Endcap Hadronic Calorimeter Testbeam Software Framework

A Proposal: The hec_adc package

M. Lefebvre
D. O'Neil

University of Victoria
Victoria, B.C.

Presented May 7th 1997

Outline

Talk Outline

- Motivations and Goals
- Development and Implementation
- The hec_adc Package
  - Input testbeam data
  - Coding convention
  - hec_adc structure
  - Pass over all events
  - Include files
  - ADC Channel systems
  - Documentation
- Current Tasks
- A Proposal
- Call for Developers
- Future Direction?

Motivations and Goals

Motivations

- The need to facilitate **access** to testbeam data;
- Use as efficiently as possible our limited manpower resources.

Goals

- Provide a **simple** and clear common **framework** for HEC offline **code development** and analysis;
- Allow independent code development by users;
- Allow easy **implementation** of debugged user code.

Scope

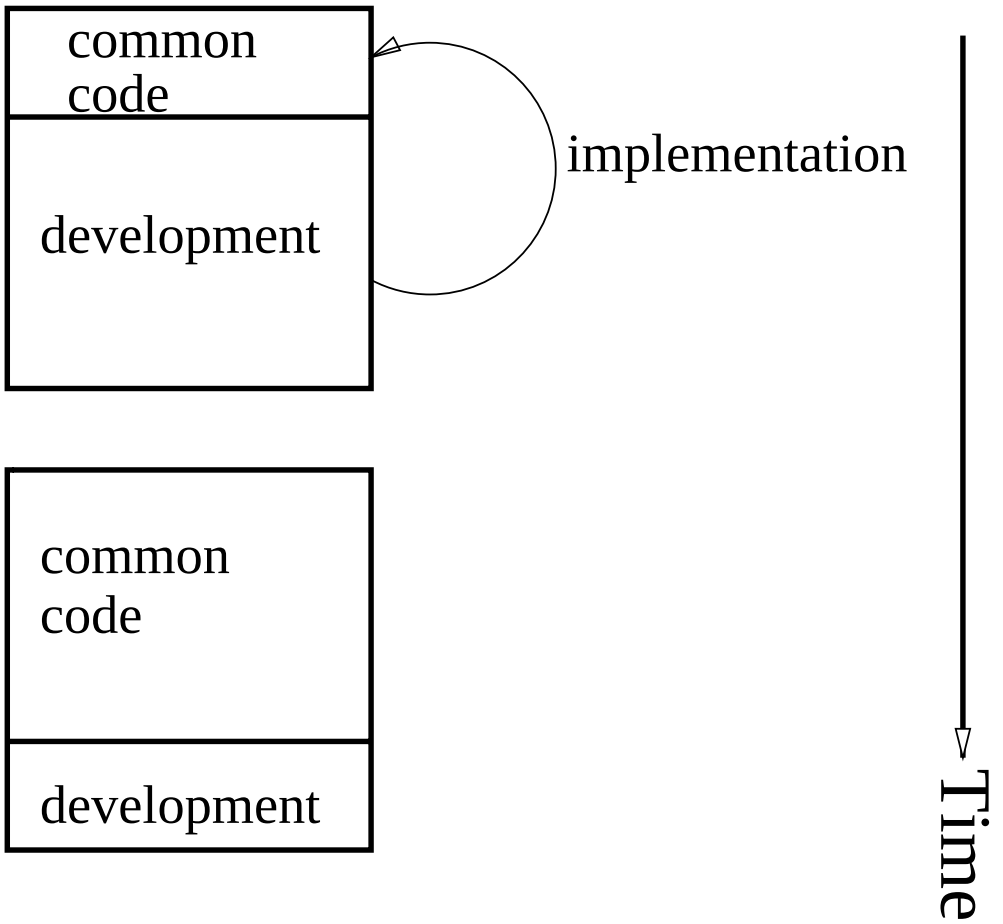
- Addresses the **immediate** HEC offline testbeam needs;
- Compatible with parallel efforts to develop longer term object oriented testbeam code.

An earlier version was presented at CERN on July 12th 1996.

