Iostreams WP changes for London

--- copyof-1\lib-io~1   Tue Jul 15 17:24:32 1997
***************
*** 142,157 ****
   namespace std {
   .DE
   .Cb
   - ." CD2-27-004. 27.2
   - ." Add forward declaration of traits and allocator class
   - templates   template<class charT> class char_traits;
   template<> class char_traits<char>;
   template<> class char_traits<wchar_t>;
   .Ce
   .Cb
   - template<class T> class allocator;
   - .Ce
   - .Cb
   template <class charT, class traits = char_traits<charT> >
   class basic_ios;
--- 142,151 ----
***************
*** 460,469 ****
   cin.tie() returns
   .CW &cout .
   - ." CD2-27-035. 27.3.2. Make explicit initialization of standard streams.
   - Its state is otherwise the same as required for
   - .CW ios_base::init
   - (_lib.basic.ios.cons_).
   ."----
   .ix "[cout]"
   .Pb
--- 454,459 ----
***************
*** 500,509 ****
   is initialized,
   .CW "cerr.flags() & unitbuf"
   is nonzero.
   - ." CD2-27-035. 27.3.1. Make explicit initialization of standard streams.
   - Its state is otherwise the same as required for
   - .CW ios_base::init
   - (_lib.basic.ios.cons_).
   ."----
   .ix "[clog]"
   .Pb
--- 490,495 ----
***************
*** 518,523 ****
--- 504,513 ----
 declared in
   .CW <cstdio> (_lib.c.files_).
+ .eN
+ This leaves other parts of the state unspecified. I believe they should be initialized as
+ if by init.
H3 "Wide stream objects" lib.wide.stream.objects

***************
*** 542,551 ****
  .CW wcin.tie()
  returns
  .CW &wcout.
  ." CD2-27-035. 27.3.2. Make explicit initialization of standard streams.
  - Its state is otherwise the same as required for
  - .CW ios_base::init
  - (_lib.basic.ios.cons_).
  ."----
  .ix "[wcout]"
  .Pb
--- 532,537 ----
***************
*** 582,591 ****
  is initialized,
  .CW "wcerr.flags() & unitbuf"
  is nonzero.
  ." CD2-27-035. 27.3.2. Make explicit initialization of standard streams.
  - Its state is otherwise the same as required for
  - .CW ios_base::init
  - (_lib.basic.ios.cons_).
  ."----
  .ix "[wclog]"
  .Pb
--- 568,573 ----
***************
*** 615,621 ****
  namespace std {
    typedef \f4OFF_T\fP  streamoff;
    typedef \f4SZ_T\fP streamsize;
  ." CD2-27-013. 27.4. Put fpos into synopsis of <ios>
  template <class stateT> class fpos;
--- 597,602 ----
***************
*** 810,823 ****
  ! ." UK 634. 27.4.2 make ios_base destructor virtual
  ! virtual ~ios_base();
  .Ce
  .Cb
  \f6// destructor \fP
! ." CD2-27-018. 27.4.2. Suppress exceptions
!  typedef void (*event_callback)(event, ios_base&, int \f6index\fP) throw();
  void register_callback(event_call_back \f6fn\fP, int \f6index\fP);
  .Ce
  .Cb
--- 791,802 ----
  ! ." CD2-27-018. 27.4.2. Suppress exceptions
  !  typedef void (*event_callback)(event, ios_base&, int \f6index\fP) throw();
  void register_callback(event_call_back \f6fn\fP, int \f6index\fP);
public:
  explicit failure(const string& msg);
  virtual ~failure();
  virtual const char* what() const throw();
};

--- 874,880 ----
public:
  explicit failure(const string& msg);
  virtual ~failure();
  virtual const char* what() const;
};

--- 896,902 ----
initializing the base class with exception(msg)
Postcondition:
  strcmp(what(), msg.c_str()) == 0

--- 1348,1353 ----
ios_base();
Effects:
The ios_base members are have an indeterminate value after construction.
--- 1460,1468 ----
ios_base();
Effects:
The ios_base members are have an indeterminate value after construction.
member function called from within
.I fn
has well defined results.
+ .eN
+ Should the fpos constructor be explicit?
+ .nE
\"==
.H3 "Template class \&\f7fpos\fP\&" lib.fpos
.ix "[fpos]"

