

C++IssueStatus

The table below tracks the status of the C++ Committee's issues in the implementation of stdcxx.

		Status of C++ Issues		
	IS S U E	DESCRIPTION	STATUS	JIRA / REVISION
	1	C library linkage editing oversight	<i>Editorial</i>	N/A
	3	Atexit registration during atexit() call is not described	<i>Editorial</i>	N/A
	5	string::compare specification questionable	DONE	N/A
{*}	17	Bad bool parsing	CHECK	
{*}	18	get(...bool&) omitted		
{*}	19	noconv definition too vague		
{*}	20	thousands_sep returns wrong type		
{*}	21	codecvt_byname<> instantiations		
{*}	22	Member open vs. flags		
{*}	24	do_convert doesn't exist		
{*}	25	String operator<< uses width() value wrong		
{*}	26	Bad sentry example		
{*}	27	string::erase(range) yields wrong iterator		
{*}	28	ctype<char> is ambiguous		
{*}	29	Ios_base::init doesn't exist		
{*}	30	Wrong header for LC_*		
{*}	31	Immutable locale values		
{*}	32	pbackfail description inconsistent		
{*}	33	codecvt<> mentions from_type		
{*}	34	true/falsename() not in ctype<>		
{*}	35	No manipulator unitbuf in synopsis		
{*}	36	lword & pword storage lifetime omitted		
{*}	37	Leftover global reference		
{*}	38	Facet definition incomplete		
{*}	39	istreambuf_iterator<>::operator++(int) definition garbled		
{*}	40	Meaningless normative paragraph in examples		
{*}	41	Ios_base needs clear(), exceptions()		
{*}	42	string ctors specify wrong default allocator		
{*}	44	Iostreams use operator== on int_type values		
{*}	46	Minor Annex D errors		
{*}	47	imbue() and getloc() Returns clauses swapped		
{*}	48	Use of non-existent exception constructor		
{*}	49	Underspecification of ios_base::sync_with_stdio		
{*}	50	Copy constructor and assignment operator of ios_base		
{*}	51	Requirement to not invalidate iterators missing		
{*}	52	Small I/O problems		
{*}	53	basic_ios destructor unspecified		
{*}	54	basic_streambuf's destructor		
{*}	55	Invalid stream position is undefined		
{*}	56	showmanyc's return type		
{*}	57	Mistake in char_traits		
{*}	59	Ambiguity in specification of gbump		
{*}	60	What is a formatted input function?		
{*}	61	Ambiguity in iostreams exception policy		
{*}	62	ync's return value		
{*}	63	Exception-handling policy for unformatted output		
{*}	64	Exception handling in basic_istream::operator>>(basic_streambuf*)		
{*}	66	strstreambuf::setbuf		
{*}	68	Extractors for char* should store null at end		
{*}	69	Must elements of a vector be contiguous?		
{*}	70	uncaught_exception() missing throw() specification		
{*}	71	do_get_monthname synopsis missing argument		
{*}	74	Garbled text for codecvt::do_max_length		

{*}	75	Contradiction in <code>codecvt::length's</code> argument types		
{*}	76	Can a <code>codecvt</code> facet always convert one internal character at a time?		
!	78	Typo: <code>event_call_back</code>	Editorial	
{*}	79	Inconsistent declaration of <code>polar()</code>		
{*}	80	Global Operators of <code>complex</code> declared twice		
{*}	83	<code>string::npos</code> vs. <code>string::max_size()</code>		
{*}	86	<code>string</code> constructors don't describe exceptions		
{*}	90	Incorrect description of <code>operator>></code> for <code>{(string)}</code> s		
{*}	91	Description of <code>operator>></code> and <code>getline()</code> for <code>string<></code> might cause endless loop		
{*}	92	Incomplete Algorithm Requirements		
{*}	98	Input iterator requirements are badly written		
{*}	103	<code>set::iterator</code> is required to be modifiable, but this allows modification of keys		
{*}	106	Numeric library private members are implementation defined		
{*}	108	Lifetime of <code>exception::what()</code> return unspecified		
{*}	109	Missing binders for non-const sequence elements		
{*}	110	<code>istreambuf_iterator::equal</code> not const		
{*}	112	Minor typo in <code>ostreambuf_iterator</code> constructor		
{*}	114	Placement forms example in error twice		
{*}	115	Typo in <code>stringstream</code> constructors		
{*}	117	<code>basic_ostream</code> uses nonexistent <code>num_put</code> member functions		
{*}	118	<code>basic_istream</code> uses nonexistent <code>num_get</code> member functions		
{*}	119	Should virtual functions be allowed to strengthen the exception specification?		