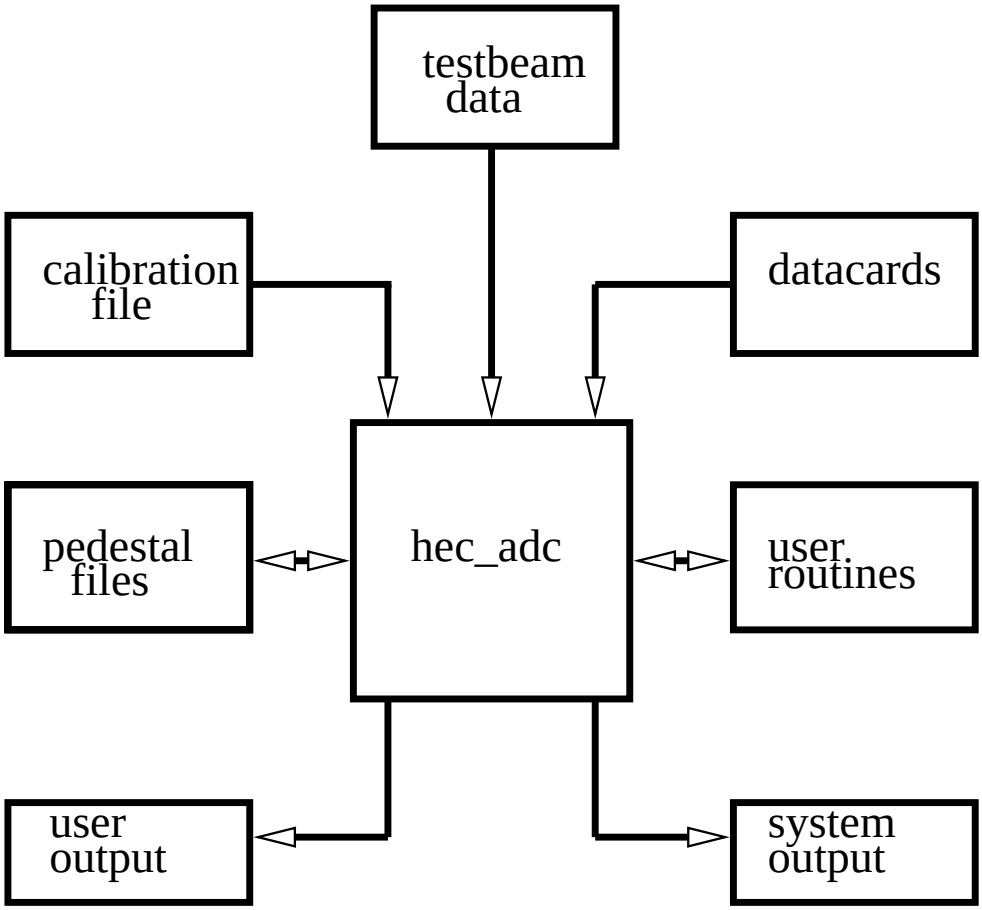
Development and Implementation

A coordinated software framework is needed to allow

- timely developement of user code;
- timely implementation of debugged user code of general interest.



The hec_adc Package



- Offline analysis and code development achieved through user routines;
- Package driven by datacards:

  - system datacards;
  - user datacards;

