| | |
|---|---|
| Doc No: | X3J16/96-0096 |
| | WG21/N0914 |
| Date: | May 23, 1996 |
| Project: | Programming Language C++ |
| Reply-To: | Stephen D. Clamage |
| | stephen.clamage@eng.sun.com |


# IOstreams WG Recommendations

Re: IOstream issues list X3J16/96-0079 WG21/N0897 of May 15, 1996.

Members of the IOstream library working group met informally on 21-22 May to review all the outstanding Chapter 27 issues. Recommendations of this group are given below. Each issue, except for those left open, was assigned to a person who will present a complete recommendation with WP language to the full IOstream working group at Stockholm. The issues left open were deemed in general too difficult to be settled by the attendees. (Note: issue numbers are not all consecutive. No issues have been omitted.)

In addition, we reviewed every editorial box. Those boxes which were not closed and not already in the issues list were added to the list. See document X3J16/96-0099 (WG21/N0917).

Attendees:

Jerry Schwarz (JS)
Tom Plum (TP)
Nathan Myers (NM)
Philippe Le Mouël (PL)
Steve Clamage (SC)

Issues:

27-001 `traits::newline` and locale (NM)
Recommend: Change description of `getline` so there are two variants instead of one with a default argument.

27-002 `traits::is_whitespace` (closed)
Made moot, since `is_whitespace` doesn't exist any more.

27-004 `is_whitespace` example (closed)
Made moot, since `is_whitespace` doesn't exist any more.

27-005 `not_eof` specification (TP)
Recommend: Yields a value which is not equal to the locale's eof. The value is unspecified. (The WP requires that `(not_eof(e)==e)` if `(e!=eof),` and forces a change from existing practice. The recommendation allows existing implementions to remain unchanged.)

27-007 `ios_traits` typedefs (closed)
Already fixed by unification of traits adopted in Santa Cruz.

27-008 `ios_traits::length` definition (closed)
Already fixed by unification of traits adopted in Santa Cruz.

27-009,010 `get_state`, `get_pos` definition (JS)
No consensus on these issues, JS will bring a proposal along these lines:
• Add a new trait type: `char_traits::upos_type, with` no constraints on the type.
• Replace `get_pos` with: `pos_type make_pos(upos_type, state_type);` because the function creates a position object.
• Add a new function: `upos_type get_pos(pos_type);`

27-011 Return type for `ios_traits::copy` (closed)
Made moot since function no longer exists.

27-106 class `Init` (closed)
Recommend no change to WP. (JS might add an example showing why Init is needed.)

27-107 `ios::failure` functionality (closed)
Recommend no change to WP, because stream lifetime might end before lifetime of the reference.

27-203 `operator bool` (open)
Depends on final resolution of `bool` properties.

27-205 `imbue` calling `rdbuf()->pubimbue` (open)
Deferred to Stockholm.

27-206 (new issue, related to 27-503): (PL)
Problem: Can use `clear(0)` to unconditionally clear error state, even if `rdbuf()` returns null. The stream then appears to be in a good state, which is wrong.
Recommend: `clear()` function should set `badbit` (independent of its argument) if `rdbuf()` returns null. Note: This means `clear()` must not be moved to `ios_base`; it must remain in `basic_ios` so it can find out if there is a streambuf.

27-301 imbuing on streambufs (NM)
Recommend: No standard function defined in Chapter 27 calls `imbue` directly, except for the `imbue` function. If any user code (e.g. if called via a virtual function) calls `imbue` during the exectuion of any of these functions, the effect is undefined. (Can't guarantee integrity of the stream otherwise.)

27-404 checking for null streambuf. (JS)
Refer to 27-503.

27-405 confusing description of formatting (closed)
Made moot by rewrite of section.

27-406 `charT*` extractor failure (JS)
Editorial, fixed.