{*}	120	Can an implementor add specializations?		
{*}	122	<code>streambuf/wstreambuf</code> description should not say they are specializations		
{*}	123	Should <code>valarray</code> helper arrays fill functions be const?		
{*}	124	<code>ctype_byname<charT>::do_scan_is</code> & <code>do_scan_not</code> return type should be <code>const charT*</code>		
{*}	125	<code>valarray<T>::operator!()</code> return type is inconsistent		
{*}	126	typos in <i>Effects</i> clause of <code>ctype::do_narrow()</code>		
{*}	127	<code>auto_ptr<></code> conversion issues		
{*}	129	Need error indication from <code>seekp()</code> and <code>seekg()</code>		
{*}	130	Return type of <code>container::erase(iterator)</code> differs for associative containers		
!	132	<code>list::resize</code> description uses random access iterators	Editorial	
{*}	133	<code>map</code> missing <code>get_allocator()</code>		
{*}	134	<code>vector</code> constructors over specified		
✖	135	Does <code>unique_copy()</code> require <code>CopyConstructible</code> and <code>Assignable</code> ?	TO DO	STDCXX-241
{*}	136	<code>seekp</code> , <code>seekg</code> setting wrong streams?		
{*}	137	Do <code>use_facet</code> and <code>has_facet</code> look in the global locale?		
{*}	139	Optional sequence operation table description unclear		
{*}	141	<code>basic_string::find_last_of</code> , <code>find_last_not_of</code> say pos instead of xpos		
{*}	142	<code>lexicographical_compare</code> complexity wrong		
{*}	144	<code>deque</code> constructor complexity wrong		
{*}	146	<code>complex<T></code> <code>Insertor</code> and <code>Extractor</code> need sentries		
{*}	147	Library Intro refers to global functions that aren't global		
{*}	148	Functions in the example facet <code>BoolNames</code> should be const		
{*}	150	<code>find_first_of</code> says integer instead of iterator		
{*}	151	Can't currently <code>clear()</code> empty container		
{*}	152	Typo in <code>scan_is()</code> semantics		
{*}	153	Typo in <code>narrow()</code> semantics		
{*}	154	Missing <code>double</code> specifier for <code>do_get()</code>		
{*}	155	Typo in naming the class defining the class <code>Init</code>		
{*}	156	Typo in <code>imbue()</code> description		
{*}	158	Underspecified semantics for <code>setbuf()</code>		

[*]	159	Strange use of <code>underflow()</code>		
[*]	160	Typo: Use of non-existing function <code>exception()</code>		
[*]	161	Typo: <code>istream_iterator</code> vs. <code>istreambuf_iterator</code>		
[*]	164	<code>do_put()</code> has apparently unused <code>fill</code> argument		
[*]	165	<code>xspn()</code> , <code>pubsync()</code> never called by <code>basic_ostream</code> members?		
[*]	167	Improper use of <code>traits_type::length()</code>		
[*]	168	Typo: formatted vs. unformatted		
[*]	169	Bad efficiency of <code>overflow()</code> mandated		
[*]	170	Inconsistent definition of <code>traits_type</code>		
[*]	171	Strange <code>seekpos()</code> semantics due to joint position		
[*]	172	Inconsistent types for <code>basic_istream::ignore()</code>		
[*]	173	Inconsistent types for <code>basic_filebuf::setbuf()</code>		
[*]	174	Typo: <code>OFF_T</code> vs. <code>POS_T</code>		
[*]	175	Ambiguity for <code>basic_streambuf::pubseekpos()</code> and a few other functions.		
