

Document No.: J16/97-0061
 WG21/N1099
 Date: 17 July 1997
 Project: Programming Language C++
 Reply-To: Steve Rumsby
 steve@maths.warwick.ac.uk

Iostreams WP changes for London
 =====

```

*** io-new.txt Wed Jul 16 16:04:32 1997
--- copyof~1\lib-io~1 Tue Jul 15 17:24:32 1997
*****
*** 142,157 ****
    .DE
    .Cb
    namespace std {
-   .\" 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 ****
    .CW 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 initializ
ed as
+   if by init.

```

```

+ .nE
.ix "[<cstdio>]"
.\"===
.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;

class ios_base;
--- 597,602 ----
*****
*** 810,823 ****
.Ce
.Cb
.\f6// destructor \fP
! .\" UK 634. 27.4.2 make ios_base destructor virtual
! virtual ~ios_base();
.Ce
.Cb
.\f6// _lib.ios.base.callback_callbacks;\fP
enum event { erase_event, imbue_event, copyfmt_event };
! .\" 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 ----
.Ce
.Cb
.\f6// destructor \fP
! ~ios_base();
.Ce
.Cb
.\f6// _lib.ios.base.callback_callbacks;\fP
enum event { erase_event, imbue_event, copyfmt_event };
! typedef void (*event_callback)(event, ios_base&, int \f6index\fP);
void register_callback(event_call_back \f6fn\fP, int \f6index\fP);

```

```

.Ce
.Cb
*****
*** 895,902 ****
    public:
        explicit failure(const string& \f6msg\fp);
        virtual ~failure();
! .\" CD2-27-007. 27.4.2.1.1. Add throw() spec to what.
!     virtual const char* what() const throw();
    };
}
.Pe
--- 874,880 ----
    public:
        explicit failure(const string& \f6msg\fp);
        virtual ~failure();
!     virtual const char* what() const;
    };
}
.Pe
*****
*** 918,925 ****
    initializing the base class with
    .CW exception(\&\f6msg\&\fp) .
    .La Postcondition:
! .\"CD2-27-001, UK 635 27.4.2.1.1
! .CW "strcmp(what(), msg.c_str()) == 0"
    .\"
    .ix "[ios_base::failure] [what]"
    .Pb
--- 896,902 ----
    initializing the base class with
    .CW exception(\&\f6msg\&\fp) .
    .La Postcondition:
! .CW "what() == \f6msg\fp.str()"
    .\"
    .ix "[ios_base::failure] [what]"
    .Pb
*****
*** 1371,1377 ****
    .CW ++ .
    .\"
    .ix "[ios_base] [iword]"
- .\" CD2-27-017. 27.4.2.5
    .Pb
    long& iword(int \f6idx\fp);
    .Pe
--- 1348,1353 ----
*****
*** 1484,1493 ****
    ios_base();
    .Pe
    .La Effects:
! .\" UK 636. 27.4.2.7
! Each
    .CW ios_base
! member has an indeterminate value after construction.
    .\"
    .\" Santa Cruz motion 36. N0842R1.
    .Pb
--- 1460,1468 ----
    ios_base();
    .Pe
    .La Effects:
! The
    .CW ios_base
! members are have an indeterminate value after construction.

```

```

.\"
.\" Santa Cruz motion 36. N0842R1.
.Pb
*****
*** 1505,1510 ****
--- 1480,1488 ----
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
! .\" CD2-27-036. 27.4.3
! explicit fpos(stateT);
\f6// _lib.fpos.members_ Members\fp
stateT state() const;
void state(stateT);
--- 1491,1497 ----
template <class stateT> class fpos {
public:
\f6// _lib.fpos.cons_ Constructors\fp
! 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 ,
- .\" 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 .
- .\" 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      \&\f5P(O(p)) == p\&\fP
-
*****
*** 1610,1627 ****
-
p != q      \&\flconvertible to\fp\& bool   \&\f5!(p==q)\fp\&
-
! q = p + o  fpos      + \&\floffset\fp\&      \&\f5q-o == p\fp\&
p += o
-
! q = p - o  fpos      - \&\floffset\fp\&      \&\f5q+o == p\fp\&
p -= o
-
o = p - q   OFF_T     \&\fldistance\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\&
-
-
- .TE
- .ad
- .Te
--- 1581,1593 ----
-
p != q      \&\flconvertible to\fp\& bool   \&\f5!(p==q)\fp\&
-
! q = p + o  POS_T     + \&\floffset\fp\&      \&\f5q-o == p\fp\&
p += o
-
! q = p - o  POS_T     - \&\floffset\fp\&      \&\f5q+o == p\fp\&
p -= o
-
o = p - q   OFF_T     \&\fldistance\fp\&      \&\f5q+o == p\fp\&
-
- .TE
- .ad
- .Te
*****
*** 1727,1738 ****
protected:
    basic_ios();
    void init(basic_streambuf<charT,traits>* \f6sb\fp);
- .Ce
- .Cb
- .\" US & Germany. CD2-27-002. Disallow copying streams.
- private:
-     basic_ios(const basic_ios& ); // not defined
-     basic_ios& operator=(const basic_ios&); // not defined
-     };
- }
- .Ce
--- 1693,1698 ----
*****
*** 1876,1884 ****
- .La Postcondition:
- .CW "\&\f6fillch\fp == fill()"
- .La Returns:
! The previous value of
! .\" Editorial Japan
! .CW "fill()" .
- .ix "[basic_ios] [copyfmt]"
- .Pb
- basic_ios& copyfmt(const basic_ios& \f6rhs\fp);
--- 1836,1842 ----
- .La Postcondition:
- .CW "\&\f6fillch\fp == fill()"
- .La Returns:

```

```

! The previous value of fill().
.ix "[basic_ios] [copyfmt]"
.Pb
basic_ios& copyfmt(const basic_ios& \f6rhs\fp);
*****
*** 1960,1967 ****
iostate rdstate() const;
.Pe
.La Returns:
! .\" CD-27-025. 27.4.5.3
! The error state of the stream buffer.
.\"
.ix "[basic_ios] [clear]"
.Pb
--- 1918,1924 ----
iostate rdstate() const;
.Pe
.La Returns:
! The control state of the stream buffer.
.\"
.ix "[basic_ios] [clear]"
.Pb
*****
*** 2568,2575 ****
.LI
the
.CW getloc()
! .\" Editorial Japan
! member to a copy the global locale,
.CW locale() ,
at the time of construction.
.La Notes:
--- 2525,2531 ----
.LI
the
.CW getloc()
! member to to a copy the global locale,
.CW locale() ,
at the time of construction.
.La Notes:
*****
*** 2941,2949 ****
.H5 "Get area" lib.streambuf.virt.get
.\"
.ix "[basic_streambuf] [showmanyc]"
- .\" CD2-27-009. 27.5.2.4.3
.Pb
! streamsize showmanyc();\*f
.Pe
.Fs
The morphemes of
--- 2897,2904 ----
.H5 "Get area" lib.streambuf.virt.get
.\"
.ix "[basic_streambuf] [showmanyc]"
.Pb
! int showmanyc();\*f
.Pe
.Fs
The morphemes of
*****
*** 3484,3495 ****
template<class charT, class traits>
basic_istream<charT,traits>& operator>>(basic_istream<charT,traits>&,
charT*);
- .\" CD2-27-020. 27.6.1.1. Correct type of args.
template<class traits>

```

```

!     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*);
}
.Ce
--- 3439,3449 ----
    template<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*);
}
.Ce
*****
*** 3580,3591 ****
    public:
        explicit sentry(basic_istream<charT,traits>& is, bool noskipws = false);
        ~sentry();
! .\" CD2-27-021. 27.6.1.1.2
!     operator bool() { return ok_; } const;
! .\" CD2-27-034. 27.6.1.1.2 sentry's should not be copyable
!     private:
!     sentry(const sentry&); // \f4 not defined\fP
!     sentry& operator=(const sentry&); // \f4 not defined\fP
    };
}
.Pe
--- 3534,3540 ----
    public:
        explicit sentry(basic_istream<charT,traits>& is, bool noskipws = false);
        ~sentry();
!     operator bool() { return ok_; }
    };
}
.Pe
*****
*** 3612,3635 ****
    .CW is.tie()->flush()
    .ix "[flush]"
    to synchronize the output sequence with any associated external
! C stream.
! .\" Germany_276112/ 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"