27-407 `charT` extractor failure (closed)
The definition of *extract* (27.6.1.1) covers setting eof. Note: footnotes could be added referring to that section. (Editorial)

27-408 `ws` manipulator (JS)
Recommend: Explanation needs a rewrite. Intent is clear: Extracts any whitespace. Sets `eofbit` if eof is reached.

27-409 `unsigned short` extractor mechanism (closed)
Already fixed.

27-410 description of `putback` (JS)
Editorial: fixed a typo.

27-411 `getline` failure (JS)
Recommend: `getline` fails if no characters are extracted. An empty line is not a failure. If WP doesn't say that, it should be changed.

27-412 `basic_streambuf` extractor with null argument (JS)
Recommend: Set `failbit`, not `badbit`.

27-501 `char` inserter formatting, padding, width (open)
The draft says (contrary to the issues list claim) no formatting. A proposal is pending regarding `char` *vs* `charT` inserters. This issue is left open for Stockholm.

27-503 checking for null streambuf (JS)
Editorial: point out where the check is not necessary. In places where check is required, recommend add the check.

27-505 `unsigned long` inserter (closed)
Made moot by rewrite of that section of the WP.

27-506 padding *vs* adjustment (closed)
Made moot by rewrite and moving discussion of fill padding to Chapter 22 (Localization).

27-601 `ios_base` manipulators (PL)
Recommend: Accept resolution in issues list.

27-602 typedefs for `pos_type` and `off_type` (closed)
Recommend: No change to WP. Does no harm to leave in the typedefs, and probably they are required by language rules anyway.

27-603 `read`/`write` functions (closed)
Recommend: No change to WP. If `read`/`write` take `void*`, what does it mean to write bytes to a wide stream? Here it is clear that you always write sequences of `charT` to a stream.

27-604 `ios::in`, `ios::out` required in `open` or ctor? (SC)
Recommend ctors/`open` for iXstream, oXstream always assumes the `in` or `out` bit. Ctors/`open` for Xstream (bidirectional) do not assume either `in` or `out` bits.

27-605 allow iterators in `get`/`put` functions (closed)
This is a proposed extension, recommend no change to WP.

27-606 `seekg`/`seekp` should have value parameter (JS)
Yes, editorial, make declaration and definition agree.

27-651 `setfill` description (open)

At issue is the `fill` manipulator and what parameter type it should accept: Certainly `charT`. What about `char` or `int`? Problem is lack of a correct implicit “widen” from `char` to `charT`, because the conversion requires knowledge of the locale. Defer to Stockholm.

27-652 `smanip` is not a single type (JS)

Editorial fix to wording, along the lines of:
Returns: A value of some class `smanip` such that

```
out << resetiosflags(m)
```

has the effect of

```
out.setf(0,m)
```

27-70x: First six issues depend on defining a consistent model for stringstreams. Should stringbufs be more like filebufs or more like strstreams? Probably more like files. The stringstream section needs a full review and possibly some rewriting to fit the model. (PL + review by JS)

27-701 return value of `basic_stringbuf::str`
27-702 String streams need allocator and traits params
27-703 `stringbuf` postconditions
27-704 `stringbuf` ctor
27-705 calls to `setg`/`setp` in Table 14
27-706 calls to `setg`/`setp` in Table 16

27-707 `setbuf` for `basic_stringbuf` (SC)

Recommend: Effects: implementation-defined, except that `setbuf(0,0)` has no effect, returns `this`. Return type in WP is correct.

27-801 `filebuf::underflow` (done)

Editorial, already done.

27-802 `filebuf::is_open` (JS)

Recommend: Resolution in issues list.

27-803,804 ofstream ctor missing `trunc` in default arg (SC)

Recommend: Fix words and Table 14 to show that `out` by itself is equivalent to `out|trunc`. Then `out|trunc` can be changed to `out` everywhere as editorial. All possible combinations of flags should be considered and given a meaning or declared invalid or undefined.

