Fixing stream and streambuf iterators

Motivation

The classes istream_iterator, ostream_iterator, istreambuf_iterator, and ostreambuf_iterator are broken in many minor ways. Some of the problems are as follows.

- Descriptions refer to the nonexistent class ios_traits, rather than to char_traits.
- Several function templates are declared to have default template arguments.
- istreambuf_iterator takes its distance type as a template parameter, rather than picking it up from char_traits.
- Comparison functions take their arguments as non-const references rather than as const references.
- The header synopsis in clause 24.2 doesn't say that these classes are derived from the class iterator.
- The header synopsis refers to the function iterator_category, which was removed by the iterator_traits proposal.

In all cases, the intent is clear; the text simply doesn't say what it should. Many of these changes are arguably editorial.

Working paper changes

- strike the text
  template <class T, class charT, class traits = ios_traits<charT>, class Distance = ptrdiff_t>
  class istream_iterator
  from clause 24.5.1, replacing it with
  template <class T, class charT, class traits = char_traits<charT>, class Distance = ptrdiff_t>
  class istream_iterator
  class istream_iterator

- strike the text
  template <class T, class Distance>
  bool operator==(const istream_iterator<T, Distance>&, const istream_iterator<T, Distance>&);
  from clause 24.5.1, replacing it with
  template <class T, class charT, class traits, class Distance>
  bool operator==(const istream_iterator<T, charT, traits, Distance>&, const istream_iterator<T, charT, traits, Distance>&);
-- strike the text
template <class T, class Distance>
bool operator==(const istream_iterator<T, Distance>&, const istream_iterator<T, Distance>&);
from clause 24.2, replacing it with
template <class T, class charT, class traits, class Distance>
bool operator==(const istream_iterator<T, charT, traits,& Distance>&, const istream_iterator<T, charT, traits,& Distance>&);

-- strike the text
template<class T, class charT, class traits = ios_traits<charT> >
class ostream_iterator
from clause 24.5.2, replacing it with
template<class T, class charT, class traits = char_traits<charT> >
class ostream_iterator

-- strike the text
template<class T, class charT, class traits = ios_traits<charT> >
class ostream_iterator;
from clause 24.5.2, replacing it with
template<class T, class charT, class traits = char_traits<charT> >
class ostream_iterator : public iterator<output_iterator_tag, void, void>;

-- strike the text
template <class charT, class traits = ios_traits<charT>, class Distance = ptrdiff_t>
class istreambuf_iterator : public iterator<input_iterator_tag, charT, Distance>
from clause 24.5.3, replacing it with
template <class charT, class traits = char_traits<charT> >
class istreambuf_iterator : public iterator<input_iterator_tag, charT, typename traits::off_type>

-- strike the text
template <class charT, class traits = ios_traits<charT> > class istreambuf_iterator;
from clause 24.2, replacing it with
template <class charT, class traits = char_traits<charT> >;
class istreambuf_iterator : public iterator<input_iterator_tag, charT, typename traits::off_type>;

-- strike the text
template <class charT, class traits = ios_traits<charT> >
bool operator==(streambuf_iterator<charT, traits>& a, istreambuf_iterator<charT, traits>& b);
from clause 24.5.3, replacing it with
template <class charT, class traits>
bool operator==(const istreambuf_iterator<charT, traits>& a, const istreambuf_iterator<charT, traits>& b);

-- strike the text
template <class charT, class traits = ios_traits<charT> >
bool operator!=(streambuf_iterator<charT, traits>& a, istreambuf_iterator<charT, traits>& b);
from clause 24.5.3, replacing it with
template <class charT, class traits>
bool operator!==(const istreambuf_iterator<charT, traits>& a, const istreambuf_iterator<charT, traits>& b);
-- remove editorial boxes 78 and 79 from clause 24.2.

-- add the sentence
Constructs an end-of-stream iterator if s.rdbuf() is null.
after the second Effects paragraph in clause 24.5.3.2.

-- insert the text
Requires: s is not null
Before the Returns section for ostreambuf_iterator(streambuf_type *s) in clause 24.5.4.1.