Document No: N3108= 10-0098
Date: 2010-08-05
Project: Programming Language C++
Reply to: Nicolai Josuttis
nico@josuttis.de

Proposed Resolution for US 114: Small-string optimization not possible with current swap() specification

Rationale

Strings are containers (they fulfill sequence container requirements, which according to table 97 fulfill container requirements), but for containers the Standard states in 23.2.1 General container requirements §9:

> Every iterator referring to an element in one container before the swap shall refer to the same element in the other container after the swap.

Proposed Wording

In **21.4 Class template basic_string, §3**

In

> The class template basic_string conforms to the requirements for a Sequence Container (23.2.3), for a Reversible Container (23.2), and for an Allocator-aware container (96), except that basic_string does not constructor destroy its elements using allocator_traits<Alloc>::construct and allocator_traits<Alloc>::destroy..
> The iterators supported by basic_string are random access iterators (24.2.7).

insert

> The class template basic_string conforms to the requirements for a Sequence Container (23.2.3), for a Reversible Container (23.2), and for an Allocator-aware container (96), except that basic_string does not construct or destroy its elements using allocator_traits<Alloc>::construct and allocator_traits<Alloc>::destroy
> **and that swap() for basic_string invalidates iterators. {FOOTNOTE: swap() invalidates iterators to enable the small-string optimization.}**
> The iterators supported by basic_string are random access iterators (24.2.7).