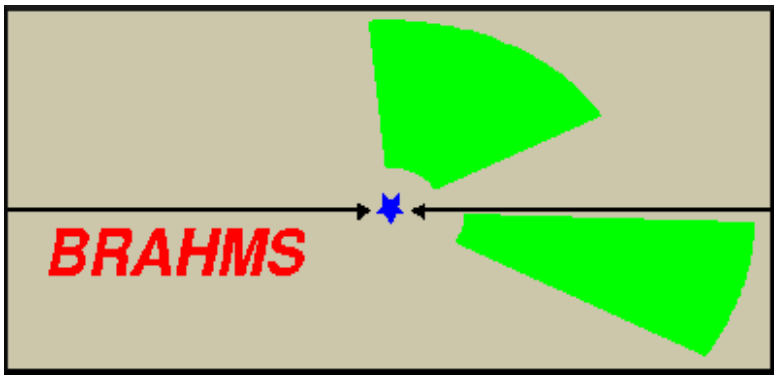
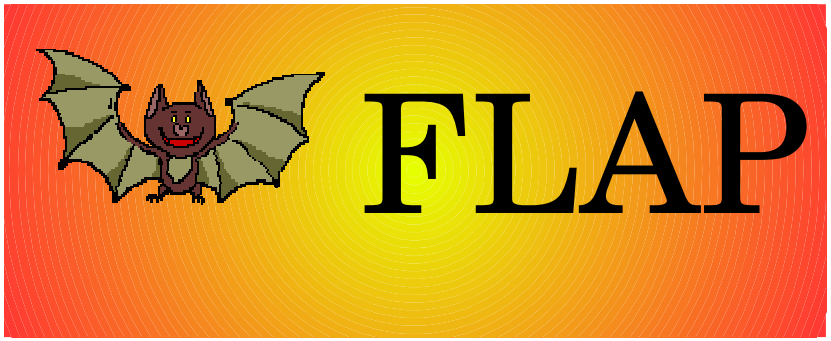
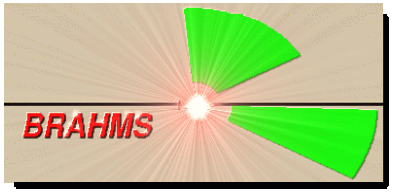
BRAHMS ana software, DSTs etc.





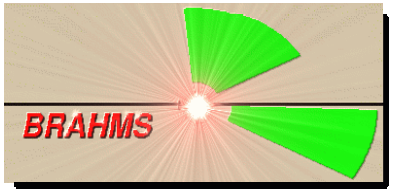
Philosophy



Why do we need software?
Why do we need COMMON software?

Quotes from students:

Philosophy

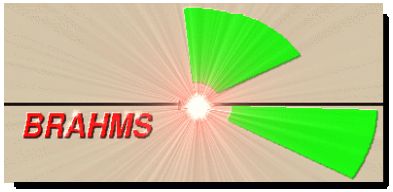


Why do we need software?
Why do we need COMMON software?

Quotes from students:

- *Was it run WITH the cut? But then all my files are useless!*

Philosophy

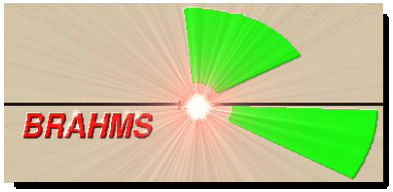


Why do we need software?
Why do we need COMMON software?

Quotes from students:

- *Was it run WITH the cut? But then all my files are useless!*
- *There's a parameter here. Do I **really** have to read Peters source code to find out what it does?*

Philosophy

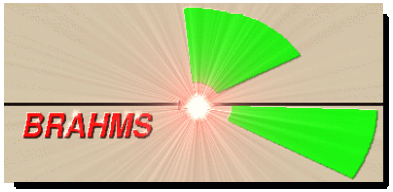


Why do we need software?
Why do we need COMMON software?

Quotes from students:

- *Was it run WITH the cut? But then all my files are useless!*
- *There's a parameter here. Do I **really** have to read Peters source code to find out what it does?*
- *You mean noone is actually **using** this software? I thought it was the official one?!?*

Philosophy

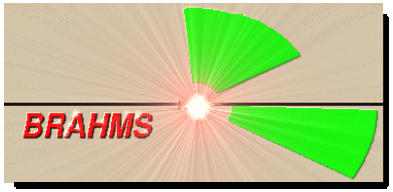


Why do we need software?
Why do we need COMMON software?

