## Report from the Library working group

*Work done Mar — Jul 90*

This report presents a summary of the work the Library working group did between the first and second meetings. It is organized along the lines of the slides from Tuesday's presentation:

### *Background*

For the benefit of those who joined the discussion after March, we reviewed the discussion about the Library portion of the standard at that meeting. The relevant aspects were the decisions made about the overall direction of the Library effort, and the resulting direction given to the working group.

#### Decisions

*Describe interfaces*
*Solicit input for consideration*
*Formulate acceptance criteria*
*C++ bindings for other standards*

At the March meeting, we had general agreement that the Library portion of the Standard would describe interfaces and remain silent on implementation details (although the Rationale might have something to say about specific cases). To the extent that the Library uses class declarations to describe interfaces, these declarations would therefore not include `private` portions.

A key decision was that the Library working group was not in the business of designing new libraries. The key idea is that the Standard would be based on existing practice. We agreed to solicit those who developed existing libraries for C++ to see if they would suggest parts of their libraries for the Standard.

Perhaps naively, we expected to have many suggestions put forward. We agreed to develop some explicit set of criteria for evaluating the submissions. We felt that there might be future delays in accepting the Standard if someone objected that their library had not been given due consideration, or had been excluded on arbitrary grounds.

We agreed not to formulate C++ interfaces for every other relevant standard (the POSIX standard was mentioned in this context). We decided to consider establishing liaisons with such standards groups on a case-by-case basis, and help co-ordinate efforts — but keep such "C++ bindings" outside the scope of the Library portion of this Standard.

#### Direction

*Document compatibility with ANSI C Library*
*Document solicitation/evaluation process*
*Describe evaluation criteria*

_*Define contents of Standard and Rationale_
_*Develop list of working papers_

The Library group was to develop 3 deliverables for the July meeting, as described in X3J16/90-0026. The statement of Goals for the July meeting (X3J16/90-0036), added the last two items.

Steve Clamage volunteered to coordinate input for the first deliverable on compatibility with the ANSI C library. Wahhab Baldwin agreed to take the lead on formulating solicitation, evaluation, and acceptance procedures. Tim O'Konski volunteered to organize our thoughts on the criteria to use to evaluate the submissions.

### Assumptions

*Some form of Exception Handling and Parameterized Types will be available*
*C++ Library can be required to be reentrant*
*Preprocessing may not be available*

Support for exceptions and type parameterization will fundamentally affect library design. Since there is widespread agreement to include these features in C++ (although the details are still to be worked out), it seems reasonable to assume their availability. Of course, this poses a problem for the direction given to the Library group that the Standard be based on existing practice: few existing C++ libraries use exceptions or type parameterization, and none use the exact form that will eventually appear in the language!

Various aspects of the C library (`errno`, locale settings, other global state) allow various implementations to be non-reentrant. Some of the discussion regarding a C++ library was that requiring a reentrant library would be helpful for users.

Some of the environment discussions in March indicated that some C++ vendors are working on more sophisticated development systems. While the current C model of preprocessing and compilation is likely to persist, there is no reason to assume that the preprocessor will be available in all C++ implementations. In the Library portion of the Standard, relying upon (and documenting) the use of `#include` directives may not be appropriate.

Too strong; it will be available but it may not be optimal to use

### *Proposed Standard Contents*

Reports from the Editorial and Environment groups indicated that the Library group was working on more than one aspect of the Standard. Most of the issues regarding compatibility with existing C libraries will be addressed in Chapter 17 — Environment. Discussions of actual C++ library contents will appear in Chapter 18 — Libraries.

### Chapter 17 — Environment

*Access to C Library*
*Access to C++ Library*
*Library interaction*
    *input/output*
    *memory management*
    *environment*

Section 3.5 of X3J16/90-0031 addresses the issues involved in providing access to C libraries in a mixed C/C++ implementation. Although a similar mechanism may be provided for gaining access to the C++ library, some C++-only implementations may choose otherwise. Whatever similarities and differences in these mechanisms, Chapter 17 seems the most appropriate place to document the details.

This chapter is also more appropriate for discussing what interactions exist in a mixed C/C++ implementation. The particular issues involved mixing input/output operations, calls to the free store allocators, and access to the program's environment.

Other

*Inter-process communication*
*Network communication*
*Parallel tasks*
*Process table*
*Relational database access*

This list was taken from the suggestion that the standard library should provide "higher level" abstractions.

***Working papers***

Based on the discussion to date, and on the example of other working groups, the Library working group needs to focus discussions into specific working papers and proposals. These will provide specific items for consideration by the full X3J16 committee. The documents will generally focus on one of two kinds of topics: Issues or Library contents.

**Issues**

*Interaction of signals and exceptions (if any)*
*C++ variable argument list manipulation*
*Memory management interaction*
- *new / malloc*
- *delete / free*
- *realloc*

*Input / output interaction*

Issues will identify topics that affect many portions of the Library group's contribution to the Standard, although they are not proposals to be included directly in the Standard. Each document will indicate a description of the issue, present the known alternatives to solving the problem, and recommend one solution. Once these kinds of issues are resolved, they will guide decisions on several proposals for contents of the Standard itself.

**Library contents**

*Types*
*System*
*Data Structures*
*Other standards efforts (POSIX)*
*Solicitation and evaluation process*

Of all the proposed Library contents discussed earlier, a few aspects will likely evolve into specific proposals for inclusion in the Standard. Those not appropriate for this Standard will have to be directed to the appropriate standards group. It also seems prudent to explicitly record the details of whatever solicitation and evaluation process the Library working group adopts.