile/Atlas/AtlasDistributedComputing?rev=1;filename=DQ2_Containers_v08.pdf

Appendix 1. Notes on Configuration Tags

A1.1 List of Production Steps; Parameters used for definitions

Table A1-1 lists the parameters used to define the configuration tags.

Note that tags can be defined only by the actor listed in the table. These are :

- TR = “Task Request”, input to the production system.
- Tier 0 = real data management at Tier 0 ,
- DPD = DPD coordination
- DAQ = Trigger and Run coordination

Table A1-1 List of production steps and the parameters used to define the config tag.

Name	Tag char	Parameter set used	Actor	Note
atlfast	a	SWReleaseCache,DBRelease,Geometry,transformation,simulJobConfig,reconJobConfig,TriggerConfig,PhysicsList,description	TR	
bytestream	b	DBRelease, description, Geometry, JobConfig, SWReleaseCache, transformation, TriggerConfig	TR	1
calproc	c	description, Geometry, ConditionsTag, SWReleaseCache, transformation	Tier 0	3
dbgrec	g	SWReleaseCache	DAQ	4
digit	d	CavernDataset, ConditionsTag, DBRelease, description, Geometry JobConfig, MinBiasDataset, PhysicsList, SeedOffset, SWReleaseCache, transformation	TR	
evgen	e	description, JobConfig, SeedOffset, SWReleaseCache, transformation	TR	
gmerge	m	description, SWReleaseCache, transformation	Tier 0	2
daq	o	HLT Version	online	3
DPD cuts		DPDName, InputTriggerStreams, Cuts, EventContent	DPD	4
select	p	ConditionsTag, DBRelease, description, Geometry, JobConfig, SWReleaseCache, transformation, TriggerConfig	TR	
	q		Tier 0	
recon	r	ConditionsTag, DBRelease, description, Geometry, JobConfig, SWReleaseCache, transformation, TriggerConfig	TR	5
	f		Tier 0	6
simul	s	ConditionsTag, DBRelease , description, Geometry, JobConfig, PhysicsList, SeedOffset, SWReleaseCache, transformation	TR	
merge	t	AODCorrection, DBRelease, description, Geometry, JobConfig, SWReleaseCache, transformation, TriggerConfig	TR	
	m		Tier 0	
Super dataset	v			

formation				

Notes :

(1) The first version of this document [1] states that the prodStep name is “bytestream” and the dataType is “RAW” TR is still using “bstream” and “BS”.

(2) “gmerge” is obsolete. It was used for merging when the input format = the output format; requested by Tier0. See “merge”.

(3) “caloproc” is obsolete.

(4) “dbgrec” is a production step which indicates that RAW datasets come from the recovered debug stream.

(5) “o” tags are now obsolete. The DAQ sets no configuration tags

(6) DPD production. These are special configuration tags for DPD types which ARE NOT PART OF THE DATASET NAME.

(7) Tags with n<50 could be Tier 0.

(8) First reconstruction. An XML representation of the parameters is also defined in AMI.

In general actors are responsible for setting the tag number and numbers should be allocated consecutively [1] It is possible to ask AMI to allocate a number. Two production steps have two config tag letters because two actors are concerned by the production step. The set of parameters was initially the same for each actor, but has since diverged.

The first version of this document [1] states that the fields for configurations are predetermined. For a particular configuration where no value is given for a field, the default neutral value of “none” is given. When a new field is added to the definition of a parameter then the value of this field for previously assigned tags is set to “N/A”.

It was not possible to follow this specification for the parameter “SeedOffset” which is an integer. The default value of “0” has been used.

Tier 0 defines the f configuration tag in AMI BEFORE processing. The tag is read from AMI.

TR defines configuration tags independently and uploads them after definition.

A1.2 Overall Format of the Configuration Tag

Ref [1] did not define a maximum length for the configuration tags.

The limiting case seems to be when data is passed through several “evgen” steps.