Quotes from students:

- *Was it run WITH the cut? But then all my files are useless!*
- *There's a parameter here. Do I **really** have to read Peters source code to find out what it does?*
- *You mean noone is actually **using** this software? I thought it was the official one?!?*
- *Aw, my gawd, you don't know what you're doing!!!*

Philosophy



Why do we need software?
Why do we need COMMON software?

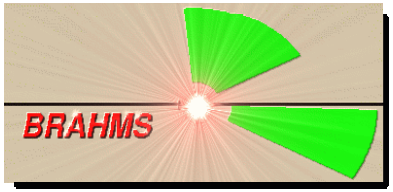
Quotes from students:

- *Was it run WITH the cut? But then all my files are useless!*
- *There's a parameter here. Do I **really** have to read Peters source code to find out what it does?*
- *You mean noone is actually **using** this software? I thought it was the official one?!?*
- *Aw, my gawd, you don't know what you're doing!!!*

We want to make our lives easier. So:

- More official and quasi-official *public* code.
- More documentation of run-conditions and functionality
- Common baseline files up to a certain point.

Analysis chain



1) Local tracking
2) Global tracking
3) Track matching
4) RDO...

- Common software (BRAT)
- Official files
- Well-documented reco-settings
- As much info as possible kept, baseline for checking calibs, data mapping etc.

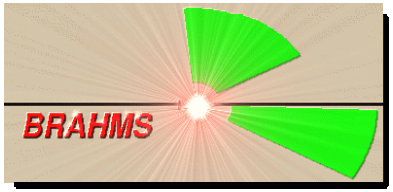
5) Generate DSTs

- NO PHYSICS! Just a selection of interesting info.
- Common, official files.
- Baseline for analysis

6) Data analysis

- bdstAna, FLAP, etc.
- Not common, but we need 1-2 *actively used* frameworks that are *well-documented* and *avaliable*.
- PID here... Personal preference...

What we want from a DST



Collaboration meeting
Krakow, 4.-7. June 2003

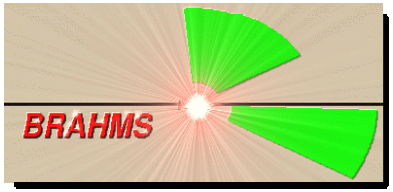
For analysis:

- Per-track information, enough for PID, corrections, checks...
- Per-event information (vertex, trigger...)
- Per-run info (magnet settings, scaledowns...)

In addition:

- Reasonable file sizes
- Fast readout
- Ease of use in final analysis frameworks (ROOT guru status *not* required!)
- Infrequent (or no) changes in structure and content.

B(E)DST
(BRAHMS Extended Data Summary Tree)



Collaboration meeting
Krakow, 4.-7. June 2003

T1

R
- Run nr.
- D1 set.
- D1 pol.
- D2 set.
- ...
- Scaledowns
- etc.

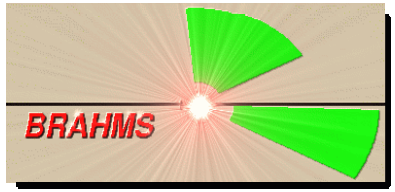
G
- z_BB
- z_INEL
- Centrality
- Multiplicity
- etc.

MRS
- Track info
- Detector info

FS
- Track info
- Detector info

TPM1
- Track info
- Detector info

B(E)DST
(BRAHMS Extended Data Summary Tree)



Collaboration meeting
Krakow, 4.-7. June 2003

T1

R
- Run nr.
- D1 set.
- D1 pol.
- D2 set.
- ...
- Scaledowns
- etc.

G
- z_BB
- z_INEL
- Centrality
- Multiplicity
- etc.