```

```

--- 3561,3572 ----
  .CW is.tie()->flush()
  .ix "[flush]"
  to synchronize the output sequence with any associated external
! C stream.\*f
  .Fs
! The call
! .CW is.tie()->flush()
! does not necessarily occur if the function can determine that no
! synchronization is necessary.
  .Fe
  If \&\f6noskipws\fP\& is zero and
  .CW "is.flags() & ios_base::skipws"
*****
*** 4068,4077 ****
  .Pb
  basic_istream<charT,traits>& get(char_type* \f6s\fP, streamsize \f6n\fP)
  .Pe
- .\" USA lib CD2-27-014 27.6.1.3 lib.istream.unformatted
  .La "Effects:"
  Calls
! .CW "get(s,n,widen('\en'))"
  .La "Returns:"
  Value returned by the call.
  .\"
--- 4005,4013 ----
  .Pb
  basic_istream<charT,traits>& get(char_type* \f6s\fP, streamsize \f6n\fP)
  .Pe
  .La "Effects:"
  Calls
! .CW "getline(s,n,widen('\en'))"
  .La "Returns:"
  Value returned by the call.
  .\"
*****
*** 4108,4117 ****
  .Pb
  basic_istream<charT,traits>& get(basic_streambuf<char_type,traits>& \f6sb\fP);
  .Pe
- .\" USA lib CD2-27-014 27.6.1.3 lib.istream.unformatted
  .La "Effects:"
  Calls
! .CW "get(s,n,widen('\en'))"
  .La "Returns:"
  Value returned by the call.
  .\"
--- 4044,4052 ----
  .Pb
  basic_istream<charT,traits>& get(basic_streambuf<char_type,traits>& \f6sb\fP);
  .Pe
  .La "Effects:"
  Calls
! .CW "getline(s,n,widen('\en'))"
  .La "Returns:"
  Value returned by the call.
  .\"
*****
*** 4258,4272 ****
  .Pb
  basic_istream<charT,traits>& read(char_type* \f6s\fP, streamsize \f6n\fP);
  .Pe
- .\" CD2-27-033 27.6.1.3 lib.istream.unformatted
  .La Effects:
! If
! .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);
! .Pe
! .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 ****
! basic_ostream<charT,traits>& operator<<(double \f6f\fp);
! basic_ostream<charT,traits>& operator<<(long double \f6f\fp);
! .Ce
- .\" USA lib CD2-27-042 & Sweden _27621/ 27.6.2.1
! .Cb
! basic_ostream<charT,traits>& operator<<(const void* \f6p\fp);
! basic_ostream<charT,traits>& operator<<
! (basic_streambuf<char_type,traits>* \f6sb\fp);
! .Ce
--- 4492,4499 ----
! basic_ostream<charT,traits>& operator<<(double \f6f\fp);
! basic_ostream<charT,traits>& operator<<(long double \f6f\fp);
! .Ce
! .Cb
! basic_ostream<charT,traits>& operator<<(void* \f6p\fp);
! basic_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
! };
}

```

```

.Pe
--- 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);
!   operator<<(void* \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
    .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\fP\& can be defined as:\*f
    .Fs
    The expression
    .CW "cin >> resetiosflags(ios_base::skipws)"
--- 5166,5172 ----
    behaves as if
    .CW f(s)
    were called,
! where \&\f6f\fP\& can be defined as:\*f
    .Fs
    The expression
    .CW "cin >> resetiosflags(ios_base::skipws)"
*****
*** 5296,5314 ****
    return \f6str\fP;

```

```

    }
.Ce
- .\" 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 .
.\"----
.ix "[setiosflags]"
.Pb
--- 5196,5201 ----
*****
*** 5324,5340 ****
    behaves as if
    .CW f(s)
    were called,
! .\" 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 \&\f6f\fP& can be defined as:
    .ix "[ios] [fmtflags]"
    .Cb
        ios_base& \f6f\fP(ios_base& \f6str\fP, ios_base::fmtflags \f6mask\fP)
--- 5211,5217 ----
    behaves as if
    .CW f(s)
    were called,
! where \&\f6f\fP& can be defined as:
    .ix "[ios] [fmtflags]"
    .Cb
        ios_base& \f6f\fP(ios_base& \f6str\fP, ios_base::fmtflags \f6mask\fP)
*****
*** 5343,5361 ****
        return \f6str\fP;
    }
.Ce
! .\" 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 ..\"----
.ix "[setbase]"
.Pb
\f4smanip\fP setbase(int \f6base\fP);

```

```

--- 5220,5226 ----
        return \f6str\fP;
    }
    .Ce
! .\"-----
    .ix "[setbase]"
    .Pb
    \f4smanip\fP setbase(int \f6base\fP);
*****
*** 5370,5386 ****
    behaves as if
    .CW f(s)
    were called,
! .\" 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 \&\f6f\fP\& can be defined as:
    .Cb
        ios_base& \f6f\fP(ios_base& \f6str\fP, int \f6base\fP)
        { \f6// set basefield\fP
--- 5235,5241 ----
    behaves as if
    .CW f(s)
    were called,
! where \&\f6f\fP\& can be defined as:
    .Cb
        ios_base& \f6f\fP(ios_base& \f6str\fP, int \f6base\fP)
        { \f6// set basefield\fP
*****
*** 5391,5409 ****
        return \f6str\fP;
    }
    .Ce
! .\" 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 ..\"-----
    .ix "[setfill]"
    .Pb
    \f4smanip\fP setfill(char_type \f6c\fP);
--- 5246,5252 ----
        return \f6str\fP;
    }
    .Ce
! .\"-----
    .ix "[setfill]"
    .Pb
    \f4smanip\fP setfill(char_type \f6c\fP);
*****
*** 5429,5441 ****
        return \f6str\fP;