[*]	176	<code>exceptions()</code> in <code>ios_base...</code> ?		
[*]	179	Comparison of <code>{{const_iterator}}</code> s to <code>{{iterator}}</code> s doesn't work		
[*]	181	<code>make_pair()</code> unintended behavior		
[*]	182	Ambiguous references to <code>size_t</code>		
[*]	183	I/O stream manipulators don't work for wide character streams		
[*]	184	<code>numeric_limits<bool></code> wording problems		
[*]	185	Questionable use of term <code>inline</code>		
[*]	186	<code>bitset::set()</code> second parameter should be <code>bool</code>		
[*]	187	<code>iter_swap</code> underspecified		
[*]	189	<code>setprecision()</code> not specified correctly		
[*]	193	Heap operations description incorrect		
[*]	195	Should <code>basic_istream::sentry</code> 's constructor ever set <code>eofbit</code> ?		
[*]	198	Validity of pointers and references unspecified after iterator destruction		
[*]	199	What does <code>allocate(0)</code> return?		
[*]	200	Forward iterator requirements don't allow constant iterators		
[*]	201	<code>numeric_limits</code> terminology wrong		
[*]	202	<code>unique()</code> effects unclear when predicate not an equivalence relation		
[*]	206	<code>operator new(size_t, nothrow)</code> may become unlinked to ordinary <code>operator new</code> if ordinary version replaced		
[*]	208	Unnecessary restriction on <i>past-the-end</i> iterators		
[*]	209	<code>basic_string</code> declarations inconsistent		
[*]	210	<code>distance</code> first and last confused		
[*]	211	<code>operator>>(istream&, string&)</code> doesn't set <code>failbit</code>		
[*]	212	Empty range behavior unclear for several algorithms		
[*]	214	<code>set::find()</code> missing <code>const</code> overload		
	217	Facets example (Classifying Japanese characters) contains errors	Editorial	
[*]	220	<code>~ios_base()</code> usage valid?		
[*]	221	<code>num_get<>::do_get</code> stage 2 processing broken		
[*]	222	Are <i>Throw</i> clauses necessary if a throw is already implied by the <i>Effects</i> clause?		
[*]	223	reverse algorithm should use <code>iter_swap</code> rather than <code>swap</code>		
[*]	224	<code>clear()</code> complexity for associative containers refers to undefined <code>N</code>		
[*]	225	<code>std::</code> algorithms use of other unqualified algorithms		
[*]	226	User supplied specializations or overloads of namespace <code>std</code> function templates		
[*]	227	<code>std::swap()</code> should require <code>CopyConstructible</code> or <code>DefaultConstructible</code> arguments		
[*]	228	Incorrect specification of <code>..._byname</code> facets		
[*]	229	Unqualified references of other library entities		
[*]	230	Assignable specified without also specifying <code>CopyConstructible</code>		
	231	Precision in <code>istream</code> ?	DONE	STDCXX-763
[*]	232	depends poorly defined in 17.4.3.1		
[*]	233	Insertion hints in associative containers		

{*}	234	Typos in allocator definition			
{*}	235	No specification of default ctor for reverse_iterator			
{*}	237	Undefined expression in complexity specification			
{*}	238	Contradictory results of stringbuf initialization.			
{*}	239	Complexity of unique() and/or unique_copy incorrect			
{*}	240	Complexity of adjacent_find() is meaningless			
{*}	241	Does unique_copy() require CopyConstructible and Assignable?			