***************
*** 1513,1520 ****
    template <class stateT> class fpos {
  public:
    \f6// _lib.fpos.cons_ Constructors\fP
      .cw CD2-27-036. 27.4.3
    !     explicit fpos(stateT);
    \f6// _lib.fpos.members_ Members\fP
      stateT state() const;
    void state(stateT);
  template <class stateT> class fpos {
    public:
      \f6// _lib.fpos.cons_ Constructors\fP
        .cw CD2-27-036. 27.4.3
    !     fpos(stateT);
    \f6// _lib.fpos.members_ Members\fP
      stateT state() const;
    void state(stateT);


***************
*** 1575,1591 ****
  .cw o
  refers to a value of type
  .cw streamoff ,
  -- .cw CD2-27-012. 27.4.4 Interconvertibility of streamsize and OFF_T.
    .li
  -- .cw sz
  - refers to a value of type
  - .cw streamsize
    and
    .li
    .cw i
  refers to a value of type
  .cw int .
  -- .cw CD2 editorial. Replace FPOS_T with fpos
    .ts "Position type requirements"
        .na
    .ts
  -- 1552,1562 ----

***************
*** 1602,1608 ****
    P p(i);
    P p = i;                      post: \&\f5p == P(i)\fP\&
    --
    ! P(o) fpos    converts from offset
    --
    O(p) OFF_T converts to offset  \&\f5P(O(p)) == p\&\fP
    --
    1573,1579 ----
    P p(i);
    P p = i;                      post: \&\f5p == P(i)\fP\&
    --
    ! P(o) POS_T converts from offset
O(p) OFF_T converts to offset \&fP(O(p)) == p\&fP

***************

*** 1610,1627 ****

p != q \&f1convertible to\&fP\& bool \&f5!(p==q)\&fP\&

! q = p + o fpos + \&f1foffset\&fP\& \&f5q-o == p\&fP\&

p += o

! q = p - o fpos - \&f1foffset\&fP\& \&f5q+o == p\&fP\&

p -= o

o = p - q OFF_T \&f1distance\&fP\& \&f5q+o == p\&fP\&

***************

--- 1581,1593 ----

p != q \&f1convertible to\&fP\& bool \&f5!(p==q)\&fP\&

! q = p + o POS_T + \&f1foffset\&fP\& \&f5q-o == p\&fP\&

p += o

! q = p - o POS_T - \&f1foffset\&fP\& \&f5q+o == p\&fP\&

p -= o

o = p - q OFF_T \&f1distance\&fP\& \&f5q+o == p\&fP\&

.\" USA & Germany. CD2-27-012. Interconvertibility of streamsize and OFF_T.
- streamsize(o) streamsize converts \&f5streamsize(O(sz)) == sz\&fP\&
- O(sz) OFF_T converts \&f5streamsize(O(sz)) == sz\&fP\&

--- 1693,1698 ----

***************

*** 1876,1884 ****

 protecting:
 basic_ios();
 void init(basic_streambuf<charT,traits>* \f6sb\&fP);
The previous value of fill().

\ix "[basic__ios] [copyfmt]"

\begin{Verbatim}
\texttt{basic\_ios\& copyfmt(const basic\_ios\& \texttt{rhs});}
\end{Verbatim}

***************

*** 1960,1967 ****

\begin{Verbatim}
\texttt{iostate rdstate() const;}
\end{Verbatim}

** La Returns:

! \" CD-27-025. 27.4.5.3
! The error state of the stream buffer.
\"
\ix "[basic\_ios] [clear]"
\begin{Verbatim}
\texttt{basic\_ios\& copyfmt(const basic\_ios\& \texttt{rhs});}
\end{Verbatim}

***************

*** 1918,1924 -----

\begin{Verbatim}
\texttt{iostate rdstate() const;}
\end{Verbatim}

** La Returns:

! \" The control state of the stream buffer.
\"
\ix "[basic\_ios] [clear]"

***************

*** 2568,2575 ****

** LI

\begin{Verbatim}
\texttt{the}
\end{Verbatim}

\ix " Editorial Japan

! member to a copy the global locale,
\ix " locale() ,

at the time of construction.