Example:

valid1.018101.PythiaB_Bd_Jpsie3e3K0s.recon.e315_e321_e304_s385_r316

The chain is needed only because schema evolution does not work and three jobs are needed to get EVNT made in 13 readable in 12. This should be a temporary situation.

The first version of this document [1] did not specifically rule out upper case letters for the config tags.

These considerations lead to two additional nomenclature rules.

(1) The length of the version tag should not exceed 32 characters.

(2) prodStep Tags are case sensitive – only lower case letters should be used.

A1.3 Modification of the parameter list for a configuration tag

Obviously a situation where different actors are using differing sets of parameters will lead to confusion. Any request for an additional parameter should be circulated to all the actors concerned.

The current sets of parameters as defined in Table A1-1 are taken as the “correct” starting point.

Any additional parameters which are found to be needed are notified (if they come from one of the named coordinators) or requested (if they come from someone else) – using the “nomenclature actors” mailing list (atlas-data-nomenclature@cern.ch). “Requested” parameters must be approved by one of the coordinators. In this way we can be sure that all actors are informed of changes, and can take the necessary steps to change implementation.

A1.4 Validity of Configuration Tags

There was no discussion of this issue in the first version of this document [1] . However the issue is the same as for the dataset names themselves. Once a tag has been defined and used for a given set of parameters, it should not be reused for a different set of parameters. Therefore if it has been declared bad for any reason, it should be kept in the databases, but marked as unusable.

The following is a suggestion for tag validity management. It has been partially implemented in AMI.

The AMI implementation uses two status values readStatus, and writeStatus. (https://atlastagcollector.in2p3.fr:8443/AMI/servlet/net.hep.atlas.Database.Bookkeeping.AMI.Servlet.Command?linkId=483)

Table A1-2 Write Status : Determines if the value may be currently used for writing

valid	A valid value for the field, which is currently available for writing
obsolete	A field value which was used in the past and is now obsolete. No longer used for writing.
invalid	Not yet validated, or readStatus = trashed.

Table A1- 3 Read Status : May be valid if legacy data exists which uses this data.

valid	A tag which has been used to label valid data, not necessarily still available for writing.
trashed	A tag which was defined for use, but has produced no valid data. Should not be read.

Any data with a value for a field which is not declared valid for reading should be ignored. Not that setting a read status to trashed sets the write status to invalid.

Here is a proposition which would complete the implementation.

Production managers can also update the status of tags.

Put into place a system which locks tag definition once a tag has been used.

Put into place a system which trashes datasets if the read status of one of the constituent tags of its version field has been marked as ”trashed”

Make this part of an integrated dataset trashing tool.

Appendix 2. Container datasets

The full description of dataset containers can be found in reference (6).

Experience with data movement shows that the best data replication performance can be achieved with relatively small datasets but small datasets are not suitable for data organization. The concept of a dataset container was introduced to solve this problem.

A dataset container is a one level hierarchical structure for ATLAS data organization. A contained dataset is the smallest portion of data movable across the GRID by ATLAS DDM. A dataset can belong to more than one container.

For example Monte Carlo datasets are produced by a set of production system tasks which all have the same parameters, except for the number of events produced. The data produced by each task of a particular data type is a contained dataset. The set of contained datasets for a particular data type are registered as members of a container.

Real data files are formed into datasets at Tier 0. As some files for a run may arrive much later than others it is convenient to start the distribution of data before all the data for a run is transferred from the SFOs. Datasets of real data are time stamped, and the real data containers contain at least one timestamped contained dataset.

The container datasets are the exact logical equivalent of the physics datasets which were described in version 1 of this document. The container names are the same as physics dataset names except for the slash character at the end.

However containers are manipulated by a different set of DDM commands. DDM commands for containers require a slash character s the last character of the name.

AMI catalogues explicitly only container/physics datasets and does not show directly the slash character, but adds it transparently when necessary as in the DDM information page of AMI.