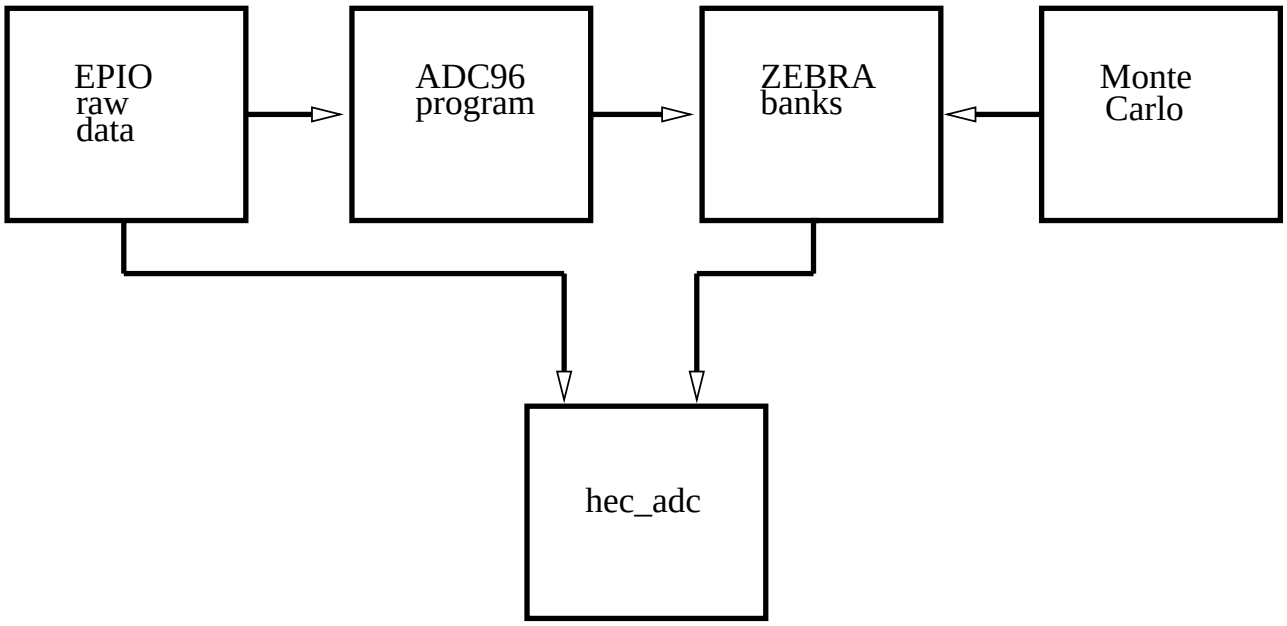
- Mature user routines can be easily incorporated in the hec_adc package by the code managers.

Input Testbeam Data

The hec_adc package accepts testbeam data in two formats:

- EPIO raw data
- ZEBRA obtained from the ADC96 (and soon ADC97!) code from Denis Salihagic et al. (Many thanks to Denis).

The ZEBRA format should in principle allow compatibility with Monte Carlo.



The banks already treated are

run header	RUNH;
event header	EVTH;
raw beam chamber data	DWPC;
raw ADC values	GEOE.

Other raw data banks can easily be made available.

Coding Conventions

- The FORTRAN language;
- The `include` statement;
- Coding conventions:

  lots of clear and correct comments;
  `implicit none`;
  `print` before `stop`;
  no automatic `save`;
  no `goto`;
  commons in documented files, one common per file;
  one routine per file;
  no compilation warnings on `atlas.wgs`.

- Code maintenance and distribution:

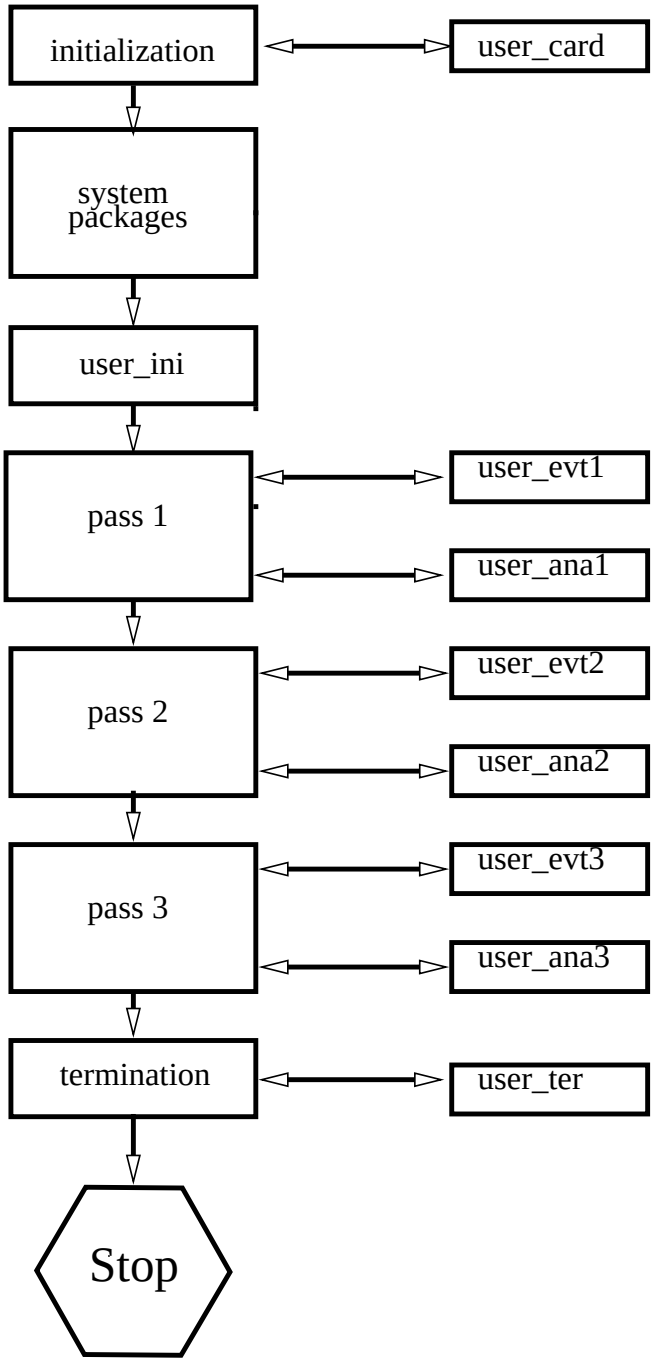
  unix `make`;
  `tar` and `gzip`;
  web page.