MRS
- Track info
- Detector info

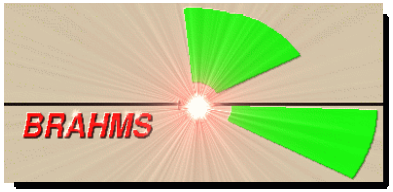
FS
- Track info
- Detector info

TPM1
- Track info
- Detector info

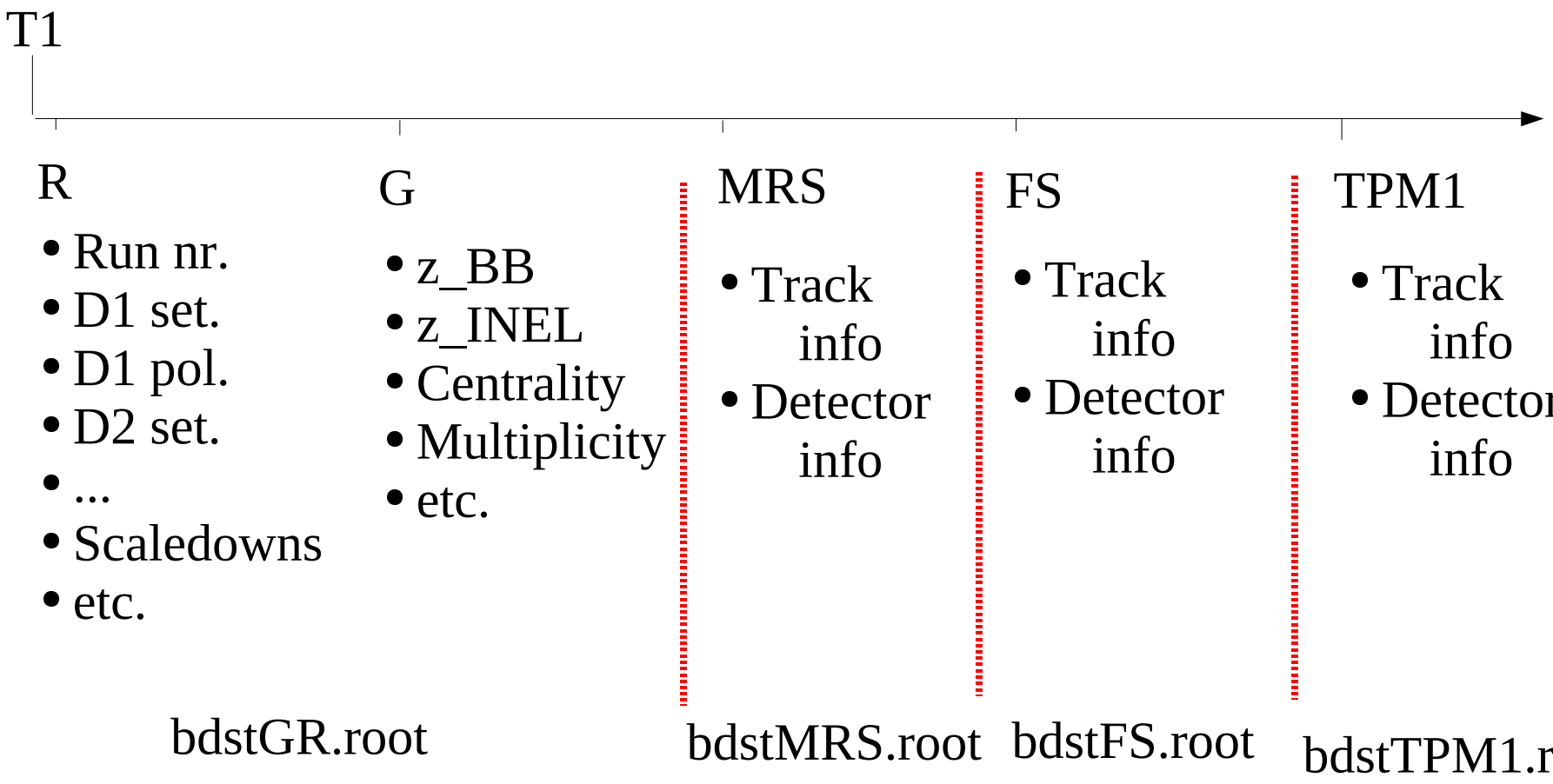
All in one file, bdstXXXX.root

1) Load tree T1 from bdstXXXX.root
2) T1->SetReadBranch(FS), T1->SetReadBranch(MRS), etc...
3) Read tree

B(E)DST
(BRAHMS Extended Data Summary Tree)