** La Notes:

--- 2525,2531 -----

** LI

\begin{Verbatim}
\texttt{the}
\end{Verbatim}

\ix " member to to a copy the global locale,
\ix " locale() ,

at the time of construction.

** La Notes:

***************

*** 2941,2949 ****

\begin{Verbatim}
\texttt{H5 "Get area" lib.streambuf.virt.get
\"
\ix "[basic\_streambuf] [showmanyc]"
\end{Verbatim}

- \" CD2-27-009. 27.5.2.4.3
\ix "[basic\_streambuf] [showmanyc]"
\begin{Verbatim}
\texttt{int showmanyc();}\*f
\end{Verbatim}

** La Notes:

--- 2897,2904 -----

\begin{Verbatim}
\texttt{H5 "Get area" lib.streambuf.virt.get
\"
\ix "[basic\_streambuf] [showmanyc]"
\end{Verbatim}

** La Notes:

***************

*** 3484,3495 ****

\begin{Verbatim}
template<class charT, class traits>
\texttt{basic\_istream\langle charT, traits\rangle\& operator\rangle(basic\_istream\langle charT, traits\rangle\&, charT*)};
\end{Verbatim}

- \" CD2-27-020. 27.6.1.1. Correct type of args.

***************
basic_istream<char,traits>& operator>>(basic_istream<char,traits>&, unsigned char*);

template<class traits>
basic_istream<char,traits>& operator>>(basic_istream<char,traits>&, signed char*);
}

class charT, class traits>
    basic_istream<charT,traits>& operator>>(basic_istream<charT,traits>&, charT*);

template<class traits>
    basic_istream<char,traits>& operator>>(basic_istream<charT,traits>&, unsigned char*);

template<class traits>
    basic_istream<char,traits>& operator>>(basic_istream<charT,traits>&, signed char*);

***************
*** 3580,3591 ****
public:
    explicit sentry(basic_istream<charT,traits>& is, bool noskipws = false);
    ~sentry();
    operator bool() { return ok_; } const;
    operator bool() { return ok_; } const;
};

***************
*** 3612,3635 ****
.CW is.tie()->flush()
.ix "[flush]"
to synchronize the output sequence with any associated external
C stream.
! CD2-27-034. 27.6.1.1.2
tied streams should be flushed just before calls to underflow
Except that this call can be suppressed if the put area of
! .CW is.tie()
is empty.
! Further an implementation is allowed to defer the call to flush until a
! call of
! .CW "is->rdbuf()->underflow"
! occurs.
! If no such call occurs before the
! .CW sentry
! object is destroyed, the call to
! .CW flush
! may be eliminated entirely*f
.Fs
! This will be possible only in functions that are part of the library.
! The semantics of the constructor used in user code is as specified.
.Fe
! If \&\f6noskipws\fP\& is zero and
! .CW "is.flags() & ios_base::skipws"
The call
.is.tie()->flush()
! does not necessarily occur if the function can determine that no
! synchronization is necessary.
! If \f6noskipws\fP is zero and
! .CW "is.flags() & ios_base::skipws"

