

From: Bob Payne
Date: Tue, 24 Mar 92 11:36:17 EST

aips++ documentaion:

Some time ago now Chris Flatters proposed a class documentation standard. There are several excellent suggestions in that document however it makes several design decisions which I would do differently. The use of C style comment fields and the tagging scheme for selecting what is being documented seems more than is needed. I would prefer the use of C++ comment delimiters and have the extractor generate documentation based on the actual code rather than on comment fields which may not reflect changes to the code or may fail to document items in the code.

There is a public domain documentation extractor for C++ code called "genman". It is a GNU awk script which looks for C++ keywords and generates ascii or man page documentation from these keywords. While this program may need to be enhanced I believe it close to what is needed and may be better at extracting documentation from code which does not meet the standards proposed in Chris's document.

In any case we will soon need decisions on coding style and documentation standards for production aips++ code.

Here is the suggested template file for genman:

```
//-----  
//  
//          foo.h  
//  
  
#ifndef FOO_H  
#define FOO_H  
  
// .NAME foo - a class to manage foo objects  
// .LIBRARY Base  
// .HEADER General utility routines  
// .INCLUDE base/foo.h  
// .FILE foo.cxx  
// .FILE foo.h  
  
// .SECTION Description  
// A paragraph describing the class the include file defines.  
// Use a blank line to terminate the paragraph.  
  
// .SECTION Caveats  
// A paragraph describing unusual features of the class.  
// This paragraph may be omitted.  
  
#ifndef STDIO_HXX  
#include <stdio.hxx>  
#endif  
  
class foo  
{  
private:  
    int *bar;  
    // Pointer to character string struct.  
  
protected:  
  
public:  
    foo( const char *str= 0 );  
    ~foo();  
};  
  
#endif // FOO_H
```

and the extracted documentation:

Mar 24 09:09

NAME

foo - a class to manage foo objects

SYNOPSIS

```
#include <base/foo.h>
```

```
class foo
```

Public members

```
foo( const char *str= 0 );  
~foo();
```

Private members

```
int *bar;  
    Pointer to character string struct.
```

DESCRIPTION

A paragraph describing the class the include file defines.
Use a blank line to terminate the paragraph.

CAVEATS

A paragraph describing unusual features of the class.
This paragraph may be omitted.

DEFINED MACROS

```
FOO_H
```

INCLUDED FILES

```
<stdio.hxx>
```

SOURCE FILES

```
foo.cxx  
foo.h
```

SUMMARY