- Keep things simple for the user. No need to know about:

  CMZ;
  ZEBRA;
  EPIO;
  file formats.

- Execution time not critical.
  Concentrate on code clarity and transparence.

hec_adc Structure



General hec_adc package flow and user routines entry points.

hec_adc Structure (continued)

- Simple structure;
- No `ipass` variable;
- The user can request up to 3 passes over all events;
- Mature user routines can be made into system packages for all to use in the next code release;
- System packages currently available (for 1996 data):
  - pedestal package:

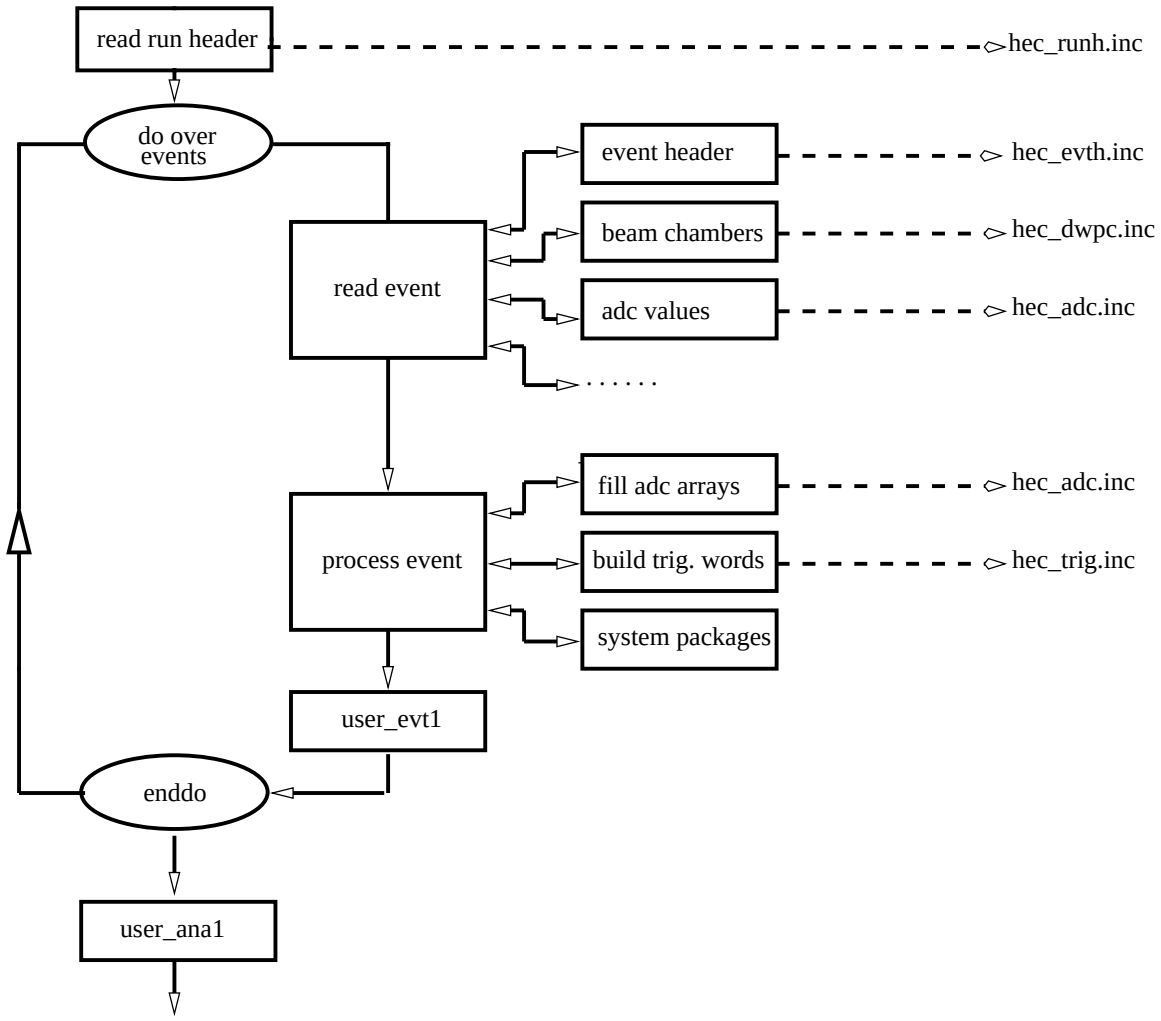
    read peds file;
    produce peds;
    output peds file.