{*}	242	Side effects of function objects			
{*}	243	get and getline when sentry reports failure			
{*}	247	vector, deque::insert complexity			
{*}	248	time_get fails to set eofbit			
{*}	250	splicing invalidates iterators			
{*}	251	basic_stringbuf missing allocator_type			
{*}	252	missing casts/C-style casts used in iostreams			
{*}	253	valarray helper functions are almost entirely useless			
{*}	254	Exception types in clause 19 are constructed from std::string			
{*}	256	typo in 27.4.4.2, p17: copy_event does not exist			
{*}	258	Missing allocator requirement			
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="24f59a16-e2f1-44c4-ac86-aa112f5c06a8"><ac:plain-text-body><![CDATA[	{*}	[259	http://tinyurl.com/ytwhttr#259]	basic_string::operator[] and const correctness	]]></ac:plain-text-body></ac:structured-macro>
{*}	260	Inconsistent return type of istream_iterator::operator++(int)			
{*}	261	Missing description of istream_iterator::operator!=			
{*}	262	Bitmask operator~ specified incorrectly			
{*}	263	Severe restriction on basic_string reference counting			
{*}	264	Associative container insert(i, j) complexity requirements are not feasible.			
{*}	265	std::pair::pair() effects overly restrictive			
{*}	266	bad_exception::~bad_exception() missing Effects clause			
{*}	268	Typo in locale synopsis			
{*}	270	binary_search requirements overly strict			
{*}	271	basic_istream missing {{typedef}}s			
	272	Missing parentheses around subexpression	Editorial		
	273	Missing ios_base qualification on members of a dependent class	Editorial		
{*}	274	a missing/impossible allocator requirement			
{*}	275	Wrong type in num_get::get() overloads			
{*}	276	Assignable requirement for container value_type overly strict			
{*}	278	What does iterator validity mean?			
{*}	280	Comparison of reverse_iterator to const reverse_iterator			
{*}	281	std::min() and max() requirements overly restrictive			
{*}	282	What types does numpunct::grouping refer to?			
{*}	283	std::replace() requirement incorrect/insufficient			
{*}	284	unportable example in 20.3.7, p6			
{*}	285	minor editorial errors in fstream ctors			
{*}	286	<cstdlib> requirements missing size_t typedef			
{*}	288	<cerrno> requirements missing macro EILSEQ			
{*}	291	Underspecification of set algorithms			
{*}	292	effects of a.copyfmt(a)			
{*}	294	User defined macros and standard headers			
{*}	295	Is abs defined in <cmath>?			
{*}	297	const_mem_fun_t<>::argument_type should be const T*			
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="dce5664e-88c5-4690-b99b-4875e8f59f2e"><ac:plain-text-body><![CDATA[	{*}	[298	http://tinyurl.com/ytwhttr#298]	::operator delete[] requirement incorrect/insufficient	]]></ac:plain-text-body></ac:structured-macro>
{*}	300	list::merge() specification incomplete			
{*}	301	basic_string template ctor effects clause omits allocator argument			
{*}	303	bitset input operator underspecified			

{*}	305	Default behavior of <code>codecvt<wchar_t, char, mbstate_t>::length()</code>			
{*}	306	<code>offsetof</code> macro and non-POD types			
{*}	307	Lack of reference typedefs in container adaptors			
{*}	308	Table 82 mentions unrelated headers			
{*}	310	Is <code>errno</code> a macro?			
{*}	311	Incorrect wording in <code>basic_ostream</code> class synopsis			
{*}	312	Table 27 is missing headers			
{*}	315	Bad "range" in <code>List::unique</code> complexity			
{*}	316	Vague text in Table 69			
{*}	317	Instantiation vs. specialization of facets			
{*}	318	Misleading comment in definition of <code>num_punct_byname</code>			
{*}	319	Storage allocation wording confuses "Required behavior", "Requires"			
{*}	320	<code>List::assign</code> overspecified			
{*}	321	Typo in <code>num_get</code>			
{*}	322	<code>iterator</code> and <code>const_iterator</code> should have the same value type			
{*}	324	Do output iterators have value types?			
{*}	325	Misleading text in <code>money_punct<>::do_grouping</code>			
{*}	327	Typo in <code>time_get</code> facet in table 52			
{*}	328	Bad <code>sprintf</code> format modifier in <code>money_put<>::do_put()</code>			
{*}	329	vector capacity, reserve and reallocation			
{*}	331	bad declaration of destructor for <code>ios_base::failure</code>			
{*}	333	does <code>endl</code> imply synchronization with the device?			