***************
*** 4068,4077 ****
  basic_istream<charT,traits>& get(char_type* \f6s\fP, streamsize \f6n\fP)
  .La "Effects:"
  Calls
  .CW "get(s,n,widen('\en'))"
  .La "Returns:"
  Value returned by the call.
.
--- 4005,4013 ----
  basic_istream<charT,traits>& get(basic_streambuf<char_type,traits>& \f6sb\fP);
.
 La Effects:"
 Calls
 .CW "getline(s,n,widen('\en'))"
 .La "Returns:"
 Value returned by the call.
.
***************
*** 4258,4272 ****
  basic_istream<charT,traits>& read(char_type* \f6s\fP, streamsize \f6n\fP);
  .La Effects:"
 Calls
 .CW "getline(s,n,widen('\en'))"
 .La "Returns:"
 Value returned by the call.
.
**************
*** 4044,4052 ****
  basic_istream<charT,traits>& get(basic_streambuf<char_type,traits>& \f6sb\fP);
  .La "Effects:"
 Calls
 .CW "getline(s,n,widen('\en'))"
 .La "Returns:"
 Value returned by the call.
.
**************
*** 4258,4272 ****
  basic_istream<charT,traits>& read(char_type* \f6s\fP, streamsize \f6n\fP);
  .La Effects:"
 Calls
 .CW !good()!
 calls
! .CW setstate(failbit)
! which may throw an exception,
! and return.
! Otherwise extracts characters and stores them
into successive locations of an array whose first element is designated by
\&\f6s\fP\&.\*f
.Fs
--- 4193,4200 ----
.Pb
basic_istream<charT,traits>& read(char_type* \f6s\fP, streamsize \f6n\fP);
.Fe
La Effects:
! Extracts characters and stores them
into successive locations of an array whose first element is designated by
\&\f6s\fP\&.\*f
.Fs
***************
*** 4281,4288 ****
.LI
end-of-file occurs on the input sequence
(in which case the function calls
! ." CD2-27-008. 27.6.1.3. Add eofbit to values set.
! .CW setstate(failbit|eofbit) ,
which may throw
.CW ios_base::failure
(_lib.iostate.flags_).
--- 4209,4215 ----
.LI
end-of-file occurs on the input sequence
(in which case the function calls
! .CW setstate(failbit) ,
which may throw
.CW ios_base::failure
(_lib.iostate.flags_).
***************
*** 4565,4573 ****
基本_ostream<charT,traits>& operator<<(double \f6f\fP);
基本_ostream<charT,traits>& operator<<(long double \f6f\fP);
.Ce
- ." USA lib CD2-27-042 & Sweden _27621/ 27.6.2.1
 .Cb
基本_ostream<charT,traits>& operator<<(const void* \f6p\fP);
基本_ostream<charT,traits>& operator<<
(basic_streambuf<char_type,traits>* \f6sb\fP);
.Ce
--- 4492,4499 ----
基本_ostream<charT,traits>& operator<<(double \f6f\fP);
基本_ostream<charT,traits>& operator<<(long double \f6f\fP);
.Ce
 .Cb
基本_ostream<charT,traits>& operator<<(void* \f6p\fP);
基本_ostream<charT,traits>& operator<<
(basic_streambuf<char_type,traits>* \f6sb\fP);
.Ce
***************
*** 4706,4717 ****
public:
  explicit sentry(basic_ostream<charT,traits>& os);
  ~sentry();
! ." CD2-27-021. 27.6.2.3
!  operator bool() const { return ok_; }
!  private
! ." CD2-27-034 27.6.2.3 lib ostream::sentry
!  sentry(const sentry&); // \f4 not defined\fP
!  sentry& operator=(const sentry&); // \f4 not defined\fP
};
--- 4632,4638 -----
    public:
    explicit sentry(basic_ostream<charT,traits>& os);
    ~sentry();
    !     operator bool() { return ok_; }
    