  - sytem histogram package:

    provides default system histograms for quick data quality check.

Pass over all Events

Each user pass over all events has the following structure:



- Note the entry point of the user routines;
- User routines access data through system include files.

Include Files

User routines access data through system include files. They are self-documented: **read them carefully**.

- Include files relevant to all `user*.f` routines:

```
hec_buf.inc        zebra and paw commons;
hec_par.inc        fixed parms, also used by other *.inc files;
hec_datacard.inc   variables associated with datacards;
```

- Include files relevant after initialization:

```
hec_cal.inc        calibration coefficients (if read from file)
hec_geo.inc        geometry correspondence tables
```

- Include files relevant for each event of a run:
  - from system packages:

```
hec_ped.inc     adc pedestal and pedestal rms values
```

  - raw event data

```
hec_runh.inc    run header info (RUNH bank)
hec_evth.inc    event number and pattern units (EVTH bank)
hec_dwpc.inc    beam chamber data
hec_adc.inc     adc values
```

  - processed data

```
hec_adc.inc     adc values
hec_trig.inc    trigger words
```

ADC Channel Systems

Each readout cell (adc channel) can be identified by:

- adc channel system:

  `ic` = adc channel index

- Database system:

  `i_mod` = module number;
  `i_seg` = segment (family) number;
  `i_pad` = pad number (odd/even scheme);

- Physics system:

  `i_eta` = tower eta label;
  `i_phi` = tower phi label;
  `i_z` = tower depth number.

The relation between the Database system and the Physics system is run dependent in general.

For each event, the user has access to:

- The adc values for each adc channel in any of the above systems;
- Correspondence tables between these systems.
  (see `hec_geo.inc`).

Documentation

- Visit the hec_adc web page at

  `http://wwwhep.phys.uvic.ca/~uvatlas/hec_adc/hec_adc.html`

- or visit the directory

  `/afs/cern.ch/atlas/testbeam/HEC/offline/dev`

Current Tasks

- Adapt to the latest EPIO format;
- The adc97 code;
- Upgrade the histogram package for 1997 data;
- Provide a bad channel list to the user.

From this version onwards, we will attempt backward compatibility.

