

# ISO/IEC TR 19768:2007-11 (E)

## Information technology - Programming languages - Technical Report on C++ Library Extensions

---

### Contents

Page

List of Tables xi	1
1.1 Relation to C++ Standard Library Introduction .....	1
1.2 Categories of extensions .....	1
1.3 Namespaces and headers .....	1
2 General Utilities	3
2.1 Reference wrappers .....	3
2.1.1 Additions to header <functional> synopsis .....	3
2.1.2 Class template reference_wrapper .....	3
2.1.2.1 reference_wrapper construct/copy/destroy .....	4
2.1.2.2 reference_wrapper assignment .....	5
2.1.2.3 reference_wrapper access .....	5
2.1.2.4 reference_wrapper invocation .....	5
2.1.2.5 reference_wrapper helper functions .....	5
2.2 Smart pointers .....	6
2.2.1 Additions to header <memory> synopsis .....	6
2.2.2 Class bad_weak_ptr .....	7
2.2.3 Class template shared_ptr .....	7
2.2.3.1 shared_ptr constructors .....	9
2.2.3.2 shared_ptr destructor .....	10
2.2.3.3 shared_ptr assignment .....	10
2.2.3.4 shared_ptr modifiers .....	11
2.2.3.5 shared_ptr observers .....	11
2.2.3.6 shared_ptr comparison .....	12
2.2.3.7 shared_ptr I/O .....	13
2.2.3.8 shared_ptr specialized algorithms .....	13
2.2.3.9 shared_ptr casts .....	13
2.2.3.10 get_deleter .....	14
2.2.4 Class template weak_ptr .....	14
2.2.4.1 weak_ptr constructors .....	15
2.2.4.2 weak_ptr destructor .....	15
2.2.4.3 weak_ptr assignment .....	16
2.2.4.4 weak_ptr modifiers .....	16
2.2.4.5 weak_ptr observers .....	16
2.2.4.6 weak_ptr comparison .....	17
2.2.4.7 weak_ptr specialized algorithms .....	17
2.2.5 Class template enable_shared_from_this .....	17
3 Function objects	21
3.1 Definitions .....	21
3.2 Additions to <functional> synopsis .....	21
3.3 Requirements .....	22
3.4 Function return types .....	23
3.5 Function template mem_fn .....	24
3.6 Function object binders .....	24
3.6.1 Class template is_bind_expression .....	24
3.6.2 Class template is_placeholder .....	25
3.6.3 Function template bind .....	25
3.6.4 Placeholders .....	26
3.7 Polymorphic function wrappers .....	26
3.7.1 Class bad_function_call .....	26
3.7.1.1 bad_function_call constructor .....	27

3.7.2	Class template function .....	27
3.7.2.1	function construct/copy/destroy .....	28
3.7.2.2	function modifiers .....	30
3.7.2.3	function capacity .....	30
3.7.2.4	function invocation .....	30
3.7.2.5	function target access .....	30
3.7.2.6	undefined operators .....	31
3.7.2.7	null pointer comparison operators .....	31
3.7.2.8	specialized algorithms .....	31
4	Metaprogramming and type traits 33 4.1 Requirements .....	33
4.2	Header <type_traits> synopsis .....	33
4.3	Helper classes .....	35
4.4	General Requirements .....	35
4.5	Unary Type Traits .....	36
4.5.1	Primary Type Categories .....	36
4.5.2	Composite type traits .....	37
4.5.3	Type properties .....	37
4.6	Relationships between types .....	39
4.7	Transformations between types .....	40
4.7.1	Const-volatile modifications .....	40
4.7.2	Reference modifications .....	41
4.7.3	Array modifications .....	41
v	CONTENTS CONTENTS 4.7.4 Pointer modifications .....	42
4.8	Other transformations .....	42
4.9	Implementation requirements .....	42
5	Numerical facilities 45 5.1 Random number generation .....	45
5.1.1	Requirements .....	45
5.1.2	Header <random> synopsis .....	49
5.1.3	Class template variate_generator .....	51
5.1.4	Random number engine class templates .....	52
5.1.4.1	Class template linear_congruential .....	53
5.1.4.2	Class template mersenne_twister .....	54
5.1.4.3	Class template subtract_with_carry .....	55
5.1.4.4	Class template subtract_with_carry_01 .....	57
5.1.4.5	Class template discard_block .....	58
5.1.4.6	Class template xor_combine .....	59
5.1.5	Engines with predefined parameters .....	61
5.1.6	Class random_device .....	62
5.1.7	Random distribution class templates .....	63
5.1.7.1	Class template uniform_int .....	63
5.1.7.2	Class bernoulli_distribution .....	64
5.1.7.3	Class template geometric_distribution .....	65
5.1.7.4	Class template poisson_distribution .....	65
5.1.7.5	Class template binomial_distribution .....	66
5.1.7.6	Class template uniform_real .....	67
5.1.7.7	Class template exponential_distribution .....	67
5.1.7.8	Class template normal_distribution .....	68
5.1.7.9	Class template gamma_distribution .....	69
5.2	Mathematical special functions .....	70
5.2.1	Additions to header <cmath> synopsis .....	70
5.2.1.1	associated Laguerre polynomials .....	73
5.2.1.2	associated Legendre functions .....	73
5.2.1.3	beta function .....	74
5.2.1.4	(complete) elliptic integral of the first kind .....	74
5.2.1.5	(complete) elliptic integral of the second kind .....	74
5.2.1.6	(complete) elliptic integral of the third kind .....	75
5.2.1.7	confluent hypergeometric functions .....	75
5.2.1.8	regular modified cylindrical Bessel functions .....	75
5.2.1.9	cylindrical Bessel functions (of the first kind) .....	75