    !.Pe

***************
*** 4870,4877 ****
operator<<(float \f6val\fP);
operator<<(double \f6val\fP);
operator<<(long double \f6val\fP);
! ." USA lib CD2-27-042 & Sweden _27621/
! operator<<(const void* \f6val\fP);
 .Pe
 ." X3J16/94-0064R1,WG21/N0451R1
 .La Effects:
--- 4791,4797 -----
operator<<(float \f6val\fP);
operator<<(double \f6val\fP);
operator<<(long double \f6val\fP);
! .Pe
 ." X3J16/94-0064R1,WG21/N0451R1
 .La Effects:
***************
*** 4909,4920 ****
 .CW ios_base .
 It provides formatting specifications such as field width, and
a locale from which to obtain other facets.
! ." USA lib CD2-27-037 27.6.2.5.2 lib.ostream.inserters.arithmetic
! If
! .CW failed
! is true then does
! .CW "setstate(badbit)" ,
! which may throw an exception, and returns.
 .La Returns:
 .CW *this .
 ." 
--- 4829,4838 -----
 .CW ios_base .
 It provides formatting specifications such as field width, and
a locale from which to obtain other facets.
! .eN
! This doesn't say what happens on failure. It needs to set one of the error
! bits. It isn't clear which one is right.
! .ne
 .La Returns:
 .CW *this .
 ." 
***************
*** 5025,5038 ****
After a
 .CW sentry
object is constructed insert characters.
! ." CD2-27-032, CD2-27-003 27.6.2.5.4 Clarify when widen is called for
! ." character inserters.
! In case \f6c\fP has type
 .CW char
! and the character type of the stream is not
! .CW char ,
! then the character to be inserted is
! .CW "out.widen(\f6c\fP)" ;
otherwise the character is \f6c\fP*f
.Fs
In case the insertion is into a
--- 4943,4952 -----
After a
.CW sentry
object is constructed insert characters.
! In case \f6c\fP's type is (signed, unsigned or plain)
.CW char
! the character to be inserted is
! .CW "widen(\f6c\fP)" ;
! otherwise the character is \f6c\fP*f
.CW .Fs
In case the insertion is into a
***************
*** 5077,5088 ****
Padding is determined as described in _lib.facet.num.put.virtuals_.
The
.CW traits::length(\f6s\fP)
! ." CD2-27-031, CD2-27-003. 27.6.2.5.4 widen characters in array inserters.
! characters starting at \f6s\fP are widened using
! .CW out widen
! (_lib.basic.ios.members_).
! The widened characters and any required padding are inserted into
! \f6out\fP.  Calls
.CW width(0) .
.La Returns:
.I out
--- 4991,4999 -----
Padding is determined as described in _lib.facet.num.put.virtuals_.
The
.CW traits::length(\f6s\fP)
! characters starting at \f6s\fP and any required padding is inserted
! into \f6out\fP.
! Calls
.CW width(0) .
.La Returns:
.I out
***************
*** 5255,5272 ****
behaves as if
.CW f(s)
were called,
! ." USA CD2-27-015. 27.6.3
! ." Manipulator are applicable to both ostream and istreams.
! and if
! .CW in
! is an (instance of)
! .CW basic_istream
! then the expression
! .CW in>>s
! behaves as if
! .CW f(s)
! were called.
! Where \&\f6f& can be defined as:\*f
.CW .Fs
The expression
.CW "cin >> resetiosflags(ios_base::skipws)"
--- 5166,5172 -----
behaves as if
.CW f(s)
were called,
! where \&\f6f& can be defined as:\*f
.CW .Fs
The expression
.CW "cin >> resetiosflags(ios_base::skipws)"
***************
*** 5296,5314 ****
return \f6str\fP;
Return value of insertion or extraction of manipulators.

The expression `out<<s` has type `ostream&` and value `out`.
The expression `in>>s` has type `istream&` and value `in`.

Where `f6f\fP\&` can be defined as:

```cpp
ios_base& f6f\fP(ios_base& \f6str\fP, ios_base::fmtflags \f6mask\fP)
```

behaves as if `f(s)` were called, where `f6f\fP\&` can be defined as:

```cpp
ios_base& f6f\fP(f6f\fP\& \f6str\fP, ios_base::fmtflags \f6mask\fP)
```

behaves as if `f(s)` were called.

Where `\$\f6f\fP\&` can be defined as:

```cpp
ios_base& \f6f\fP\&(ios_base& \f6str\fP, ios_base::fmtflags \f6mask\fP)
```

behaves as if `f(s)` were called,

Where `\$\f6f\fP\&` can be defined as:

```cpp
ios_base& \f6f\fP(f6f\fP\& \f6str\fP, ios_base::fmtflags \f6mask\fP)
```

return `\f6str\fP`;

```cpp
f4smanip\fP setbase(int \f6base\fP);
```
setwbase(int base);