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="fcac6127-4fc8-4f0a-b203-27d313e52980"><ac:plain-text-body><![CDATA[		{*} [334	http://tinyurl.com/ytwhht#334	<code>map::operator[]</code> specification forces inefficient implementation	>></ac:plain-text-body></ac:structured-macro>
{*}	335	minor issue with <code>char_traits</code> , table 37			
{*}	336	Clause 17 lack of references to deprecated headers			
{*}	337	<code>replace_copy_if</code> 's template parameter should be <code>InputIterator</code>			
{*}	338	is whitespace allowed between '.' and a digit?			
{*}	339	definition of <code>bitmask</code> type restricted to clause 27			
{*}	340	interpretation of <code>has_facet<Facet>(loc)</code>			
{*}	341	Vector reallocation and swap			
{*}	345	type <code>tm</code> in <code><cwchar></code>			
{*}	346	Some iterator member functions should be <code>const</code>			
{*}	347	<code>locale::category</code> and <code>bitmask</code> requirements			
{*}	349	Minor typographical error in <code>ostream_iterator</code>			
{*}	352	missing <code>fpos</code> requirements			
{*}	354	Associative container <code>lower/upper_bound</code> requirements			
{*}	355	Operational semantics for <code>a.back()</code>			
{*}	358	interpreting <code>thousands_sep</code> after a <code>decimal_point</code>			
{*}	359	<code>num_put<>::do_put(..., bool)</code> undocumented			
{*}	360	locale mandates inefficient implementation			
{*}	362	<code>bind1st/bind2nd</code> type safety			
{*}	363	Missing exception specification in 27.4.2.1.1			
{*}	364	Inconsistent wording in 27.5.2.4.2			
{*}	365	Lack of <code>const</code> -qualification in clause 27			
{*}	369	io stream objects and static ctors			
{*}	370	Minor error in <code>basic_istream::get</code>			
{*}	371	Stability of <code>multiset</code> and <code>multimap</code> member functions			
{*}	373	Are <code>basic_istream</code> and <code>basic_ostream</code> to use (<code>exceptions() & badbit</code>) <code>!= 0</code> ?			
{*}	375	<code>basic_ios</code> should be <code>ios_base</code> in 27.7.1.3			
{*}	376	<code>basic_streambuf</code> semantics			
{*}	379	nonsensical <code>ctype::do_widen()</code> requirement			
{*}	380	typos in <code>codecvt</code> tables 53 and 54			
{*}	381	detection of invalid <code>mbstate_t</code> in <code>codecvt</code>			
{*}	383	Bidirectional iterator assertion typo			
{*}	384	<code>equal_range</code> has unimplementable runtime complexity			

<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="3f4b8c0c-1e2b-4a65-a570-6535c31bf4eb"><ac:plain-text-body><![CDATA[	{*}	[386	http://tinyurl.com/ytwhtt#386]	Reverse iterator's operator [] has impossible return type		]]></ac:plain-text-body></ac:structured-macro>
{*}	387	std::complex over-encapsulated				
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="bbf2db71-4f37-4d6e-88ef-fbd668de8fa3"><ac:plain-text-body><![CDATA[	{*}	[389	http://tinyurl.com/ytwhtt#389]	Const overload of valarray::operator[] returns by value		]]></ac:plain-text-body></ac:structured-macro>
{*}	391	non-member functions specified as const				
{*}	395	inconsistencies in the definitions of rand() and random_shuffle()				
{*}	400	redundant type cast in lib.allocator.members				
{*}	401	incorrect type casts in table 32 in lib.allocator.requirements				
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="7628610c-f986-4de1-8c11-7a41135f2134"><ac:plain-text-body><![CDATA[	{*}	[402	http://tinyurl.com/ytwhtt#402]	wrong new expression in [some_]allocator::construct		]]></ac:plain-text-body></ac:structured-macro>
{*}	403	basic_string::swap should not throw exceptions				
{*}	404	May a replacement allocation function be declared inline?				
{*}	405	qsort and POD				
{*}	406	vector::insert(s) exception safety				
{*}	407	Can singular iterators be destroyed?				
