The PrimEx Bank Definition Markup Language
PrimEx Note 17

Mark Ito

December 10, 2003

Abstract

This note describes an XML-based markup language for defining banks (BDML) for PrimEx analysis software. The C++ code for the header file which defines the bank structures and the initialization sequence for the banks are generated from an ascii BDML file by a perl script. A web page documenting the bank structure is also generated. Syntax of the BDML and procedures for using it are provided.

1 Introduction

Adding a new bank to the PrimEx software system requires modifications to the code in at least two places: in the bank definition header file, `src/libraries/include/bankdef.h`, and in the routine where each bank is initialized, `src/libraries/codaIO/PEvent.cc`. In addition, we will maintain documentation where the bank format is described, perhaps in more than one place. To insure consistency among these locations, we define each bank in a single place: `src/include/bankdef.xml` and generate the source code files and (in the future) documentation files from this source. To define the banks in `bankdef.xml` we use a custom mark-up language, the Bank Definition Markup Language (BDML) to specify all of the necessary information. This is a form of the Extensible Mark-up Language (XML[1]). This standard is used widely and is maintained by the World Wide Web Consortium[2].

For example, the BDML snippet

```
<bank name="hycalhit" comment="HYCAL hit bank (calibrated)">
<column name="id" type="int" comment="HYCAL block ID" />
<column name="E" type="float" comment="energy deposited in block (GeV)" />
</bank>
```

will generate the code snippet

```
/*----- start of hycalhit bank -----*/
/* hycalhit: HYCAL hit bank (calibrated) */

typedef struct {
  int id; /* HYCAL block ID */
  float E; /* energy deposited in block (GeV) */
} hycalhit_t;

typedef struct {
  bankHeader_t bank;
```

```
  hycalhit_t *hycalhit;
} primHYCALHIT_t;

/*----- end of hycalhit bank -----*/
```

in `bankdef.h` and the code

```
InitializeBank((void**)&banks.HYCALHIT, "HYCALHIT", sizeof(hycalhit_t), BANKFLAG_NONE);
```

in `PEvent.cc` (via an include file).

In addition, structures can be defined outside of banks. For example

```
<struct name="bankHeader_t" comment="this is a general bank header definition">
<member name="nrow" type="int" comment="number of rows" />
<member name="maxrows" type="int" comment="maximum number of rows" />
<member name="size" type="int" comment="number of 4-byte words in a row" />
<member name="name[16]" type="char" comment="bank name" />
<member name="flags" type="int" comment="Flags defined from above enum" />
</struct>
```

generates the code

```
/* ----- start of bankHeader_t structure -----*/
/* bankHeader_t: this is a general bank header definition */

typedef struct {
  int nrow; /* number of rows */
  int maxrows; /* maximum number of rows */
  int size; /* number of 4-byte words in a row */
  char name[16]; /* bank name */
  int flags; /* Flags defined from above enum */
} bankHeader_t;

/*----- end of bankHeader_t structure -----*/
```

in `bankdef.h`. For structures, no code is generated in `PEvent.cc`.

2 Making the Source Code

The directory `src/libraries/include` contains a makefile. When invoked it will create two files in that directory: `bankdef.h.body` and `bankinit.cc` using as input `bankdef.xml`, also found in `src/libraries/include`. It uses the perl script `scripts/make_bank_code.pl` to do this. The directory also contains `bankdef.h.header` and `bankdef.h.footer`. `bankdef.h` is a concatenation of `bankdef.h.header`, `bankdef.h.body` and `bankdef.h.footer`. This operation is also performed by the makefile. `bankinit.cc` is included in `PEvent.cc` (found in `src/libraries/codaIO`) using the standard C-preprocessor command.

3 Creating the Web Page

Most modern browsers are able to interpret the raw BDML file if that file references the appropriate XSLT[3] stylesheet. Here the stylesheet is src/libraries/include/bankdef.xsl. These files have been checked out in our web accessible area. To view the current bank definitions, point your web browser to

`http://www.jlab.org/primex/src/libraries/include/bankdef.xml`

Note that in this scheme, the html file does not exist on disk anywhere.

If your web browser does not support XSLT stylesheets, you can find a plain html version at

`http://www.jlab.org/primex/primex_notes/bankdef_xml/bankdef_mozilla.html`

4 Markup Language Syntax

In this section we describe all defined tags and their attributes.

4.1 `<bankdef>`

The top level tag. All BDML documents begin with `<bankdef>` and end with `</bankdef>`.

Parent tag: none.

Child tags: `<struct>`, `<bank>`.

4.2 `<struct>`

This tag defines a structure in bankdef.h.

Parent tag: `<bankdef>`.

Child tags: `<member>`.

Attributes are:

name The name of the structure.

comment (optional) A comment to appear in `bankdef.h` associated with the structure as a whole.

4.3 `<member>`

This tag describes a member of a structure.

Parent tag: `<structure>`.

Child tags: none.

Attributes are:

name The name of the member variable.

type The data type of the member variable, *e. g.*, `int`, `float`, `rawhit_t`, *etc.*

comment (optional) A comment specific to this variable.

4.4 `<bank>`

Describes a PrimEx bank. Each bank tag generates a structure definition in `bankdef.h` and in the appropriate line of bank initialization code in `bankinit.cc`.

Parent tag: `<bankdef>`.

Child tag: `<column>`.

Attributes are:

`name` The name of the bank.

`comment` (optional) Comment associated with the bank as a whole.

`typedef` (optional) If this attribute is specified, the bank is based entirely on a previously defined structure, just with a different name. The value of the attribute is the name of the type of the defining structure.

`size` (required if `typedef` specified) The name of the structure which defines a row of the bank on which the current bank is based. Used to specify the size of a bank row for initialization.

`bankflag` (required if `typedef` specified) Specifies the bankflag to be used during initialization.

4.5 `<column>`

Specifies one column of a row of bank.

Parent tag: `<bank>`.

Child tags: none.

Attributes are:

`name` The name of the column.

`type` The data type of the column, *e. g.*, `int`, `float`, `rawhit_t`, *etc.*

`comment` (optional) A comment specific to this column.

4.6 `<CVSID>`

Parent tag: `<bankdef>`.

A convenient place to put version control information, RCS keywords in particular.

Child tags: none.

No attributes.

References

[1] A simple introduction to XML syntax can be found at http://studio.tellme.com/general/xmlprimer.html.

[2] The World-Wide Web Consortium's homepage is at http://www.w3.org.

[3] Extensible Stylesheet Language Transformations. See http://www.w3.org/TR/xslt.

CVS information:

$Id: bankdef_xml.tex,v 1.11 2003/12/10 22:40:39 marki Exp $