***************
*** 5370,5386 ***
behaves as if
! where \&f\& can be defined as:
.Cb
ios_base& f(ios_base& str, int base)
{ // set basefield
--- 5235,5241 ----
behaves as if
! where \&f\& can be defined as:
.Cb
ios_base& f(ios_base& str, int base)
{ // set basefield
***************
*** 5391,5409 ***
return str;
}

! ." USA CD2-27-015. Manipulator are applicable to both ostream and
! ." istreams. and if
! .CW in
! is an (instance of)
! .CW basic_istream
! then the expression
! .CW in>>s
! behaves as if
! .CW f(s)
! were called.
! Where \&f\& can be defined as:
.Cb
ios_base& f(ios_base& str, int base)
{ // set basefield
--- 5246,5252 ----
return str;
}

! ."----
! .ix "[setfill]"
.Pb
\4smanip\FP setfill(char_type \6c\FP);
--- 5246,5252 ----
return \6str\FP;
}

! ."----
! .ix "[setfill]"
.Pb
\4smanip\FP setfill(char_type \6c\FP);
***************
*** 5429,5441 ***
return \6str\FP;

Page 13
The expression .CW out<<s has type .CW "ostream&" and value .CW out.

\verbatim
ios_base& \f6f\fP(ios_base& \f6str\fP, int \f6n\fP)
{
  // set precision\fP
  return \f6str\fP;
}
\endverbatim

The expression .CW in>>s has type .CW "istream&" and value .CW in.

\verbatim
4smanip\fP setw(int \f6n\fP);
\endverbatim
Manipulator

behave as if

were called,

Manipulator are applicable to both ostream and

streams. and if

where \&f\& can be defined as:

\begin{verbatim}
ios_base& f(ios_base& str, int n)
{ // set width
--- 5310,5316 ----
behave as if
.f(s)
were called,

where \&f\& can be defined as:

\begin{verbatim}
ios_base& f(ios_base& str, int n)
{ // set width

***************

behave as if

were called.

Where \&f\& can be defined as:

\begin{verbatim}
ios_base& f(ios_base& str, int n)
{ // set width

***************

return str;

\end{verbatim}

--- 5310,5316 ----
behave as if

were called,

where \&f\& can be defined as:

\begin{verbatim}
ios_base& f(ios_base& str, int n)
{ // set width

***************

return f6strfp;

\end{verbatim}

! ." USA CD2-27-028. Return value of insertion or extraction of manipulators.
! The expression
! .CW out<<s
! has type
! .CW "ostream"
! and value
! .CW out .
! The expression
! .CW in>>s
! has type
! .CW "istream"
! and value
! .CW in . ."-------------------------------------------------------------------------

--- 5310,5316 ----
behave as if

were called,

where \&f\& can be defined as:

\begin{verbatim}
ios_base& f(ios_base& str, int n)
{ // set width

***************

return f6strfp;

\end{verbatim}

! ." USA CD2-27-028. Return value of insertion or extraction of manipulators.
! The expression
! .CW out<<s
! has type
! .CW "ostream"
! and value
! .CW out .
! The expression
! .CW in>>s
! has type
! .CW "istream"
! and value
! .CW in . ."-------------------------------------------------------------------------
typedef charT char_type;

typedef typename traits::int_type int_type;

typedef typename traits::pos_type pos_type;

typedef typename traits::off_type off_type;

typedef traits traits_type;

_LIB_STRINGBUF_CONSTRUCTORS:

explicit basic_stringbuf(ios_base::openmode which)

class Allocator = allocator<charT> >

class basic_stringbuf : public basic_streambuf<charT,traits> {
public:

_TYPES:

typedef charT char_type;

typedef typename traits::int_type int_type;

typedef typename traits::pos_type pos_type;

typedef typename traits::off_type off_type;

_LIB_STRINGBUF_CONSTRUCTORS:

explicit basic_stringbuf(ios_base::openmode which)

--- 5381,5392 ----
class Allocator = allocator<charT> >

class basic_stringbuf : public basic_streambuf<charT,traits> {
public:

_TYPES:

typedef charT char_type;

typedef typename traits::int_type int_type;

typedef typename traits::pos_type pos_type;

typedef typename traits::off_type off_type;