{*}	409	Closing an fstream should clear error state				
{*}	410	Missing semantics for stack and queue comparison operators				
{*}	411	Wrong names of set member functions				
{*}	412	Typo in 27.4.4.3				
{*}	413	Proposed resolution to LDR#64 still wrong				
{*}	414	Which iterators are invalidated by v.erase()?				
{*}	415	behavior of std::ws				
{*}	416	definitions of XXX_MIN and XXX_MAX macros in climits				
{*}	420	is std::FILE a complete type?				
{*}	422	explicit specializations of member functions of class templates				
{*}	425	return value of std::get_temporary_buffer				
{*}	426	search_n(), fill_n(), and generate_n() with negative n				
{*}	428	string::erase(iterator) validity				
{*}	432	stringbuf::overflow() makes only one write position available				
{*}	434	bitset::to_string() hard to use				
{*}	435	bug in DR 25				
{*}	436	are cv-qualified facet types valid facets?				
{*}	438	Ambiguity in the "do the right thing" clause				
{*}	441	Is fpos::state const?				
{*}	442	sentry::operator bool() inconsistent signature				
{*}	443	filebuf::close() inconsistent use of EOF				
{*}	444	Bad use of casts in fstream				
{*}	445	iterator_traits::reference unspecified for some iterator categories				
{*}	448	Random Access Iterators over abstract classes				
{*}	449	Library Issue 306 Goes Too Far				
✓	453	basic_stringbuf::seekoff need not always fail for an empty stream	DONE	STDCXX-332		
{*}	455	cerr::tie() and wcerr::tie() are overspecified				
{*}	456	Traditional C header files are overspecified				
{*}	457	bitset constructor: incorrect number of initialized bits				
{*}	460	Default modes missing from basic_fstream member specifications				
{*}	461	time_get hard or impossible to implement				
{*}	464	Suggestion for new member functions in standard containers				
{*}	465	Contents of <ciso646>				
✓	467	char_traits::lt(), compare(), and memcmp()	DONE	STDCXX-214		
{*}	468	unexpected consequences of ios_base::operator void*()				
{*}	469	vector<bool> ill-formed relational operators				
{*}	474	confusing Footnote 297				
{*}	475	May the function object passed to for_each modify the elements of the iterated sequence?				

[*]	478	Should forward iterator requirements table have a line for <code>r->m</code> ?		
[*]	488	<code>rotate</code> throws away useful information		
[*]	495	Clause 22 template parameter requirements		
[*]	496	Illegal use of <code>T</code> in <code>vector<bool></code>		
[*]	497	meaning of <code>numeric_limits::traps</code> for floating point types		
[*]	505	<code>result_type</code> in random distribution requirements		
[*]	507	Missing requirement for <code>variate_generator::operator()</code>		
[*]	508	Bad parameters for <code>ranlux64_base_01</code>		
[*]	518	Are <code>insert</code> and <code>erase</code> stable for <code>unordered_multiset</code> and <code>unordered_multimap</code> ?		
[*]	519	<code>data()</code> undocumented		
[*]	520	<code>result_of</code> and pointers to data members		
[*]	521	Garbled requirements for <code>argument_type</code> in <code>reference_wrapper</code>		
[*]	524	regex named character classes and case-insensitivity don't mix		
[*]	527	<code>tr1::bind</code> has lost its <i>Throws</i> clause		
[*]	530	Must elements of a <code>string</code> be contiguous?		
[*]	531	array forms of unformatted input functions		
[*]	533	typo in 2.2.3.10/1		
[*]	534	Missing <code>basic_string</code> members		
[*]	535	<code>std::string::swap</code> specification poorly worded		
[*]	537	Typos in the signatures in 27.6.1.3/42-43 and 27.6.2.4		
[*]	538	241 again: Does <code>unique_copy()</code> require <code>CopyConstructible</code> and <code>Assignable</code> ?		
[*]	540	<code>shared_ptr<void>::operator*()</code>		
[*]	541	<code>shared_ptr</code> template assignment and void		
[*]	542	<code>shared_ptr</code> observers		
[*]	543	<code>valarray</code> slice default constructor		
[*]	545	When is a deleter deleted?		