Collaboration meeting
Krakow, 4.-7. June 2003

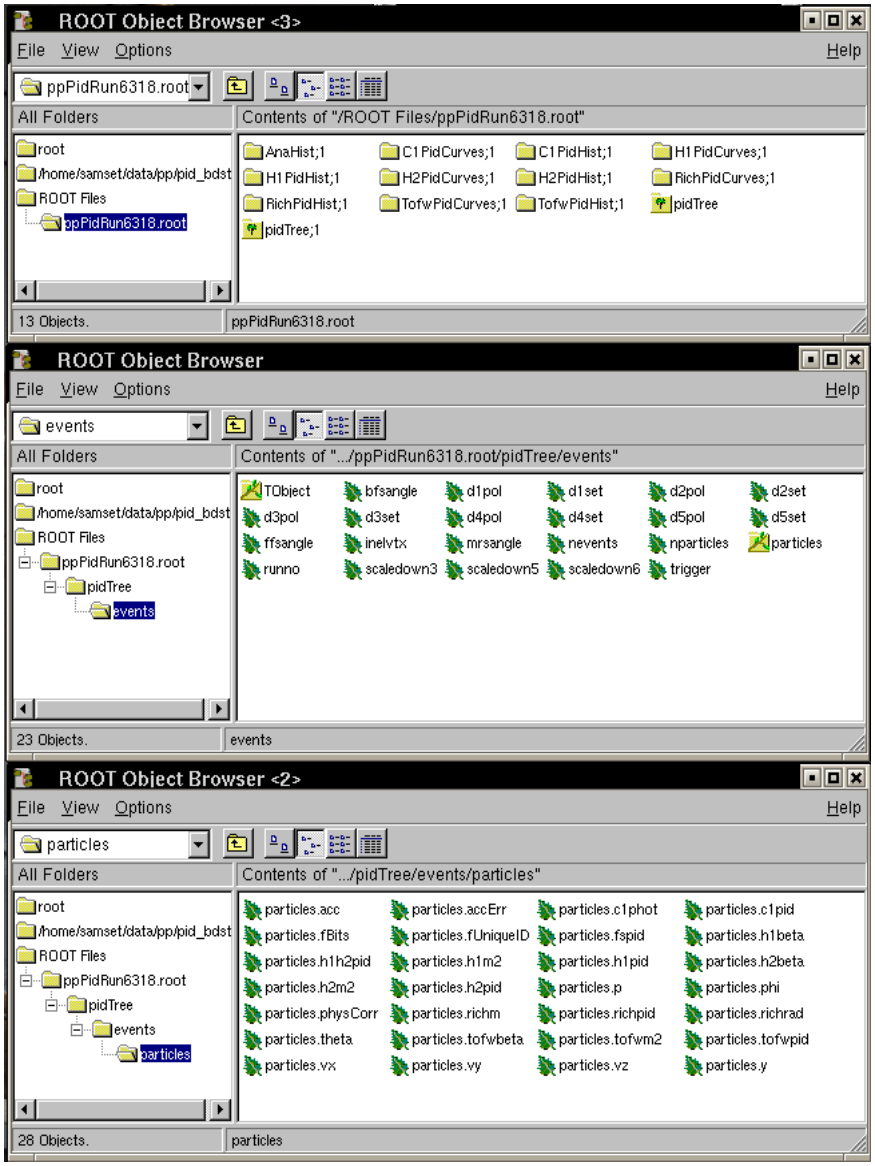


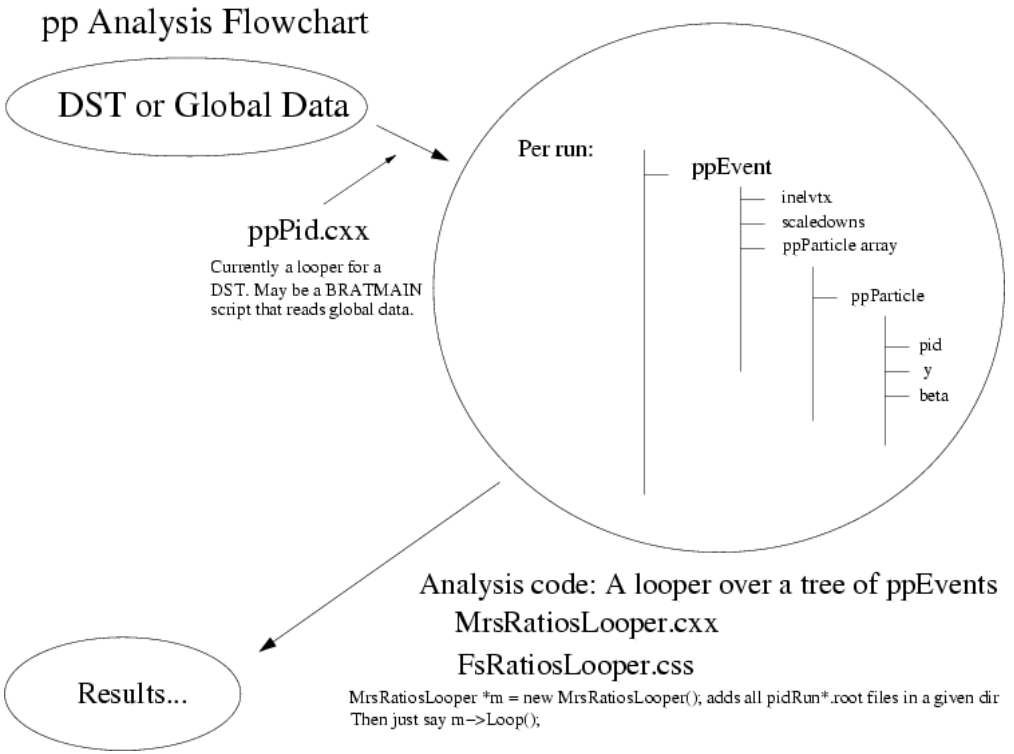
1) Load tree T1 from bdstGR.root
2) T1->AddFriend(FS), T1->AddFriend(MRS), etc...
3) Read tree

Analysis framework: FLAP

(Final Looping Analysis and Pid)

Collaboration meeting
Krakow, 4.-7. June 2003



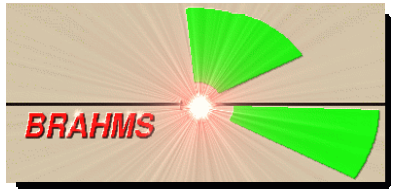


- Analysis is quick (~1hr for full 2001 dataset)
- Files are small (full dataset is 350Mb)
- bhs_app/flap/

Software and DSTs
Bjørn H. Samset

Software rules

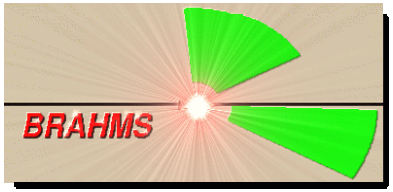
(NOT for the individual analyzer, for the rest of the world!)



Collaboration meeting
Krakow, 4.-7. June 2003

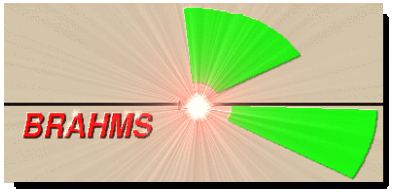
1) All analysis software up to and including DST generation is public, i.e. in BRAT or CVS. Also, we keep one or two *actively used* post-dst frameworks that are *well-documented* and *avaliable* in CVS outside nn_app.
2) All software is self-documenting, i.e. it saves its run settings in an ascii-file, root-file, whatever.
3) All software from local tracking to “public” ana software must have a README or doc.tex that explains usage and pitfalls, and names an expert to contact.

TODO



- Make BRAT modules self-documenting
- Expand bdst to include run info, pp/dAu specific stuff etc.
- Remove the old bdstAna?
- Claus: Put the new bdstAna and dstReader somewhere public and documented
- Bjørn: Put FLAP somewhere public and documented
- Restructure disk use on RCF so that we have official places for official files - and make or designate official files.

TODO



- Make BRAT modules self-documenting
- Expand bdst to include run info, pp/dAu specific stuff etc.
- Remove the old bdstAna?
- Claus: Put the new bdstAna and dstReader somewhere public and documented
- Bjørn: Put FLAP somewhere public and documented
- Restructure disk use on RCF so that we have official places for official files - and make or designate official files.