_CLASS_LOG:

_LIB_STRINGBUF_CONSTRUCTORS:

explicit basic_stringbuf(ios_base::openmode which)

***************

*** 5994,6006 ****
class Allocator = allocator<charT> >

class basic_stringbuf : public basic_streambuf<charT,traits> {
public:

_TYPES:

typedef charT char_type;

typedef typename traits::int_type int_type;

typedef typename traits::pos_type pos_type;

typedef typename traits::off_type off_type;

_LIB_STRINGBUF_CONSTRUCTORS:

explicit basic_stringbuf(ios_base::openmode which)

***************

*** 5784,5794 ****
class Allocator = allocator<charT> >

class basic_stringbuf : public basic_streambuf<charT,traits> {
public:

_TYPES:

typedef charT char_type;

typedef typename traits::int_type int_type;

typedef typename traits::pos_type pos_type;

typedef typename traits::off_type off_type;

_LIB_STRINGBUF_CONSTRUCTORS:

explicit basic_stringbuf(ios_base::openmode which)

***************

*** 6109,6115 ****
explicit basic_ostreamstream(

const basic_string<charT,traits,Allocator>& f6str,
ios_base::openmode f6which) = default;

" CD2-27-010. 27.7.3.1. Eliminate basic_ostreamstream destructor.

CLASS_LOG:

_LIB_OSTREAMSTREAM_CONSTRUCTORS:

explicit basic_ostreamstream(

const basic_string<charT,traits,Allocator>& f6str,
ios_base::openmode f6which) = default;

virtual ~basic_ostreamstream();

_LIB_OSTREAMSTREAM_CONSTRUCTORS:

***************

*** 6422,6429 ****

Page 16
.. protected:
   \f6// _lib.filebuf.virtuals_ Overridden virtual functions:\fP
   ! \f6/ CD2-27-009. 27.5.2 return type of showmanyc
   ! virtual streamsize showmanyc();
   virtual int_type underflow();
   virtual int_type uflow();
   virtual int_type pbackfail(int_type \f6c\fP = traits::eof());
--- 6210,6216 ----
.. protected:
   \f6/ _lib.filebuf.virtuals_ Overridden virtual functions:\fP
   ! virtual int      showmanyc();
   virtual int_type underflow();
   virtual int_type uflow();
   virtual int_type pbackfail(int_type \f6c\fP = traits::eof());
***************
*** 6666,6674 ****
.. H4 "Overridden virtual functions" lib.filebuf.virtuals
  \f6/" .ix "[basic__filebuf] [showmanyc]"
- \f6/ USA CD2-27-009. 27.8.1.4. return type of showmanyc
  .Pb
! streamsize showmanyc();
  .Pe
  .La Effects:
  Behaves the same as
--- 6453,6460 ----
.. H4 "Overridden virtual functions" lib.filebuf.virtuals
  \f6/" .ix "[basic__filebuf] [showmanyc]"
  .Pb
! int showmanyc();
  .Pe
  .La Effects:
  Behaves the same as
***************
*** 6869,6877 ****
.. H4 "Overridden virtual functions" lib.filebuf.virtuals
  \f6/" .ix "[basic__filebuf] [seekoff]"
- \f6/ CD2-27-024. 27.8.1.4. seekpos's argument is ignored.
  pos_type seekoff(off_type \f6off\fP, ios_base::seekdir \f6way\fP,
  !                  ios_base::openmode
                      = ios_base::in | ios_base::out);
  .Pe
  .La Effects:
--- 6655,6662 ----
  \f6/" .ix "[basic__filebuf] [seekoff]"
  .Pb
  pos_type seekoff(off_type \f6off\fP, ios_base::seekdir \f6way\fP,
  !                  ios_base::openmode \f6which\fP
                      = ios_base::in | ios_base::out);
  .Pe
  .La Effects:
***************
*** 6879,6888 ****
.. CW "is_open() == false", the positioning operation fails.
  Otherwise, if
- \f6/ CD2-27-023. 27.8.1.4
  .. CW "a_codecvt.encoding()>0"
  ! repositions the sequence by \f6\f6off\f6*a_codecvt.encoding()\fP\f6& positions in
  ! \f6/\f6file\fP
  This is done by some combination of manipulating the put or get area and
repositioning of the \f6\f6file\fP\f6 ("as if" by computing \f6\f6distance\fP,
the number of characters to be moved in `\&f6\fP\&` and calling
--- 6664,6671 -----
  \.CW "is_open() == false",
the positioning operation fails.
Otherwise, if
  \.CW "a_codecvt.encoding()>0"
! repositions the sequence by `\&f6off*e\fP\&` positions in `\&f6\fP\&`.
This is done by some combination of manipulating the put or get area and
repositioning of the `\&f6\fP\&` ('`as if'') by computing `\&f6distance\fP`,
the number of characters to be moved in `\&f6\fP\&` and calling
*******************
*** 6933,6945 ****
stream position, if possible.
If the positioning operation fails, or
if the object cannot represent the resultant stream position,
! \" Editorial Japan
! returns an invalid stream position (_lib.fpos_).
 \" [basic__filebuf] [seekpos]
- \" CD2-27-024. 27.8.1.4
.Pb
! pos_type seekpos(pos_type `\f6sp\fP`, ios_base::openmode
    = ios_base::in | ios_base::out);
  \Pe
Alters the file position, if possible, to correspond to the position
--- 6616,6726 -----
stream position, if possible.
If the positioning operation fails, or
if the object cannot represent the resultant stream position,
! returns an invalid stream position (_lib.iostreams.pos.t_).
! \" [basic__filebuf] [seekpos]
.Pb
! pos_type seekpos(pos_type `\f6sp\fP`, ios_base::openmode `\f6which\fP`
    = ios_base::in | ios_base::out);
  \Pe
Alters the file position, if possible, to correspond to the position
*******************
*** 6963,6970 ****
on the same file the effects are undefined.
! \La "Returns:"
  \.I sp
! \" Editorial Japan
! on success. Otherwise returns an invalid stream
position(_lib.iostreams.definitions_ .)
  \" [basic__filebuf] [sync]
--- 6744,6750 -----
on the same file the effects are undefined.
! \La "Returns:"
  \.I sp
! [basic__filebuf] [sync]

