Clause 19 (Diagnostics Library) Issues List - Version 1

Revision History


Introduction

This document is a summary of the issues identified in Clause 19. For each issue the status, a short description, and pointers to relevant reflector messages and papers are given.

Active Issues

-----------------------------------------------

Work Group: Library Clause 19
Issue Number: 19-001
Title: Use and Treatment of Clause 19 Predefined Exceptions
Inconsistent
Sections: 19 Diagnostics Library [lib.diagnostics]
Status: active
Description: Jonathan Schilling in a private mail:

>During the Santa Cruz straw-vote discussion on adding underflow_error
>as a predefined exception, someone asked whether the WP should state
>in what situations this exception is thrown. Beman (or someone else,
>I’m not sure) said that this was not necessary, since for example
>nnowhere is it stated where overflow_error is thrown.
>
>Well, that’s not exactly correct, since bitset::to_ulong() [WP 23.2.1.2]
is documented as potentially throwing overflow_error.
>
>More generally, the use and treatment of the Clause 19 predefined
>exceptions doesn’t seem very consistent in the WP. Some libraries
>(string, locale, bitset) document that they may throw them in certain
>situations, while the other libraries have no "Throws:" specifications
>at all (other than the "default" one of [lib.res.on.exception.handling]).
>Some of the predefined exceptions get "used" by classes in the
>standard library (e.g. out_of_range is used by string and bitset) while
>others are not "used" at all (e.g. domain_error, which would seem to be
>a good candidate for use by the numerics library).
>

I understand that in the spirit of the original Clause 19 design (Keffer’s 94-0021/N0408 paper), the predefined exceptions don’t have to be used by the standard library in order to be of value -- they exist to provide a framework for programmers to define exception classes in their own applications. But surely the predefined exceptions would also provide value in allowing people to write narrow but portable exception handlers in code that makes use of the standard library.

(By comparison, in Ada predefined exceptions are treated very consistently – all standard exceptions are "used", and there is a complete list of the situations in which each standard exception will be raised.)

My question is, are there cases now in the standard library where designers are expecting that one of the predefined exceptions might be thrown, but this is not documented in the WP "Throws:" specifications? Is this the case with underflow_error, or domain_error, for instance? If not, I have no issue. But if so, then I think there would be a real benefit in adding these specifications to the WP. I am _not_ proposing that any redesign of libraries be done to throw exceptions where it wasn’t intended (e.g. STL).

Proposed Resolution:

Make sure that the standard library consistently documents all throw specifications which throw predefined exceptions.
(Needs a specific recommendation)

Requestor:      Jonathan Schilling, jls@sco.com
Owner:          Sandra Whitman
Emails:         None.
Papers:         None.