Doc. No. : WG21/N0958
X3J16/96-0140
Date: July 27, 1996
Project: Programming Language C++
Reply to : Jerry Schwarz
jerry@intrinsa.com

Iostream Action Items -- Jerry Schwarz

The following paragraphs contain exact wordings for proposals that were approved in principle by the iostream wg at the Menlo Park meeting.

27-404 Checking for null

27-406 Extractor Failure

Editorial clarification of order of operations is present in current WP. However, the edited paragraph now has some typo's it should read:

> A null byte `(traits::eos())` is stored in the next position, which may be the first position if no characters were extracted. Then call `width(0)`.

27-408 ws manipulator

Ws should not set failbit when it reaches end-of-file. But the way the effects paragraph is written it does because the sentry constructor does. The proposed words are copied from the words in

> **Effects**: Extracts characters as long as the next available character `c` is a whitespace or there are no more characters in the sequence. Whitespace characters are distinguished with the same criterion as used by `sentry::sentry`. [lib.istream::sentry]. If `ws` stops extracting characters because there are no more it sets `eofbit`, but not `failbit`.

27-410 Description of putback

This is an editiorial change. Description of `putback` (27.6.1.3) refers to `sputback`. The should be `sputbackc`.

27-411 getline failures

`getline`[27.6.1.3] should not fail when the line is empty. That is, it fails when it extracts no characters, not when it stores no characters. The working paper currently says that, so no change is required.

27-412 basic_streambuf extractor with null argument

This extractor `operator>>(basic_streambuf<charT,traits>* sb)` [27.6.1.2.3] should set `failbit` when `sb` is null, not badbit. Badbit should only be set when the stream itsel f is bad. The first sentence of the effects paragraphs should be:

> If sb is null, calls setstate(failbit), which may throw ios_base::failure.

27-503 null streambuf's

27-606 seekg/seekp should have value parameter

This has been corrected in the current working paper because the type had been approved in Santa Cruz.

27-652 smanip is not a single type

[27.6.3] This section was written assuming more about smanip than is stated.

I propose to replace the second paragraph (which begin "In describing functions ..." with:

> The type designated smanip in each of the following function descriptions is implementation specified and may be different for each function.

The first sentence of the returns section of each of these function s(`restiosflags`, `setiosflags`, `setbase`, `setfill`, `setprecision`, `setw`) descriptions is "*smanip*(f,mask) where f can be defined as ..." Replace those sentences by :

> An object `s` of implementation specified type such that if `out` is an (instance of) `basic_ostream` then the expression `out<<s` behaves as if `f(s)` were called where `f` could be defined as

27-802 filebuf::is_open

Propose replacing "true if the associated file is available and open" in 27.8.1.2 with

> **Returns:** true if a previous call lto open succeeded (returned a non-null value)and there has been no intervening call to close.

27-807 filebuf::uflow

The current WP says that filebuf overrides underflow, but not uflow. It isn't clear to me that the WP should be specific about which virtuals are overridden, but it is clear that underflow and uflow ought to be treated the same way. I propose to add a member function declaration to 27.8.1.1

```
virtual int_type uflow();
```

and and a description to 27.8.1.4

```
int_type uflow()
```

> **Effects**: Behaves according to the description of `basic_streambuf<charT,traits>::uflow` with the specialization that a sequence of characters is read from the input sequence with the same method as used by `underflow`.

27-810 editorial change to declarations

Change the declaration of the basic_ifstream constructor in 27.8.1.5 to

explicit basic_ifstream(const char* s, ios_base::openmode mode = ios_base::in);

and similarly change the basic_ofstream constructor in 27.8.1.8 to

explicit basic_ofstream(const char* s, ios_base::openmode mode = ios_base::out);

27-813 filebuf::overflow needs to specify return value

Current WP specifies a return value on failure, but not on success. Replace the returns clause by

> **Returns**: If underflow succeeds it returns `traits::not_eof(c)` otherwise it return `traits::eof()` to indicate failure.
>
> **Notes**: If `is_open == false`, the function always fails.

27-909 Typos in footnote

footnote 228 (in 27.1.2) should read

> It is usually a synonym for one of the signed basic integral types whose representation is at least as many bits as type `long`.