```

```

    }
.Ce
- .\" 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 .
. \"----
.ix "[setprecision]"
.Pb
--- 5272,5277 ----
*****
*** 5451,5467 ****
    behaves as if
    .CW f(s)
    were called,
! .\" 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 \&\f6f\fp\& can be defined as:
.Cb
    ios_base& \f6f\fp(ios_base& \f6str\fp, int \f6n\fp)
    { \f6// set precision\fp
--- 5287,5293 ----
    behaves as if
    .CW f(s)
    were called,
! where \&\f6f\fp\& can be defined as:
.Cb
    ios_base& \f6f\fp(ios_base& \f6str\fp, int \f6n\fp)
    { \f6// set precision\fp
*****
*** 5469,5487 ****
        return \f6str\fp;
    }
.Ce
! .\" 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 ..\"----
.ix "[setw]"
.Pb
\f4smanip\fp setw(int \f6n\fp);
--- 5295,5301 ----
    return \f6str\fp;
    }
.Ce
! .\"----
.ix "[setw]"

```

```

.Pb
\f4smanip\fp setw(int \f6n\fp);
*****
*** 5496,5512 ****
behaves as if
.CW f(s)
were called,
! ." 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 \&\f6f\fp\& can be defined as:
.Cb
    ios_base& \f6f\fp(ios_base& \f6str\fp, int \f6n\fp)
    { \f6// set width\fp
--- 5310,5316 ----
behaves as if
.CW f(s)
were called,
! where \&\f6f\fp\& can be defined as:
.Cb
    ios_base& \f6f\fp(ios_base& \f6str\fp, int \f6n\fp)
    { \f6// set width\fp
*****
*** 5514,5532 ****
        return \f6str\fp;
    }
.Ce
! ." 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 ..\"-----
-----
.H2 "String-based streams" lib.string.streams
.P
The header
--- 5318,5324 ----
    return \f6str\fp;
}
.Ce
! ."-----
.H2 "String-based streams" lib.string.streams
.P
The header
*****
*** 5589,5602 ****
        class Allocator = allocator<charT> >
        class basic_stringbuf : public basic_streambuf<charT,traits> {
public:
! ." CD2-27-011. Eliminate redundant typedef's
! \f6// Types (inherited from basic_streambuf\c
! (_lib.ios_):\fp

```

```

! // 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;.Ce
.Cb
  \f6// _lib.stringbuf.cons_ Constructors:\fP
    explicit basic_stringbuf(ios_base::openmode \f6which\fP
--- 5381,5392 ----
        class Allocator = allocator<charT> >
        class basic_stringbuf : public basic_streambuf<charT,traits> {
        public:
! \f6// Types:\fP
!   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;
! .Ce
.Cb
  \f6// _lib.stringbuf.cons_ Constructors:\fP
    explicit basic_stringbuf(ios_base::openmode \f6which\fP
*****
*** 5994,6006 ****
        class Allocator = allocator<charT> >
        class basic_istream : public basic_istream<charT,traits> {
        public:
! \f6// Types (inherited from basic_istream\c
! (_lib.input.streams_):\fP
! // 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;
! .Ce
.Cb
  \f6// _lib.istream.cons_ Constructors:\fP
--- 5784,5794 ----
        class Allocator = allocator<charT> >
        class basic_istream : public basic_istream<charT,traits> {
        public:
! \f6// Types:\fP
!   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;
! .Ce
.Cb
  \f6// _lib.istream.cons_ Constructors:\fP
*****
*** 6109,6115 ****
    explicit basic_ostringstream(
        const basic_string<charT,traits,Allocator>& \f6str\fP,
        ios_base::openmode \f6which\fP = ios_base::out);
! ." CD2-27-010. 27.7.3.1. Eliminate basic_ostringstream destructor.
! .Ce
.Cb
  \f6// _lib.ostringstream.members_ Members:\fP
--- 5897,5903 ----
    explicit basic_ostringstream(
        const basic_string<charT,traits,Allocator>& \f6str\fP,
        ios_base::openmode \f6which\fP = ios_base::out);
!   virtual ~basic_ostringstream();
! .Ce
.Cb
  \f6// _lib.ostringstream.members_ Members:\fP
*****
*** 6422,6429 ****