[*]	550	What should the return type of <code>pow(float, int)</code> be?		
[*]	551	<code><complex></code>		
[*]	552	<code>random_shuffle</code> and its generator		
✓	559	<code>numeric_limits<const T></code>	DONE	STDCXX-151
[*]	561	<code>inserter</code> overly generic		
[*]	562	<code>stringbuf</code> ctor inefficient		
[*]	563	<code>stringbuf</code> seeking from end		
[*]	566	array forms of unformatted input function undefined for zero-element arrays		
[*]	567	<code>streambuf</code> inserter and extractor should be unformatted		
[*]	574	DR 369 Contradicts Text		
[*]	575	the specification of <code>~shared_ptr</code> is MT-unfriendly, makes implementation assumptions		
[*]	576	<code>find_first_of</code> is overconstrained		
[*]	577	<code>upper_bound(first, last, ...)</code> cannot return last		
[*]	578	purpose of hint to <code>allocator::allocate()</code>		
[*]	581	<code>flush()</code> not unformatted function		
[*]	586	<code>string</code> inserter not a formatted function		
[*]	589	Requirements on iterators of member template functions of containers		
[*]	593	<code>__STDC_CONSTANT_MACROS</code>		
[*]	595	<code>TR1/C++0x: fabs(complex<T>)</code> redundant / wrongly specified		
[*]	596	27.8.1.3 Table 112 omits "a+" and "a+b" modes		
[*]	598	Decimal: Conversion to integral should truncate, not round.		
[*]	599	Decimal: Say "octets" instead of "bytes."		
[*]	600	Decimal: Wrong parameters for <code>wcstod*</code> functions		
[*]	601	Decimal: <code>numeric_limits</code> typos		
[*]	602	Decimal: "generic floating type" not defined.		
[*]	603	Decimal: Trivially simplifying decimal classes.		
[*]	604	Decimal: Storing a reference to a facet unsafe.		
[*]	605	Decimal: <code><decimal.h></code> doesn't live here anymore.		
[*]	607	Concern about short seed vectors		

{*}	608	Unclear seed_seq construction details				
{*}	609	missing static const				
{*}	610	Suggested non-normative note for C++0x				
{*}	611	Standard library templates and incomplete types				
{*}	612	numeric_limits::is_modulo insufficiently defined				
{*}	613	max_digits10 missing from numeric_limits				
{*}	616	missing typename in ctype_byname				
{*}	618	valarray::cshift() effects on empty array				
{*}	619	longjmp wording problem				
{*}	620	valid uses of empty valarrays				
{*}	621	non-const copy assignment operators of helper arrays				
{*}	622	behavior of filebuf dtor and close() on error				
{*}	623	pubimbue forbidden to call imbue()				
{*}	624	valarray assignment and arrays of unequal length				
✓	625	mixed up Effects and Returns clauses	DONE	STDCXX-272		
{*}	628	Inconsistent definition of basic_regex constructor				
{*}	629	complex<T> insertion and locale dependence				
{*}	634	allocator.address() doesn't work for types overloading operator&				
{*}	638	deque end invalidation during erase				
{*}	640	27.6.2.5.2 does not handle (unsigned) long long				
{*}	643	Impossible "as if" clauses				
{*}	644	Possible typos in 'function' description				
{*}	646	const incorrect match_result members				
{*}	650	regex_token_iterator and const correctness				
{*}	651	Missing preconditions for regex_token_iterator ctors				
{*}	652	regex_iterator and const correctness				
{*}	654	Missing IO roundtrip for random number engines				
{*}	655	Signature of generate_canonical not useful				
{*}	660	Missing Bitwise Operations				
{*}	661	New 27.6.1.2.2 changes make special extractions useless				
{*}	664	do_unshift for codecvt<char, char, mbstate_t>				
{*}	665	do_unshift return value				
{*}	666	moneypunct::do_curr_symbol()				
{*}	672	Swappable requirements need updating				
{*}	673	unique_ptr update				
{*}	674	shared_ptr interface changes for consistency with N1856				