27-805 `filebuf::imbue` semantics (open)

Resolve in Stockholm.

27-806 `filebuf::seekoff` description (TP)

Recommend: Accept in principle the recommendations in the public comment. Eight-bit streams behave like stdio `FILE`s. For wide streams, seeks are allowed provided the external file meets certain criteria to be worked out by TP.

27-807 `filebuf::underflow` performance (JS)
Recommend: `Filebuf` overrides both `uflow` and `underflow`, instead of just `underflow`. Semantics are left as-is.

27-808 editorial (done)
Make it clear that although `FILE` is a typedef in C, that is not necessarily so in C++.

27-809 `setbuf` for `filebuf` (SC)
Recommend: Effects: `setbuf(0,0)`, if called before any I/O occurs on the stream, makes the stream unbuffered, returns `this`. Otherwise, the effects are implementation-defined. (Define *unbuffered* in terms of `underflow`/`uflow` and `overflow`.) Return type in WP is correct.

27-810 `openmode` notation (JS)
`trunc` issue handled by 27-803,804. Remainder is editorial.

27-811 missing `filebuf::sync` (TP)
Recommend: For output, perhaps adapt wording from C standard. (Intent is clear.) For in/out, flush output then really seek to current file position. For input files, implementation-defined.

27-812 `filebuf::overflow` example (done)
Editorial.

27-813 `basic_filebuf::overflow` return value(JS)
Recommend: Returns `not_eof(c)` on success.

27-814 `basic_filebuf::imbue` (open)
To be resolved in Stockholm.

27-815 `seekpos` description (PL)
Recommend: Sets file position to value of its argument, which must have come from `seekoff(0)`.

27-816 [`i`|`o`]`fstream::open` using `is_open` (PL)
Recommend: As in proposed resolution.

27-901 I/O of `unsigned charT` (open)
deferred until Stockholm.

27-904 `<iosfwd>` declarations (NM)
Recommend: Add decls for bidirectional streams: `fstream`, `iostream`, `strstream`, `stringstream`. Add decl for `streampos`. (See 27-910)

27-906 typedef for traits (NM)
Recommend: as in issues list.

27-907,908 editorial
Done.

27-909 mostly editorial (JS)
#3 done, #2 now moot, #1 to be done.

27-910 `streampos` type (TP)
Recommend: Resolve as in issues list, and add `streampos` to `<iosfwd>`.

27-911 stdio synchronization (NM)
Recommend: Synchronization should be the default (status quo). NM will write proposal about adding in synch/unsynch functions, and will discuss when synch/unsynch can be called.

27-912,913 “Notes” clauses. (JS)
Recommend: Non-normative notes to “[Note:” style, change normative “Notes” to some other heading word.

27-914 rethrowing exceptions (JS)
We want to add an example (editorial).

27-915 “instance” vs “specialization” (open)
Editorial, but more work is needed to preserve distinction between explicit and implicit specialization.

27-916 missing descriptions of specializations (done)
Proposal is subsumed by proposed change to `<iosfwd>` (27-904). Recommend: No change to WP.

27-918 editorial (done)
Added a footnote.

27-919 Table 2 assertions (PL)
Larger issue: This table was voted out of the iostream chapter as part of traits consolidation, but needs to be somewhere. The discussions of `OFF_T` and `POS_T` should be consolidated in iostream chapter, with a note added to string chapter referring to this chapter. Re this question, recommend accept proposal in 27-919.

27-920 destination of `clog` and `wclog` (no change)
Recommend: No change to WP, delete the editorial box.

27-921 default locale argument to ctor (no change)
Recommend: No change to WP.

27-1001 `setbuf` for `strstreambuf` (SC)
Recommend: Effects: implementation-defined, except that `setbuf(0,0)` has no effect, returns `this`. Return type in WP is correct.

27-1002,3,4 (JS)
Editorial corrections already made. Recommend: Add typedefs as in 1002.