N1135 Issues
Approach

• Discuss against last committee draft
• Many changes in editorial committee
• Discuss significant issues
• Discuss significant new material
• If desired, walkthrough changed sections (change bars)
Title of TR

• Security is unacceptable
• Safer is unacceptable
• “Extension to the C Library, Part 1”
• Changes macro names
  – __STDC_LIB_EXT1__
  – __STDC_WANT_LIB_EXT1__
“Constraints”

- Was diagnosed undefined behavior
- Committee requested constraints
  - But, that is only translation-time
  - Objections on the reflector
- I suggested “usage requirement”
- Editorial board picked “runtime-constraint”
  - Plum’s paper N1134
Runtime-constraint

- Defined in 3.1 and 4.0
- Model in 6.1.4
- typedef for handler in 6.6
- `set_constraint_handler_s` is in 6.6.1
abort_handler_s, etc.

• Suggested by editorial committee
  – abort_handler_s (Subclause 6.6.1.2)
  – ignore_handler_s (Subclause 6.6.1.3)
  – strict_handler_s (Subclause 6.6.1.4)

• Is strict_handler_s right?
Overlapping Operands

• Three cases:
  – Simple: memcpy, wmemcpy
  – Intermediate: strcpy, strcat, wcscpy, wcscat
  – Too hard: scanf, printf

• I argued that overlapping needs a better definition, if required to detect it
Overlapping Operands 2

- Much discussion in editorial committee
  - printf, scanf too hard (but vulnerability)
  - memcpy definitely
  - Intermediate: try

- Definition using pointer comparisons
  - relationals are not defined for different objects
  - Just state in English
Overlapping Operands 3

• “Copying shall not take place between objects that overlap.”

• Add to: strcpy_s, strncpy_s, strcat_s, strncat_s, wcsncpy_s, wcsncpy_s, wcscat_s, wcsncat_s?
Open mode “u”

• For fopen_s, freopen_s
• Means use system-default protections when creating a file
• Two cases:
  – “w” creating a file
  – “a” append creating a file
• Two letters needed, or is “u” a flag
The set_constraint_handler_s function sets the runtime-constraint handler to be handler. The runtime-constraint handler is the function to be called when a library function detects a runtime-constraint violation.
gets_s

- Provided for when fgets is not as compatible when gets as needed
- In registration draft
- Rewritten due to “constraints” edit
- Not quite right at editorial meeting
- Should be OK now
printf_s family functions

• Added to forbid %n (security vulnerability)
sprintf_s return value

- Editorial Committee changed return value

```c
count += sprintf_s(dest, sizeof dest, fmt1, arg1, arg2);
count += sprintf_s(dest+count, sizeof dest-count, fmt2, arg3, arg4);
```
mbstowcs_s, etc

- mbstowcs_s, wcstombs_s, mbsrtowcs_s, wcsrtombs_s,
- Should always null terminate results?
- Should *retval count the null terminator?
- Runtime-constraint: if dst is a null pointer, dstmax shall be zero.
strtok_s, wcstok_s

• New parameter to make sure function does not store outside of string tokenized
• wcstok_s added by Editorial committee
bsearch_s

• When can key be null?

• “If nmemb is not equal to zero, then none of key, base, or compar shall be a null pointer.”
Known Defects

- Make sure strnlen_s and wcsnlen_s are not called strnlen or wcsnlen
- Page 4, references to Clause 5 should be to Clause 6
- Title page
- (I hope) Delete footnote 70