***************
*** 6984,6991 ****
! void imbue(const locale& `\f6loc\fP`);
  \Pe
! \" Editorial Japan
! " Precondition:"
! If the file is not positioned at its beginning and the encoding of the current
locale as determined by
  \.CW "a_codecvt.encoding()"
--- 6764,6770 -----
.Pb
void imbue(const locale& loc);

**Precondition**
If the file is not positioned at its beginning and the encoding of the current locale as determined by
CW "a_codecvt.encoding()"

***************
*** 7003,7017 ****
Causes characters inserted or extracted after this call to be converted according to &loc& until another call of &imbue&.

-La "Effects:"
- Causes characters inserted or extracted after this call
- to be converted according to &f6loc& until another call of
- CW imbue.
- La "Note:"
- This may require reconversion of previously converted characters.
- This in turn may require the implementation to be able to reconstruct
- the original contents of the file.

--- 6782,6787 -----
***************
*** 7020,7033 ****

--- 6790,6800 -----
***************
*** 7156,7169 ****

--- 7159,7169 ----
```cpp
// _lib.ofstream.cons_ Constructors:
--- 6923,6933 ------
template <class charT, class traits = char_traits<charT> > 
class basic_ofstream : public basic_ostream<charT,traits> { 
  public:
    typedef charT                     char_type;
    typedef typename traits::int_type int_type;
    typedef typename traits::pos_type pos_type;
    typedef typename traits::off_type off_type;

  public:
    // CD2-27-011. Eliminate redundant typedef's 
    // Types (inherited from basic_istream):
    // typedef charT                     char_type;
    // typedef typename traits::int_type int_type;
    // typedef typename traits::pos_type pos_type;
    // typedef typename traits::off_type off_type;
    // typedef traits                    traits_type;

  public:
    // constructors/destructor 
--- 7056,7066 ------

```