```



```

.Cb
protected:
  \f6// _lib.filebuf.virtuals_ Overridden virtual functions:\fP
! .\" 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 ----
.Cb
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
.\"
.ix "[basic_filebuf] [showmanyc]"
- .\" 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
.\"
.ix "[basic_filebuf] [showmanyc]"
.Pb
! int showmanyc();
.Pe
.La Effects:
Behaves the same as
*****
*** 6869,6877 ****
.\" 95-0074/N0674
.ix "[basic_filebuf] [seekoff]"
.Pb
- .\" 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 ----
.\" 95-0074/N0674
.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
- .\" CD2-27-023. 27.8.1.4
.CW "a_codecvt.encoding()>0"
! repositions the sequence by \&\f6off*a_codecvt.encoding()\fP& positions in
! \&\f6file\fP.
This is done by some combination of manipulating the put or get area and
repositioning of the \&\f6file\fP& (`as if' by computing \&\f6distance\fP,

```

```

the number of characters to be moved in \&\f6file\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 \&\f6file\fp.
This is done by some combination of manipulating the put or get area and
repositioning of the \&\f6file\fp\& (`as if'' by computing \&\f6distance\fp,
the number of characters to be moved in \&\f6file\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_).
."
.ix "[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
--- 6716,6726 ----
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_).
."
.ix "[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_ .)
."
.ix "[basic_filebuf] [sync]"
--- 6744,6750 ----
on the same file the effects are undefined.
.La "Returns:"
.I sp
! on success. Otherwise returns an invalid stream
position(_lib.iostreams.definitions_ .)
."
.ix "[basic_filebuf] [sync]"
*****
*** 6984,6991 ****
.Pb
void imbue(const locale& \f6loc\fp);
.Pe
! ." Editorial Japan
! .La "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& \f6loc\fp);
.Pe
! .La "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 \&\f6loc\fp\& until another call of \&\f5imbue\fp.
.nE
- .\ " CD2-27-038. 27.8.1.4. Add effects for imbue
- .La "Effects:"
- Causes characters inserted or extracted after this call
- to be converted according to \&\f6loc\fp 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.
.\ "----
.H4 "Template class \&\f7basic_ifstream\fp\&" lib.ifstream
.ix "[basic_ifstream]"
--- 6782,6787 ----
*****
*** 7020,7033 ****
  template <class charT, class traits = char_traits<charT> >
  class basic_ifstream : public basic_istream<charT,traits> {
  public:
! .\ " CD2-27-011. Eliminate redundant typedef's
! \f6// Types (inherited from basic_istream\c
! (_lib.istream_)):\fp
! // 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;
.Ce
.Cb
  \f6// _lib.ifstream.cons_ Constructors:\fp
--- 6790,6800 ----
  template <class charT, class traits = char_traits<charT> >
  class basic_ifstream : public basic_istream<charT,traits> {
  public:
! \f6// Types:\fp
! 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;
.Ce
.Cb
  \f6// _lib.ifstream.cons_ Constructors:\fp
*****
*** 7156,7169 ****
  template <class charT, class traits = char_traits<charT> >
  class basic_ofstream : public basic_ostream<charT,traits> {
  public:
! .\ " CD2-27-011. Eliminate redundant typedef's
! \f6// Types (inherited from basic_ostream\c
! (_lib.ostream_)):\fp
! // 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;
.Ce
.Cb

```

```

    \f6// _lib.ofstream.cons_ Constructors:\fP
--- 6923,6933 ----
    template <class charT, class traits = char_traits<charT> >
    class basic_ofstream : public basic_ostream<charT,traits> {
    public:
    !   \f6// Types:\fP
    !   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;
    .Ce
    .Cb
    \f6// _lib.ofstream.cons_ Constructors:\fP
*****
*** 7292,7305 ****
    .Ce
    .Cb
    public:
    !   ." CD2-27-011. Eliminate redundant typedef's
    !   \f6// Types (inherited from basic_istream\c
    !   (_lib.istream_)):\fP
    !   // 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;
    .Ce
    .Cb
    \f6// constructors/destructor \fP
--- 7056,7066 ----
    .Ce
    .Cb
    public:
    !   \f6// Types\fP
    !   typedef charT                                char_type;
    !   typedef typename traits::int_type            ins_type;
    !   typedef typename traits::pos_type            pos_type;
    !   typedef typename traits::off_type            off_type;
    .Ce
    .Cb
    \f6// constructors/destructor \fP

```