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="f03cf1fa-3d04-4479-99ed-f4dc57d17b6a"><ac:plain-text-body><![CDATA[		{*}	[677]	http://tinyurl.com/ytwhht#677]	Weaknesses in seed_seq::randomize [rand.util.seedseq]	C H E C K
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="25e0a770-c168-4ac0-a4ba-d3ebb5ff4471"><ac:plain-text-body><![CDATA[		{*}	[678]	http://tinyurl.com/ytwhht#678]	Changes for [rand.req.eng]	
		{*}	679	resize parameter by value		
		{*}	680	move_iterator operator-> return		
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="ec82e37b-104e-4611-8807-62fcdcf04128"><ac:plain-text-body><![CDATA[		{*}	[681]	http://tinyurl.com/ytwhht#681]	Operator functions impossible to compare are defined in [re.submatch.op]	
✗	682	basic_regex ctor takes InputIterator or ForwardIterator?	TO DO			
✗	687	shared_ptr conversion constructor not constrained	TO DO			
✗	689	reference_wrapper constructor overly constrained	TO DO			
{*}	693	std::bitset::all() missing	CHECK			
{*}	694	std::bitset and long long	CHECK			
✗	695	ctype<char>::classic_table() not accessible	TO DO	STDCXX-996		
✗	699	N2111 changes min/max	TO DO			
{*}	700	N1856 defines struct identity	CHECK			
i	703	map::at() need a complexity specification	Editorial	N/A		
{*}	705	type-trait decay incompletely specified	CHECK			
✗	706	make_pair() should behave as make_tuple() wrt. reference_wrapper()	TO DO			

		712	seed_seq::size no longer useful	TO DO			
{*}		715	minmax_element complexity is too lax				
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="6d4d01a9-c2a7-41c2-8e58-5d4b1db49f92"><ac:plain-text-body><![CDATA[	{*}	722		http://tinyurl.com/ytwhtt#722	Missing [c.math] functions nanf and nanl		]]></ac:plain-text-body></ac:structured-macro>
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="999af249-d013-4b45-82cf-15e01de2d35f"><ac:plain-text-body><![CDATA[	{*}	740		http://tinyurl.com/ytwhtt#740	Please remove *_ptr<T[N]>		]]></ac:plain-text-body></ac:structured-macro>
{*}		743	rvalue swap for shared_ptr				
{*}		744	What is the lifetime of an exception pointed to by an exception_ptr?				
{*}		746	current_exception may fail with bad_alloc				
{*}		749	Currently has_nothrow_copy_constructor<T>::value is true if T has 'a' nothrow copy constructor.				
		755	std::vector and std::string lack explicit shrink-to-fit operations	TO DO	STDCXX-1000		
		759	A reference is not an object	Editorial			
{*}		761	unordered_map needs an <tt>at()</tt> member function				
{*}		766	Inconsistent exception guarantees between ordered and unordered associative containers				
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="7ae3dfb1-631f-4ca3-8a09-50c67cfe4cd6"><ac:plain-text-body><![CDATA[	{*}	768		http://tinyurl.com/ytwhtt#768]	Typos in [atomics]?		]]></ac:plain-text-body></ac:structured-macro>
{*}		770	std::function should use rvalue swap				
		775	Tuple indexing should be unsigned?	TO DO			
{*}		777	Atoms Library Issue				
		778	std::bitset does not have any constructor taking a string literal	DONE			
{*}		781	std::complex should add missing C99 functions				
{*}		782	Extended seed_seq constructor is useless				
{*}		783	thread::id reuse				
{*}		789	xor_combine_engine(result_type) should be explicit				
{*}		792	piecewise_constant_distribution is undefined for a range with just one endpoint				
{*}		798	Refactoring of binders lead to interface breakage				

Legend:

	Codes and Descriptions	
ICON	STATUS	DESCRIPTION
	Editorial	Either an editorial issue or one that requires no code changes.
	DONE	Change fully implemented and exercised by one or more tests.
{*}	CHECK	Further analysis needed or implementation status unknown.
	INCOMPLETE	Change only partially implemented or not exercised by a test.
	TO DO	Change not implemented yet.