5.2.1.10	irregular modified cylindrical Bessel functions .....	76
5.2.1.11	cylindrical Neumann functions .....	76
5.2.1.12	(incomplete) elliptic integral of the first kind .....	77
5.2.1.13	(incomplete) elliptic integral of the second kind .....	77
5.2.1.14	(incomplete) elliptic integral of the third kind .....	77
5.2.1.15	exponential integral .....	77
5.2.1.16	Hermite polynomials .....	78
<b>CONTENTS CONTENTS vi 5.2.1.17 hypergeometric functions .....</b>		<b>78</b>
5.2.1.18	Laguerre polynomials .....	78
5.2.1.19	Legendre polynomials .....	79
5.2.1.20	Riemann zeta function .....	79
5.2.1.21	spherical Bessel functions (of the first kind) .....	79
5.2.1.22	spherical associated Legendre functions .....	80
5.2.1.23	spherical Neumann functions .....	80
5.2.2	Additions to header <math.h> synopsis .....	80
<b>6</b>	<b>Containers 81 6.1 Tuple types .....</b>	<b>81</b>
6.1.1	Header <tuple> synopsis .....	81
6.1.2	Additions to header <utility> synopsis .....	82
6.1.3	Class template tuple .....	83
6.1.3.1	Construction .....	83
6.1.3.2	Tuple creation functions .....	84
6.1.3.3	Tuple helper classes .....	85
6.1.3.4	Element access .....	86
6.1.3.5	Relational operators .....	86
6.1.4	Pairs .....	87
6.2	Fixed size array .....	88
6.2.1	Header <array> synopsis .....	88
6.2.2	Class template array .....	88
6.2.2.1	array constructors, copy, and assignment .....	90
6.2.2.2	array specialized algorithms .....	90
6.2.2.3	array size .....	90
6.2.2.4	Zero sized arrays .....	90
6.2.2.5	Tuple interface to class template array .....	90
6.3	Unordered associative containers .....	91
6.3.1	Unordered associative container requirements .....	91
6.3.1.1	Exception safety guarantees .....	96
6.3.2	Additions to header <functional> synopsis .....	97
6.3.3	Class template hash .....	97
6.3.4	Unordered associative container classes .....	98
6.3.4.1	Header <unordered_set> synopsis .....	98
6.3.4.2	Header <unordered_map> synopsis .....	98
6.3.4.3	Class template unordered_set .....	99
6.3.4.3.1	unordered_set constructors .....	101
6.3.4.3.2	unordered_set swap .....	102
6.3.4.4	Class template unordered_map .....	102
6.3.4.4.1	unordered_map constructors .....	104
6.3.4.4.2	unordered_map element access .....	105
6.3.4.4.3	unordered_map swap .....	105
6.3.4.5	Class template unordered_multiset .....	105
6.3.4.5.1	unordered_multiset constructors .....	107
6.3.4.5.2	unordered_multiset swap .....	108
vii	<b>CONTENTS CONTENTS 6.3.4.6 Class template unordered_multimap .....</b>	<b>108</b>
6.3.4.6.1	unordered_multimap constructors .....	110
6.3.4.6.2	unordered_multimap swap .....	111
<b>7</b>	<b>Regular expressions 113 7.1 Definitions .....</b>	<b>113</b>
7.2	Requirements .....	113
7.3	Regular expressions summary .....	115
7.4	Header <regex> synopsis .....	115

