Doc. No.:	WG21/N1016 X3J16/96-0198
Date:	November 11, 1996
Project:	C++ Standard Library
Reply to:	Pete Becker pbecker@oec.com

Clause 25 (Algorithms Library) Issues

Revision 2

Revision History:

Revision 1: September 26, 1995
Revision 2: November 11, 1996

Introduction:

This document summarizes the open issues for clause 25, including recommended actions, and indicates resolutions as they occur.

Issue Number:	006
Title:	Missing description of direction of copy
Section:	25.2.1
Requester:	Pete Becker
Owner:	Pete Becker
Status:	Resolved in Stockholm, not yet in WP

Description:

In [lib.copy.backward] the direction of the copy is specified as “starting from last-1 and proceeding to first.” [lib.copy] has no such specification, suggesting that the direction of copying is immaterial.

Discussion:

It should not be left to the reader to infer that [lib.copy] requires forward copying. It should be explicit.

Proposed Resolution:

Change the “Effects:” section of [lib.copy] to read:

Effects: Copies elements in the range [first, last) into the range [result, result + (last - first)) starting from first and proceeding to last. For each non-negative integer N < (last - first), performs *(result + n) = *(first + n).

Issue Number:	015
Title:	The templates min, max, defined in [lib.alg.min.max], should be moved to lib.utility.
Section:	25.3.7
Requester:	German Delegation
Owner:	Pete Becker
Status:	Open

Description:

The templates min and max are currently defined in the algorithms clause of the working paper. They seem to fit better in the general utilities subclause, since they are not algorithms that apply to containers but are of much broader utility.

Discussion:

These templates came into the working paper through STL, where they were classified as algorithms. The working paper has a broader category for functions and templates that are of more general use and do not depend on container semantics. These functions should be moved to that section of the paper.

Proposed Resolution:

Move the text describing the templates min and max from [lib.alg.min.max] to [lib.utility], leaving min_element and max_element in [lib.alg.min.max].

Issue Number:	016
Title:	The templates swap, defined in [lib.alg.swap], should be moved to lib.utility.
Section:	25.2.2
Requester:	German Delegation
Owner:	Pete Becker
Status:	Open

Description:

The template swap is currently defined in the algorithms clause of the working paper. It seems to fit better in the general utilities subclause, since it is not an algorithm that applies to containers but is of much broader utility.

Discussion:

This template came into the working paper through STL, where it was classified as an algorithm. The working paper has a broader category for functions and templates that are of more general use and do not depend on container semantics. This function should be moved to that section of the paper.

Proposed Resolution:

Move the text describing the templates swap from [lib.alg.swap] to [lib.utility].

Issue Number:	017
Title:	The template iter_swap, defined in [lib.alg.swap] should be moved to lib.iter.primitives.
Section:	25.2.2
Requester:	German Delegation
Owner:	Pete Becker
Status:	Open

Description:

The template iter_swap is currently defined in the algorithms clause of the working paper. It seems to fit better in the iterators subclause, since it is not an algorithm that applies to ranges of iterators but to two unrelated iterators.

Discussion:

The subclause [lib.iterator.primitives] contains several sections that deal with individual iterators and their properties in order to support implementation of algorithms. iter_swap seems to fit better in that subclause than in the algorithms clause.

Proposed Resolution:

Move the text describing the template iter_swap from [lib.alg.swap] to [lib.iterator.primitives].