

Multibyte Character Handling in I/O Library of C++

C++WG, ITSCJ, JAPAN

March 1992

1 Introduction

This article summarizes our activities for investigating the multibyte character handling in I/O library of C++.

— 2 Basic Concept and Design Principle

It is useful to introduce the basic concepts defined in MSE(Multibyte Support Extensions) of C(IS-9899) proposed by Japanese National Body as an addendum for the IS-9899. In that proposal, there are several basic definitions for multibyte and `wchar_t` in C language. In this basic design principle, they consider to invent several additional utilization functions for handling `wchar_t` data, e.g. `wprintf`. In C++ I/O library design for multibyte character handling, this basic definitions can be used. It is important to make the additional mechanisms for handling multibyte characters compatible with existing functionalities for single byte characters in C++ stream I/O.

3 Alternatives for I/O Library Design in C++

There are two alternatives for designing the I/O library, mainly concentrated in multibyte character oriented stream I/O. One is to introduce the `wstreambuf` and `wistream/wostream` as new classes in stream I/O hierarchical architecture. Another is to add only new member functions to already proposed classes like `streambuf` and `istream/ostream`. In either case, it is necessary to prepare a conversion mechanism between multibyte character streams and wide character streams.

3.1 Case 1: Extension of `wstreambuf` and `wistream/wostream`

The hierarchical architecture of stream I/O in this case is shown in Figure 1.

`Wistream/wostream` has wide character strings, and the data transfer can be supported among wide character strings and variables in user programs with given data types like `char`, `int` and so on. `Wstreambuf` can have multibyte character strings. Conversion mechanism is realized using additional class `wconvstreambuf`. This conversion mechanism supports multibyte characters to wide characters and vice versa. It can understand encoding