A Proposal to the HEC Community

- The hec_adc package is maintained on atlas.wgs and updated with mature user packages.
  (Michel Lefebvre, Dugan O'Neil);

Remarks:

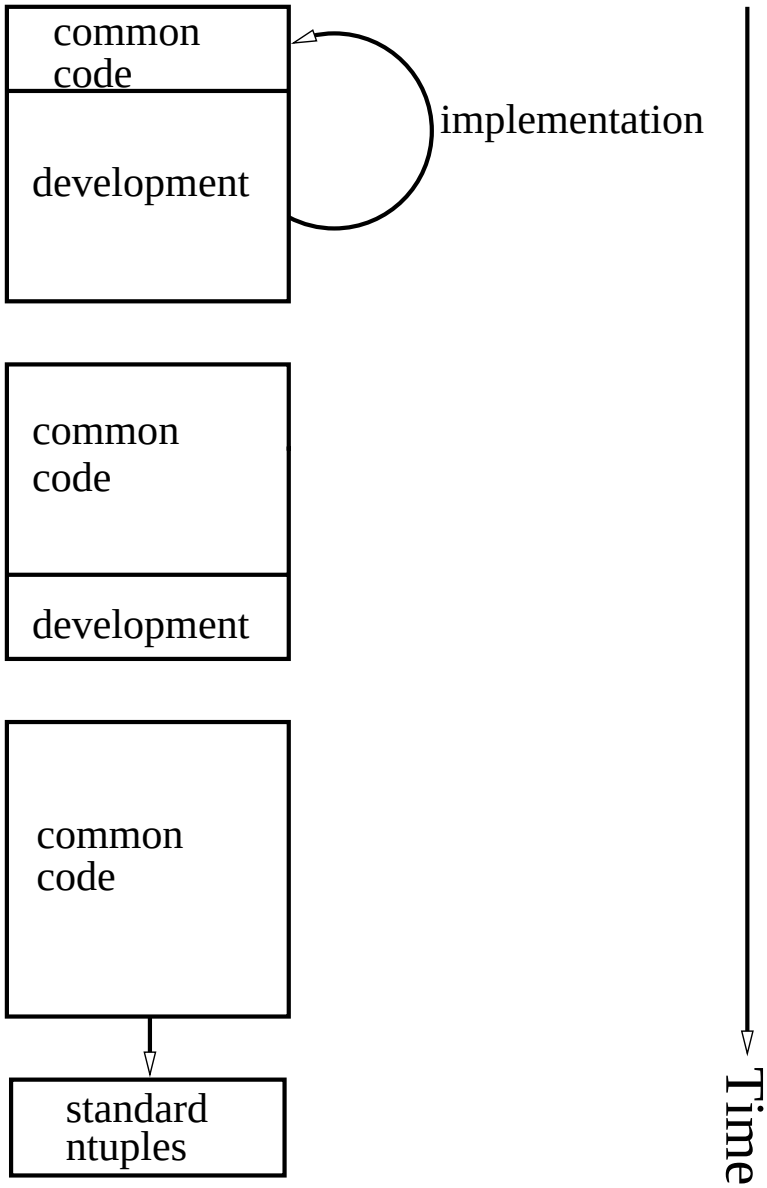
- The success of this package and, we believe, the efficiency of our efforts, require this package **to be used** by all as the framework for testbeam data access and analysis;
- The EPIO–to–ZEBRA program should be maintained to allow compatibility with Monte Carlo
  (Denis Salihagic et al);
- This proposal addresses the **immediate** HEC offline testbeam needs;
- This proposal is compatible with parallel efforts to develop longer term object oriented testbeam code.

Call for Developers

We should identify people that can take responsibilities in developing system packages:

- For each event, process raw beam chamber data into useful quantities;
- For each event, apply calibrations to adc values and fill nA arrays;
- For each event, find the signal height and timing from the multiple time sampling data;
- For each event, obtain pedestals from the first few time samplings;
- others?

Future Direction?



When sufficient development has taken place, we could consider the creation of a standard ntuple for each run. The hec_adc package could then allow the reading of many such ntuples for user analysis.

Acknowledgements

The hec_adc package greatly benefited from the advices of **Hans-Peter Wellish**, and from a heroic coding help from **Pierre Savard** for the first version released July 96.

Access to the data was and is made possible thanks to the crucial support of **Denis Salihagic**.

We are also indebted to **Dieter Striegel** and **Hasko Stenzel** for sharing with us their EPIO expertise.

- Visit the hec_adc web page at

  `http://wwwhep.phys.uvic.ca/~uvatlas/hec_adc/hec_adc.html`

- or visit the directory

  `/afs/cern.ch/atlas/testbeam/HEC/offline/dev`