7.5	Namespace <code>tr1::regex_constants</code> .....	121
7.5.1	Bitmask Type <code>syntax_option_type</code> .....	121
7.5.2	Bitmask Type <code>regex_constants::match_flag_type</code> .....	122
7.5.3	Implementation defined <code>error_type</code> .....	124
7.6	Class <code>regex_error</code> .....	125
7.7	Class template <code>regex_traits</code> .....	125
7.8	Class template <code>basic_regex</code> .....	127
7.8.1	<code>basic_regex</code> constants .....	129
7.8.2	<code>basic_regex</code> constructors .....	129
7.8.3	<code>basic_regex</code> <code>assign</code> .....	131
7.8.4	<code>basic_regex</code> constant operations .....	131
7.8.5	<code>basic_regex</code> locale .....	132
7.8.6	<code>basic_regex</code> swap .....	132
7.8.7	<code>basic_regex</code> non-member functions .....	132
7.8.7.1	<code>basic_regex</code> non-member swap .....	132
7.9	Class template <code>sub_match</code> .....	132
7.9.1	<code>sub_match</code> members .....	133
7.9.2	<code>sub_match</code> non-member operators .....	133
7.10	Class template <code>match_results</code> .....	138
7.10.1	<code>match_results</code> constructors .....	139
7.10.2	<code>match_results</code> size .....	140
7.10.3	<code>match_results</code> element access .....	140
7.10.4	<code>match_results</code> formatting .....	141
7.10.5	<code>match_results</code> allocator .....	141
7.10.6	<code>match_results</code> swap .....	142
7.11	Regular expression algorithms .....	142
7.11.1	exceptions .....	142
7.11.2	<code>regex_match</code> .....	142
7.11.3	<code>regex_search</code> .....	144
7.11.4	<code>regex_replace</code> .....	145
7.12	Regular expression Iterators .....	146
7.12.1	Class template <code>regex_iterator</code> .....	146
7.12.1.1	<code>regex_iterator</code> constructors .....	147
7.12.1.2	<code>regex_iterator</code> comparisons .....	147
7.12.1.3	<code>regex_iterator</code> dereference .....	148
7.12.1.4	<code>regex_iterator</code> increment .....	148
7.12.2	Class template <code>regex_token_iterator</code> .....	149
<b>CONTENTS CONTENTS viii 7.12.2.1 <code>regex_token_iterator</code> constructors</b> .....		<b>150</b>
7.12.2.2	<code>regex_token_iterator</code> comparisons .....	151
7.12.2.3	<code>regex_token_iterator</code> dereference .....	151
7.12.2.4	<code>regex_token_iterator</code> increment .....	151
7.13	Modified ECMAScript regular expression grammar .....	152
8	C compatibility 155 8.1 Additions to header <code>&lt;complex&gt;</code> .....	155
8.1.1	Synopsis .....	155
8.1.2	Function <code>acos</code> .....	155
8.1.3	Function <code>asin</code> .....	155
8.1.4	Function <code>atan</code> .....	156
8.1.5	Function <code>acosh</code> .....	156
8.1.6	Function <code>asinh</code> .....	156
8.1.7	Function <code>atanh</code> .....	156
8.1.8	Function <code>fabs</code> .....	156
8.1.9	Additional Overloads .....	156
8.2	Header <code>&lt;ccomplex&gt;</code> .....	157
8.3	Header <code>&lt;complex.h&gt;</code> .....	157
8.4	Additions to header <code>&lt;cctype&gt;</code> .....	157
8.4.1	Synopsis .....	157
8.4.2	Function <code>isblank</code> .....	157
8.5	Additions to header <code>&lt;ctype.h&gt;</code> .....	157
8.6	Header <code>&lt;cfenv&gt;</code> .....	157
8.6.1	Synopsis .....	157

8.6.2	Definitions .....	158
8.7	Header <fenv.h> .....	158
8.8	Additions to header <float> .....	158
8.9	Additions to header <float.h> .....	158
8.10	Additions to header <ios> .....	158
8.10.1	Synopsis .....	158
8.10.2	Function hexfloat .....	158
8.11	Header <inttypes> .....	159
8.11.1	Synopsis .....	159
8.11.2	Definitions .....	159
8.12	Header <inttypes.h> .....	159
8.13	Additions to header <climits> .....	160
8.14	Additions to header <limits.h> .....	160
8.15	Additions to header <locale> .....	160
8.16	Additions to header <cmath> .....	160
8.16.1	Synopsis .....	160
8.16.2	Definitions .....	164
8.16.3	Function template definitions .....	164
8.16.4	Additional overloads .....	165
8.17	Additions to header <math.h> .....	166
8.18	Additions to header <csdarg> .....	166
ix	CONTENTS CONTENTS 8.19 Additions to header <stdarg.h> .....	166
8.20	The header <csdbool> .....	166
8.21	The header <stdbool.h> .....	166
8.22	The header <csdint> .....	166
8.22.1	Synopsis .....	166
8.22.2	Definitions .....	167
8.23	The header <stdint.h> .....	167
8.24	Additions to header <csdio> .....	167
8.24.1	Synopsis .....	167
8.24.2	Definitions .....	168
8.24.3	Additional format specifiers .....	168
8.24.4	Additions to header <stdio.h> .....	168
8.25	Additions to header <csdlib> .....	168
8.25.1	Synopsis .....	168
8.25.2	Definitions .....	169
8.25.3	Function abs .....	169
8.25.4	Function div .....	169
8.26	Additions to header <stdlib.h> .....	169
8.27	Header <ctgmath> .....	169
8.28	Header <tgmath.h> .....	169
8.29	Additions to header <ctime> .....	169
8.30	Additions to header <cwchar> .....	169
8.30.1	Synopsis .....	169
8.30.2	Definitions .....	170
8.30.3	Additional wide format specifiers .....	170
8.31	Additions to header <wchar.h> .....	170
8.32	Additions to header <wctype> .....	170
8.32.1	Synopsis .....	170
8.32.2	Function iswblank .....	170
8.33	Additions to header <wctype.h> .....	170